intel®

8XC51RA/RB/RC
Hardware Description

February 1995

Order Number: 272668-001

Contact your local Intel sales office or your distributor to obtain the latest specifications before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

8XC51RA/RB/RC HARDWARE DESCRIPTION

CONTENTS

PAGE

CONTENTS

PAGE

1.0 INTRODUCTION

The 8XC51RA/RB/RC is a highly integrated 8-bit microcontroller based on the MCS® 51 architecture. As a member of the MCS 51 family, the 8XC51RA/RB/RC is optimized for control applications. Its key features are 512 bytes on-chip RAM and watchdog timer. Also included are an enhanced serial port for multi-processor communications, an up/down timer/counter and enhanced program lock scheme for the on-chip program memory. Since the 8XC51RA/RB/RC products are CHMOS, they have two software selectable reduced power modes: Idle Mode and Power Down Mode.

The 8XC51RA/RB/RC uses the standard 8051 instruction set and is pin-for-pin compatible with the existing MCS 51 family of products.

This document presents a comprehensive description of the on-chip hardware features of the 8XC51RA/RB/RC. It begins with a discussion of the on-chip memory and then discusses each of the peripherals listed below.

- 512 Bytes On-Chip RAM
- Hardware Watchdog Timer (One-Time Enabled with Reset-Out)
- ALE Disable/Enable
- Four 8-Bit Bidirectional Parallel Ports
- Three 16-Bit Timer/Counters with

— One Up/Down Timer/Counter
— Clock Out

- Full-Duplex Programmable Serial Port with

— Framing Error Detection
— Automatic Address Recognition

- Interrupt Structure with

— Seven Interrupt Sources
— Four Priority Levels

- Power-Saving Modes

— Idle Mode
— Power Down Mode

- Three-Level Program Lock System

Table 1 summarizes the product names and memory differences of the various 8XC51RA/RB/RC products currently available. Throughout this document, the products will generally be referred to as the C51RX.

Table 1. C51RX Family of Microcontrollers

ROM Device	EPROM Version	ROMless Version	ROM/ EPROM Bytes	RAM Bytes
83C51RA	87C51RA	80C51RA	8K	512
83C51RB	87C51RB	80C51RA	16K	512
83C51RC	87C51RC	80C51RA	32K	512

2.0 MEMORY ORGANIZATION

All MCS-51 devices have a separate address space for Program and Data Memory. Up to 64 Kbytes each of external Program and Data Memory can be addressed.

2.1 Program Memory

If the $\overline{EA}$ pin is connected to V_{SS}, all program fetches are directed to external memory. On the 83C51RA (or 87C51RA), if the $\overline{EA}$ pin is connected to V_{CC}, then program fetches to addresses 0000H through 1FFFH are directed to internal ROM and fetches to addresses 2000H through FFFFH are to external memory.

On the 83C51RB (or 87C51RB) if $\overline{EA}$ is connected to VCC, program fetches to addresses 0000H through 3FFFH are directed to internal ROM, and fetches to addresses 4000H through FFFFH are to external memory.

On the 83C51RC (or 87C51RC) if $\overline{EA}$ is connected to V_{CC}, program fetches to addresses 0000H through 7FFFH are directed to internal ROM or EPROM and fetches to addresses 8000H through FFFFH are to external memory.

2.2 Data Memory

The C51RX has internal data memory that is mapped into four separate segments: the lower 128 bytes of RAM, upper 128 bytes of RAM, 128 bytes special function register (SFR) and 256 bytes expanded RAM (ERAM).

The four segments are:

1. The Lower 128 bytes of RAM (addresses 00H to 7FH) are directly and indirectly addressable.
2. The Upper 128 bytes of RAM (addresses 80H to FFH) are indirectly addressable only.
3. The Special Function Registers, SFRs, (addresses 80H to FFH) are directly addressable only.
4. The 256-bytes expanded RAM (ERAM, 00H–FFH) are indirectly accessed by move external instruction, MOVX, and with the EXTRAM bit cleared.

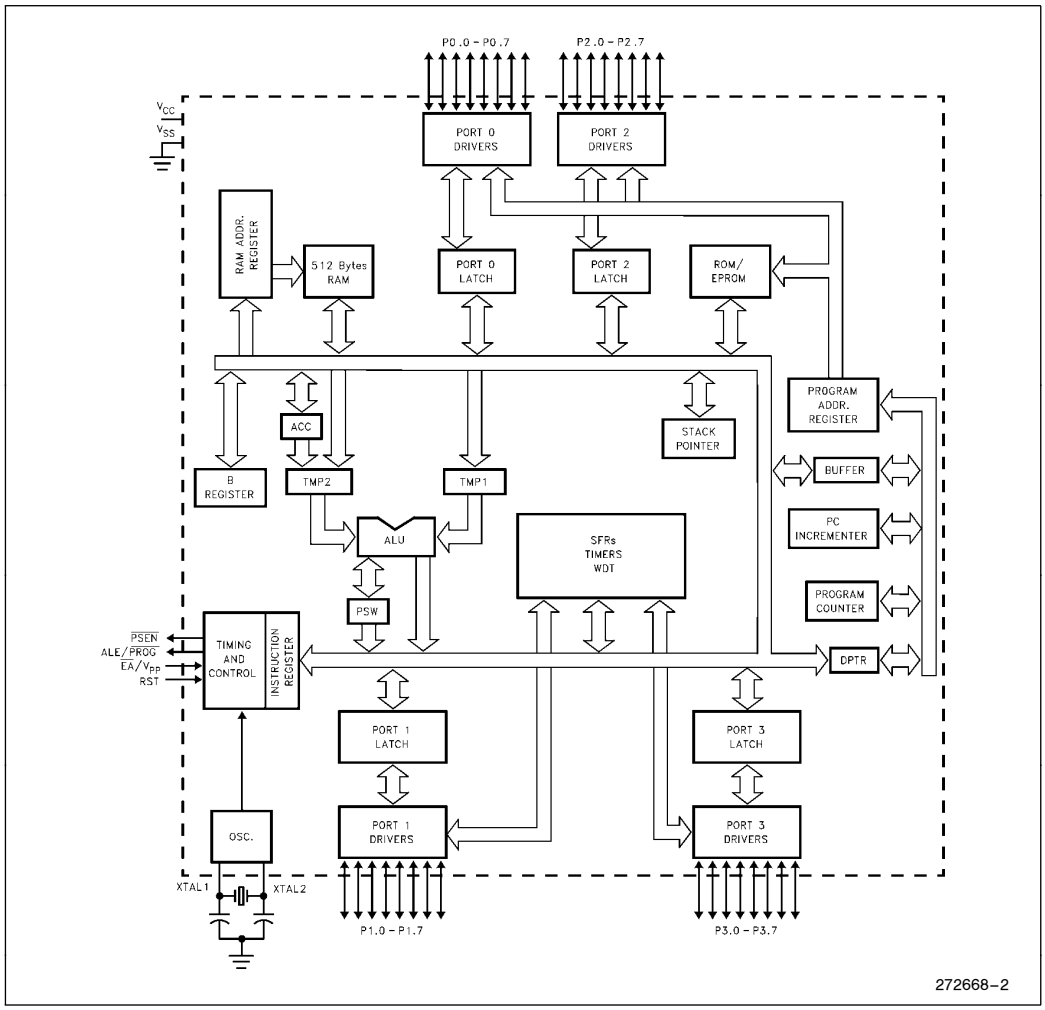


Figure 1. 8XC51RA/RB/RC Functional Block Diagram

The Lower 128 bytes can be accessed by either direct or indirect addressing. The Upper 128 bytes can be accessed by indirect addressing only. The Upper 128 bytes occupy the same address space as the SFR. That means they have the same address, but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the CPU knows whether the access is to the upper 128 bytes of data RAM or to SFR space by the addressing mode used in the instruction. Instructions that use direct addressing access SFR space. For example:

MOV 0A0H, #data

accesses the SFR at location 0A0H (which is P2). Instructions that use indirect addressing access the Upper 128 bytes of data RAM. For example:

MOV @R0, #data

where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

The 256 bytes ERAM can be accessed by indirect addressing, with EXTRAM bit cleared and MOVX instructions. This part of memory is physically located on-chip, logically occupies the first 256-bytes of external data memory.

Table 2. AUXR: Auxiliary Register

AUXR Address = 8EH Reset Value = xxxx xx00B

Not Bit Addressable

—	—	—	—	—	—	EXTRAM	DISALE
7	6	5	4	3	2	1	0

Bit

DISALE Disable/Enable ALE

DISALE	Operating Mode
0	ALE is emitted at a constant rate of 1/6 the oscillator frequency
1	ALE is active only during a MOVX or MOVC instruction

EXTRAM Internal/External RAM (00H–FFH) access using MOVX @Ri/@DPTR

— Reserved for future expansion

EXTRAM	Operating Mode
0	Internal ERAM (00H–FFH) access using MOVX @Ri/@DPTR
1	External data memory access

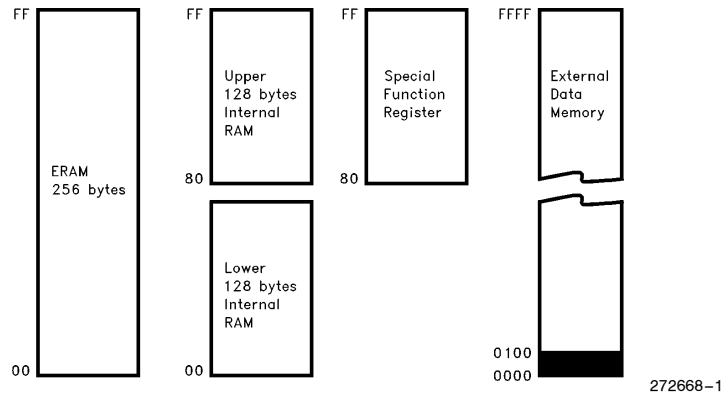


Figure 2. Internal and External Data Memory Address Space with EXTRAM = 0

With EXTRAM = 0, the ERAM is indirectly addressed, using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. An access to ERAM will not affect ports P0, P2, P3.6 (WR#) and P3.7 (RD#). For example, with EXTRAM = 0,

MOVX @R0,#data

where R0 contains 0A0H, access the ERAM at address 0A0H rather than external memory. An access to external data memory locations higher than FFH (i.e. 0100H to FFFFH) will be performed with the MOVX DPTR instructions in the same way as in the standard MCS® 51, so with P0 and P2 as data/address bus, and P3.6 and P3.7 as write and read timing signals. Refer to Figure 2.

With EXTRAM = 1, MOVX @Ri and MOVX @DPTR will be similar to the standard MCS® 51. MOVX @Ri will provide an eight-bit address multiplexed with data on Port0 and any output port pins can be used to output higher order address bits. This is to provide the external paging capability. MOVX @DPTR will generate a sixteen-bit address. Port2 outputs the high-order eight address bits (the contents of DPH) while Port0 multiplexes the low-order eight address bits (DPL) with data. MOVX @Ri and MOVX @DPTR will generate either read or write signals on P3.6 (#WR) and P3.7 (#RD).

The stack pointer (SP) may be located anywhere in the 256 bytes RAM (lower and upper RAM) internal data memory. The stack may not be located in the ERAM.

3.0 SPECIAL FUNCTION REGISTERS

A map of the on-chip memory area called the SFR (Special Function Register) space is shown in Table 3.

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

User software should not write 1s to these unimplemented locations, since they may be used in future MCS-51 products to invoke new features. In that case the reset or inactive values of the new bits will always be 0, and their active values will be 1.

The functions of the SFRs are outlined below. More information on the use of specific SFRs for each peripheral is included in the description of that peripheral.

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

Table 3. 8XC51RA/RB/RC SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000	SADEN 00000000							0BFH
0B0H	P3 11111111							IPH XX000000	0B7H
0A8H	IE 0X000000	SADDR 00000000							0AFH
0A0H	P2 11111111						WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXXXXX00		8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000				PCON 00000000	87H

Table 4. PSW: Program Status Word Register

PSW Address = 0D0H Reset Value = 0000 0000B

Bit Addressable

	CY	AC	F0	RS1	RS0	OV	—	P
Bit	7	6	5	4	3	2	1	0

Symbol	Function
CY	Carry flag.
AC	Auxiliary Carry flag. (For BCD Operations)
F0	Flag 0. (Available to the user for general purposes).
RS1	Register bank select bit 1.
RS0	Register bank select bit 0.

RS1	RS0	Working Register Bank and Address	
0	0	Bank 0	(00H–07H)
0	1	Bank 1	(08H–0FH)
1	0	Bank 2	(10H–17H)
1	1	Bank 3	(18H–1FH)

OV	Overflow flag.
—	User definable flag.
P	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the Accumulator, i.e., even parity.

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

Stack Pointer: The Stack Pointer Register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. The stack may reside anywhere in on-chip RAM. On reset, the Stack Pointer is initialized to 07H causing the stack to begin at location 08H.

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

Program Status Word: The PSW register contains program status information as detailed in Table 4.

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

Timer Registers: Register pairs (TH0, TL0), (TH1, TL1), and (TH2, TL2) are the 16-bit count registers for Timer/Counters 0, 1, and 2 respectively. Control and status bits are contained in registers TCON and TMOD for Timers 0 and 1 and in registers T2CON and T2MOD for Timer 2. The register pair (RCAP2H, RCAP2L) are the capture/reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Serial Port Registers: The Serial Data Buffer, SBUF, 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. Register SCON contains the control and status bits for the Serial Port. Registers SADDR and SADEN are used to define the Given and the Broadcast addresses for the Automatic Address Recognition feature.

Interrupt Registers: The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the 7 interrupts in the IP register.

Power Control Register: PCON controls the Power Reduction Modes. Idle and Power Down Modes.

WatchDog Timer ReSet: (WDTRST) register is used to keep the watchdog timer from periodically resetting the part.

4.0 PORT STRUCTURES AND OPERATION

All four ports in the C51RX 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 1 and Port 3 pins are multifunctional. They are not only port pins, but also serve the functions of various special features as listed in Table 5.

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.

4.1 I/O Configurations

Figure 3 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 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 pin" signal. See the Read-Modify-Write Feature section.

As shown in Figure 3, the output drivers of Ports 0 and 2 are switchable to an internal ADDRESS and ADDRESS/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.

Table 5. Alternate Port Functions

Port Pin	Alternate Function
P0.0/AD0–P0.7/AD7	Multiplexed Byte of Address/Data for External Memory
P1.0/T2	Timer 2 External Clock Input/Clock-Out
P1.1/T2EX	Timer 2 Reload/Capture/Direction Control
P2.0/A8–P2.7/A15	High Byte of Address for External Memory
P3.0/RXD	Serial Port Input
P3.1/TXD	Serial Port Output
P3.2/$\overline{\text{INTO}}$	External Interrupt 0
P3.3/$\overline{\text{INT}}$	External Interrupt 1
P3.4/T0	Timer 0 External Clock Input
P3.5/T1	Timer 1 External Clock Input
P3.6/$\overline{\text{WR}}$	Write Strobe for External Memory
P3.7/$\overline{\text{RD}}$	Read Strobe for External Memory

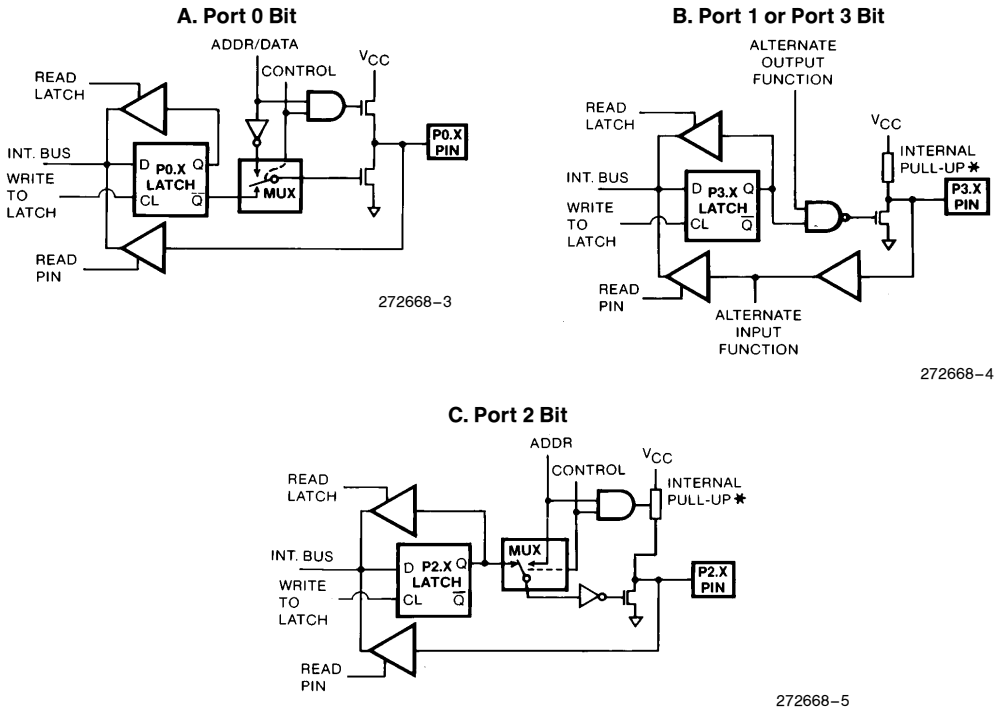


Figure 3. C51RX Port Bit Latches and I/O Buffers

Also shown in Figure 3 is that if a P1 or P3 latch contains a 1, then the output level is controlled by the signal labeled "alternate output function." The actual pin level is always available to the pin's alternate input function, if any.

Ports 1, 2, and 3 have internal pullups. Port 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 ADDRESS/DATA BUS). To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. On 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 differs from the other ports in not having internal pullups. The pullup FET in the P0 output driver (see Figure 3) is used only when the Port is emitting 1s during external memory accesses. Otherwise the pullup FET is off. Consequently P0 lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, which floats the pin and allows it to be used as a high-impedance input. Because Ports 1 through 3 have fixed internal pullups they are sometimes call "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 it floats when configured as an input.

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

4.2 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 State 6 Phase 2 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 will be at S1P1 of the next machine cycle. Refer to Figure 4. For more information on internal timings refer to the CPU Timing section.

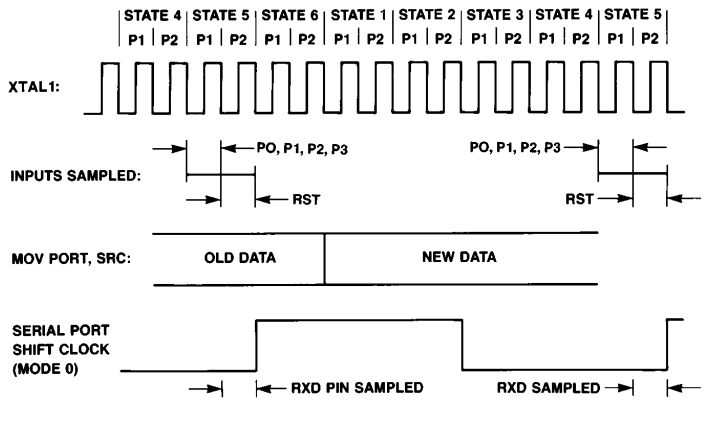


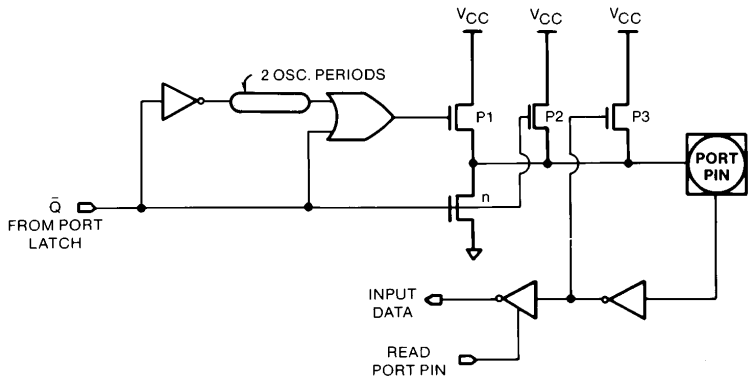
Figure 4. Port Operation

If the change requires a 0-to-1 transition in Ports 1, 2, and 3, an additional pullup 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 pullup can source about 100 times the current that the normal pullup can. The internal pullups are field-effect transistors, not linear resistors. The pull-up arrangements are shown in Figure 5.

The pullup consists of three pFETs. Note 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 is the transistor that is turned on for 2 oscillator periods after a 0-to-1 transition in the port latch. A 1 at the port pin turns on pFET3 (a weak pull-up), through the invertor. This invertor and pFET form a latch which hold the 1.

If the pin is emitting 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. 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.



CHMOS Configuration. pFET 1 is turned on for 2 osc. periods after $\overline{Q}$ makes a 0-to-1 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 pullup on while emitting 1s that are address bits. (See text, "Accessing External Memory".)

Figure 5. Ports 1 and 3 Internal Pullup Configurations

4.3 Port Loading and Interfacing

The output buffers of Ports 1, 2, and 3 can each sink 1.6 mA at 0.45 V. These port pins can be driven by open-collector and open-drain outputs although 0-to-1 transitions will not be fast since there is little current pulling the pin up. 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 each sink 3.2 mA at 0.45 V. However, as port pins they require external pullups to be able to drive any inputs.

See the latest revision of the data sheet for design-in information.

4.4 Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Which ones do which? The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions. Listed below are the 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.

4.5 Accessing External Memory

Accesses to external memory are of two types: accesses to external Program Memory and accesses to external Data Memory. Accesses to external Program Memory use signal $\overline{\text{PSEN}}$ (program store enable) as the read strobe. Accesses to external Data Memory use $\overline{\text{RD}}$ or $\overline{\text{WR}}$ (alternate functions of P3.7 and P3.6) to strobe the memory. Refer to Figures 6 through 8.

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).

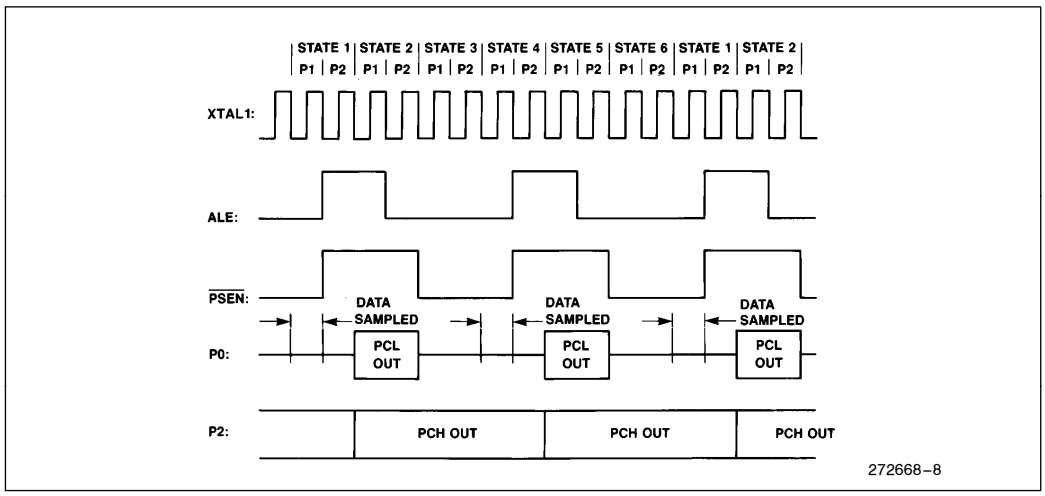


Figure 6. External Program Memory Fetches

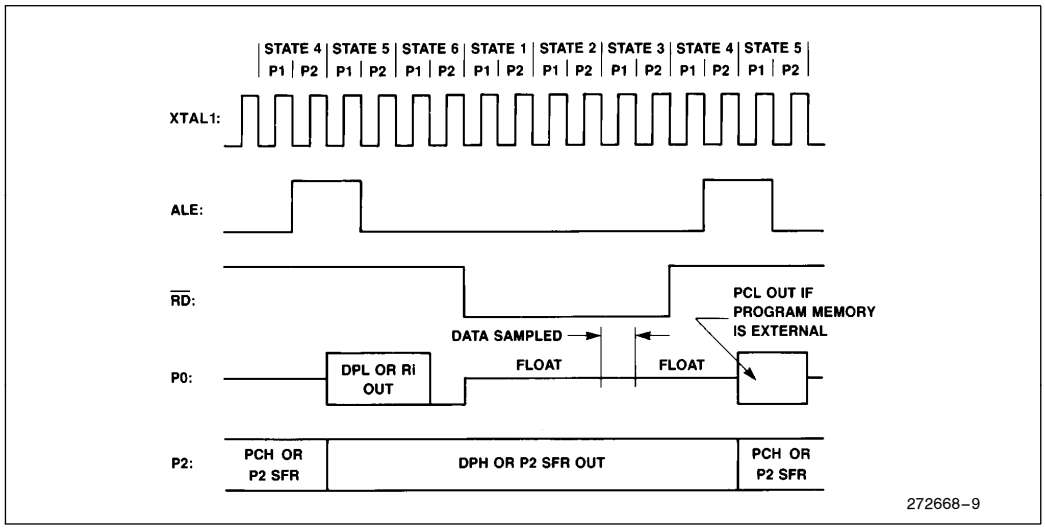


Figure 7. External Data Memory Read Cycle

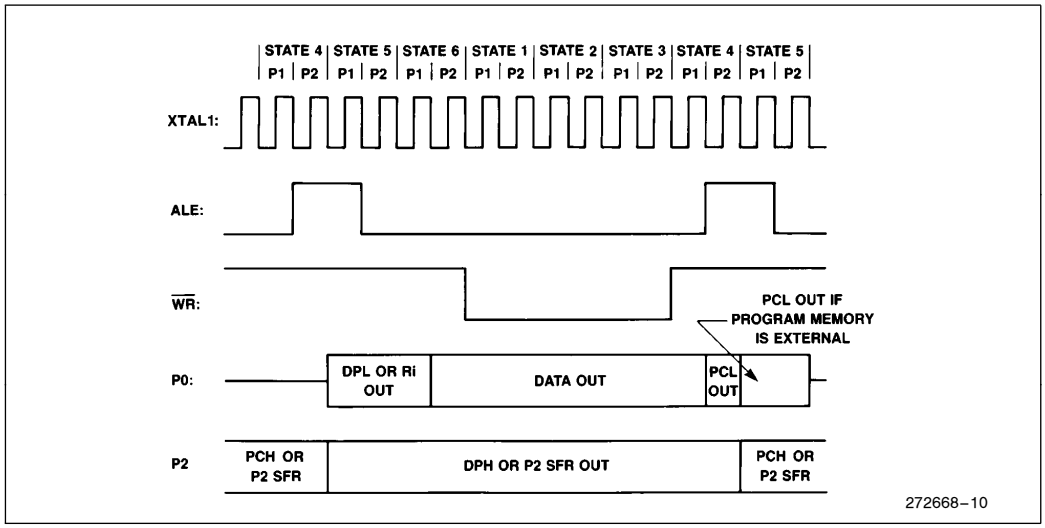


Figure 8. External Data Memory Write Cycle

When EXTRAM = 1, AUXR SFR bit 1 is set, and 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. The Port 2 drivers use the strong pullups during the entire time that they are emitting address bits that are 1s. This occurs when the MOVX @ DPTR instruction is executed. During this time the Port 2 latch (the Special Function Register) does not have to contain 1s, and the contents of the Port 2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the Port 2 SFR will reappear in the next cycle.

When EXTRAM = 1, AUXR SFR bit 1 is set, and 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. In this case, Port 2 pins can be used to page the external data memory.

In either case, the low byte of the address is time-multiplexed with the data byte on Port 0. The ADDRESS/DATA signal drives both FETs in the Port 0 output buffers. Thus, in external bus mode the Port 0 pins are not open-drain outputs and do not require external pullups. The ALE (Address Latch Enable) 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 ($\overline{RD}$) is deactivated.

During any access to external memory, the CPU writes 0FFH to the Port 0 latch (the Special Function Register), thus obliterating the information in the Port 0 SFR. Also, a MOV P0 instruction must not take place during external memory accesses. 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 two conditions:

1. Whenever signal $\overline{EA}$ is active, or
2. Whenever the program counter (PC) contains an address greater than 1FFFH (8K) for the 8XC51RA or 3FFFH (16K) for the 8XC51RB, or 7FFFH (32K) for the 87C51RC.

This requires that the ROMless versions have $\overline{EA}$ wired to V_{SS} enable the lower 8K, 16K, or 32K 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 with the Port 2 drivers using the strong pullups to emit bits that are 1s.

5.0 TIMERS/COUNTERS

The C51RX has three 16-bit Timer/Counters: Timer 0, Timer 1, and Timer 2. Each consists of two 8-bit registers, THx and TLx, (x = 0, 1, and 2). All three can be configured to operate either as timers or event counters.

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

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin—T0, T1, or T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is $1/_{24}$ of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

In addition to the Timer or Counter selection, Timer 0 and Timer 1 have four operating modes from which to select: Modes 0 – 3. Timer 2 has three modes of operation: Capture, Auto-Reload, and Baud Rate Generator.

5.1 Timer 0 and Timer 1

The Timer or Counter function is selected by control bits C/$\overline{T}$ in the Special Function Register TMOD (Table 5). 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. Mode 3 operation is different for the two timers.

MODE 0

Either Timer 0 or Timer 1 in Mode 0 is an 8-bit Counter with a divide-by-32 prescaler. Figure 9 shows the Mode 0 operation for either timer.

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 TFx. The counted input is enabled to the Timer when TRx = 1 and either GATE = 0 or $\overline{INTx}$ = 1. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INTx}$, to facilitate pulse width measurements). TRx and TFx are

control bits in SFR TCON (Table 7). The GATE bit is in TMOD. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

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

MODE 1

Mode 1 is the same as Mode 0, except that the Timer register uses all 16 bits. Refer to Figure 10. In this mode, THx and TLx are cascaded; there is no prescaler.

MODE 2

Mode 2 configures the Timer register as an 8-bit Counter (TLx) with automatic reload, as shown in Figure 11. Overflow from TLx not only sets TFx, but also reloads TLx with the contents of THx, which is preset by software. The reload leaves THx unchanged.

Table 6. TMOD: Timer/Counter Mode Control Register

TMOD Address = 89H Reset Value = 0000 0000B

Not Bit Addressable

	TIMER 1				TIMER 0			
	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Bit	7	6	5	4	3	2	1	0

Symbol	Function
GATE	Gating control when set. Timer/Counter 0 or 1 is enabled only while $\overline{INT0}$ or $\overline{INT1}$ pin is high and TR0 or TR1 control pin is set. When cleared, Timer 0 or 1 is enabled whenever TR0 or TR1 control bit is set.
C/$\overline{T}$	Timer or Counter Selector. Clear for Timer operation (input from internal system clock). Set for Counter operation (input from T0 or T1 input pin).

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

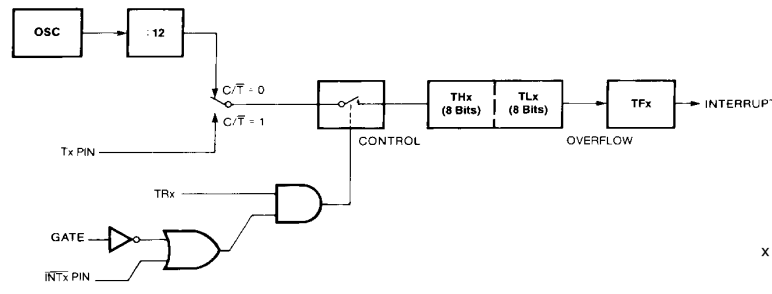


Figure 9. Timer/Counter 0 or 1 in Mode 0: 13-Bit Counter

Table 7. TCON: Timer/Counter Control Register

TCON Address = 88H Reset Value = 0000 0000B

Bit Addressable

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

Symbol	Function
TF1	Timer 1 overflow Flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter 1 on/off.
TF0	Timer 0 overflow Flag. Set by hardware on Timer/Counter 0 overflow. Cleared by hardware when processor vectors to interrupt routine.
TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter 0 on/off.
IE1	Interrupt 1 flag. Set by hardware when external interrupt 1 edge is detected (transmitted or level-activated). Cleared when interrupt processed only if transition-activated.
IT1	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupt 1.
IE0	Interrupt 0 flag. Set by hardware when external interrupt 0 edge is detected (transmitted or level-activated). Cleared when interrupt processed only if transition-activated.
IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupt 0.

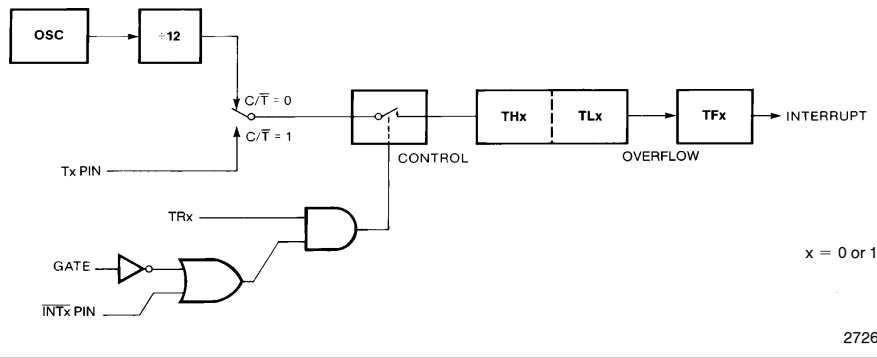


Figure 10. Timer/Counter 0 or 1 in Mode 1: 16-Bit Counter

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 12. TL0 uses the Timer 0 control bits: C/T, GATE, TR0, INT0, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus TH0 now controls the Timer 1 interrupt.

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

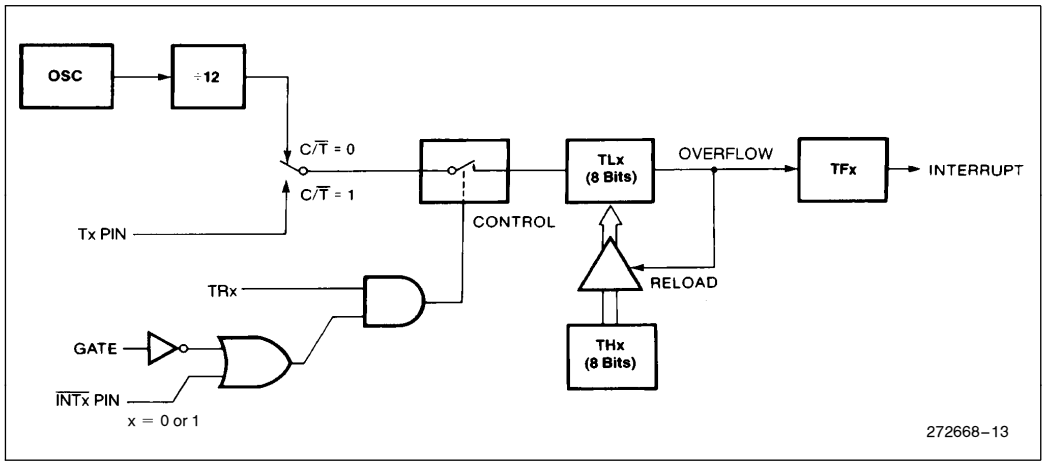


Figure 11. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload

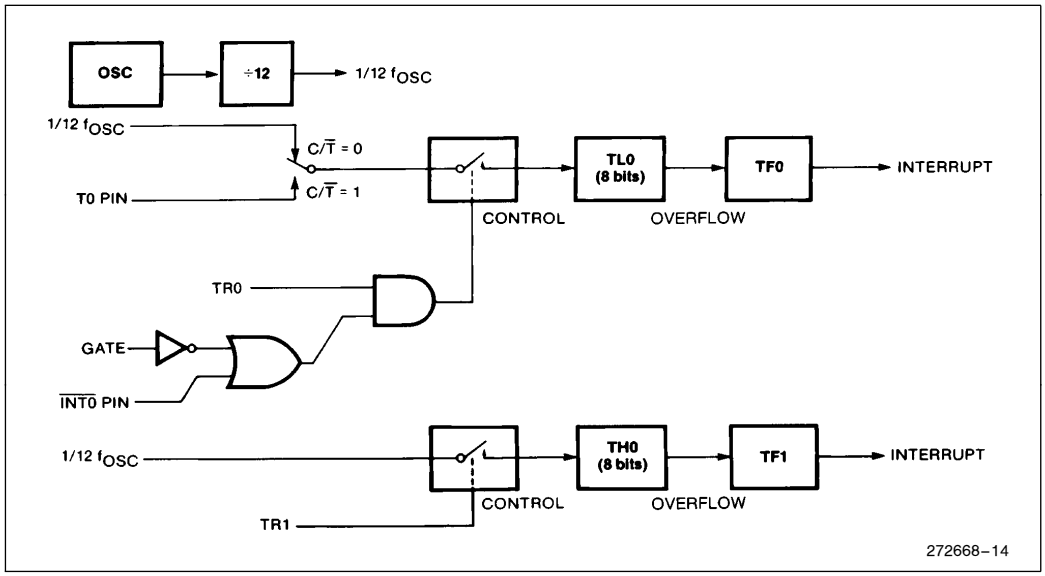


Figure 12. Timer/Counter 0 Mode 3: Two 8-Bit Counters

5.2 Timer 2

Timer 2 is a 16-bit Timer/Counter which can operate either as a timer or as an event counter. This is selected by bit C/$\overline{T2}$ in the Special Function Register T2CON (Table 9). It has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON as shown in Table 8.

Table 8. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	T2*OE	TR2	Mode
0	0	0	1	16-Bit Auto-Reload
0	1	0	1	16-Bit Capture
1	X	X	1	Baud Rate Generator
X	0	1	1	Clock-Out on P1.0
X	X	X	0	Timer Off

Table 9. T2CON: Timer/Counter 2 Control Register

T2CON Address = 0C8H Reset Value = 0000 0000B

Bit Addressable

	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	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. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	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	Start/stop control for Timer 2. A logic 1 starts the timer.
C/$\overline{T2}$	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12 or OSC/2 in baud rate generator mode). 1 = External event counter (falling edge triggered).
CP/$\overline{RL2}$	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.

CAPTURE MODE

In the capture mode there are two options selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 still does the above, but with the added feature that a 1-to-0 tran-

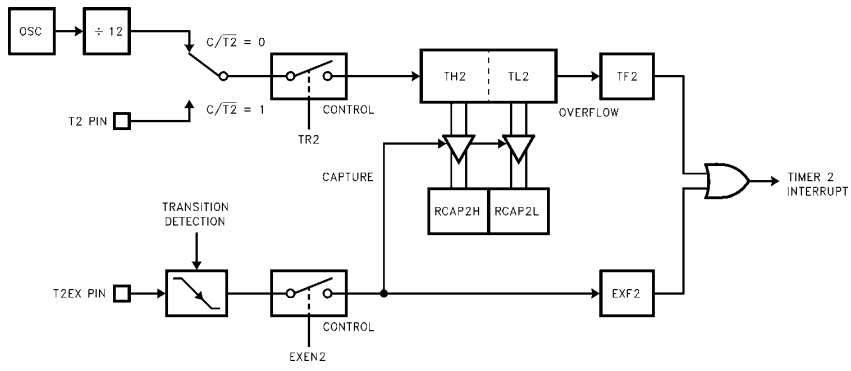


Figure 13. Timer 2 in Capture Mode

sition at external input T2EX causes the current value in the Timer 2 registers, TH2 and TL2, to be captured into registers RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 13.

AUTO-RELOAD MODE (UP OR DOWN COUNTER)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by a bit named DCEN (Down Counter Enable) located in the SFR T2MOD (see Table 10). Upon reset the DCEN bit is set to 0 so that Timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down depending on the value of the T2EX pin.

Figure 14 shows Timer 2 automatically counting up when DCEN = 0. In this mode there are two options selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Either the TF2 or EXF2 bit can generate the Timer 2 interrupt if it is enabled.

Table 10. T2MOD: Timer 2 Mode Control Register

T2MOD Address = 0C9H Reset Value = XXXX XX00B

Not Bit Addressable

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	T2OE	DCEN

Symbol	Function
—	Not implemented, reserved for future use.*
T2OE	Timer 2 Output Enable bit.
DCEN	Down Count Enable bit. When set, this allows Timer 2 to be configured as an up/down counter.

*User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

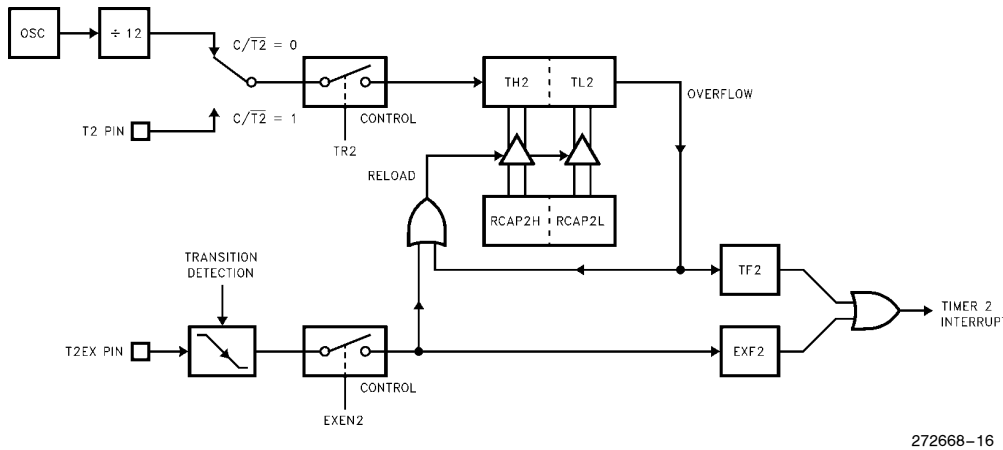


Figure 14. Timer 2 Auto Reload Mode (DCEN = 0)

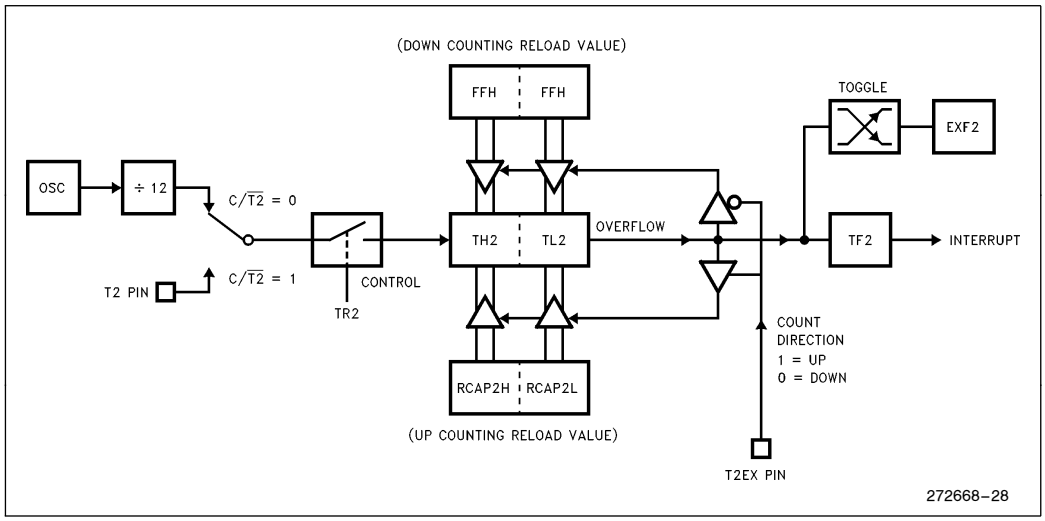


Figure 15. Timer 2 Auto Reload Mode (DCEN = 1)

Setting the DCEN bit enables Timer 2 to count up or down as shown in Figure 15. In this mode the T2EX pin controls the direction of count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit which can then generate an interrupt if it is enabled. This overflow also causes a the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. Now the timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows. This bit can be used as a 17th bit of resolution if desired. In this operating mode, EXF2 does not generate an interrupt.

BAUD RATE GENERATOR MODE

The baud rate generator mode is selected by setting the RCLK and/or TCLK bits in T2CON. Timer 2 in this mode will be described in conjunction with the serial port.

PROGRAMMABLE CLOCK OUT

A 50% duty cycle clock can be programmed to come out on P1.0. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed (1) to input the external clock for Timer/Counter 2 or (2) to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/T2 (in T2CON) must be cleared and bit T2OE in T2MOD must be set. Bit TR2 (T2CON.2) also must be set to start the timer (see Table 9 for operating modes).

The Clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L) as shown in this equation:

$$\text{Clock-out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times (65536 - \text{RCAP2H, RCAP2L})}$$

In the Clock-Out mode Timer 2 roll-overs will not generate an interrupt. This is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and Clock-out frequencies cannot be determined independently of one another since they both use the values in RCAP2H and RCAP2L.

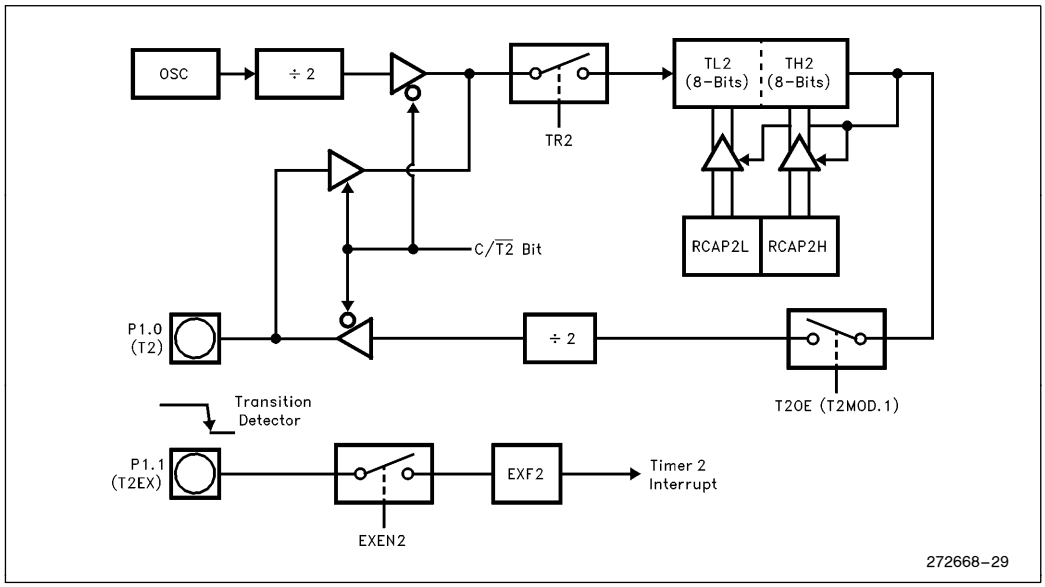


Figure 16. Timer 2 in Clock-Out Mode

6.0 SERIAL INTERFACE

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a 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 through Special Function Register SBUF. Actually, SBUF is two separate registers, a transmit buffer and a receive buffer. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port control and status register is the Special Function Register SCON, shown in Table 11. This register contains the mode selection bits (SM0 and SM1); the SM2 bit for the multiprocessor modes (see Multiprocessor Communications section); the Receive Enable bit (REN); the 9th data bit for transmit and receive (TB8 and RB8); and the serial port interrupt bits (TI and RI).

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). Refer to Figure 17. 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 SCON, while the stop bit is ignored. (The validity of the stop bit can be checked with Framing Error Detection.) The baud rate is programmable to either $1/32$ or $1/64$ the oscillator frequency.

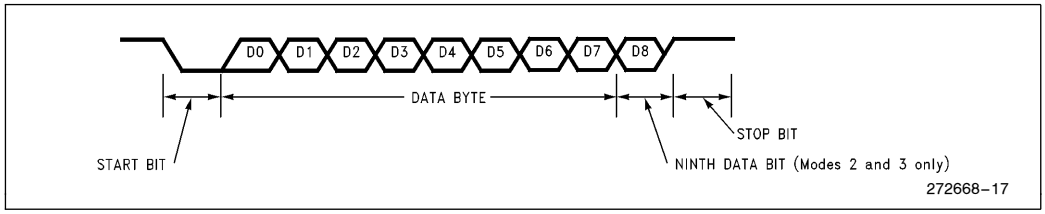


Figure 17. Data Frame: Modes 1, 2 and 3

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 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. For more detailed information on each serial port mode, refer to the "Hardware Description of the 8051, 8052, and 80C51."

6.1 Framing Error Detection

Framing Error Detection allows the serial port to check for valid stop bits in modes 1, 2, or 3. A missing stop bit can be caused, for example, by noise on the serial lines, or transmission by two CPUs simultaneously.

If a stop bit is missing, a Framing Error bit FE is set. The FE bit can be checked in software after each reception to detect communication errors. Once set, the FE bit must be cleared in software. A valid stop bit will not clear FE.

The FE bit is located in SCON and shares the same bit address as SM0. Control bit SMOD0 in the PCON register (location PCON.6) determines whether the SM0 or FE bit is accessed. If SMOD0 = 0, then accesses to SCON.7 are to SM0. If SMOD0 = 1, then accesses to SCON.7 are to FE.

6.2 Multiprocessor Communications

Modes 2 and 3 provide a 9-bit mode to facilitate multiprocessor comunication. The 9th bit allows the controller to distinguish between address and data bytes. The 9th bit is set to 1 for address bytes and set to 0 for data bytes. When receiving, the 9th bit goes into RB8 in SCON. When transmitting, TB8 is set or cleared in software.

The serial port can be programmed such that when the stop bit is received the serial port interrupt will be activated only if the received byte is an address byte (RB8 = 1). This feature is enabled by setting the SM2 bit in SCON. A way to use this feature in multiprocessor systems is as follows.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. Remember, an address byte has its 9th bit set to 1, whereas a data byte has its 9th bit set to 0. All the slave processors should have their SM2 bits set to 1 so they will only be interrupted by an address byte. In fact, the C51RX has an Automatic Address Recognition feature which allows only the addressed slave to be interrupted. That is, the address comparison occurs in hardware, not software. (On the 8051 serial port, an address byte interrupts all slaves for an address comparison.)

The addressed slave's software then clears its SM2 bit and prepares to receive the data bytes that will be coming. The other slaves are unaffected by these data bytes. They are still waiting to be addressed since their SM2 bits are all set.

6.3 Automatic Address Recognition

Automatic Address Recognition reduces the CPU time required to service the serial port. Since the CPU is only interrupted when it receives its own address, the software overhead to compare addresses is eliminated. With this feature enabled in one of the 9-bit modes, the Receive Interrupt (RI) flag will only get set when the received byte corresponds to either a Given or Broadcast address.

The feature works the same way in the 8-bit mode (Mode 1) as in the 9-bit modes, except that the stop bit takes the place of the 9th data bit. If SM2 is set, the RI flag is set only if the received byte matches the Given or Broadcast Address and is terminated by a valid stop bit. Setting the SM2 bit has no effect in Mode 0.

The master can selectively communicate with groups of slaves by using the Given Address. Addressing all slaves at once is possible with the Broadcast Address. These addresses are defined for each slave by two Special Function Registers: SADDR and SADEN.

A slave's individual address is specified in SADDR. SADEN is a mask byte that defines don't-cares to form the Given Address. These don't-cares allow flexibility in the user-defined protocol to address one or more slaves at a time. The following is an example of how the user could define Given Addresses to selectively address different slaves.

Slave 1:

SADDR	=	1111	0001
SADEN	=	1111	1010
GIVEN	=	1111	0X0X

Slave 2:

SADDR	=	1111	0011
SADEN	=	1111	1001
GIVEN	=	1111	0XX1

Table 11. SCON: Serial Port Control Register

SCON Address = 98H Reset Value = 0000 0000B

Bit Addressable

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

Bit: (SMOD0 = 0/1)*

Symbol	Function
FE	Framing Error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0* bit must be set to enable access to the FE bit.
SM0	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)
SM1	Serial Port Mode Bit 1

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

SM2	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.
REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
RB8	In modes 2 and 3, 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	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	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.

NOTE:
*SMOD0 is located at PCON6.
**F_{OSC} = oscillator frequency

The SADEN byte are selected such that each slave can be addressed separately. Notice that bit 1 (LSB) is a don't-care for Slave 1's Given Address, but bit 1 = 1 for Slave 2. Thus, to selectively communicate with just Slave 1 the master must send an address with bit 1 = 0 (e.g. 1111 0000).

Similarly, bit 2 = 0 for Slave 1, but is a don't-care for Slave 2. Now to communicate with just Slave 2 an address with bit 2 = 1 must be used (e.g. 1111 0111).

Finally, for a master to communicate with both slaves at once the address must have bit 1 = 1 and bit 2 = 0. Notice, however, that bit 3 is a don't-care for both slaves. This allows two different addresses to select both slaves (1111 0001 or 1111 0101). If a third slave was added that required its bit 3 = 0, then the latter address could be used to communicate with Slave 1 and 2 but not Slave 3.

The master can also communicate with all slaves at once with the Broadcast Address. It is formed from the logical OR of the SADDR and SADEN registers with zeros defined as don't-cares. The don't-cares also allow

flexibility in defining the Broadcast Address, but in most applications a Broadcast Address will be 0FFH.

SADDR and SADEN are located at address A9H and B9H, respectively. On reset, the SADDR and SADEN registers are initialized to 00H which defines the Given and Broadcast Addresses as XXXX XXXX (all don't-cares). This assures the C51RX serial port to be backwards compatibility with other MCS®-51 products which do not implement Automatic Addressing.

6.4 Baud Rates

The baud rate in Mode 0 is fixed:

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

The baud rate in Mode 2 depends on the value of bit SMOD1 in Special Function Register PCON. If SMOD1 = 0 (which is the value on reset), the baud rate is $1/64$ the oscillator frequency. If SMOD1 = 1, the baud rate is $1/32$ the oscillator frequency.

$$\text{Mode 2 Baud Rate} = 2^{\text{SMOD1}} \times \frac{\text{Oscillator Frequency}}{64}$$

The baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate, or by Timer 2 overflow rate, or by both (one for transmit and the other for receive).

6.5 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 SMOD1 as follows:

$$\text{Modes 1 and 3 Baud Rate} = 2^{\text{SMOD1}} \times \frac{\text{Timer 1 Overflow Rate}}{32}$$

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 most applications, it is configured for "timer" operation in the auto-reload mode (high nibble of TMOD = 0010B). In this case, the baud rate is given by the formula:

$$\text{Modes 1 and 3 Baud Rate} = \frac{2^{\text{SMOD1}} \times \text{Oscillator Frequency}}{32 \times 12 \times [256 - (\text{TH1})]}$$

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

Table 12 lists various commonly used baud rates and how they can be obtained from Timer 1.

6.6 Using Timer 2 to Generate Baud Rates

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 8). Note that 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 18.

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

Table 12. Timer 1 Generated Commonly Used Baud Rates

Baud Rate	f_{OSC}	SMOD	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

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

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

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

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

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

Timer 2 as a baud rate generator is shown in Figure 18. This figure is valid only if RCLK and/or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the Timer is being incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read, but shouldn't be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Table 13 lists commonly used baud rates and how they can be obtained from Timer 2.

Table 13. Timer 2 Generated Commonly Used Baud Rates

Baud Rate	Osc Freq	Timer 2	
		RCAP2H	RCAP2L
375K	12 MHz	FF	FF
9.6K	12 MHz	FF	D9
4.8K	12 MHz	FF	B2
2.4K	12 MHz	FF	64
1.2K	12 MHz	FE	C8
300	12 MHz	FB	1E
110	12 MHz	F2	AF
300	6 MHz	FD	8F
110	6 MHz	F9	57

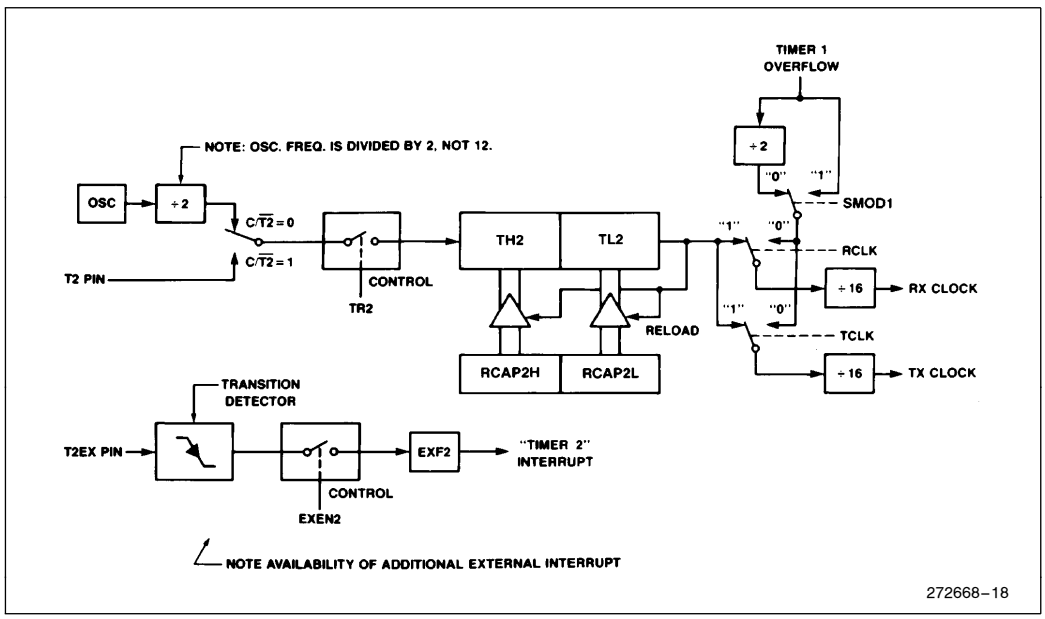


Figure 18. Timer 2 in Baud Rate Generator Mode

7.0 HARDWARE WATCHDOG TIMER (ONE-TIME ENABLED WITH RESET-OUT)

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upset. The WDT consists of a 14-bit counter and the WatchDog Timer ReSeT (WDTRST) SFR. The WDT is default to disable from exiting reset. To enable the WDT, user must write 01EH and 0E1H in sequence to the WDTRST, SFR location 0A6H. When WDT is enabled, it will increment every machine cycle while the oscillator is running and there is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST-pin.

7.1 Using the WDT

To enable the WDT, user must write 01EH and 0E1H in sequence to the WDTRST, SFR location 0A6H. When WDT is enabled, the user needs to service it by writing to 01EH and 0E1H to WDTRST to avoid WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH) and this will reset the device. When WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycle. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST-pin. The RESET pulse duration is 98 x T_{OSC}, where $T_{OSC} = 1/F_{OSC}$. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

7.2 WDT during Power Down and Idle

In Power Down mode the oscillator stops, which means the WDT also stops. While in Power Down mode the user does not need to service the WDT. There are 2 methods of exiting Power Down mode: by a hardware reset or via a level activated external interrupt which is enabled prior to entering Power Down mode. When Power Down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the C51RX is reset. Exiting Power Down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power Down.

To ensure that the WDT does not overflow within a few states of exiting of powerdown, it is best to reset the WDT just before entering powerdown.

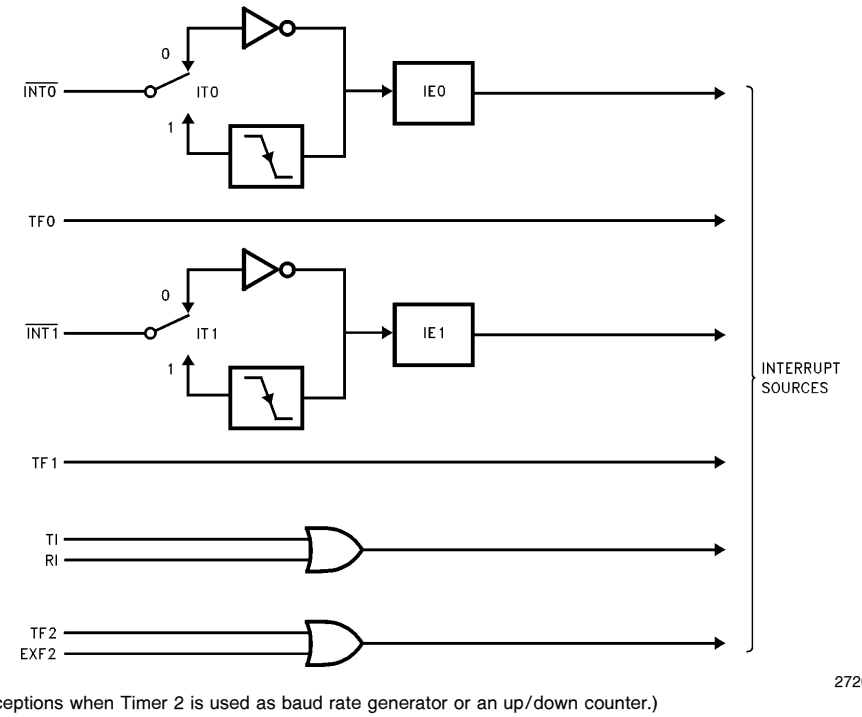
In the Idle mode, the oscillator continues to run. To prevent the WDT from resetting the C51RX while in Idle mode, the user should always set up a timer that will periodically exit Idle, service the WDT, and re-enter Idle mode.

8.0 INTERRUPTS

The C51RX has a total of 6 interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 19.

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

Each of these interrupts will be briefly described followed by a discussion of the interrupt enable bits and the interrupt priority levels.



(See exceptions when Timer 2 is used as baud rate generator or an up/down counter.)

Figure 19. Interrupt Sources

8.1 External Interrupts

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. If ITx = 0, external interrupt x is triggered by a detected low at the $\overline{INTx}$ pin. If ITx = 1, external interrupt x is negative edge-triggered. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. These flags are cleared by hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

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 external interrupt $\overline{INT0}$ or $\overline{INT1}$ is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

8.2 Timer Interrupts

Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1 in register TCON, which are set by a rollover in their respective Timer/Counter registers (except see Timer 0 in Mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

Timer 2 Interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared in software.

8.3 Serial Port Interrupt

The serial port interrupt is generated by the logical OR of bits RI and TI in register SCON. Neither of these flags is cleared by hardware when the service routine is vectored to. The service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared in software.

8.4 Interrupt Enable

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in the Interrupt Enable (IE) register. (See Table 14.) Note that IE also contains a global disable bit, EA. If EA is set (1), the interrupts are individually enabled or disabled by their corresponding bits in IE. If EA is clear (0), all interrupts are disabled.

8.5 Priority Level Structure

Each interrupt source can also be individually programmed to one of two priority levels, by setting or clearing a bit in the Interrupt Priority (IP) register shown in Table 15. A low-priority interrupt can itself be interrupted by a higher priority interrupt, but not by another low-priority interrupt. A high priority interrupt cannot be interrupted by any other interrupt source.

Table 14. IE: Interrupt Enable Register

IE Address = 0A8H Reset Value = 0X00 0000B

Bit Addressable

	EA	—	ET2	ES	ET1	EX1	ET0	EX0
Bit	7	6	5	4	3	2	1	0

Enable Bit = 1 enables the interrupt.
Enable Bit = 0 disables it.

Symbol	Function
EA	Global disable bit. If EA = 0, all Interrupts are disabled. If EA = 1, each Interrupt can be individually enabled or disabled by setting or clearing its enable bit.
—	Not implemented, reserved for future use.*
ET2	Timer 2 interrupt enable bit.
ES	Serial Port interrupt enable bit.
ET1	Timer 1 interrupt enable bit.
EX1	External interrupt 1 enable bit.
ET0	Timer 0 interrupt enable bit.
EX0	External interrupt 0 enable bit.

Table 15. IP: Interrupt Priority Registers

IP Address = 0B8H Reset Value = XX00 0000B

Bit Addressable

	—	—	PT2	PS	PT1	PX1	PT0	PX0
Bit	7	6	5	4	3	2	1	0

Priority Bit = 1 assigns high priority
Priority Bit = 0 assigns low priority

Symbol	Function
—	Not implemented, reserved for future use.*
PT2	Timer 2 interrupt priority bit.
PS	Serial Port interrupt priority bit.
PT1	Timer 1 interrupt priority bit.
PX1	External interrupt 1 priority bit.
PT0	Timer 0 interrupt priority bit.
PX0	External interrupt 0 priority bit.

NOTE:
*User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

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 shown in Table 16.

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

Table 16. Interrupt Priority within Level Polling Sequence

1 (Highest)	$\overline{INT0}$
2	Timer 0
3	$\overline{INT1}$
4	Timer 1
5	Serial Port
6 (Lowest)	Timer 2

8XC51RA/RB/RC Interrupt Priority Structure

In the 8XC51RA/RB/RC, a second Interrupt Priority register (IPH) has been added, increasing the number of priority levels to four. Table 17 shows this second register. The added register becomes the MSB of the priority select bits and the existing IP register acts as the LSB. This scheme maintains compatibility with the rest of the MCS-51 family. Table 18 shows the bit values and priority levels associated with each combination.

Table 18. Priority Level Bit Values

Priority Bits		Interrupt Priority Level
IPH.x	IP.x	
0	0	Level 0 (Lowest)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (Highest)

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 Timer 2 interrupt cycle is slightly 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.

Table 17. IPH: Interrupt Priority High Register

IPH Address = 0B7H Reset Value = XX00 0000B

Not Bit Addressable

	—	—	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
Bit	7	6	5	4	3	2	1	0

Symbol	Function
—	Not implemented, reserved for future use.
PT2H	Timer 2 interrupt priority high bit.
PSH	Serial Port interrupt priority high bit.
PT1H	Timer 1 interrupt priority high bit.
PX1H	External interrupt 1 priority high bit.
PT0H	Timer 0 interrupt priority high bit.
PX0H	External interrupt priority high bit.

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 write 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 the interrupt flag for a *level-sensitive* external interrupt is active but not being responded to for 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 20.

Note that if an interrupt of a higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 20, then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine having been executed.

Thus the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. 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 in Table 19.

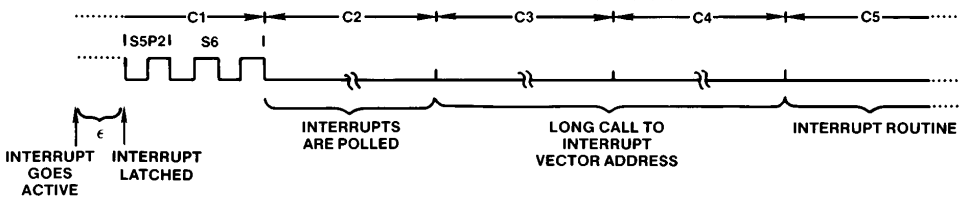
Table 19. Interrupt Vector Address

Interrupt Source	Interrupt Request Bits	Cleared by Hardware	Vector Address
$\overline{\text{INT0}}$	IE0	No (level) Yes (trans.)	0003H
TIMER 0	TF0	Yes	000BH
$\overline{\text{INT1}}$	IE1	No (level) Yes (trans.)	0013H
TIMER 1	TF1	Yes	001BH
SERIAL PORT	RI, TI	No	0023H
TIMER 2	TF2, EXF2	No	002BH

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

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking interrupt was still in progress.

Note that the starting addresses of consecutive interrupt service routines are only 8 bytes apart. That means if consecutive interrupts are being used (IE0 and TF0, for example, or TF0 and IE1), and if the first interrupt routine is more than 7 bytes long, then that routine will have to execute a jump to some other memory location where the service routine can be completed without overlapping the starting address of the next interrupt routine.



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

Figure 20. Interrupt Response Timing Diagram

8.6 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 to be executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles elapses between activation of an external interrupt request and the beginning of execution of the service routine's first instruction. Figure 20 shows interrupt response timing.

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 write to IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one or 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.

9.0 RESET

The reset input is the RST pin, which has a Schmitt Trigger input. 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 21.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. ALE and PSEN 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. The port pins are driven to their reset state as soon as a valid high is detected on the RST pin, regardless of whether the clock is running.

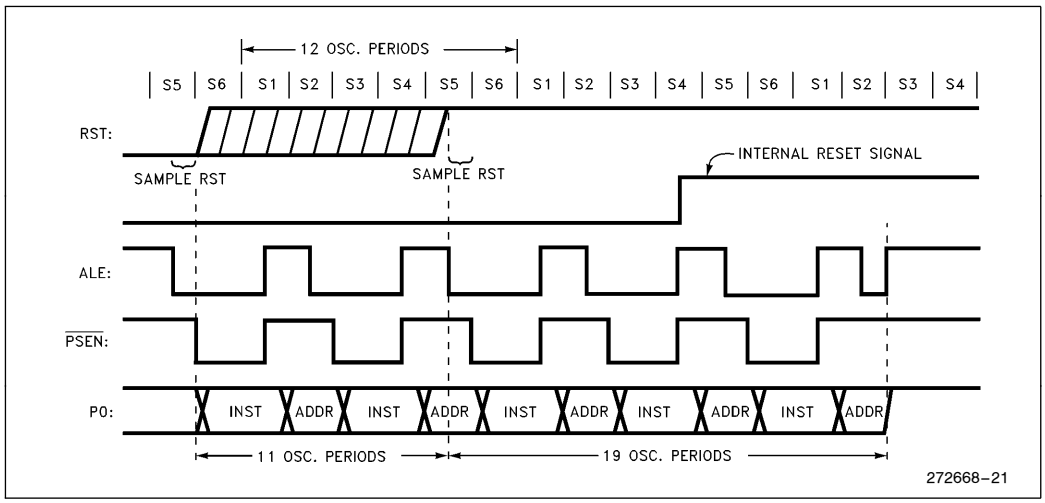


Figure 21. Reset Timing

While the RST pin is high, the port pins, 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 C51RX.

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 redefines all the SFRs. Table 1 lists the SFRs and their reset values. The internal RAM is not affected by reset. On power up the RAM content is indeterminate.

9.1 Power-On Reset

For CHMOS devices, when VCC is turned on, an automatic reset can be obtained by connecting the RST pin to VCC through a 1 μF capacitor (Figure 22). The CHMOS devices do not require an external resistor like the HMOS devices because they have an internal pull-down on the RST pin.

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.

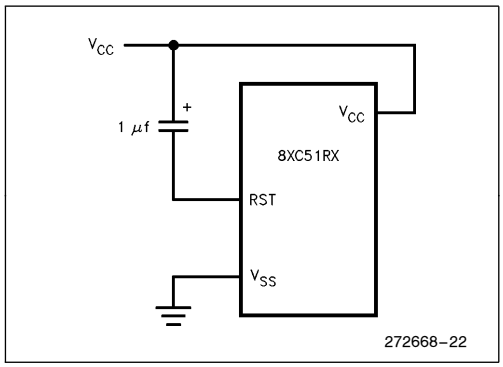


Figure 22. Power on Reset Circuitry

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

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 that 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.

10.0 POWER-SAVING MODES OF OPERATION

For applications where power consumption is critical, the C51RX provides two power reducing modes of operation: Idle and Power Down. The input through which backup power is supplied during these operations is V_{CC}. Figure 23 shows the internal circuitry which implements these features. In the Idle mode (IDL = 1), the oscillator continues to run and the Interrupt, Serial Port, PCA, 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 (Table 20).

10.1 Idle Mode

An instruction that sets PCON.0 causes that to be the last instruction executed before going into the Idle mode. In the Idle mode, the internal clock signal is gated off to the CPU, but not to the Interrupt, Timer, and Serial Port functions. The PCA can be programmed either to pause or continue operating during Idle (refer to the PCA section for more details). 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 Mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into Idle.

The flag bits (GF0 and GF1) can be used to give an indication if an interrupt occurred during normal operation or during 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 needs to 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 21, two or three machine cycles of program execution may take place before the

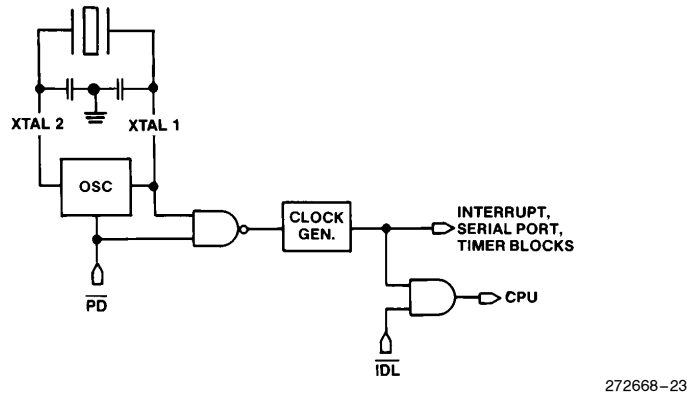


Figure 23. Idle and Power Down Hardware

Table 20. PCON: Power Control Register

PCON Address = 87H Reset Value = 00XX 0000B

Not Bit Addressable

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

Symbol	Function
SMOD1	Double Baud rate bit. When set to a 1 and Timer 1 is used to generate baud rates, and the Serial Port is used in modes 1, 2, or 3.
SMOD0	When set, Read/Write accesses to SCON.7 are to the FE bit. When clear, Read/Write accesses to SCON.7 are to the SM0 bit.
—	Not implemented, reserved for future use.*
POF	Power Off Flag. Set by hardware on the rising edge of V_{CC}. Set or cleared by software. This flag allows detection of a power failure caused reset. V_{CC} must remain above 3V to retain this bit.
GF1	General-purpose flag bit.
GF0	General-purpose flag bit.
PD	Power Down bit. Setting this bit activates Power Down operation.
IDL	Idle mode bit. Setting this bit activates idle modes operation. If 1s are written to PD and IDL at the same time, PD takes precedence.

NOTE:
*User software should not write 1s to unimplemented bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate

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 be one that writes to a port pin or to external Data RAM.

10.2 Power Down Mode

An instruction that sets PCON.1 causes that to be the last instruction executed before going into the Power Down mode. In this mode the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and Special Function Registers are held. The port pins output the values held by their respective SFRs, and ALE and $\overline{PSEN}$ output lows. In Power Down V_{CC} can be reduced to as low as 2V. Care must be taken, however, to ensure that V_{CC} is not reduced before Power Down is invoked.

The C51RX can exit Power Down with either a hardware reset or external interrupt. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10 msec).

With an external interrupt, $\overline{INT0}$ or $\overline{INT1}$ must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator and bringing the pin back high completes the exit. After the RETI instruction is executed in the interrupt service routine, the next instruction will be the one following the instruction that put the device in Power Down.

10.3 Power Off Flag

The Power Off Flag (POF) located at PCON.4, is set by hardware when V_{CC} rises from 0 to 5 Volts. POF can also be set or cleared by software. This allows the user to distinguish between a "cold start" reset and a "warm start" reset.

A cold start reset is one that is coincident with V_{CC} being turned on to the device after it was turned off. A warm start reset occurs while V_{CC} is still applied to the device and could be generated, for example, by a Watchdog Timer or an exit from Power Down.

Immediately after reset, the user's software can check the status of the POF bit. POF = 1 would indicate a cold start. The software then clears POF and commences its tasks. POF = 0 immediately after reset would indicate a warm start.

V_{CC} must remain above 3 volts for POF to retain a 0.

11.0 ALE DISABLE

ALE operation can be disabled by setting bit 0 of AUXR SFR (08EH). With this bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

12.0 EPROM VERSIONS

The 8XC51RA/RB/RC uses the Improved "Quick-Pulse" programming algorithm. These devices program at V_{PP} = 12.75V (and V_{CC} = 5.0V) using a series of five 100 μs $\overline{PROG}$ pulses per byte programmed. This results in a total programming time of approximately 5 seconds for the 87C51RA's 8 Kbytes, 10 seconds for the 87C51RB's 16 Kbytes, and 20 seconds for the 87C51RC's 32 Kbytes.

Exposure to Light: The EPROM window must be covered with an opaque label when the device is in operation. This is not so much to protect the EPROM array from inadvertent erasure, but to protect the RAM and other on-chip logic. Allowing light to impinge on the silicon die while the device is operating can cause logical malfunction.

13.0 PROGRAM MEMORY LOCK

In some microcontroller applications, it is desirable that the Program Memory be secure from software piracy. The C51RX has varying degrees of program protection depending on the device. Table 21 outlines the lock schemes available for each device.

Encryption Array: Within the EPROM/ROM is an array of encryption bytes that are initially unprogrammed (all 1's). For EPROM devices, the user can program the encryption array to encrypt the program code bytes during EPROM verification. For ROM devices, the user submits the encryption array to be programmed by the factory. If an encryption array is submitted, LB1 will also be programmed by the factory. The encryption array is not available without the Lock Bit. Program

code verification is performed as usual, except that each code byte comes out exclusive-NOR'ed (XNOR) with one of the key bytes. Therefore, to read the ROM/EPROM code, the user has to know the encryption key bytes in their proper sequence.

Unprogrammed bytes have the value 0FFH. If the Encryption Array is left unprogrammed, all the key bytes have the value 0FFH. Since any code byte XNOR'ed with 0FFH leaves the byte unchanged, leaving the Encryption Array unprogrammed in effect bypasses the encryption feature.

When using the encryption array feature, one important factor should be considered. If a code byte has the value 0FFH, verifying the byte will produce the encryption byte value. If a large block (>64 bytes) of code is left unprogrammed, a verification routine will display the encryption array contents. For this reason all unused code bytes should be programmed with some value other than 0FFH, and not all of them the same value. This will ensure maximum program protection.

Program Lock Bits: Also included in the Program Lock scheme are Lock Bits which can be enabled to provide varying degrees of protection. Table 22 lists the Lock Bits and their corresponding influence on the microcontroller. Refer to Table 21 for the Lock Bits available on the various products. The user is responsible for programming the Lock Bits on EPROM devices. On ROM devices, LB1 is automatically set by the factory when the encryption array is submitted. The Lock Bit is not available without the encryption array on ROM devices.

Table 21. C51RX Program Protection

Device	Lock Bits	Encrypt Array
83C51RA	None	None
83C51RB	LB1	64 Bytes
83C51RC	LB1	64 Bytes
87C51RA	LB1, LB2, LB3	64 Bytes
87C51RB	LB1, LB2, LB3	64 Bytes
87C51RC	LB1, LB2, LB3	64 Bytes

14.0 ONCE MODE

The ONCE (ON-Circuit Emulation) mode facilitates testing and debugging of systems using the C51RX without having to remove the device from the circuit. The ONCE mode is invoked by:

1. Pulling ALE low while the device is in reset and $\overline{PSEN}$ is high;
2. Holding 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, 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.

Normal operation is restored after a valid reset is applied.

Table 22. Lock Bits

	Program Lock Bits			Protection Type
	LB1	LB2	LB3	
1	U	U	U	No program lock features enabled. (Code verify will still be encrypted by the encryption array if programmed.)
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 EPROM 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.

P = Programmed
U = Unprogrammed
Any other combination of the Lock Bits is not defined.

15.0 ON-CHIP OSCILLATOR

The on-chip oscillator for the CHMOS devices, shown in Figure 24, consists of a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator. In this application the crystal is operating in its fundamental response mode as an inductive reactance in parallel resonance with capacitance external to the crystal (Figure 25).

The oscillator on the CHMOS devices can be turned off under software control by setting the PD bit in the PCON register. The feedback resistor R_f in Figure 24 consists of paralleled n- and p-channel FETs controlled by the PD bit, such that R_f is opened when PD = 1. The diodes D1 and D2, which act as clamps to V_{CC} and V_{SS}, are parasitic to the R_f FETs.

The crystal specifications and capacitance values (C1 and C2 in Figure 25) are not critical. 30 pF can be used in these positions at any frequency with good quality crystals. In general, crystals used with these devices typically have the following specifications:

ESR (Equivalent Series Resistance)	see Figure 27
C_O (shunt capacitance)	7.0 pF maximum
C_L (load capacitance)	30 pF ±3 pF
Drive Level	1 MW

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

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 as higher values, typically 47 pF. The manufacturer of the ceramic resonator should be consulted for recommendations on the values of these capacitors.

A more in-depth discussion of crystal specifications, ceramic resonators, and the selection of values for C1 and C2 can be found in Application Note AP-155, "Oscillators for Microcontrollers" in the Embedded Applications handbook.

To drive the CHMOS parts with an external clock source, apply the external clock signal to XTAL1 and leave XTAL2 floating as shown in Figure 26. This is an important difference from the HMOS parts. With HMOS, the external clock source is applied to XTAL2, and XTAL1 is grounded.

An external oscillator may encounter as much as a 100 pF load at XTAL1 when it starts up. This is due to interaction between the amplifier and its feedback capacitance. Once the external signal meets the V_{IL} and V_{IH} specifications the capacitance will not exceed 20 pF.

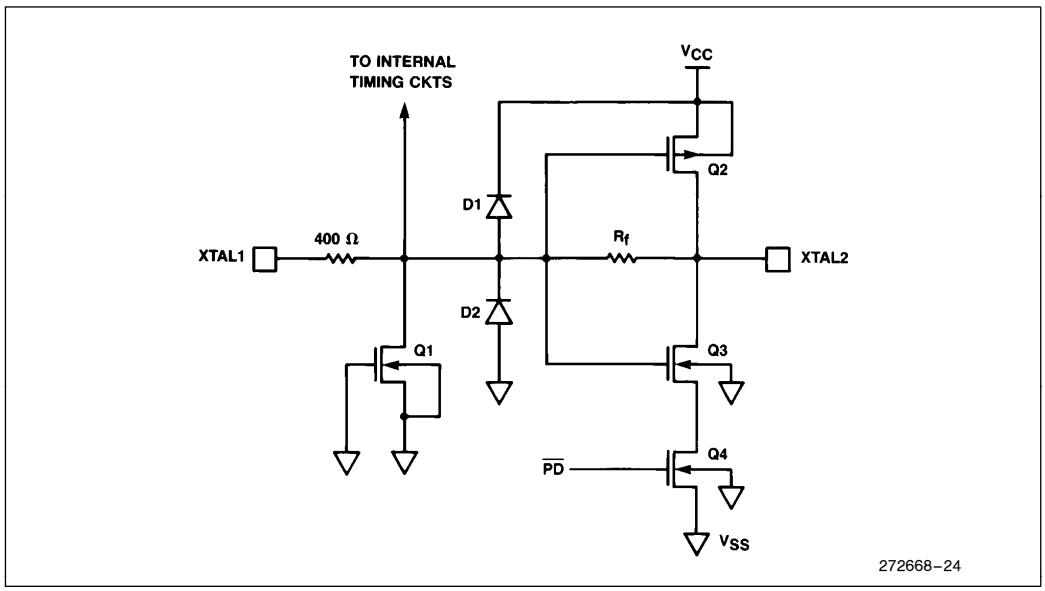


Figure 24. On-Chip Oscillator Circuitry

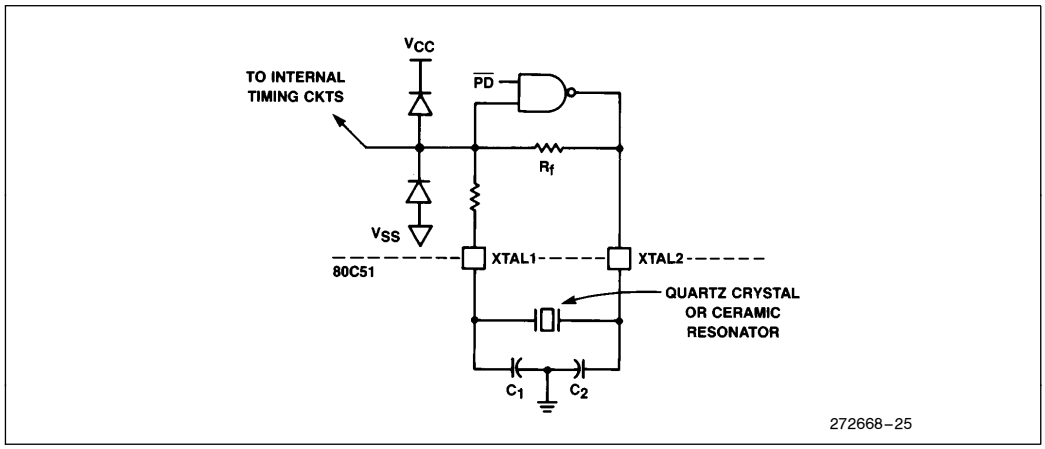


Figure 25. Using the CHMOS On-Chip Oscillator

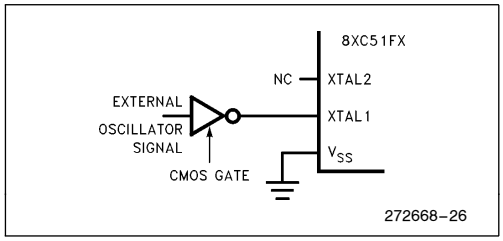


Figure 26. Driving the CHMOS Parts with an External Clock Source

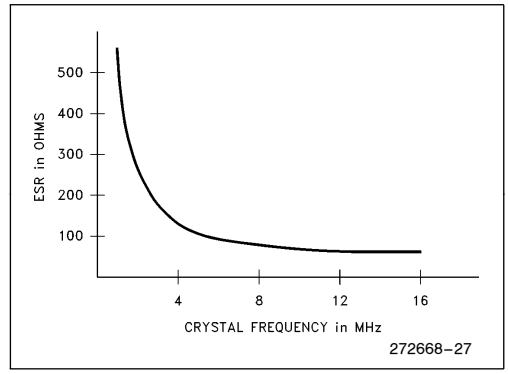


Figure 27. ESR vs Frequency

16.0 CPU TIMING

The internal clock generator defines the sequence of states that make up a machine cycle. A machine cycle consists of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1 microsecond if the oscillator frequency is 12 MHz. Each state is then divided into a Phase 1 and Phase 2 half.

Rise and fall times are dependent on the external loading that each pin must drive. They are approximately 10 nsec, measured between 0.8V and 2.0V.

Propagation delays are different for different pins. For a given pin they vary with pin loading, temperature, V_{CC}, and manufacturing lot. If the XTAL1 waveform is taken as the timing reference, propagation delays may vary from 25 to 125 nsec.

The AC Timings section of the data sheets do not reference any timing to the XTAL1 waveform. Rather, they relate the critical edges of control and input signals to each other. The timings published in the data sheets include the effects of propagation delays under the specified test condition.

ADDITIONAL REFERENCES

The following application notes provide supplemental information to this document and can be found in the *Embedded Applications* handbook.

1. AP-125 "Designing Microcontroller Systems for Electrically Noisy Environments"
2. AP-155 "Oscillators for Microcontrollers"
3. AP-252 "Designing with the 80C51BH"
4. AP-410 "Enhanced Serial Port on the 83C51FA"
5. AP-415 "83C51FA/FB PCA Cookbook"
6. AB-41 "Software Serial Port Implemented with the PCA"
7. AP-425 "Small DC Motor Control"
8. The appropriate data sheet.