Introduction

This chapter presents a comprehensive description of the on-chip hardware features of Atmel's Flash-based microcontrollers. Included in this description are the following items.

- 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 and Low Power Idle

The devices under consideration are listed in Table 1.

Figure 1 shows a functional block diagram of the AT89 Series microcontrollers.

Table 1. Atmel's Flash Microcontrollers

Device Name	Program Memory	Data Memory Bytes	16-bit Timers	Technology
AT89C1051	1K Flash	64 RAM	1	CMOS
AT89C2051	2K Flash	128 RAM	2	CMOS
AT89C51	4K Flash	128 RAM	2	CMOS
AT89C52	8K Flash	256 RAM	3	CMOS
AT89C55	20K Flash	256 RAM	3	CMOS
AT89S8252	8K Flash	256 RAM 2K EEPROM	3	CMOS
AT89S53	12K Flash	256 RAM	3	CMOS

Special Function Registers

A map of the on-chip memory area called Special Function Register (SFR) space is shown in Figure 2. SFRs marked by parentheses are resident in the AT89C52 but not in the AT89C51.

AT89 Series Hardware Description

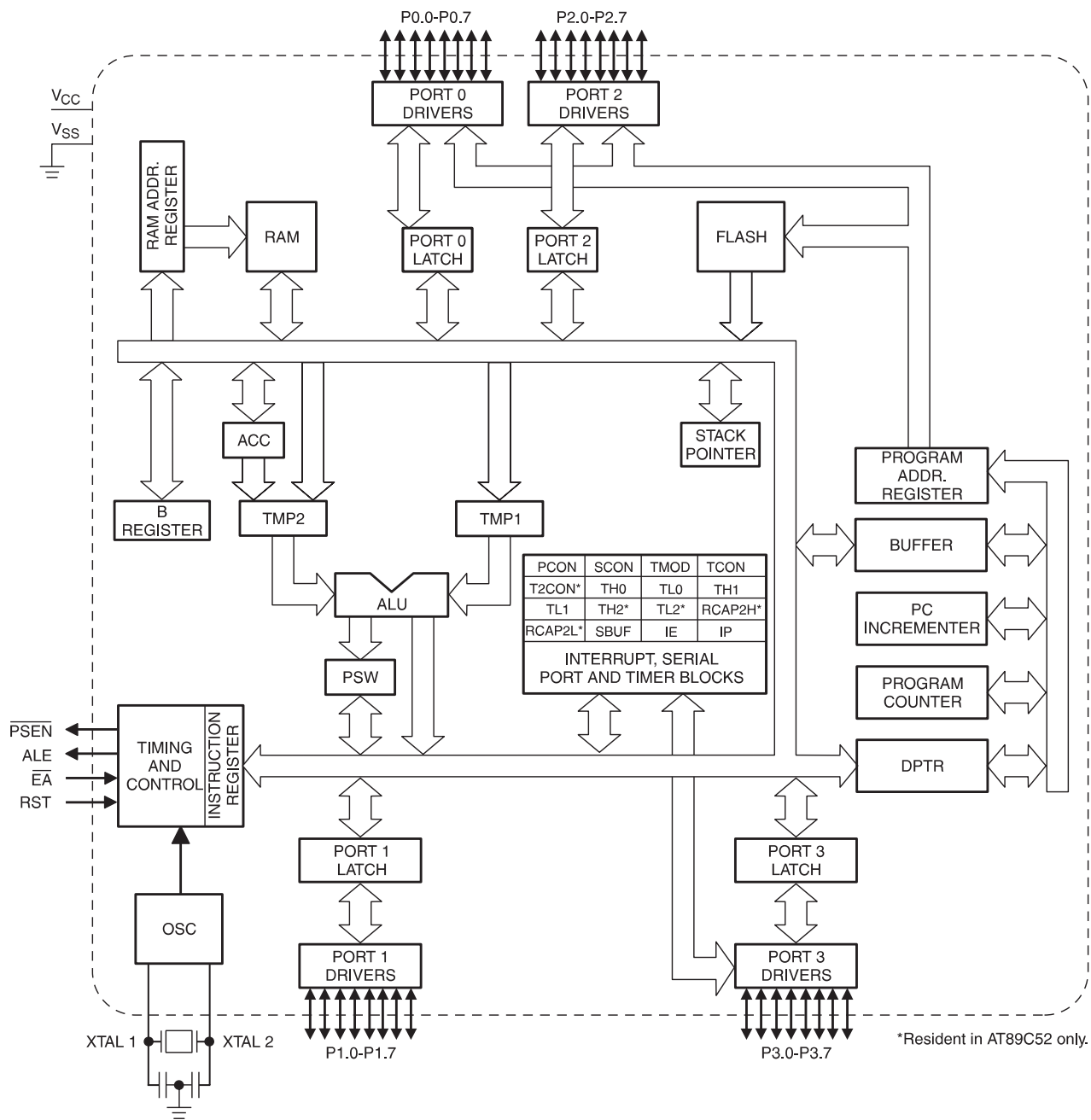**Figure 1.** AT89 Series Flash-Based Microcontroller Architectural Block Diagram

Hardware Description

Figure 2. SFR Map. (...) Indicates Resident in AT89C52, not in AT89C51.

8 Bytes

F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	(T2CON)	(T2MOD)	(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

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

User software should not write 1s to these unimplemented locations, since they may be used in future microcontrollers 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 outlined in the following sections.

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 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 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 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 Counter registers for Timer/Counters 0, 1, and 2, respectively.

Capture Registers

The register pair (RCAP2H, RCAP2L) are the Capture registers for the Timer 2 Capture Mode. In this mode, in

response to a transition at the AT89C52's T2EX pin, TH2 and TL2 are copied into RCAP2H and RCAP2L. Timer 2 also has a l6-bit auto-reload mode, and RCAP2H and RCAP2L hold the reload value for this mode.

Control Registers

Special Function Registers IP, IE, TMOD, TCON, T2CON, T2MOD, 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 of this chapter.

Figure 3. PSW: Program Status Word Register

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

Symbol	Position	Name and Significance	Symbol	Position	Name and Significance
CY	PSW.7	Carry flag.	OV	PSW.2	Overflow flag.
AC	PSW.6	Auxillary Carry flag. (For BCD operations.)	—	PSW.1	User definable flag.
F0	PSW.5	Flag 0. (Available to the user for general purposes.)	P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of 1 bits in the Accumulator, that is, even parity.
RS1	PSW.4	Register bank select control bits 1 and 0.			
RS0	PSW.3	Set/cleared by software to determine working register bank.[1]	Note:	1.	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)

Figure 4. AT89C51 and AT89C52 Port Bit Latches and I/O Buffers

*See Figure 5 for details of the internal pullup.

A. PORT 0 BIT

B. PORT 1 BIT

C. PORT 2 BIT

D. PORT 3 BIT

Hardware Description
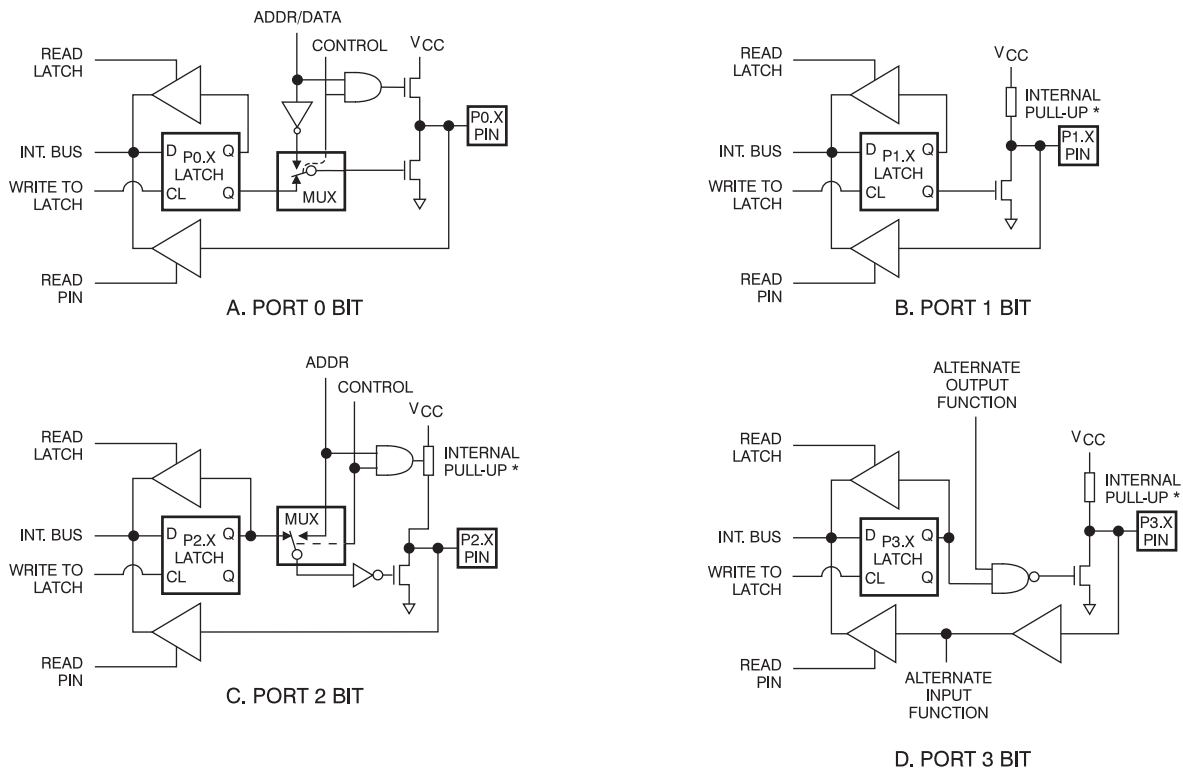
Port Structures and Operation

All four ports in the AT89C51 and AT89C52 are bidirectional. Each consists of a latch (Special Function Registers P0 through P3), an output driver, and an input buffer.

The output drivers of Ports 0 and 2, and the 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 two Port 1 pins (in the AT89C52) are multifunctional. They are not only port pins, but also provide the special features listed in the following table.

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

Note: 1. P1.0 and P1.1 serve these alternate functions only on the AT89C52.

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 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 clocks 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 pin" signal.

As shown in Figure 4, the output drivers of Ports 0 and 2 can be switched 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 1s are written to the P0 SFR.

If a P3 bit latch contains a 1, then the output level is controlled by the alternate output function signal, as shown in Figure 4. The actual P3.X pin level is always available to the pin's alternate input function.

Ports 1, 2, and 3 have internal pullups. Port 0 has open drain outputs. Each I/O line can be used independently 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 pullup but can be pulled low by an external source.

Port 0 has no internal pullups. The FET pullup in the P0 output driver (see Figure 4) is used only when the Port emits 1s during external memory accesses. Otherwise, the FET pullup is off. Consequently, P0 lines that are used as output port lines are open drain. Writing a 1 to the bit latch leaves both FET outputs off, so the pin floats. In this condition, it can be used as a high-impedance input.
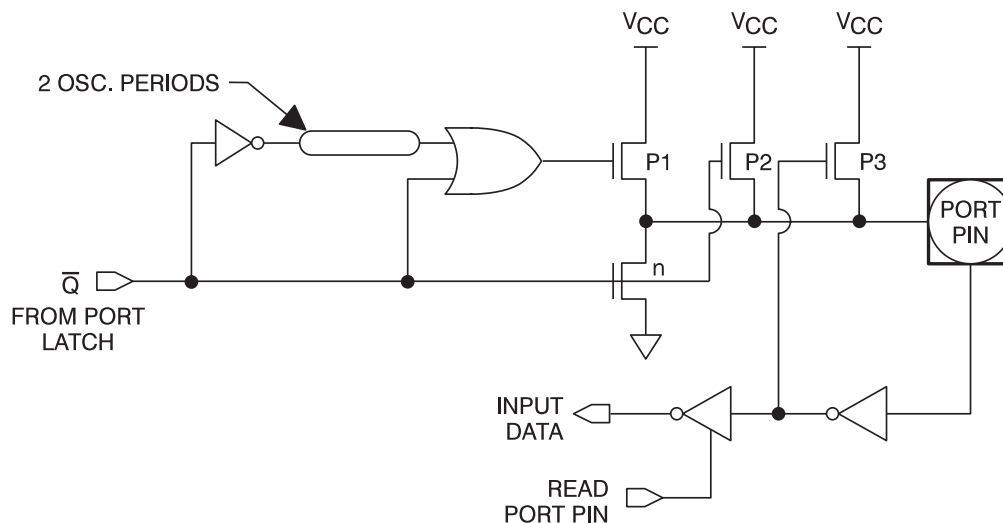
Because Ports 1, 2, and 3 have fixed internal pullups, they are sometimes called quasi-bidirectional ports. When configured as inputs, they pull high and source current (I_{IL}) when externally pulled low. Port 0, on the other hand, is considered truly bidirectional, because it floats when configured as an input.

The reset function writes 1s to all the port latches in the AT89C51and AT89C52. If a 0 is subsequently written to a port latch, the latch can be reconfigured as an input if a 1 is written to it.

Writing to a Port

When an instruction changes a port latch value, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are sampled by their output buffers only during Phase 1 of any clock period. (During Phase 2, the output buffer holds the value sampled during the previous Phase 1). Consequently, the new value in the port latch does not actually appear at the output pin until the next Phase 1, which is at S1P1 of the next machine cycle. See Figure 39 in the Internal Timing section.

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

Figure 5. Ports 1 and 3 Internal Pullup Configurations. Port 2 is similar except that it holds the strong pullup on while emitting 1s that are address bits. (See text, "Accessing External Memory".)

Note: pFET1 is turned on for 2 osc. periods after $\overline{Q}$ makes a 0-to-1 transition. During this time, pFET1 also turns on pFET3 through the inverter to form a latch which holds the 1. pFET2 is also on.

The pullup consists of three pFETs. An n-channel FET (nFET) turns on when a logical 1 is applied to its gate, and turns 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.

The pFET1 transistor in Figure 5 is turned on for 2 oscillator periods after a 0-to-l transition in the port latch. While pFET1 is on, it turns on pFET3 (a weak pullup) through the inverter. This inverter and pFET3 form a latch that holds the 1.

If the pin emits a 1, a negative glitch on the pin from some external source can turn off pFET3, causing the pin to go into a float state. pFET2 is a very weak pullup which is on whenever the nFET is off, in traditional CMOS style. pFET2 is only about 1/10 the strength of pFET3. Its function is to restore a 1 to the pin in the event the pin lost a 1 in a glitch.

Port Loading and Interfacing

The output buffers of Ports 1, 2, and 3 can each drive 4 LS TTL inputs. CMOS pins can be driven by open-collector and open-drain outputs, but 0-to-l transitions will not be fast. An input 0 turns off pullup pFET3, leaving only the very weak pullup pFET2 to drive the transition.

In external bus mode, Port 0 output buffers can drive 8 LS TTL inputs. As port pins, they require external pullups to drive any inputs.

Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Read-modify-write instructions read the latch rather than the pin, and these instructions read a value, possibly change it, and then rewrite it to the latch. When the destination operand is a port, or a port bit, the read-modify-write instructions given in the following table read the latch rather than the pin.

Mnemonic	Instruction	Example
ANL	Logical AND	ANL P1, A
ORL	Logical OR	ORL P2, A
XRL	Logical EX-OR	XRL P3, A
JBC	Jump if bit = 1 and clear bit	JBC P1.1, LABEL
CPL	Complement bit, CPL P3.0	
INC	Increment	INC P2
DEC	Decrement	DEC P2
DJNZ	Decrement and jump if not zero	DJNZ P3, LABEL
MOV, PX.Y, C	Move carry bit to bit Y of Port X	
CLR PX.Y	Clear bit Y of Port X	
SETB PX.Y	Set bit Y of Port X	

The last three instructions in this list are read-modify-write instructions, because they read all 8 bits of the port byte, modify the addressed bit, then write the new byte back to the latch.

Read-modify-write instructions are directed to the latch rather than the pin in order to avoid misinterpreting 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

Hardware Description

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.

Accessing External Memory

Accesses to external memory are either to program memory or to data memory. Accesses to external program memory use the $\overline{PSEN}$ (program store enable) signal as the read strobe. Accesses to external data memory use $\overline{RD}$ or $\overline{WR}$ (alternate functions of P3.7 and P3.6) to strobe the memory. Refer to Figures 36 through 38 in the Internal Timing section for more information.

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 l6-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 emit address bits that are 1s (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 reappear in the next cycle.

If an 8-bit address is used (MOVX @Ri), the contents of the Port 2 SFR remain at the Port 2 pins throughout the external memory cycle, which facilitates 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 pullups. The Address Latche Enable (ALE) signal should be used to capture the address byte into an external latch. The address byte is valid at the negative transition 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 any information in the Port 0 SFR. If the user writes to Port 0 during an external memory fetch, the incoming code byte is corrupted. Therefore, do not write to Port 0 if external program memory is used.

External program memory is accessed under the following two conditions.

1. When the $\overline{EA}$ signal is active; or
2. When the program counter (PC) contains a number larger than 0FFFH (1FFFH for the AT89C52).

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.

Timer/Counters

The AT89C51 has two 16-bit Timer/Counter registers: Timer 0 and Timer 1. The AT89C52 has these two Timer/Counters, and in addition Timer 2. All three can be configured to operate either as Timers or event Counters.

As a Timer, the register is incremented every machine cycle. Thus, the register counts machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

As a Counter, the register is incremented in response to a l-to-0 transition at its corresponding external input pin, T0, T1, or (in the AT89C52) T2. 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 2 machine cycles (24 oscillator periods) are required to recognize a l-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 it should be held for at least one full machine cycle to ensure that a given level is sampled at least once before it changes.

In addition to the Timer or Counter functions, Timer 0 and Timer 1 have four operating modes: (13 bit timer, 16 bit timer, 8 bit auto-reload, split timer). Timer 2 in the AT89C52 has three modes of operation: Capture, Auto-Reload, and baud rate generator.

Timer 0 and Timer 1

Timer/Counters 1 and 0 are present in both the AT89C51 and AT89C52. The Timer or Counter function is selected by control bits $C/\overline{T}$ in the Special Function Register TMOD (Figure 6). 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, but Mode 3 is different. The four modes are described in the following sections.

Mode 0

Both Timers in Mode 0 are 8-bit Counters with a divide-by-32 prescaler. Figure 7 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 $\overline{INT1}$ = 1. Setting GATE=1 allows the Timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements. TR1 is a control bit in the Special Function Register TCON (Figure 8). GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH1and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

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

Figure 6. TMOD: Timer/Counter Mode Control Register

(MSB)							(LSB)
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Timer1				Timer0			

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.	Timer 0 gate bit Timer 0 counter/timer select bit
C/$\overline{T}$	Timer or Counter Selector cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).	Timer 0 M1 bit Timer 0 M0 bit
M1	Mode bit 1	
M0	Mode bit 0	

M1	M0	Mode	Operating Mode
0	0	0	13-bit Timer Mode. 8-bit Timer/Counter THz with TLx as 5-bit prescaler.
0	1	1	16-bit Timer Mode. 16-bit Timer/Counters THx and TLx are cascaded; there is no prescaler.
1	0	2	8-bit Auto Reload. 8-bit auto-reload Timer/Counter THx holds a value which is to be reloaded into TLx each time it overflows.
1	1	3	Split Timer Mode. (Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

Timer SFR	Purpose	Address	Bit-Addressable
TCON	Control	88H	Yes
TMOD	Mode	89H	No
TL0	Timer 0 low-byte	8AH	No
TL1	Timer 1 low-byte	8BH	No
TH0	Timer 0 high-byte	8CH	No
TH1	Timer 1 high-byte	8DH	No
T2CON[1]	Timer 2 control	C8H	Yes
T2MOD[1]	Timer 2 Mode	C9H	No
RCAP2L[1]	Timer 2 low-byte capture	CAH	No
RCAP2H[1]	Timer 2 high-byte capture	CBH	No
TL2[1]	Timer 2 low-byte	CCH	No
TH2[1]	Timer 2 high byte	CDH	No

Note: 1.   AT89C52 only.

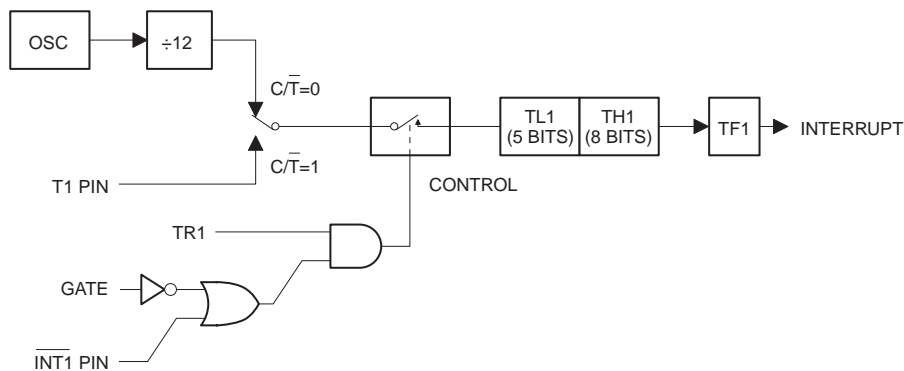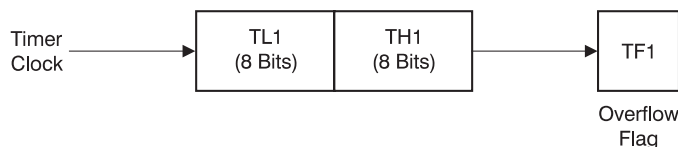**Figure 7.** Timer/Counter 1 Mode 0: 13-Bit Counter

Figure 8. Timer/Counter 1 Mode 1: 16-Bit Counter
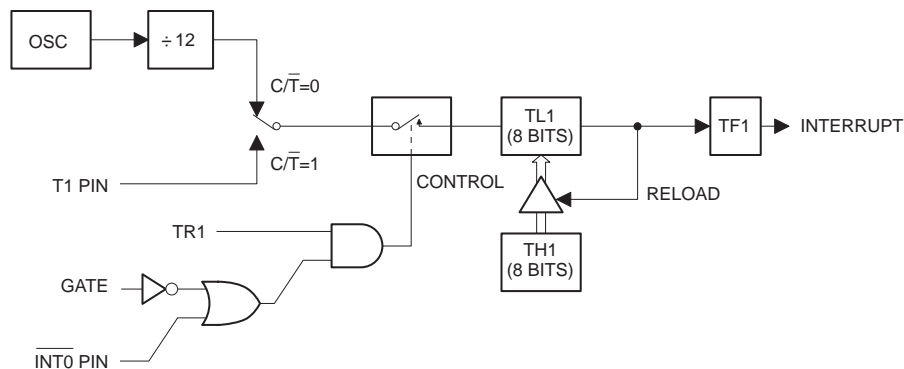

Mode 1

Mode 1 is the same as Mode 0, except that the Timer register is run with all 16 bits. The clock is applied to the combined high and low timer registers (TL1/TH1). As clock pulses are received, the timer counts up: 0000H, 0001H, 0002H, etc. An overflow occurs on the FFFFH-to-0000H overflow flag. The timer continues to count. The overflow flag is the TF1 bit in TCON that is read or written by software. See Figure 8.

Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in Figure 9. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. 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.

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 10. 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 over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt.

Mode 3 is for applications requiring an extra 8-bit timer or counter. With Timer 0 in Mode 3, the AT89C51 can appear to have three Timer/Counters, and an AT89C52, can appear to have 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. In this case, Timer 1 can still be used by the serial port as a baud rate generator or in any application not requiring an interrupt.
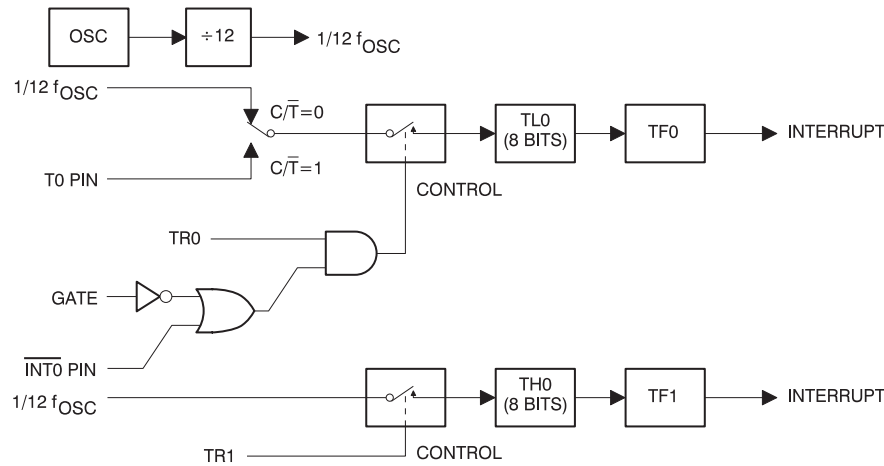
Figure 9. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload

Hardware Description

Figure 10. Timer/Counter 0 Mode 3: Two 8-Bit Counters

Figure 11. 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 2

Timer 2 is a 16-bit Timer/Counter present only in the AT89C52. This is a powerful addition to the other two just discussed. Five extra special function registers are added to accommodate Timer 2 which are: the timer registers, TL2 and TH2, the timer control register, T2CON, and the capture registers, RCAP2L and RCAP2H. Like Timers 0 and 1, it can operate either as a timer or as an event counter, depending on the value of bit C/$\overline{\text{T2}}$ in the Special Function Register T2CON (Figure 12). Timer 2 has three operating modes: capture, auto-reload, and baud rate generator, which are selected by bits in T2CON, as shown in Table 2.

Table 2. Timer 2 Operation 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)

Figure 12. T2CON Timer/Counter 2 Control Register

(MSB)							(LSB)
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 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, 3 and Timer 1 provides transmit baud rate. 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, 3 and Timer 1 provides transmit baud rate. 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/$\overline{T2}$	T2CON.1	Timer or counter select. (Timer 2)
		    0 = Internal timer (OSC/12)
		    1 = External event counter (falling edge triggered).

CP/$\overline{RL2}$	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.

In the Capture Mode, the EXEN2 bit in T2CON selects two options. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter whose overflow sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer2 performs the same way, but a l-to-0 transition at external input T2EX also causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into the RCAP2L and RCAP2H registers, respectively. (RCAP2L and RCAP2H are new Special Function Registers in the AT89C52.) In addition, the transition at T2EX sets the EXF2 bit in T2CON, and EXF2, like TF2, can generate an interrupt.

The Capture Mode is illustrated in Figure 13.

In the auto-reload mode, the EXEN2 bit in T2CON also selects two options. If EXEN2 = 0, then when Timer 2 rolls over it sets TF2 and also reloads the Timer 2 registers with the 16-bit value in the RCAP2L and RCAP2H registers, which are preset by software. If EXEN2 = 1, then Timer 2 performs the same way, but a 1-to-0 transition at external input T2EX also triggers the 16-bit reload and sets EXF2.

The auto-reload mode is illustrated in Figure 14.

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1. This mode is described in conjunction with the serial port. (Figure 17)

Serial Interface

The serial port is full duplex, which means it can transmit and receive simultaneously. It is also receive-buffered, which means it can begin receiving a second byte before a previously received byte has been read from the receive register. (However, if the first byte still has not been read when reception of the second byte is complete, one of the bytes will be lost.) The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in the following four modes.

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

Hardware Description

Figure 13. Timer 2 In Capture Mode
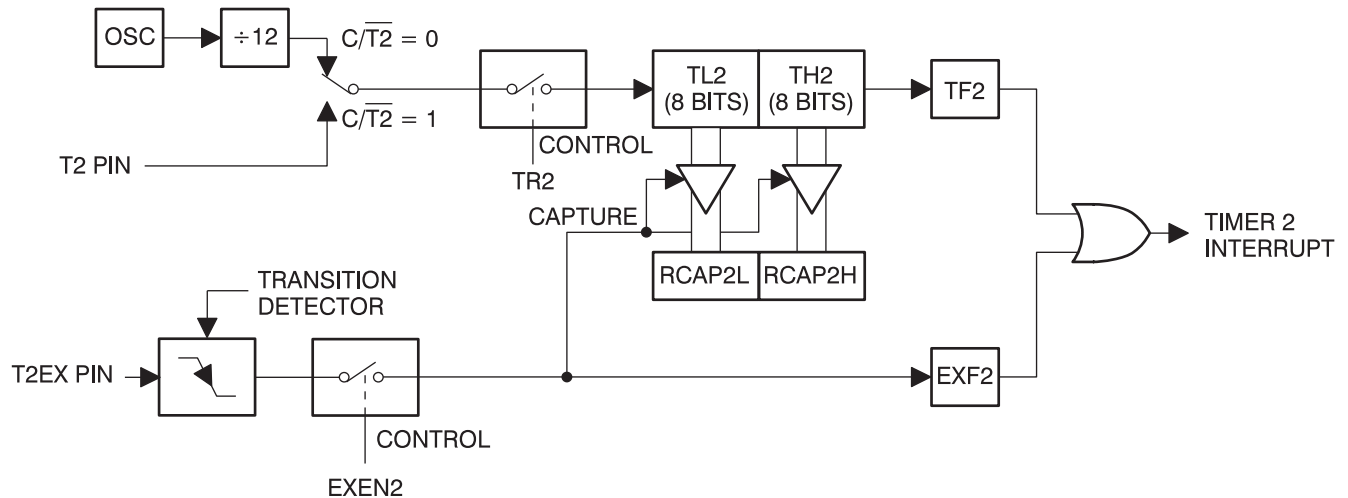
Mode 1: 10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable ninth 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) can 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 ninth data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate, which is variable in Mode 3.

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

Multiprocessor Communications

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

The following example shows how to use the serial interrupt for multiprocessor communications. When the master processor must transmit a block of data to one of several slaves, it first sends out an address byte that 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 is interrupted by a data byte. An address byte, however, interrupts all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave clears its SM2 bit and prepares to receive the data bytes that follows. The slaves that are not addressed set their SM2 bits and ignore the data bytes.

SM2 has no effect in Mode 0 but can be used to check the validity of the stop bit in Mode 1. In a Mode 1 reception, if SM2=1, the receive interrupt is not 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 15. This register contains the mode selection bits, the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).
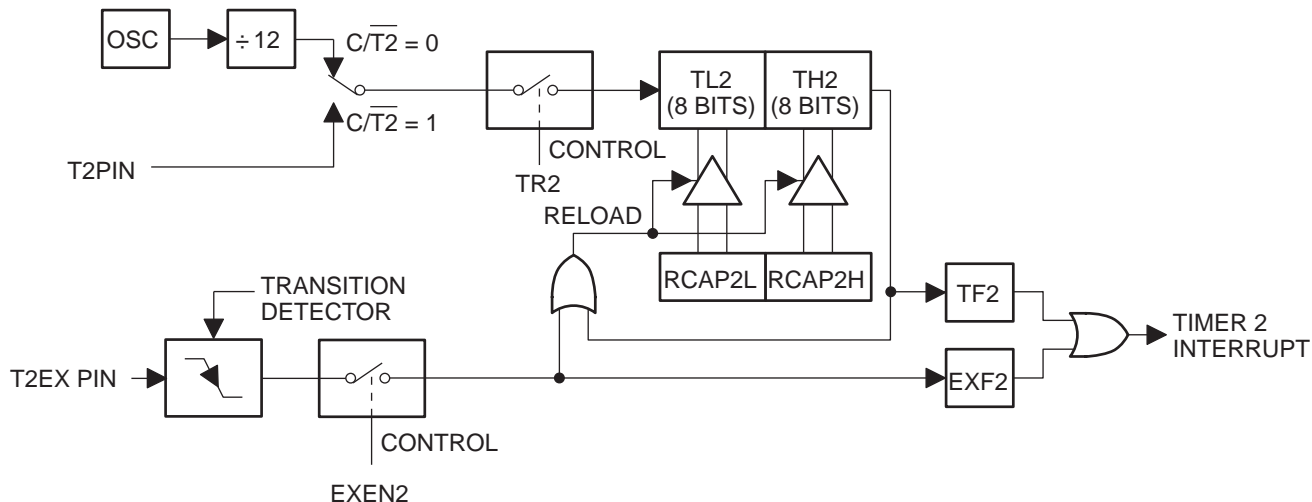
Figure 14. Timer 2 in Auto-Reload Mode (DCEN = 0)

Figure 15. SCON: Serial Port Control Register

(MSB)							(LSB)
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Symbol	Position	Name and Significance
SM0	SCON.7	Serial port mode bit 0 (see table below).
SM1	SCON.6	Serial port mode bit 1 (see table below).
SM2	SCON.5	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 activated if a valid stop bit was not received. In Mode 0, SM2 should be 0.
REN	SCON.4	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
TB8	SCON.3	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software.
RB8	SCON.2	In Modes 2 and 3, 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	SCON.1	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	SCON.0	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.

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

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	fixed ($f_{OSC.}/12$)
0	1	1	8-bit UART	variable (set by timer)
1	0	2	9-bit UART	fixed ($f_{OSC.}/64$ or $f_{OSC.}/32$)
1	1	3	9-bit UART	variable (set by timer)

Hardware Description

Baud Rates

The baud rate in Mode 0 is fixed as shown in the following equation.

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

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

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

In the AT89C51, the Timer 1 overflow rate determines the baud rates in Modes 1 and 3. In the AT89C52, these baud rates can be determined by Timer 1, 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 the baud rate generator, the baud rates in Modes l and 3 are determined by the Timer 1 overflow rate and the value of SMOD according to the following equation.

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

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

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

Programmers can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, 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 16 lists commonly used baud rates and how they can be obtained from Timer 1.

Using Timer 2 to Generate Baud Rates

In the AT89C52, setting TCLK and/or RCLK in T2CON selects Timer 2 as the baud rate generator (Figure 11). Under these conditions, 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 17.

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

In this case, the baud rates in Modes 1 and 3 are determined by the Timer 2 overflow rate according to the following equation.

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

Timer 2 can be configured for either timer or counter operation. In the most typical applications, it is configured for timer operation (C/T2 = 0). Normally, a timer increments every machine cycle (thus at 1/12 the oscillator frequency), but timer operation is different for Timer 2 when it is used as a baud rate generator. As a baud rate generator, Timer 2 increments every state time (thus at 1/2 the oscillator frequency). In this case, the baud rate is given by the following formula.

$$\frac{\text{Modes 1, 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

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

Figure 16. Commonly Used Baud Rates Generated by Timer 1

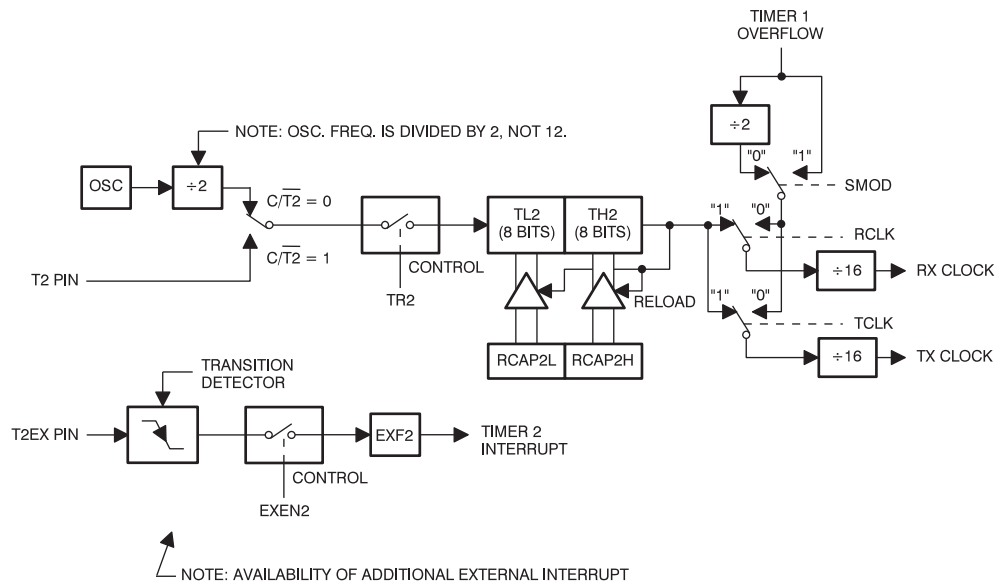| Baud Rate	f$_{OSC}$	SMOE	Timer 1		
			C/$\overline{T}$	Mode	Reload Value
Mode 0 Max: 1 MHz	12 MHz	X	X	X	X
Mode 2 Max: 375K	12 MHz	1	X	X	X
Modes 1, 3: 62.5K	12 MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

Figure 17. Timer 2 in Baud Rate Generator Mode

Figure 17 shows Timer 2 as a baud rate generator. This figure is valid only if RCLK + TCLK = 1 in T2CON. A rollover in TH2 does not set TF2 and does 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. If EXEN2 is set, a l-to-0 transition in T2EX sets EXF2 but does not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus, when Timer 2 is used as a baud rate generator, T2EX can be used as an extra external interrupt.

When Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, programmers should not read from or write to TH2 or TL2. Under these conditions, Timer 2 is incremented every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but should not be written to, because a write might overlap a reload and cause write and/or reload errors. Turn Timer 2 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. Eight data bits are transmitted/received, with the LSB first. The baud rate is fixed at 1/12 the oscillator frequency.

Figure 18 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 ninth position of the transmit shift register and tells the TX Control block to begin a transmission. The internal timing is such that one full machine cycle will elapse between "write to SBUF" and activation of SEND.

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

As data bits shift out to the right, 0s come in from the left. When the MSB of the data byte is at the output position of the shift register, the 1 that was initially loaded into the ninth position is just to the left of the MSB, and all positions to the left of that contain 0s. This condition flags the TX Control block to do one last shift, then deactivate SEND and set TI. Both of these actions occur at S1P1 of the tenth machine cycle after "write to SBUF."

Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register and activates RECEIVE in the next clock phase.

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

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the right-most position arrives at the left-most position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the l0th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.
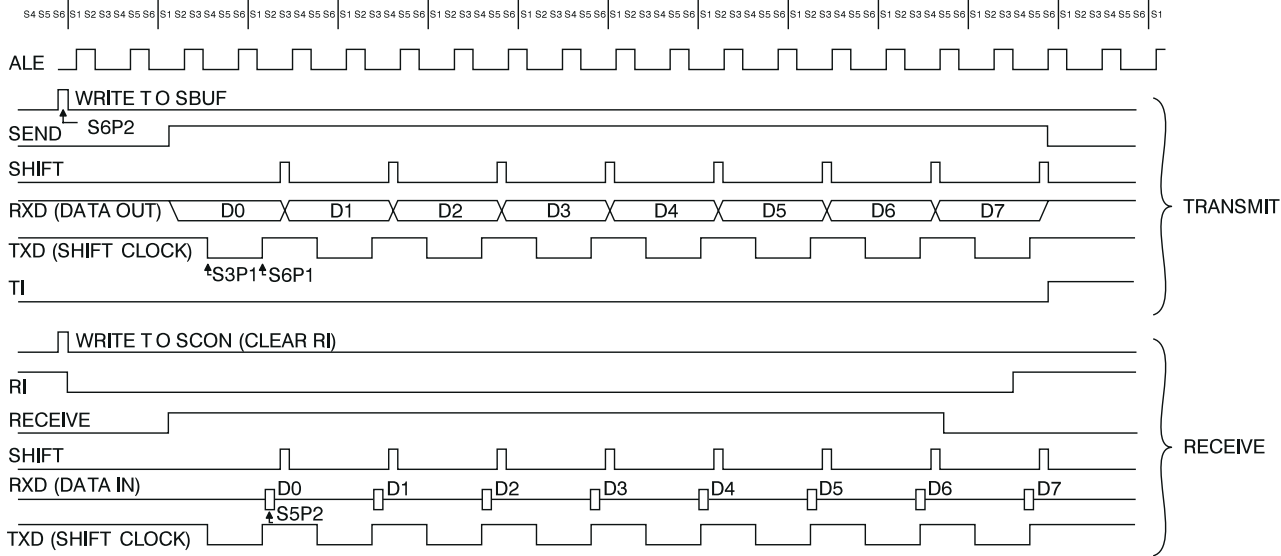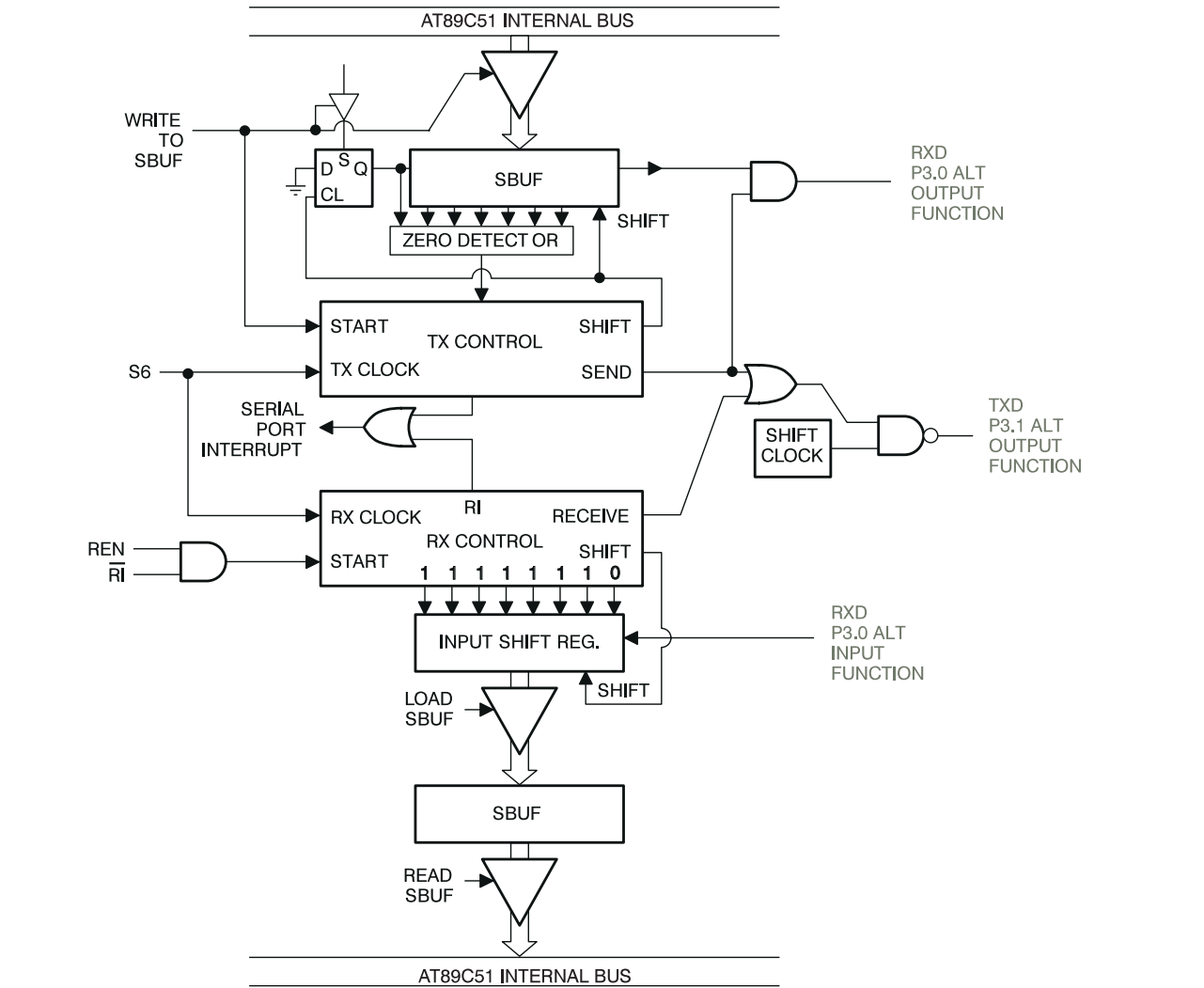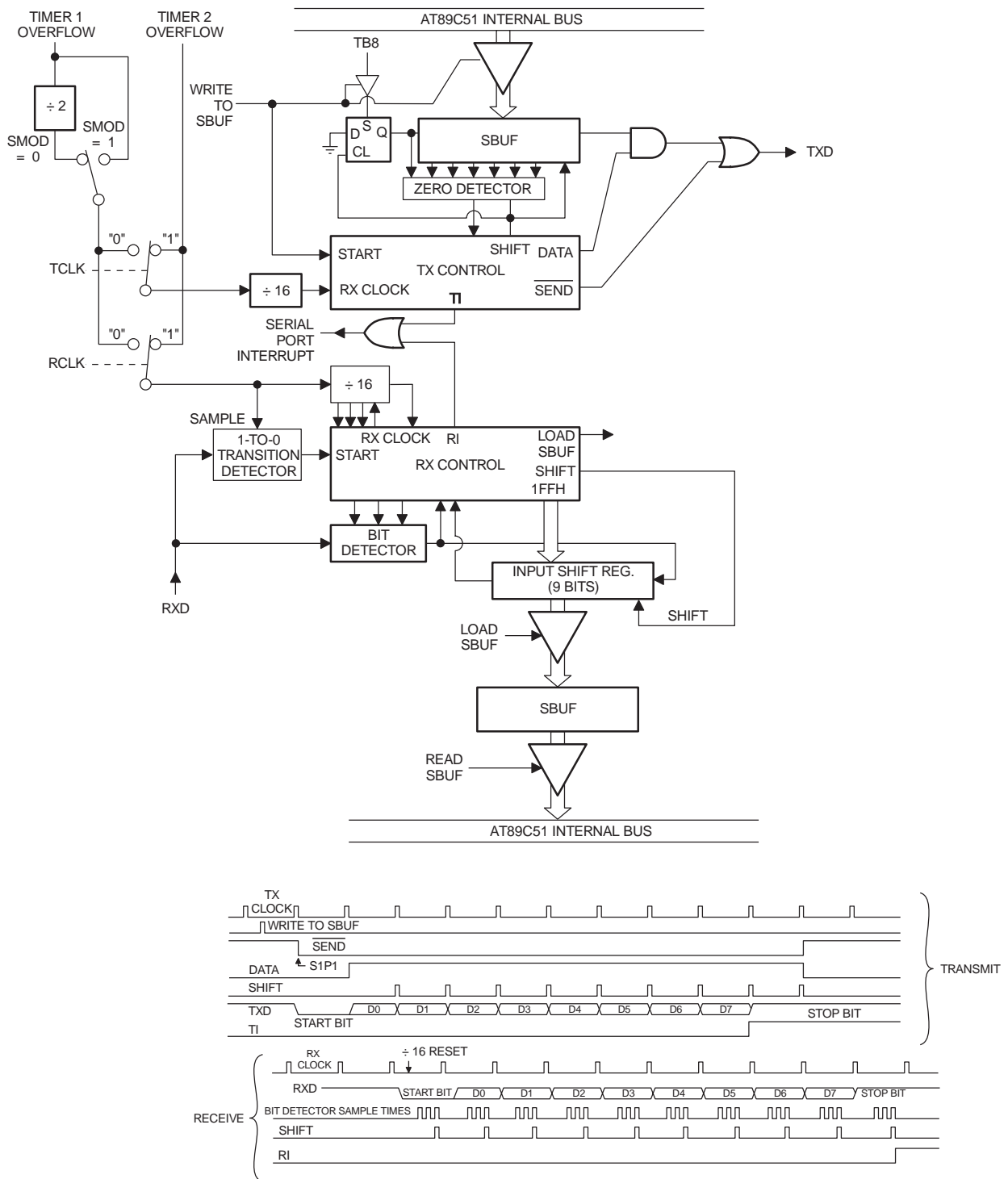
Figure 18. Serial Port Mode 0

Figure 19. Serial Port Mode 1. TCLK, RCLK and Timer 2 are Present In the AT89C52 Only.

Hardware Description

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 AT89C51, the baud rate is determined by the Timer 1 overflow rate. In the AT89C52 the baud rate is determined either by the Timer 1 overflow rate, the Timer 2 overflow rate, or both. In this case, one Timer is for transmit, and the other is for receive.

Figure 19 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 ninth 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 when $\overline{SEND}$ is activated, 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, 0s are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, the 1 that was initially loaded into the ninth position is just to the left of the MSB, and all positions to the left of that contain 0s. This condition flags the TX Control unit to do one last shift, then deactivate $\overline{SEND}$ and set TI. This occurs at the tenth divide-by-16 rollover after "write to SBUF."

Reception is initiated by a 1-to-0 transition detected at RXD. For this purpose, RXD is sampled at a rate of 16 times the established baud rate. 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.

The 16 states of the counter divide each bit time into 16ths. At the seventh, eighth, and ninth 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 to reject noise. In order to reject false bits, if the value accepted during the first bit time is not 0, the receive circuits are reset and the unit continues looking for another l-to-0 transition. If the start bit is valid, it is shifted into the input shift register, and reception of the rest of the frame proceeds.

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

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

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether or not the above conditions are met, the unit continues looking for a l-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 ninth data bit, and a stop bit (1). On transmit, the ninth data bit (TB8) can be assigned the value of 0 or 1. On receive, the ninth data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 of 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 20 and 21 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 ninth 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 ninth 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 when $\overline{SEND}$ is activated, 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 ninth bit position of the shift register. Thereafter, only 0s are clocked in. Thus, as data bits shift out to the right, 0s 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 0s. This condition flags the TX Control unit to do one last shift, then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after "write to SBUF."

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

Figure 20. Serial Port Mode 2
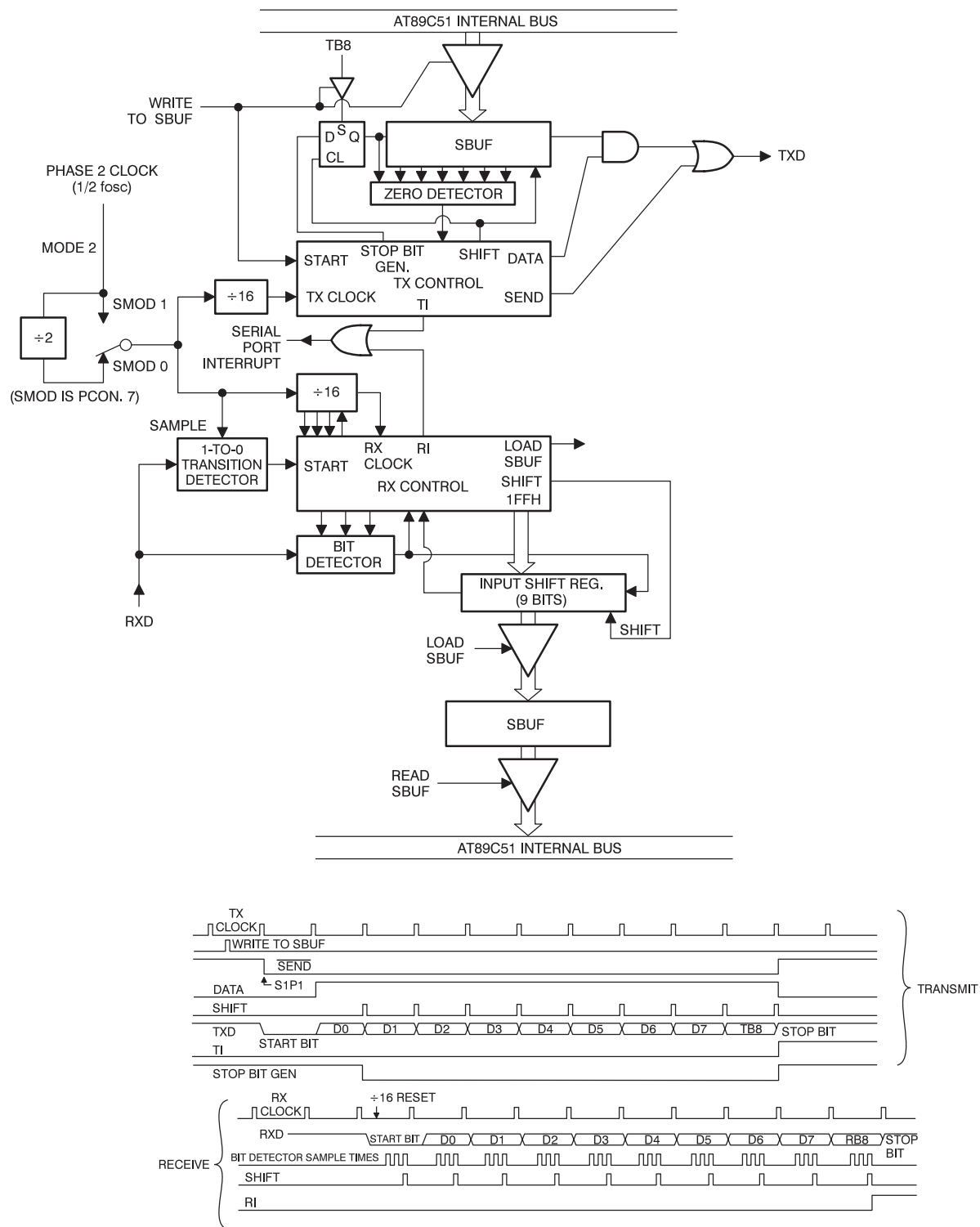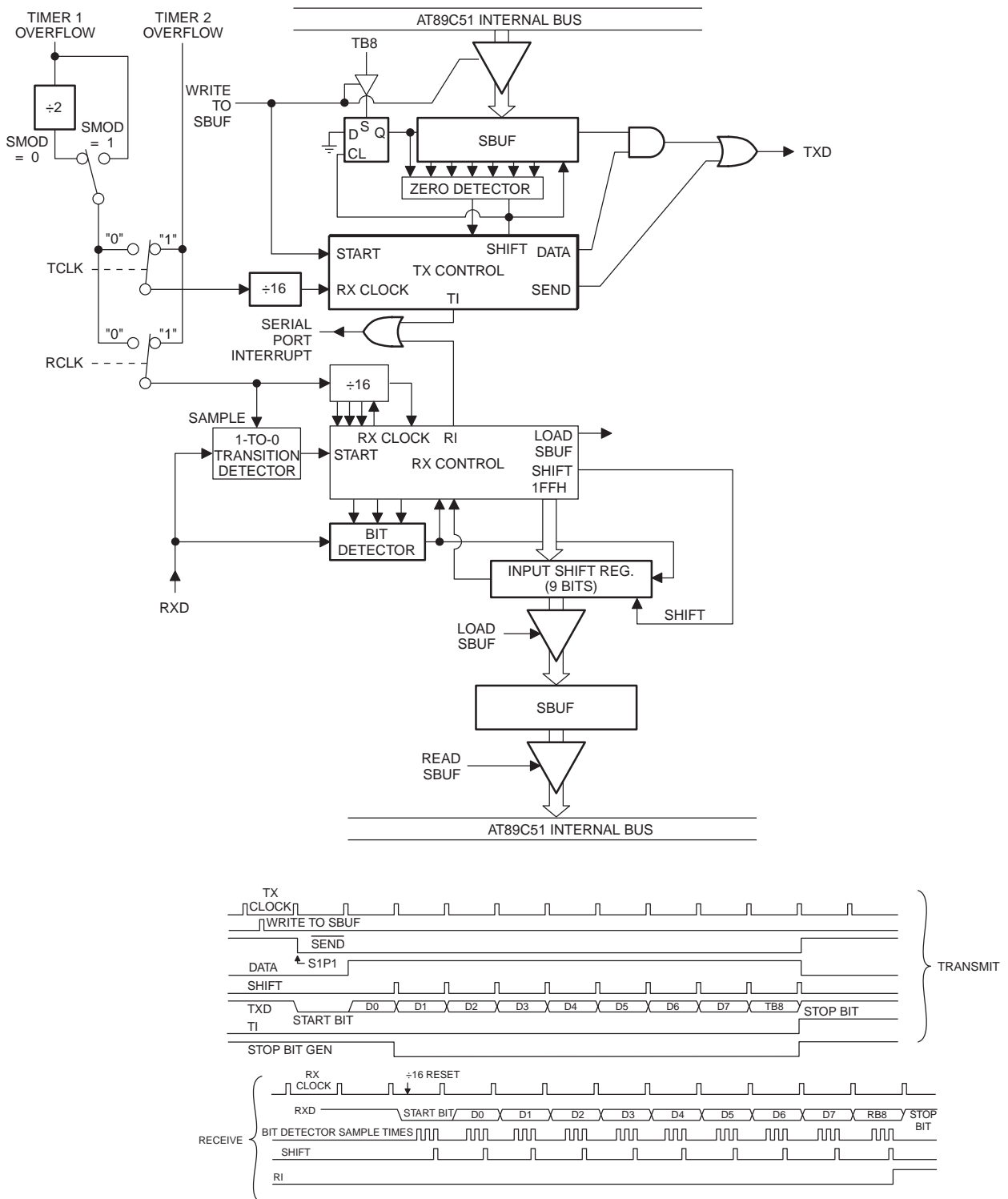

Hardware Description

Figure 21. Serial Port Mode 3. TCLK, RCLK, and Timer 2 are Present in AT89C52 only

At the seventh, eighth and ninth 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 continues looking for another l-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 proceeds.

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 is 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 ninth 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 continues 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.

Interrupts

The AT89C51 provides 5 interrupt sources: two external interrupts, two timer interrupts, and a serial port interrupt. The AT89C52 provides 6 with the extra timer. These are shown in Figure 22.

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 the IE0 and IE1 bits in TCON. When the service routine is vectored to, hardware clears the flag that generated an external interrupt only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source (rather than the on-chip hardware) controls the request flag.

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 for Timer 0 in Mode 3). When a timer interrupt is generated, the on-chip hardware clears the flag that generated it 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 normally must determine whether RI or TI generated the interrupt, and the bit must be cleared in software.

In the AT89C52, 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 TF2 or EXF2 generated the interrupt, and the bit must be cleared in software.
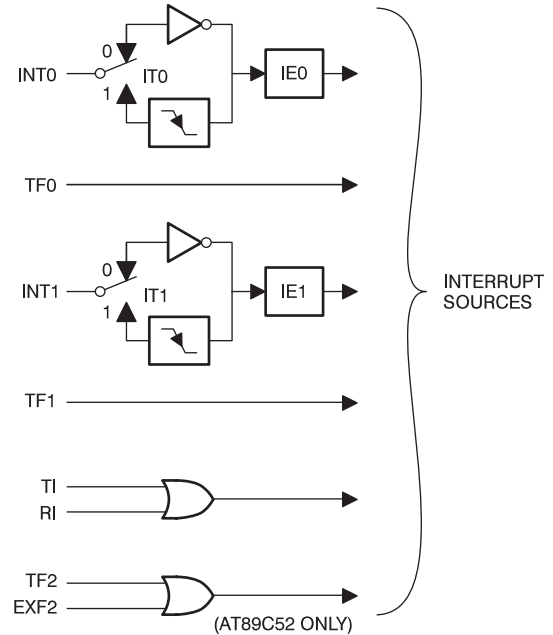
Figure 22. Interrupt Sources

Figure 23. IE: Interrupt Enable Register

(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable bit = 1 enables the interrupt.

Enable bit = 0 disables it.

Symbol	Position	Function
EA	IE.7	Global enable/disable. 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 enable bit.
—	IE.6	Undefined/reserved.
ET2	IE.5	Timer 2 Interrupt enable bit (AT89C52).
ES	IE.4	Serial Port Interrupt enable bit.
ET1	IE.3	Timer 1 Interrupt enable bit.
EX1	IE.2	External Interrupt 1 enable bit.
ET0	IE.1	Timer 0 Interrupt enable bit.
EX0	IE.0	External Interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, since they may be used in future AT89 Series products.

Figure 24. IP: Interrupt Priority Register

(MSB)							(LSB)
—	—	PT2	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority.

Priority bit = 0 assigns low priority.

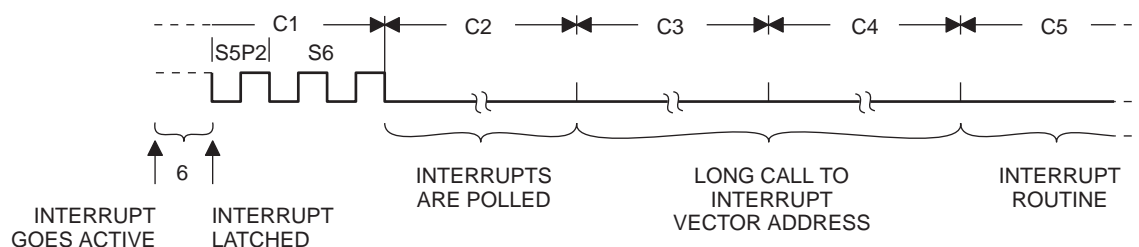| Symbol	Position	Function
—	IP.7	reserved
—	IP.6	reserved
PT2	IP.5	Timer 2 Interrupt priority bit.
PS	IP.4	Serial Port Interrupt priority bit.
PT1	IP.3	Timer 1 Interrupt priority bit.
PX1	IP.2	External Interrupt 1 priority bit.
PT0	IP.1	Timer 0 Interrupt priority bit.
PX0	IP.0	External Interrupt 0 priority bit.

User software should never write 1s to unimplemented bits, since they may be used in future AT89 Series products.

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

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (interrupt enable) at address OA8H. As well as individual enable bits for each interrupt source, there is a global enable/disable bit that is cleared to disable all interrupts or set to turn on interrupts (see Figure 23).

Figure 23 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future microcontrollers.

Figure 25. Interrupt Response Timing Diagram

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 (interrupt priority) at address 0B8H (Figure 24). IP is cleared after a system reset to place all interrupts at the lower priority level by default. A low-priority interrupt can be interrupted by a high-priority interrupt but not by another low-priority interrupt. A high-priority interrupt can not be interrupted by any other interrupt source.

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 determines which request is serviced. Thus within each priority level there 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*.

The IP register contains a number of unimplemented bits. IP.7 and IP.6 are vacant in the AT89C52, and in the AT89C51 these bits and IP.5 are vacant. User software should not write 1s to these bit positions, since they may be used in future products.

INTERRUPT GOES ACTIVE
INTERRUPT LATCHED
INTERRUPTS ARE POLLED
LONG CALL TO INTERRUPT VECTOR ADDRESS
INTERRUPT ROUTINE

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.

How Interrupts Are Handled

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. The AT89C52 Timer 2 interrupt cycle is different, as described in the Response Time Section. 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 blocked 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 write 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. If an active interrupt flag is not being serviced because of one of the above conditions and is not still active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in Figure 25.

Note that if an interrupt of higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 25, then in accordance with the above rules it will be serviced during C5 and C6, without any instruction of the lower priority routine having been executed.

Thus, the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it does not. It never clears the Serial Port or Timer 2 flags. This must be done in the user's software. The processor 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 serviced, as shown in the following table.

Interrupt	Source	Vector Address
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RI or TI	0023H
Timer 2	TF2 or EXF2	002BH
System Reset	RST	0000H

Note: When vectoring to an interrupt the flag that caused the interrupt is automatically cleared by hardware. The exceptions are RI and TI for serial port interrupts, and TF2 and EXF2 for Timer 2 interrupts. Since there are two possible sources for each of these interrupts, it is not practical for the CPU to clear the interrupt flag. These bits must be tested in the ISR to determine the source of the interrupt, and then the interrupting flag is cleared by software.

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.

Interrupt Flag Bits

Interrupt	Flag	SFR Register and Bit Position
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 1	TF1	TCON.7
Timer 0	TF0	TCON.5
Serial port	TI	SCON.1
Serial port	RI	SCON.0
Timer 2	TF2	T2CON.7 (AT89C52)
Timer 2	EXF2	T2CON.6 (AT89C52)

Hardware Description

When an interrupt is accepted the following action occurs:

1. The current instruction completes operation.
2. The PC is saved on the stack.
3. The current interrupt status is saved internally.
4. Interrupts are blocked at the level of the interrupts.
5. The PC is loaded with the vector address of the ISR (interrupt service routine).
6. The ISR executes.

The ISR executes and takes action in response to the interrupt. The ISR finishes with RETI (return from interrupt) instruction. This retrieves the old value of the PC from the stack and restores the old interrupt status. Execution of the main program continues where it left off.

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 INTx pin. If ITx = 1, external interrupt x is edge-triggered. In this mode if successive samples of the $\overline{INTx}$ pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx 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 machine cycle, and then hold it low for at least one machine 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 the external source must deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

Response Time

The $\overline{INT0}$ and $\overline{INT1}$ levels are inverted and latched into the interrupt flags IE0 and IE1 at S5P2 of every machine cycle. Similarly, the Timer 2 flag EXF2 and the Serial Port flags RI and TI are set at S5P2. The values are not actually polled by the circuitry until the next machine cycle.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag TF2 is set at S2P2 and is polled in the same cycle in which the timer overflows.

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 executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles elapsed between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine. Figure 25 shows interrupt response timings.

A longer response time results 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 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. 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 9 cycles.

Single-Step Operation

The AT89C51 interrupt structure allows single-step execution with very little software overhead. As previously noted, an interrupt request will not be serviced while an interrupt of equal priority level is still in progress, nor will it be serviced 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 one instruction of the interrupted program is executed. One way to use this feature for single-stop operation is to program one of the external interrupts (for example, $\overline{INT0}$) to be level-activated. The service routine for the interrupt will terminate with the following code:

```
JNB    P3.2,$  ;Wait Here Till INT0 Goes High
JB     P3.2,$  ;Now Wait Here Till it Goes Low
RETI           ;Go Back and Execute One Instruction
```

If the $\overline{INT0}$ pin, which is also the P3.2 pin, is held normally low, the CPU will go right into the External Interrupt 0 routine and stay there until $\overline{INT0}$ is pulsed (from low to high to low). Then it will execute RETI, go back to the task program, execute one instruction, and immediately reenter 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.

Reset

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

A reset is 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 generating an internal reset, with the timing shown in Figure 26.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. The port pins will maintain their current activities for 19 oscillator periods after a logic 1 has been sampled at the RST pin; that is, for 19 to 31 oscillator periods after the external reset signal has been applied to the RST pin.

While the RST pin is high, ALE and $\overline{PSEN}$ are weakly pulled high. After RST is pulled low, it will take 1 to 2 machine cycles for ALE and $\overline{PSEN}$ to start clocking. For this reason, other devices can not be synchronized to the internal timings of the AT89C51.

Driving the ALE and $\overline{PSEN}$ pins to 0 while reset is active could cause the device to go into an indeterminate state.

The internal reset algorithm writes 0s to all the SFRs except the port latches, the Stack Pointer, and SBUF. The port latches are initialized to FFH, the Stack Pointer to 07H, and SBUF is indeterminate. Table 3 lists the SFRs and their reset values.

The internal RAM is not affected by reset. On power-up the RAM content is indeterminate.

Note: There is no internal pulldown reset pin on NMOS devices, unlike that of Atmel's CMOS microcontroller devices.

Figure 26. Reset Timing
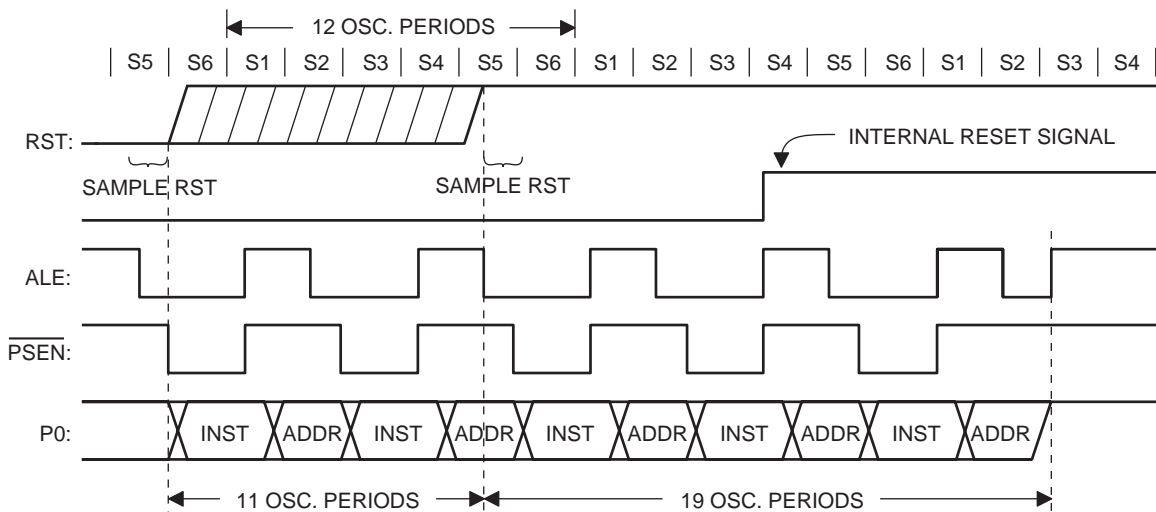
Power-On Reset

For CMOS devices, the external resistor can be removed because the RST pin has an internal pulldown. The capacitor value can then be reduced to 1 µF in Figure 27.

When power is turned on, the circuit holds the RST pin high for an amount of time that depends on the capacitor value and the rate at which it charges. To ensure a valid reset, the RST pin must be held high long enough to allow the oscillator to start up plus two machine cycles.

On power-up, V_{CC} should rise within approximately 10 ms. The oscillator start-up time depends on the oscillator frequency. For a 10 MHz crystal, the start-up time is typically 1 ms. For a 1 MHz crystal, the start-up time is typically 10 ms.

With the given circuit, reducing V_{CC} quickly to 0 causes the RST pin voltage to momentarily fall below 0V. However, this voltage is internally limited and will not harm the device.

Note: The port pins will be in a random state until the oscillator has started and the internal reset algorithm has written 1s to them.
Powering up the device without a valid reset could cause the CPU to start executing instructions from an indeterminate location. This is because the SFRs, specifically the Program Counter, may not get properly initialized.

Hardware Description

Figure 27. Power-On Reset Circuit
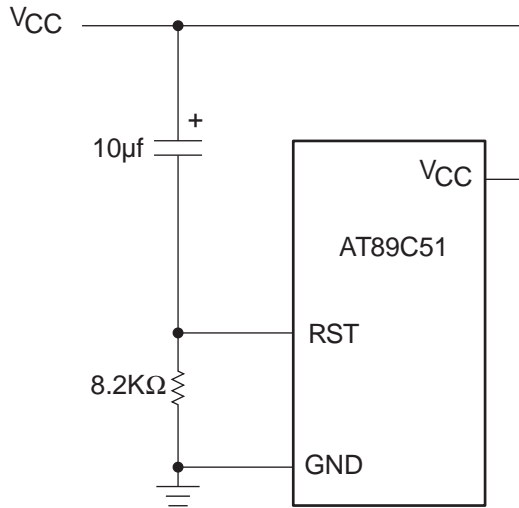
Power-Saving Modes of Operation

The Atmel Microcontrollers have two power-reducing modes, Idle and Power Down. The input through which backup power is supplied during these operations is V_{CC}. Figure 28 shows the internal circuitry which implements these features. In the Idle mode (IDL = 1), the oscillator continues to run and the Interrupt, Serial Port, and Timer blocks continue to be clocked, but the clock signal is gated off to the CPU. In Power Down (PD = 1), the oscillator is frozen. The Idle and Power Down modes are activated by setting bits in Special Function Register PCON. The address of this register is 87H. Figure 29 details its contents.

Idle Mode

An instruction that sets PCON.0 is the last instruction executed before the Idle mode begins. 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 entirety: 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.

Table 3. Reset Values of the SFRs

SFR Name	Reset Value
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	FFH
IP (AT89C51)	XXX00000B
IP (AT89C52)	XX000000B
IE (AT89C51)	0XX00000B
IE (AT89C52)	0X000000B
TMOD	00H
T2MOD (AT89C52)	XXXXXX00B
TCON	00H
T2CON (AT89C52)	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2 (AT89C52)	00H
TL2 (AT89C52)	00H
RCAP2H (AT89C52)	00H
RCAP2L (AT89C52)	00H
SCON	00H
SBUF	Indeterminate
PCON (CHMOS)	0XXX0000B

The flag bits GF0 and GF1 can be used to indicate whether an interrupt occurred during normal operation or during an 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 other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset must be held active for only two machine cycles (24 oscillator periods) to complete the reset.

The signal at the RST pin clears the IDL bit directly and asynchronously. At this time, the CPU resumes program execution from where it left off; that is, at the instruction following the one that invoked the Idle Mode. As shown in Figure 26, two or three machine cycles of program execution

may take place before the internal reset algorithm takes control. On-chip hardware inhibits access to the internal RAM during this time, but access to the port pins is not inhibited. To eliminate the possibility of unexpected outputs at the port pins, the instruction following the one that invokes Idle should not write to a port pin or to external data RAM.
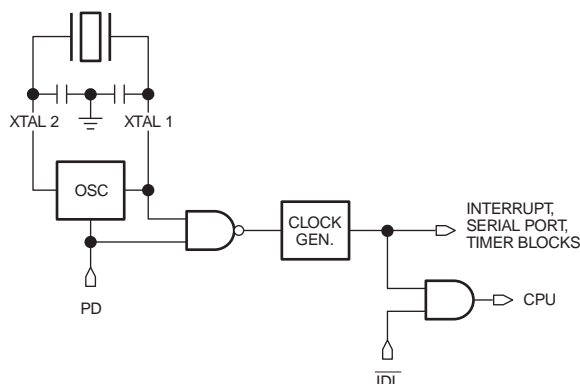
Figure 28. Idle and Power-Down Hardware

Figure 29. PCON Power Control Register

(MSB)							(LSB)
SMOD	—	—	—	GF1	GF0	PD	IDL

Symbol	Position	Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1 and Timer 1 is used to generate baud rate, and the Serial Port is used in 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 (0XXX000).User software should never write 1s to unimplemented bits, since they may be used in future products.

Power Down Mode

An instruction that sets PCON 1 is the last instruction executed before Power Down mode begins. In the Power Down mode, the on-chip oscillator stops. 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.

The only exit from Power Down for the AT89C series is a hardware reset. Reset redefines all the SFRs but does not change the on-chip RAM.

In the Power Down mode of operation, V_{CC} can be reduced to as low as 2V. However, V_{CC} must not be reduced before the Power Down mode is invoked, and V_{CC} must be 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 V_{CC} 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).

Programming

The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. A list of programming companies that support Atmel's products can be found on the Atmel Bulletin Board and in the Microcontroller Programmer Support section of this Data Book. To access the bulletin board, dial 408-436-4309.

The AT89C51/52 programs at V_{PP} = 12V using one 100-msec $\overline{PROG}$ pulse per byte programmed. This results in a programming time of approximately 1.5 msec per byte, for a total programming time of 6 sec for the 4K byte device and 12 sec for the 8K byte device.

Detailed procedures for programming and verifying each device are given in the data sheets.

Program Memory Locks

In some microcontroller applications, the program memory must be secure from software piracy. Atmel has responded to this need by implementing a program memory locking scheme in all of its devices. While it is impossible for anyone to guarantee absolute security against all levels of technological sophistication, the program memory locks present a substantial barrier against illegal readout of protected software.

Table 4. Flash AT89C51 and AT89C52

Device Name	Flash Bytes	Ckt Type	V_{PP}	Time Required to Program Entire Array
AT89C51	4K	CMOS	12V	6 seconds
AT89C52	8K	CMOS	12V	12 seconds

The procedure for programming the lock bits is detailed in the data sheets.

Hardware Description

Table 5 lists the Lock Bits and their corresponding effects on the microcontroller.

Erasing the Flash also erases the Lock Bits, returning the microcontroller to full functionality.

Table 5. Program Lock Bits and Their Features

Program Lock Bits			Protection Type
Mode	LB1	LB2	LB3	
1	U	U	U	No program lock features enabled.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, EA is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as 2, also verify is disabled.
4	P	P	P	Same as 3, also external execution is disabled.

Notes: 1. P = Programmed; U = Unprogrammed

2. Any other combination of the Lock Bits is not defined.

Table 6. Program Protection

Device	Lock Bits
AT89C51	LB1, LB2, LB3
AT89C52	LB1, LB2, LB3
AT89C2051	LB1, LB2
AT89C1051	LB1, LB2

When Lock Bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

ONCE™ Mode

The ONCE ("on-circuit emulation") mode facilitates testing and debugging of systems using the device without requiring the device to be removed from the circuit. The ONCE mode is invoked by taking the following steps.

1. Pull ALE low while the device is in reset and PSEN is high;

2. Hold ALE low as RST is deactivated.

While the device is in ONCE mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the device is in this mode, an emulator or test CPU can be used to drive the circuit. A reset restores normal operation.

On-Chip Oscillators

The crystal specifications and capacitance values (C1 and C2 in Figure 30) are not critical. 30 pF can be used in these positions at any frequency with good quality crystals. A ceramic resonator can be used in place of the crystal in cost-sensitive applications. When a ceramic resonator is used, C1 and C2 are normally selected to be of somewhat higher values, typically, 47 pF. The manufacturer of the ceramic resonator should be consulted for recommendations on the values of these capacitors.

In general, crystals used with these devices typically have the following specifications.

ESR (Equivalent Series Resistance) see Figure 31

CO (Shunt Capacitance)　　　　7.0 pF max.

CL (Load Capacitance)　　　　30 pF + 3 pF

Drive Level　　　　　　　　1 mW

Frequency, tolerance and temperature range are determined by the system requirements.

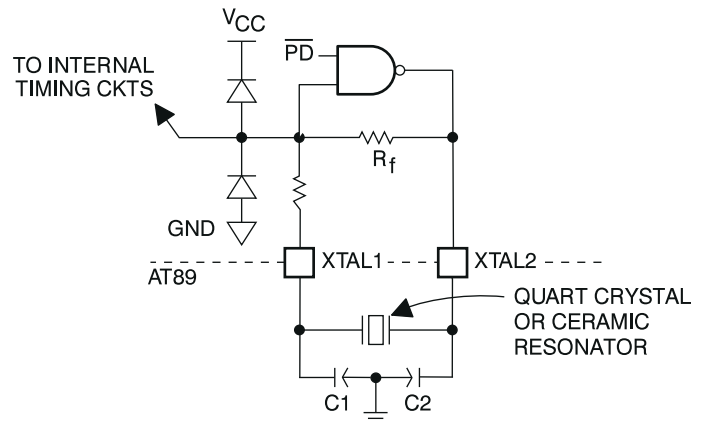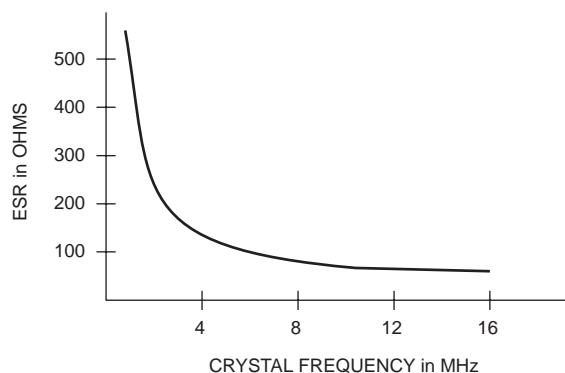**Figure 30.** Using the On-Chip Oscillator

Figure 31. ESR versus Frequency

The on-chip oscillator circuitry shown in Figure 32, consists of a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator.
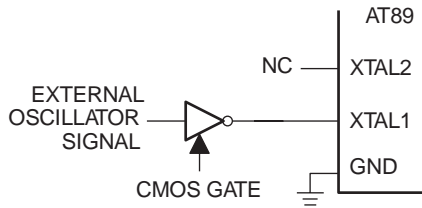
To drive the parts with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 floating, as shown in Figure 33.

Figure 32. On-Chip Oscillator Circuitry for the AT89C51
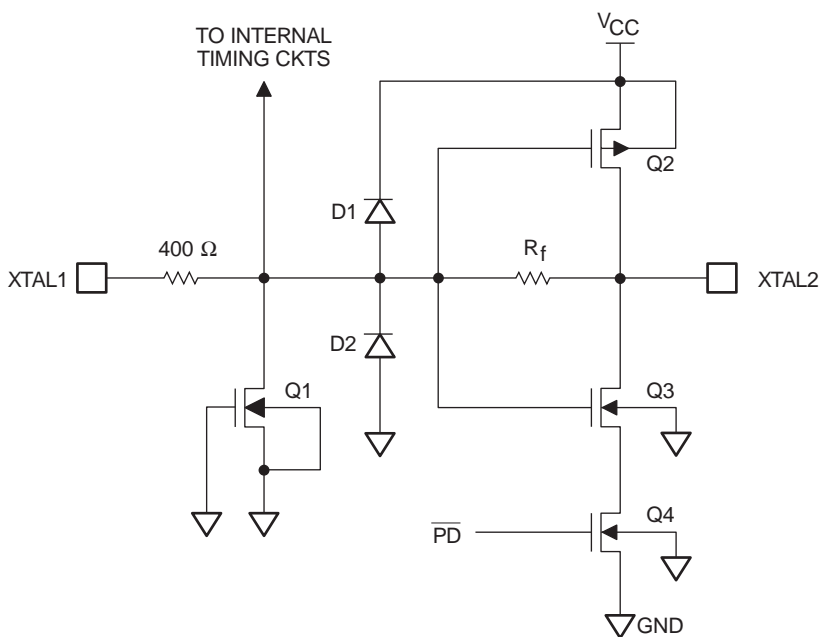
Note: In Atmel's CMOS microcontrollers the Oscillator Specification differs from that in NMOS versions.

Figure 33. Using an External Clock Source

Internal Timing

Figures 34 through 37 show the various strobe and port signals being clocked internally. The figures do not show rise and fall times of the signals, nor do they show propagation delays between the XTAL signal and events at other pins.

Rise and fall times are dependent on the external loading that each pin must drive. They are often taken to be about 10 ns, measured between 0.8V and 2.0V.

Propagation delays are different for different pins. For a given pin the delays vary with pin loading, temperature, V_{CC}, and manufacturing lot. If the XTAL waveform is taken as the timing reference, propagation delays may vary from 25 to 125 ns.

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

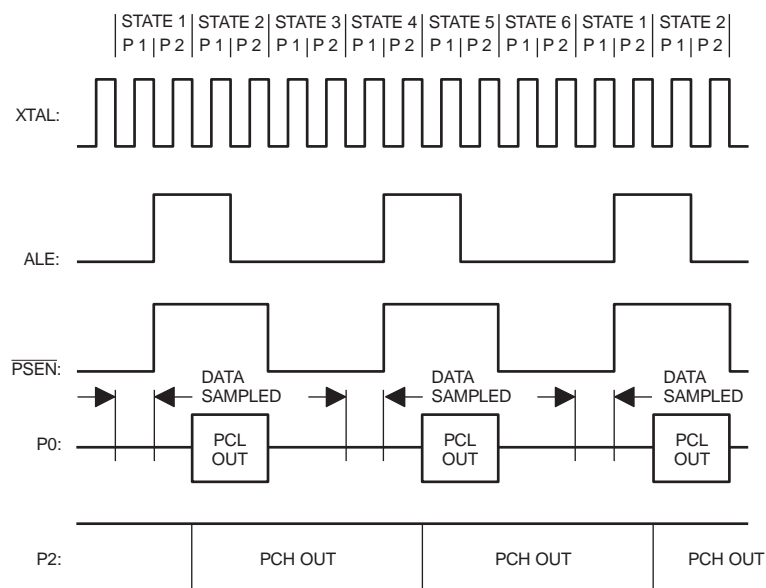**Figure 34.** External Program Memory Fetches
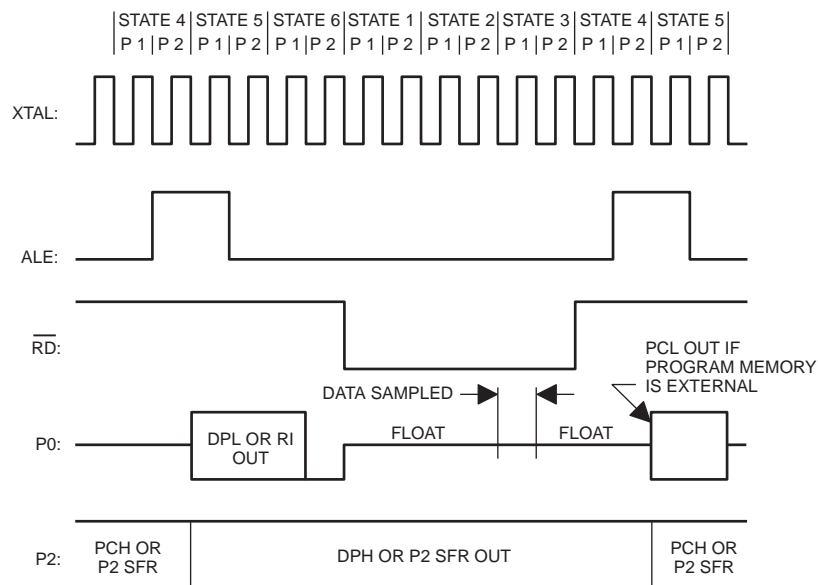

Figure 35. External Data Memory Read Cycle
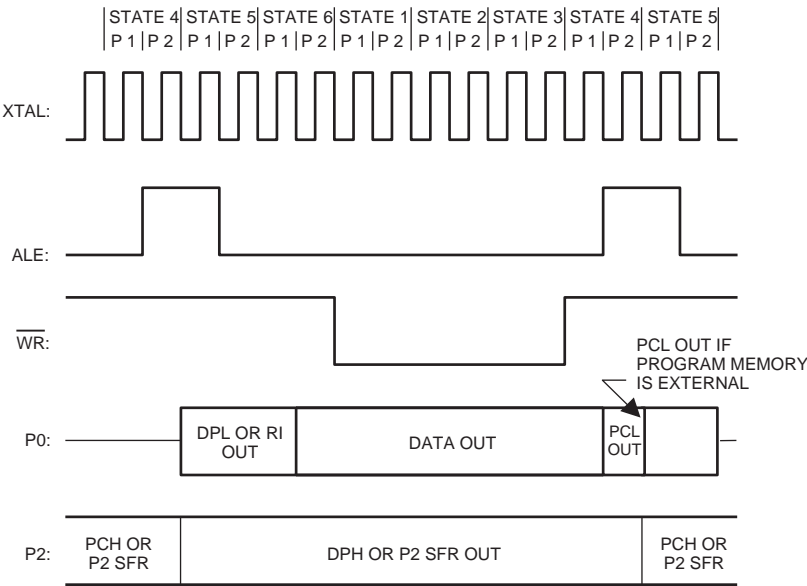
Hardware Description

Figure 36. External Data Memory Write Cycle

Figure 37. Port Operation