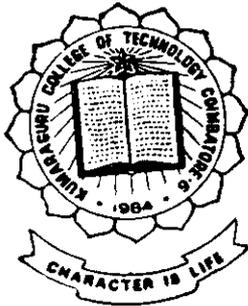


Micro Controller Based Printer Sharer

Project Report



P-209

Submitted by

K. B. Aswin
P. R. Muralidharan
K. Sundararaju

Under the Guidance of

Mr. U. Sivakumar, B.E.,

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
BACHELOR OF ENGINEERING
IN ELECTRICAL AND ELECTRONICS ENGINEERING
OF THE BHARATHIAR UNIVERSITY, COIMBATORE

1994-95

Department of Electrical and Electronics Engineering
Kumaraguru College of Technology

Coimbatore - 641 006

ACKNOWLEDGEMENT

We are highly grateful to **Mr. V. SIVAKUMAR B.E.**, Associate Lecturer in Electrical and Electronics Engineering Department for his able guidance and readiness to extend all the required help to do this project.

P-209

We express our sincere thanks to our beloved professor **DR. K.A. PALANISWAMY B.E.**, M.Sc (ENGG)., Ph.D., M.I.S.T.E., C.ENGG(I)., F.I.E., Head of the Department of Electrical and Electronics Engineering for his constant inspiration and continued encouragement to do this project.

We thank our principal **DR. S. SUBRAMANIAN B.E.**, M.Sc (ENGG)., Ph.D., SMIEEE for providing all facilities in the college to carry out this project.

We express our gratitude to **MR. E. PALANIAPPAN B.E.**, **MR. D. SENTHIL KUMAR B.E.**, AGT ELECTRONICS, Coimbatore, and **MR. E. JAGANATHAN B.E.**, GURUKULAM ELECTRONICS for the help in the successful completion of this project.

Thanks are due to all those who directly or indirectly helped us for the completion of this project. Thanks are also due to members of staff in Electrical and Electronics Engineering Department for their continued support and help.

SYNOPSIS

The Microcontroller based printer sharer holds its importance on time consumption and cost. Here the concept of one printer for one PC has been replaced by one printer for many PC's. The microcontroller chooses a particular PC according to the logic programmed in it. This logic fed to the decoder enables the tratchpatch latch which acknowledges the PC, which in turn sends the data through the another latch to the rotating buffer in the microcontroller. The microcontroller is programmed in such a way that all the PC's are given equal priority in a sequence to send their data into the buffer. The priority for printing is given to that buffer in which the information for storing starts first.

The present project deals with the design of 8031 microcontroller and 8255A programmable pheripheral interface card which are used to control the PC's and the printer. The hardware also contains decoders and latches to select the

PC's and to transfer the data in to the rotating buffer . An interactive software has been developed in " C " language to handle all the signals and interrupts. A cross complier has been used which converts " C " language in to 8031 source code. Here an interface board has been designed in such a way that 2 PC's share a single printer.

CONTENTS

	PAGE NO.
CERTIFICATE	(i)
ACKNOWLEDGEMENT	(ii)
SYNOPSIS	(iii)
CONTENTS	
I. INTRODUCTION	1
1.1 Personal computers	2
1.2 Interfacing unit using 8031	2
1.3 Printer	3
II. MICRO-CONTROLLER	6
2.1 Introduction	6
2.2 Hardware description	7
2.3 Special function registers	8
2.4 Port structures and operations	10
2.5 Port loading and interface	11
2.6 Timer/counter	12
2.7 Timer/counter mode control (TMOD) register	12

2.8 Standard serial interface	16
2.9 Serial interface	18
2.9 Serial interface	18
2.10 Multiprocessor communication	18
2.11 Serial port control (SCON) registers	19
2.12 Interrupts	20
2.13 Reset	21
III. 8255-A PROGRAMMABLE PERIPHERAL INTERFACE	27
IV. 8279 PROGRAMMABLE DISPLAY INTERFACE	35
4.1 Scan section	36
4.2 Display section	36
4.3 Interface section	37
V. HANDSHAKING	40
VI. PARALLEL INTERFACE	45
VII. INTERFACING	52
7.1 Interfacing unit	53
7.2 Interfacing memory with 8031	54
7.3 Interfacing 8255A	55
7.4 Interfacing 8279	56
7.5 Protection circuits	56

VIII. PCB FABRICATION	66
8.1 PCB drawing in smart work	66
8.2 Fabrication	67
8.3 Testing	63
IX. SOFTWARE	73
X. CONCLUSION	83
REFERENCES	84

CHAPTER I

INTRODUCTION

With the help of micro controller based printer sharer, one printer can be shared by 2 PC's and can be extended to any number of PC's. Hence it becomes economical when compared to one PC for one printer.

The block diagram of a overview of the system connected to the printer is shown in the Fig.1.1.

OVERVIEW OF THE SYSTEM

The system as a whole consists of three areas namely

1. The personal computers
2. Interfacing unit using 8031
3. Printer.

1.1 PERSONAL COMPUTERS

The personal computer could be of any form available. Whenever a user wants to print a matter, he gives an appropriate DOS command to the PC which is identified by the microcontroller. Whenever a PC needs its data to be printed, the microcontroller identifies the strobe signal and activates the buffer and the required task is performed.

1.2 INTERFACING UNIT USING 8031

The interfacing unit designed is placed in between the PC and the printer. This unit consists of latches, decoders, microcontrollers and programmable peripheral interface.

The strobe, Acknowledge, Busy, Printer empty signals of PC are connected to 74LS373 latch and the data lines of the PC are connected to another latch. The decoder 74138 selects the particular latch 74LS373 according to the logic pro-

grammed in the microcontroller. Any number of PC's can be connected in parallel through the data lines and the latches.

The microcontroller sends busy signals to all the PC's at a time. Then it checks for a strobe signal from each PC's for a fixed period of time in a particular sequence. Once it receives the strobe signal it collects the data and places it on the rotating buffer. A separate rotating buffer is created for each PC. The buffer in which the information starts storing first is activated by the microcontroller to be printed by the printer.

1.3 PRINTER

The printer could be of any brand provided it has the suitable adapter for interfacing. The acknowledgement and the busy signals of the printer is given to the interfacing unit to perform its task effectively. The printers usually consists of two ports namely a serial port and a parallel

port, the commercially available such ports are RS 232C and Centronics respectively. The RS 232C is a 25 pin serial port where as the Centronics is a 32 pin serial port. The printer which we use do have a Centronics adapter.

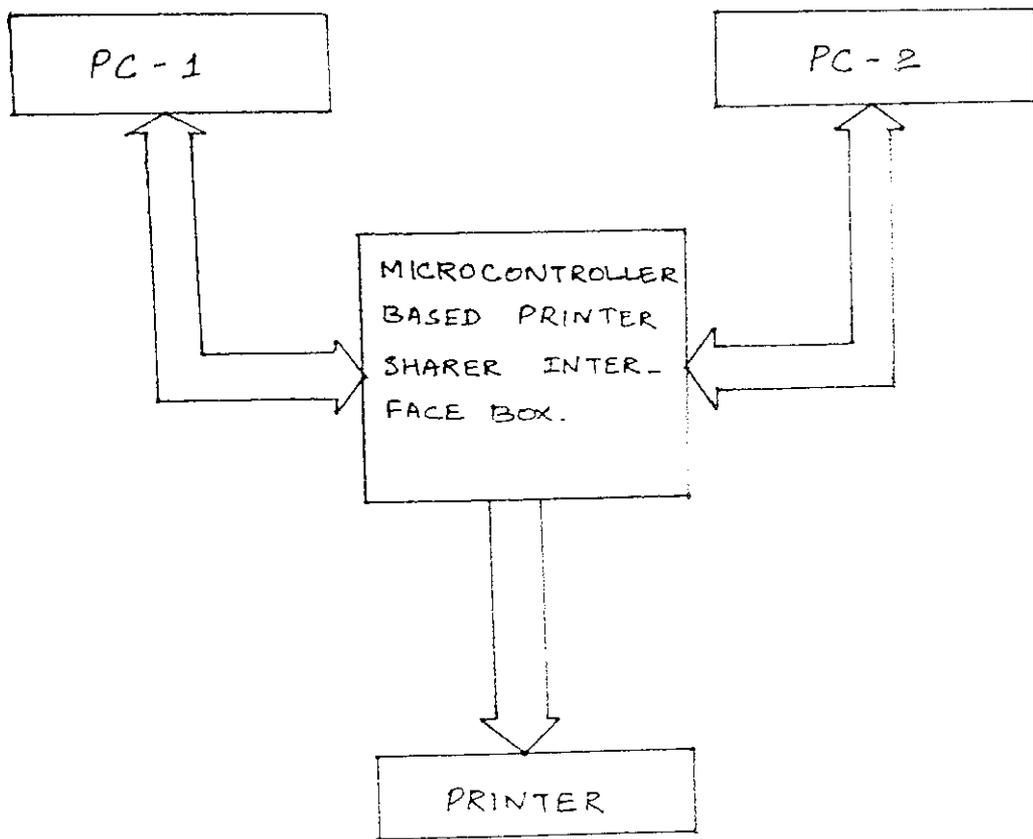


Fig. 1.

SCHAMATIC DIAGRAM OF MICROCONTROLLER BASED PRINTER SHAR

CHAPTER II

MICROCONTROLLER

2.1 INTRODUCTION

The 8051 family of products is based on the industry standard for 8-bit high performance microcontroller. The architecture for the family has been optimized for sequential real time control applications. The 8051 family of products are used in a wide range of applications from those that are relatively simple to applications in medical instrumentation and automobile control system. All of the devices included in the family are available in versions that have either internal ROM, EPROM or CPU only.

In this project 8031 micro controller which does not have any internal program memory is used for controlling the input to the printer . Fig.2.1 gives the block diagram of 8031.

2.2 HARDWARE DESCRIPTION

The hardwares used in a microcontroller are

1. 8 bit CPU optimized for control applications.
2. Extensive boolean processing (single bit logic) capabilities.
3. 32 bi directional and individual addressable I/O lines.
4. 128 bytes of an chip data RAM.
5. Two 16 Timer / Counters.
6. Full Duplex UART.
7. Source Interrupt structure with two priority levels.
8. On chip clock oscillator.
9. 64 KB programmable memory address space.
10. 64 KB data memory address space.
11. 40 pin DIP

The port drives and how they function both as ports and for port 0 and port 2 in bus operation, the timer/counters, the serial interface, the interrupt system and reset are described below.

SYMBOL	POSITION	NAME AND SIGNIFICANCE
CY	PSW.7	Carry Flag.
AC	PSW.6	auxillary carry flag (for BCD operations)
F0	PSW.5	flag 0
RS1	PSW.4	register bank select control bits 0 set
RS0	PSW.3	cleared by software to determine working register bank.
OV	PSW.2	overview flag.
-	PSW.1	user definable flag.
P	PSW.0	parity flag.

STACK POINTER

The stack pointer register is 8 bit wide. PUSH and CALL (POP) executions can be done.

DATA POINTER

DPTR consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16 bit address.

PORTS 0 - 3

P0, P1, P2 and the SFR latches of port 0,1,2 and 3 respectively.

2.4 PORT STRUCTURES AND OPERATIONS

All ports in 8051 family are bi-directional. Each consists of a latch as an output driver and an input buffer.

The output drivers of ports 0 and 2 and the input buffer of port 0 are used to access the external memory. In this application port 0 output, the low byte of external memory address, time multiplexed with the byte being written or read.

All the port 3 pins are multi-functional. They are port pins, but also serve the functions of varied features as listed below.

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

2.5 PORT LOADING AND INTERFACING

The output buffers of port 1,2 and 3 can each drive 4 LS TTL inputs. These ports on HMOS versions can be driven

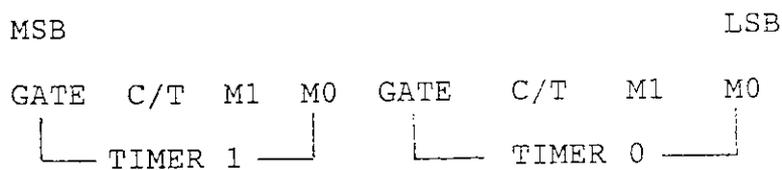
in a normal manner by any TTL or NMOS circuit. Both HMOS and CHMOS pins can be driven by open collector and open drain outputs but note that 0 to -1 transition will not be fast.

2.6 TIMER/COUNTER

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

In the counter function a register is incremented in response to a 1 or 0 transition at the corresponding external input pin t0 or t1. The timer 0 and 1 have four operating modes to select.

2.7 TIMER/COUNTER MODE CONTROL (TMOD) REGISTER



GATE - Gating control when set. Timer/counter 'x' is enabled only while $\overline{\text{TNTX}}$ pin is high and 'IRX' control pin is set. When cleared Timer is enabled whenever 'TRX' control bit is set.

C/ $\overline{\text{T}}$ - Timer/Counter selector Clear for timer operation (input from internal system clock). Set for counter operation.

M1	M0	: OPERATION MODE
0	0	8048 Timer TLX serves as 5 bit pre-scaler.
0	1	16 - bit timer/counter. THX and TLX are cascaded. There is no pre-scaler.
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)TLO is an 8 bit timer/counter controlled by standard timer 0 control bits. THO is an 8 bit timer only, controlled by timer 1 control bits.
1	1	(Timer 1) Timer/Counter 1 stopped.

MODE 0

Putting either timer into mode 0 makes it look like a timer, which is an 8 bit counter with a divide by 32 prescaler. Fig. 2.3 shows the mode 0 operation as it applies to timer 1.

In this mode the timer register is configured as a 32 bit register. As the counter rolls over, from all 1's to all 0's it sets the timer interrupt flag $TF1 = 1$ and either $GATE = 0$ or $INT1 = 1$. Setting $GATE = 1$ allows the lower to be controlled by external input $INT1$ to facilitate pulse width measurements. $TR1$ is a control bit in the special function register $TCON$ $GATE$ is in $TMOD$.

The 13 bit register consists of all 8 bits of $TH1$ as the lower 5 bits of $TL1$. The upper 3 bits of $TL1$ are indeterminate and should be ignored. Setting the run flag $TR1$ does not clear the register.

Mode 0 operation is the same for timer 0 as timer 1. Substitute TRO, TF0 and INTO for the corresponding timer 1 signals in figure. There are two different gate bits, one for timer 1 (TMOD.7) and the other for timer 0 (TMOD.3).

MODE 1

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

MODE 2

The mode 2 configuration, the timer register is an 8 bit (TL1) with automatic reload as shown in Fig.2.4.

MODE 3

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

2.8 STANDARD SERIAL INTERFACE

MSB
LSB
TF1
TR1
TF0
TR0
IE1
IT1
IE0
IT0

SYMBOL	POSITION	NAME AND SIGNIFICANCE
TF1	TCON.7	Timer 1 overflow flag.
TR1	TCON.6	Timer 1 run control bit set/cleared by software to turn Timer/Counter ON/OFF.
TF0	TCON.5	Timer 0 overflow flag set by hardware on timer/counter overflow.
TR0	TCON.4	Timer 0 run control bit.
IE1	TCON.3	Interrupt 1 edge flag set by hardware when external interrupt edge detected. Cleared when interrupt processed.

IT1	TCON.2	Interrupt 1 type control bit/ cleared by software to specify falling edge/ low level triggered external interrupts.
IE0	TCON.1	Interrupt 0 edge flag set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	TCON.0	Interrupt 0 type control bit set / cleared by software to specify falling edge/low level triggered external interrupts.

2.9 SERIAL INTERFACE

The serial port is a full duplex, meaning it can transmit and receive simultaneously. It is also 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 as special function registers SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

2.10 MULTIPROCESSOR COMMUNICATION

Modes 2 and 3 in serial interface are having the provision for the multiprocessor communication. 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 slaves. Using the 9th bit and SM 2 values with

mode 2 and 3 will decide the address. It has no value in mode 0 and 1.

2.11 SERIAL PORT CONTROL REGISTERS (SCON)

The serial port control and status register is a special function register. SCON is described as follows.

MSB							LSB	
SM0	SM1	SM2	REN	TB8	RB8	T1	RL	

SM0, SM1	Specify the serial port mode.
SM2	Enables the multiprocessor communication feature
REN	Enables the reception
TB8	Set or clear by the software is desired.
RB8	In 9th pin it is received.
T1	Transmitted interrupt flag.
R1	Receive interrupt flag.

2.12 INTERRUPTS

The 8031 provides 5 interrupt sources. These are shown in Fig.2.5. The external interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level activated or transition activated, depending on bits IT0 and IT1 in register TCON. The flags that actually generate these interrupt bits are bit IE0 and IE1 in TCON. When the external interrupt is generated, the flag that is generated is cleared by the hardware the service routine is vectored to only if the interrupt is transition activated.

PRIORITY OF INTERRUPT

SOURCE	PRIORITY LEVEL	VECTOR ADDRESS
IE0	Highest	0003H
TE0		000BH
IE1		0013H
TF1		001BH
R1+T1	Lowest	0023H

2.13 RESET

The reset input is the RST pin which is the input to a schmitt trigger. A reset is accomplished by holding the RST pin high for atleast two machine cycles, while the oscillator in running. The CPU responds by generating an external reset.

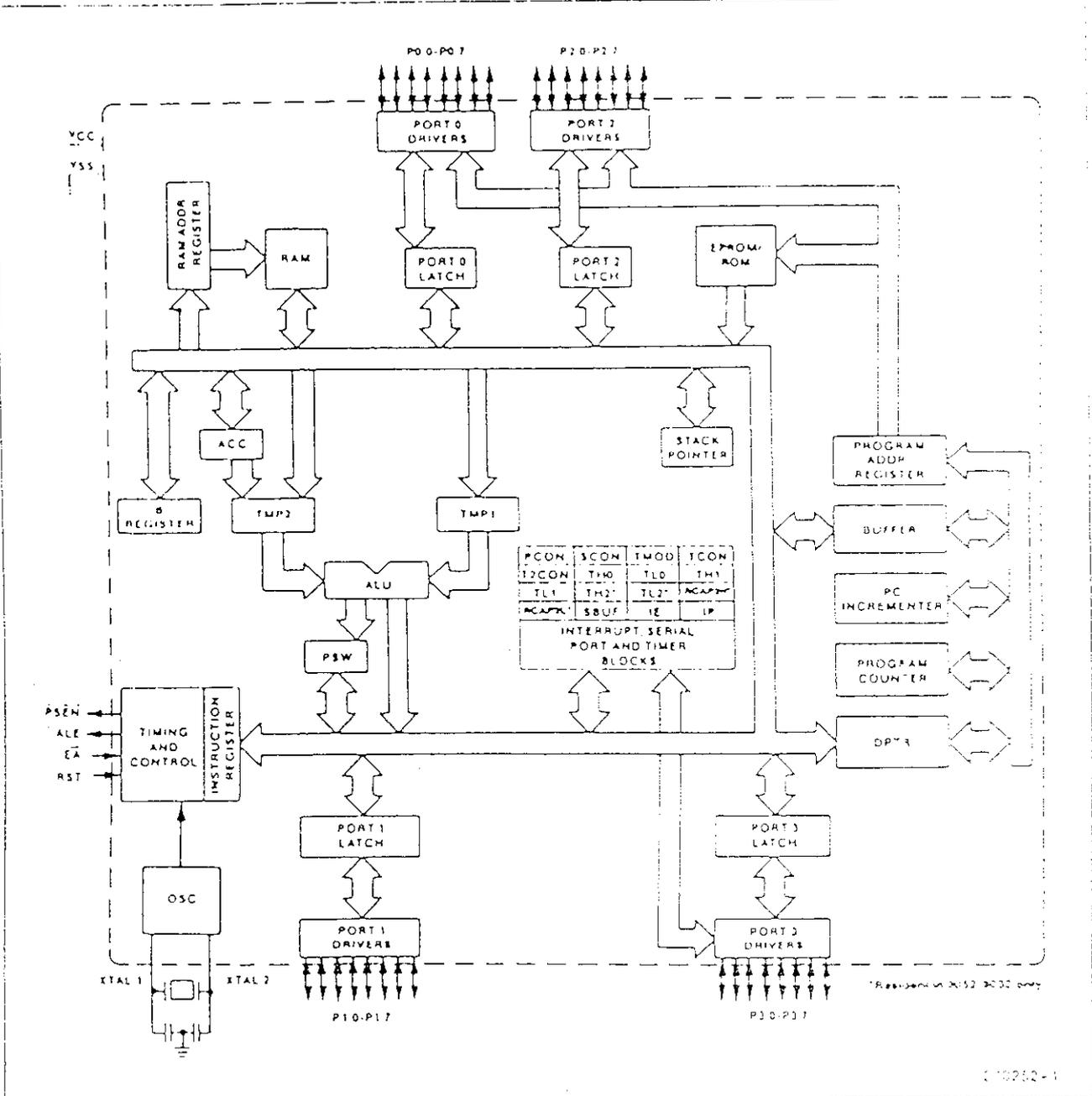


Fig. 2.1. ARCHITECTURE OF 8031 .

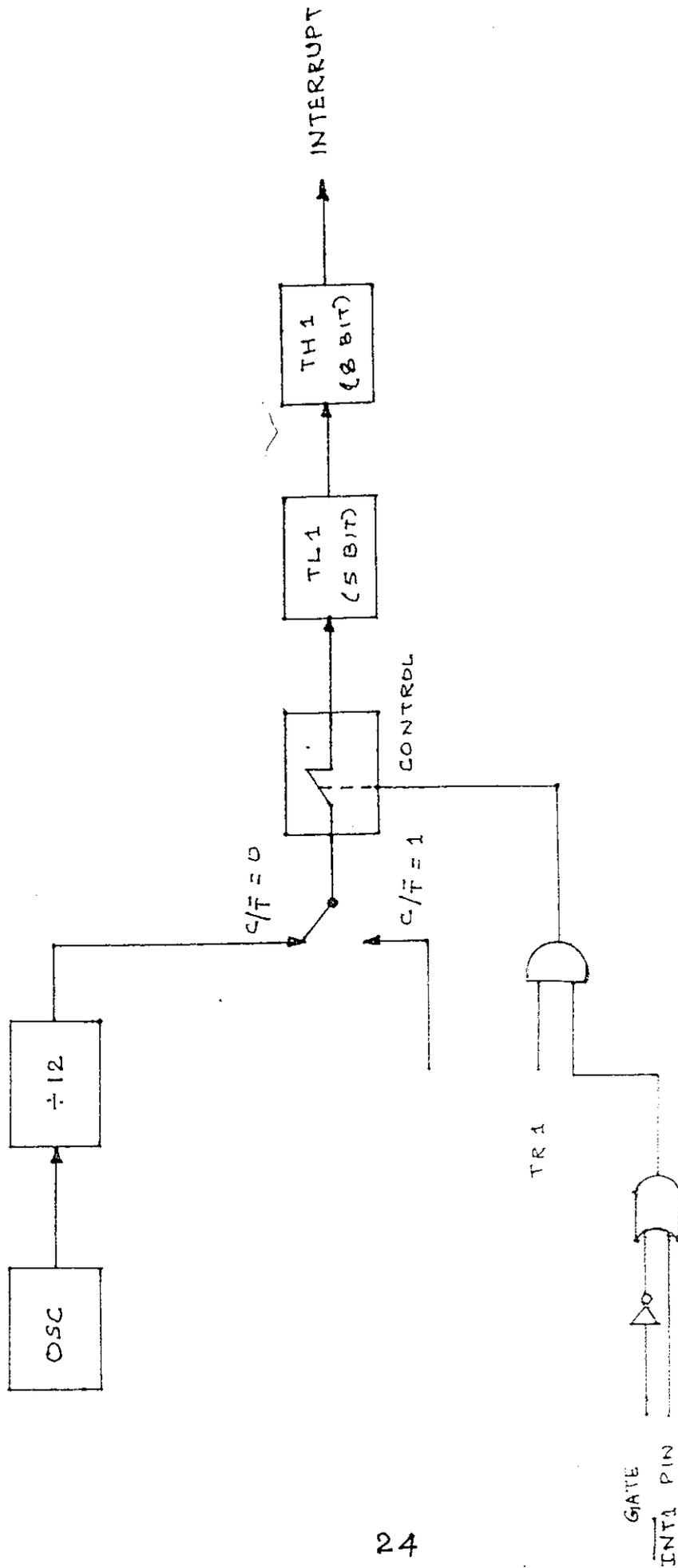


Fig. 2.3. TIMER/COUNTER 1 MODE 2 : 8 BIT AUTO RELOAD.

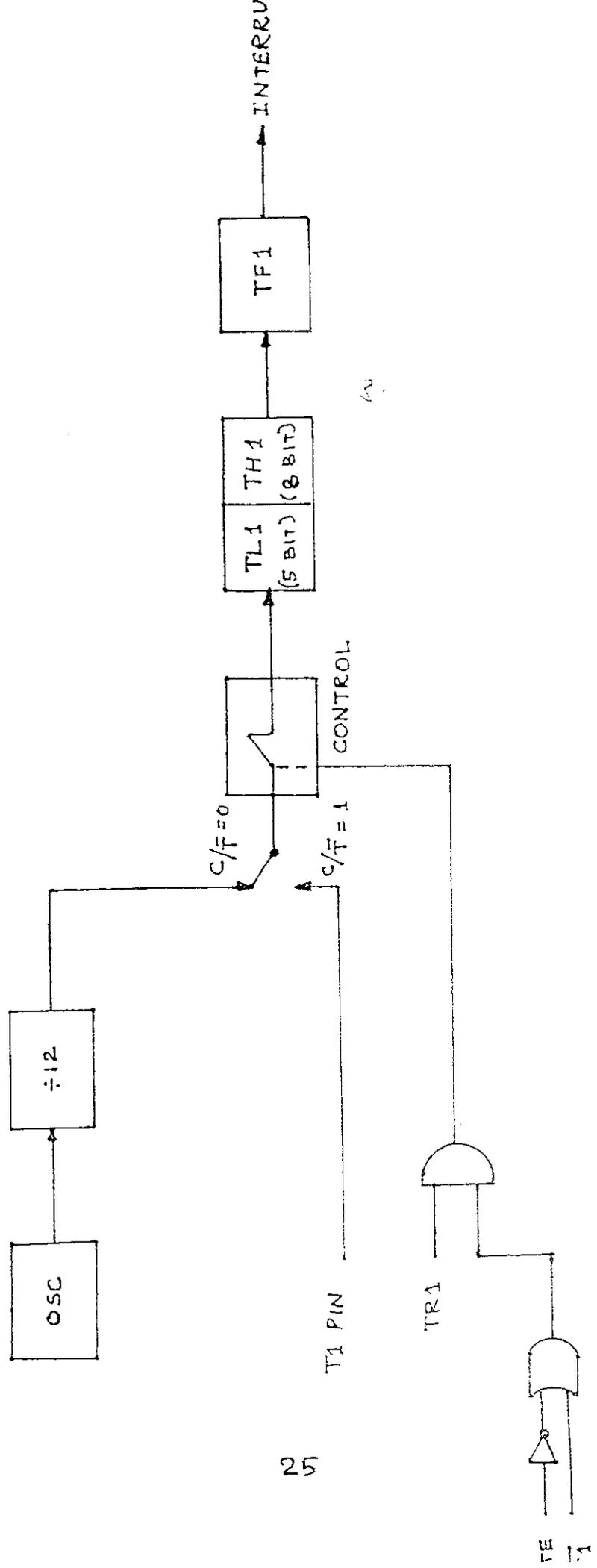
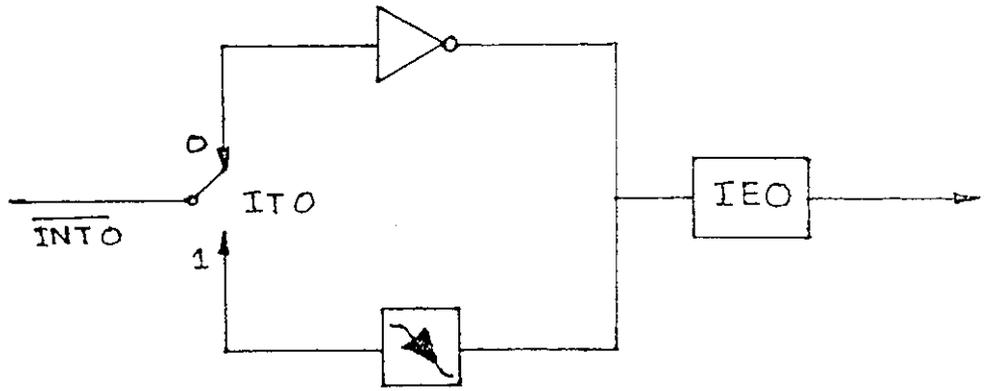
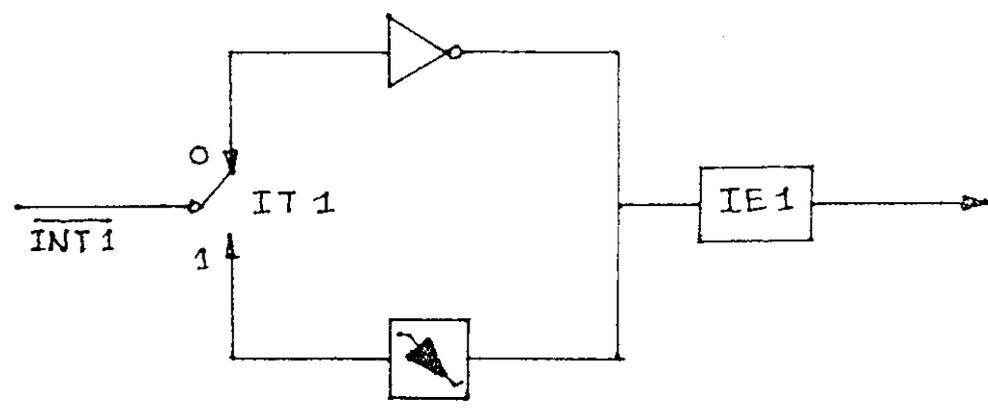


Fig. 2.4f. TIMER/COUNTER 1 MODE 0 : 13 BIT COUNTER .



TFO



TF 1

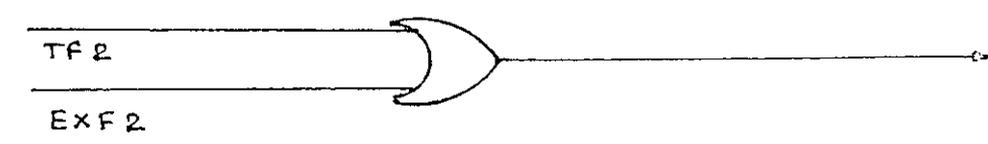
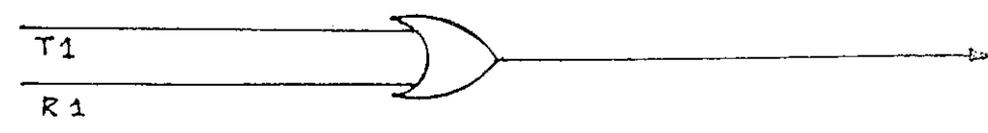


Fig. 2.5. INTERRUPT SOURCES

CHAPTER III

8255-A PROGRAMMABLE PERIPHERAL INTERFACE

The 8255-A is widely used programmable, parallel input output device. The pin configuration of 8255-A is shown in fig 3.1. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O. It is flexible, versatile and economical (When multiple I/O ports are required). It is an important general purpose I/O device that can be used with almost any microprocessors.

The 8255-A has 24 I/O pins that can grouped primarily in two 8 bit parallel ports : A and B, the remaining 8 bits as port C. The 8 bits of port C can be used as individual bits or be grouped in two four bit ports, C upper (CU) and C lower (CL) as in Fig 3.2. The functions of these ports are defined by writing a control word in the control register.

Fig 3.3. shows all the functions of the 8255-A classified according to two modes. The Bit Set / Reset (BSR) mode and I/O mode. The BSR mode is used to set or reset the bits in port C. The I/O mode is further divided into three modes: MODE 0, MODE 1 and MODE 2, all ports function as simple I/O ports. MODE 1 is a handshake mode whereby port A and/or port B use bits from port C as handshake signals.

In the handshake mode, two types of I/O data transfer can be implemented: Status check and interrupt. In MODE 2, port A can be set up for bi-directional data transfer using handshake signals from port C and port B can be set up either in MODE 0 or MODE 1.

3.1 BLOCK DIAGRAM OF 8255-A

The fig 3.4 shows two 8 bit ports (A and B), two four bit ports (CU and CL). It's a simplified but expanded version of the internal structure including a control register. Port C performs functions similar to that of the status register in addition to providing handshake signals.

Pin Configuration

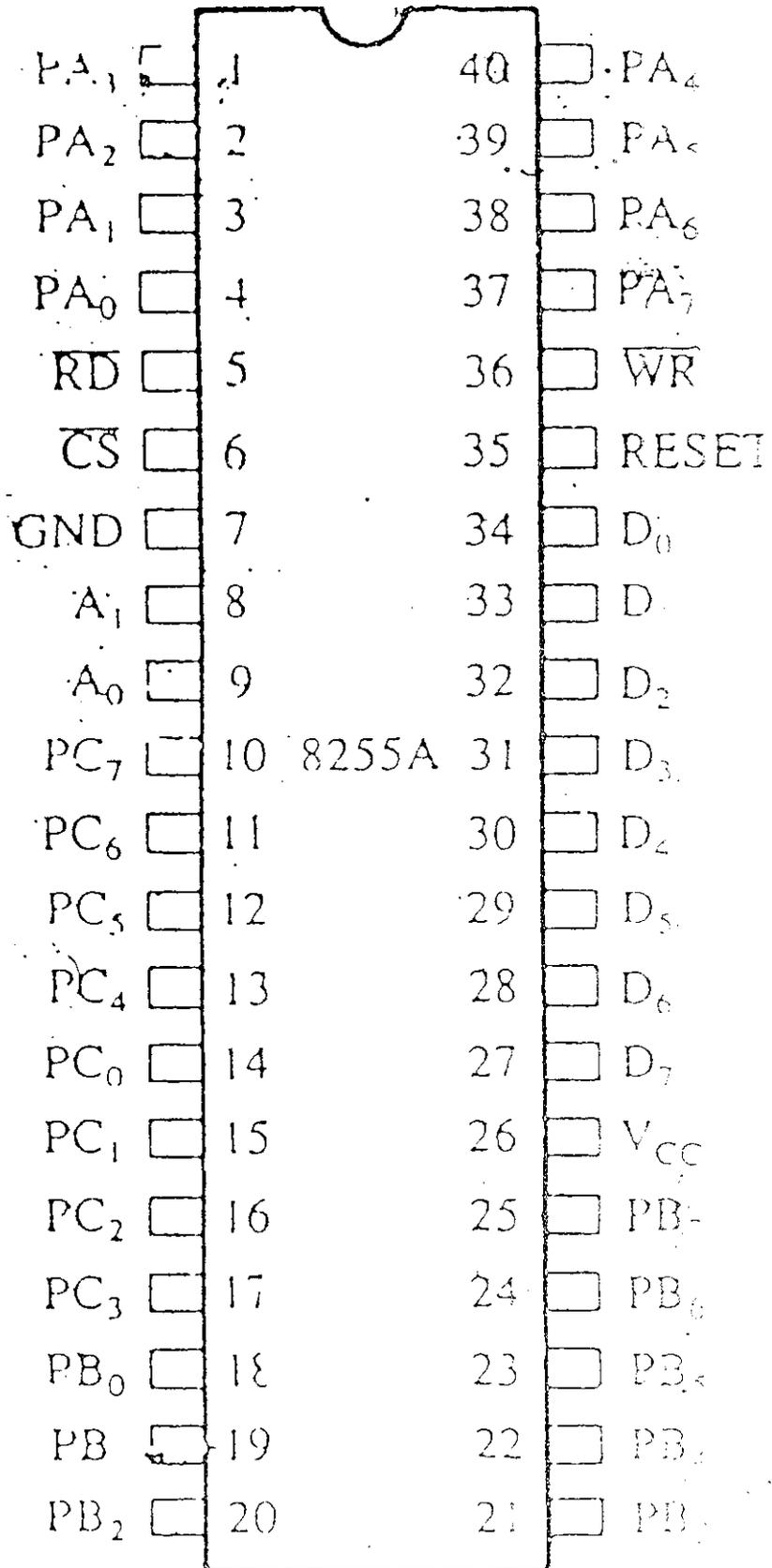
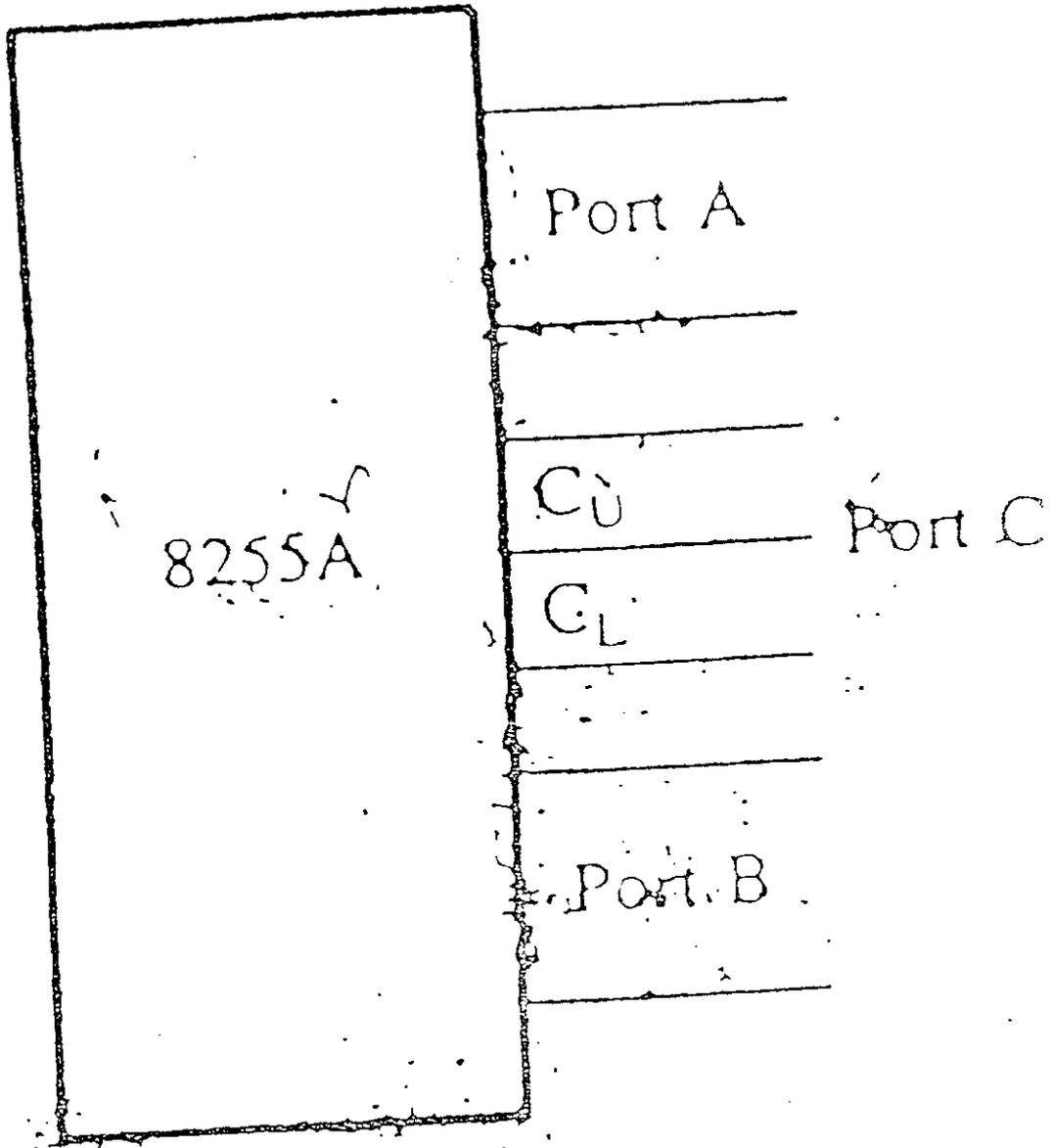


Fig. 3.1. 8255A PIN CONFIGURATION



(a)

Fig. 3.2. 8255A PORTS.

Control Word

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀

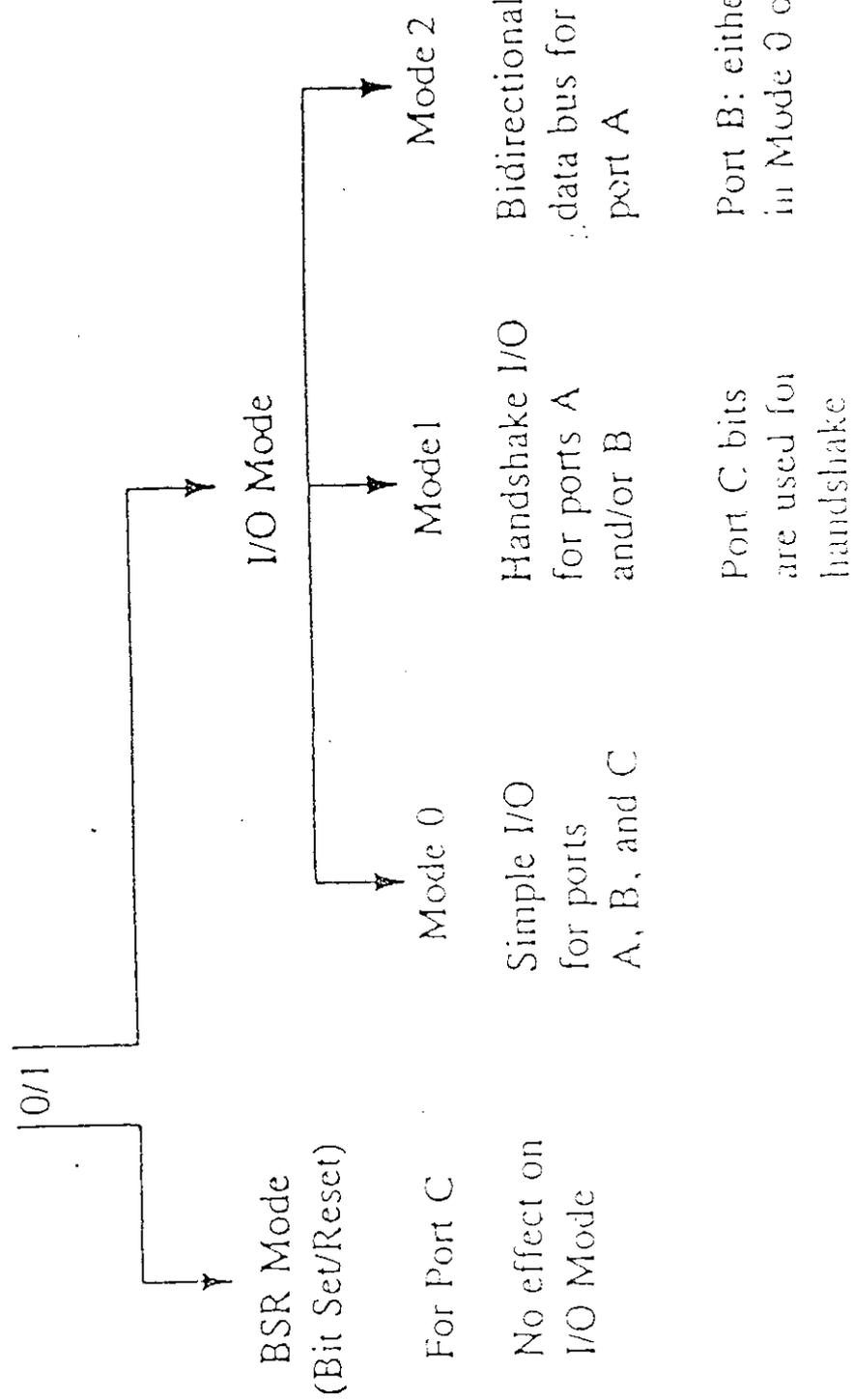


Fig. 3.5. 6855-A modes

(b)

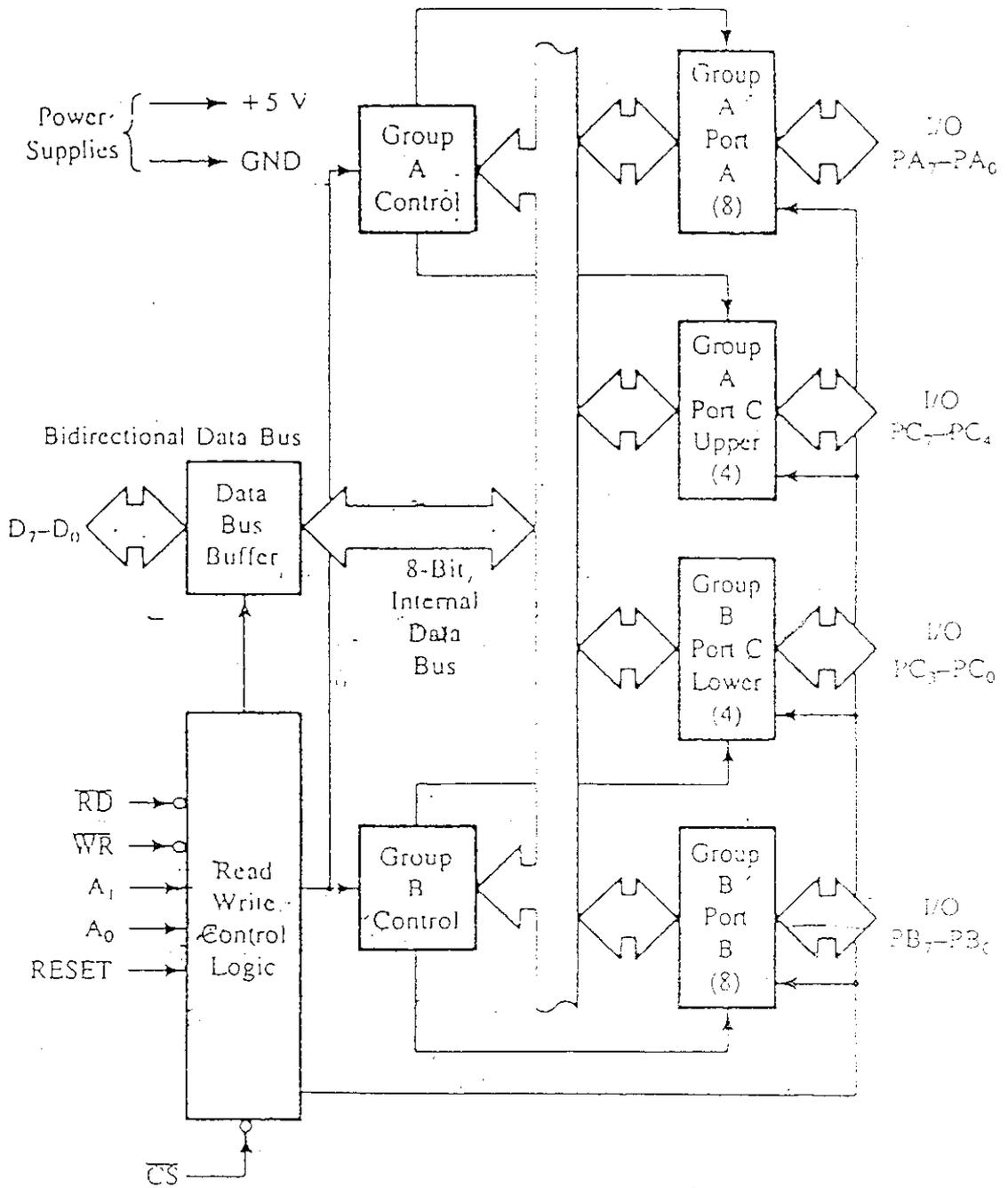


Fig. 3.4. 8255 A BLOCK DIAGRAM.

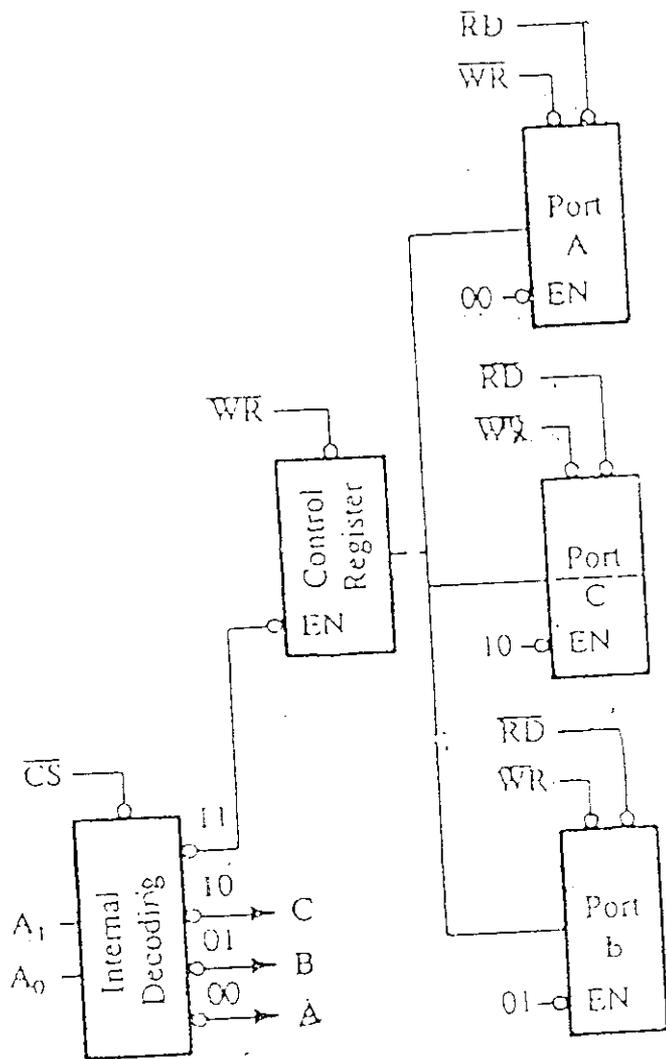


Fig. 3.4 . CONTROL LOGIC AND I/O PORTS.

(b)

Pin Names

D_7-D_0	Data Bus (Bidirectional)
RESET	Reset Input
\overline{CS}	Chip Select
\overline{RD}	Read Input
\overline{WR}	Write Input
A_0, A_1	Port Address
PA_7-PA_0	Port A (Bit)
PB_7-PB_0	Port B (Bit)
PC_7-PC_0	Port C (Bit)
V_{CC}	+5 Volts
GND	0 Volts

Fig. 3.4. PIN NAMES.

CHAPTER IV

8279 PROGRAMMABLE DISPLAY INTERFACE

The 8279 is a hardware for interfacing a matrix keyboard and a multiplexed display. Software approach for the above purpose is also possible but major disadvantage is that microprocessor is occupied for a considerable amount of time in checking the keyboard and refreshing the display. The 8279 relieves the processor from these two tasks.

The 8279 is 40p pin device with two major segments viz keyboard and display. The keyboard segment can be connected to a 64 contact key matrix. Keyboard entries are debounced and stored in the internal FIFO (first in first out) memory while an interrupt signal is generated with each entry. The display segment can provide a 16 character scanned display interface with such devices as LEDs. This segment has 16 x 8 R/W memory (RAM). which can be used to read / write information for display purposes. The display can be setup either in right entry or left entry format.

The block diagram of 8279 shows four sections. These sections are.

4.1 SCAN SECTION

The scan section has a scan counter and 4 scan lines (SLO - SL3). These four scan lines can be decoded using a 4 to 16 decoder to generate 16 lines for scanning. These lines can be drives of a multiplexed display.

4.2 DISPLAY SECTION

The display section has 8 output lines divided into two groups, A0 to A3 and B0 to B3. These lines can be used either as a group of 8 lines or as two groups of four in conjunction with the scan lines for a multiplexed display. The display can be blanked by using the BD line. This section includes 16 x 8 display RAM. The MPU can read from or write into any of these registers.

4.3 MPU INTERFACE SECTION

This section includes 8 bi-directional data lines (DB0-DB7), one interrupt request line (IRQ) and 6 lines for interfacing, including the buffer address line (A0). When A0 is high, the signals are interpreted as control words are status. When A0 is low, the signals are interpreted as data. The IRQ line goes high whenever data entries are stored in the FIFO. This signal is used to interrupt the MPU to indicate the availability of data.

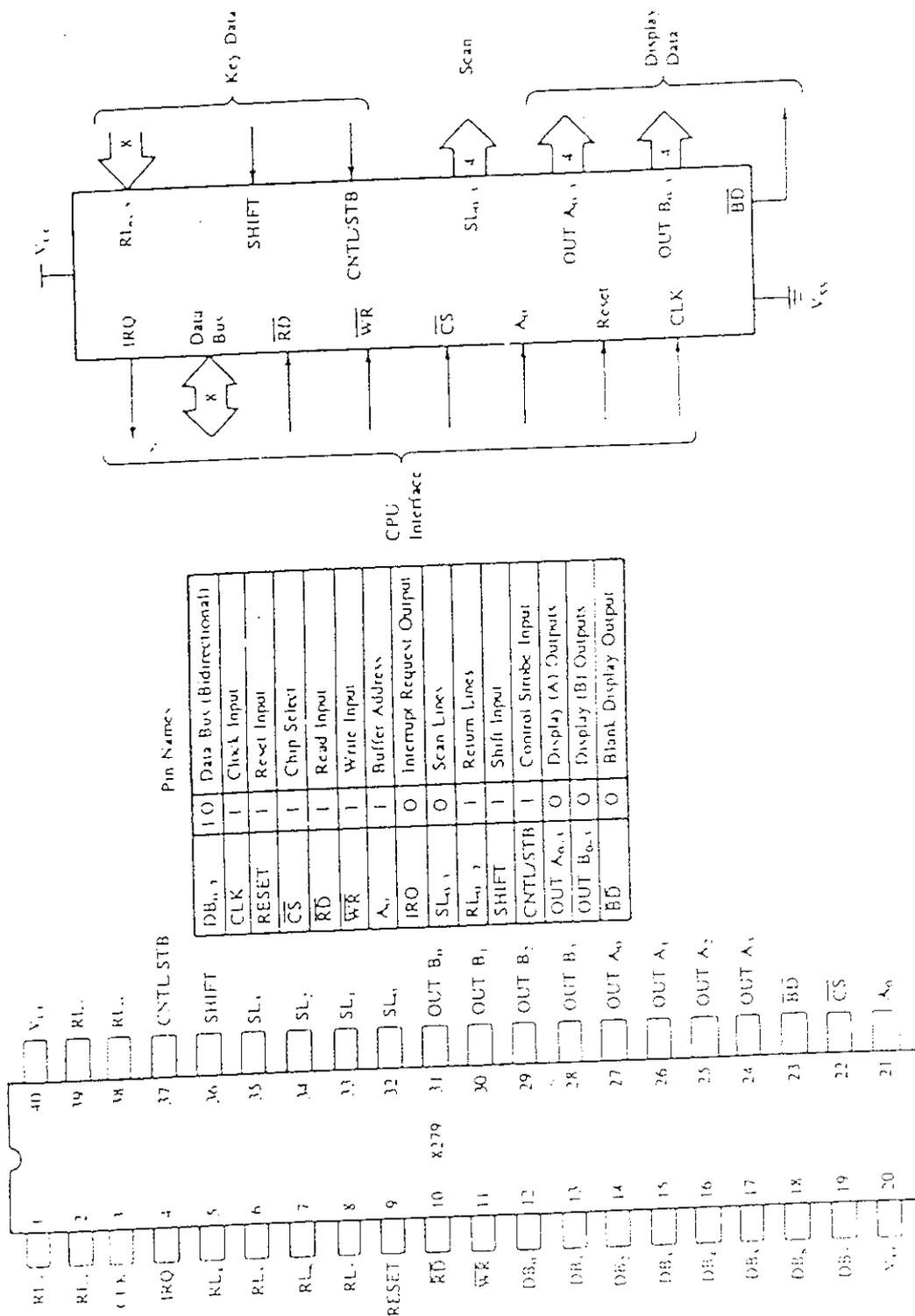


Fig. 4.1. 8279 PIN CONFIGURATION.

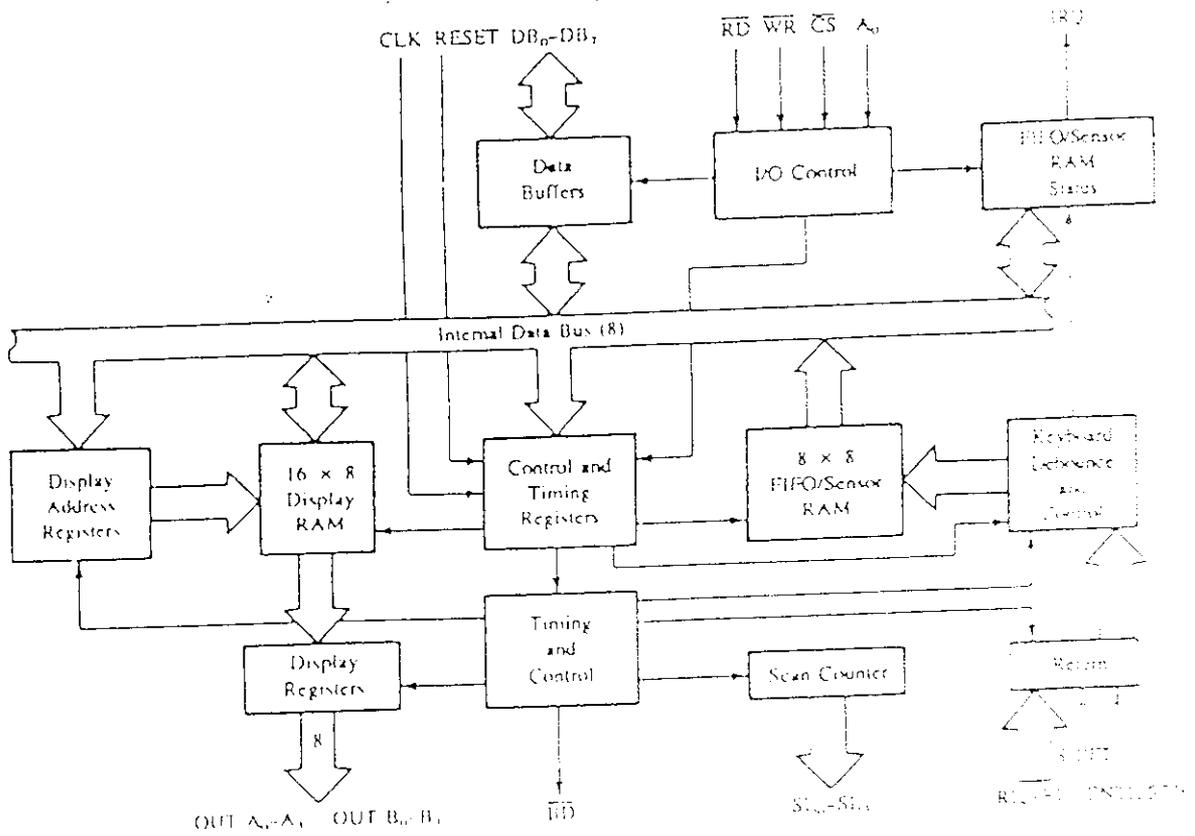


Fig. 4.2. 8279 BLOCK DIAGRAM.

CHAPTER V

HANDSHAKING

The basic principle of the two wire handshaking method of data transfer is as follows.

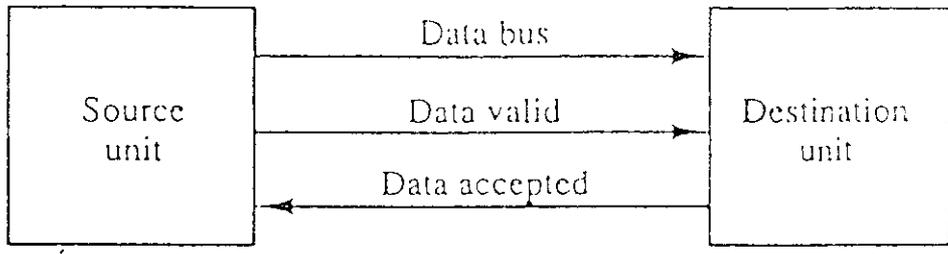
One control line is in same direction as the data flow in the bus from the source to the destination. It is used by the source unit to inform the destination unit whether there are valid data in the bus. The other control line is in the other directions from destination to the source. It is used by the destination unit to inform the source whether it can accept data. The sequence of control during the transfer depends in the unit that initiates the transfer.

There are two types of handshaking one in source initiated transfer and the other is destination initiated transfer.

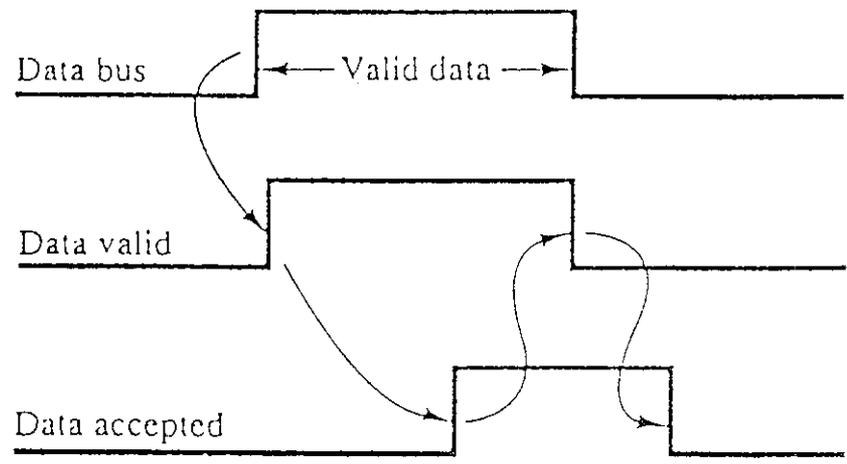
The source initiated transfer using handshaking having two lines called data valid, which is generated by the source unit, and the data accepted generated by destination unit. The timing diagram shows the exchange of signals between the two units. The sequence of events listed in the fig 5.1 shows the four possible states that the system can be at any given time. The source unit initiates the transfer by placing the data on the bus and enabling its data valid signal. The data accepted signal is activated by the destination unit after it accepts the data from other bus. The source unit then disables its data valid signal, which invalidates the data on the bus. The destination unit then disables its data accepted signal and the system goes into its initial state.

The source does not send the next data item until after the destination unit shows it's readiness to accept new data by disabling its data accepted signal. This scheme allows arbitrary delay from one state to the next and permits each unit to respond at its own data transfer rate.

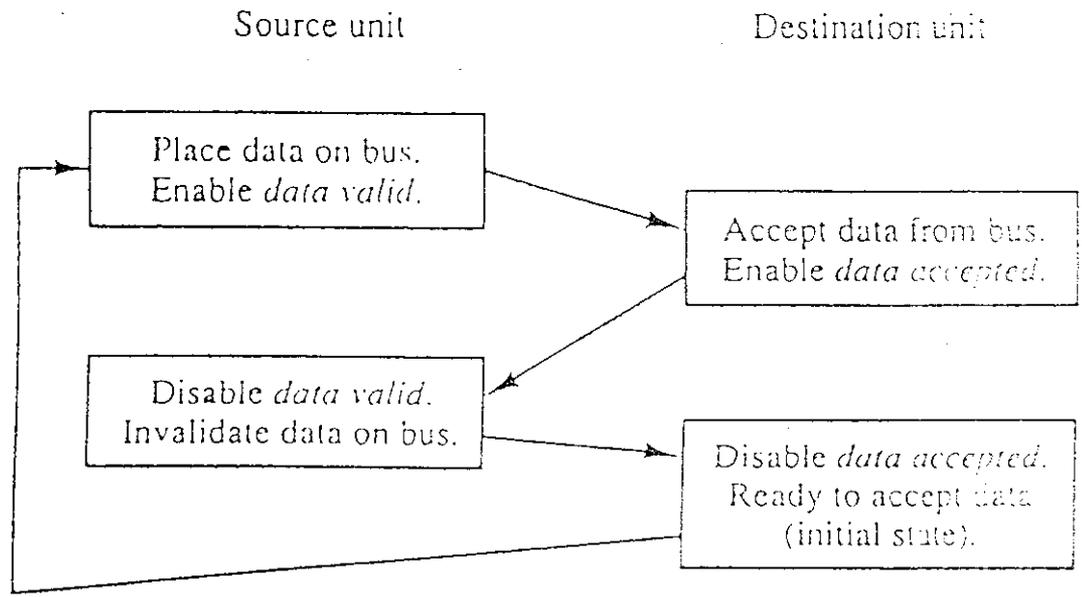
The another method, destination initiated transfer using handshaking lines is shown in fig 5.2. Note that the name of the signals generated by the destination unit has been changed to ready for data to reflect its new meaning. The source unit in this case does not place data on the bus until after it receives the ready for data signal from the destination unit. From there on, the handshaking procedure follows the same pattern as in the source initiated case. The sequence of events in both cases would be identical if we consider the ready for data signal as the complement of data accepted. In fact the only difference between the source initiated and destination initiated transfer is in the choice of initial state. This advantage had been made use of in this project to select the input to the printer.



(a) Block diagram

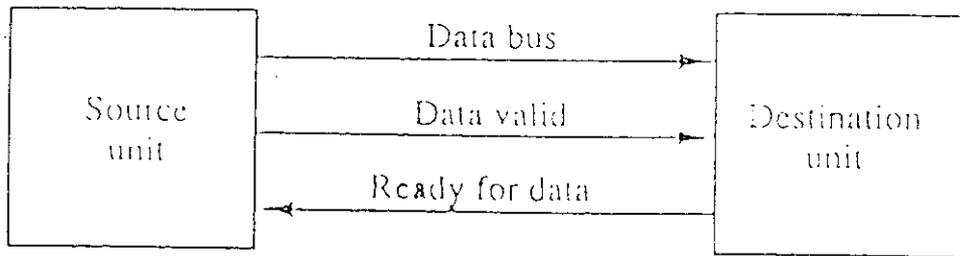


(b) Timing diagram

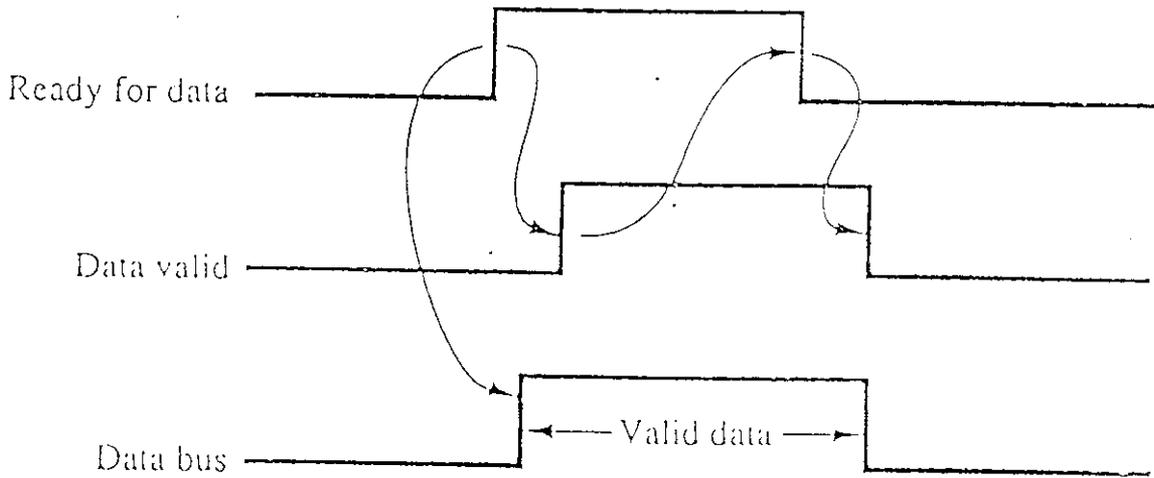


(c) Sequence of events

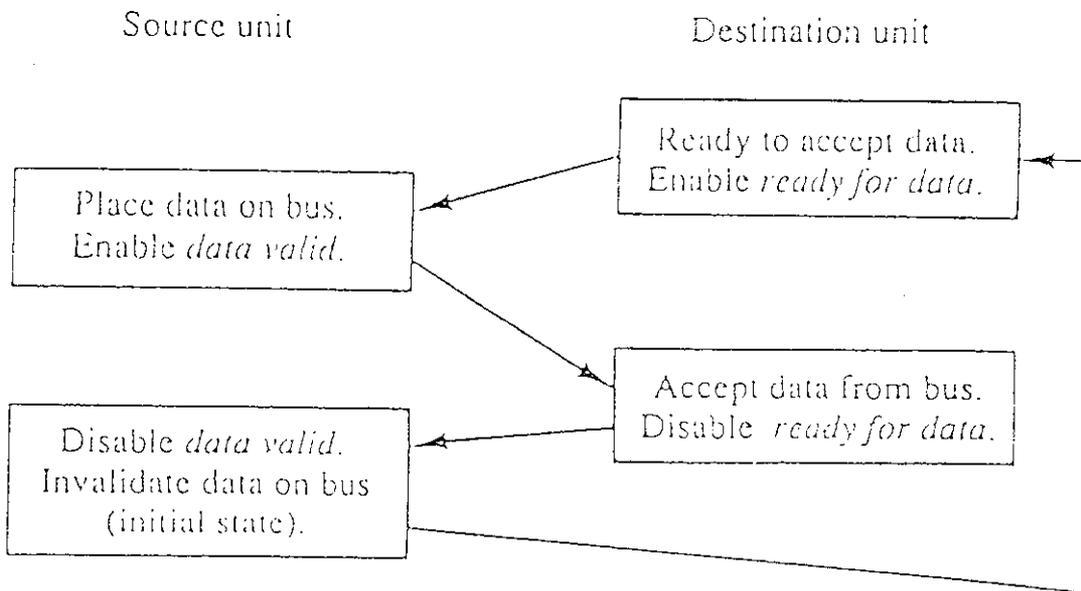
Fig. 5.1. SOURCE INITIATED TRANSFER.



(a) Block diagram



(b) Timing diagram



(c) Sequence of events

Fig. 5.2. DESTINATION INITIATED TRANSFER.

CHAPTER VI

PARALLEL INTERFACE

Micro computers transmit data to printers circuitry inside the printer must be capable of determining when information is being transferred. Sometimes the printer has to acknowledge that data was transferred successfully. The printer circuitry that listens to data coming from the computer comes in five different varieties. Two of the five are common: parallel and serial. Video and HPiB are significant minorities. The last SCSI is a future possibility

With a parallel interface, each bit that represents a character or instruction has its own wire. For eg, the bit pattern representing a A is 01000001. To transmit the letter A at once, at least eight wires are needed : one wire for each of the 8 bits representing the letter A as shown in Fig 6.1.

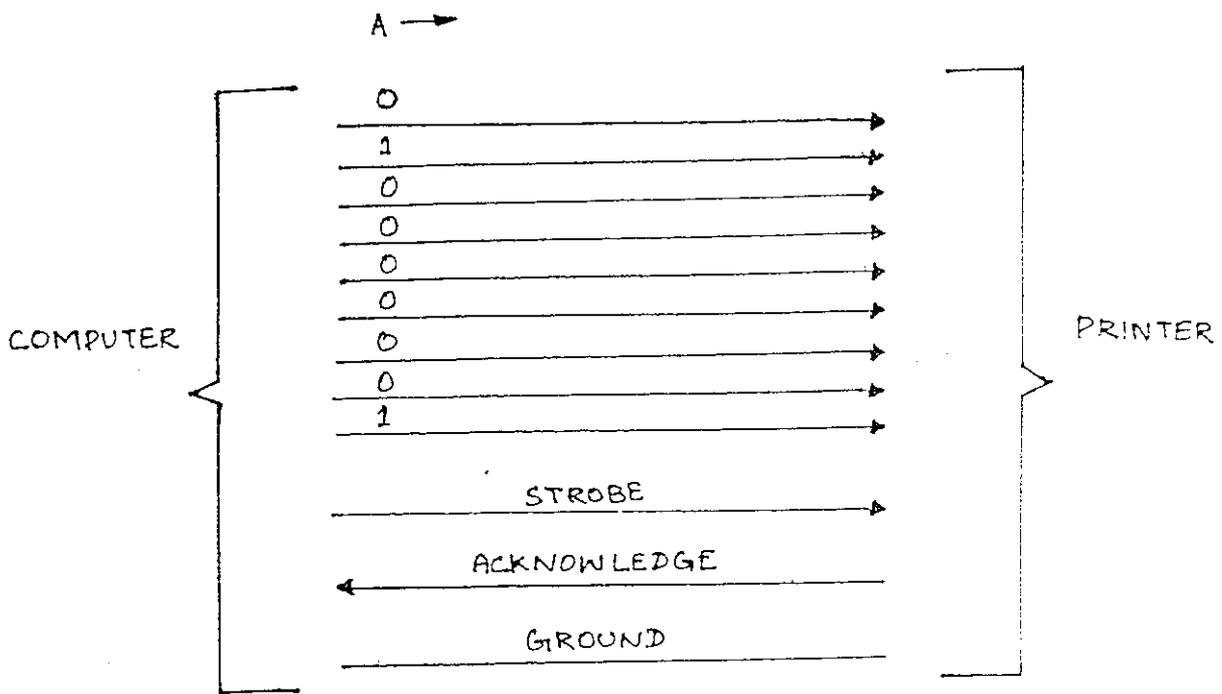


Fig. B.1. A SIMPLE PARALLEL CABLE

Centronics Parallel Port

<i>Computer Pin</i>			<i>Printer Pin</i>	
1	strobe	----->	1	tells printer to latch onto data
2	data bit 0	----->	2	data
3	data bit 1	----->	3	data
4	data bit 2	----->	4	data
5	data bit 3	----->	5	data
6	data bit 4	----->	6	data
7	data bit 5	----->	7	data
8	data bit 6	----->	8	data
9	data bit 7	----->	9	data
10	ack.	<-----	10	printer tells computer received
11	busy	<-----	11	printer tells computer buffer full
12	paper out	<-----	*12	not necessary
13	select	<-----	*13	not necessary
14	auto feed	----->	*14	not necessary
15	error	<-----	*15	not necessary
16	initialize	----->	*16	not necessary
17	select	<-----	*17	not necessary
18-25	ground	-----	18-30	

Parallel Interface Description

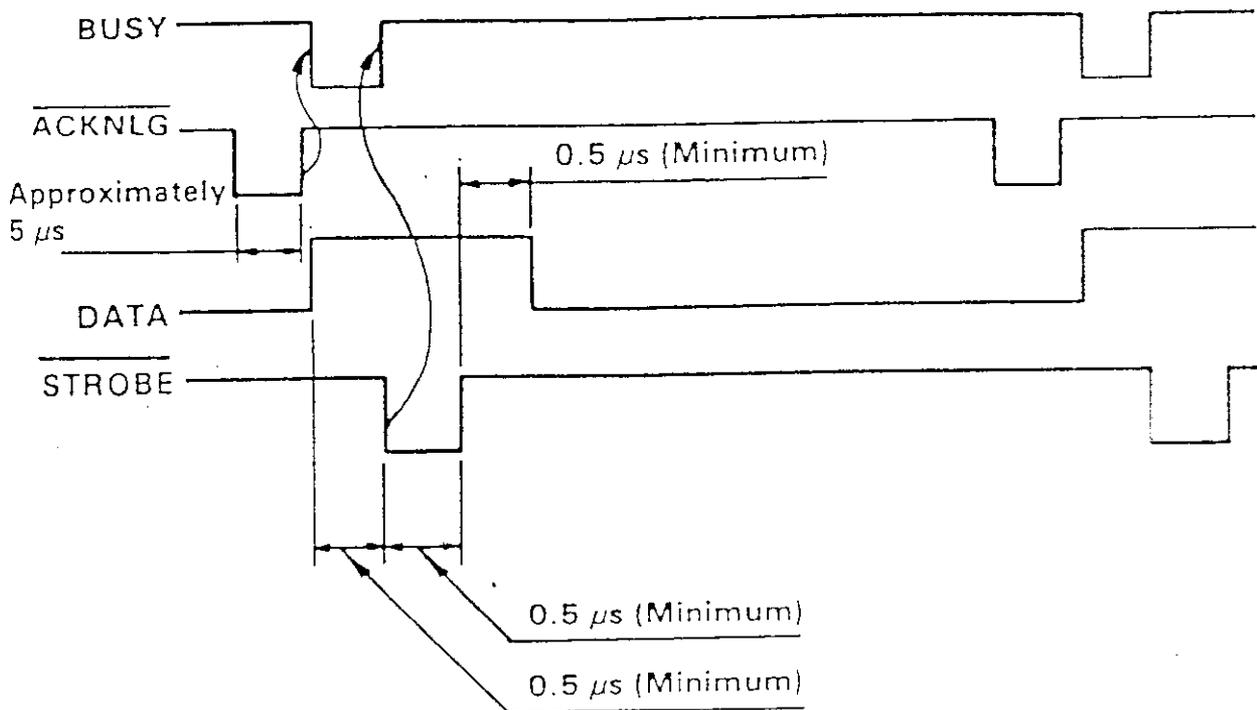
Specifications:

- Data transfer rate: 1000 cps (maximum)
- Synchronization: By externally-supplied STROBE pulses.
- Handshaking $\overline{\text{ACKNLG}}$ or BUSY signals.
- Logic level: Input data and all interface control signals are compatible with the TTL level.

Connector: Plug: 57-30360 (Amphenol)

Connector pin assignment and descriptions of respective interface signals are provided on the following pages.

Data transfer sequence:



Parallel Interface Timing Diagram

Signal Pin No.	Return Pin. No.	Signal	Direction	Description
1	19	$\overline{\text{STROBE}}$	In	STROBE pulse to read data in. Pulse width must be more than 0.5 μs at receiving terminal. The signal level is normally "high"; read-in of data is performed at the "low" level of this signal.
2	20	DATA 1	In	These signals represent information of the 1st to 8th bits of parallel data respectively. Each signal is at "high" level when data is logical "1" and "low" when logical "0."
3	21	DATA 2	In	
4	22	DATA 3	In	
5	23	DATA 4	In	
6	24	DATA 5	In	
7	25	DATA 6	In	
8	26	DATA 7	In	
9	27	DATA 8	In	
10	28	$\overline{\text{ACKNLG}}$	Out	Approximately 5 μs pulse: "low" indicates that data has been received and the printer is ready to accept other data.
11	29	BUSY	Out	A "high" signal indicates that the printer cannot receive data. The signal becomes "high" in the following cases: 1. During data entry. 2. During printing operation. 3. In "offline" state. 4. During printer error status.

Connector Pin Assignment and Descriptions of Interface Signals
(Part 1 of 3)

Signal Pin No.	Return Pin No.	Signal	Direction	Description
12	30	PE	Out	A "high" signal indicates that the printer is out of paper.
13	—	SLCT	Out	This signal indicates that the printer is in the selected state.
14	—	$\overline{\text{AUTO FEED XT}}$	In	With this signal being at "low" level, the paper is automatically fed one line after printing. (The signal level can be fixed to "low" with DIP SW pin 2-3 provided on the control circuit board.)
15	—	NC		Not used.
16	—	0V		Logic GND level.
17	—	CHASSIS-GND	—	Printer chassis GND. In the printer, the chassis GND and the logic GND are isolated from each other.
18	—	NC	—	Not used.
19-30	—	GND	—	"Twisted-Pair Return" signal; GND level.
31	—	$\overline{\text{INIT}}$	In	When the level of this signal becomes "low" the printer controller is reset to its initial state and the print buffer is cleared. This signal is normally at "high" level, and its pulse width must be more than 50 μs at the receiving terminal.

Connector Pin Assignment and Descriptions of Interface Signals
(Part 2 of 3)

Signal Pin No.	Return Pin No.	Signal	Direction	Description
32		$\overline{\text{ERROR}}$	Out	The level of this signal becomes "low" when the printer is in "Paper End" state, "Offline" state and "Error" state.
33	—	GND	—	Same as with pin numbers 19 to 30.
34	—	NC	—	Not used.
35				Pulled up to +5 Vdc through 4.7 k-ohms resistance.
36	—	$\overline{\text{SLCT IN}}$	In	Data entry to the printer is possible only when the level of this signal is "low". (Internal fixing can be carried out with DIP SW 1-8. The condition at the time of shipment is set "low" for this signal.)

- Notes:**
- "Direction" refers to the direction of signal flow as viewed from the printer.
 - "Return" denotes "Twisted-Pair Return" and is to be connected at signal-ground level.
When wiring the interface, be sure to use a twisted-pair cable for each signal and never fail to complete connection on the return side. To prevent noise effectively, these cables should be shielded and connected to the chassis of the system unit and printer, respectively.
 - All interface conditions are based on TTL level. Both the rise and fall times of each signal must be less than 0.2 μs .
 - Data transfer must not be carried out by ignoring the $\overline{\text{ACKNLG}}$ or BUSY signal. (Data transfer to this printer can be carried out only after confirming the $\overline{\text{ACKNLG}}$ signal or when the level of the BUSY signal is "low.")

Connector Pin Assignment and Descriptions of Interface Signals (Part 3 of 3)

CHAPTER VII

INTERFACING

The interfacing unit consist of latches 8031 and decoders. The 74LS373 is used for the tratchpatch latch. The PC consist of 8 bit data bus, strobe , Busy signal, acknowl- edge, and paper empty signals. The direction of signal flow is as shown in the Fig. 7.1.

In this project it has been designed such that two PC's share a single printer. One PC consist of two latches, one latch for data transfer and other for strobe, paper empty, busy, and Acknowledge signals. The two corresponding latches of the PC's are in parallel so that the output to the printer in only one.

Two decoders are used in the interface system. One for selecting data latch and other for selecting the signal latch. The signal to the decoder is programmed in the micro- controller and it is controlled by the controller unit.

7.1 INTERFACING UNIT

Here the busy signal is given to all the PC's by the microcontroller. Whenever the strobe signal is received by the latch, the latch through the decoder gets enabled and the data passes to the rotating Buffer through another latch. The decoder will select the PC according to the logic in the microcontroller .

For eg, if the logic from the microcontroller is 11 then the decoder will select the first computer and the data passes to the buffer that is to be printed. If it is 10, then it will select the second computer data to be printed.

The data from the latch is stored in the rotating buffer in the 8031 controller. The buffer is divided into two parts. One for computer 1 and other for computer 2 and so on.

If the logic of the controller is 11 then the data from the computer 1 is stored in buffer 1 and after a fixed

period of time, the busy signal is given to computer 1. Now the logic is changed and it will check whether there is any data from computer 2. If so the data is stored in buffer 2. This process is done for every fixed period of time in sequence simultaneously.

The printer will print the data from only one of the buffers depending upon the control signal from the microcontroller. The printer prints the computer data from one buffer and then only goes to the next buffer. It won't print other buffer's data while printing one buffer's data.

7.2 INTERFACING MEMORY WITH 8031

The ports P0 and P2 given the address of the memory. The port P0 constitute address and data line. The address lines are latched with Arithmetic Logic Enable (ALE) signal. This signal is given to IC 74LS373 as shown in Fig. 7.2. In this kit, jumpers are provided in order to address 6264 (8K RAM) as in Fig. 7.3. and 62256 (32K RAM) as in Fig 7.3. These two RAMs are selected due to the availability of the compon-

ents for addressing. The data lines are pulled up with the help of 10K resistance.

A12-A14 are decoded by IC 74138. The decoded output is used for selecting the chips out of 64K data memory the lower 32K space is allotted for RAM. The upper 32K space is allotted for other peripherals.

7.3 INTERFACING 8255A

Data lines D7 to D0 are given to pins 27 to 34 shown in Fig. 7.5. For selecting the control word and port address A0 and A1 is used. The chip select signal is obtained from the decoded output of 74138 as in Fig 7.6.

The Address for Control word	- 8003
port A	- 8000
port B	- 8001
port C	- 8002

7.4 INTERFACING 8279

8279 is used for displaying the data and for keyboard interfacing. The control and data word address in A0000 and A0001 respectively. Because, A0 of 8031 is connected to A0 of 8279. Address lines A000 to BFFF form a shadow for control and data register.

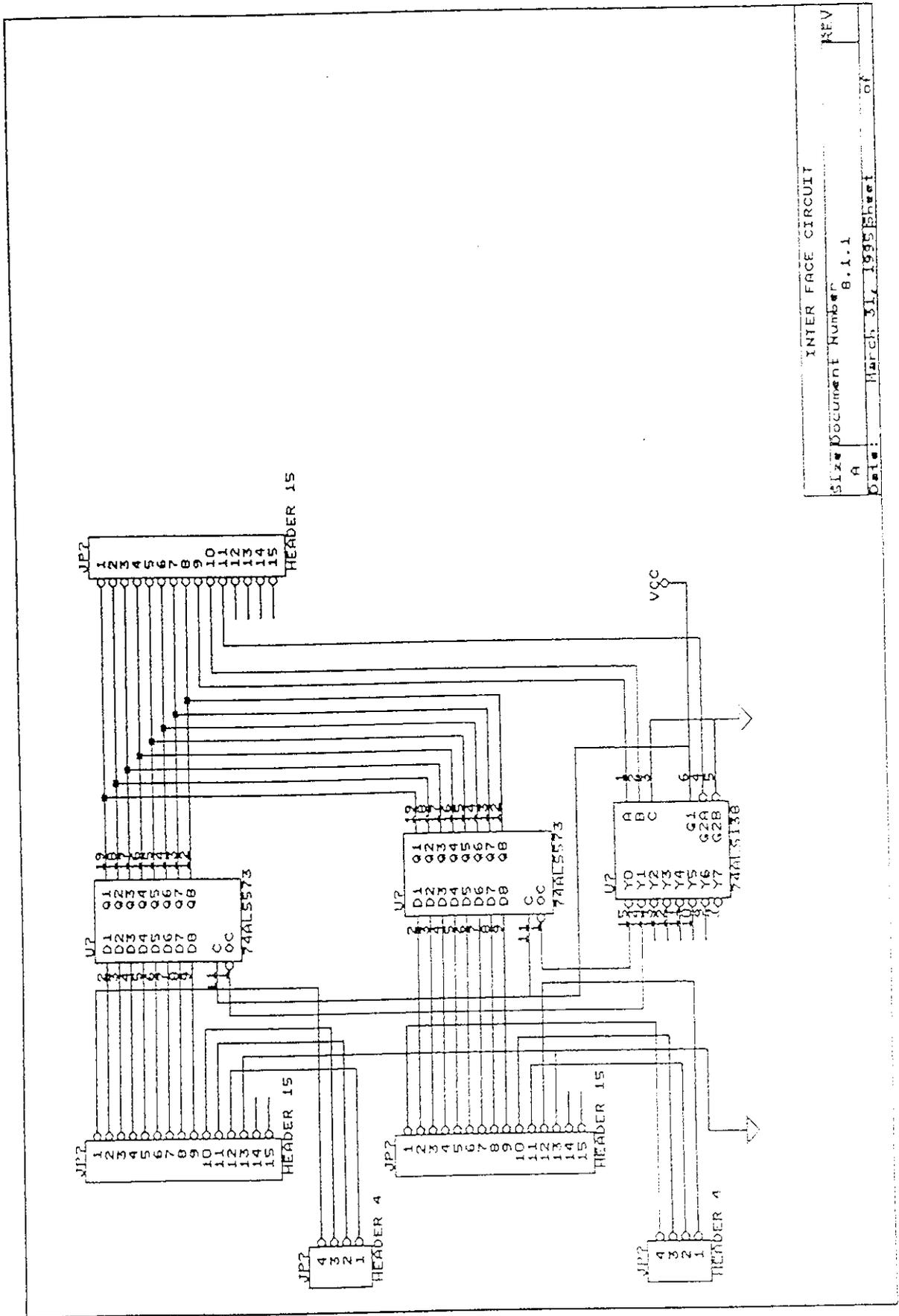
7.5 PROTECTION CIRCUIT

POWER DOWN SIGNAL

When there is a power fluctuation, the voltage applied may go out of operating range of the microcontroller circuit. There is every chance of memory being corroded. For protecting the memory, a power down signal is given to the chip select 2 of 6264 which freeze the chip when the voltage drops below operating voltage. The circuit is shown in Fig. 7.7.

WATCH DOG CIRCUIT

When there is power fluctuation, there is every chance of the microcontroller hanging in same location. In this case the processor should be automatically reseted. It actually consists of a astable mode multivibrator. This will start giving out pulse when there is no input pulse. The input pulse is given by enabling and disabling a port pin. This pulse should of such a duration, that the capacitor should be between $1/3$ and $2/3$ of the voltage. The output from the multivibrator is given to the reset pin as in the Fig. 7.7.



INTER FACE CIRCUIT		REV
Size Document Number	B.1.1	
A		
Date:	March 31, 1995	Sheet
		of

Fig. 7.1 . INTERFACE CIRCUIT .

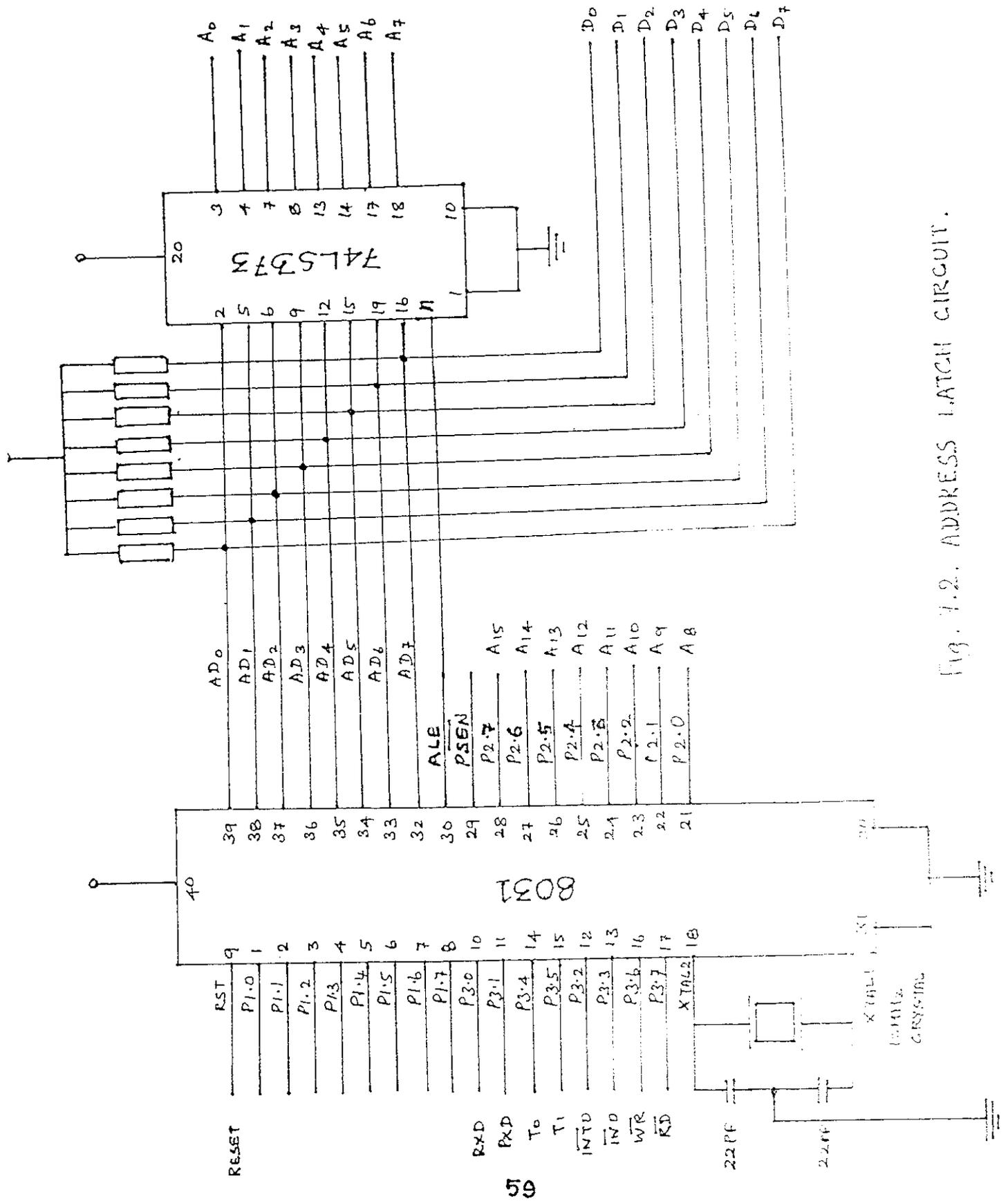


Fig. 7.2. ADDRESS LATCH CIRCUIT.

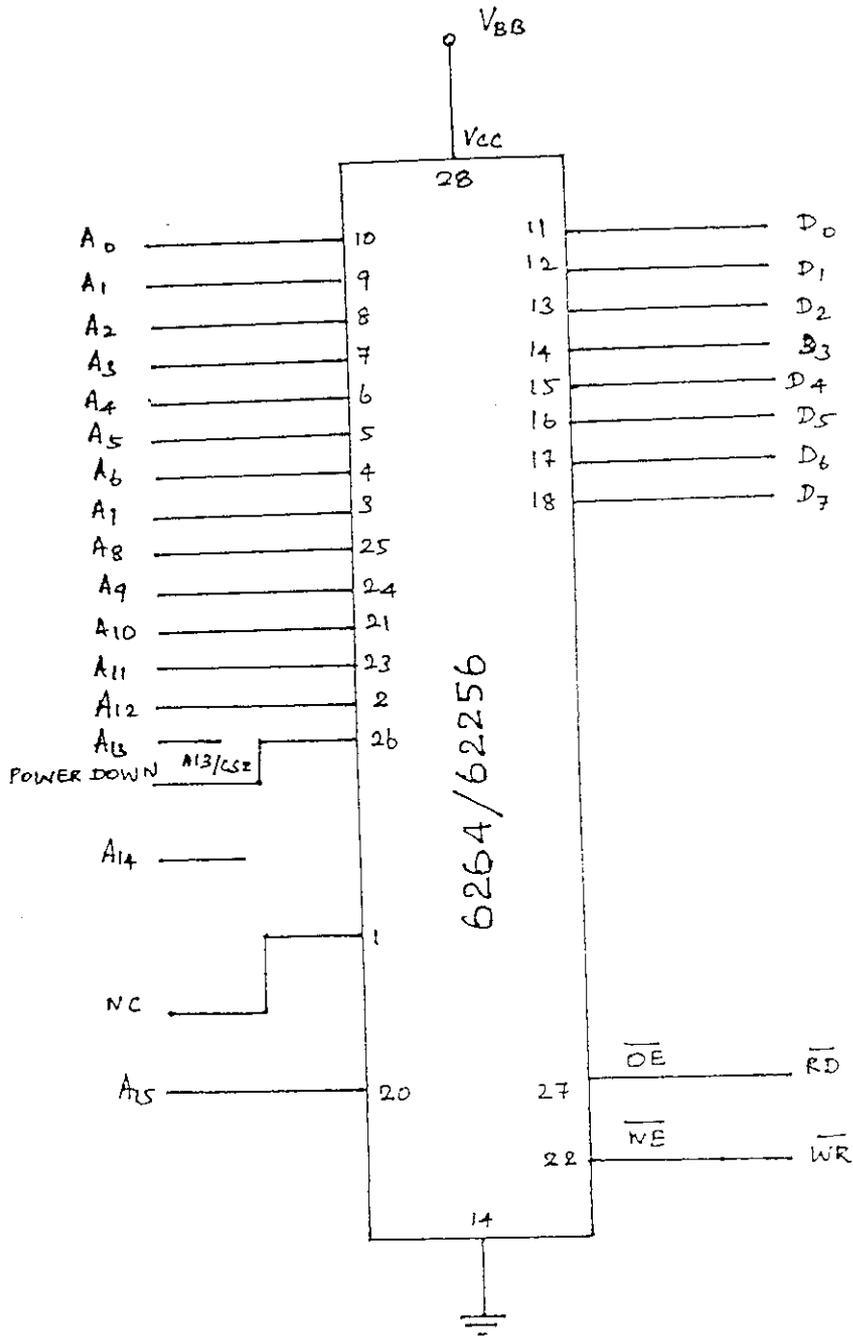


Fig. 7.3. 32 K RAM.

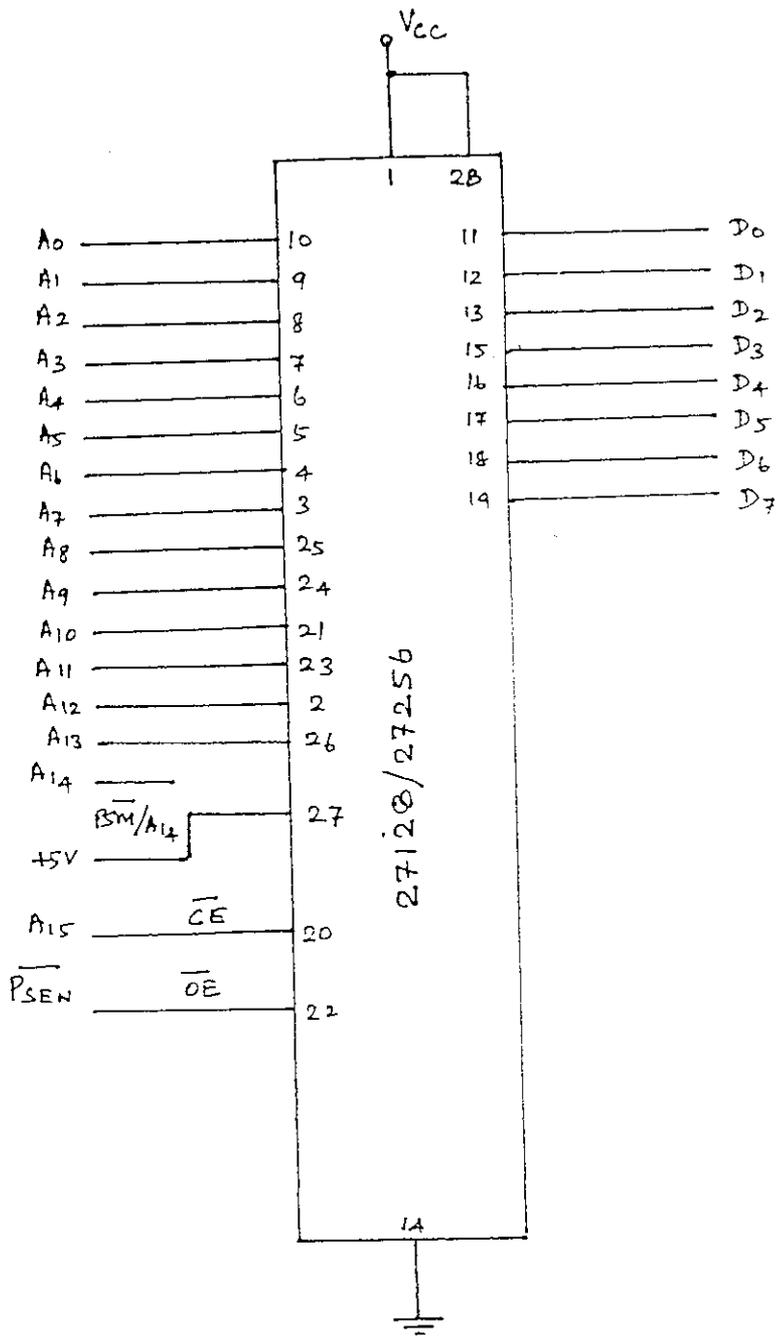


Fig. 7.4. MEMORY INTERFACE.

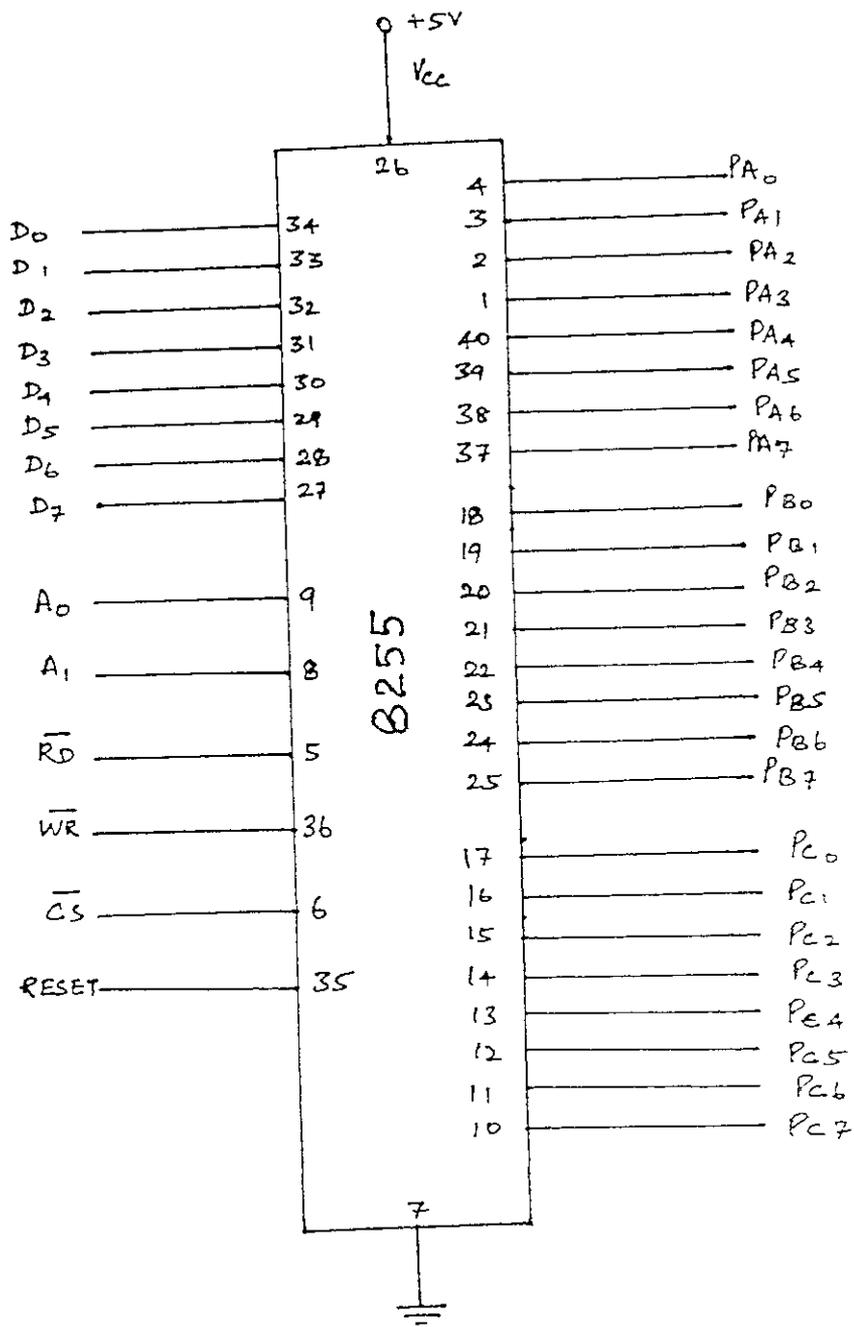


Fig. 7.5. 8255 INTERFACING.

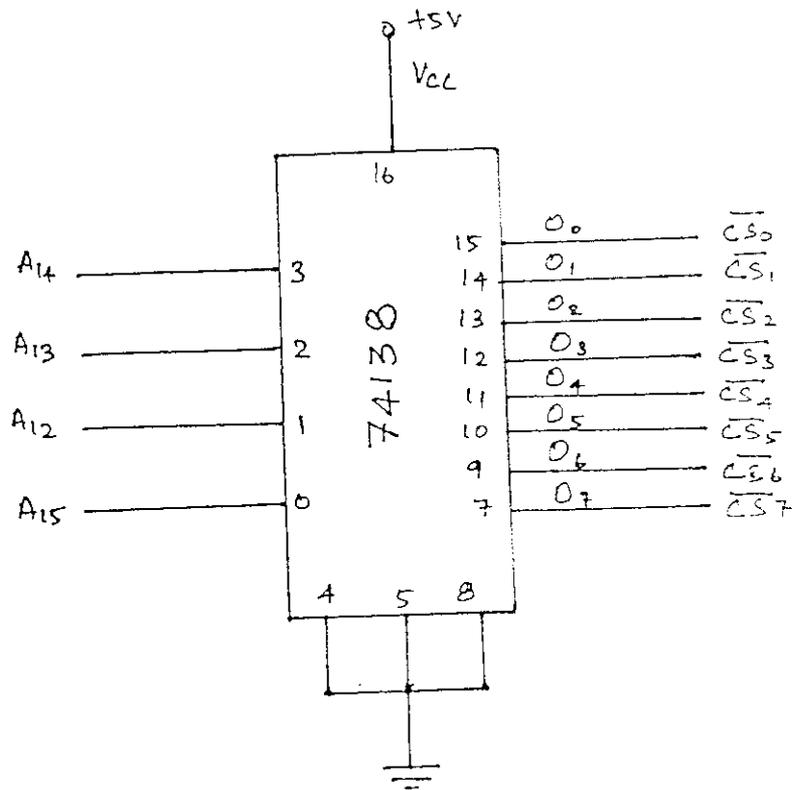
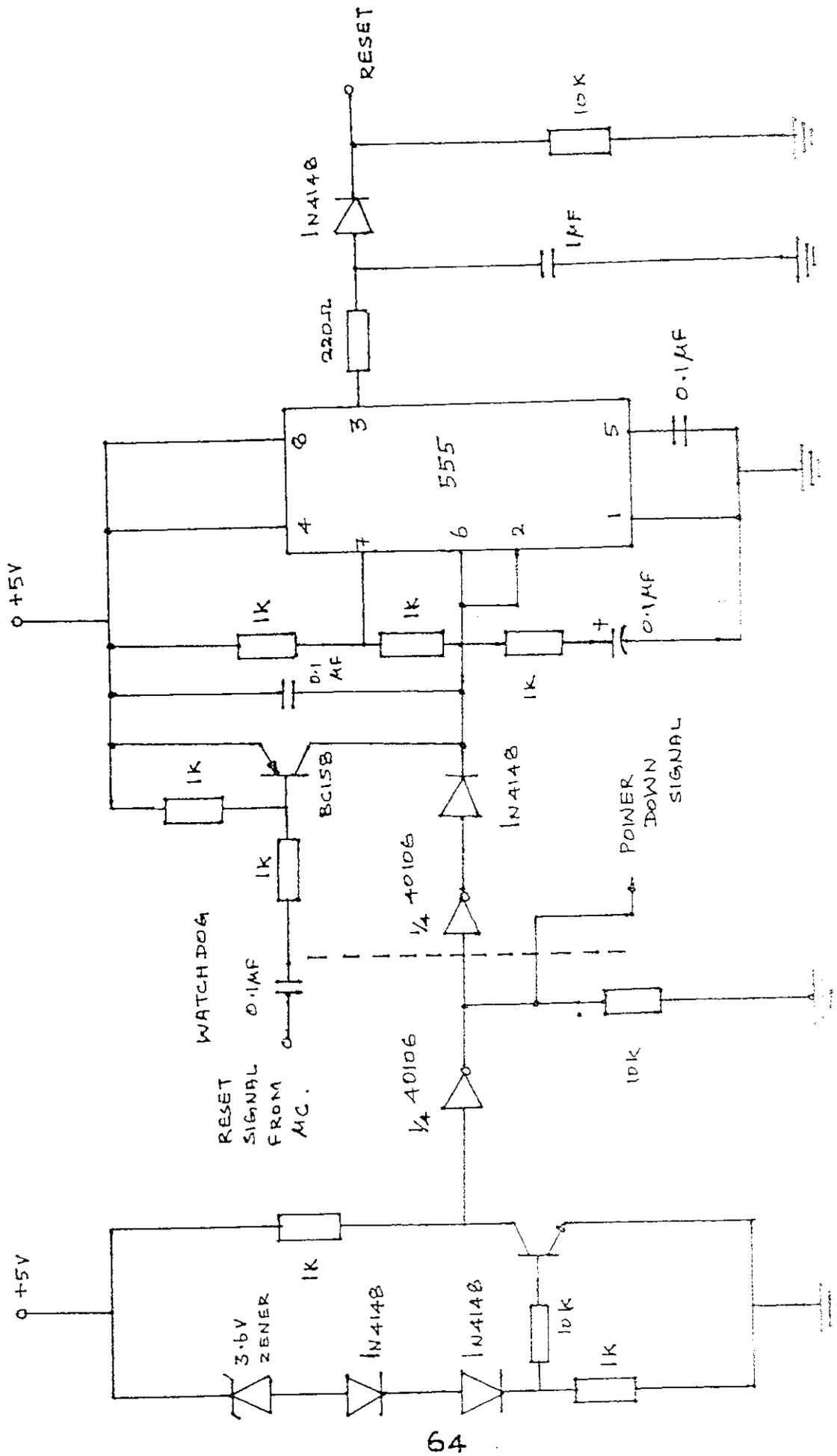


Fig. 7.6. 10 F-8 DECODER/DEMULTIPLEXER.



WATCH DOG CIRCUIT.

Fig. 7.7.

POWER DOWN SIGNAL.

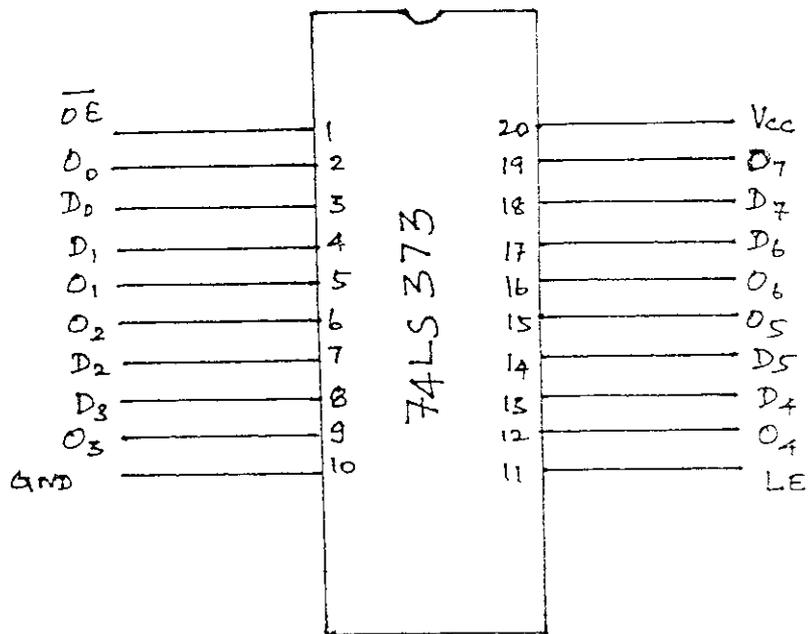
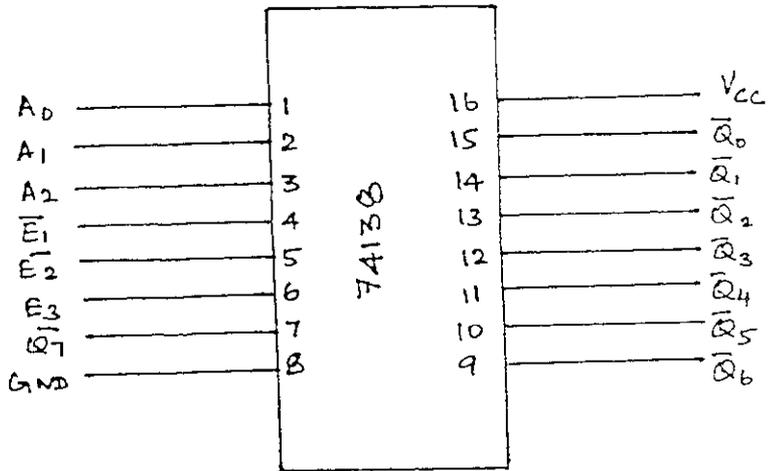


Fig. 7.8. PIN CONFIGURATION.

CHAPTER VIII
PCB FABRICATION

8.1 PCB DRAWING IN SMART WORK

The printed circuit board (PCB) making is the arrangement of components in a neat compact way on a copper board circuit connections.

The various components used in PCB assembly have standard dimensions. Based on this suitable spacing is to be provided while drawing a PCB. First the circuit diagram is thoroughly studied. The size of the different components are noted down. Approximate placing of the different components are taken in preparing a PCB layout. The layout should be a compact, arrange the components neatly, spacing is to be provided sufficiently according to the size of the components so that the leads do not break by bending or the components do not get crowded and all the connecting lines are drawn in the board.

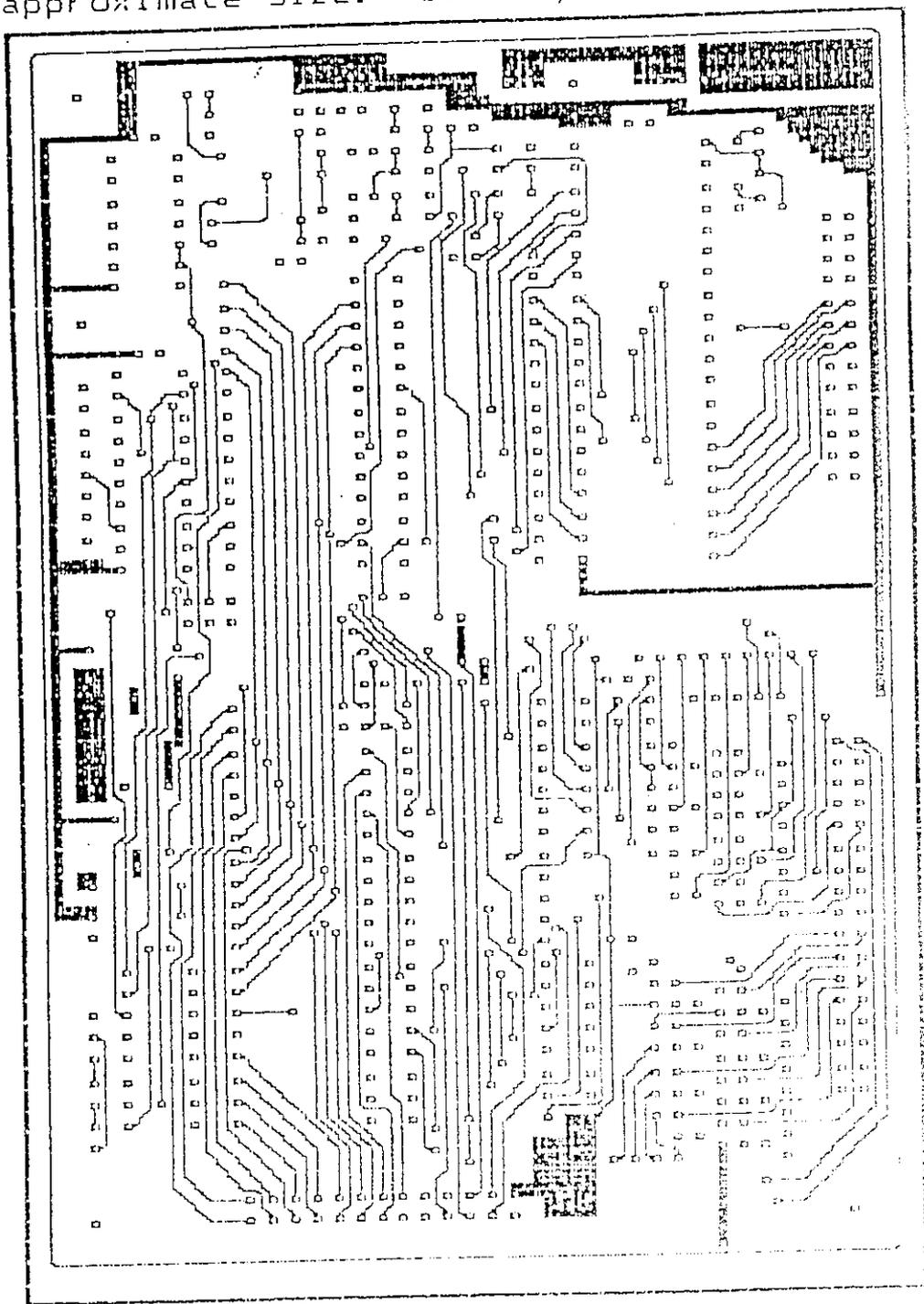
A rough layout for the circuit is prepared on a paper indicating the placing of the components and lines interconnecting them. This diagram is then converted to required PCB layout with the help of SMART WORK which is software design producing a printout of the layout. The SMART WORK commands are SAVE, LOAD, CLEAR, QUIT, DIP and function keys F1, F2, F3, F4, F5 and F6. The solder side and component side layout are shown in the figure.

8.2 FABRICATION

The layout of the PCB drawn in graph sheet is transferred to the copper side of the board. This may be done with help of carbon paper. Then the lines and pads on the board are printed with acid resist and using fine brush. Then the board is put in a solution of Ferric Chloride in water with little quantity of Hcl and the solutions is lightly stirred. Thus process is called etching. After this only copper lines and pads are available for connection. Then the holes are drilled (1mm) for mounting the components.

8.3 TESTING

The complete PCB is tested for the continuity wherever required by multimeter. Shorts between lines are also checked and removed if found. Then assembly of componenets is done by inserting the leads of the components in to the holes and soldered using lead. Then the assembled PCB is tested and checked whether the required output is obtained.

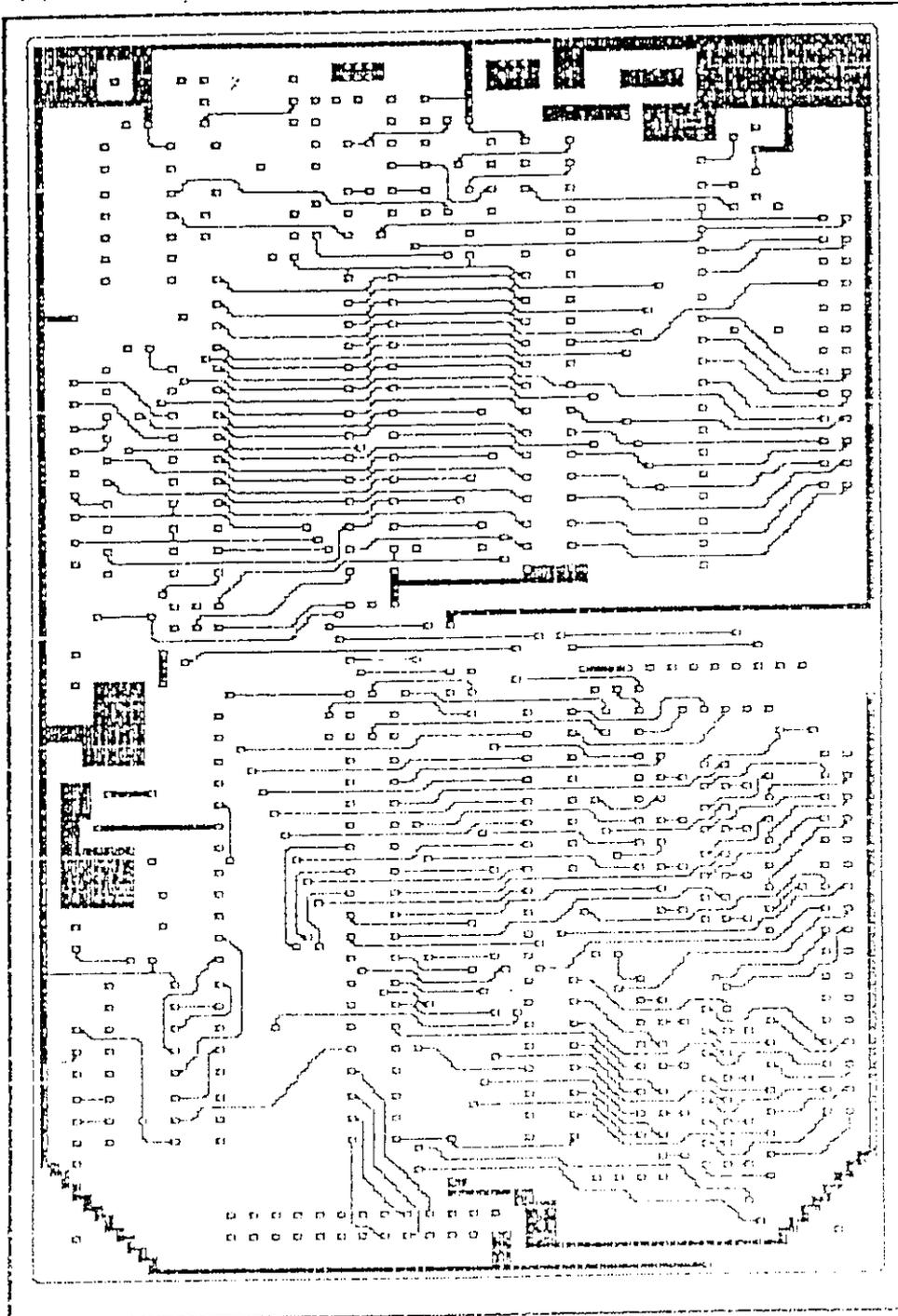


1X checkplot 9 Apr 1994 14:24:14

pdm5.pcb

v1.2 r2 holes: 714 component side

approximate size: 3.90 by 5.65 inches



1X checkplot

9 Apr 1994

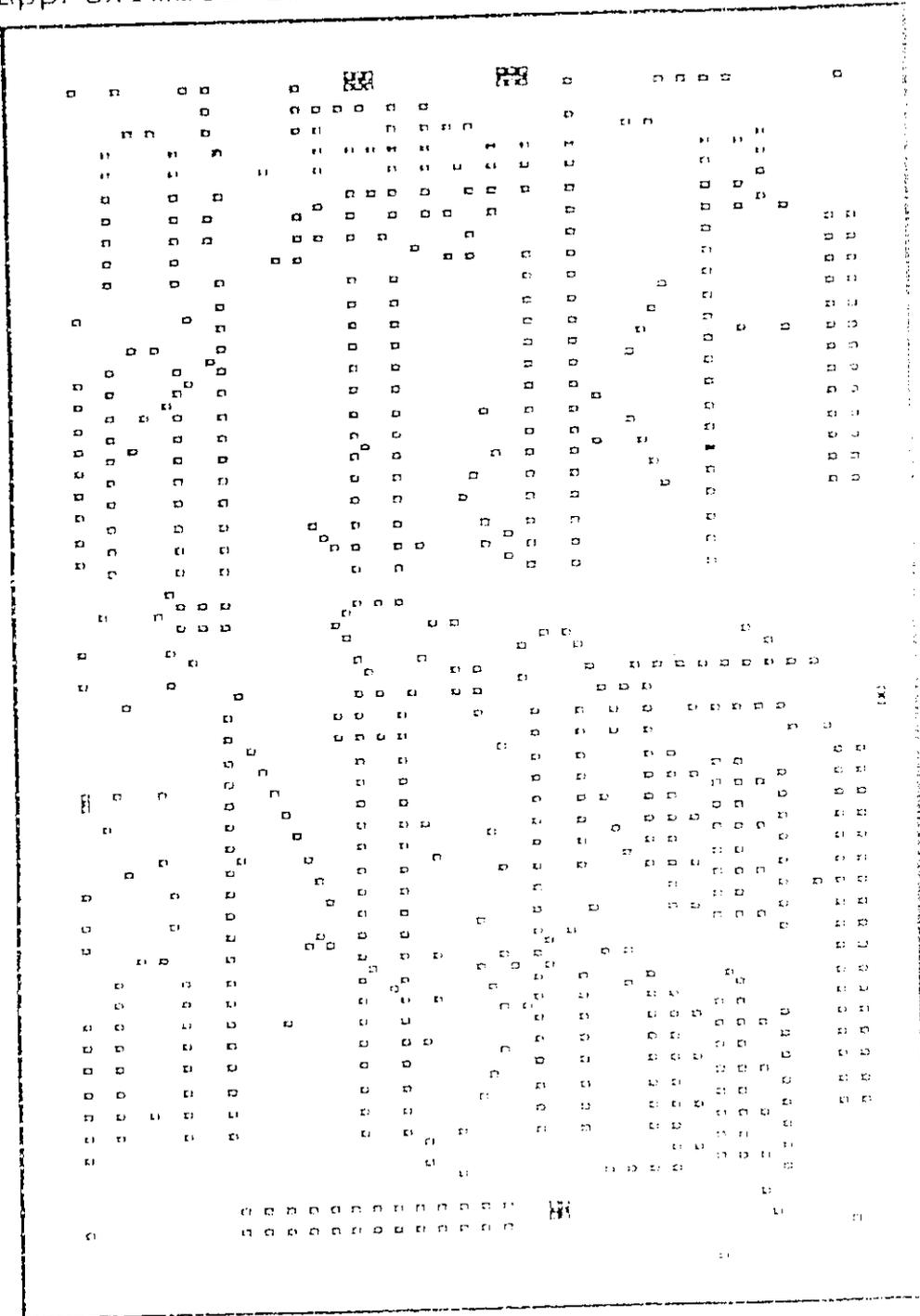
14:29:23

pdm5.pcb

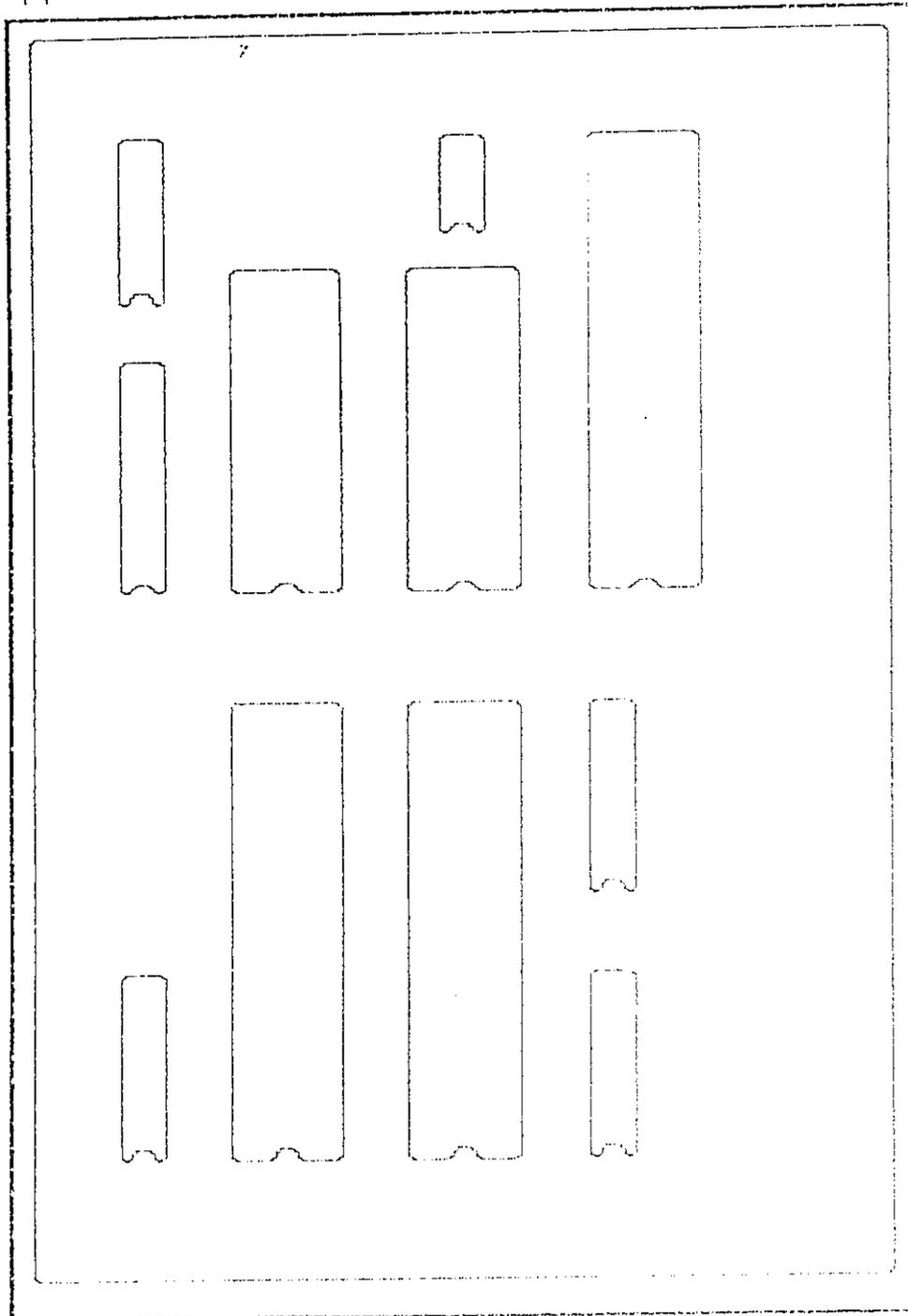
v1.2 r2 holes: 714

pad master

approximate size: 3.90 by 5.65 inches



1X checkplot 9 Apr 1994 14:27:03
pdm5.pcb
v1.2 r2 holes: 714 silkscreen
approximate size: 3.90 by 5.65 inches



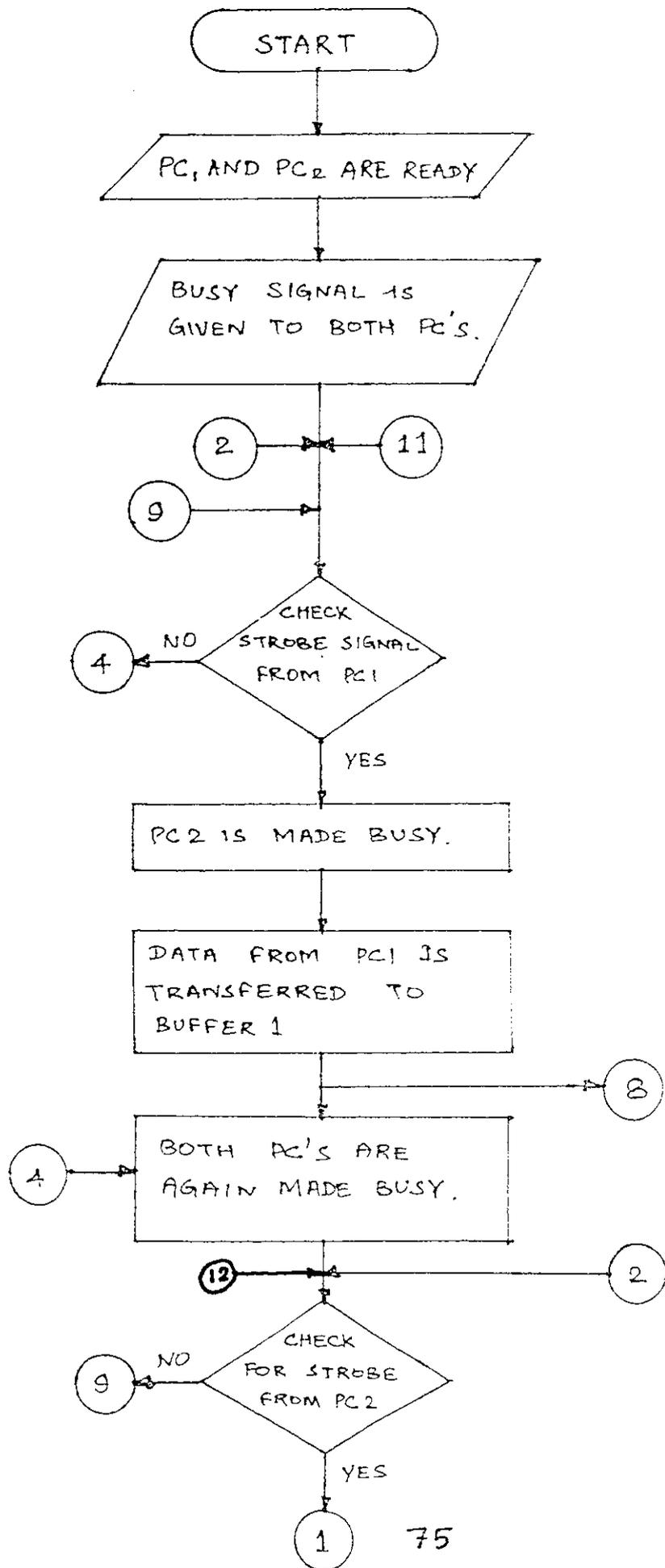
CHAPTER IX

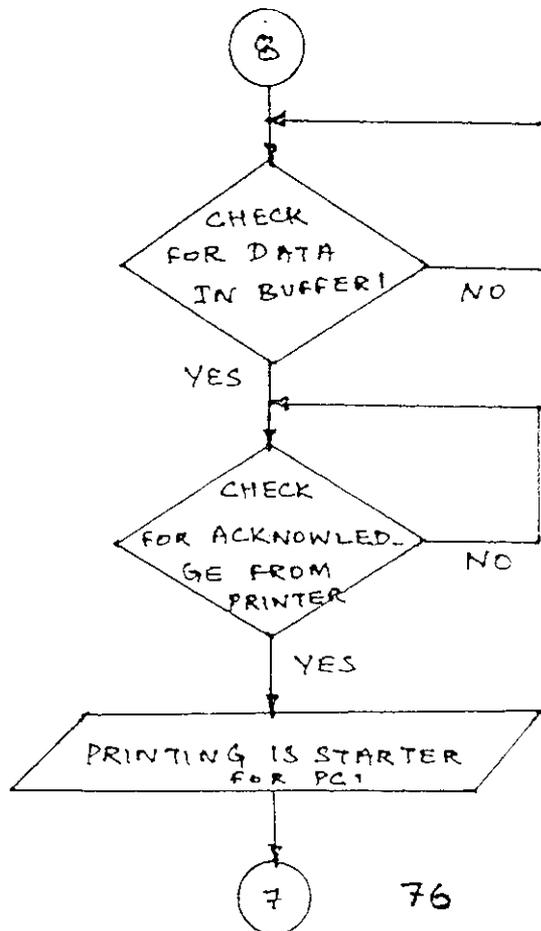
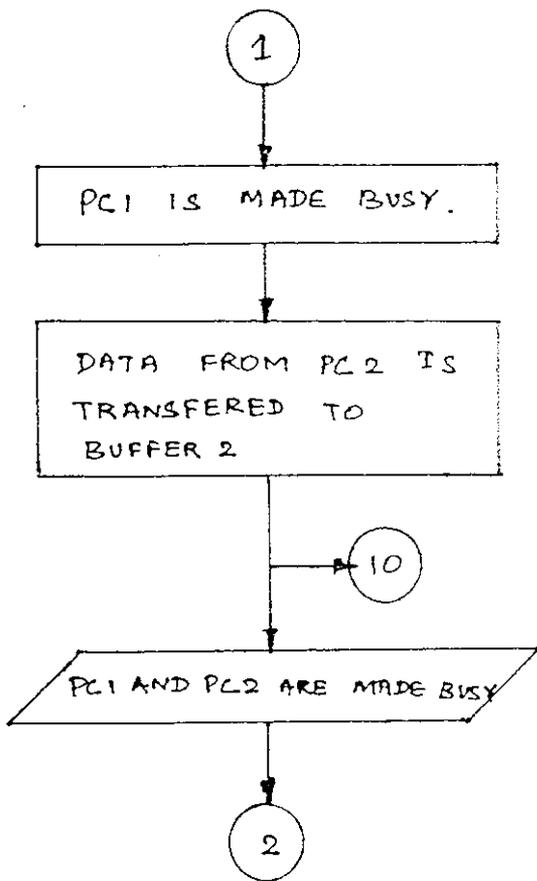
SOFTWARE

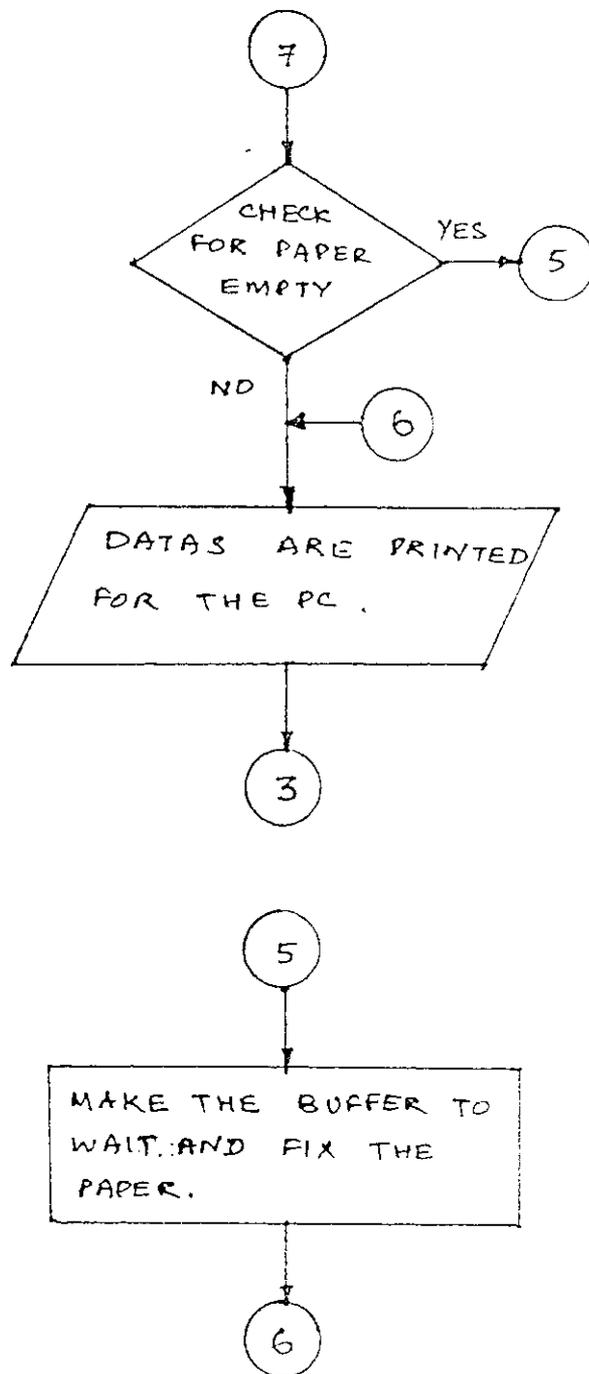
An interactive software has been developed in 'C' language which controls all the operations. A cross compiler has been used which converts the 'C' language into 8031 source code.

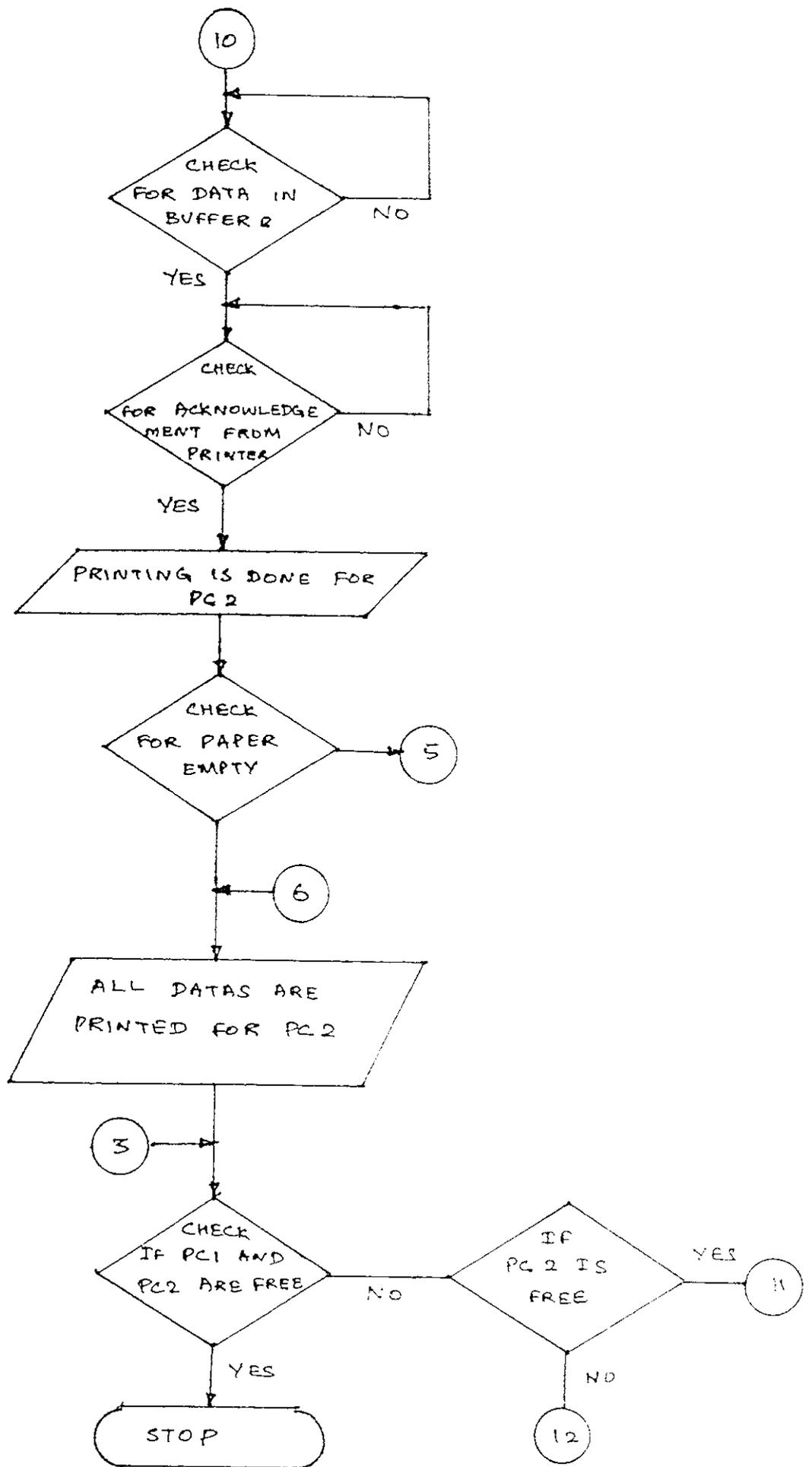
ALGORITHM

1. Start.
2. PC1 and PC2 are made busy.
3. Check for the strobe signal from PC1.
4. If yes, make PC2 busy and transfer the data from PC1 to the buffer 1.
5. After transferring data for fixed period of time, both PC's are again made busy.
6. Repeat the steps from step 3 to 5 for PC2.









```

/*****
/* Program for Printer Sharrer
/* 2 Computers shares one Printer
/* DEVICE LOCKING & TIMEOUT INTRODUCED
/*****
# pragma DEBUG OBJECTTEXTEND

# include <reg51.h>
# include <stdio.h>

unsigned char xdata CTRL_B255 _at_ 0x8003;
unsigned char xdata PORTA_B255 _at_ 0x8000;
unsigned char xdata PORTB_B255 _at_ 0x8001;
unsigned char xdata PORTC_B255 _at_ 0x8002;

unsigned char xdata buffer1[4096] _at_ 0x0000; /* End address 0x0FFF */
unsigned char xdata buffer2[4096] _at_ 0x1000; /* End address 0x1FFF */
unsigned int from_ptr1=0;
unsigned int to_ptr1=0;
unsigned int from_ptr2=0;
unsigned int to_ptr2=0;

bit byte_updated1;
bit byte_updated2;
bit device_locked;

unsigned char code DEVICE1 _at_ 0x8000;

/* sfr P1 = 0x90; */

sbit STROBE1 = 0x90;
sbit PE1 = 0x91;
sbit BUSY1 = 0x92;
sbit ACK1 = 0x93;

unsigned char code DEVICE2 _at_ 0xA000;
sbit STROBE2 = 0x94;
sbit PE2 = 0x95;
sbit BUSY2 = 0x96;
sbit ACK2 = 0x97;

unsigned char curr_device = 0;
unsigned int time_out = 0;

void init_devices(void);
void scan_device1(void);
void scan_device2(void);
void print_device(void);
void insert_q(unsigned char ,unsigned char );
void delete_q(unsigned char );

main()
{
    init_devices();
    P1 = 0x33;
    CTRL_B255 = 0x08; /* Port A - output */
                    /* Port C Lower - output,Port C Upper - input */

    PORTC_B255 = 0x01;

    BUSY1 = 1;
    ACK1 = 0;

```

```

        }
        BUSY1 = 1;
    }
void scan_device2()
{
    unsigned char byte;

    BUSY2 = 0;
    if (STROBE2 == 0)
    {
        if (!device_locked)
        {
            device_locked = 1;
            curr_device = 2;
        }
        if (!byte_updated2)
        {
            byte_updated2 = 1;
            BUSY2 = 1;
            byte = DEVICE2;
            insert_q(2,byte);
        }
    }
    else
    {
        if (device_locked)
        {
            if (curr_device == 2)
            {
                if (time_out++ == 1000)
                {
                    time_out = 0;
                    device_locked = 0;
                    curr_device = 0;
                }
            }
        }
        byte_updated2 = 0;
    }
    BUSY2 = 1;
}

void print_device(void)
{
    switch(curr_device)
    {
        case 0x01:    delete_c(1);
                    break;

        case 0x02:    delete_c(2);
                    break;
    }
}

void insert_q(unsigned char device,unsigned char byte)
{
    switch(device)
    {
        case 1: buffer1[to_ptr1] = byte;
               to_ptr1++;
               break;

        case 2: buffer2[to_ptr2] = byte;
               to_ptr2++;
    }
}

```

CHAPTER X
CONCLUSION

A software has been developed in "C" language for controlling the PC's to share a single printer. The software has been tested for 2 PC's sharing a single printer using PC's in the Department of Computer Science and Engineering at the college. It has been found that all the PC's share the printer successfully. This interface has been designed to make a correlation between cost and utilisation.

The same logic can be used to develop printer sharer interface connecting 'n' PC's. This can be obtained by connecting all the PC's in parallel through corresponding latches and decoders. The time delay programmed can also be changed to meet our requirement. The memory capacity of the buffer can be increased upto 1MB using DRAM.

REFERENCES

1. MORRIS. M. MANO, "COMPUTER SYSTEM ARCHITECTURE". THIRD Edition, Prentice-Hall of India Pvt. Ltd., NEW DELHI, 1993.
2. RAMESH S. GOANKAR, "MICROPROCESSOR ARCHITECTURE, PROGRAMMING AND APPLICATIONS", Wiley Eastern Company, 1993.
3. MORRIS MANO, "DIGITAL LOGIC AND COMPUTER DESIGN", SECOND Edition, Prentice Hall India Pvt. Ltd., NEW DELHI, 1987.
4. "MICROPROCESSOR DATA HAND BOOK", FIRST Edition, BPB Publications, 1989.
5. " EMBEDDED MICROCONTROLLERS AND PROCESSORS", Vol.I, INTEL., 1993.
6. SCOTT FOERSTER, "THE PRINTER BIBLE", Que Corporation., 1990.
7. TTL Logic IC's Manual., 1991.
8. BYRON S. GOTTFRIED, "PROGRAMMING WITH C", Tata McGraw Hill, NEWDELHI, 1991.