
PC BOARD DESIGN FOR 8051

Akhil Guliani



Summer Internship
Instrument Design and Development Center
Indian Institute of Technology, Delhi
July 2010

CERTIFICATE

This is to certify that the dissertation titled "PC Board Design for 8051", submitted by Akhil Guliani entry no. 2009EE19128 as a report for the work done during his Internship at the Instrument Design and Development Center, Indian Institute of Technology, New Delhi, is a record of bonafide work done by him under my supervision and guidance.

Dr. I P Singh

Associate Professor
Instrument Design and Development Center
Indian Institute of Technology, Delhi

Instrument Design and Development Center
Indian Institute of Technology, Delhi

July 2010

ACKNOWLEDGEMENT

I would like to thank my guide Professor I P Singh for his continued support and guidance without which this dissertation was not possible. I would also like to thank everyone else who has been supportive and helpful during my stay at IIT Delhi.

Thank you

Akhil Guliani

CONTENTS

Certificate	2
Acknowledgement	3
Abstract.....	5
Introduction	6
1. THE 8051 MICROCONTROLLER	8
1.1 Introduction.....	8
1.2 8051 Pin Out	9
1.3 The Architecture of 8751.....	11
1.4 Memory Organization in 8751	13
1.5 The Special Function Registers.....	15
1.6 The Timer/Counters 0 and 1	18
1.7 The Serial Interface.....	19
1.7.1 Using the Serial Interrupt	20
1.8 The Interrupts.....	21
1.9 The Power Control register PCON.....	22
1.10 THE 8051 INSTRUCTION SET	23
1.10.1 Addressing Modes.....	23
1.10.2 Instruction Timing.....	25
2 The Printed Circuit Board	27
2.1 Block diagram.....	27
2.2 Circuit details.....	30
2.2.1 Bill of materials	30
2.2.2 Netlist.....	31
2.3 Circuit Components.....	35
2.4 PCB Design Guidelines	40
2.4.1 Circuit Board Layout (Component Selection and Placement)	40
2.4.2 Circuit Board Layout (Trace Routing)	41
References	42
Appendix.....	43

ABSTRACT

The main theme of this report is to design a PC board for interfacing the 8751 with different kinds of peripherals. The paper presents an in depth view of the Printed Circuit board design from an engineer's perspective. The report also presents the microcontroller 8751 from the view point of its architecture, programming and interfacing.

INTRODUCTION

A computer system consists of three main components, namely, the Central Processing Unit (CPU), the Memory and the Input/output Interface. Apart from these it has the clock and reset circuit. The Memory used in the system is of two types, i.e., the Read Only memory (ROM) and the Read Write Memory (RAM). With the advent of better fabrication technology we were able to miniaturize the CPU to a single silicon chip and this was known as a microprocessor. Further miniaturization lead to the integration of the RAM, the ROM and the I/O Interface with the CPU. These microcomputers on a single chip are known as microcontrollers.

The evolution of microprocessor technology launched off in two different directions. One of them is the evolution of faster and more powerful microprocessors of increasing word lengths and memory addressing capabilities. The other is the integration of memory and I/O interface with the CPU on a single chip and providing more and more I/O facilities on it.

The former branch leads to development of 16-bit and 32-bit microprocessors with increasing power and speed. These microprocessors find application in personal computers and work stations, as well as sophisticated instruments because of their speed. An example of this is the Personal Computer, which was initially based on 8088, then 80286, 80386, 80486, Pentium 1, 2, 3, 4 and now Core i7.

The other branch developed because of the need of fast real time control requiring the integration of the CPU, the memory and the I/O facilities on a single chip. More and more I/O facilities are being provided in microcontrollers such as D/A and A/D. The compactness of these microcontrollers has made it possible to provide efficient and economical automation to small and low cost systems and even toys.

Some of the common facilities found on microcontrollers include the CPU, On-chip RAM, On-chip ROM, Parallel I/O Ports, Timer/Counter, Interrupts, On-chip Clock Generator, and Facilities for Expanded Mode. Apart from these common features, the other resources which may also be present, such as Digital to Analog Conversion, Analog to Digital Conversion, Watch Dog Timer, Power Down Mode, Serial I/O and so on.

Microcontroller applications are wide ranging these include the domain of low cost toys to Hand-Held instruments to expensive guidance control systems. They are used in many peripheral devices attached to a main system. For example, the keyboard is based on the 8048 microcontroller. Some stand alone devices use the microcontroller to control their functions. The paper copier, the electronic typewriter and the plotters are examples of these.

Microcontrollers are also used in ignition control, transmission control, Anti-skid brakes and pollution control in automobiles. The more sophisticated high end microcontrollers are used for missile control, torpedo guidance, and control intelligent ammunition etc.

Many microcontrollers are available in the market. In this text I have used the 8051. The 8051, a part of the larger MCS-51 family of microcontrollers introduced by Intel, has been an industry standard 8-bit microcontroller since its introduction in 1980. The main reason behind its popularity has been the ease and flexibility of its use in various applications due to the presence of a large number of manufacturers producing a large number of variants. It is mainly used in automotive applications.

Once we have a microcontroller we need to provide it with some additional circuitry such as a reset circuit, a crystal circuit, connectors and certain IC's associated with interfacing of the microcontroller to its peripherals. Here comes the role of the Printed Circuit (PC) Board which offers to mechanically support and electrically connect electronic components using conductive pathways, tracks and signal traces etched from copper sheets embedded in non-conducting substrate.

The popularity of the PCB stems from the fact that it is a cheap and robust method of circuit assembly. The PCB requires a robust design to reduce and eliminate certain parasitic effects caused due to its inherent structure and external environment at the application site.

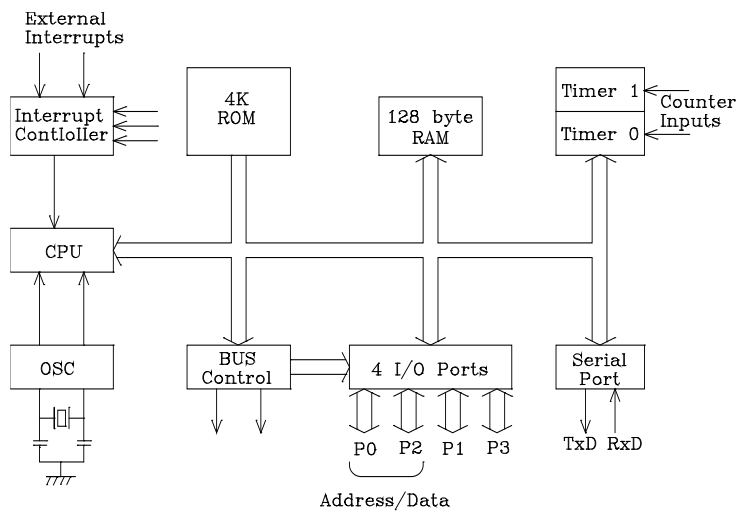
This paper gives a brief about the 8051 and its related circuitry, some peripheral IC's and then moves on to the designing the PCB of the final circuit.

1. THE 8051 MICROCONTROLLER

1.1 INTRODUCTION

MCS-51 family was introduced by Intel in 1980. Since then it has become an industrial standard for 8-bit high performance microcontrollers. The architecture has been optimized for sequential real time control. These are available from several manufactures in several versions, with and without a ROM. The MCS-51 family has found acceptance in a very wide range of applications. These include medical instrumentation, automobile applications etc. Dedicated controllers are now available which have the basic 8051 CPU core embedded in them. Examples of these are the 80C152 the universal communication controller, 83C053 the microcontroller for television and video from Philips. The block diagram of 8051 gives a pictorial representation of the resources. The common features of the MCS-51 family are:

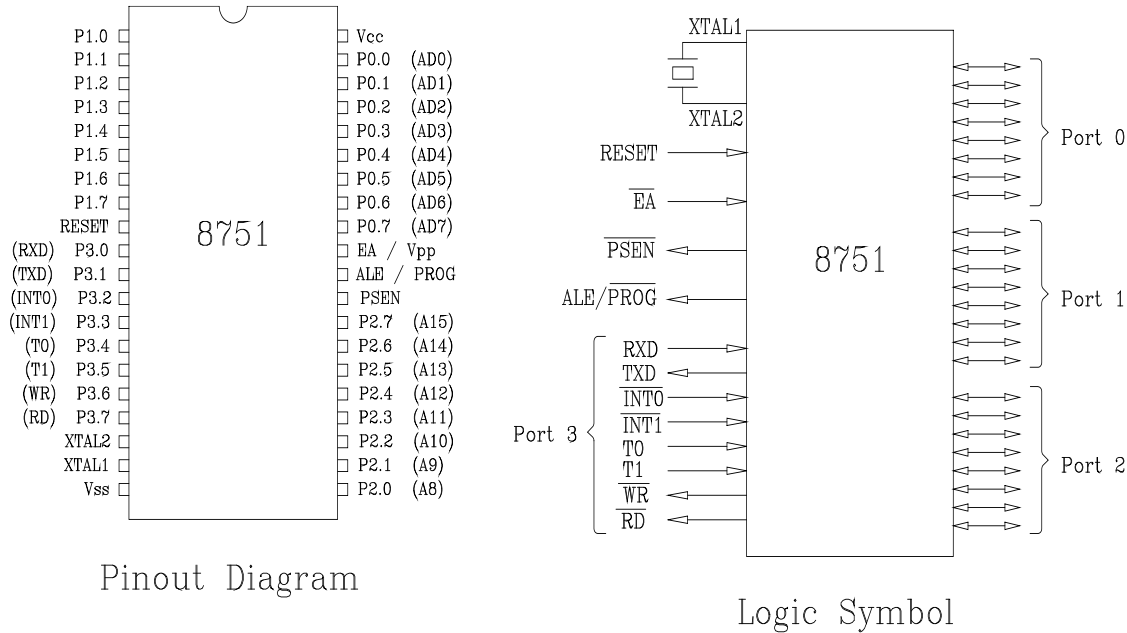
- a. 8-bit CPU optimized for control applications.
- b. Extensive Boolean Processing capabilities.
- c. 32 bidirectional individually addressable I/O lines.
- d. Full Duplex UART
- e. 5 source interrupt structure with 2 priority levels.
- f. 2 -16-bit timer/counters.
- g. 64K bytes maximum program size
- h. 256 bytes maximum on-board RAM size.
- i. 64K bytes of maximum external RAM size.
- j. 1 μ sec instruction cycle with 12MHz crystal.
- k. Hardware multiply and divide instructions in 4 μ sec.
- l. Upward compatibility with 8048 software.



Basic block diagram of 8751.

1.2 8051 PIN OUT

The pin out and the Logic symbol are shown in figure. In order to reduce the chip count some of the pins have dual functions. This method of reducing the chip count is also used in microprocessors like 8085 and 8086 by multiplexing. The 8051 has 40 pins (It is also available in the 44 pin quad flat pack). The four ports take up 32 pins (Figure 9.18). These are described below.

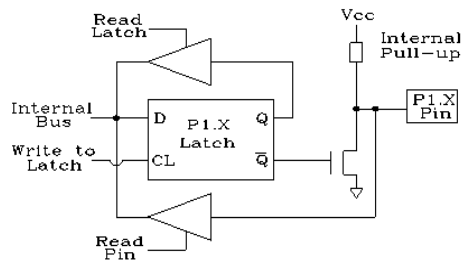


Port P0 : This is an open drain bi-directional port. This also serves as the lower multiplexed address and data bus AD0-AD7 when 8751 is used in the expanded mode. As an output it can sink 8 LSTTL load. If '1' is written to bits of this port, the pins are in a high impedance state, and can be used as high impedance inputs.

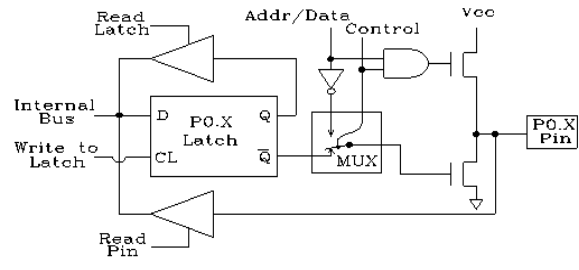
Port P1 : This is a bi-directional port with internal pull-ups. When a '1' is written to bits of this port they are pulled high by internal pull-ups and can be used as inputs.

Port P2 : This is a bi-directional port with internal pull-ups. It also serves as the higher order address bus in the expanded mode. When a '1' is written to bits of this port they are pulled high by internal pull-ups and can be used as inputs.

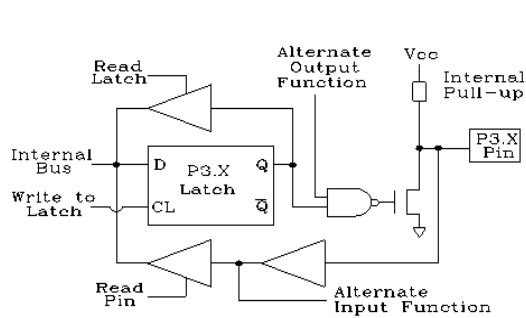
Port P3 : This is a bi-directional port with internal pull-ups. When a '1' is written to bits of this port they are pulled high by internal pull-ups and can be used as inputs. This port also has alternate functions as indicated below. The details are given in Table i.



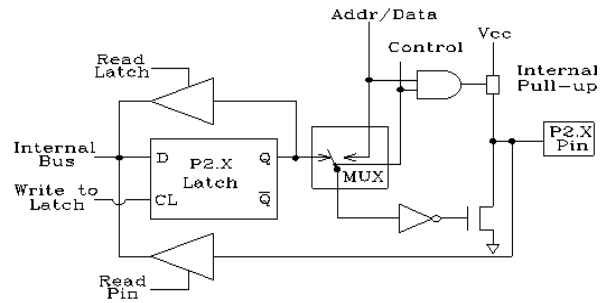
Port 1 Bit



Port 0 Bit



Port 3 Bit



Port 2 Bit

Detail diagram of ports

Table i. Alternate Functions of Port P3.

<u>Pin</u>	<u>Name</u>	<u>Alternate Function</u>
P3.0	RXD	Serial input Line
P3.1	TXD	Serial Output Line
P3.2	INT0	External Interrupt 0
P3.3	INT1	External Interrupt 1
P3.4	T0	Timer 0 External Input
P3.5	T1	Timer 1 External Input
P3.6	WR	External Data Memory Write Strobe
P3.7	RD	External Data Memory Read Strobe

ALE :

Address Latch Enable is used for demultiplexing the AD0-AD7 when the 8751 has external memory added to it (expanded mode). A latch is used to latch the address at the falling edge of the ALE when ever external memory is addressed. This pin is also used during the programming of the on-board EPROM. The ALE is normally emitted at 1/6 of oscillator frequency.

RESET :

A HIGH on this input for two machine cycles resets the micro-controller. A capacitor connected to this pin can provide for power-on reset.

PSEN :

The PSEN known as the Program Store Enable is used as the Read Strobe for the Program Memory. This signal is inactive when program fetches are taking place from the internal ROM.

EA/Vpp :

This pin must be grounded when the Program Fetches are required to be made from the External Memory Locations 0000H to 0FFFH, ie, the Internal ROM is not to be used. Else, for normal operation the EA must be connected to Vcc. This pin is also used in Internal EPROM programming for applying the programming pulse.

XTAL1, XTAL2 :

These inputs are the crystal. The clock circuit on board generates the clock. If an external clock is being used, the input clock is given to XTAL1, and XTAL2 is left unconnected.

Vcc and Vss:

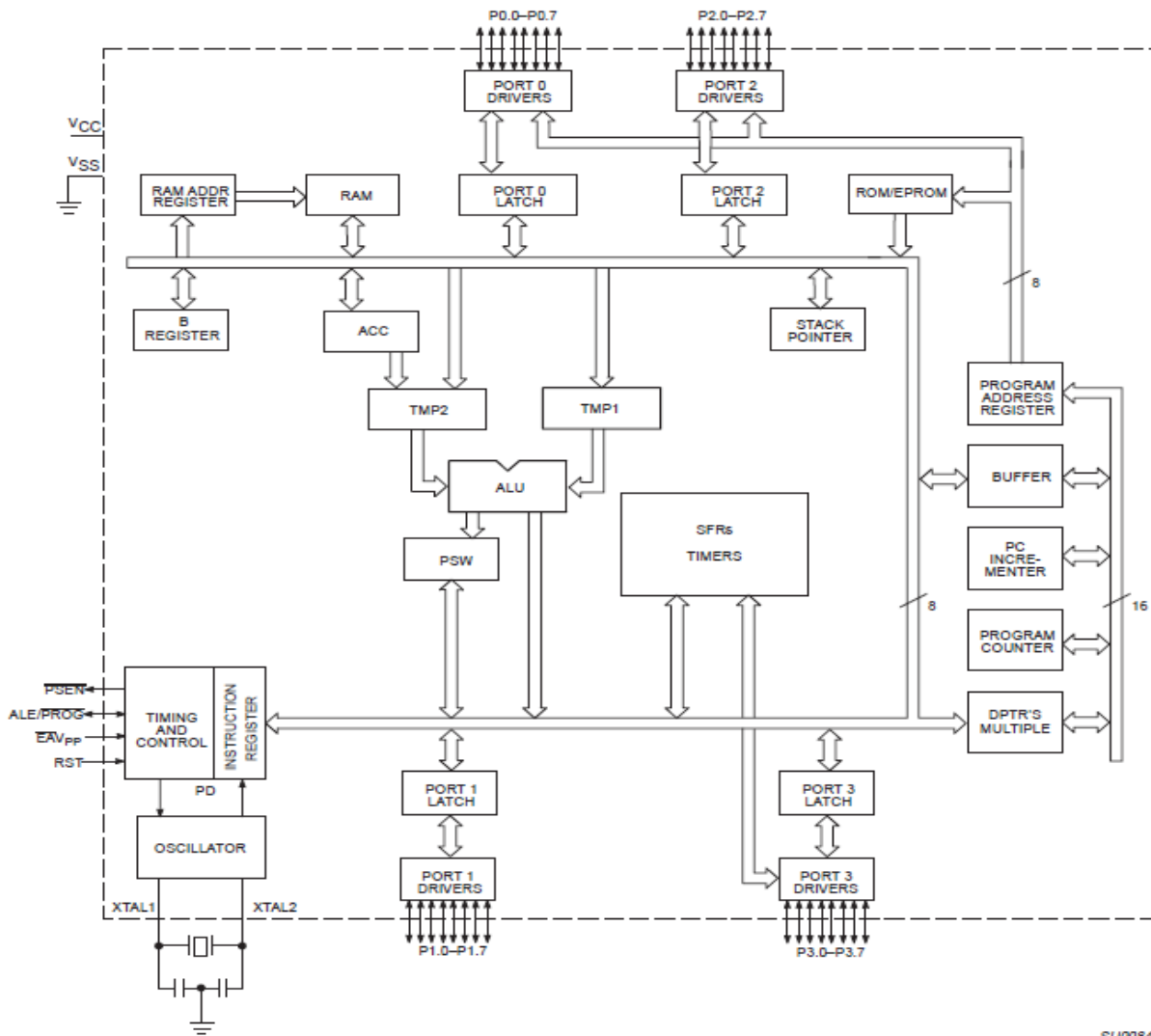
These are the power 5V and ground respectively.

1.3 THE ARCHITECTURE OF 8751

The Central Processing Unit is responsible for fetching and executing the instructions. Its main elements of interest to the user are the 8-bit Arithmetic Logic Unit, the Accumulator A, the Multiplication Register B, the Program Status Word PSW, the Stack Pointer SP, the 16-bit Program Counter PC(PCH and PCL) for Program Memory and the Data Pointer DPTR(DPH & DPL) for External Data Memory. The registers A, B, PSW, SP, DPH AND DPL are some of the Special Function Registers (SFR), and are mapped onto the Internal Data Memory. These are described in detail later.

The I/O ports have also been shown at the four corners of the figure. Port P2 and Port P0 are shown to be connected to the Program Counter and the Data Pointer. It is as stated earlier that the memory expansion for both data as well as program memory is done. The latches P0, P1, P2, P3 of all the ports are associated with SFRs, which form a part of the Internal Data Memory. Details of Port bit latches are shown in Figure 9.18.

The eight pins of Port P3 are also having alternate functions. Similarly all the alternate functions of Port P3 are also associated with SFRs. These include Serial Port Control Register SCON, Serial Port Buffer SBUF, Interrupt Enable Register IE, Interrupt Priority Register IP, Timer Register TCON, Timer Mode Register TMOD, and the Timer 0 and Timer 1 high and low count bytes - TL0, TH0, TL1 and TH1. Since these form a part of the internal memory these will be discussed in the next section.



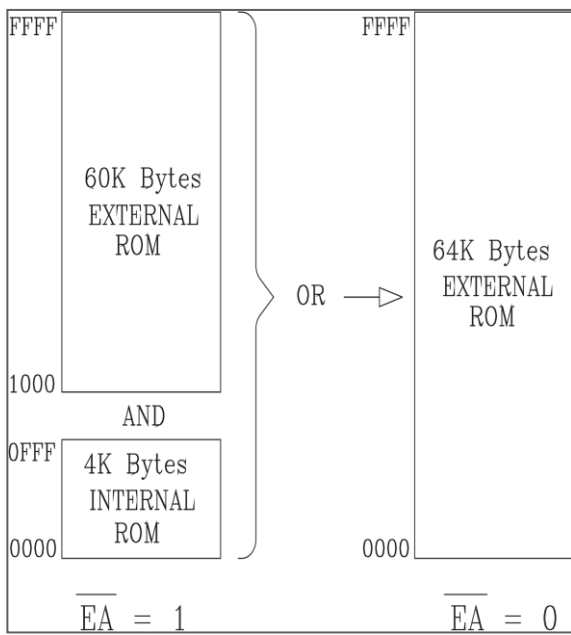
SU00545

Functional Block Diagram of 8751

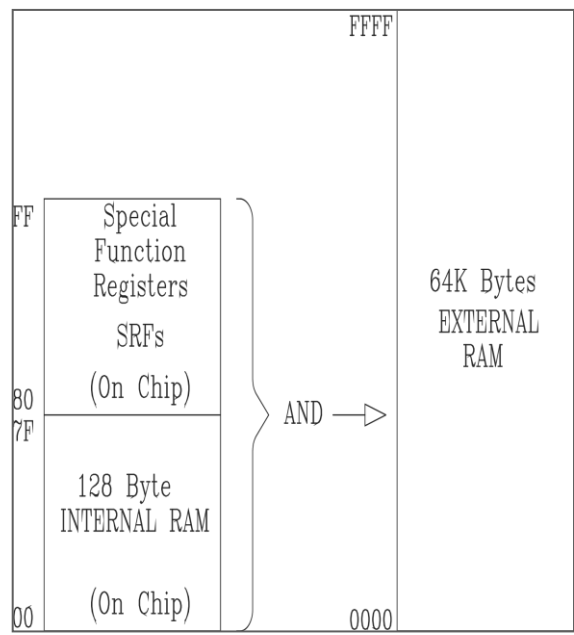
1.4 MEMORY ORGANIZATION IN 8751

The 8051 has separate address spaces for program memory and the data memory. Both these address spaces can be expanded through External memories. Therefore the Total Memory Space in the 8051 can be divided into four distinct parts. These are the Internal Data Memory, the External Data Memory, the Internal Program Memory, and the External Program memory.

The 8051 can be expanded for up to 64K of external data memory (RAM) and 64K of external code memory (ROM/EPROM). The disadvantage of adding external memory is that ports P0 and P2 are used for the address and data bus and are not available of I/O operations. Furthermore, if the Data Memory is expanded the read and write strobes (RD and WR) for it are available as alternate functions on P3. Thus two more bits are not available for I/O.



The Program Memory



The Data Memory

The Program Memory

The program counter starts at 0000H on RESET and therefore the first instruction to be executed lies at this location. Normally (if EA is connected to Vcc) this location lies in the Internal ROM, but if is grounded the program fetches are made from the External Memory. When EA/Vpp is made high, the first 4K of program instructions are fetched from the Internal ROM. The fetches from Program Memory above 4K (1000H to FFFFH) are made from the External ROM. The layout of the Program Memory are shown in Figure 9.19.

The Data Memory

The Data Memory Space can be divided into two separate regions. The 8051 can address up to 64K Bytes of External Data Memory (RAM). The other region is the Internal Data Memory of 128 bytes of RAM and a further 128 bytes of hardware register address space on the chip itself. This is shown in Figure 9.20.

The External Data Memory :

Only if the application calls for large amounts of data memory, then external RAM is interfaced. There are only instruction, the MOVX instruction which deals with data transfers to and from the external RAM. If the data memory is expanded two bits of Port P3 are also required apart from P0 and P2, thus reducing the I/O pins available.

The Internal Data Memory:

This is a 256 byte memory on the chip itself. The first 128 bytes form the internal ram space. The other 128 bytes from 80H to FFH are registers concerned with hardware, known as the Special Function Registers SFRs. Not all of 80H to FFH are occupied. The 128 bytes of RAM can be divided into three segments as detailed below. The use of the address space in the 128-byte RAM is detailed in Figure 9.21.

a) The Register Banks 0 to 3. The first 32 locations contain the four banks (RB0 to RB3) of data registers. Each bank contains 8 registers. Only one of these banks can be active at a time. The program addresses the registers of the active bank as R0 to R7. The bank that is presently in use is defined by the setting of the two "bank select bits" in the PSW (Program Status Word). The Registers R0 and R1 of the selected register have a special significance as they can be used for indirect addressing. After the hardware RESET the register bank RB0. Also the Stack Pointer is initialized to 08H. Therefore if the second bank is to be used the Stack Pointer should be re-initialized to a place in the RAM which will not be used for data.

b) The Bit Address Space: RAM addresses 20H through 2FH are designated as bit-addressable memory locations. These 128 bits can be directly addressed. There are two ways of addressing them. One way is to refer to them as bits starting from 00H to 7FH. The other method is to address them in reference to the byte, ie, 21.7 refers to the 7th bit of the 21H byte of address. This is the same as 0F bit. The 16 bytes constituting the Bit Address Space can also be addressed in the normal manner as bytes. This method of addressing is useful for control applications for a large number of ON/OFF bit flags.

c) The Scratch Pad Area: The last 80 bytes are available to the user as RAM for general-purpose data storage as well as stack operations.

1.5 THE SPECIAL FUNCTION REGISTERS.

The Special Function Registers (SFRs) are contained in the next 128 bytes of the internal data memory corresponding to address 80H to FFH. The SFRs are registers required for software instruction execution as well as for controlling the special hardware features. These SFRs can either be addressed by their hexadecimal address or by their symbolic name. Table ii lists the SFRs' symbolic name, its address and its value at RESET.

Special Function Bit Addressable Registers

	Internal RAM Address		Byte Addr.	Bit Addr		Name	
<div style="display: flex; justify-content: space-between;"> MS Bit LS Bit </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> Scratch Pad Area 80 Bytes </div> <div style="display: flex; justify-content: space-between;"> 7F 7E 7D 7C 7B 7A 79 78 </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> Bit Addressable Area 20H-2FH </div> <div style="display: flex; justify-content: space-between;"> 0F 0E 0D 0C 0B 0A 09 08 </div> <div style="display: flex; justify-content: space-between;"> 07 06 05 04 03 02 01 00 </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Register Bank 3 18H-1FH </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Register Bank 2 10H-17H </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Register Bank 1 08H-0FH </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Register Bank 0 00H-07H </div>	7FH		F0H	F7 F6 F5 F4 F3 F2 F1 F0		B	
		30H		E0H	E7 E6 E5 E4 E3 E2 E1 E0		Acc
		2FH		D0H	D7 D6 D5 D4 D3 D2 D1 D0		PSW
				B8H	- - - BC BB BA B9 B8		IP
				B0H	B7 B6 B5 B4 B3 B2 B1 B0		P3
				A8H	AF - - AC AB AA A9 A8		IE
				A0H	A7 A6 A5 A4 A3 A2 A1 A0		P2
				98H	9F 9E 9D 9C 9B 9A 99 98		SCON
				90H	97 96 95 94 93 92 91 90		P1
				88H	8F 8E 8D 8C 8B 8A 89 88		TCON
				80H	87 86 85 84 83 82 81 80		P0

Table ii. Special Function Registers at their value at Reset.

Symbol	Name of Special Function Register	Address	Value at Reset
* ACC	Accumulator	E0H	00000000B
* B	B Register	F0H	00000000B
* PSW	Program Status Word	D0H	00000000B
SP	Stack Pointer	81H	07H
DPTR	Data Pointer		
DPH	Low Byte	82H	00H
DPL	High Byte	83H	00H
* P0	Port 0	80H	11111111B
* P1	Port 1	90H	11111111B
* P2	Port 2	A0H	11111111B
* P3	Port 3	B0H	11111111B
* IP	Interrupt Priority Control	B8H	XXX00000B
* IE	Interrupt Enable Control	A8H	0XX00000B
TMOD	Timer/Counter Mode Control	89H	00H
* TCON	Timer/Counter Control	88H	00000000B
TH0	Timer/Counter 0 High Byte	8CH	00H
TL0	Timer/Counter 0 Low Byte	8AH	00H
TH1	Timer/Counter 1 High Byte	8DH	00H
TL1	Timer/Counter 1 Low byte Serial Control	8BH	00H
* SCON	Serial Data Buffer	98H	00000000B
SBUF	Power Control	99H	XXH
PCON		87H	0XXXXXXXB

* These SFRs are bit addressable.

The Register for Port 0 can be addressed either by P0 or by its address 80H. Several of these registers are bit addressable (Figure 9.22). These have been indicated in the Table VI by a '*'. The first bit of Port P0 can be addressed as 80H.0 or P0.0. Similarly the carry bit in PSW can be addressed as PSW.7 or D0H.7

Some of these registers have been discussed below. Reference to the User Manual should be made for a complete description of these registers.

a) The Accumulator and Register B.

The 8051 is also an accumulator based processor, like the 8048 and 8085 it contains one of the operands as well as the results of an arithmetic operations. In the instructions the Accumulator is referred to as A. The Register B is used together with the register A in multiply and divide instructions.

b) The Program Status Word.

Most of the arithmetic and logical operations affect the status flags. These flags are grouped together to form the Program Status Word, PSW. The PSW also contains information about which is the active register bank of the four register bank. Details are given below:

PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
CY	AC	F0	RS1	RS0	OV	-	P

CY: This is the Carry flag. It is used by the addition, subtraction and rotate instructions. Apart from that it is used as a Boolean Accumulator for the one bit logical and bit manipulation instructions.

AC: This bit is used for BCD operations and denotes carry or borrow out of bit 3 of accumulator.

OV: This indicates whether an overflow has occurred. This makes signed arithmetic possible.

P : This is the parity flag. It is set/cleared depending upon whether the number of '1' in the Accumulator is odd/even.

F0: This is the user defined flag F0. This can be set, cleared or tested through software.

RS1 & RS0: These two bits indicate which one of the four register banks is active. '00' indicates Bank 0, '01' indicates Bank 1, '10' indicates Bank 2 and '11' indicates Bank 3.

c) The P0, P1, P2, P3 are the SRF latches registers of the respective ports.

d) **The Stack Pointer SP** is 8 bit wide and can point to data anywhere in the Internal RAM. This is initialized to 07 after reset, so the stack starts from 08H.

e) **The Data Pointer DPTR (DPH & DPL)** is a 16-bit register and holds a external Data address. It can also be manipulated as two separate 8-bit registers.

1.6 THE TIMER/COUNTERS 0 AND 1

These have a 16-bit counter each. These are TH0, TL0, TH1, TL1. Apart from these there are two registers TMOD, the Mode Control Register, and the TCON, the Control Register. TMOD register determines for both the timer/counters whether they are to be used as a timer or a counter, selects one of the four modes of the timer/counter operations and defines the gating control. The TCON on the other hand contains four control bits to turn the two timer/counters on/off as well as to specify whether the interrupts are falling edge triggered or low-level triggered. There are two flags to indicate an overflow of the two timer/counters. There are two other flags which indicate whether an external interrupt edge was detected at INT0 and INT1. Details of TMOD and TCON are given in below

The TCON and TMOD Control Registers.

TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.

GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer 1				Timer 0			

GATE When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control).
When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).

C/T Timer or Counter selector. Cleared for Timer operation (input from Internal system clock). Set for Counter operation (input from Tx input pin).

M1 Mode selector bit. (NOTE 1)

M0 Mode selector bit. (NOTE 1)

NOTE 1:

M1	M0	Operating Mode
0	0	0 13-bit Timer (8048 compatible)
0	1	1 16-bit Timer/Counter
1	0	2 8-bit Auto-Reload Timer/Counter
1	1	3 (Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standart Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3 (Timer 1) Timer/Counter 1 stopped.

The Serial Port Control Register SCON.

SCON: SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0	SCON.7	Serial Port mode specifier. (NOTE 1)
SM1	SCON.6	Serial Port mode specifier. (NOTE 1)
SM2	SCON.5	Enables the multiprocessor communication feature in modes 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 6.)
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.
RB8	SCON.2	In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 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

SERIAL PORT SET-UP:

Table 6.

MODE	SCON	SM2 VARIATION
0	10H	Single Processor Environment (SM2 = 0)
1	50H	
2	90H	
3	D0H	
0	NA	Multiprocessor Environment (SM2 = 1)
1	70H	
2	80H	
3	F0H	

1.7 THE SERIAL INTERFACE.

The Serial Interface is full duplex, and the data is transmitted and received by the SBUF register. On writing to SBUF it loads the transmit register and on reading the SBUF a separate receive buffer is accessed. The serial transmission is controlled by the special function register SCON (Figure 9.24). There are 4 modes of serial operation. In modes 1 and 3, 10 and 11 bits are transmitted and the baud rate is variable. Mode 2 is similar to mode 3, except that the baud rate is fixed at 1/32 or 1/64 of oscillator frequency. For the variable baud rates the overflow of Timer 1 is used.

In mode 0 the TXD outputs a clock at fixed baud rate of 1/12 of the frequency. The data either enters or exits through the RXD. Using this mode it is possible to expand the ports by using the serial I/O expansion methods. This is done by using serial parallel out shift registers with latch, ie, 74HC595 for output and parallel in serial out shift registers, ie 74HC167 or 74HC589 for input.

1.7.1 USING THE SERIAL INTERRUPT

For generating the variable baud rate for serial transmission in mode 1 and 3, the Timer 1 is used in mode 2 (8-bit timer with auto reload). The Timer 1 interrupt should be disabled. The Baud Rate is set by loading TH1 as follows

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

Details of load count of TH1 for various baud rates is shown in the Table iii.

Table iii Timer 1 Generated Baud Rates.

Baud Rate	f _{osc}	SMOD	Timer 1		TH1
			C/T	Mode	
Mode 0, Max 1MHz	12 MHz	X	X	X	X
MOde 2, Max 375 K	12 MHz	1	X	X	X
MOde 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
137.5K	11.059 MHz	0	0	2	1DH
110K	6 MHz	0	0	2	72H
110K	12 MHz	0	0	1	FEEBH

1.8 THE INTERRUPTS.

There are 5 sources of interrupts in 8051. These are two external- interrupts, INT0 and INT1, two timer-interrupts TF0 and TF1 and a single Serial Interrupt, which is a logical OR of receive and transmit interrupts. The external interrupts can be either falling edge or low level triggered, but these are programmed by the TCON associated with the timer/counters. The interrupts are enabled or disabled through the IE special function register. Each of these interrupts has a priority attached. This is determined by the Interrupt Priority Register, the IP. Within each priority there is a priority determined in order of the interrupts listed below in the interrupt types, with IE0 having the highest priority.

The Interrupt Control Registers IE and IP.

IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	-	-	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
-	IE.6	Not implemented, reserved for future use.*
-	IE.5	Not implemented, reserved for future use.*
ES	IE.4	Enable or disable the serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External Interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

IP: INTERRUPT PRIORITY REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt has a lower priority and if the bit is 1 the corresponding interrupt has a higher priority.

-	-	-	PS	PT1	PX1	PT0	PX0
---	---	---	----	-----	-----	-----	-----

-	IP.7	Not implemented, reserved for future use.*
-	IP.6	Not implemented, reserved for future use.*
-	IP.5	Not implemented, reserved for future use.*
PS	IP.4	Defines the Serial Port interrupt priority level.
PT1	IP.3	Defines the Timer 1 interrupt priority level.
PX1	IP.2	Defines External Interrupt 1 priority level.
PT0	IP.1	Defines the Timer 0 interrupt priority level.
PX0	IP.0	Defines the External Interrupt 0 priority level.

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

In order to use the interrupts that interrupt will have to be first unmasked individually as well globally through the IE register. If external interrupts are being used, the triggering method is programmed through TCON. The service routine will then have to lie at the Vector Address in the Program Memory of that Interrupt as indicated below.

Interrupt Type	Name	Address in Prog. Memory
External Interrupt 0	IE0	0003H
Timer Interrupt 0	TF0	000BH
External Interrupt 1	IE1	0013H
Timer Interrupt 1	TF1	001BH
Serial Interrupt	RI & TI	0023H

1.9 THE POWER CONTROL REGISTER PCON.

There are power saving modes of operation of 8051. These modes are only available for CHMOS devices. These are the Idle Mode and the Power Down Mode. In the idle mode the oscillator continues to run, but the clock signal is not supplied to the CPU. The oscillator continues to clock the interrupts, the serial port and the timer block. This mode is set by setting the IDL bit of PCON to '1'. This mode can be terminated through an interrupt or a reset. In the Power Down Mode the oscillator is frozen. This mode can be set by setting PD bit of PCON to '1'. The only way to come out of this mode is through a hardware RESET. The GF1 and GF0 are general purpose flags.

The SMOD bit of PCON doubles the baud rate when it is set to '1'. This is used in baud rate settings for serial interface. In HMOS devices this is the only available bit in PCON. The various bits are shown below.

PCON.7	PCON.6	PCON.5	PCON.4	PCON.3	PCON.2	PCON.1	PCON.0
SMOD	--	--	--	GF1	GF0	PD	IDL

1.10 THE 8051 INSTRUCTION SET

There are a number of versions of the 8051, belonging to the MCS-51 family. All the members of the 8051 family use the same instruction set. A summary of instruction set is given in Table VIII (The complete instruction set is given in the Appendix). The instruction set of 8051 make it especially well suited for control applications.

In the description only the mnemonics are discussed. In order to write a program one will therefore need an assembler, such as ASM-51 from Intel. For hand assembly further details will have to be referred for the MCS-51 User Manual.

From a programmers point of view there are five address spaces. These are

- a. The directly addressable data address space (DATA).
- b. The bit address space (BIT).
- c. The external data address space (XDATA).
- d. The code or program address space (CODE).
- e. The indirectly addressable space (IDATA).

These address spaces may have the same numeric value, but the context will determine the space being referred to.

1.10.1 ADDRESSING MODES

There are various addressing modes for addressing the operands, as well as the various address spaces in 8051. The Instructions can be up to three bytes long. In terms of assembly language, the operation mnemonic is written first, followed by operands separated by commas ','. Most of the instructions have one or two operands, but a few also have three operands.

a) Direct Byte Addressing: In this mode the address of the operand is directly specified. By this method only the Internal RAM and the SRF can be accessed. The address is a 8-bit address. The names of the SFR can also be given instead of the address. For example "MOV A,90H" is the same as "MOV A,P1".

b) Register Addressing: In this mode the registers of the active bank Rn (R0 to R7) can be specified as the address of the operand. The active register bank can be found or changed from the PSW register. An example of this would be "ADD A,R7".

c) Register Indirect Addressing: In this mode a particular register is specified, which points to (contains the address of) the location of the operand. This indirect addressing is indicated by the prefix '@'. The indirect addressing registers for 8-bit addresses are the R0 and R1 registers of the active register bank or the stack pointer SP. The indirect addressing register for 16-bit addresses required for accessing external ram is the data pointer DPTR. Example of this mode is "MOV A,@R0". If R0 contains 7FH, then the contents of location 7FH in the Internal RAM is transferred to A. "MOVX A,@DPTR" moves the content of memory location whose address is in DPTR to A.

d) Register Specific or Implicit Addressing: In these instructions the instructions always operate on an implied register, such as the Accumulator or the DPTR. Therefore no operands have to be specified, eg, DA A. The A is specified only for clarity.

e) Immediate Addressing: The operands are specified in the instruction itself. The immediate data is indicated by the prefix '#'. An example of this is MOV A, #A5H.

f) Indexed addressing: This mode of addressing is only valid for the program memory. It can only be used to read locations from the code and is basically used for table-look-up. The 16-bit registers which can be used for indexing are the data pointer DPTR and the program counter PC. In this mode the actual address of the operand in the memory is calculated by adding the accumulator A to the specified index register, DPTR or PC. MOVC A,@A+DPTR and MOVC A,@A+PC instructions move the operands at the memory locations

g) Bit Addressing: This mode of addressing is used to access bits of the bit address space having bit addresses of 00 to 7F and the bit addressable SFRs (Figures 9.21 & 9.22 and Table VI). The bit 7 of the PSW can also be addressed as PSW.7 or D0.7.

This would correspond to bit address D0H (byte address of PSW) + 7 (the bit number = 87 (the bit address). As another example the bit address of Timer 0 overflow flag is at bit 5 of the special

1.10.2 INSTRUCTION TIMING

The execution of instructions takes a finite amount of time. Most instructions take one or two machine cycles. Each machine cycle corresponds to 12 oscillator clocks. Thus if the XTAL frequency is 12MHz, each instruction takes 1 μ sec. For example a NOP takes 1 machine cycle, ie, 1 μ secs. Consider the following delay subroutine

```
DLY:  MOV  R1,#FFH    ; 1 cycle
      MOV  R0,#00H    ; 1 cycle
LP:   DJNZ R0,LP     ; 2 cycles
      DJNZ R1,LP     ; 2 cycles
      RET                ; 2 cycles
```

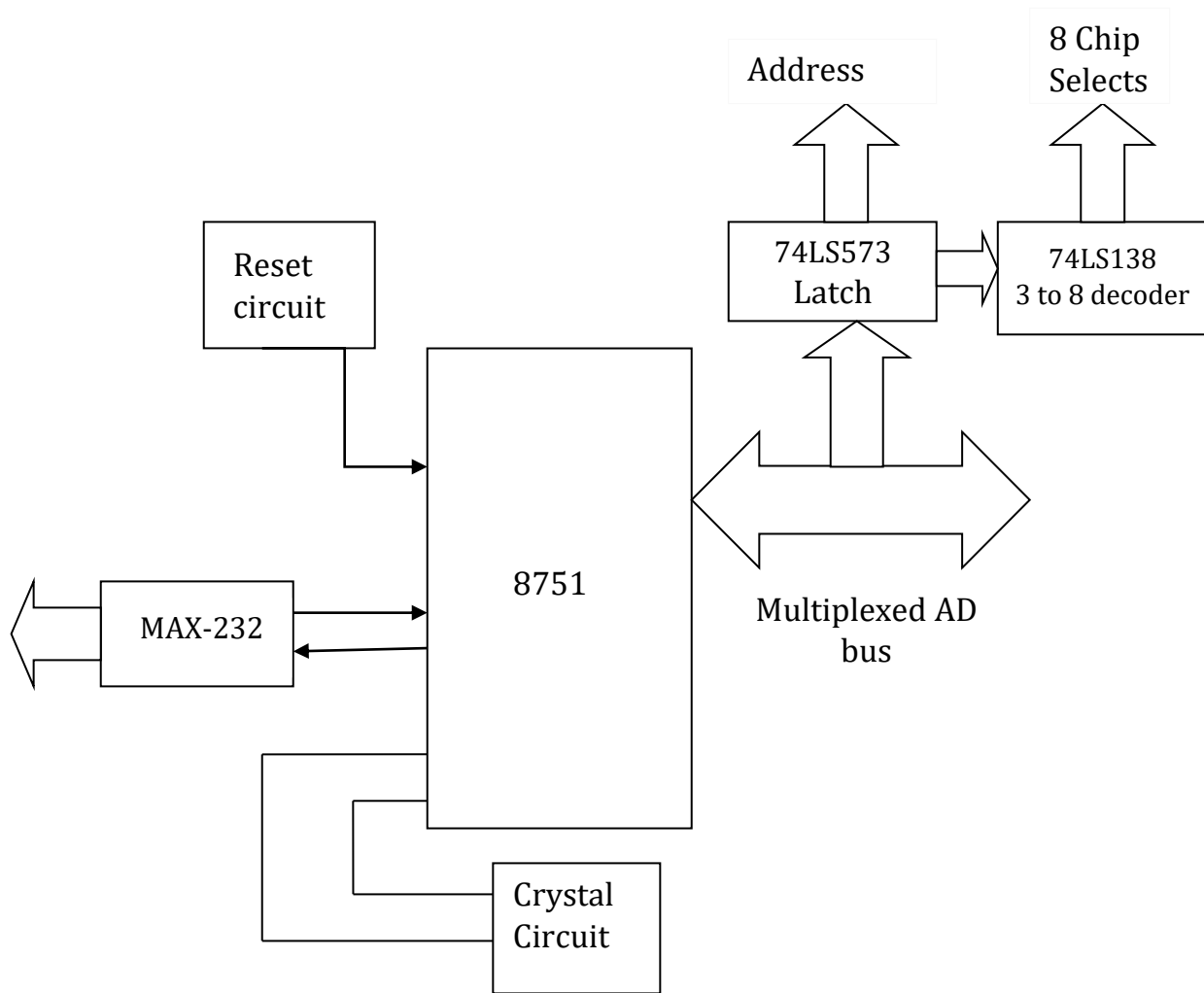
This will take a total of 131082 machine cycles. If the XTAL frequency is 12 MHz, then this will provide a delay of .131082 sec.

TABLE iv . MCS-51 INSTRUCTION DESCRIPTION

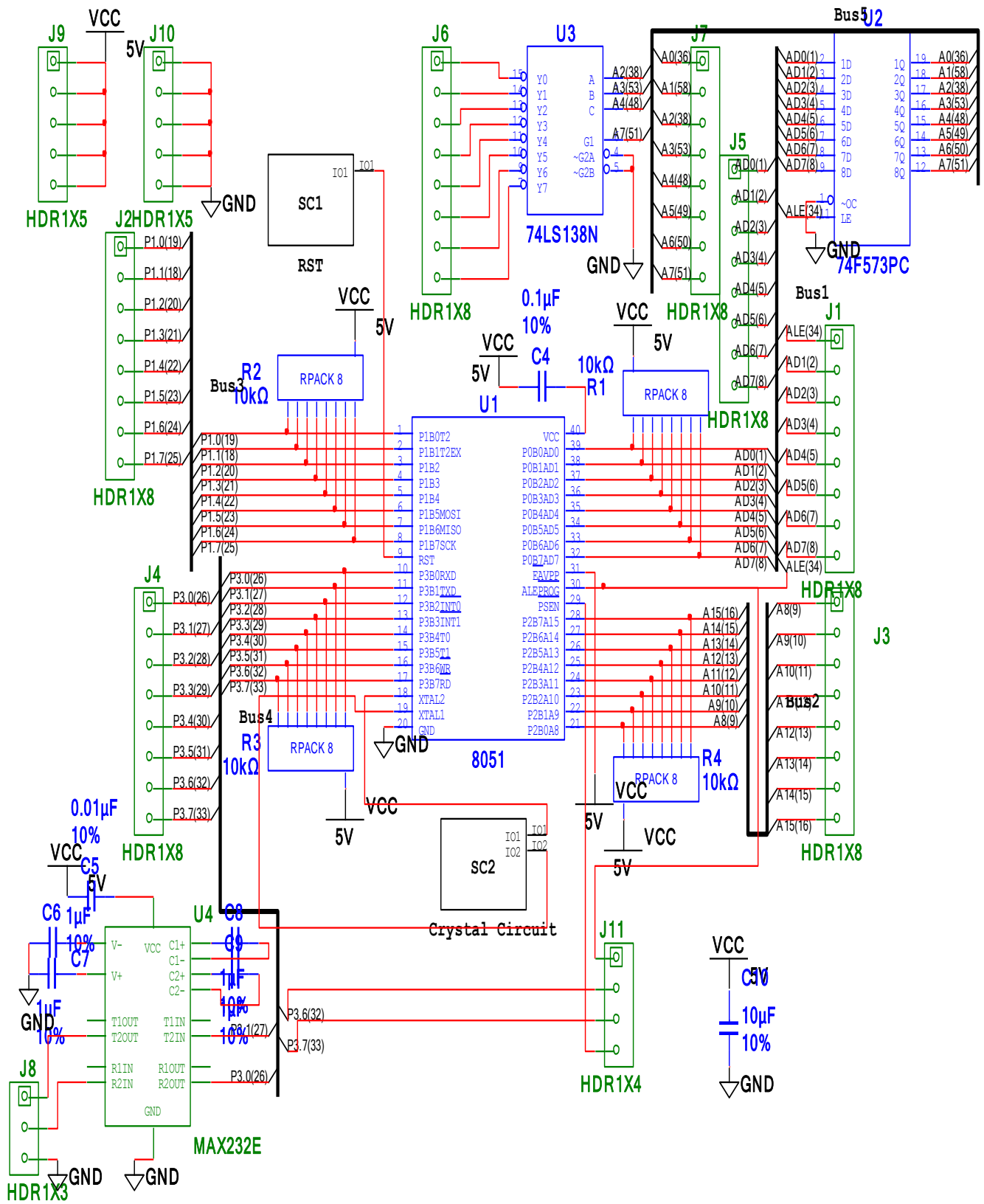
ARITHMETIC OPERATIONS				DATA TRANSFER (cont.)					
Mnemonic		Description	Byte	Cyc	Mnemonic	Description	Byte	Cyc	
ADD	A,Rn	Add register to Accumulator	1	1	MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
ADD	A,direct	Add direct byte to Accumulator	2	1	MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1	MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
ADD	A,#data	Add immediate data to Accumulator	2	1	MOVX	A,DPTR	Move External RAM (16-bit addr) to A	1	2
ADDC	A,Rn	Add register to Accumulator with Carry	1	1	MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
ADDC	A,direct	Add direct byte to A with Carry flag	2	1	MOVX	DPTR,A	Move A to External RAM (16-bit addr)	1	2
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1	PUSH	direct	Push direct byte onto stack	2	2
ADDC	A,#data	Add immediate data to A with Carry flag	2	1	POP	direct	Pop direct byte from stack	2	2
SUBB	A,Rn	Subtract register from A with Borrow	1	1	XCH	A,Rn	Exchange register with Accumulator	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1	XCH	A,direct	Exchange direct byte with Accumulator	2	1
SUBB	A,@Ri	Subtract indirect RAM from A w/Borrow	1	1	XCH	A,@Ri	Exchange indirect RAM with A	1	1
SUBB	A,#data	Subtract immed. data from A w/Borrow	2	1	XCHD	A,@Ri	Exchange low-order 16-bit ind. RAM w/A	1	1
INC	A	Increment Accumulator	1	1	BOOLEAN VARIABLE MANIPULATION				
INC	Rn	Increment register	1	1	Mnemonic	Description	Byte	Cyc	
INC	direct	Increment direct byte	2	1	CLR	C	Clear Carry flag	1	1
INC	@Ri	Increment indirect RAM	1	1	CLR	bit	Clear direct bit	2	1
DEC	A	Decrement Accumulator	1	1	SETB	C	Set Carry flag	1	1
DEC	Rn	Decrement register	1	1	SETB	bit	Set direct Bit	2	1
DEC	direct	Decrement direct byte	2	1	CPL	C	Complement Carry flag	1	1
DEC	@Ri	Decrement indirect RAM	1	1	CPL	bit	Complement direct bit	2	1
INC	DPTR	Increment Data Pointer	1	2	ANL	C,bit	AND direct bit to Carry flag	2	2
MUL	AB	Multiply A & B	1	4	ANL	C,/bit	AND complement of direct bit to Carry	2	2
DIV	AB	Divide A by B	1	4	ORL	C,bit	OR direct bit to Carry flag	2	2
DA	A	Decimal Adjust Accumulator	1	1	ORL	C,/bit	OR complement of direct bit to Carry	2	2
LOGICAL OPERATIONS				PROGRAM AND MACHINE CONTROL.					
Mnemonic		Destination	Byte	Cyc	Mnemonic	Description	Byte	Cyc	
ANL	A,Rn	AND register to Accumulator	1	1	ACALL	addr11	Absolute Subroutine Call	2	2
ANL	A,direct	AND direct byte to Accumulator	2	1	LCALL	addr16	Long Subroutine Call	3	2
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1	RET		Return from subroutine	1	2
ANL	A,#data	AND immediate data to Accumulator	2	1	RETI		Return from interrupt	1	2
ANL	direct,A	AND Accumulator to direct byte	2	1	AJMP	addr11	Absolute Jump	2	2
ANL	direct,#data	AND immediate data to direct byte	3	2	LJMP	addr16	Long Jump	3	2
ORL	A,Rn	OR register to Accumulator	1	1	SJMP	rel	Short Jump (relative addr)	2	2
ORL	A,direct	OR direct byte to Accumulator	2	1	JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1	JZ	rel	Jump if Accumulator is Zero	2	2
ORL	A,#data	OR immediate data to Accumulator	2	1	JNZ	rel	Jump if Accumulator is Not Zero	2	2
ORL	direct,A	OR Accumulator to direct byte	2	1	JC	rel	Jump if Carry flag is set	2	2
ORL	direct,#data	OR immediate data to direct byte	3	2	JNC	rel	Jump if No Carry flag	2	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1	JB	bit,rel	Jump if direct Bit set	3	2
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1	JNB	bit,rel	Jump if direct Bit Not set	3	2
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1	JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
XRL	A,#data	Exclusive-OR immediate data to A	2	1	CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1	CJNE	A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2	CJNE	Rn,#data,rel	Comp. immed. to reg. & Jump if Not Equal	3	2
CLR	A	Clear Accumulator	1	1	CJNE	@Ri,#data,rel	Comp. immed. to ind. & Jump if Not Equal	3	2
CPL	A	Complement Accumulator	1	1	DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
RL	A	Rotate Accumulator Left	1	1	DJNZ	direct,rel	Decrement direct & Jump if Not Zero	3	2
RLC	A	Rotate A Left through the Carry flag	1	1	NOP		No operation	1	1
RR	A	Rotate Accumulator Right	1	1	Notes on data addressing modes:				
RRC	A	Rotate A Right through Carry flag	1	1	Rn	Working register R0-R7			
SWAP	A	Swap nibbles within the Accumulator	1	1	direct	128 internal RAM locations, any I/O port, control or status register			
DATA TRANSFER				Notes on program addressing modes:					
Mnemonic		Description	Byte	Cyc	addr16	Destination address for LCALL & LJMP may be anywhere within the 64-Kilobyte program memory address space.			
MOV	A,Rn	Move register to Accumulator	1	1	addr11	Destination address for ACALL & AJMP will be within the same 2-Kilobyte page of program memory as the first byte of the following instruction.			
MOV	A,direct	Move direct byte to Accumulator	2	1	rel	SJMP and all conditional jumps include an 8-bit offset byte. Range is +127/-128 bytes relative to first byte of the following instruction.			
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1	All mnemonics copyrighted © Intel Corporation 1979				
MOV	A,#data	Move immediate data to Accumulator	2	1					
MOV	Rn,A	Move Accumulator to register	1	1					
MOV	Rn,direct	Move direct byte to register	2	2					
MOV	Rn,#data	Move immediate data to register	2	1					
MOV	direct,A	Move Accumulator to direct byte	2	1					
MOV	direct,Rn	Move register to direct byte	2	2					
MOV	direct,direct	Move direct byte to direct	3	2					
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2					
MOV	direct,#data	Move immediate data to direct byte	3	2					
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1					
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2					
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1					
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	2					

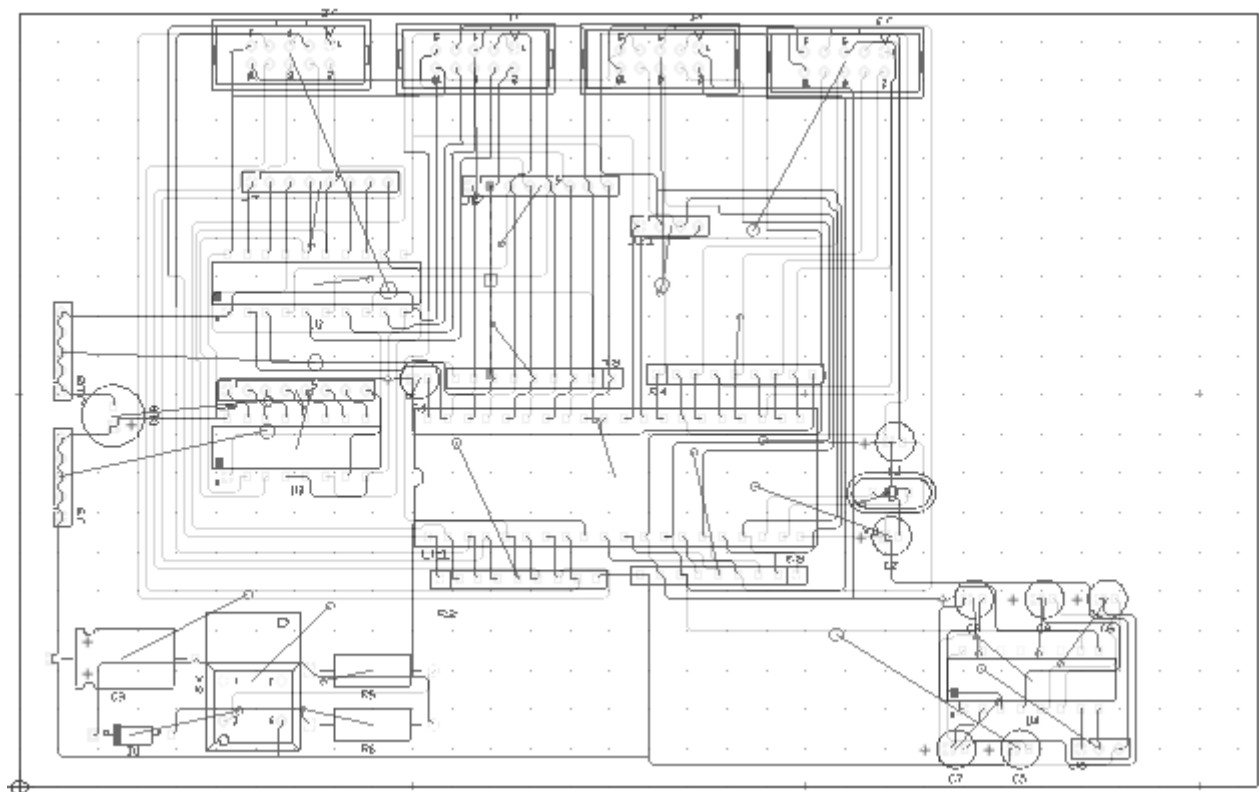
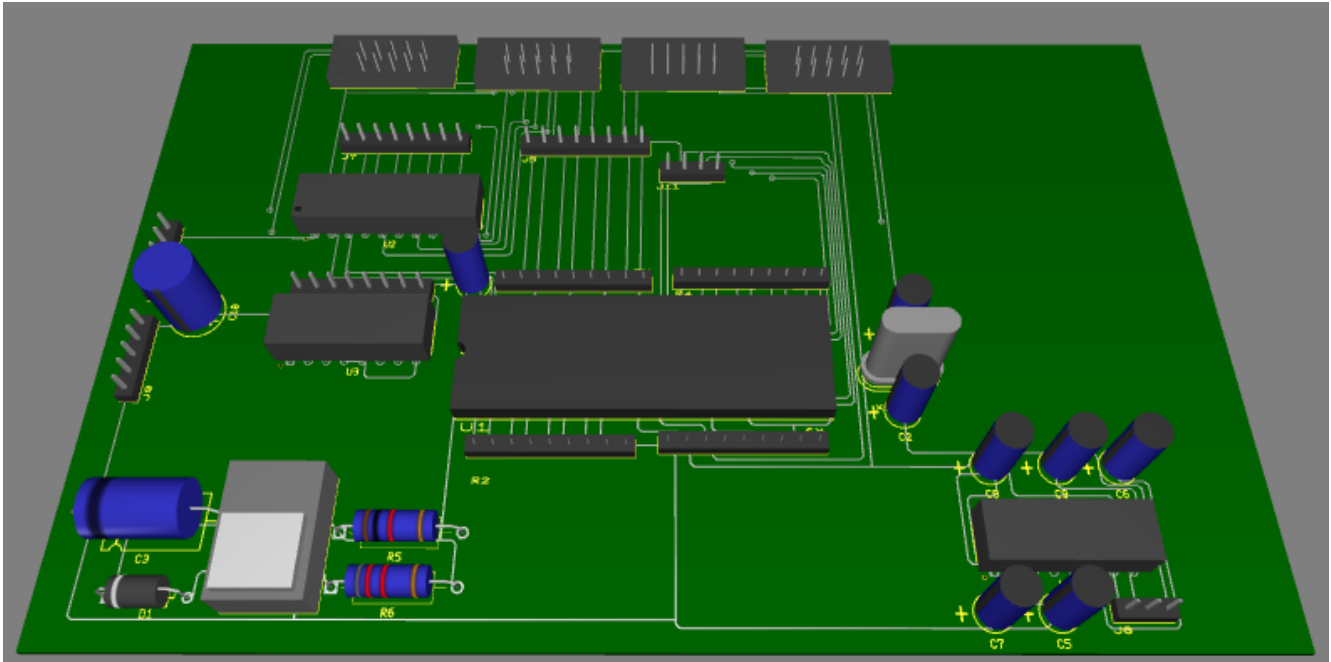
2 THE PRINTED CIRCUIT BOARD

2.1 BLOCK DIAGRAM



Block diagram of PCB for 8751





2.2 CIRCUIT DETAILS

2.2.1 BILL OF MATERIALS

Qty	Description	RefDes	Package	Type
1	CRYSTAL, HC-49/U_11MHz	X1	Generic\HC-49U	
2	CAPACITOR, 33F 10%	C1, C2	IPC-2221A/2222\CAPPR150-400X500	Ceramic
1	RESISTOR, 1k Ω 5%	R5	IPC-2221A/2222\RES1600-1000X400	Carbon Composition
1	RESISTOR, 8.2k Ω 5%	R6	IPC-2221A/2222\RES1600-1000X400	Carbon Composition
1	SCH_CAP_SYMS, PUSH_BUTTON_DPST	X2	Ultiboard\DIGTAS1	
1	CAPACITOR, 1 μ F 5%	C3	IPC-2221A/2222\CAPPA1870-1270X770	Electrolytic
1	DIODE, 1N4149	D1	IPC-2221A/2222\DO-35	
1	805x, 8051	U1	Generic\DIP-40	
4	RPACK_VARIABLE_1X8, 10k Ω	R1, R2, R3, R4	Generic\SIP-9	
7	CONNECTORS, HDR1X8	J1, J2, J3, J4, J5, J6, J7	Generic\HDR1X8	
1	CAPACITOR, 0.1 μ F 10%	C4	IPC-2221A/2222\CAPPR200-500X1100	Electrolytic
1	74F, 74F573PC	U2	IPC-2221A/2222\N20A	
1	74LS, 74LS138N	U3	IPC-2221A/2222\NO16	
1	LINE_TRANSCEIVER, MAX232E	U4	IPC-2221A/2222\DIP16	
1	CAPACITOR, 0.01 μ F 10%	C5	IPC-2221A/2222\CAPPR200-500X1100	Electrolytic
1	CONNECTORS, HDR1X3	J8	Generic\HDR1X3	
4	CAPACITOR, 1 μ F 10%	C6, C7, C8, C9	IPC-2221A/2222\CAPPR200-500X1100	Electrolytic
2	CONNECTORS, HDR1X5	J9, J10	Generic\HDR1X5	
1	CONNECTORS, HDR1X4	J11	Generic\HDR1X4	
1	CAPACITOR, 10 μ F 10%	C10	IPC-2221A/2222\CAPPR250-800X1150	Tantalum

2.2.2 NETLIST

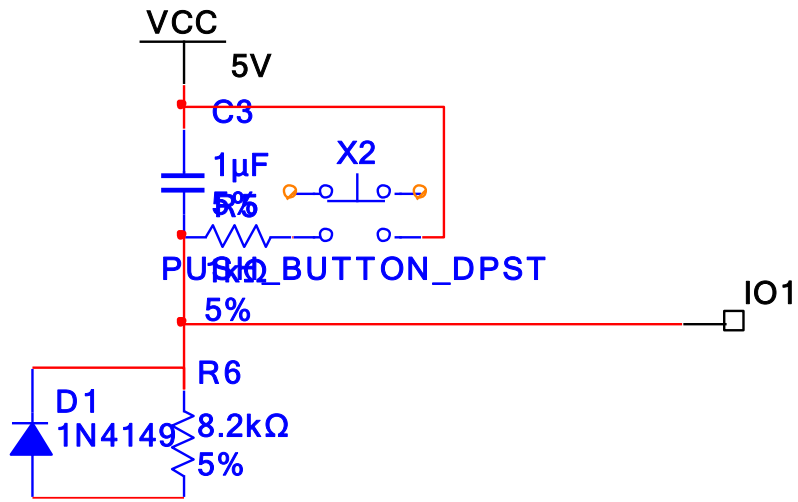
Net	Sheet	Component	Pin	Net	Sheet	Component	Pin
1	Design1	U1	P0B0AD0	9	Design1	R4	8
1	Design1	U2	1D	9	Design1	U1	P2B0A8
1	Design1	J5	P1	10	Design1	R4	7
1	Design1	R1	8	10	Design1	J3	P2
2	Design1	U1	P0B1AD1	10	Design1	U1	P2B1A9
2	Design1	J1	P2	11	Design1	U1	P2B2A10
2	Design1	U2	2D	11	Design1	R4	6
2	Design1	R1	7	11	Design1	J3	P3
2	Design1	J5	P2	12	Design1	J3	P4
3	Design1	J1	P3	12	Design1	U1	P2B3A11
3	Design1	U1	P0B2AD2	12	Design1	R4	5
3	Design1	R1	6	13	Design1	J3	P5
3	Design1	J5	P3	13	Design1	R4	4
3	Design1	U2	3D	13	Design1	U1	P2B4A12
4	Design1	R1	5	14	Design1	J3	P6
4	Design1	J5	P4	14	Design1	R4	3
4	Design1	U1	P0B3AD3	14	Design1	U1	P2B5A13
4	Design1	J1	P4	15	Design1	U1	P2B6A14
4	Design1	U2	4D	15	Design1	J3	P7
5	Design1	J5	P5	15	Design1	R4	2
5	Design1	J1	P5	16	Design1	R4	1
5	Design1	U2	5D	16	Design1	J3	P8
5	Design1	U1	P0B4AD4	16	Design1	U1	P2B7A15
5	Design1	R1	4	17	RST(SC1)	R6	1
6	Design1	U2	6D	17	RST(SC1)	D1	A
6	Design1	R1	3	18	Design1	R2	2
6	Design1	J5	P6	18	Design1	U1	P1B1T2EX
6	Design1	J1	P6	18	Design1	J2	P2
6	Design1	U1	P0B5AD5	19	Design1	J2	P1
7	Design1	U2	7D	19	Design1	R2	1
7	Design1	J1	P7	19	Design1	U1	P1B0T2
7	Design1	U1	P0B6AD6	20	Design1	U1	P1B2
7	Design1	J5	P7	20	Design1	R2	3
7	Design1	R1	2	20	Design1	J2	P3
8	Design1	U1	P0B7AD7	21	Design1	U1	P1B3
8	Design1	J5	P8	21	Design1	R2	4
8	Design1	R1	1	21	Design1	J2	P4
8	Design1	U2	8D	22	Design1	U1	P1B4
8	Design1	J1	P8	22	Design1	R2	5

9	Design1	J3	P1	22	Design1	J2	P5
Net	Sheet	Component	Pin	Net	Sheet	Component	Pin
23	Design1	U1	P1B5MOSI	34	Design1	J1	P1
23	Design1	J2	P6	35	Design1	U1	VCC
23	Design1	R2	6	35	Design1	C4	2
24	Design1	J2	P7	36	Design1	J7	P1
24	Design1	U1	P1B6MISO	36	Design1	U2	1Q
24	Design1	R2	7	37	Design1	J11	P4
25	Design1	R2	8	37	Design1	U1	^PSEN^
25	Design1	U1	P1B7SCK	38	Design1	U3	A
25	Design1	J2	P8	38	Design1	J7	P3
26	Design1	R3	8	38	Design1	U2	3Q
26	Design1	U1	P3B0RXD	40	Design1	U3	Y0
26	Design1	U4	R2OUT	40	Design1	J6	P1
26	Design1	J4	P1	41	Design1	J6	P2
27	Design1	J4	P2	41	Design1	U3	Y1
27	Design1	U4	T2IN	42	Design1	U3	Y2
27	Design1	R3	7	42	Design1	J6	P3
27	Design1	U1	P3B1TXD	43	Design1	U3	Y3
28	Design1	U1	P3B2^INT0^	43	Design1	J6	P4
28	Design1	J4	P3	44	Design1	U3	Y4
28	Design1	R3	6	44	Design1	J6	P5
29	Design1	R3	5	45	Design1	J6	P6
29	Design1	U1	P3B3^INT1^	45	Design1	U3	Y5
29	Design1	J4	P4	46	Design1	U3	Y6
30	Design1	U1	P3B4T0	46	Design1	J6	P7
30	Design1	R3	4	47	Design1	U3	Y7
30	Design1	J4	P5	47	Design1	J6	P8
31	Design1	J4	P6	48	Design1	J7	P5
31	Design1	U1	P3B5T1	48	Design1	U3	C
31	Design1	R3	3	48	Design1	U2	5Q
32	Design1	J4	P7	49	Design1	J7	P6
32	Design1	R3	2	49	Design1	U2	6Q
32	Design1	U1	P3B6^WR^	50	Design1	U2	7Q
32	Design1	J11	P2	50	Design1	J7	P7
33	Design1	J11	P3	51	Design1	J7	P8
33	Design1	R3	1	51	Design1	U2	8Q
33	Design1	J4	P8	51	Design1	U3	G1
33	Design1	U1	P3B7^RD^	53	Design1	U2	4Q
34	Design1	J11	P1	53	Design1	J7	P4
34	Design1	U2	LE	53	Design1	U3	B
34	Design1	U1	ALE^PROG^	54	RST(SC1)	R5	2

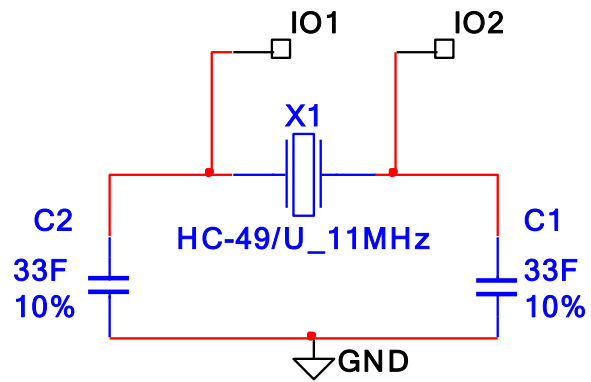
Net	Sheet	Component	Pin	Net	Sheet	Component	Pin
54	RST(SC1)	X2	3	GND	Design1	U1	GND
58	Design1	J7	P2	GND	Design1	GND	GND
58	Design1	U2	2Q	GND	Design1	U4	GND
61	Design1	C5	2	GND	Design1	J10	P2
61	Design1	U4	VCC	GND	Design1	J10	P1
62	Design1	U4	R2IN	GND	Design1	U3	GND
62	Design1	J8	P2	GND	Design1	J10	P4
63	Design1	U4	T2OUT	GND	Design1	GND	GND
63	Design1	J8	P1	SC1/IO1	Design1	U1	RST
64	Design1	U4	V-	SC1/IO1	RST(SC1)	D1	K
64	Design1	C6	2	SC1/IO1	RST(SC1)	R5	1
65	Design1	U4	V+	SC1/IO1	RST(SC1)	C3	2
65	Design1	C7	1	SC1/IO1	RST(SC1)	R6	2
66	Design1	C7	2	SC2/IO1	Design1	U1	XTAL2
66	Design1	C6	1	SC2/IO1	Crystal Circuit(SC2)	C2	2
67	Design1	U4	C1+	SC2/IO1	Crystal Circuit(SC2)	X1	X1
67	Design1	C8	1	SC2/IO2	Design1	U1	XTAL1
68	Design1	U4	C2+	SC2/IO2	Crystal Circuit(SC2)	X1	X2
68	Design1	C9	1	SC2/IO2	Crystal Circuit(SC2)	C1	1
69	Design1	C8	2	VCC	Design1	VCC	VCC
69	Design1	U4	C1-	VCC	Design1	VCC	VCC
70	Design1	C9	2	VCC	Design1	U2	VCC
70	Design1	U4	C2-	VCC	RST(SC1)	VCC	VCC
GND	Crystal Circuit(SC2)	C2	1	VCC	Design1	VCC	VCC
GND	Crystal Circuit(SC2)	GND	GND	VCC	RST(SC1)	C3	1
GND	Crystal Circuit(SC2)	C1	2	VCC	Design1	VCC	VCC
GND	Design1	GND	GND	VCC	Design1	VCC	VCC
GND	Design1	U3	~G2A	VCC	Design1	J9	P3
GND	Design1	U3	~G2B	VCC	Design1	J9	P4
GND	Design1	C10	2	VCC	Design1	J9	P5
GND	Design1	J10	P3	VCC	Design1	J9	P2
GND	Design1	GND	GND	VCC	Design1	J9	P1
GND	Design1	J8	P3	VCC	RST(SC1)	X2	4
GND	Design1	GND	GND	VCC	Design1	C5	1
GND	Design1	GND	GND	VCC	Design1	VCC	VCC
GND	Design1	GND	GND	VCC	Design1	VCC	VCC
GND	Design1	U2	GND	VCC	Design1	U1	^EA^VPP
GND	Design1	J10	P5	VCC	Design1	VCC	VCC

GND	Design1	U2	~OC	VCC	Design1	R1	9
GND	Design1	GND	GND	VCC	Design1	R2	9
Net	Sheet	Component	Pin				
VCC	Design1	R4	9				
VCC	Design1	R3	9				
VCC	Design1	C4	1				
VCC	Design1	VCC	VCC				
VCC	Design1	U3	VCC				
VCC	Design1	C10	1				

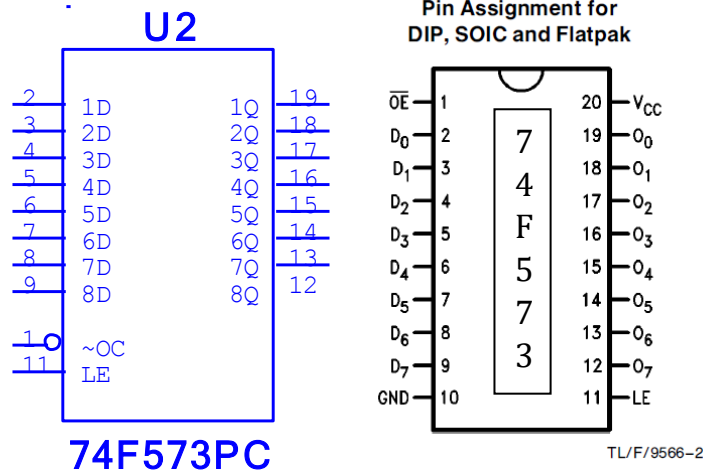
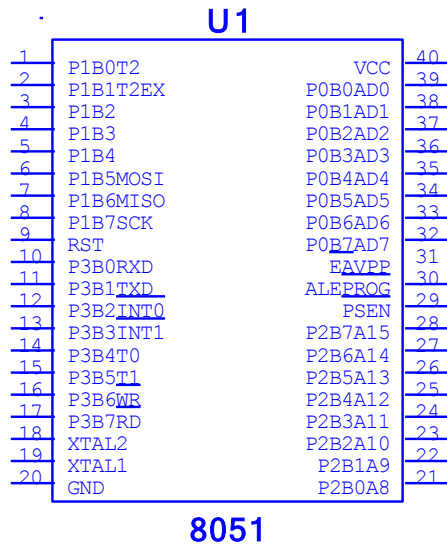
2.3 CIRCUIT COMPONENTS



Reset Circuit (SC-1)



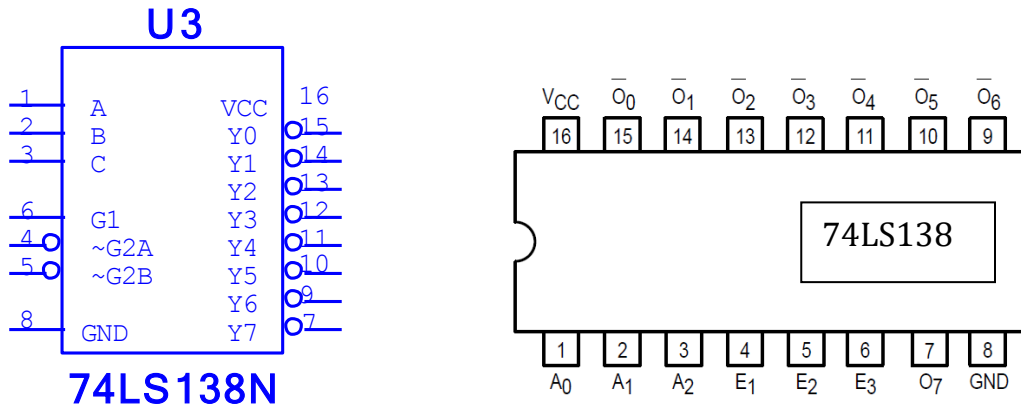
Crystal Circuit (SC-2)



The 'F573 contains eight D-type latches with 3-state output buffers.

Pin 11 or LE (Latch enable) goes high the D-inputs (1-9) are latched and outputted when Pin 1(OC') is given a logic low.

Its primary use in the circuit is that it will latch the lower Address bus on receiving the ALE from the 8051



74LS 138 is a 3-8 decoder

It has been used here to provide 8 chip select lines of the following addresses:

Address	A7	A6	A5	A4	A3	A2	A1	A0	CS
	E1(4)	X	X	A2(3)	A1(2)	A0(1)	X	X	
80	1	0	0	0	0	0	0	0	Y0
84	1	0	0	0	0	1	0	0	Y1
88	1	0	0	0	1	0	0	0	Y2
8C	1	0	0	0	1	1	0	0	Y3
90	1	0	0	1	0	0	0	0	Y4
94	1	0	0	1	0	1	0	0	Y5
98	1	0	0	1	1	0	0	0	Y6
9C	1	0	0	1	1	1	0	0	Y7

The above table has been provided for the reference of the programmer. This table provides the addresses to be used to provide the corresponding chip selects

MAX232(A) DIP Package Pin Layout No.	Name	Purpose	Signal Voltage	Capacitor Value MAX232	Capacitor Value MAX232A
1	C1+	+ connector for capacitor C1	capacitor should stand at least 16V	1 μ F	100nF
2	V+	output of voltage pump	+10V, capacitor should stand at least 16V	1 μ F to VCC	100nF to VCC
3	C1-	- connector for capacitor C1	capacitor should stand at least 16V	1 μ F	100nF
4	C2+	+ connector for capacitor C2	capacitor should stand at least 16V	1 μ F	100nF
5	C2-	- connector for capacitor C2	capacitor should stand at least 16V	1 μ F	100nF
6	V-	output of voltage pump / inverter	-10V, capacitor should stand at least 16V	1 μ F to GND	100nF to GND
7	T2out	Driver 2 output	RS-232		
8	R2in	Receiver 2 input	RS-232		
9	R2out	Receiver 2 output	TTL		
10	T2in	Driver 2 input	TTL		
11	T1in	Driver 1 input	TTL		
12	R1out	Receiver 1 output	TTL		
13	R1in	Receiver 1 input	RS-232		
14	T1out	Driver 1 output	RS-232		
15	GND	Ground	0V	1 μ F to VCC	100nF to VCC
16	VCC	Power supply	+5V	see above	see above

V+(2) is also connected to VCC via a capacitor (C3). V-(6) is connected to GND via a capacitor (C4). And GND(16) and VCC(15) are also connected by a capacitor (C5), as close as possible to the pins.

2.4 PCB DESIGN GUIDELINES

2.4.1 CIRCUIT BOARD LAYOUT (COMPONENT SELECTION AND PLACEMENT)

1. Connectors should be located on one edge or on one corner of a board. This makes it much easier to fix the potential of all connectors to the same reference. This is extremely important for boards with high-frequency components that will not be housed in a shielded enclosure.
2. A device on the board that communicates with a device off the board through a connector should be located as close as possible (e.g. within 2 cm) to that connector. This is another way of saying keep you I/O traces short. If an I/O trace comes too far on to the board, it is more likely to couple noise to or from other circuits on the board.
3. Active digital components should be selected that have maximum acceptable off-chip transition times. This means we need to control the rise and fall times.
4. All off-board communication from a single device should be routed through the same connector. Common-mode currents are much easier to control in wires that are part of the same cable bundle.
5. Components not connected to an I/O net should be located at least 2 cm away from I/O nets and connectors. This reduces the noise coupled on or off the board through the I/O.
6. Any component (such as a clock oscillator) that communicates exclusively with another component (such as a clock driver) should be located close to that component; especially if the communication is a high-frequency signal.

2.4.2 CIRCUIT BOARD LAYOUT (TRACE ROUTING)

1. Critical signal traces should be buried between power/ground planes.
2. Metal plane layers both above and below signal traces help to contain the electric and magnetic fields associated with signals. Signals routed between planes are less likely to radiate and less likely to be susceptible to external noise.
3. No trace unrelated to I/O should be located between an I/O connector and the device(s) sending and receiving signals using that connector. This is another way of saying "Don't route high-speed signals between connectors."
4. All power planes and traces should be routed on the same layer. This is almost always a very good idea. It ensures that different voltage planes don't overlap and helps to minimize noise coupling between power buses. An exception might be made for plane pairs such as + and - 12 V, where it is generally best to maintain balance by routing identical planes on different layers.
5. Critical nets should be routed at least 2X from the board edge, where X is the distance between the trace and its return current path. Traces routed too close to the board edge exhibit poor field containment and more easily couple to external cables and other objects.
6. Signals with high-frequency content should not be routed beneath components used for board I/O. "Keep noisy signals and components away from the I/O".
7. Be sure to maintain balance when routing the differential signals in order to ensure that one can derive the noise canceling benefits of a differential signal.
8. The length of high-frequency nets should be minimized. "Control the loop areas."
9. On a board with power and ground planes, no traces should be used to connect to power or ground. Connections should be made using a via adjacent to the power or ground pad of the component. Traces on power or ground take up space on the board and add inductance to the connection. There is generally no good reason to have these traces, so they should be eliminated.

REFERENCES

A. Text Books :

1. Kleitz, William, "Digital and Microprocessor Fundamentals", Prentice-Hall, New Jersey, 1990.
2. Hall, Douglas V., "Microprocessors and Interfacing. Programming and Hardware", 2nd Ed., Macmillan / McGraw-Hill, Illinois, 1992.
3. Peatman, John B., "Design with Microcontrollers", McGraw-Hill Book Company, New York, 1988.
4. Wiatrowski C and House C.H., "Logic Circuits and Microcomputer systems", McGraw-Hill International Book Co., New Delhi, 1980.
5. Motorola, " M68HC11 Reference Manual", Motorola Inc., Pnoenix, Arizona, 1990.
6. Intel, "The MCS 96 Users's Manual", Intel Corporation, Santa Clara, CA, 1984.
7. Intel, "Embedded Controller Handbook, Vol I 8-Bit", Intel Corporation, Santa Clara, CA, 1988.
8. Intel, "Embedded Controller Handbook, Vol II 16-Bit", Intel Corporation, Santa Clara, CA, 1988.
9. Intel, "Embedded Control Applications Handbook", Intel Corporation, Santa Clara, CA, 1989.
10. Philips, "80C51 Based 8-Bit Microcontrollers" Philips Semiconductors, Netherlands, 1993.
11. Hintz, K. and Daniel Tabak, "Microcontrollers, Architecture, Implementation & Programming", McGraw-Hill International Edition, New York, 1992.

B. Web sites

1. http://www.cvel.clemson.edu/emc/tutorials/guidelines/Good_Guidelines.html - EMC Design Guidelines by Clemson Vehicular Electronic Laboratory.
2. http://en.wikipedia.org/wiki/Printed_circuit_board - PCB, Wikipedia Page.
3. http://en.wikibooks.org/wiki/Serial_Programming/MAX232_Driver_Receiver Wikibook on Serial programming using Max232.

APPENDIX
