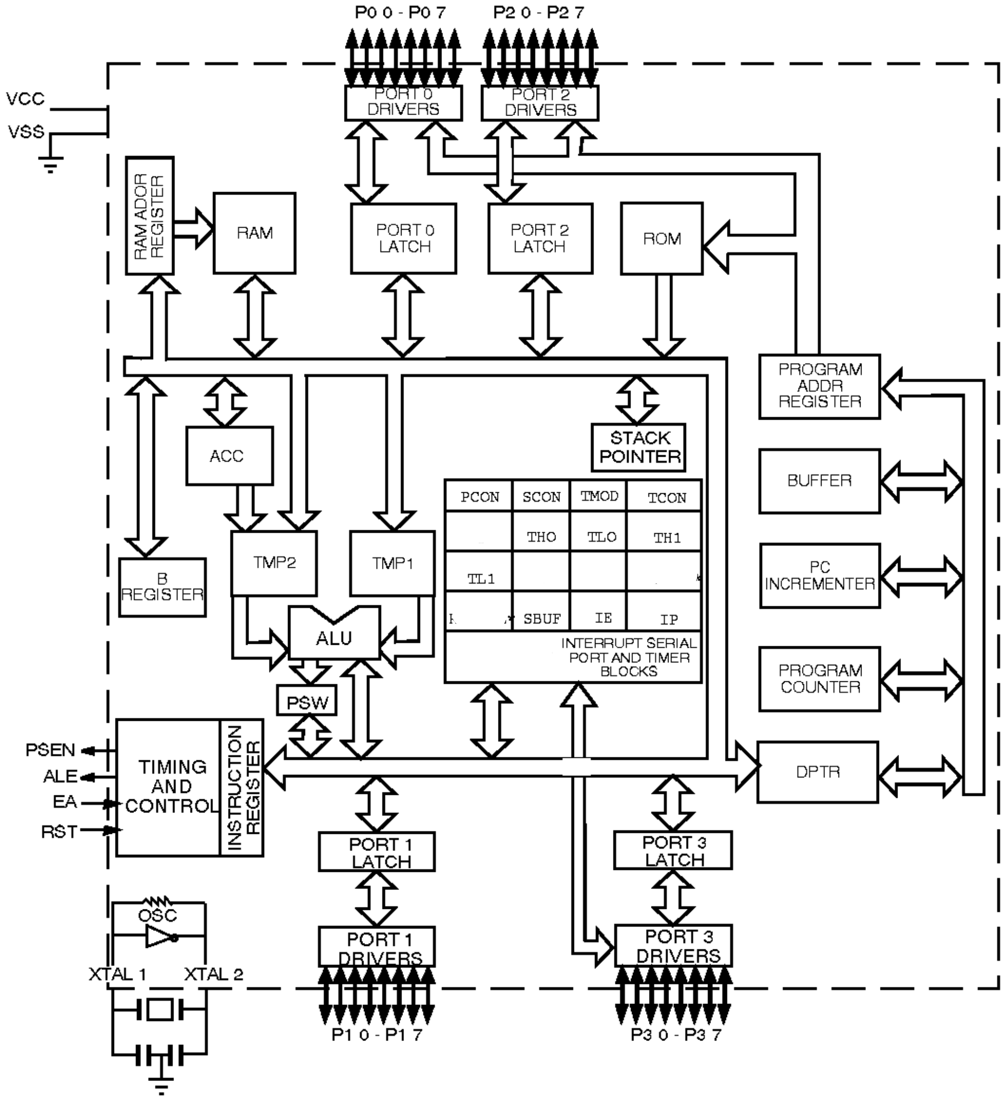
Figure 2-1. 8051 Architecture Block Diagram



Note: (*)For Timer 2 only.

Figure 2-1 shows a functional block diagram of the 80C51s.

2.2 Special Function Registers

A map of the on-chip memory area called SFR (Special Function Register) space is shown in Figure 2-1. SFRs marked by parentheses are resident in the microcontroller which have the Timer2 feature.Note that not all of the addresses are occupied. Read accesses to these addresses will in general return random data.

Table 2-1. SFR Map

	Bit Addressable	**8 Bytes Non-bit Addressable**							
F8h	▨	▨	▨	▨	▨	▨	▨	▨	FFh
F0h	B	▨	▨	▨	▨	▨	▨	▨	F7h
E8h	▨	▨	▨	▨	▨	▨	▨	▨	EFh
E0h	ACC	▨	▨	▨	▨	▨	▨	▨	E7h
D8h	▨	▨	▨	▨	▨	▨	▨	▨	DFh
D0h	PSW	▨	▨	▨	▨	▨	▨	▨	D7h
C8h		▨					▨	▨	CFh
C0h	▨	▨	▨	▨	▨	▨	▨	▨	C7h
B8h	IP	▨	▨	▨	▨	▨	▨	▨	BFh
B0h	P3	▨	▨	▨	▨	▨	▨	▨	B7h
A8h	IE	▨	▨	▨	▨	▨	▨	▨	AFh
A0h	P2	▨	▨	▨	▨	▨	▨	▨	A7h
98h	SCON	SBUF	▨	▨	▨	▨	▨	▨	9Fh
90h	P1	▨	▨	▨	▨	▨	▨	▨	97h
88h	TCON	TMOD	TL0	TL1	TH0	TH1			8Fh
80h	P0	SP	DPL	DPH	▨	▨	▨	PCON	87h
	0/8	**1/9**	**2/A**	**3/B**	**4/C**	**5/D**	**6/E**	**7/F**	

Note: Reserved ▨

User software should not write to the reserved locations, since they may be used in derivative Atmel 8051 products to invoke new features. The functions of the SFRs are described as below.

2.2.1 Accumulator

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

2.2.2 B Register

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

2.2.3 Program Status Word

The PSW register contains program status information as detailed in Table 2-2.

Table 2-2. PSW: Program Status Word 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	
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)		

2.2.4 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.

2.2.5 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.

2.2.6 Ports 0 to 3

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

2.2.7 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.

2.2.8 Timer Registers

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

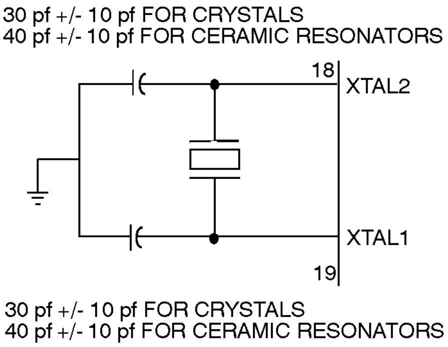
2.2.10 Control Registers

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.

2.3 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 2-2. The on-chip circuitry, and selection of off-chip components to configure the oscillator are discussed in Section 1.12.

Figure 2-2. 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

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.

2.3.1 More about the On-chip Oscillator

This section not yet available.

2.4 CPU Timing

2.4.1 X1 Mode (Standard Mode)

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.

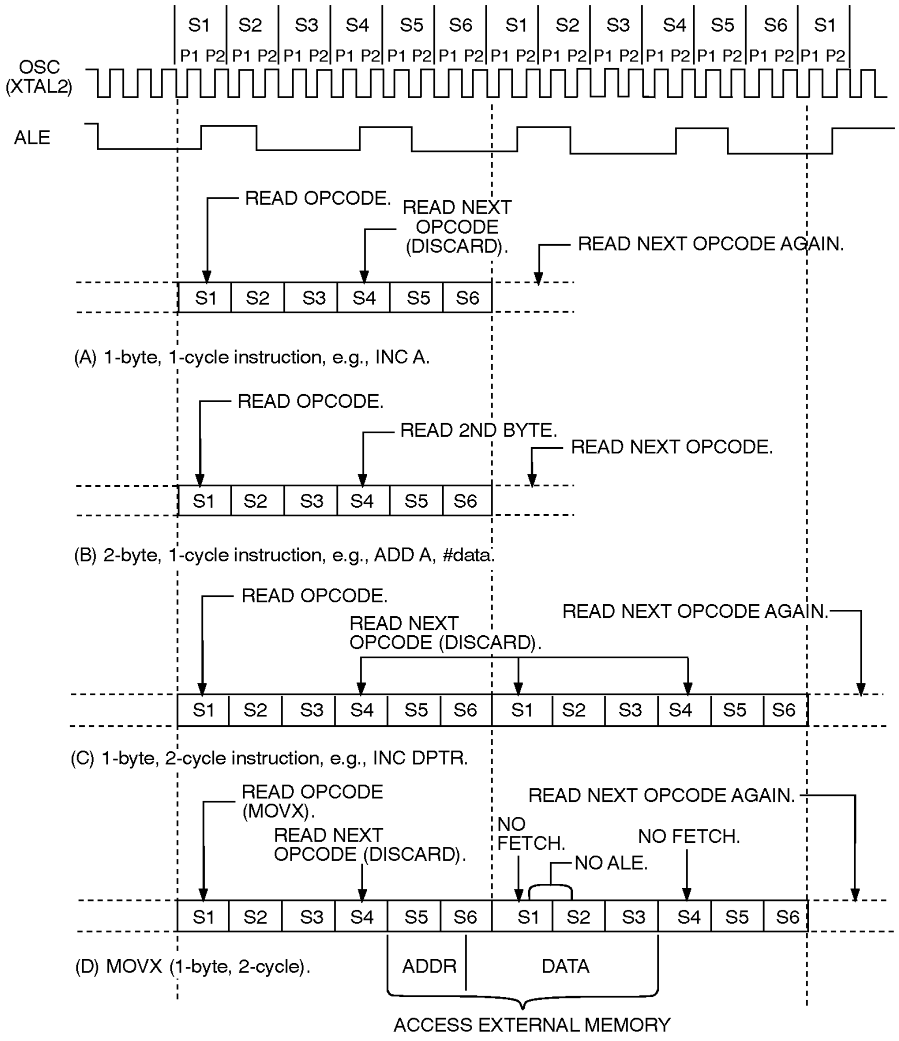
The diagrams in Figure 2-3 show the fetch/execute timing referenced to the internal states and phases. Since these internal clock signals are not user accessible, the XTAL2 oscillator signal and the ALE (Address Latch Enable) signal are shown for external reference. ALE is normally activated twice during 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. Figure 2-3A and Figure 2-3B 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.

Separately, 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 2-3C and Figure 2-3D show the timing for a normal 1-byte, 2-cycle instruction and for a MOVX instruction.

Figure 2-3. 80C51 fetch/Execute Sequences.


S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1
P1 P2
OSC (XTAL2)
ALE
READ OPCODE.
READ NEXT OPCODE (DISCARD).
READ NEXT OPCODE AGAIN.
(A) 1-byte, 1-cycle instruction, e.g., INC A.
READ OPCODE.
READ 2ND BYTE.
READ NEXT OPCODE.
(B) 2-byte, 1-cycle instruction, e.g., ADD A, #data.
READ OPCODE.
READ NEXT OPCODE (DISCARD).
READ NEXT OPCODE AGAIN.
(C) 1-byte, 2-cycle instruction, e.g., INC DPTR.
READ OPCODE (MOVX).
READ NEXT OPCODE (DISCARD).
NO FETCH.
NO ALE.
NO FETCH.
READ NEXT OPCODE AGAIN.
(D) MOVX (1-byte, 2-cycle).
ADDR
DATA
ACCESS EXTERNAL MEMORY

2.4.2 X2 Mode

This section not yet available.

2.5 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.

All the Port 3 pins, and (in the case of Timer2) 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
[1]P1.0	T2 (Timer/Counter 2 external input)
[1]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)

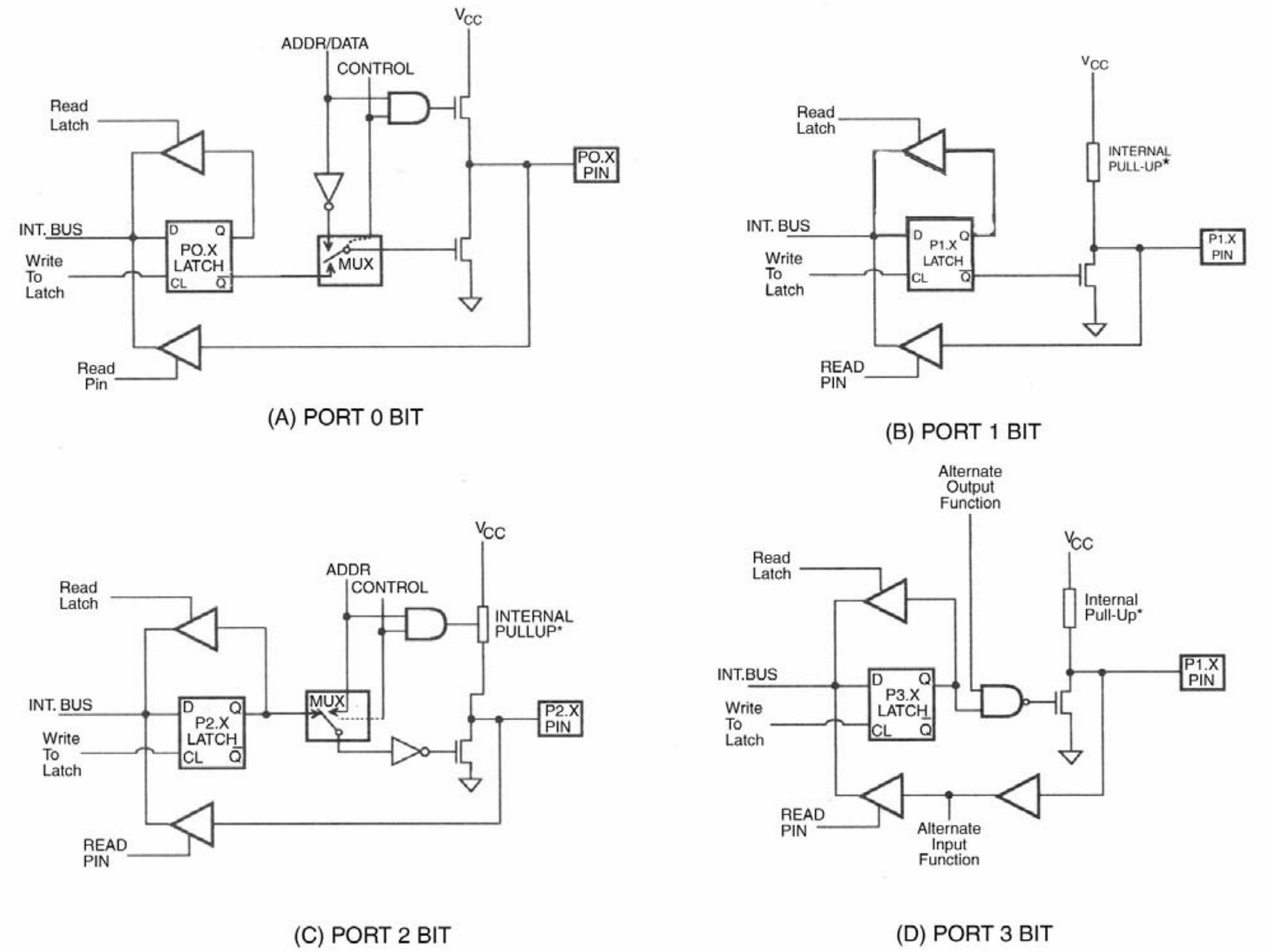
Note: 1. P1.0 and P1.1 serve these alternate functions only on the microcontrollers including Timer2.

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.

2.5.1 I/O Configurations

Figure 2-4 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 2-4. 80C51 Port Bit Latches and I/O Buffers.



(A) PORT 0 BIT

(B) PORT 1 BIT

(C) PORT 2 BIT

(D) PORT 3 BIT

As shown in Figure 2-4, 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 2-4, 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 pull-ups. The pull-up FET in the P0 output driver (see Figure 2-4A) is used only when the Port is emitting 1’s during external memory accesses. Otherwise the pull-up 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 pull-ups 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 1’s written to them by the reset function. If a 0 is subsequently written to a port latch, it can be re configured as an input by writing a 1 to it.

2.5.2 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.

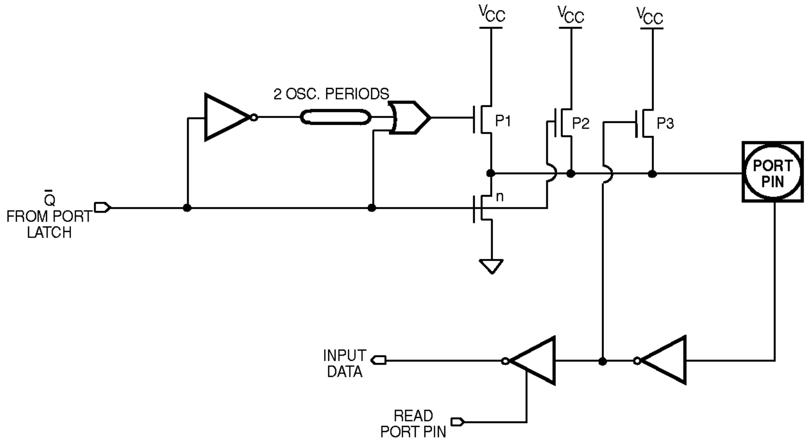
If the change requires a 0-to-1 transition in Port 1, 2, or 3, an additional pull-up is turned on during S1P1 and S1P2 of the cycle in which the transition occurs. This is done to increase the transition speed. The extra pull-up can source about 100 times the current that the normal pull-up can. It should be noted that the internal pull-ups are field-effect transistors, not linear resistors. The pull-up arrangements are shown in Figure 2-5.

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 2-5 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 strength 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 2-5. Ports 1 and 3 CMOS Internal Pull-up Configurations.



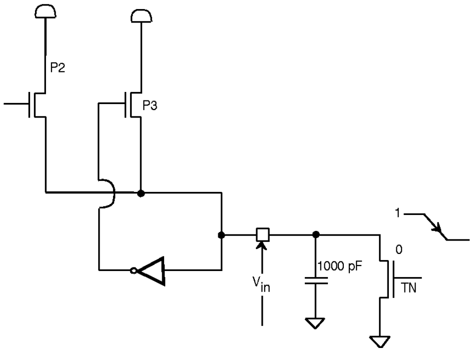
CMOS Configuration. pFET 1 is turned on for 2 osc. periods after Q makes a 1-to-0 transition. During this time, pFET 1 also turns on pFET 3 through the inverter to form a latch which holds the 1. pFET 2 is also on.

Port 2 is similar except that it holds the strong pull-up on while emitting 1s that are address bits. (See “Accessing External Memory”.)

2.5.3 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 pull-up P3, leaving only the weak pull-up P2 to drive the transistor. Figure 2-6 shows an example where the port is driven by an open drain transistor t_N. The parasitic capacitance is equal to 1000pF.

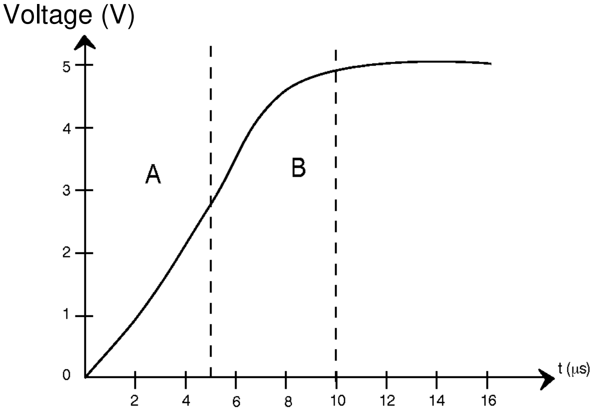
Figure 2-6. Port Interfacing



The above diagram show the behavior of the port during 0 to 1 transition.

In the area A only pull-up P2 sinks the capacitor and takes 5 µs to switch from 0 volt to 2 volts. In the area B, pull-up 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 2-7. Port Behavior During 0-to-1 Transition



2.5.4 Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Which instructions perform what functions? 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.

2.6 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 $\overline{PSEN}$ (program store enable) as the read strobe. Accesses to external Data Memory use $\overline{RD}$ or $\overline{WR}$ (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 pull-ups during the entire time that they are emitting address bits that are 1’s. 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 1’s, 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 $\overline{WR}$ is activated, and remains there until after $\overline{WR}$ is deactivated. In a read cycle, the incoming byte is accepted at Port 0 just before the read strobe is deactivated.

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.

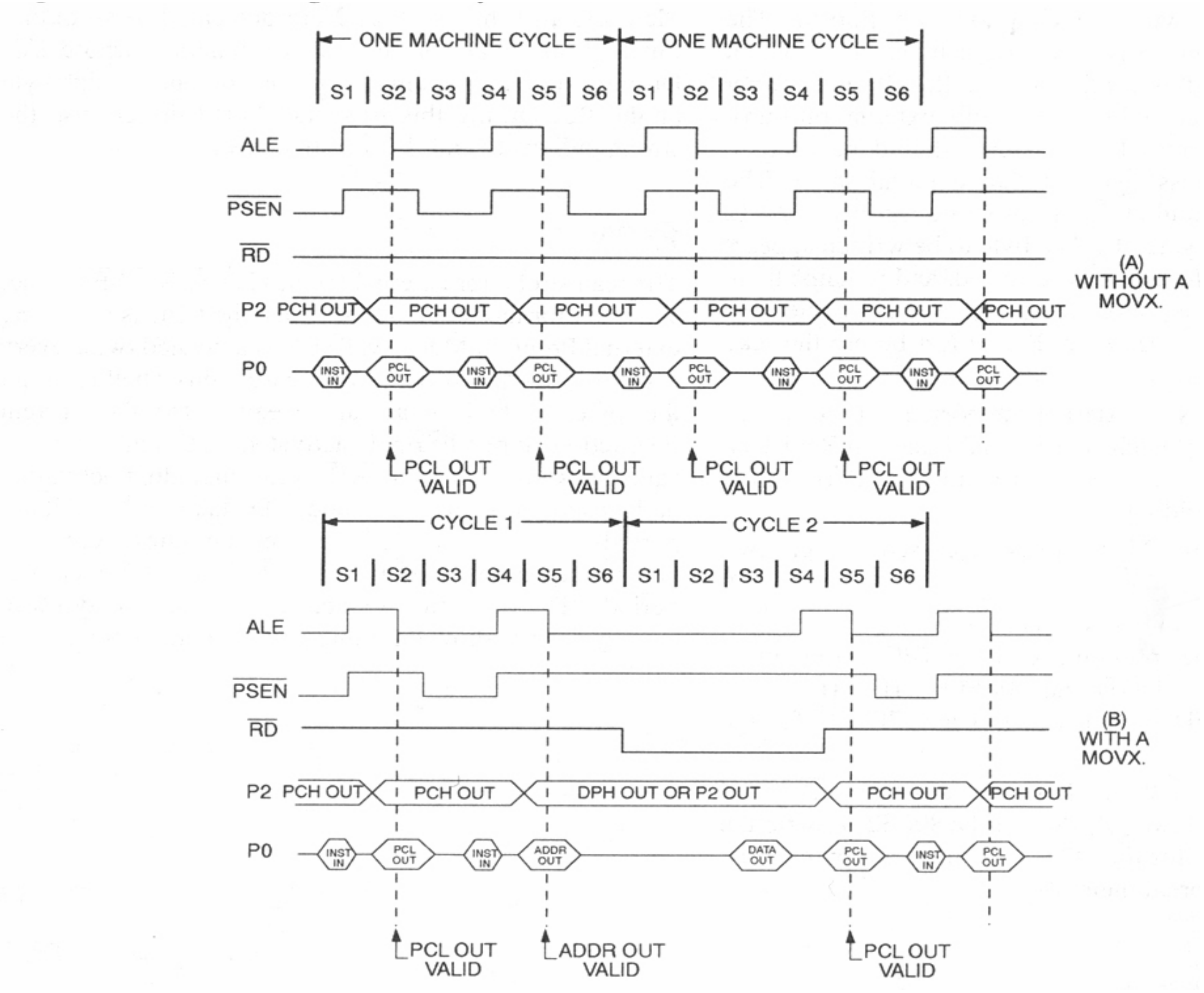
External program Memory is accessed under two conditions:

1. Whenever signal $\overline{EA}$ is active or
2. Whenever the program counter (PC) contains a number that is larger than the memory size.

This requires that the ROMless versions have $\overline{EA}$ wired low to enable the lower 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 pull-ups to emit PC bits that are 1's.

Figure 2-8. External Program Memory Execution



2.7 PSEN

The read strobe for external fetches is PSEN. PSEN is not activated for internal fetches. When the CPU is accessing external Program Memory, 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 PSEN is activated its timing is not the same as RD. A complete RD cycle, including activation and deactivation of ALE and RD, takes 12 oscillator periods. A complete PSEN cycle, including activation and deactivation of ALE and PSEN, takes 6 oscillator periods. The execution sequence for these two types of read cycles are shown in Figure 2-8 for comparison.

2.8 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. The ALE disable mode, described in Section 2.8.2, disables the ALE output. 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.

2.8.1 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 $\overline{\text{PSEN}}$ and $\overline{\text{RD}}$. A positive-logic AND of these two signals produces an active-low read strobe that can be used for the combined physical memory. Since the $\overline{\text{PSEN}}$ cycle is faster than the $\overline{\text{RD}}$ cycle, the external memory needs to be fast enough to accommodate the $\overline{\text{PSEN}}$ cycle.

2.8.2 ALE Disable Mode

The ALE signal is used to demultiplex address and data buses on port 0 when used with external program or data memory. Nevertheless, during internal code execution, ALE signal is still generated.

In order to reduce EMI, ALE signal can be disabled by setting AO bit.

The AO bit is located in AUXR register at bit location 0 (See Table 2-3). As soon as AO is set, ALE is no longer output but remains active during MOVX and MOVC instructions and external fetches. During ALE disabling, ALE pin is weakly pulled high.

Table 2-3. AUXR Register
AUXR (S:8Eh)
Auxiliary Register

7	6	5	4	3	2	1	0
-	-	-	-	-	-	EXTRAM	AO

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	EXTRAM	**EXTRAM select** Clear to map XRAM data in internal XRAM memory. Set to map XRAM data in external XRAM memory.
0	AO	**ALE Output bit** Clear to restore ALE operation during internal fetches. Set to disable ALE operation during internal fetches.

Reset Value = XXXX XX00b

2.9 Timer/Counters

The Atmel 80C51 Microcontrollers implement two general purpose, 16-bit timers/counters. They are identified as Timer 0 and Timer 1, and can be independently configured to operate in a variety of modes as a timer or as an event counter. When operating as a timer, the timer/counter runs for a programmed length of time, then issues an interrupt request. When operating as a counter, the timer/counter counts negative transitions on an external pin. After a preset number of counts, the counter issues an interrupt request.
The various operating modes of each timer/counter are described in the following sections.

2.9.1 Timer/Counter Operations

A basic operation consists of timer registers THx and TLx (x= 0, 1) connected in cascade to form a 16-bit timer. Setting the run control bit (TRx) in TCON register (see Figure 2-4) turns the timer on by allowing the selected input to increment TLx. When TLx overflows it increments THx; when THx overflows it sets the timer overflow flag (TFx) in TCON register. Setting the TRx does not clear the THx and TLx timer registers. Timer registers can be accessed to obtain the current count or to enter preset values. They can be read at any time but TRx bit must be cleared to preset their values, otherwise the behavior of the timer/counter is unpredictable.

The C/Tx# control bit (in TCON register) selects timer operation, or counter operation, by selecting the divided-down peripheral clock or external pin Tx as the source for the counted signal. TRx bit must be cleared when changing the mode of operation, otherwise the behavior of the timer/counter is unpredictable.

For timer operation (C/Tx# = 0), the timer register counts the divided-down peripheral clock. The timer register is incremented once every peripheral cycle (6 peripheral clock periods). The timer clock rate is F_{PER} / 6, i.e. F_{OSC} / 12 in standard mode or F_{OSC} / 6 in X2 mode.

For counter operation (C/Tx# = 1), the timer register counts the negative transitions on the Tx external input pin. The external input is sampled every peripheral cycle. When the sample is high in one cycle and low in the next one, the counter is incremented. Since it takes 2 cycles (12 peripheral clock periods) to recognize a negative transition, the maximum count rate is F_{PER} / 12, i.e. F_{OSC} / 24 in standard mode or F_{OSC} / 12 in X2 mode. 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 peripheral cycle.

In addition to the “timer” or “counter” selection, Timer 0 and Timer 1 have four operating modes from which to select which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both timer/counters. Mode 3 is different. The four operating modes are described below.

Timer 2, has three modes of operation: ‘capture’, ‘auto-reload’ and ‘baud rate generator’.

2.10 Timer 0

Timer 0 functions as either a timer or event counter in four modes of operation. Figure 2-9 to Figure 2-12 show the logical configuration of each mode.

Timer 0 is controlled by the four lower bits of the TMOD register (see Table 2-5) and bits 0, 1, 4 and 5 of the TCON register (see Table 2-4). TMOD register selects the method of timer gating (GATE0), timer or counter operation (T/C0#) and mode of operation (M10 and M00). The TCON register provides timer 0 control functions: overflow flag (TF0), run control bit (TR0), interrupt flag (IE0) and interrupt type control bit (IT0).

For normal timer operation (GATE0= 0), setting TR0 allows TL0 to be incremented by the selected input. Setting GATE0 and TR0 allows external pin INT0# to control timer operation.

Timer 0 overflow (count rolls over from all 1s to all 0s) sets TF0 flag, generating an interrupt request.
It is important to stop timer/counter before changing mode.

2.10.1 Mode 0 (13-bit Timer)

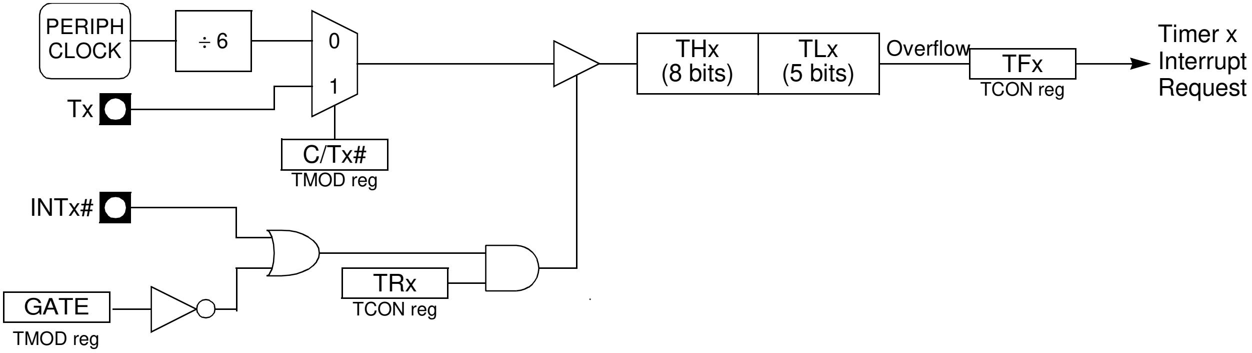
Mode 0 configures timer 0 as a 13-bit timer which is set up as an 8-bit timer (TH0 register) with a modulo 32 prescaler implemented with the lower five bits of the TL0 register (see Figure 2-9). The upper three bits of TL0 register are indeterminate and should be ignored. Prescaler overflow increments the TH0 register.

As the count rolls over from all 1's to all 0's, it sets the timer interrupt flag TF0. The counted input is enabled to the Timer when TR0 = 1 and either GATE = 0 or $\overline{\text{INT0}}$ = 1. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{\text{INT0}}$, to facilitate pulse width measurements). TR0 is a control bit in the Special Function register TCON (Table 2-4). GATE is in TMOD.

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

Mode 0 operation is the same for Timer 0 as for Timer 1. Substitute TR0, TF0 and $\overline{\text{INT0}}$ for the corresponding Timer 1 signals in Table 2-10. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

Figure 2-9. Timer/Counter x (x = 0 or 1) in Mode 0

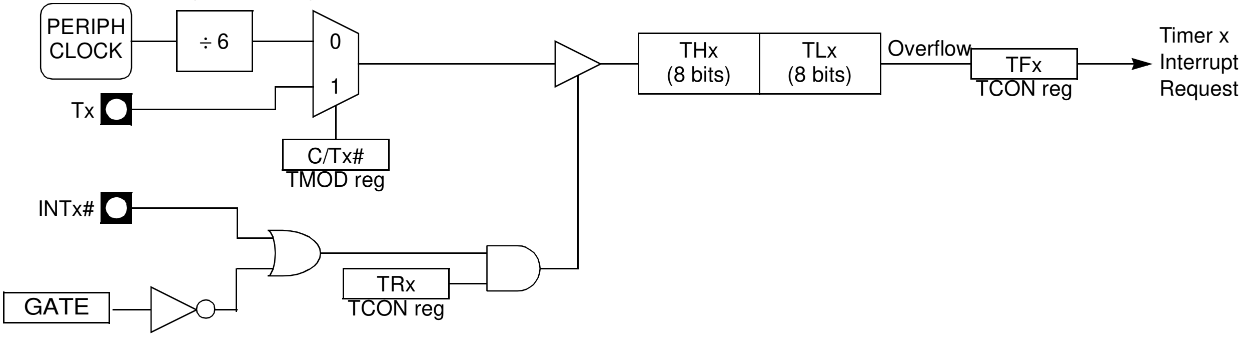


2.10.2 Mode 1 (16-bit Timer)

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

Mode 1 configures timer 0 as a 16-bit timer with the TH0 and TL0 registers connected in cascade (see Figure 2-10). The selected input increments the TL0 register.

Figure 2-10. Timer/Counter x (x = 0 or 1) in Mode 1

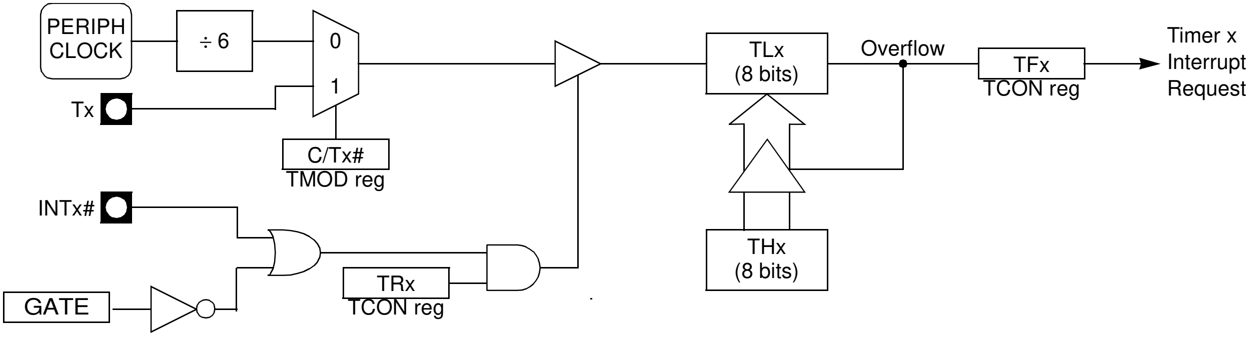


2.10.3 Mode 2 (8-bit Timer with Auto-Reload)

Mode 2 configures timer 0 as an 8-bit timer (TL0 register) that automatically reloads from the TH0 register (see Table 2-6 on page 37). TL0 overflow sets TF0 flag in the TCON register and reloads TL0 with the contents of TH0, which is preset by software. When the interrupt request is serviced, hardware clears TF0. The reload leaves TH0 unchanged. The next reload value may be changed at any time by writing it to the TH0 register.

Mode 2 operation is the same for Timer/Counter 1.

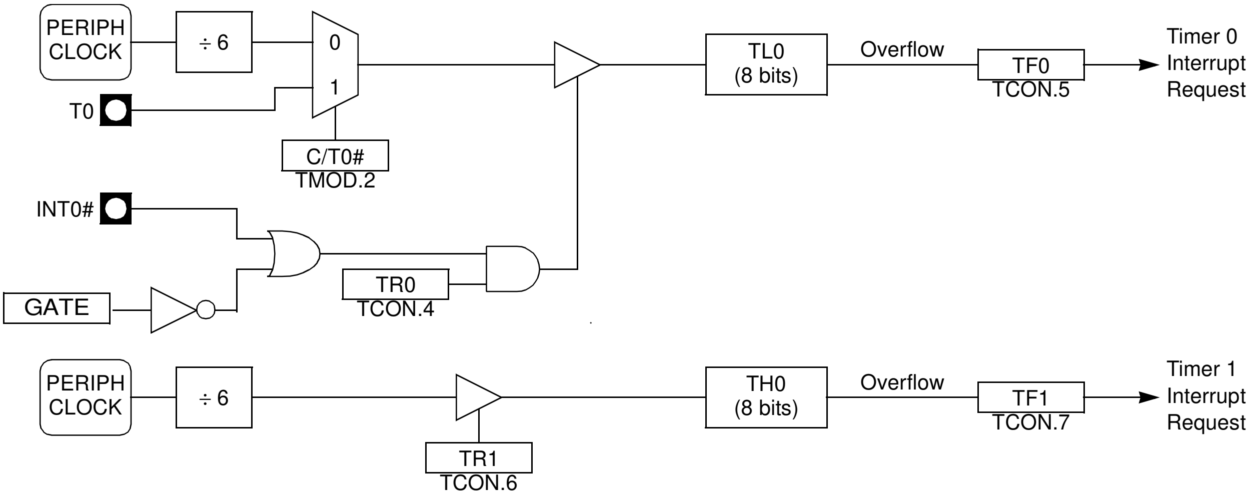
Figure 2-11. Timer/Counter x (x = 0 or 1) in Mode 2



2.10.4 Mode 3 (Two 8-bit Timers)

Mode 3 configures timer 0 so that registers TL0 and TH0 operate as separate 8-bit timers (see Figure 2-12). This mode is provided for applications requiring an additional 8-bit timer or counter. TL0 uses the timer 0 control bits C/T0# and GATE0 in the TMOD register, and TR0 and TF0 in the TCON register in the normal manner. TH0 is locked into a timer function (counting F_{PER} /6) and takes over use of the timer 1 interrupt (TF1) and run control (TR1) bits. Thus, operation of timer 1 is restricted when timer 0 is in mode 3.

Figure 2-12. Timer/Counter 0 in Mode 3: Two 8-bit Counters



2.11 Timer 1

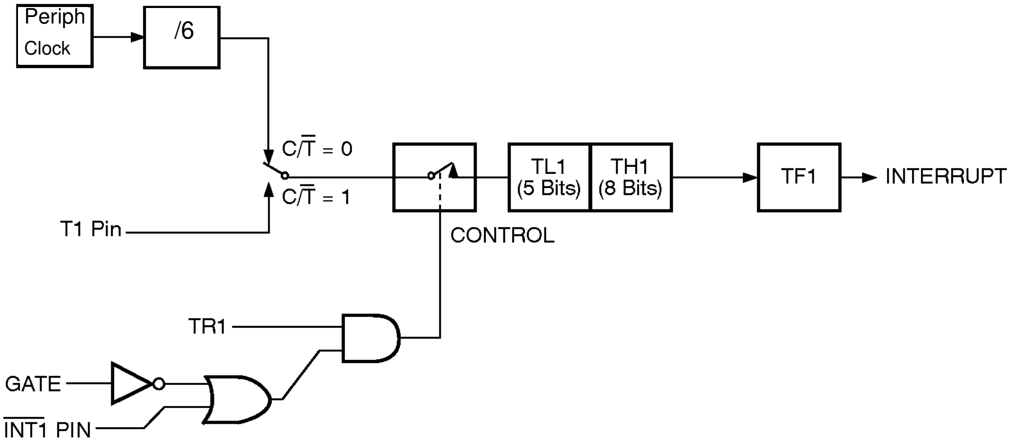
Timer 1 is identical to timer 0, except for mode 3, which is a hold-count mode. The following comments help to understand the differences:

- Timer 1 functions as either a timer or event counter in three modes of operation. Figure 2-9 to Figure 2-11 show the logical configuration for modes 0, 1, and 2. Timer 1's mode 3 is a hold-count mode.
- Timer 1 is controlled by the four high-order bits of the TMOD register (see Table 2-5 on page 37) and bits 2, 3, 6 and 7 of the TCON register (see Table 2-4 on page 36). The TMOD register selects the method of timer gating (GATE1), timer or counter operation (C/T1#) and mode of operation (M11 and M01). The TCON register provides timer 1 control functions: overflow flag (TF1), run control bit (TR1), interrupt flag (IE1) and interrupt type control bit (IT1).
- Timer 1 can serve as the baud rate generator for the serial port. Mode 2 is best suited for this purpose.
- For normal timer operation (GATE1 = 0), setting TR1 allows TL1 to be incremented by the selected input. Setting GATE1 and TR1 allows external pin INT1# to control timer operation.
- Timer 1 overflow (count rolls over from all 1s to all 0s) sets the TF1 flag generating an interrupt request.
- When timer 0 is in mode 3, it uses timer 1's overflow flag (TF1) and run control bit (TR1). For this situation, use timer 1 only for applications that do not require an interrupt (such as a baud rate generator for the serial port) and switch timer 1 in and out of mode 3 to turn it off and on.
- It is important to stop timer/counter before changing modes.

2.11.1 Mode 0 (13-bit Timer)

Mode 0 configures Timer 1 as a 13-bit timer, which is set up as an 8-bit timer (TH1 register) with a modulo-32 prescaler implemented with the lower 5 bits of the TL1 register (see Figure 2-9). The upper 3 bits of the TL1 register are ignored. Prescaler overflow increments the TH1 register.

Figure 2-13. Timer/Counter 1 Mode 0: 13-bit Counter



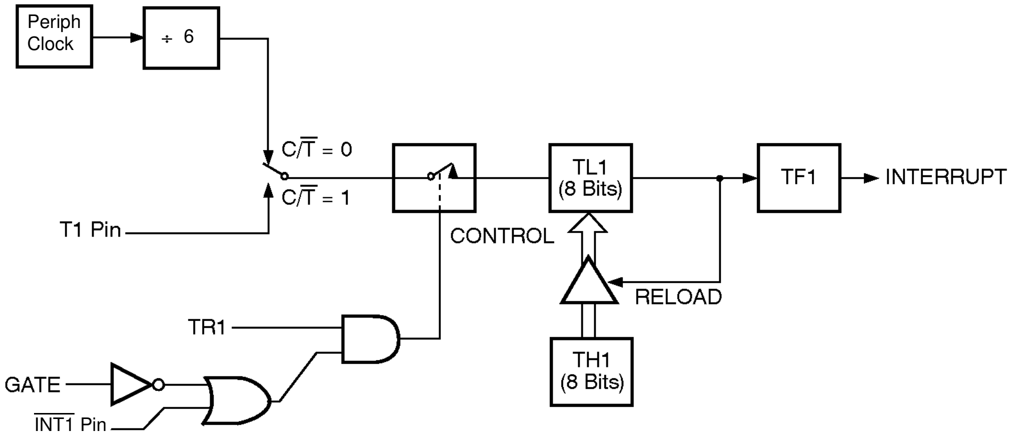
2.11.2 Mode 1 (16-bit Timer)

Mode 1 configures Timer 1 as a 16-bit timer with the TH1 and TL1 registers connected in cascade (see Figure 2-10). The selected input increments the TL1 register.

2.11.3 Mode 2 (8-bit Timer with Auto Reload)

Mode 2 configures Timer 1 as an 8-bit timer (TL1 register) with automatic reload from the TH1 register on overflow (see Figure 2-11). TL1 overflow sets the TF1 flag in the TCON register and reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

Figure 2-14. Timer/Counter 1 Mode 2: 8-bit Auto-reload



2.11.4 Mode 3 (Halt)

Placing Timer 1 in mode 3 causes it to halt and hold its count. This can be used to halt Timer 1 when TR1 run control bit is not available i.e., when Timer 0 is in mode 3.

2.11.5 Interrupt

Each timer handles one interrupt source; that is the timer overflow flag TF0 or TF1. This flag is set every time an overflow occurs. Flags are cleared when vectoring to the timer interrupt routine. Interrupts are enabled by setting ETx bit in IE0 register. This assumes interrupts are globally enabled by setting EA bit in the IE0 register.

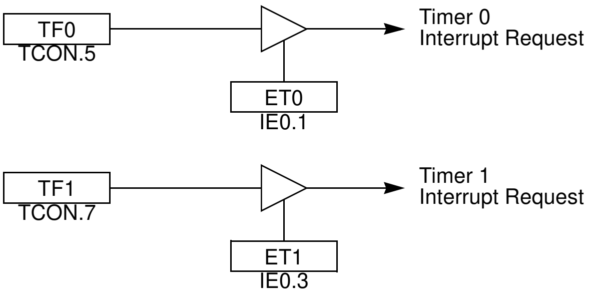


Figure 2-15. Timer Interrupt System

2.11.6 Timer Registers

Table 2-4. TCON Register - *TCON (S:88h)*
Timer/Counter Control Register.

7	6	5	4	3	2	1	0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**

Bit Number	Bit Mnemonic	Description
7	TF1	**Timer 1 Overflow Flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on timer/counter overflow, when the timer 1 register overflows.
6	TR1	**Timer 1 Run Control Bit** Clear to turn off timer/counter 1. Set to turn on timer/counter 1.
5	TF0	**Timer 0 Overflow Flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on timer/counter overflow, when the timer 0 register overflows.
4	TR0	**Timer 0 Run Control Bit** Clear to turn off timer/counter 0. Set to turn on timer/counter 0.
3	IE1	**Interrupt 1 Edge Flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT1). Set by hardware when external interrupt is detected on INT1# pin.
2	IT1	**Interrupt 1 Type Control Bit** Clear to select low level active (level triggered) for external interrupt 1 (INT1#). Set to select falling edge active (edge triggered) for external interrupt 1.
1	IE0	**Interrupt 0 Edge Flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT0). Set by hardware when external interrupt is detected on INT0# pin.
0	IT0	**Interrupt 0 Type Control Bit** Clear to select low level active (level triggered) for external interrupt 0 (INT0#). Set to select falling edge active (edge triggered) for external interrupt 0.

Reset Value = 0000 0000b

Table 2-5. TMOD Register - *TMOD (S:89h)*
Timer/Counter Mode Control Register

7	6	5	4	3	2	1	0
GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00

Bit Number	Bit Mnemonic	Description
7	GATE1	**Timer 1 Gating Control Bit** Clear to enable timer 1 whenever the TR1 bit is set. Set to enable timer 1 only while the INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select Bit** Clear for timer operation: timer 1 counts the divided-down system clock. Set for Counter operation: timer 1 counts negative transitions on external pin T1.
5	M11	**Timer 1 Mode Select Bits** M11 M01 Operating mode 0 0 Mode 0: 8-bit timer/counter (TH1) with 5-bit prescaler (TL1). 0 1 Mode 1: 16-bit timer/counter. 1 0 Mode 2: 8-bit auto-reload timer/counter (TL1). Reloaded from TH1 at overflow. 1 1 Mode 3: timer 1 halted. Retains count.
4	M01	
3	GATE0	**Timer 0 Gating Control Bit** Clear to enable timer 0 whenever the TR0 bit is set. Set to enable timer/counter 0 only while the INT0# pin is high and the TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select Bit** Clear for timer operation: timer 0 counts the divided-down system clock. Set for counter operation: timer 0 counts negative transitions on external pin T0.
1	M10	**Timer 0 Mode Select Bit** M10 M00 Operating mode 0 0 Mode 0: 8-bit timer/counter (TH0) with 5-bit prescaler (TL0). 0 1 Mode 1: 16-bit timer/counter. 1 0 Mode 2: 8-bit auto-reload timer/counter (TL0). Reloaded from TH0 at overflow. 1 1 Mode 3: TL0 is an 8-bit timer/counter. TH0 is an 8-bit timer using timer 1's TR0 and TF0 bits.
0	M00	

Reset Value = 0000 0000b

Table 2-6. TH0 Register - TH0 (S:8Ch)
Timer 0 High Byte Register.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 0.**

Reset Value = 0000 0000b

Table 2-7. TL0 Register - *TL0 (S:8Ah)*
Timer 0 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 0.**

Reset Value = 0000 0000b

Table 2-8. TH1 Register - *TH1 (S:8Dh)*
Timer 1 High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 1.**

Reset Value = 0000 0000b

Table 2-9. TL1 Register - *TL1 (S:8Bh)*
Timer 1 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 1.**

Reset Value = 0000 0000b

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.

2.13 Serial Interface

It provides both synchronous and asynchronous communication modes. It operates as a Universal Asynchronous Receiver and Transmitter (UART) in three full-duplex modes (Modes 1, 2 and 3). Asynchronous transmission and reception can occur simultaneously and at different baud rates.

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

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.

Serial I/O port includes the following enhancements:

- Framing error detection
- Automatic address recognition

The serial port control and status register is the Special Function Register SCON, shown in Table 2-17. 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).

Table 2-17. SCON Register

SCON - Serial Control Register (98h)

7	6	5	4	3	2	1	0
FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Bit Number	Bit Mnemonic	Description
7	FE	**Framing Error bit (SMOD0=1**) Clear to reset the error state, not cleared by a valid stop bit. Set by hardware when an invalid stop bit is detected. SMOD0 must be set to enable access to the FE bit
	SM0	**Serial port Mode bit 0** Refer to SM1 for serial port mode selection. SMOD0 must be cleared to enable access to the SM0 bit
6	SM1	**Serial port Mode bit 1** **SM0** **SM1** **Mode** **Description** **Baud Rate** 0 0 0 Shift Register $F_{CPU\ PERIPH}/6$ 0 1 1 8-bit UART Variable 1 0 2 9-bit UART $F_{CPU\ PERIPH}$ /32 or /16 1 1 3 39-bit UART Variable
5	SM2	Serial port Mode 2 bit / Multiprocessor Communication Enable bit Clear to disable multiprocessor communication feature. Set to enable multiprocessor communication feature in mode 2 and 3, and eventually mode 1. This bit should be cleared in mode 0.
4	REN	**Reception Enable bit** Clear to disable serial reception. Set to enable serial reception.
3	TB8	Transmitter Bit 8 / Ninth bit to transmit in modes 2 and 3 o transmit a logic 0 in the 9th bit. Set to transmit a logic 1 in the 9th bit.
2	RB8	**Receiver Bit 8 / Ninth bit received in modes 2 and 3** Cleared by hardware if 9th bit received is a logic 0. Set by hardware if 9th bit received is a logic 1. In mode 1, if SM2 = 0, RB8 is the received stop bit. In mode 0 RB8 is not used.
1	TI	**Transmit Interrupt flag** Clear to acknowledge interrupt. 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.
0	RI	**Receive Interrupt flag** Clear to acknowledge interrupt. Set by hardware at the end of the 8th bit time in mode 0, see Figure 2-26. and Figure 2-27. in the other modes.

Reset Value = 0000 0000b
Bit addressable

2.13.4 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:

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

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 2-21 lists various commonly used baud rates and how they can be obtained from Timer 1.

Figure 2-21. Timer 1 Generated Commonly Used Baud Rates

Fosc (MHz)	11.0592	12	14.7456	16	20	SMOD
Baudrate						
150	40h	30h	00h			0
300	A0h	98h	80h	75h	52h	0
600	D0h	CCh	C0h	BBh	A9h	0
1200	E8h	E6h	E0h	DEh	D5h	0
2400	F4h	F3h	F0h	EFh	EAh	0
4800		F3h	EFh	EFh		1
4800	FAh		F8h		F5h	0
9600	FDh		FCh			0
9600					F5h	1
19200	FDh		FCh			1
38400			FEh			
76800			FFh			

2.13.6 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 2-24 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, 1's shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift 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.

2.13.7 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 microcontroller having Timer 2 feature, 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 2-25 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit and receive.

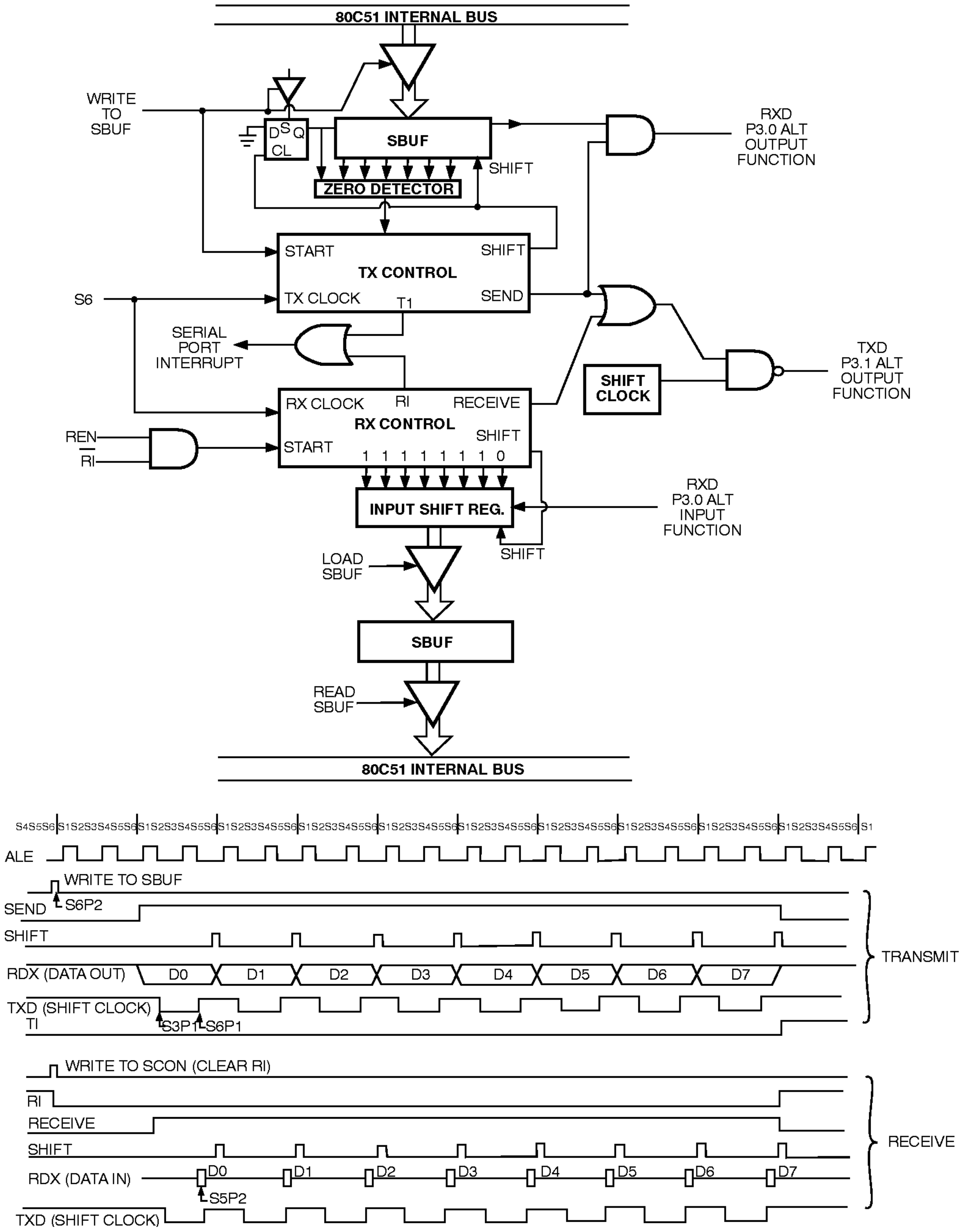
Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal).

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

As data bits shift out to the right, 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 rollover after "write to SBUF".

Reception is initiated by a detected 1-to-0 transition at RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 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 2-24. Serial Port Mode 0



UART Registers

Table 2-22. SCON Register

SCON - Serial Control Register (98h)

7	6	5	4	3	2	1	0
FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Bit Number	Bit Mnemonic	Description
7	FE	**Framing Error bit (SMOD0=1**) Clear to reset the error state, not cleared by a valid stop bit. Set by hardware when an invalid stop bit is detected. SMOD0 must be set to enable access to the FE bit
	SM0	**Serial port Mode bit 0** Refer to SM1 for serial port mode selection. SMOD0 must be cleared to enable access to the SM0 bit
6	SM1	Serial port Mode bit 1 SM0 SM1 Mode Description Baud Rate 0 0 0 Shift Register $F_{CPU\ PERIPH}/6$ 0 1 1 8-bit UART Variable 1 0 2 9-bit UART $F_{CPU\ PERIPH}$/32 or /16 1 1 3 9-bit UART Variable
5	SM2	**Serial port Mode 2 bit / Multiprocessor Communication Enable bit** Clear to disable multiprocessor communication feature. Set to enable multiprocessor communication feature in mode 2 and 3, and eventually mode 1. This bit should be cleared in mode 0.
4	REN	**Reception Enable bit** Clear to disable serial reception. Set to enable serial reception.
3	TB8	Transmitter Bit 8 / Ninth bit to transmit in modes 2 and 3. o transmit a logic 0 in the 9th bit. Set to transmit a logic 1 in the 9th bit.
2	RB8	**Receiver Bit 8 / Ninth bit received in modes 2 and 3** Cleared by hardware if 9th bit received is a logic 0. Set by hardware if 9th bit received is a logic 1. In mode 1, if SM2 = 0, RB8 is the received stop bit. In mode 0 RB8 is not used.
1	TI	**Transmit Interrupt flag** Clear to acknowledge interrupt. 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.
0	RI	**Receive Interrupt flag** Clear to acknowledge interrupt. Set by hardware at the end of the 8th bit time in mode 0, see Figure 2-26. and Figure 2-27. in the other modes.

Reset Value = 0000 0000b
Bit addressable

Table 2-28. PCON Register

PCON - Power Control Register (87h)

7	6	5	4	3	2	1	0
SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Serial port Mode bit 1 for UART** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0	Serial port Mode bit 0 for UART Cleared to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	POF	**Power-Off Flag** Cleared to recognize next reset type. Set by hardware when VCC rises from 0 to its nominal voltage. Can also be set by software.
3	GF1	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
2	GF0	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
1	PD	**Power-Down mode bit** Cleared by hardware when reset occurs. Set to enter power-down mode.
0	IDL	**Idle mode bit** Cleared by hardware when interrupt or reset occurs. Set to enter idle mode.

Reset Value = 00X1 0000b
Not bit addressable

Power-off flag reset value will be 1 only after a power on (cold reset). A warm reset doesn't affect the value of this bit.

2.16 Interrupts

If two requests of different priority levels are received simultaneously, 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:

Table 2-30. Interrupt Priority Level

	Source	Priority Within Level
1	IE0	(highest)
2	TF0	
3	IE1	
4	TF1	
5	RI + TI	
		(lowest)

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

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
[illegible]	

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.

2.16.2 External Interrupts

The external sources can be programmed to be level-activated or transition-activated by setting or clearing bit IT1 or ITO in Register TCON.

If ITx = O, 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 el se another interrupt will be generated.

2.16.3 Response Time

The $\overline{\text{INTO}}$ and $\overline{\text{INT1}}$ levels are inverted and latched into IEO 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 29. 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