MICROPOWER - i

LCD - Technical Reference

Version - 20

Technical Clarification / Suggestion :
✍ / ☎

Technical Support Division,
Vi Microsystems Pvt. Ltd.,
Chennai - 600 096.
Ph: 044-2496 3142, 044-5201 5899

Vi Microsystems Pvt. Ltd.,
Chennai - 96

PREFACE

The "Micropower-i (8085A/Z80A) LCD Technical Reference". Manual provides you with all the basic details of the trainer namely Specifications, the I/O and Memory Mapping, Key Functions, Connector Details and the Circuit Diagram for 8085A&Z80A CPU.

Chapter-1 briefs the Hardware and Software features of the trainer in a nutshell. This chapter gives full details of the facilities that are available with the trainer. The forth-coming Chapters elaborate the features mentioned in this Chapter.

Chapter-2 details the Hardware of the trainer, illustrating the I/O addresses of the various peripherals used.

Chapter-3 shall equip you with the allocation of memory in the trainer and details concerning memory expansion.

Chapter-4 emphasizes on the Keyboard Functions & Display Notations.

Chapter-5 introduces you to the actual implementations of program with our trainer the methodology in which you can execute your program and debug it.

Chapter-6 provides you with a host of useful system call which you shall definitely find to be of great use, while developing software.

Chapter-7 illustrates the connector details used in the trainer to aid you in interfacing your trainer with an external system.

Chapter-8 illustrators the fault analysis points of the trainer kit.

The Appendices provides the CPU data sheets and the Instruction Set summary for 8085A. It also contains the Ics Pinouts, Component & Connector Layouts, and the Circuit Diagrams for Micropower-i.

Suggestions are welcome for further improvement on this manual.

Write to:

The Customer-Support Division,
Vi Microsystems Pvt. Ltd.,
Plot.No.75, Electronics Estate,
Perungudi, Chennai - 600 096.
☎ # 2496 3142, 5201 5899
FAX # 2496 1536.

PREFACE

The "Micropower-i (8085A/Z80A) LCD Technical Reference". Manual provides you with all the basic details of the trainer namely Specifications, the I/O and Memory Mapping, Key Functions, Connector Details and the Circuit Diagram for 8085A&Z80A CPU.

Chapter-1 briefs the Hardware and Software features of the trainer in a nutshell. This chapter gives full details of the facilities that are available with the trainer. The forth-coming Chapters elaborate the features mentioned in this Chapter.

Chapter-2 details the Hardware of the trainer, illustrating the I/O addresses of the various peripherals used.

Chapter-3 shall equip you with the allocation of memory in the trainer and details concerning memory expansion.

Chapter-4 emphasizes on the Keyboard Functions & Display Notations.

Chapter-5 introduces you to the actual implementations of program with our trainer the methodology in which you can execute your program and debug it.

Chapter-6 provides you with a host of useful system call which you shall definitely find to be of great use, while developing software.

Chapter-7 illustrates the connector details used in the trainer to aid you in interfacing your trainer with an external system.

Chapter-8 illustrators the fault analysis points of the trainer kit.

The Appendices provides the CPU data sheets and the Instruction Set summary for 8085A. It also contains the Ics Pinouts, Component & Connector Layouts, and the Circuit Diagrams for Micropower-i.

Suggestions are welcome for further improvement on this manual.

Write to:

The Customer-Support Division,
Vi Microsystems Pvt. Ltd.,
Plot.No.75, Electronics Estate,
Perungudi, Chennai - 600 096.
☎ # 2496 3142, 5201 5899
FAX # 2496 1536.

CONTENTS

CHAPTER ***PAGE NO.***

LIST OF APPENDICES

APPENDIX ***PAGE NO.***

CAPABILITY OVERVIEW

The Micropower-i obtained by you is truly an 8/16 Microprocessor development system furnished with the undermentioned capabilities and can be used either as a stand alone system or as a system with serial interface to a personal Computer to further enhance its perfomanc and exploit to your maximum benefit.

The following are the features of the trainer that can be underlined as being remarkable.

CPUs SUPPORTED:

The Micropower-i is as such a Multi-CPU system with auto-swithing facility between different CPUs. The following are the CPUs with which the Micropower-i can be used as separate systems:

INTEL	8085A
ZILOG	Z80A
SCL	6502
INTEL	8088
MOTOROLA	6800
MOTOROLA	6809

In Micropower-i, the 8085A is supported in the base board. The other CPUs are provided as separate plug-in boards called PIGGY BACK BOARDS. For working with any processor, just plug-in the respective piggy back onto the between INTEL, ZILOG, SYSNERTEK and MOTOROLA family processors. The trainer also has many pheripharals for interface and experimentation.

The monitor EPROMs provided on-board, switch automatically between the different CPUs, the moment the piggy back board is plugged in the kit. These monitor EPROMs occupy 24k bytes of memory.

DOCUMENTATION SUPPORT:

Your trainer is accompained by excellent documentation. For each CPU we provide two manuals, namely a Technical Reference and a User Manual/work Book. These manuals are aimed at transforming you into a handware/software engineer of very high calibre in the field of microprocessors.

CPU	MANUAL	CAT NO
8085A	Micropower-i Z80A/8085A Technical Reference	#MP1-85/80-001
	Micropower-i 8085A User Manual/Student work book	#MP1-85-002
Z80A	Micropower-i Z80A/8085A Technical Reference	#MP1-85/80-001
	Micropower-i Z80 User Manual/Student work book	#MP1-80-002
8088	Micropower-i 8088 Technical Reference	#MP1-88-001
	Micropower-i 8088User Manual/Student work book	#MP1-88-002
6800	Micropower-i 6800/6809 Technical Reference	#MP1-68-001
	Micropower-i 6800 User Manual/Student work book	#MP1-00-002
6809	Micropower-i 6800/6809 Technical Reference	#MP1-68-001
	Micropower-i 6800 User Manual/Student work book	#MP1-09-002
6502	Micropower-i 6502 Technical Reference	#MP1-65-001
	Micropower-i 6502 User Manual/Student work book	#MP1-65-002

SYSTEM CAPABILITY

The trainer contains the CPU, memory and the peripherals. The trainer as a stand alone system can be utilised to your advantage using the onboard keypad and seven segment display of the trainers for your interaction.

The simplest configuration of Micropower-i in a stand alone mode, is to use it as a TRAINING SYSTEM and TEACHING AID in Micro processor Labs. By adding more peripheral / system units such as printer, audio casette recorder, RS232 terminal and IBM compatible PC / XT / AT, you can transform the Micropower -i. A DEVELOPMENT SYSTEM for ADVANCED EXPERIMENTATION and RESEARCH APPLICATIONS. MicroPower-i Depending on the Need and Application, you can use Micropower-i in any one of the following modes.

MODE 1 STANDALONE TRAINER

In this simplest mode, the user has to use the trainer LCD display module available on-board. The user has to key in commands and hex data using the keypad and visualise data and addresses LCD display. Programs can be stored in an Audio tape using an AUDIO CASSETTE RECORDER and a hard copy printout canbe taken from a CENTRONICS COMPATIBLE PRINTER using the printer interface.

The compromise in this mode of operation is that you can view and edit programs one step at a time.

INSTALLATION PROCEDURE

The Micropower -i which you have acquired is a product with many good features. A few guidelines as to how youshould use the trainer are given below. Please follow these hints while working with the trainer.

Unpacking the trainer package you will observe the following.

i. Micropower -i Trainer.
ii. A Technical Reference Manual
iii. A User Manual / Workbook.
iv. A Power Card.

The Micropower-i has multi - output linear power supply and can directly 230V AC / 50Hz mains. Plug in the power cord to the kit as shown in the figure and connect the other end of the pwoer cord to the Ac mains at your place.

(I) INSTALLING WITH 8085A CPU

The base board of Micropower-i directly supports Intel 8085A.

Turn ON the power-switch at the front panel of the trainer. The light glows to indicate the power.

Now, the trainer gets resessted and the message as given below is displayed.

If the message displayed is not proper, then press the RES key and check for the following message.

If still it is a problem, then please contact our customer Support Department for further clarifications.

```
Micro-85AD
#
```

INSTALLATION WITH Z80A CPU

The zilog Z80A CPU is supported as a Piggy Back board as already stated. To use the Micropower-i with Z80, follow the steps as given below.

Make sure the power to the trainer is OFF.

Take the Z80A Piggy Back board and insert it on the CPU bus connector provided just close to the 8085A. The connector is polaised and hence will fix properly with the Z80A Piggy in only one way.

After making sure the Piggy Back board is fixed properly, power-ON the trainer.

The trainer gets resetted and the following message is displayed. Press RES key again, if the display is not proper.

```
Micro-85AD
#
```

DONT'S

i. Please do not insert any add-on card while trainer is powered ON.

ii. Please do not tamper with any of the components in the trainer.

iii. Please do not solder any wire from connectors when the power is up.

iv. Wires are to be soldered only from the solder side of the board.

v. The 26 pin headers should be used only with cables and not with wires soldered from the pins.

vi. Do not attempt to service the trainer in case of problems.

CHAPTER - 1

GENERAL INFORMATION

1.1 INTRODUCTION

This chapter briefs the hardware and software facilities available on Micropower-i and Micro-Z80 CPU. The capabilities of the trainer with regard to memory, peripherals, key functions are all mentioned here. For further explanations you are requested to consult the respect chapters.

1.2 SPECIFICATIONS

1.2.1 HARDWARE SPECIFICATIONS

1. PROCESSOR, CLOCK FREQUENCY :

Intel 8085A at 3.072 MHz

Zilog Z80A at 4MHz

2. MEMORY ALLOCATION / CONFIGURATION :

Monitor EPROM	:	0000	-	1FFFH
EPROM Expansion	:	2000	-	3FFFH
System RAM	:	4000	-	5FFFH
Monitor Data Area	:	4000	-	40FFH
User Ram Area	:	4100	-	5FFFH
EPROM / RAM Expansion	:	6000	-	FFFFH (40K)
		Without Auto Switching facility for other CPU		

3. **INPUT / OUTPUT**

Parallel	:	48 I/O lines using two numbers of 8255.
Serial	:	2 Nos. of RS232 serial interface (one optional) using 8251A USART. Baud rate selectable from 300 to 9600 under software control for both the serial ports.
Timer	:	Three channel16-bit programmable timer 8253. Channel 0 is used as baud rate clock generator for 8251A (com1) USART and Channel 2 for com2.
Printer	:	One centronics compatible parallel printer port.
Interrupts	:	8 Interrupts lines using 8259.

4. LCD INTERFACE

16 × 2 / 20 × 4 LCD Display Module

5. KEYBOARD

IBM PC keyboard interface.

6. LCD INTERFACE

16 x 2 / 20 x 4 display Module.

7. SPEAKER AND SPEAKER INTERFACE

A 2.25" diameter speaker is provided for user expansion.

8. ONBOARD BATTERY BACKUP

On board battery backup facility is provided for 16K RAM (Address 4000 - 7FFF).

9. ONBOARD EPROM PROGRAMMER

28 Pin ZIF socket with circuitry mounted on the board to blank check, program and verify EPROMS 2716, 2732, 2764, 2764A, 27128 and 27128A. ALL operations (Selection of EPROM, programming mode, programming voltage etc.) are software controlled.

10. SYSTEM POWER CONSUMPTION

+5V	:	1Amp
+12V	:	200mA
-12V	:	100mA
+30V	:	300mA if EPROM programmer is used.

11. POWER SUPPLY SPECIFICATIONS:

Model	:	LPOW-001A
Mains	:	220 Volts/AC at 50 Hz
Outputs	:	1) +5 Volts, 3 Amps Rgulated 2) +2 Volts, 150 ma Regulated 3) -12 Volts, 150 ma Regulated 4) +30 Volts, 500 ma Unregulated

12. PHYSICAL CHARACTERISTICS

Micropower-i PCB	:	313mm × 245mm
Cabinet	:	315 mm × 245mm × 48mm
Weight	:	4.1 Kg.

13. BUS EXPANSION

A new concept of VXT bus and VME_Microbus has been incorporated in Micropower-i which facilitates to patch up any extra hardware to Micropower-i. An unlimited number of add-on boards could be added this way to interface to the hardware available on Micropower -i. All buffered address, data and control signal are brought out to this bus. One VME_MICROBUS CONNECTOR IS PROVIDED ON-BOARD.

Note

The power supply used is meant only for the trainer and the add-on boards used alongwith the trainer. The user is therefore, requested not to use this power supply for any external applications.

CHAPTER - 2

HARDWARE DETAILS

2.1 INTRODUCTION

The detailed hardware anatomy of Micropower -i, explained in this chapter, highlights on each and every technical aspect concerned with memory and peripheral interfacing.

The Standard configuration comes with Intel 8085 on the base board. The Z80 CPU is supported as piggy back both Intel 8085 and Zilog Z80.

Note

All addresses either memory or I/O are in hex, unless otherwise specified.

2.2 CONTROL SIGNALS DESCRIPTION

The following explanation outlines the working of the trainer as a system.

i) Address and Data bus

The 8 bit bidirectional data lines and the 16 address lines are brought to buffers. The 8 data lines are buffered through an octal transceiver. The..outputs of these ICs comprise the 16 bit address and 8 bit data bus. The direction of data transfer is decided by the. direction select input of the transceiver depending upon a RD* or WR* cycle.

ii) Control Bus

The other bus is the control bus. The control signals required for proper operation of the system are the IOR* (I/O Read), IOW* (I/O Write), MR* (Memory Read) and MW* (Memory Write). The peripherals on the trainer are all I/O mapped and hence to input from or output to a peripheral IOR* and IOW* are utilised. The memory read and memory write signals are used to output enable an EPROM & RAM and write into a RAM respectively. These signals are generated from the IORQ*,MRQ* and WR*, RD* signals.

2.3a. 8085 CPU (U1)

The Intel 8085 used in Micropower-i is operated at a clock rate of 3.072 Mhz with the peripherals and memory as mentioned in Hardware specifications.

Using its 16-bit address and 8-bit data bus the memory is configured for 64 K Using eight numbers of 2764/6264 (0000 to FFFF).

The peripherals are all I/O mapped to 8085 using IO/M* signal.

The RESET IN* signal provided from the keyboard RESET switch.

2.3b. Z80 CPU

The Z80 CPU available on the Piggy Board also is operated at 4 Mhz clock rate. The peripherals are all I/O mapped using IORQ* signal.

Similar to 8085, it too can access 64 K bytes of memory using its 16-bit address bus and 8-bit data bus.

2.4 MEMORY CONFIGURATION

Please refer to Chapter 3 of this manual for a detailed description on memory configuration of Micropower-i.

2.5 BATTERY BACKUP

Battery backup facility is provided optionally for 16K RAM from address 4000 - 7FFF at sockets U2 and U3. A 3.0V Lithium battery is used which provides power to RAM during power - off and gets charged during power - on. The data in RAM can be retained to a maximum of 12 Hours if the battery has been charged to its maximum.

2.6 ALLOCATION OF I/O ADDRESS

The peripherals available on Micropower -i are all I/O mapped. The complete I/O allocation table for the full fledged professional trainer is given in figure 2.1. As seen from the table, the on-board peripherals occupy I/O addresses from 00 to 3F. The add- on application boards occupy I/O addresses, the rest are available to the users for their own development purposes. Succeeding sections briefly describe the various I/O peripherals and interfaces available on Micropower -i.

I/O ALLOCATION TABLE

FOR 8085 & Z80

F0	NOT USED LEFT FOR USER DEVELOPMENT	FF
80	USED FOR ADD-ON APPLICATION BOARDS	EF
40	NOT USED LEFT FOR USER DEVELOPMENT	7F
3C	LCD CONTROL AND DATA	3D
38	A/D CONVERTER CHANNEL SELECT(ADC0809)	3A
34	PRINTER DATA	37
30	PRINTER CONTROL	33
2C	PRINTER STATUS	2F
26	A/D CONVERTER SOC(ADC0809)	27
24	A/D CONVERTER DATA(ADC0809)	25
20	D/A CONVERTER(DAC0809)	23
1C	EPROM CHIP ENABLE	1F
18	PROGRAMMABLE INTERRUPT CONTROLLER(8259)	1B
14	EPROM 8255 SEL	17
0C	PARALLEL PORTS(1&2)(8255)	13
08	USART-I & USART-II(8251)	0B
04	PROGRAMMABLE INTERVAL TIMER(8253)	07
00	KEYBOARD/DISPLAY CONTROLLER(8279)	03

Figure 2.1

2.7 TIMER INTERFACE : (U4)

i. Device Used : Intel 8253 - Programmable interval timer.

ii. Key features of 8253 : 3 independent 16-bit counters. Programmable counter modes. Input clock can be upto 2MHz. Programmable count from 2 to 2^{16}.

iii. System Mapping : I/O Mapped I/O.

iv. Channel 0 :

Input Clock	:	1.5MHz
Output Clock	:	Depends on selection of baud rate 0
Used for	:	Baud Rate generation for 8251 USART.

v. Channel 1 :

Input Clock	:	1.5MHz
Output Clock	:	Depends on selection of baud rate.
Used for	:	Reserved for single step operation.

vi. Channel 2 :

Input Clock	:	Brought to connector. Hence a HL clock of amplitude from 0 to +5V and frequency less than 2 Mhz com given by user.
Output Clock	:	User determined
Used for	:	Available for user development

vii. Connector Termination : Clock input and clock outputs are brought out to connector P5.

viii. I/O Address:

FUNCTION	I/O ADDRESS	CONNECTOR NO.	SOCKET NO.
Control Register	07		U4
Channel 0	04		U4
Channel 1	05	P5	U4
Channel 2	06	P5	U4

2.8 RS232 SERIAL INTERFACE (U2, U3, U16, U27)

i. Device Used : Intel 8251A -Universal synchronous/ Asynchronous Receiver / Transistor.

- Two numbers are available.

ii. Key Features of 8251 : Both Synchronous and asynchronous operations. Full duplex, double buffered, transmitter and receiver. Parity, overrun and framing error detection.

iii. System Mapping : I/O mapped I/O.

iv. Input clock for 8251 : 3.072MHz

v. Baud Rate : 9600,4800,2400,1200,600,300

vi. Baud Clock : Baud clock for first 8251 (U16) provided by channel 0 of 8253 (U4).
Baud clock for second 8251 (U27) provide by channel 2 of 8253 (U4).

vii. Initialised to : 9600 Baud rate, 8 bits, one stop bit and no parity during transmission / reception.

viii. Drivers Used : MAX232 - Serial Transmitter.
RS232 Serial Receiver.

ix. Connector Termination : The RTS, CTS RxD and TxD lines are terminated at the connector P3 and P4 for two respective 8251, the connector being Amphenol standard 9 pin D type male.

x. I/O Address :

Function	I/O Address	Connector No.	Socket No.
I 8251 Control / Status	09	P3	U16
I 8251 Data	08	P3	U16
II 8251 Control / Status	0B	P4	U27
II 8251 Data	0A	P4	U27

2.9 PARALLEL I / O INTERFACE (U5, U6)

i. Device Used : Intel 8255 - Programmable Peripheral Interface.

ii. Key Features of 8255 : 24 Programmable I/O lines configured as three 8-bit ports. Direct bit set / reset capability. Three modes of operation namely basic I/O, strobed I/O and Bi directional bus.

iii. System Mapping : I/O Mapped I/O

iv. Connector Termination : 48 of the TTL I/O lines are terminated at two 26 pin IDC headers P6 and P7.

v. I/O ADDRESS

FUNCTION	I/O ADDRESS	CONNECTOR NO.	SOCKET NO.
I 8255 control register	0F		U5
I 8255 Port A	0C	P6	U5
I 8255 Port B	0D	P6	U5
I 8255 Port C	0E	P6	U5
II 8255 control register	13		U6
II 8255 Port A	10	P7	U6
II 8255 Port B	11	P7	U6
II 8255 Port C	12	P7	U6

2.10 INTERRUPT INTERFACE (U52)

i) Device Used : Interl 8259 -Programmable Interrupt Controller.

ii) Key Features of 8259 : Eight - Level Priority Controller expandable to 64 levels by cascading 8259s. Programmable Interrupt modes. Individual request mask capability.

iii) System Mapping : I/O mapped I/O.

iv) INT Interrupt of 8259 : Connected to Intel 8085 INT line. Not connected in Z80.

v) Initialised To : Fully nested mode in 8085 CPU. Polled mode in Z80 CPU.

vi) Connector : Eight lines of interrupt terminated to connector P18. (S401 10 pin connector).

FUNCTION	CONNECTOR NO.	SOCKET NO.
ICW1	P18	U52
ICW2	P18	U52
ICW3	P18	U52
ICW4	P18	U52
0CW1	P18	U52
0CW2	P18	U52
0CW3	P18	U52

2.11 IBM PC KEYBOARD INTERFACE

i. Device Used - 4015 Shift register

ii. Mode Used - Software Polled Mode

iii. System Mapping - I/O Mapped I/O

iv. Keyboard Used - IBM PC Keyboard

FUNCTION	I/O ADDRESS	CONNECTOR NO.	SOCKET NO.
Data read	00	-	U19

2.12 LCD INTERFACE

i. Device Used - 16 × 2 / 20 × 4 LCD Module

ii. System Mapping - I/O Mapped I/O

FUNCTION	I/O ADDRESS	CONNECTOR NO.	SOCKET NO.
Diplay / Control Data	3C	LCD	-
RS & DIOW	3D	LCD	U21

2.13 PRINTER INTERFACE (U11, U19, U22)

i. Device Used : 74LS273, 74LS244, 74LS174.

ii. Printer Status : 74LS244 (U22)
5 Input Buffer

iii. Printer Control : 74LS174 (U11)
4 Bit Output Latch

iv. Printer Data : 74LS273 (U19)
8 Bit Output Latch

v. Key Features : The lines are centronics compatible. Can be used for printing or as a general purpose input / output port that matches the design.

vi. System Mapping : I/O Mapped I/O.

vii. Connector Termination : Terminated at P14, Ampenol 25 pin D female connector, are the centronics printer port compatible lines.

viii. I / O Address

FUNCTION	I / O ADDRESS	CONNECTOR NO.	SOCKET NO.
Printer Control	30	P14	U22
Printer Status	2C	P14	U11
Printer Data	34	P14	U19

2.14 AUDIO CASSETTE INTERFACE

i. Device Used : 74LS74, 74LS74
-Filtering and wave shaping networks are provided for reading and writing tapes.
-74LS74, used for tape write.
-74LS74, used for tape read.

ii. Key features : File handling capability
Maximum of FF files can be stored in an audio file.

iii. System Mapping : I/O Mapped

iv. Connector termination : The output of -74LS74 is terminated at MIC socket P12 and Tape read connector is terminated at EAR socket P11.

v. I/O Address

FUNCTION	CONNECTOR NO.	SOCKET NO.
Audio Cassette Write	P12	U16
Audio Cassette Read	P11	U16

2.15. ADC INTERFACE (U51, U47, U48, U49)

i) Device Used : National Semiconductor's - ADC0809

ii) Key Features : Resolution of 8-bits.
100 micro seconds conversion time.
Uses Successive approximation as conversion technique.
8-Channel multiplexer.

iii) System Mapping : I/O mapped I/O.

iv) ADC Data : Connected to D0 - D7 through the 74LS245 (U47) upon selection of I/O address 24.

v) ADC Channel select : The Channel selects ADDA, ADDB, ADDC and ALE are connected to D0-D3 through 74LS174 (U48) upon selection of I/O address 38.

vi) ADC SOC : The SOC is connected to D0 through a 74LS74 (U49) upon selection of I/O address 26.

vii) ADC EOC : The EOC is connected to one line of 74LS244 (U8) upon selection of I/O address 2C.

viii) 8 Channels of ADC : A 1K Trimpot is connected to Channels, 0 of ADC. The other channels, left free, are terminated at connector P17.

ix) Connector Termination : The 8 Channels of the ADC0809 are terminated to the 16-pin IDC header P13 which is usually referred to as the AN-1 Bus.

x) I/O Address :

FUNCTION	I / O ADDRESS	CONNECTOR NO.	SOCKET NO.
ADC Data	24	P16	U47
ADC Address	38	P17	U48
ADC SOC	26		U49
ADC EOC	24		U8

2.16. DAC INTERFACE (U40)

i) Device Used : National Semiconductor's DAC800.

ii) Key Features : 8-bit high speed current output D to A convertor. Allows differential output Voltages of 20V peak to peak with changes in resistor loads.

iii) System Mapping : I/O mapped I/O.

iv) DAC Data : It is connected to D0 to D7 through a 74LS273 (U44).

v) DAC output : The Current output of the DAC0800 is converted to its equivalent voltage using an LM741 - OP Amp. and the output voltage peak to peak value can be changed by simple changing the load resistor value. Usually it is -5V for hex data 00 and 5V for hex data FF.

vi) Connector Termination : The voltage output of LM741 is terminated at the 16-pin IDC header P17.

vii) I/O Address :

FUNCTION	I/O ADDRESS	CONNECTOR NO	SOCKET NO
DAC	20	P17	U40

2.17 BUS EXPANSION

In Micropower -i, the hardware expansion facility has been provided through a bus based architecture consisting of VXT bus. All the CPU address lines, Data lines and control signals are brought out through address latches, Data transceivers and control signal buffers and terminated at the VXT bus connector. The interface our VBMB series of boards, VXT bus connector are provided (P10).

The bus based expansion helps in the easy plugging of a range of our add-on boards for further experimentation and also makes it convenient for the users to build their own add-on modules.

2.18 POWER SUPPLY: [LPOW 001A]

Model LPOW 001A power supply used in Micropower-i is a high efficiency, multi output, linear power supply. In this model the temperature performance, load, line regulation are better.

POWER SUPPLY SPECIFICATIONS

INPUT	:	230V AC 50HZ
OUTPUT	:	DC Outputs +5V/3A Regulated +12V/150mA Regulated +30v/300mA Unregulated

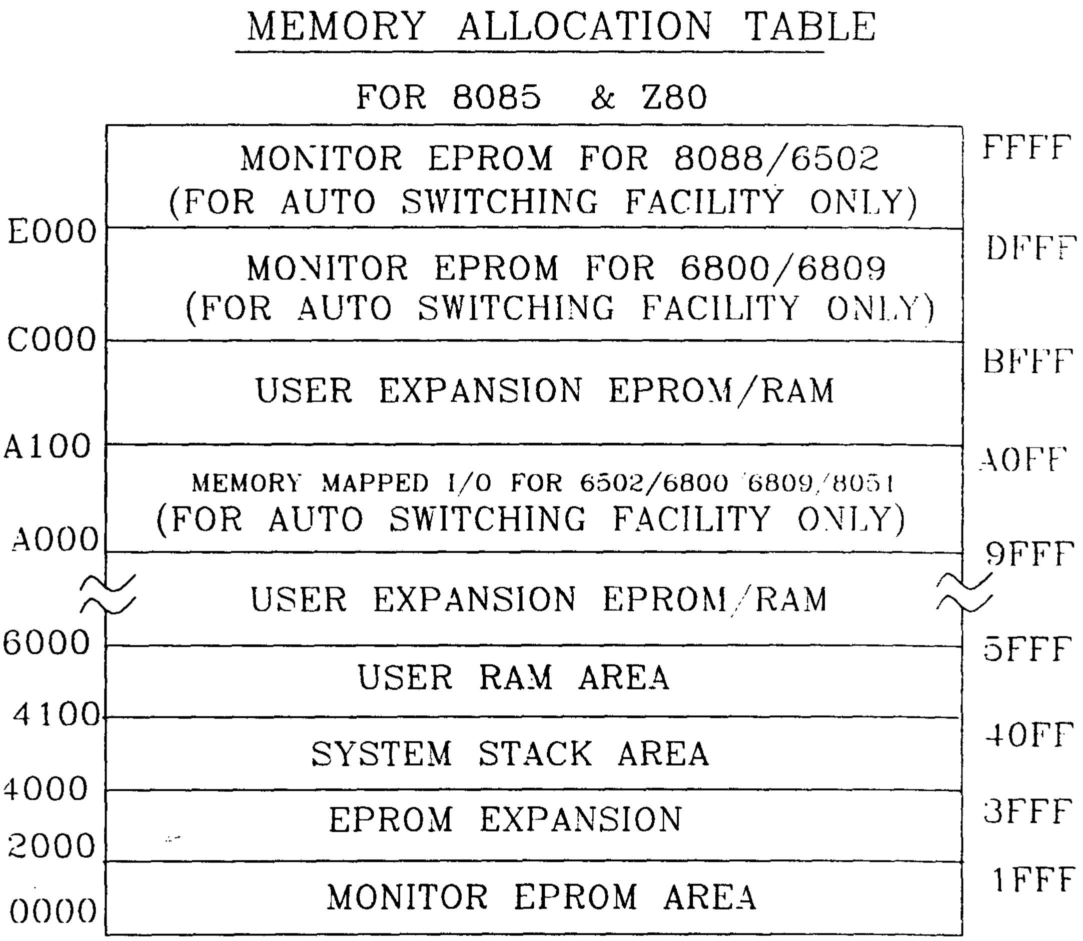
This power supply has four stable outputs with different current ratings.

CHAPTER - 3

MEMORY ALLOCATION

3.1 MEMORY CONFIGURATION

This section explains the memory mapping facilities available on the Micropower-i. The following descriptions on memory is the same for both Intel 8085 and Zilog Z80. The memory configuration of Micropower-i is given in figure 3.1 from which a clear idea about the memory allocation can be had.



MEMORY ALLOCATION TABLE

FOR 8085 & Z80

E000	MONITOR EPROM FOR 8088/6502 (FOR AUTO SWITCHING FACILITY ONLY)	FFFF
C000	MONITOR EPROM FOR 6800/6809 (FOR AUTO SWITCHING FACILITY ONLY)	DFFF
A100	USER EXPANSION EPROM/RAM	BFFF
A000	MEMORY MAPPED I/O FOR 6502/6800 6809/8051 (FOR AUTO SWITCHING FACILITY ONLY)	A0FF
6000	USER EXPANSION EPROM/RAM	9FFF
4100	USER RAM AREA	5FFF
4000	SYSTEM STACK AREA	40FF
2000	EPROM EXPANSION	3FFF
0000	MONITOR EPROM AREA	1FFF

Figure 3.1

3.2 ALLOCATION OF EPROM

The Micropower-i has a standard EPROM configuration of 8K bytes using one 2764 (8K × 8 EPROM). The address for the basic EPROM is 0000 - 1FFF. The EPROM expansion can be available for another 8K using one more 2764 whose address will be from 2000 - 3FFF.

START ADDRESS	END ADDRESS	SOCKET NO.	IC USED	TOTAL CAPACITY
0000	1FFF	U12	2764 × 1	8k Bytes
0000	3 FFF	U12, U23	2764x2	16K Bytes

3.3 ALLOCATION OF RAM

The standard Micropower-i comes with 8K bytes of Random Access Memory using one (8K× 8 RAM), whose address is from 4000 - 5FFF, out of which the first FF locations are used by the system for its data buffers. Hence User RAM area starts from 4100.

START ADDRESS	END ADDRESS	SOCKET NO.	IC USED	TOTAL CAPACITY
4000	5FFF	U13	6264 × 1	8k Bytes

3.4 MEMORY EXPANSION

Apart from the basic EPROM and RAM at addresses 0000 - 1FFF and 4000 - 5FFF respectively, the address 2000 - 3FFF is meant for EPROM expansion. So the remaining 40k is available to the users n which either EPROM or RAM can be used. The addresses for the expansion shall be as follows.

START ADDRESS	END ADDRESS	SOCKET NO.	IC USED	TOTAL CAPACITY
6000	7FFF	U24	6264 / 2764 × 1	8k Bytes
6000	9FFF	U24, U14	6264 / 2764 ×2	16k Bytes
6000	BFFF	U24, U25	6264 / 2764 × 3	24k Bytes
6000	DFFF	U24, U15	6264 / 2764 × 4	32k Bytes
6000	FFFF	U24, U26	6264 / 2764 × 5	40k Bytes

It auto-switching facility is provided for the six CPUs, then the address from C000-FFFF will be occupied by 6800/6809 and 8088/6502 monitor EPROM each occupying 8K bytes of memory. While using 6502, 6800 the address from A000 to A0FF cannot be used for memory expansion for those addresses are used for the memory I/O of these CPUs.

CHAPTER - 4

SOFTWARE FEATURES

4.1 INTRODUCTION

This chapter describes the commands involved in Micro-85 AD which can be given from an IBM-PC keyboard. The Micro-85 AD can accept any command related to the following short listed functions once in the command prompt mode, indicated by a "#" in the leftmost position of the second row in the LCD Module. The functions that can be performed to using the commands are as follows.

* Display and substitute memory locations.

* Display and modify the registers of the CPU.

* Enter the initialise execution of your own programs.

* Debug your program through the single step facility provided by the monitor.

* Write or Read to or from an audio tape.

* Fill a block of RAM memory.

* Move a block of memory to RAM within RAM.

* Compare two blocks of memory.

* Insert bytes into RAM memory.

* Delete bytes from RAM memory.

* Input a byte from an input port.

* Output a byte to an output port.

* Add-on board routines.

* Transmits or receive block of data through the serial port.

* Print a block of memory.

* Use onboard EPROM programmer.

In Micro-85 AD commands can be given from an IBM-PC keyboard. So the keyboard standard is universal. Each command in dealt in detail in the following sections.

The reset message in Micro-85 AD is as follows.

```
Micro-85AD
```

The command can be entered from the # prompt.

The syntax notations used in the following descriptions are as follows,

i) <Addr>, <Start Addr>,
<End Addr>, <Dest Addr>
<Program End Addr>,
<Insert Addr> -- 16 bit hex address to be entered by user.

ii) <Data> -- 8 bit data to be entered by user.

iii) <File Name> -- An 8-bit hex data which specifies the file name in Tape commands.

iv) <Port Addr> -- An 8-bit hex data that specifies an I/O port.

v) <CR> -- Carriage Return.

Note:

1) A command should be entered immediately after the # prompt without blank space.

2) Unless otherwise specified the space in he commands may be used or may not be used.

4.2 COMMAND LINE EDITOR FEATURES:

i) Single line editor can process upto 40 charactrs.

ii) Valid key functions are:

Enter - To validate an entry.
To Increment memory location.
To select from menu.

___ -	-	To decrement memory location To select from menu.
> .	-	To terminate a command
Back Space	-	To delete a character and comeback one position
4 Left Arrow	-	Left arrow key, to come back one position without deleting a character.
6 Right Arrow	-	Right arrow key, to move right one position wihtout deleting a character.
. Del	-	To delete a character in a command line.
0-9	-	Numeric characters from 0 to 9.
A-Z	-	Alphabetic characters from A to Z.

4.3 SUBSTITUTE MEMORY COMMAND

FUNCTION

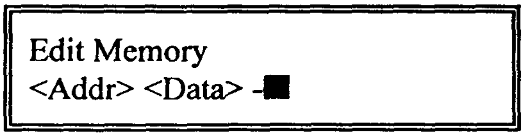
This command is used to examine the contents of selected memory locations & modify the RAM contents of desired.

SYNTAX

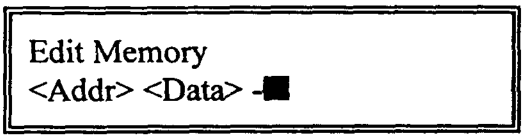
SU - Substitute byte command word.
<Addr> - 16 bit RAM address
<CR> - Keyboard Return

PROCEDURE

i. After entering a substitute command, the contents of that address are displayed as follows.



```
Edit Memory
<Addr> <Data> -■
```

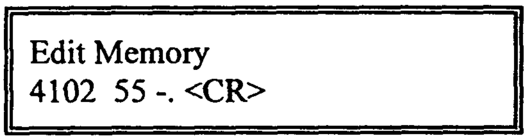


```
Edit Memory
<Addr> <Data> -■
```

ii. Use enter to examine the next location or to increment the memory location and the '-' key to view previous locations. (i.e) to decrement the memory location.

iii. To modify the contents, enter the new data and press enter key.

iv. To terminate this command press a dot (.) and then enter carriage return.



```
Edit Memory
4102  55 -. <CR>
```

```
Edit Memory
4102  55 -. <CR>
```

ERROR CONDITIONS

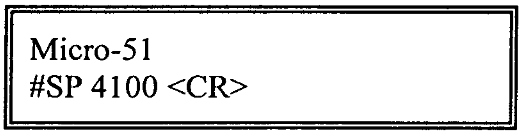
i. Attempting to modify the contents of a read only or non-existent memory locations.

ii. Invalid address (i.e) not within the range of 0000 - FFFFH.

iii. Invalid data (i.e) not within the range of 00 to FFH.

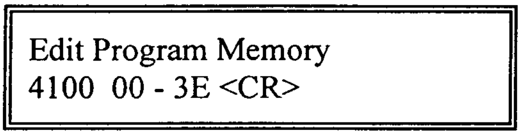
EXAMPLE

To modify the contents of 4100 to 3E and 4101 to 22.



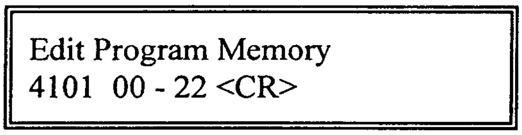
```
Micro-51
#SP 4100 <CR>
```

Enter the substitute command as shown in this figure



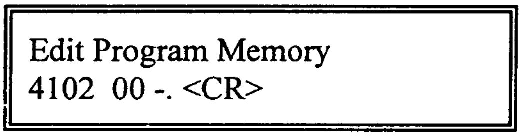
```
Edit Program Memory
4100  00 - 3E <CR>
```

The Data 3E is entered and carriage return is pressed. Now 74 is entered in 4100 location.



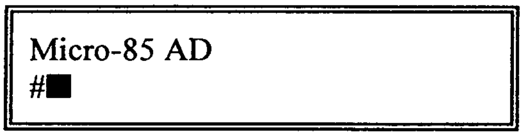
```
Edit Program Memory
4101  00 - 22 <CR>
```

The data 22 is entered.



```
Edit Program Memory
4102  00 -. <CR>
```

To terminate this command press dot (.) and carriage return.



```
Micro-85 AD
#■
```

Now Micro-51 EB is at command prompt.

4.4 REGISTER COMMAND

FUNCTION

To examine and modify the special function register contents of the CPU.

SYNTAX

#R <CR>

R	-	Register View / Modify Command Word.
<CR>	-	Carriage Return.

PROCEDURE

i. Enter register command when prompted for entry.

ii. Initially "A" register is displayed.

PROCEDURE

i. Enter register command when prompted for entry.

ii. Initially "A" register is displayed.

```
Register View
A = 80 - ■
```

iii. Press enter or "-" key to view the subsequent or the previous register in the order as given in the table below.

Register	Description
A	Accumulator
PSW	Process Status Word
PCH	Program counter high byte
PCL	program counter low byte
SPH	Stack Pointer High byte
SPL	Stack Pointer Low byte
H	Register H
L	Register L] HL Pair
D	Register D
E	Register E] DE Pair
B	Register B
C	Register C] BC Pair
INT-REG	Interrupt Register

iv. Change data to any register if desired and press carriage return.

v. The sequence is circular and therefore pressing enter key after the last register in order, will again display the first one.

vi. To transmit this command press dot (.) and carriage return.

EXAMPLE

To modify the contents of DPH register to 55, follow the sequence given below.

Micro-85 AD #R < CR>	Enter Register command
Register View A = 0B<CR>	Skipping "A" register by pressing enter.
Register View PSW=44<CR>	Skipping "B" register by pressing enter.
Register View PCH =13 <CR>	Skip program counter high byte.
Register View PCL =63 <CR>	Skip program counter low byte.
Register View SPH = 40<CR>	Skip stack pointer high byte.
Register View SPL= 42 <CR>	Skip program counter low byte.
Register View H= 1C 55 <CR>	Modify H contents to 55H.

```
Register View
L= BF <->
```

Decrement and again view the contents of H register.

```
Register View
H=55. <CR>
```

```
Micro - 85AD
#■
```

Back to command prompt.

4.5 GO AND EXECUTE COMMAND

4.5.1 GO & EXEC COMMAND

FUNCTION

The GO command is used to RUN a program. This command transfers control of the 8085 CPU from the monitor program to user programs.

SYNTAX

G <Addr> <CR>

G	-	GO Command.
<Addr>	-	16 Bit Address
<CR>	-	Carriage Return

PROCEDURE

i. Type GO command with address from where execution should start.

ii. Now, the control is transferred to the address entered by you and the display will be as shown below.

```
Go Execute
#G <Addr>
```

iii. To exit from the execution and to return control to the monitor, press Reset key in Micro-power-i LCD kit.

NOTE

Before transferring control from monitor program to user programs, the register contents are updated using their respective buffers.

For instance, if you initialise A - 23, using register command, and then use the GO command. "A" will be updated to 23 before executing the user program.

EXAMPLE

To start the execution of a user program at 4100.

```
Micro - 85AD
G 4100 <CR>
```

Enter GO Command

```
Go Execute
#G 4100
```

Transfer Control to user program

4.5.2 GO & EXEC WITH BREAK POINT

FUNCTION

This command executes a block of program whose start and end address specified.

SYNTAX

#G <Start Addr> <End Addr> <CR>

G	-	GO command, with break point.
<Start Addr>	-	Program Start Address.
<End Addr>	-	Program End Address. i.e Break Point address.
<CR>	-	Carriage Return

PROCEDURE

i. Enter GO command with the above said syntax.

ii. The program is executed only till the break-point. After reaching breakpoint, the registers are all saved up and break address with its contents are displayed. After some time the control automatically returns to command prompt. After reaching the breakpoint the display will be as follows.

```
Brk Point Reached
<Break Pt. Addr> - <Data>
```

NOTE

i. You cannot break point as instruction in a ROM (Read Only Memory).

ii. The break point once set will be cleared after reaching it. So break point address must be specified each time the program is to be executed with break point.

iii The break point address specified must contain data which will taken by the processor as an opcode and not as an operand.

ERROR CONDITIONS

Attempting to break point a ROM location or Non-existent memory location.

```
Micro - 85AD
#G 4100 4125 <CR>
```

Enter program block address.

```
Brk Point Reached
4125 - 74.
```

Break point Reached indication.

```
Micro - 85AD
#■
```

Back to Command prompt.

4.6 TRACE COMMAND

This command helps the user to execute programs in steps i.e instruction by instruction. This command will be very helpful while debugging programs.

SYNTAX

#TR <Addr> <CR>

TR	-	Trace Command Word
<Addr>	-	Program Starting Address
<CR>	-	Carriage Return

PROCEDURE

Fi. Enter the trace command with the above syntax.

ii. Now, the memory address and its contents (i.e) opcode are displayed. Press enter to execute this instruction. Now the first instruction is executed and the registers are updated.

iii. The next instruction to be executed and the memory location are displayed now.

iv. Since, registers are updated after each individual instruction, you can view the register contents after any execution of individual instructions.

v. Press dot(.) and enter key to return to monitor and view registers or memory locations as desired.

vi. Continue single stepping after viewing register contents by again executing trace command.

NOTE

Please refer chapter - 5 "User Program Execution" to know the working of this command during a user program debugging.

4.7 FILL COMMAND

FUNCTION

This command permits a block of RAM memory to be filled with desired data byte.

SYNTAX

#F <Start Addr> <End Addr> <Data> <CR>

F	-	Fill Command Word
<Start Addr>	-	Program Starting Address
<End Addr>	-	Program Ending Address
<Data>-		Data to be filled
<CR>	-	Carriage Return

PROCEDURE

i. Enter the fill command with the above syntax.

ii. Now, the block will be filled with the specified byte and control returns to monitor.

iii. You can monitor the current address being filled in the display.

ERROR CONDITION

i. If the start address > end address.

ii. Attempting to fill ROM location or a non-existent memory location.

EXAMPLE

To fill a block from 4100 to 4200 with data 33.

```
Micro - 85AD
#FP 4100 4200 33 <CR>
```

Enter Fill Command

```
Fill Block
Address - 4100
```

The filling operation is being performed and the current address is displayed.

```
Fill Block
Address - 4200
```

After Fill operation is performed.

```
Micro-85AD
#■
```

Back to Prompt.

4.8 BLOCK MOVE COMMAND

FUNCTION

This command moves the contents of a specified block of memory to another block whose start address is specified [to RAM].

SYNTAX

#MP <Start Addr> <End Addr> <Dest Addr> <CR>

M	-	Block Move Command word
<Start Addr>	-	Program Starting Address
<End Addr>	-	Program Ending Address
<Dest Addr>	-	Destination Address
<CR>	-	Carriage Return

PROCEDURE

i. Enter the block move command with the above syntax.

ii. Now, the data is moved from source block to destination block and the control is transferred to monitor program.

iii. Address can be monitored in the display throughout the move block function.

NOTE

i. Actually this command copies the specified block to the destination block byte by byte. The contents of the source block are not altered.

ii. So, if the destination address specified falls within the source block, then it does not do the work of the MOVE. The memory locations overlap and filling of locations with a similar data happens.

iii. The EPROM locations can be from but cannot be copied into.

iv. While entering the move command the display automatically will shift left when the command exceed 16 bytes.

ERROR CONDITIONS

i. Start Address > end address of the source block.

ii. Attempting to move data to EPROM or Non-existent RAM locations.

EXAMPLE

To move the contents of the block from 4100 to 4200 to the block starting at 5100.

```
Micro-85AD
#MP 4100 4200 5100
```

Enter Move Command.

```
Move Block
ADDRESS-5100
```

Move Block is in progress. Current address can be monitored.

```
Move Block
ADDRESS-5200
```

End address is reached.

```
Micro-85AD
#■
```

Back to Command prompt.

4.9 BLOCK COMPARE COMMAND

FUNCTION

This command compares the contents of two block of memory locations and displays whose contents are also be modified.

SYNTAX

#BC <Start Addr> <End Addr> <Dest Addr> <CR>

BC	-	Block Compare command word
<Start Addr>	-	Program Starting Address.
<End Addr>	-	Program Ending Address.
<CR>	-	Carriage Return

PROCEDURE

i. Enter the block compare command with the above syntax.

ii. Now, both blocks will be compared.

iii. If both blocks contains identical data, then system returns to prompt to indicate that both the blocks are equal.

iv. if unequal data is found anywhere then comparative display is made as shown below.

```
<Start Addr> = <Data> / <Dest Addr> = <Data>
B1K. Compare
```

Now, the destination data contents can be modified or just enter can be pressed for further comparison to be done.

v. So press enter key to skip editing of destination block data.

vi. The system displays command prompt, if the rest of the blocks are equal, or after it has finished the full comparison between the two blocks.

In the example given here, first fill two blocks with a similar data and then compare them both.

Now, change data within the blocks and check once again using the compare command. This is much easier way to understand the block compare command.

EXAMPLE - 1

First fill a block of memory from 4100 to 4500 with a hex data say 66, and compare the block from 4100 to 4200 with the block from 4300 to 4400. They should be equal.

```
Micro-85AD
#BC 4100 4200 4300 <CR>
```

Issue Block Compare command

```
Block Compare
Address - 4300 66
```

The two blocks are being compared and now address is displayed along with data.

```
Block Address
Address - 4400 66
```

End address is reached

```
Micro-85AD
#■
```

Now, Micro-51 EB is at command prompt.

Since the two blocks specified contain the same data at respective address locations (i.e) (4100) = (4300); (4101) = (4301) etc., the compare command returns to the command prompt, to indicate that the block are identical.

EXAMPLE - 2

Now, edit memory location 4300 contents to 77 and issue the Block Compare Command

```
Micro-85AD
#BC 4100 4200 4300 <CR>
```

Issue Block Compare command

```
4100=66/4300 = 77
BIK. Compare
```

Since data differs both contents are displayed for comparative study and data can be modified or just enter key can be press for further comparison.

```
Block Compare
Address - 4301 66
```

Since rest of the data in the blocks are identical address and data of the destination block is displayed.

```
Block Compare
Address - 4400  66
```

End address is reached

```
Micro-85AD
#■
```

Now, Micro-51 EB is at command prompt.

4.10 INSERT COMMAND

FUNCTION

This command inserts the specified bytes in the desired memory locations.

SYNTAX

#IS <Insert Addr> <Program End Addr> <No. of Bytes> <CR>

IS	-	Insert Command word
<Insert Addr>	-	Insert Address
<Program End Addr>	-	End Address of the block
<No. of Bytes>	-	No. of bytes to be inserted
<CR>	-	Carriage Return

PROCEDURE

i. Enter the insert command in the above given syntax.

ii. Now, the display will be as follows.

```
Insert Bytes
<Insert Addr> <Data> -#
```

iii. Now, enter the data to be inserted against the address and its current data displayed followed by enter key every time.

iv To terminate this command at any time, press dot (.) and then carriage return.

v. After the specified number of bytes are over, the control is transferred to command prompt.

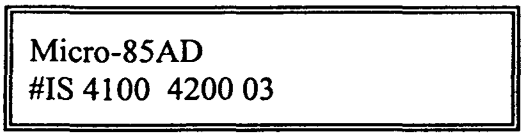
vi. The data entered are written into memory starting from the insert address and therefore the entire block is shifted below one by one to make space for bytes to be inserted.

ERROR CONDITIONS

i. Inserting into Read-only or Non-existent memory locations.
ii. If the insert address is > Program end address.
iii. Invalid data or address.

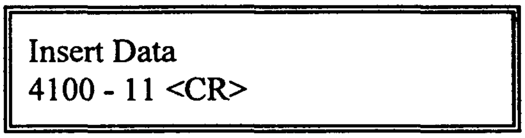
EXAMPLE

To insert three bytes 11, 22, and 33 starting at address 4100, the specified block ending at 4200.



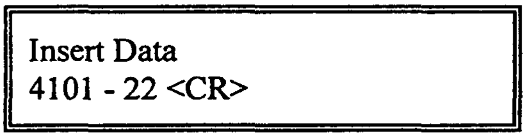
```
Micro-85AD
#IS 4100  4200 03
```

Enter Insert command



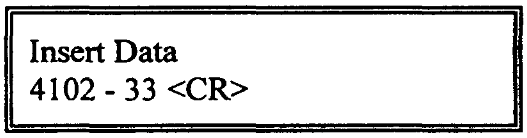
```
Insert Data
4100 - 11 <CR>
```

Insert byte "11" at 4100.



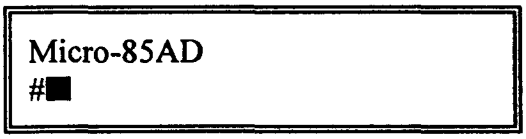
```
Insert Data
4101 - 22 <CR>
```

Insert byte "22" at 4101.



```
Insert Data
4102 - 33 <CR>
```

Insert Byte "33" at 4102



```
Micro-85AD
#■
```

After three valid insertions, control is transferred to command prompt.

NOTE

i. To insert bytes into memory, the block from insert address + 1 to the end address specified is shifted down by one. Now a memory location is free to enter the data. This is what is actually done in the insert command.

ii. For each and every data to be inserted, the same procedure is followed. If you are to insert within a program you have entered, check the jump address to ensure they are correct since you have inserted data into the program.

4.11 DELETE COMMAND

FUNCTION

This command deletes a block of bytes from memory.

SYNTAX

#D <Start Addr> <End Addr> <Program End Addr> <CR>

D	-	Delete Command Word
<Start Addr>	-	Delete block start address.
<End Addr>	-	Delete block end address.
<Program End Addr>	-	Program end address.
<CR>	-	Carriage Return

PROCEDURE

i. Enter the delete command in the above given syntax.

ii. Now, the block starting from block end address +1 to the program end address, will be moved to the block starting from block start address.

ERROR CONDITIONS

i. Attempting to delete bytes from Read-only or Non-existent memory locations.

ii. Block start address > Block end address

EXAMPLE

Delete the block from 4100 to 4105. The end address of the program that contains the block is 4200.

```
Micro-85AD
#D 4100 4150 4200 <CR>
```

Enter Delete Block Command

```
Delete Block
Address - 4150
```

Block from 4150 to 4200 will be moved to 4100, so that previous contents are deleted.

```
Delete Block
Address - 4200
```

Program end addrss is reached

```
Micro-85AD
#■
```

Control is transferred to command prompt.

NOTE

Now if you view location 4100, the data at 4151 would have been moved up here, similarly, data from 4152 to 4101 and so on till 4200.

4.12 INPUT COMMAND

FUNCTION

This command inputs data from the desired port.

SYNTAX

#IN <Port Addr> <CR>

IN	-	Input command word.
<Port Addr>	-	Valid input address (i.e 00 to FFH).
<CR>	-	Carriage Return

PROCEDURE

i. Enter IN command in the above stated syntax.

ii. Now, the data would be read and displayed as shown below

```
Input Data <Data>
IN <Port Addr>
```

Data> - Data read from the input port

iii. To read successive data, just go on pressing enter.

iv. you can also modify the port address.

v. Press dot (.) to terminate the command.

EXAMPLE

Input data from Port Address 11.

```
Micro-85AD
#IN 11 <CR>
```

Enter Input Command

```
Input data 55
IN 11
```

The data from the pot is displayed

```
Input Data 55
IN .<CR>
```

To terminate this command press dot(.) and carriage return.

```
Micro-85AD
#■
```

Back to prompt

4.13 OUTPUT COMMAND

FUNCTION

This command output data to the desired port.

SYNTAX

#0U <Port Addr> <Data> <CR>

OU	-	Output Command word
<Port Addr>	-	Output port address (00 to FFH
<Data>-		Output Data (00 to FFH).
<CR>	-	Carriage Return

PROCEDURE

i. Enter the output command in the above syntax.

ii. Now, the data entered is output to the port specified and the display will be as follows.

```
Output Data <Data>
OU <Port Addr> <Data>
```

iii. Now, again you are in out command mode, so that, you can continuously execute out command or this command can be terminated with a dot (.) followed by a carriage return.

EXAMPLE

To output data 80 to port address 13 and 55 to port address 12.

```
Micro-85
#0U 13  80  <CR>
```

Now, data 80H is output to port 13.

```
Output Data 80
OUT 13 80
```

Now, data 80H is output to port 13.

```
Output Data 80
OUT 12  55 <CR>
```

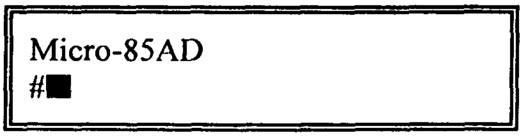
Modify the port address and data.

```
Output Data 55
OUT 12 55
```

Now, data 55 is output to port 12.

```
Output Data 55
OUT .<CR>
```

Terminate the command using dot (.) followed by carriage return.



```
Micro-85AD
#■
```

Back to command prompt

4.14. SERIAL INPUT COMMAND

FUNCTION

This command inputs data from the serial port of Micro-51 EB asynchronously.

SYNTAX

#5I <Start Addr> <CR>
PI - Input Command word.
<Start Addr> - Address to store the serial input data.
<CR> - Carriage Return.

PROCEDURE

i. Enter the serial input command in the above said format.

ii. Now, transmit data from the host systems. (Please refer through 'Note-2' for further discussion on DATA TRANSMISSION FROM HOST SYSTEMS).

iii. The Micro-51EB receives data and stores it from the given starting address and returns to the command prompt on receiving the EOF MARK. (Refer Note - 2).

iv. Using edit memory command check for the data from the starting address specified.

NOTE - 1 :

i. An RS232C serial cable should be connected between the two systems.

ii. Ensure that the baud rate setting of the host system is identical to that set in Micro-51EB.

iii. In Micro-51 EB, the serial data format is set as undermentioned. It should be similar at the host system too.

Serial Data Format of Micro-51 EB

Data	:	8 Bit
Stop Bits(s)	:	1
Parity	:	None

Note - 2

i. The EOF mark is a data string consisting of 5 "?" (3F).

ii. The host may be any system having RS232C serial port. It can be IBM PC/XT/AT, a kit etc.

On a PC to kit communication the "DATACOM" package available with us, directly supports the EOF mark.

In a kit to kit communication, the EOF data is directly supported.

While transmitting serial data from any other system ensure that the transmitting systems adds 5 "?" data at the end of transmission.

ERROR CONDITIONS

i. Writing into a Read-only or a non - existent RAM location.

ii. Due to mismatched reception, which can occur if the baud rates, and other communication parameters are not identical in both the transmitting and the receiving systems.

EXAMPLE

To serial input from USART starting from address 4100.

```
Micro-85AD
#■ SI 4100 <CR>
```

Enter the serial input command.

```
Receiving Data
ADDRESS-4100
```

Waiting for data serial port.

```
Receiving Data
ADDRESS-4125
```

Data Received and address can be monitored throughout.

```
Micro-85AD
#■
```

Back to command prompt

NOTE

Transmit data from the transmitting system only after making the kit into receive mode. If the trainer receives five 3F, it automatically comes to prompt, else reset the kit.

4.15 SERIAL OUTPUT COMMAND

FUNCTION

This command output data serially through serial port of the Micro-51EB asynchronously.

SYNTAX

#SO <Start Addr> <End Addr> <CR>

SO	-	Output command word.
<Start Addr>	-	Start Address of the transmit data block.
<End Addr>	-	End Address of the transmit data block.
<CR>	-	Carriage Return.

PROCEDURE

i. Enter the serial output command in the above said format.

ii. Ensure that the host system is set in the receive mode.

iii. Now, data will be transmitted serially. Micro-51 EB returns to command prompt after data transmission is over.

NOTE

i. If serial port of the receiving system is not proper, then the trainer waits with the message.

ii. Make sure your end of file contains five 3F.

EXAMPLE

To serial output data from 4100 to 4200.

```
Micro-85AD
#SO 4100  4200 <CR>
```

Enter the serial output command

```
Tx-ing Data
ADDRESS - 4100
```

Transmitting data from 4100 to 4200 and address can be monitored throughout the process.

```
Tx-ing Data
ADDRESS - 4200
```

End address reached.

```
Micro - 85AD
#█
```

Back to command prompt.

4.16 ASSEMBLE COMMAND

```
Micro- 85AD
#A <CR>
```

Enter Assembler Command

```
4200
```

Now 4200 is displayed

```
4200 ORG 4100
```

Enter the origin of the program

```
4100
```

Address is displayed

```
4100 MVI A,05
```

Enter the instruction

```
4102 ADI  05
```

Enter <CR> for next instruction and so on. Enter dot(.) for return back.

4.17 UNASSEMBLE COMMAND

```
Micro-85AD
#U <CR>
```

Enter the unassemble command

```
ORG?
DISASSEMBLER
```

Display the origin.

```
ORG? 4100
DISASSEMBLER
```

Enter the origin

```
4100   3E  05
MVI    A,  05
```

Unassemble the opcode and displayed.

```
4102    C6   05
ADI          05
```

Enter <CR> for next address Enter dot(.) for return back.

4.18 PRINT COMMAND

FUNCTION

This command prints the contents of a block of memory onto a centronics compatible parallel printer.

SYNTAX

```
#P      <Start addr>  <End Addr> <CR>
```

P	-	Print Command Word.
<Start Addr>	-	Print block start address.
<End Addr>	-	Print blcok end address.
<CR>	-	Carriage Return.

PROCEDURE

i) Enter print command in the abover format. If a parallel printer is kept connected to the printer port, the printing will be done else printer error will be reported.

ii) The printing will be in the following format.

Address ... 16 data in hex ... ASCII codes for the 16 data.

ERROR CONDITIONS

i) If the printer is not ready or if the interface is not centronics compatible.

ii) If the start address is > end address.

EXAMPLE

```
Micro-85AD
#P 4100 4200 <CR>
```

Enter Print command.

```
Printing Block
```

Printing in prograss, address can be monitored.

```
Printing Block
```

End address reached.

```
Micro-85AD
#
```

Back to command prompt.

Note:

If the printer is not available, or if there is No Paper sensed, error message is displayed.

4.19 BLOCK SEARCH COMMAND

FUNCTION

This command searches a block of memory for a particular byte and that byte canbe modified.

SYNTAX

```
#BS   <Start addr>  <End Addr>   <Byte> <CR>
```

BS	-	Block Search Command Word.
<Start Addr>	-	Start address of the block to be searched.
<End Addr>	-	End address of the block to be searched.
<Byte>	-	Byte to be searched.
<CR>	-	Carriage Return.

PROCEDURE:

i) Enter the Block Search command in the above format.

ii) If no matching bytes ae found then the control goes to command prompt. But if any data matches with the search byte then the display will be as follows.

```
Byte Search
Addr-<Address> <Data>
```

iii) Now, this data can be modified or this command can be transmitted using a dot (.) followed by carriage return.

ERROR CONDITIONS

i) Start address > End address.

ii) Attempt to write in to Read Only or No-Existant memory location.

EXAMPLE

Search from 4100 to 4200 for the byte AA and to modify it to BB. Assume that 4150 location has got AA data.

Enter the Block Search command.

```
Micro-85AD
#P 4100 4200 AA<CR>
```

Search for AA, modify AA to BB.

```
Byte Search
Addr-4150 AA-BB<CR>
```

```
Byte Search
Addr-4200 00
```

Block search completed.

```
Micro-85AD
#
```

Control returns to command prompt.

4.20 EPROM PROGRAMMER:

FUNCTION

This set of commands can be used to blank check, read or write into a blank EPROM from 2716 to 27256A.

SYNTAX

```
#EP<CR>
```

EP - Eprom Programmed Command Word.
<CR> - Carriage Return.

After entering this command the display will be,

```
EP
<EPROM No.>-
```

<EPROM No.>-This can be selected like in a menu. Press enter to forwared select and "-" key for reverse select (refer the given below EPROM selection table). Press "Y" to confirm selection.

Key Pressed	EPROM Selected
0	2716
1	2732
2	2732A
3	2764
4	2764A
5	27128
6	27128A
7	27256
8	27256A

After EPROM selection process the display will be as follows.

```
EP <EPROM NO.>
<FUNCTION>
```

<Function> - These can be read, write (or) blank check. Use cursor key to forward select the functions and "-" key to reverse select. Use "." key and <CR> keys to terminate this command. Press "Y" key to select the function.

EXAMPLE

```
EP
2764A-Y<CR>
```

To select 2764A EPROM

```
EP 64A
Read <CR>
```

To Select read option.

```
EP 64A READ
Addr?#
```

Display after the above selection.

Now address range should be give in the following format.

```
EP <IC NO.> <Function>
Addr? <Start Addr> <End Addr> <Dest Addr> <CR>
```

PROCEDURE:

Insert EPROM into the ZIF Socket with proper polarity.

i) Enter the EPROM Programming command in the above discussed syntax.

ii) Select the EPROM number by confirming with "Y" key.

iii) Select the EPROM function by confirming with "Y" key.

iv) Enter address range.

v) If the function is blank, then blank or not blank message will be displayed, then control goes to EPROM No. selection.

vi) After performing a function (Read, Write or Blank Check) control goes to EPROM selection.

EXAMPLE:

To blank check, program and verify the contents of a 2764 EPROM.

Insert a blank 2764 EPROM in the ZIF Socket and proceed as below. A 2764 EPROM can store upto 8K bytes of data. Hence the starting address is 0000 and the ending address if 1FFF. Let us copy the contents of the monitor from 0000 to 1FFF onto this EPROM.

```
Micro-85AD
#EP <CR>
```

Enter EPROM Programmer command.

```
EP
2716-#
```

Select 2764 IC No.

```
EP
2716-<CR>
```

```
EP
2732-<CR>
```

```
EP
2732A -<CR>
```

```
EP
2764-Y<CR>
```

Confirm selection fof 2764 IC with a "Y" key followed by carriage return.

```
EP 64
Blank Check-<CR>
```

Select blank check option

```
EP 64 Blnk.Chk
Addr? #
```

Now, you are prompted for address entry.

```
EP  64 Blnk.Chk
Address-0000
```

Address is displayed for monitoring the checking process.

```
EP  64 Blnk.Chk
Address-1FFF
```

```
EP  64 Blnk.Chk
PASSED
```

```
EP
2716-<CR>
```

```
EP
2732-<CR>
```

```
EP
2732A-<CR>
```

```
EP
2764-<CR>
```

Select 2764 IC.

```
EP 64
Blanck Check-#
```

You are prompted for function select.

```
EP 64
Blanck Check-<CR>
```

```
EP 64
Program-Y<CR>
```

Select program function

```
EP 64  Program
Addr?#
```

You are prompted for address entry.

```
EP 64  Program
Addr? 0000 1FFF 0000<CR>
```

Enter valid address range.

```
EP 64  Program
Address- 0000
```

EPROM is now programmed.

```
EP 64  Program
Address- 1FFF
```

```
EP 64  Program
Success
```

```
EP
2716-<CR>
```

Now, you are prompted for device select.

```
EP
2732-<CR>
```

```
EP
2732A-<CR>
```

```
EP
2764-Y<CR>
```

Select 2764 IC.

```
EP 64
Blank Check-#
```

Now, you are prompted for function select.

```
EP 64
Blank Check-<CR>
```

```
EP 64
Program-<CR>
```

```
EP 64
Read-Y<CR>
```

Select Read Function.

```
EP 64 Read
Addr?#
```

```
EP 64 Read
Addr? 0000 1FFF 6000<CR>
```

Enter valid address range.

```
EP 64 Read
Address- 6000
```

Data is stored from 6000- 7FFF.

```
EP 64 Read
Address- 7FFF
```

```
EP
2716-.<CR>
```

End EPROM Program command.

```
Micro-85AD
#
```

Back to the command prompt.

Use COMPARE command and compare 0000 to 1FFF with 6000 to 7FFF. They must be identical.

ERROR CONDITIONS

At any point of the above sequence an error message shall be displayed if,

i) A blank-checked EPROM is not blank.

ii) The programmed is not able to write into the EPROM.

iii) The destination address for a read is an EPOM address, or a RAM address without a RAM.

CHAPTER -5

USER PROGRAM EXECUTION

Now that you learnt that key function available on Micropower-i, it should be every easy for you to write a program and either execute it or single step it. This fact may be new to initial programmer and this chapter follows a sequel.

This chapter first explains as to how an 8-bit addition program with object codes is to be written. The procedure to enter the program into the trainer and execute it are given. Execution of a program can be done either using the unconditional GO and EXEC command or using the STEP command.

WRITING A PROGRAM

The following is a simple 8-bit addition program which involves immediate mode of addressing.

```
MVI     A,23      ;First operand to A
MVI     B,22      ;Second operand to B.
ADD     B         ;Add two operands
STA     @4200     ;Store result at 4200.
HLT               ;Halt the Program
```

Now this program is to be entered into the micropower-i kit. These are called object codes which can be found by referring to the instruction set given in the Micropower-i User Manual. The object codes for the above program are as follows.

Hex Address from where program is to be entered	Object Codes	Mnemonics
4100	3E	MVI A,23
4101	23	
4102	06	MVI B,22
4103	22	
4104	80	ADD B
4105	32	STA @4200
4106	00	
4107	42	
4108	76	HLT

ENTERING THE PROGRAM AND EXECUTING IT

The above example program can be executed by directly placing the opcodes in the memory and entering a GO command. The procedure to be followed is as follows.

```
Micro - 85AD
#SU  4100 <CR>
```

Enter substitute byte command

```
Edit  memory
4100 00-3E <CR>
```

Enter the opcodes from 4100 location

```
Edit  memory
4101 00-23 <CR>
```

Enter 23H

```
Edit memory
4102 00-06 <CR>
```

Enter 06H

```
Edit  memory
4103 00-22 <CR>
```

Enter 22H.

```
Edit  memory
4104 00-80 <CR>
```

Enter 80H.

Enter 32H

```
Edit  memory
4105 00-32 <CR>
```

```
Edit  memory
4106 00-00<CR>
```

Enter 00H.

(or)

```
Edit  memory
4107 00- <CR>
```

If previous entry 0 skip this location

```
Edit memory
4108 00-42<CR>
```

Enter 42H.

```
Edit  memory
4109 00-76 <CR>
```

Enter 76H.

```
Edit memory
410A 00-. <CR>
```

After entering the program came back to command prompt.

```
Micro-85AD
#■
```

```
Micro-85AD
#G 4100 <CR>
```

Now, issue the Go command

```
Go Excute
#G 4100
```

The program is now executed.

Now using substitute command verify that the location 4200 has the byte 45.

TRACING THE PROGRAM

```
Micro-85AD
#TR 4100 <CR>
```

Enter the trace command, you can see the instruction at 4100 location is executed and next instruction is displayed.

```
Tracing
4100 3E-<CR>
```

By pressing the carriage return you can execute this instruction

```
Tracing ...
4102 06-
```

The instruction to be executed 24 is displayed. Now register A can be viewed.

```
Tracing ...
4102 06-.<CR>
```

```
Micro-85AD
#■
```

Goes back to command prompt

```
Micro-85AD
#R<CR>
```

Issue register view command

```
Register View
A = 23.
```

Register A content is 23 after execution of MVI A,23 instruction.

```
Register View
A = 23. <CR>
```

```
Micro-85AD
#■
```

Go back to command prompt.

```
Micro-85AD
#TR 4102 <CR>
```

Again trace from 4102.

```
Tracing ...
4102 06-<CR>
```

Trace the full program

```
Tracing ....
4104   80-<CR>
```

Execute ADD B Instruction

```
Tracing ...
4108  32-<CR>
```

Execute STA 4200 instruction

```
Tracing ...
4108 76-<CR>
```

```
Tracing ...
4109 00-.<CR>
```

Execute the location 4200 and verify the further result.

ASSEMBLER LISTOUT

```
Line No   Address   Opcode & operand   Label    Mnemonics   Operand

1                                               ; Program to familiarise with
2                                               ;Assembler listouts!
3                   00    42                    Result:EQU       4200H
4         4100                                         ORG       4100H
5         4100      3A    0C    41                     LDA       (DATA1)
6         4103      47                                 MOV       B,A
7         4104      3A    0D    41                     LDA       (DATA2)
8         4107      80                                 ADD       B
9         4108      32    00    42                     STA       (RESULT)
10        410B      76                                 HLT
11        410C      23                 DATA1:          DB        23H
12        410D      22                 DATA2:          DB        22H
13        410E                                         END
```

The above is a simple assembler listout written in 8085 mnemonics. As seen, the program involves a lot of assembler directives like the ORG, EQU, END and many more which are not specified here. It is a must that one should use these directive while assembling programs in PC. These directives which are used above may be detailed as,

i. Any line starting a ";" can be used to write comments for they will not be taken into assembling.

ii. The "ORG" directive sets the program assembly start address. If this directive is not executed, the assembly start address defaults to 0000.

iii. The "EQU" directive equates the labels used in the program to the specified values. The value specified can be another symbol or any legal arithmetic expression. A label defined using an EQU" directive cannot be redefined again.

iv. The "DB" directives stores the value of the expression specified in an 8-bit storage location, the address of which is got in the order of assembling the program. ASCII character must be bracketed by apostrophes ('A'). If no expression is given, one byte is reserved and zeroed.

v. Finally, the "END" directive defines the end of a program or an included file. This directive is a compulsion to all programs.

Usage of these assembler directives makes the writing of programs much easier. This format is followed in the chapter - 3 of the user manual. Follow up these directives to understand those programs.

CHAPTER - 6

MONITOR SYSTEM CALLS

6.1 INTRODUCTION

The monitor program is written in a MODULAR APPROACH. The users can now write their own program easily and very efficiently, using the monitor program resources.

These resources are essentially SYSTEM SUBROUTINES accessed through a PRINCIPAL ENTRY POINT. Instead of providing as an Interrupt function, it has been implemented as a CALL function to the monitor.

Before making a SYSTEM CALL to the monitor, initialise the Function number, Information number and the pointer, if necessary. The mentioned task under that Function number would be done after the CALL is made and system returns back to the main program.

The principal entry point for this CALL function is 0005. So, to call a function, set the specified registers so that the function you are calling knows what to do. Now CALL 0005 and the desired task is done.

The use of these CALLs makes your programs small and efficient. The following is a summary of the FUNCTION CALLS.

6.2 SYSTEM CALLS DESCRIPTION

6.2.1 TO RESET IBM-PC KEYBOARD (Function 00)

This function resets the IBM-PC Keyboard

Input	:	A=00
		CALL 05
Output	:	None
Registers affected	:	A
Result	:	The Keyboard is reset

6.2.2 TO RESET LCD (Function 01)

This function resets the controller in the LCD module and initialises it for 16 × 2 display mode.

Input	:	A = 01
		CALL = 0005

Output	:	None
Result	:	The display is reset (or) reinitialised.
Registers affected	:	A, C

6.2.3 TO CLEAR LCD (Function 02)

This function clears the displayed matter in the LCD.

Input	:	A = 02 CALL 0005
Output	:	None
Result	:	Display is cleared
Registers affected	:	A

6.2.4 TO ON THE CURSOR IN THE LCD (Function 03)

This function display the cursor in the LCD module.

Input	:	A = 03 CALL 0005
Output	:	None
Registers affected	:	A
Result	:	If the cursor is not present, it is displayed.

6.2.5 TO OFF THE CURSOR IN THE LCD (Function 04)

This function is used to disable the cursor in the LCD Module

Input	:	A = 04
		CALL 0005
Output	:	None
Registered affected	:	A
Result	:	If the cursor is ON, then it will be disabled.

6.2.6 TO SET THE ADDRESS TO DISPLAY A CHARACTER OR A STRING (Function 05)

The LCD Module can be display two rows of 40 characters each. The address for the position in the LCD module is as follows.

```
00 ................ 0F ........................... 27
40 ................ 4F ........................... 67
```

So the valid address for displaying in the display is 00 - 27 (for the first row) or 40 - 67 (for the second row).

Input	:	C	=	00 - 27 (or) 40 - 67
		B	=	ASCII data to be displayed
		CALL 0005		
Output	:	None		
Registered affected	:	A		
Result	:	The LCD initialised to display from the set address location.		

6.2.7 TO DISPLAY AN ASCII CHARACTER (Function 06)

To display an ASCII character in the specified position using function 05.

Input	:	A = 06
		B = ASCII data to be displayed
		CALL 0005
Output	:	None
Registered affected	:	A
Result	:	The ASCII character is displayed in the specified position.

6.2.8 TO DISPLAY A STRING IN THE LCD (FUNCTION 07)

Input	:	A = 07
		HL = Starting address of the data in the memory. Place ASCII value $ (24H) to terminate the string.
		C = Address 00 - 27 (or) 40 - 67 CALL 0005
Output	:	None
Registered affected	:	HL, A
Result	:	The string will be displayed on the LCD screen.

6.2.9 TO GET A STRING INPUT FROM THE IBM- PC KEYBOARD (Function 08)

This function performs like a single line editor and stores the ASCII data entered in the RAM location 4080H. Reserved for monitor usage from 4080H upto locations 40 can be used as string input buffer (i.e.) 4080 - 40A7H. When enter key is pressed this function is terminated and the entered characters ASCII value can be read from 4080A until ASCII character "$" (24H) is encountered.

Input	:	A = 08
		B = Get string location value 00H - 27H (or) 40H - 67H
		CALL 0005
Output	:	From 4080H till data 24H is encountered
Registered affected	:	All registers
Result	:	The input string is stored from 4080H till you encounter 24H data.

6.2.10 TO CONVERT HEX DATA TO ASCII DATA (Function 09)

Input	:	A = 09
		D = Hex data
		CALL 0005
Output	:	D contains the ASCII data
Registered affected	:	A, D
Result	:	D will have the e ASCII value.

6.2.11 TO CONVERT ASCII INPUT FROM A MEMORY ARRAY TO HEX VALUE (Function 0A)

Input	:	A	=	0AH
		HL	=	Starting address of memory array
		B	=	22H, A2H, EEH, 00H
		CALL 0005		

If B = 22H	-	One byte address only. Memory array should be terminated by 24H.
If B = A2H	-	One byte address followed by data First valid two bytes from memory array is used for conversion.
If B = EEH	-	To byte address only. Memory array should be terminated by 24H
If B = 00H	-	Two byte address followed by address / data. First valid four bytes are used for conversion.

Note :

i. This function will ignore preceding blanks (i.e.) ASCII data 20H.

ii. Address error will be reported depending on "A" register initialisation.

iii. For A = 22H and A = EEH, remaining locations must be blank (i.e) 20H else error will be reported.

Output : DE Registers pair will have the Hex value.

D - Higher byte

E - Lower byte

Registers affected : All registers.

6.2.12 TO DISPLAY TWO BYTE ADDRESS IN THE DISPLAY (Function 0B)

Input : A = 0BH

C = Address locations 00 - 27H (or) 40 - 67H

H = Higher byte Hex address

L = Lower byte Hex address

Output : The contents of HL will be displayed at the specified locations.

Registers affected : A, B

Result : The contents of the HL will be displayed.

6.2.13 TO DISPLAY ONE BYTE DATA IN THE LCD (Function 0C)

Input : A = 0CH
C = Address locations
D = Data to be displayed

Output : The contents of HL will be displayed at the specified locations.

Registers affected : A, B, D, E

Result : The contents of the D Register will be displayed.

6.2.14 TO CONVERT TWO CONSECUTIVE ASCII DATA TO HEX DATA (Function 0D)

Input	:	A = 0DH
		HL = Pointer to ASCII data.
		i.e., Two bytes to represent a hex data followed by 24H
		Note : If invalid data is present then data error will be reported.

Output	:	E = Contains the hex data.
Registers affected	:	DE, HL, A
Result	:	The Hex byte equivalent of the ASCII data is given in E registers.

6.2.15 DELAY ROUTINE (Function 0E)

Input	:	A = 0EH DE = contains the delay value CALL 0005

Delay calculation

No. of delay cycles, that can be produced using this function are

63 + 24 (DE contents - 1) clock cycles.

Calculation of Delay - Cycles - Equivalent of required delay

$$\frac{\text{Required Delay}}{1/3\ \mu\text{ sec.}} = \text{No of cycles.}$$

Calculations of DE counter value

63 + 24 (DE Contents - 1) = No. of Delay Cycles = D

24 (DE Contents - 1) = D - 63

DE Contents = $\frac{(D - 63)}{24} + 1$

= Value in Decimal (Convert to Hex)

Output : None

Registers affected : A, DE

Result : Delay is generated depending on the counter value in DE registers

CHAPTER - 7

CONNECTOR DETAILS

7.1 INTRODUCTION

The following is a short list of the connectors available on Micro - Z80. The sections below define more on the pin assignment of the connectors listed below. Each signal referred to below is given a brief description.

P1	-	Power Connector.
P2	-	Casio Socket Connector.
P3,P4	-	Serial port connector.
P5	-	Timer Port Connector.
P6,P7	-	Parallel Port Connector.
P8	-	Interrupt Connector.
P9	-	Serial I/O Connector.
P10	-	VXT Bus Connector.
P11,P12	-	EP Socket Connector.
P13	-	48 Pin EuroConnector.
P14	-	Printer Port Connector.
P15	-	6 Pin PS2 Keyboard Connector.
P16	-	Analog Input & Output Connector.
P17	-	Digital I/O Lines Connector.
P18	-	Interrupt Port Connector.
LCD	-	LCD Connector.

i. The signals VCC and GND have common definitions for all connectors. They are

VCC	=	+5V Power Supply
GND	=	0V Reference GND

ii. Signals with an "*" to its right are active low signals

7.2 POWER CONNECTOR (P1)

Connector Used

Single Row 5 pin male connector
Spacing between pins 2,3,4,5 : 5mm
Spacing between pins 1 and 2 : 7.5mm

Signal Description

Pin	Details
1	GND
2	-12V
3	+12V
4	+30V
5	V_{CC}

7..3POWER CONNECTOR (Casio Socket) (P2)

Connector type and Description

Casio Socket

Signal Description

Pin	Details
CHASIS	GND
SIGNAL	+5V

Mating Connector

Casio Socket female

7.4 SERIAL PORT CONNECTOR (P3,P4)

Connector Used

9 Pin D type male Connector
Pin arranged in two rows of 5 and 4 pins.
Grid pitch 2.76mm × 2.84mm.
The connector is amphenol standard.

SIGNAL DESCRIPTION

Pin	Details
1	NC
2	TxD
3	RxD
4	RTS*
5	CTS*
6	NC
7	GND
8	NC
9	NC

Signal Definition

NC	-	No Connection
TxD	-	Transmit Data
RxD	-	Receive Data
RTS*	-	Request to send
CTS*	-	Clear to send

Mating Connector :

9 positions D female connector with same specifications.

7.5 TIMER PORT CONNECTOR (P5)

Connector Used

10 pin male straight connector
pins arranged in a single row.
pitch = 2.5mm

SIGNAL DESCRIPTION

Pin	Details
1	OUT1*
2	OUT2*
3	NC
4	GATE1*
5	CLK 2
6	CLK1*
7	GATE2*
8	GND
9	VCC
10	NC

** ----- These pins are connected to P6 through a jumper configuration. The jumpers should be closed to use these signals from P6.

Signal Definition

CLK1	-	Clock input of Channel 1
OUT 0	-	Clock output of channel 1
GATE1	-	Gate input of channel 1
CLK 2	-	Clock input of channel 2
OUT 2	.-	Clock output of channel 2
GATE2	-	Gate input of channel 2.

Mating Connector S401 Female straight connector with pitch 2.5mm.

7.6 PARALLEL PORT CONNECTORS (P6, P7)

Connector Used

26 Pin IDC Male Connector
13 Pins arranged in two rows
Pin to Pin pitch distances = 2.54mm.

Signal Description

Pin	Details	Pin	Details
1	PA0	14	PB5
2	PA1	15	PB6
3	PA2	16	PB7
4	PA3	17	PC0
5	PA4	18	PC1
6	PA5	19	PC2
7	PA6	20	PC3
8	PA7	21	PC4
9	PB0	22	PC5
10	PB1	23	PC6
11	PB2	24	PC7
12	PB3	25	GND
13	PB4	26	VCC

Signal Definition

PA0 - PA7	=	PORT A I/O LINES
PB0 - PB7	=	PORT B I /O LINES
PC0 - PC7	=	PORT C I /O LINES

Mating Connector

26 pin FRC dual row connector
Pitch = 2.54mm
The flat cable used should be of pitch 1.27mm.

7.7 INTERRUPT CONNECTOR (P8)

Connector used:

5 Pin Connector
Pins are arranged in a single row.

Signal Definition

PIN	DETAILS
1	INT
2	RST7.5
3	RST6.5
4	RST5.5
5	GND

7.8 SERIAL I/O CONNECTOR (P9)

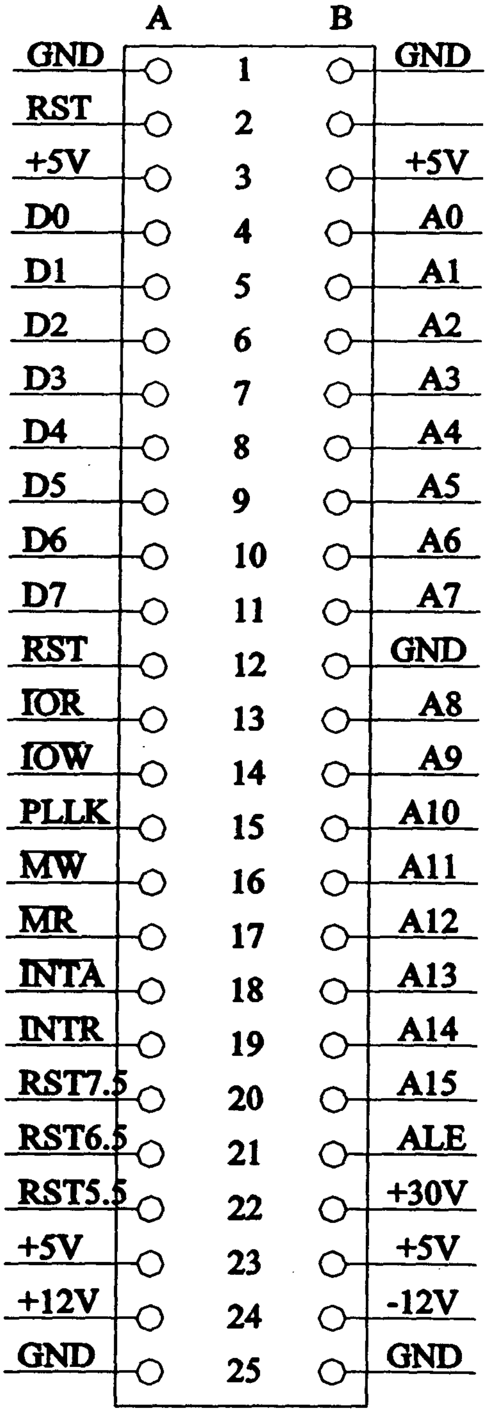
Connector used:

5 Pin Connector
Pins are arranged in a single row.

Signal Definition

PIN	DETAILS
1	GND
2	VCC
3	SOD
4	SID
5	STAT0

7.9 VXT BUS (P10)



	A		B	
GND	○	1	○	GND
RST	○	2	○	
+5V	○	3	○	+5V
D0	○	4	○	A0
D1	○	5	○	A1
D2	○	6	○	A2
D3	○	7	○	A3
D4	○	8	○	A4
D5	○	9	○	A5
D6	○	10	○	A6
D7	○	11	○	A7
$\overline{RST}$	○	12	○	GND
$\overline{IOR}$	○	13	○	A8
$\overline{IOW}$	○	14	○	A9
PLLK	○	15	○	A10
$\overline{MW}$	○	16	○	A11
$\overline{MR}$	○	17	○	A12
$\overline{INTA}$	○	18	○	A13
INTR	○	19	○	A14
RST7.5	○	20	○	A15
RST6.5	○	21	○	ALE
RST5.5	○	22	○	+30V
+5V	○	23	○	+5V
+12V	○	24	○	-12V
GND	○	25	○	GND

7.10 EP SOCKET POWER CONNECTOR (P11, P12)

Connector type and Description

EP Socket

Signal Description

Pin	Details
CHASIS	GND
SIGNAL	+5V

Mating Connector

Audio jack(mono)

7.11 48 PIN EURO CONNECTOR (P13)

7.12 PRINTER PORT CONNECTOR (P14)

Connector type and Description

25 position D female connector
Pins arranged in two rows of 12 and 13 pins.
Grid pitch is 2.76mm × 2.84 mm.
The connector is amphenol standard.

Signal Description

Pin	Details	Pin	Details
1	DATA STROBE*	10	ACK*
2	DATA BIT 0	11	BUSY
3	DATA BIT 1	12	PAPER ERROR
4	DATA BIT 2	13	SELECT
5	DATA BIT 3	14	AUTO*
6	DATA BIT 4	15	ERROR*
7	DATA BIT 5	16	INIT*
8	DATA BIT 6	17	SELECT IN*

9	DATA BIT 7	18 - 25	GND

Signal Definition

Data Strobe* - Strobe Line to printer

Data Bit0 -7 - 8 Data lines to printer

ACK* - Acknowledge from printer cannot receive data.

Paper Error - Paper Error signal from printer.

Select - Indicates printer is in selected state.

Auto* - With this signal being low, the paper is automatically fed one line after printing.

INIT* - Resets printer and clears print buffer.

Select In* - Data entry to printer is possible only if this is low.

Mating Connector

25 Position D male connector

The mating Connector can have a 25 core flat cable.

7.13 PS2 KEYBOARD CONNECTOR (P15)

Pin	Details
1	NC
2	KBDATA
3	KVCC
4	GND
5	NC
6	KBCLK

7.14 ANALOG INPUT & OUTPUT CONNECTOR (P16)

Pin	Details	Pin	Details
1	IN0	7	IN6
2	IN1	8	IN7
3	IN2	9	GND
4	IN3	10	DAC1
5	IN4	11	GND
6	IN5	12	VCC

7.15 DIGITAL I/O LINES CONNECTOR (P17)

Pin	Details	Pin	Details
1	AOUT1	9	EO1
2	AIN1	10	EO2
3	AIN2	11	VCC
4	RP1	12	AGND
5	RP2	13	DGND
6	VEXT	14	NC
7	EI1	15	NC
8	EI2	16	NC

7.16 INTERRUPT PORT (P18)

Pin	Details	Pin	Details
1	IRQ0	6	IRQ5
2	IRQ1	7	IRQ6
3	IRQ2	8	IRQ7
4	IRQ3	9	IRQ8
5	IRQ4	10	IRQ9

IRQ0-IRQ7 = Eight Interrupt Request inputs to 8259.

7.17 LCD CONNECTOR (P19)

Pin	Details	Pin	Details
1	GND	9	D2
2	VCC	10	D3
3	CONTRAST	11	D4
4	RS-	12	D5
5	DIOW-	13	D6
6	LCDSEL	14	D7
7	D0	15	VCC
8	D1		

CHAPTER - 8

FAULT ANALYSIS POINTS

8.1 INTRODUCTION

Trouble - shotting and repair of a µp (i) (8085A/Z80A) trainer involves many intense procedures to be followed. This chapter describes the methods and steps to be followed to maintain a µp (i) (8085A/Z80A) trainer and is peripherals.

The µp (i) (8085A/Z80A) trainer has a different faults analysis points to simulate the various faults, covering individual sections of a µp (i) (8085A/Z80A) trainer. The fault points are,

SW2	-	Permanent wait state logic
SW3	-	Permanent Reset state logic
SW4	-	Disable Read signal logic
SW5	-	Disable write signal logic
SW6	-	Shorts 2 address bus logic
SW7	-	Shorts 2 data lines logic
SW8	-	Shorts address and data lines logic
SW9	-	Stops peripheral clock
SW10	-	Disable 7 - segment data.
SW11 & SW12 -		Disable 7 segment scanning.

The description problem cause the possible location of each fault is explained in detail. The main advantage in this system is the user can introduce the fault in the particular section and then identify the problem. After identification he can solve the problem.

8.1a. SW2 - PERMANENT WAIT STATE

Ready - Ready signal is used to delay the microprocessor internal processing, the microprocessor wait for an integral number clock cycles until it goes high.

Fault - SW2 is connected to GND (down mode) the ready pin is goes to low. the microprocessor stops the execution & wait for ready pin goes to high (up mode)

8.1b. SW3 - PERMANENT RESET STATE

RSTSW - RST signal in Microprocessor goes low, the program counter is set to zero the buses are tri-stated, the microprocessor is reset.

Fault - SW3 is connected to GND (down mode), the RST pin is goes to low. The Microprocessor enter into the reset state, until the reset pin goes high (up mode)

8.1 c. SW4 DISABLE READ SIGNAL

Read - Read signal indicates that peripheral (or) memory device is to be read and datas are available in the data lines.

Fault - SW4 is connected to NC (down mode), Read signal does not go to the peripherals, so that Read process will not be Read the datas from peripherals.

8.1d SW5 - DISABLE WRITE SIGNAL

Write - Write signal indicates that the data on the data bus to be written into the selected memory peripherals.

Fault - SW5 is connected to Nc (up mode), the write signal process will not be write the data in selected memory (or) peripheral.

8.1e SW6 - SHORTS 2 ADDRESS BUS

Write - Address bus is a group of lines that are used to send a memory address or a device address from Microprocessor to the memory location or the peripheral. These lines are used to identify a selected memory (or) peripherals.

Fault - SW6 is connected in up mode, the two address lines are shorted. So the Microprocessor identify the wrong memory location, the basic function will be affected.

8.1f SW7 - SHORTS 2 DATA LINES

The data bus is a group of lines used for data flow. These lines are bidirectional, data flow in both directions between Microprocessor and peripheral devices.

Fault - SW7 is connected in up mode, the data lines are shorted. The Read data from the peripherals (or) selected memory is wrong process. So Microprocessor execution is done in wrong process.

8.1g SW8 SHORTS ADDRESS AND DATA LINES

The address bus identify the selected memory (or) peripheral. The data bus write a data to the memory (or) Read the data from the memory.

Fault - SW8 is connected to up mode, the address line and data line is shorted. So up identify the wrong memory location (or) peripherals at same times the data to be write (or) Read is wrong.

8.1h SW9 - STOPS PERIPHERAL CLOCK

Clk output signal can be used as system clock for other device.

Fault - The clock output is connected to down mode, the pclk will not go to the other peripherals. The peripherals is not working.

8.1i SW10 - DISABLE 7 SEGMENT DATA

The 7 segment display is used to display the address & data lines

Fault - SW10 is connected to down, one display is not working

8.1j SW11 & SW12 - DISABLE 7 SEGMENT SCANNING

SW11 & SW12 is connected to down, the segment is not closing in all display.

APPENDIX - A

8085A DATASHEETS

8085A/8085A-2
SINGLE CHIP 8-BIT N-CHANNEL MICROPROCESSOR

- Single +5V Power Supply
- 100% Software Compatible with 8080A
- 1.3uS Instruciont Cycle (8085A); 0.8uS (8085A-2)
- On-Chip Clock Generator (With External Crystal, LC or RC Network)
- On-Chip System Controller; Advanced Cycle Status Information Available for Large System Control
- Four Vectored Interrupt Inputs (one is Non-Maskable) Plus an 8080A-Compatible Interrupt.
- Serial In/Serial Out Port
- Decimal, Binary and Double Precision Arithmetic
- Direct Addressing Capability to 64K Bytes of Memory

The Intel 8085A is a complete 8 bit parallel Central Processing Unit (CPU). Its instruction set is 100% software compatible with the 8080A microprocessor, and it is designed to improve the present 8080A's performance by higher system speed.

Its high level of system integration allows a minimum system of three IC's (8085A (CPU), 8156 (RAM/IO) and 8355/8755A (ROM/PROM/IO) while maintaining total system expandability. The 8085A-2 is a faster version of the 8085A

The 8085 incorporates all of the features that the 8224 (clock generator) and 8228 (System controller) provede for the 8080A, Thereby offering a high level of system integration.

The 8085A uses a multiplexed data bus. The address is split between the 8 bit address bus and the 8 bit data bus. The on-chip address latches of 8155/8156/8755A memory products allow a direct interface with the 8085A.

Table 1. Pin Description

A8-A15	O	**Address Bus:** The most significan 8 bits of the memory address or the 8 bits of the I/O address, 3-stated during Hold and Halt modes and during RESET
AD0-AD7	I/O	**Multiplexed Address/Data Bus:** Lower 8 bits of the memory address (or I/O address) appear on the bus furing the first clock cycle (T state) of a machine Cycle. It then becomes the data bus furing the second and third clock cycles.
ALE	O	**Address Latch Enable:** It occurs during the first clock state of a machine cycle and enables the address to get latched into the on-chip latch of peripherals. The falling edge of ALE is set to guarantee setup and hold times for the address information. The falling edge of ALE can also be used to strobe the status information. ALE is never 3-stated.
S0,S1 and IO/M	O	**Machine Cycle Status:**

IO/M	S1	S0	Status
0	0	1	Memory Write
0	1	0	Memory Read
1	0	1	I/O Write
1	1	0	I/O Read
0	1	1	Opcode fetch
1	1	1	Opcode fetch
1	1	1	Interrupt Acknlo.
*	0	0	Halt

*	X	X	Hold
*	X	X	Reset

* = 3-state (high impedance)
X= unspecified

S1 can be used as an advanced R/W status. IO/M, S0 and s1 become valid at the beginning of a machine cycle and remain stable throughout the cycle. The falling edge of ALE may be used to latch the state of these lines.

RD	O	**Read Control:**A low level on RD indicates the selected memory or I/O device is to be read and that the Data Bus is available for the ata transfer, 3-stated during Hold and Halt modes and during RESET.
WR	O	**Write Control:** A low level on WR indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of WR. 3-stated during Hold and Halt modes and during RESET.
READY	I	**Ready:** If READY is high during a read or write cycle, it indicates that the momory or peripheral is ready to send or receive data. If READY is low, the cpu will wait an integral number of clock cycles for READY to go high before completing the read or write cycle. READY must conform to specified setup and hold times.
HOLD	I	**Hold:** Indicates that another master is requesting the use of the address and data buses. The cpu, upon receiving the hold request, will relinquish the use of the bus as soon as the completion of the current bus transfer. Internal processing can continue. The processor can ragain the bus only after the HOLD is remobed. When the HOLD is acknowledged, the Address, Data, RD, WR , and IO/M lines are 3-stated
HLDA	O	**Hold Acknowledge:** Indicates that the cpu has received the HOLD request and that it will relinquish the bus in the next clock cycle. HLDA goes low after the Hold request is removed. The Cpu tekes the bus one half clock cycle after HLDA goes low.
INTR	I	**Interrupt request:** Is used as a general purpose Interrupt. It is sampled only during the next to the last clock cycle of an instruction and during the next to the last clock cycle of an instrucion and during HOld and Halt states. IF it is active, the Program Counter (PC) will be inhibited from incrementing and an INTA will be issued. During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routing. The INTR is enabled and disabled by software. It is isabled by Reset and immediately after an interrupt is accepted.
INTA	O	**Interrupt Acknowledge:** Is used instead of (and has the same timing as) RD during the instruction cycle after an INTR is accepted. It can be used to activate an 8259A Interrupt chip or some other interrupt port.
RST 5.5 RST 6.5 RST 7.5	I	**Restart Interrupts:** These three intputs have the same timings as INTR except they cause an internal RESTART to be automatically inserted. The priority of these interrupts is ordered as shown in Table 2. These interrupts have a higher priority then INTR, In addition, they may be individually masked out using the SIM instruction.
TRAP	I	**Trap:** Trap interrupt is a non-maskable RESTART interrupt. It is recognized at the same time as INTR or RST5.5-7.5. It is unaffecte by any mask or interrupt Enable. It has the highest priority of any interrupt. (see Table 2.)
RESET IN	I	**Reset In:** Sets the Program Counter to zero and resets the interrupt Enable and HOLD flip-flops. The data and adress buses and the control lines are 3-stated during RESET and because of the asynchronous nature of RESET, the processor's internal registers and flags may be altered by RESET with unpredictable results. RESET IN is a Schmitt-triggered input, allowing connection to an R-C network for power-on RESET delay. The cpu is held in the reset condition as long as RESET IN is applied.
RESET OUT	O	**Reset Out:** Reset Out indicates cpu is being reset. Can be used as a system reset. The signal is synchronized to the processor clock and lasts an integral number of clock periods.
X1,X2	I	**X1 and X2:** Are connected to a crystal, LC or RC network to drive the internal clock

		generator. X1 can also be an external clock input from a logic gate. The input frequency is divided by 2 to give the processor's internal operating frequency.
CLK	O	**Clock:** Clock output for use as a system clock. The period of CLK is twice the X1,X2 input period.
SID	I	**Serial Input Data Line:** The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed
SOD	O	**Serial Output Data Line:** The putput SOD is set or reset as specified by the SIM instruction.
VCC		**Power:** +5 volt supply.
VSS		**Ground:** Reference

Table 2. Interrupt Priority, Restart Address, and Sensitivity

Name	Priority	Address Branched To (1) When Interrupt Occurs	Type Trigger
TRAP	1	24H	Rising edge AND high level until sampled.
RST 7.5	2	3CH	Rising edge (latched).
RST 6.5	3	34H	High level until sampled.
RST 5.5	4	2CH	High level until sampled.
INTR	5	See Note (2).	High level until sampled.

NOTES:

1. the processor pushes the PC on the stack before branching to the indicated address.
2. The address branched to depends on the instruction provided to the cpu when the interrupt is acknowledged.

FUNCTIONAL DESCRIPTION

The 8085A is a complete 8-bit parallel central processor. It is designed with N-channel depletion loads and requires a single +5 volt supply. Its basic clock speed is 3Mhz (8085A) or 5MHZ (8085A-2), thus improving on the present 8080A's performance with hight system speed. Also it is designed to fit into a minimum system of three IC's: The cpu (8085A), a RAM/IO (8156), and a ROM or EPROM /IO chip (8355 or 8755A).

The 8085A has twelve addressable 8-bit registers. our of them can funtion only as two 16-bit register pairs. Six other can be used interchangeably as a 8-bit register or a 16-bit register pairs. The 8085A register set is as follows:

Mnemonic	Register	Contents
ACC or A	Accumulator	8 bits
PC	Program Counter	16-bit address
BC, DE, HL	General-Purpose Registers; data pointer (HL)	8 bits x 6 or 16 bits x 3
SP	Stack Pointer	16-bit address
Flags or F	Flag Register	5 flags (8 bit space)

The 8085A uses a multiplexed Data Bus. The address is split between the higher 8-bit Address Bus and the lower 8-bit Address/Data Bus. During the first T state (clock cycle) of a machine cycle the low order address is sent out on the Address/Data bus. These lower 8 bits may be latched externally by the Address Latch Enable signal (ALE). During the rest of the machine cycle the data bus is used for memory or I/O data.

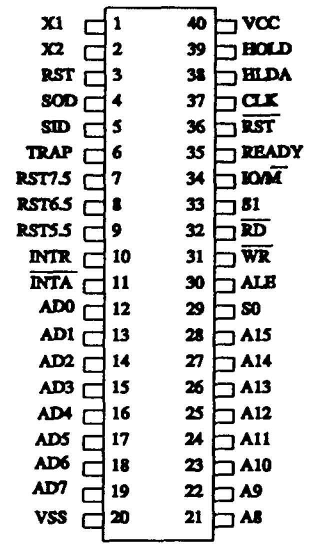
The 8085A provides **RW, WR** S0, S1 and IO/**M** signals for bus control. An interrupt acknowledge signal (**INTA**) is also provided. HOLD and all interrupts are synchronized with the processor's internal clock. The 8085A also provides serial Input Data (SID) and Serial Output Data (SOD) lines for simple serial interface.

In addition to these features, the 8085A has three maskable, vector interrupts pins and on non-maskable TRAP interrupt. ");.//-->

APPENDIX - B

IC PINOUTS

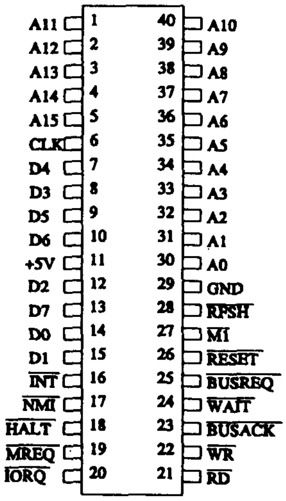
8085 8-BIT MICROPROCESSOR



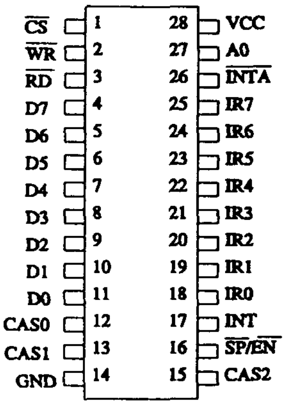
SIGNAL	INPUT/ OUTPUT	DESCRIPTION
A8 - A15	O	ADDRESS BUS
AD0 - AD7	I/O	ADDRESS/DATABUS
ALE	O	ADRESS LATCH ENABLE
S0 ,S1	O	STATUS LINES
IO/$\overline{\text{M}}$	O	INPUT OUTPUT/MEMORY
$\overline{\text{RD}}$	O	READ CONTROL
$\overline{\text{WR}}$	O	WRITE CONTROL
READY	I	READY
HOLD	I	HOLD
HLDA	O	HOLD ACKNOWLEDGE
INTR	I	INTERRUPT REQUEST
$\overline{\text{INTA}}$	O	INTERRUPT ACKNOWLEDGE
RST5.5		
RST6.5	I	RESTART INTERRUPTS
RST7.5		
TRAP	I	TRAP
$\overline{\text{RST}}$	I	RESET IN
RST	O	RESET OUT
X1 ,X2	I	CRYSTAL INPUTS
CLK	O	CLOCK OUTPUT
SID	I	SERIAL INPUT DATA LINE
SOD	O	SERIAL OUTPUT DATA LINE
VCC		+5V POWER SUPPLY
GND		GND

Z80 8-BIT MICROPROCESSOR

SIGNAL	INPUT/ OUTPUT	DESCRIPTION
A0-A15	O	ADDRESS BUS
$\overline{\text{BUSACK}}$	O	BUSS ACKNOWLEDGE
$\overline{\text{BUSREQ}}$	I	BUS REQUEST
D0-D7	I/O	DATA BUS
$\overline{\text{HALT}}$	I	HALT STATE
$\overline{\text{INT}}$	O	INTERRUPT REQUEST
$\overline{\text{IORQ}}$	O	INPUT / OUTPUT REQUEST
$\overline{\text{M1}}$	O	MACHINE CYCLE ONE
$\overline{\text{MREQ}}$	O	MEMORY REQUEST
$\overline{\text{NMI}}$	I	NON-MASKABLE INTERRUPT
$\overline{\text{RD}}$	O	READ
$\overline{\text{RESET}}$	I	RESET
$\overline{\text{RFSH}}$	O	REFRESH
$\overline{\text{WAIT}}$	I	WAIT
$\overline{\text{WR}}$	O	WRITE
CLK	I	CLOCK INPUT
VCC		+5V POWER SUPPLY
GND		GROUND

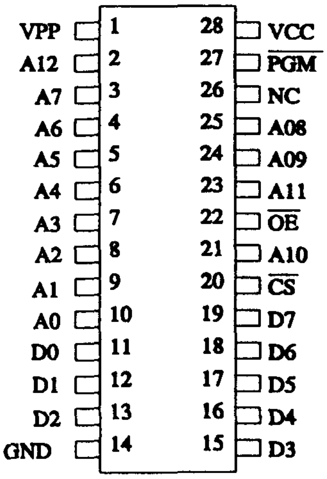

A11 1
A12 2
A13 3
A14 4
A15 5
CLK 6
D4 7
D3 8
D5 9
D6 10
+5V 11
D2 12
D7 13
D0 14
D1 15
$\overline{\text{INT}}$ 16
$\overline{\text{NMI}}$ 17
$\overline{\text{HALT}}$ 18
$\overline{\text{MREQ}}$ 19
$\overline{\text{IORQ}}$ 20
40 A10
39 A9
38 A8
37 A7
36 A6
35 A5
34 A4
33 A3
32 A2
31 A1
30 A0
29 GND
28 $\overline{\text{RFSH}}$
27 $\overline{\text{M1}}$
26 $\overline{\text{RESET}}$
25 $\overline{\text{BUSREQ}}$
24 $\overline{\text{WAIT}}$
23 $\overline{\text{BUSACK}}$
22 $\overline{\text{WR}}$
21 $\overline{\text{RD}}$

8259-PROGRAMMABLE PRIORITY CONTROLLER



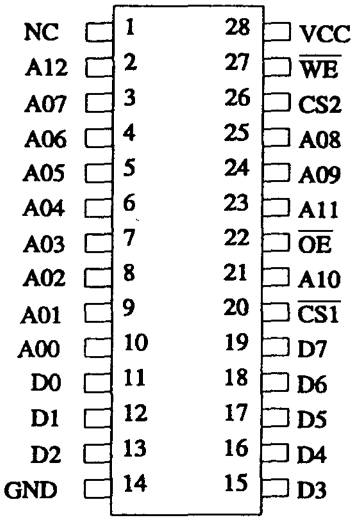
SIGNAL	INPUT/ OUTPUT	DESCRIPTION
D7 - D0	I/O	BIDIRECTIONAL 8-BIT DATA BUS
$\overline{CS}$	I	CHIP SELECT
$\overline{RD}$	I	READ OUT PUT
$\overline{WR}$	I	WRITE INPUT
INT	O	INERRUPT OUTPUT INTERRUPTS THE PROCESSOR IF ANY OF ITS INPUT LINES ARE HIGH
IR0 - IR7	I	8-INTERRUPT REQUSTS TO 8259
$\overline{INTA}$	I	INTERRUPT ACKNOWLEDGE FROM PROCESSOR
A0	I	ADDRESS LINE
CAS0 CAS2	I/O	CASCADE LINES-OUTPUTS FOR MASTER 8259 & INPUT FOR SLAVE 8259
$\overline{SP}/\overline{EN}$	I/O	SLAVE PROGRAM / ENABLE BUFFERS IN BUFFERED MODE CONTROLS BUFFER TRANSCEIVERS (EN) $\overline{ELSE}$ USED AS INPUT TO DESIGNATE & MASTER (SP=1) OR A SLSVE (SO=0)
VCC		+5V POWER SUPPLY
GND		GROUND

2764 - 8K UVPROM



SIGNAL	DESCRIPTION
A0 - A12	ADDRESS BUS
D0 - D7	DATA BUS
$\overline{OE}$	OUTPUT ENABLE
$\overline{CS}$	CHIP SELECT
$\overline{PGM}$	PROGRAM
VPP	PROGRAMMING VOLTAGE
NC	NO CONNECTION
VCC	+5V POWER SUPPLY
GND	GROUND

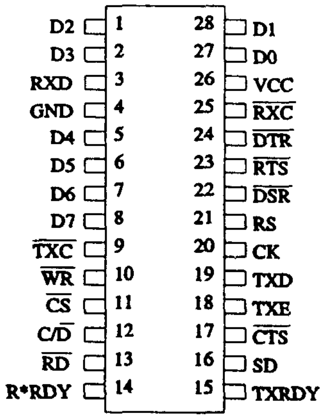
6264 - 8K STATIC RAM



SIGNAL	DESCRIPTION
A0 - A12	ADDRESS BUS
D0 - D7	DATA BUS
$\overline{RD}$	READ
$\overline{WR}$	WRITE
$\overline{CS}$	CHIP SELECT
NC	NO CONNECTION
VCC	+5V POWER SUPPLY
GND	GROUND

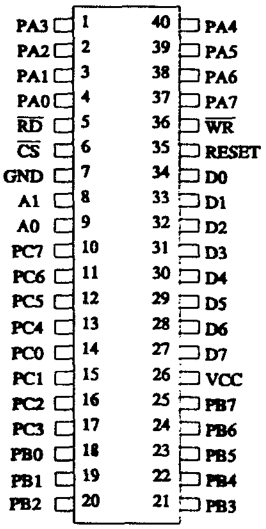
8251 - PROGRAMMABLE COMMUNICATION

SIGNAL	INPUT/ OUTPUT	DESCRIPTION
D0 - D7	I/O	DATA BUS
RESET	I	RESET
CLK	I	CLOCK INPUT
C/$\overline{D}$	I	CONTROL / DATA
$\overline{RD}$	I	READ
$\overline{WR}$	L	WRITE
$\overline{CS}$	I	CHIP SELECT
$\overline{DSR}$	I	DATA SET READY
$\overline{DTR}$	O	DATA TERMINAL READY
$\overline{CTS}$	I	CLEAR TO SEND
$\overline{RTS}$	O	REQUEST TO SEND
TXD	O	TRANSMIT DATA
TXRDY	O	TRANSMITTER READY
TXE	O	TRANSMITTER EMPTY
$\overline{TXC}$	I	TRANSMITTER CLOCK
TXC	I	RECEIVE DATA
RXRDY	O	RECEIVE READY
$\overline{RXC}$	I	REEIVE CLOCK
SYNDET	I/O	SYNC / BREAK DEFECT

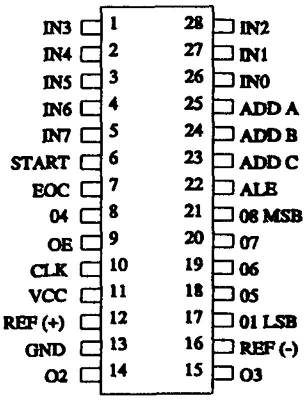


8255 - PROGRAMMABLE PERIPHERAL INTERFACE

SIGNAL	INPUT/ OUTPUT	DESCRIPTION
D0 - D7	I/O	DATA BUS
$\overline{\text{WR}}$	I	WRITE
$\overline{\text{RD}}$	I	READ
$\overline{\text{CS}}$	I	CHIP SELECT
RESET	I	RESET
A0 ,A1	I	PORT SELECT 0 PORT SELECT 1
PA0 - PA7	I/O	PORT A
PB0 - PB7	I/O	PORT B
PC0 - PC7	I/O	PORT C
VCC		+5V POWER SUPPLY
GND		GROUND



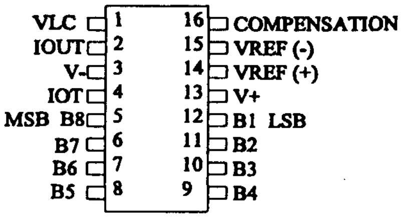
ADC0809 - ANALOG TO DIGITAL CONVERTER



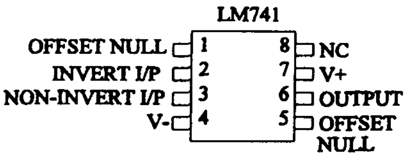
SIGNAL	INPUT/ OUTPUT	DESCRIPTION
IN0 - IN7	I	8 INPUT CHANNELS
01 - 08	O	DIGITAL OUTPUT LINES
ADD A ADD B ADD C	I	CHANNEL SELECTION
START	I	START OF CONVERSION
EOC	O	END OF CONVERSION
OE	I	OUTPUT ENABLE
CLK	I	CLOCK
ALE	I	ADDRESS SELECTION
REF (+) REF (-)	I	REFERENCE VOLTAGES
VCC		+5V POWER SUPPLY
GND		GROUND

DAC0800 - DIGITAL TO ANALOG CONVERTER

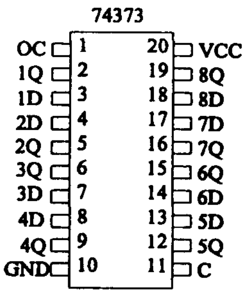
SIGNAL	INPUT/ OUTPUT	DESCRIPTION
B1 -B8	I	DIGITAL INPUTS
VLC	I	THRESHOLD CONTROL
VREF (-) VREF(+)	I	REFERENCE VOLTAGES
IOUT IOUT	O	COMPLEMENTRY CURRENT OUTPUTS
V+		+12V POWER SUPPLY
V-		-12V POWER SUPPLY.



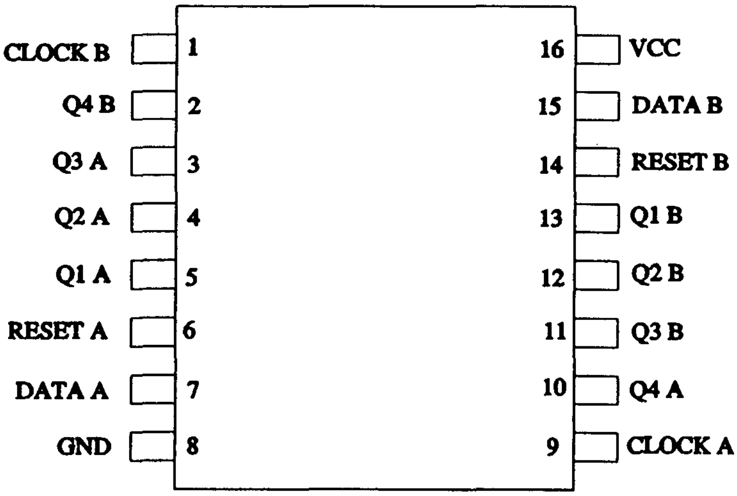
OPERATIONAL AMPLIFIERS



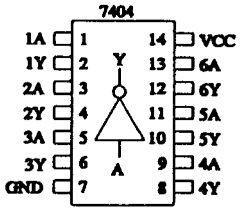
OCTAL D LATCHES



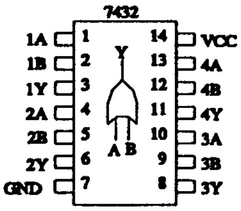
4015 - Shift Register



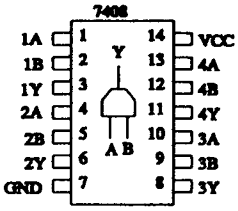
HEX INVERTOR



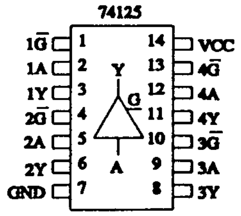
QUAD 2 I/P OR GATE



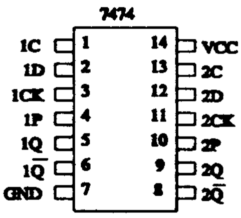
QUAD 2 I/P AND GATE



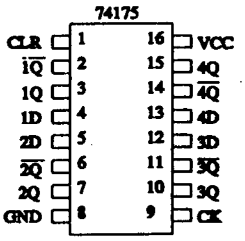
QUAD BUS BUFFER



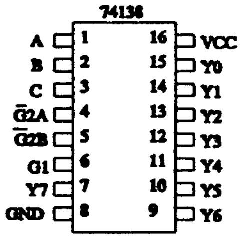
DUAL D FLIP-FLOP



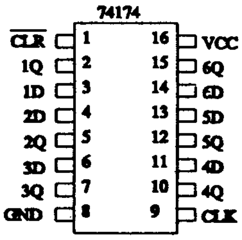
QUAD D FLIP-FLOP



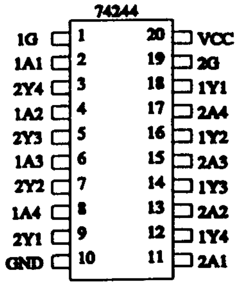
3 TO 8 LINE DECODER



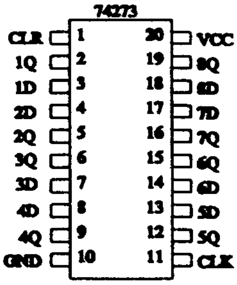
HEX D FLIP-FLOP



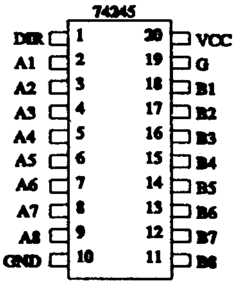
OCTAL BUFFER



OCTAL D FLIP-FLOP

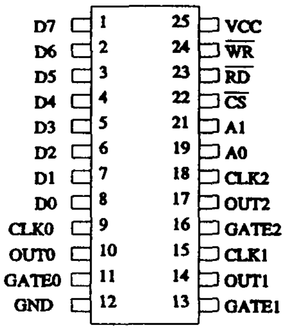


OCTAL BUS TRANSCEIVERS



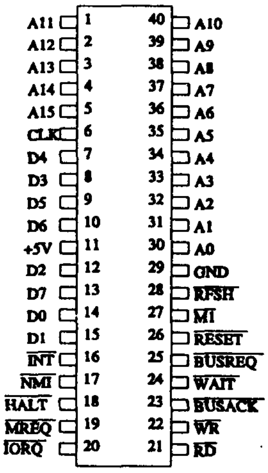
8253 - PROGRAMMABLE INTERVAL

SIGNAL	INPUT/ OUTPUT	DESCRIPTION
D0-D7	I/O	DATA BUS
$\overline{\text{WR}}$	O	WRITE
$\overline{\text{RD}}$	O	READ
$\overline{\text{CS}}$	I	CHIP SELECT
A0 ,A1	I	COUNTER SELECT INPUTS
CLK0	I	CLOCK INPUTS FOR 3 CHANNELS
CLK0		
CLK0		
OUT0	O	CLOCK OUTPUTS FOR 3 CHANNELS
OUT0		
OUT0		
GATE0		
GATE0	I	GATE NPUTS FOR 3 CHANNELS
GATE0		
GATE0		
VCC		+5V POWER SUPPLY
GND		GROUND



Z80 8-BIT MICROPROCESSOR

SIGNAL	INPUT/ OUTPUT	DESCRIPTION
A0-A15	O	ADDRESS BUS
$\overline{\text{BUSACK}}$	O	BUSS ACKNOWLEDGE
$\overline{\text{BUSREQ}}$	I	BUS REQUEST
D0-D7	I/O	DATA BUS
$\overline{\text{HALT}}$	I	HALT STATE
$\overline{\text{INT}}$	O	INTERRUPT REQUEST
$\overline{\text{IORQ}}$	O	INPUT / OUTPUT REQUEST
$\overline{\text{M1}}$	O	MACHINE CYCLE ONE
$\overline{\text{MREQ}}$	O	MEMORY REQUEST
$\overline{\text{NMI}}$	I	NON-MASKABLE INTERRUPT
$\overline{\text{RD}}$	O	READ
$\overline{\text{RESET}}$	I	RESET
$\overline{\text{RFSH}}$	O	REFRESH
$\overline{\text{WAIT}}$	I	WAIT
$\overline{\text{WR}}$	O	WRITE
CLK	I	CLOCK INPUT
VCC		+5V POWER SUPPLY
GND		GROUND


A11 1
A12 2
A13 3
A14 4
A15 5
CLK 6
D4 7
D3 8
D5 9
D6 10
+5V 11
D2 12
D7 13
D0 14
D1 15
$\overline{\text{INT}}$ 16
$\overline{\text{NMI}}$ 17
$\overline{\text{HALT}}$ 18
$\overline{\text{MREQ}}$ 19
$\overline{\text{IORQ}}$ 20
40 A10
39 A9
38 A8
37 A7
36 A6
35 A5
34 A4
33 A3
32 A2
31 A1
30 A0
29 GND
28 $\overline{\text{RFSH}}$
27 $\overline{\text{M1}}$
26 $\overline{\text{RESET}}$
25 $\overline{\text{BUSREQ}}$
24 $\overline{\text{WAIT}}$
23 $\overline{\text{BUSACK}}$
22 $\overline{\text{WR}}$
21 $\overline{\text{RD}}$

APPENDIX - C

8085A INSTRUCTION SET

The 8085 Instruction Set

As we promised, in an earlier lesson, we are going to go through an in-depth explanations of ALL the 8085 instructions.

```
|         88888     000       88888    5555555      A          |
|        8     8    0   0    8     8   5           A A         |
|        8     8   0   0 0   8     8   5          A   A        |
|         88888   0  0  0     88888    555555   AAAAAAA       |
|        8     8   0 0   0   8     8         5   A     A       |
|        8     8    0   0    8     8         5   A     A       |
|         88888     000       88888    555555   A     A       |
|         8085A MICROPROCESSOR Instruction Set Summary         |
|                                                              |
|                    _________    _________                    |
|                  _|         \__/         |_                  |
|          --> X1 |_|1                   40|_| Vcc (+5V)       |
|                  _|                      |_                  |
|          --> X2 |_|2                   39|_| HOLD <--        |
|                  _|                      |_                  |
|   <-- RESET OUT |_|3                   38|_| HLDA -->        |
|                  _|                      |_                  |
|         <-- SOD |_|4                   37|_| CLK (OUT) -->   |
|                  _|                      |_  ________        |
|         --> SID |_|5                   36|_| RESET IN <--    |
|                  _|                      |_                  |
|        --> TRAP |_|6                   35|_| READY <--       |
|                  _|                      |_     _            |
|     --> RST 7.5 |_|7                   34|_| IO/M -->        |
|                  _|                      |_                  |
|     --> RST 6.5 |_|8                   33|_| S1 -->          |
|                  _|                      |_  __              |
|     --> RST 5.5 |_|9                   32|_| RD -->          |
|                  _|                      |_  __              |
|        --> INTR |_|10      8085A       31|_| WR -->          |
|            ____  _|                      |_                  |
|        <-- INTA |_|11                  30|_| ALE -->         |
|                  _|                      |_                  |
|        <--> AD0 |_|12                  29|_| S0 -->          |
|                  _|                      |_                  |
|        <--> AD1 |_|13                  28|_| A15 -->         |
|                  _|                      |_                  |
|        <--> AD2 |_|14                  27|_| A14 -->         |
|                  _|                      |_                  |
|        <--> AD3 |_|15                  26|_| A13 -->         |
|                  _|                      |_                  |
|        <--> AD4 |_|16                  25|_| A12 -->         |
|                  _|                      |_                  |
|        <--> AD5 |_|17                  24|_| A11 -->         |
|                  _|                      |_                  |
|        <--> AD6 |_|18                  23|_| A10 -->         |
|                  _|                      |_                  |
|        <--> AD7 |_|19                  22|_| A9 -->          |
|                  _|                      |_                  |
|       (Gnd) Vss |_|20                  21|_| A8 -->          |
|                   |______________________|                   |
```

Data Transfer Croup:

The data transfer instructions move data between registers or between memory and registers.

MOV	Move
MVI	Move Immediate
LDA	Load Accumulator Directly from Memory
STA	Store Accumulator Directly in Memory
LHLD	Load H & L Registers Directly from Memory
SHLD	Store H & L Registers Directly in Memory

An 'X' in the name of a data transfer instruction implies that it deals with a register pair (16-bits);

LXI	Load Register Pair with Immediate data
LDAX	Load Accumulator from Address in Register Pair
STAX	Store Accumulator in Address in Register Pair
XCHG	Exchange H & L with D & E
XTHL	Exchange Top of Stack with H & L

Arithmetic Group:

The arithmetic instructions add, subtract, increment, or decrement data in registers or memory.

ADD	Add to Accumulator
ADI	Add Immediate Data to Accumulator
ADC	Add to Accumulator Using Carry Flag
ACI	Add Immediate data to Accumulator Using Carry
SUB	Subtract from Accumulator
SUI	Subtract Immediate Data from Accumulator
SBB	Subtract from Accumulator Using Borrow (Carry) Flag
SBI	Subtract Immediate from Accumulator Using Borrow(Carry) Flag
INR	Increment Specified Byte by One
DCR	Decrement Specified Byte by One
INX	Increment Register Pair by One
DCX	Decrement Register Pair by One
DAD	Double Register Add; Add Content of Register Pair to H & L Register Pair

Logical Group:

This group performs logical (Boolean) operations on data in registers and memory and on condition flags.

The logical AND, OR, and Exclusive OR instructions enable you to set specific bits in the accumulator ON or OFF.

ANA	Logical AND with Accumulator
ANI	Logical AND with Accumulator Using Immediate Data
ORA	Logical OR with Accumulator
OR	Logical OR with Accumulator Using Immediate Data
XRA	Exclusive Logical OR with Accumulator
XRI	Exclusive OR Using Immediate Data

The Compare instructions compare the content of an 8-bit value with the contents of the accumulator;

CMP	Compare
CPI	Compare Using Immediate Data

The rotate instructions shift the contents of the accumulator one bit position to the left or right:

RLC	Rotate Accumulator Left
RRC	Rotate Accumulator Right
RAL	Rotate Left Through Carry
RAR	Rotate Right Through Carry

Complement and carry flag instructions:

CMA	Complement Accumulator
CMC	Complement Carry Flag
STC	Set Carry Flag

Branch Group:

The branching instructions alter normal sequential program flow, either unconditionally or conditionally. The unconditional branching instructions are as follows:

JMP	Jump
CALL	Call
RET	Return

Conditional branching instructions examine the status of one of four condition flags to determine whether the specified branch is to be executed. The conditions that may be specified are as follows:

NZ	Not Zero (Z = 0)
Z	Zero (Z = 1)
NC	No Carry (C = 0)
C	Carry (C = 1)
PO	Parity Odd (P = 0)
PE	Parity Even (P= 1)
P	Plus (S = 0)
M	Minus (S = 1)

Thus, the conditional branching instructions are specified as follows:

Jumps	Calls	Returns	
C	CC	RC	(Carry)
INC	CNC	RNC	(No Carry)
JZ	CZ	RZ	(Zero)
JNZ	CNZ	RNZ	(Not Zero)
JP	CP	RP	(Plus)
JM	CM	RM	(Minus)
JPE	CPE	RPE	(Parity Even)
JPO	CPO	RPO	(Parity Odd)

Two other instructions can affect a branch by replacing the contents or the program counter:

PCHL	Move H & L to Program Counter
RST	Special Restart Instruction Used with Interrupts

Stack I/O, and Machine Control Instructions:
The following instructions affect the Stack and/or Stack Pointer:

PUSH	Push Two bytes of Data onto the Stack
POP	Pop Two Bytes of Data off the Stack
XTHL	Exchange Top of Stack with H & L
SPHL	Move content of H & L to Stack Pointer

The I/0 instructions are as follows:

IN	Initiate Input Operation
OUT	Initiate Output Operation

The Machine Control instructions are as follows:

EI	Enable Interrupt System
DI	Disable Interrupt System
HLT	Halt
NOP	No Operation

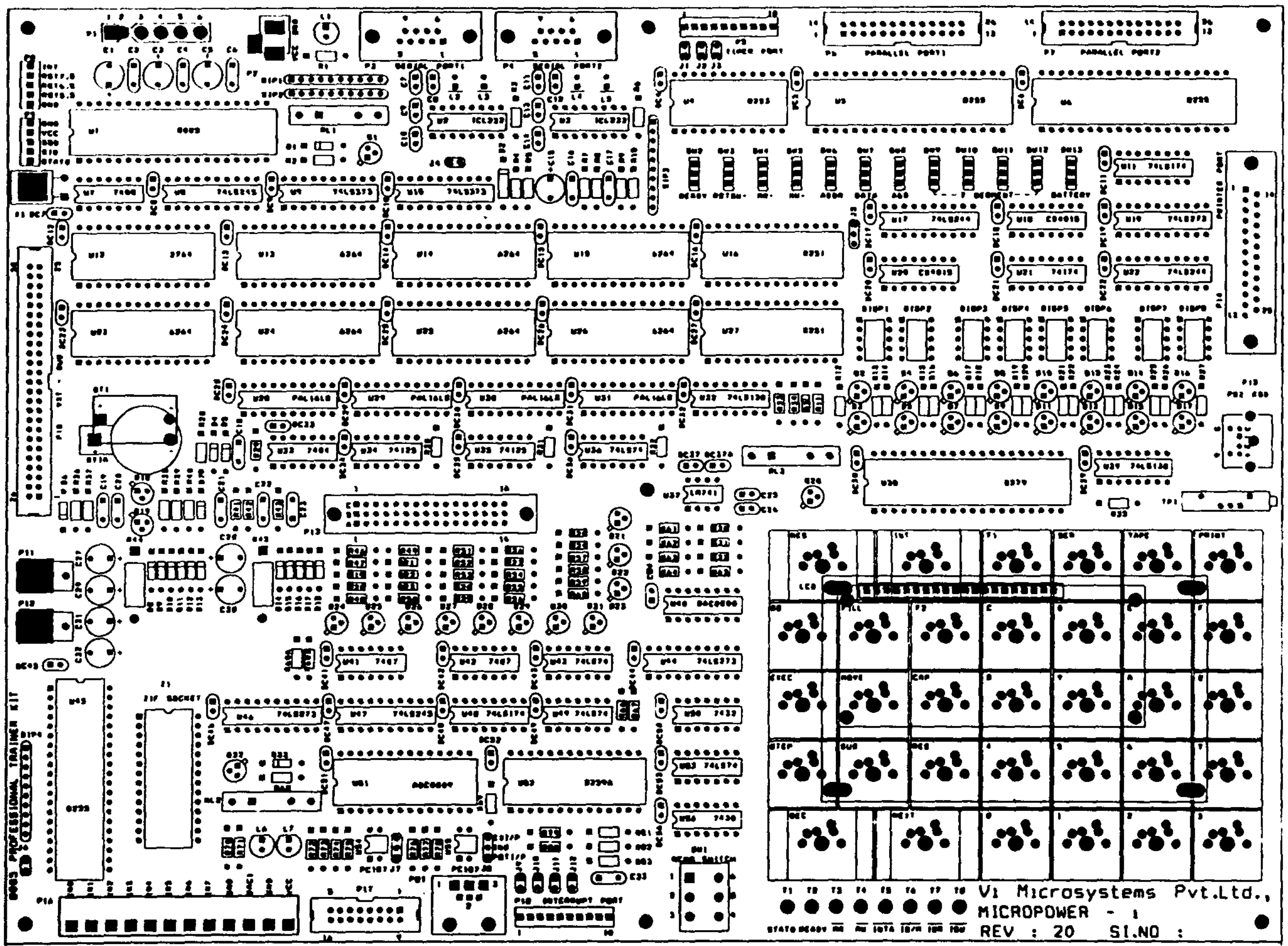
Mnemonic	Op	SZAPC	~s	Description	Notes
ACI n	CE	*****	7	Add with Carry Immediate	A=A+n+CY
ADC r	8F	*****	4	Add with Carry	A=A+r+CY(21X)
ADC M	8E	*****	7	Add with Carry to Memory	A=A+[HL]+CY
ADD r	87	*****	4	Add	A=A+r (20X)
ADD M	86	*****	7	Add to Memory	A=A+[HL]
ADI n	C6	*****	7	Add Immediate	A=A+n
ANA r	A7	****0	4	AND Accumulator	A=A&r (24X)
ANA M	A6	****0	7	AND Accumulator and Memory	A=A&[HL]
ANI n	E6	**0*0	7	AND Immediate	A=A&n
CALL a	CD	-----	18	Call unconditional	-[SP]=PC,PC=a
CC a	DC	-----	9	Call on Carry	If CY=1(18~s)
CM a	FC	-----	9	Call on Minus	If S=1 (18~s)
CMA	2F	-----	4	Complement Accumulator	A=~A
CMC	3F	----*	4	Complement Carry	CY=~CY
CMP r	BF	*****	4	Compare	A-r (27X)
CMP M	BF	*****	7	Compare with Memory	A-[HL]
CNC a	D4	-----	9	Call on No Carry	If CY=0(18~s)
CNZ a	C4	-----	9	Call on No Zero	If Z=0 (18~s)
CP a	F4	-----	9	Call on Plus	If S=0 (18~s)
CPE a	EC	-----	9	Call on Parity Even	If P=1 (18~s)
CPI n	FE	*****	7	Compare Immediate	A-n
CPO a	E4	-----	9	Call on Parity Odd	If P=0 (18~s)
CZ a	CC	-----	9	Call on Zero	If Z=1 (18~s)
DAA	27	*****	4	Decimal Adjust Accumulator	A=BCD format
DAD B	09	----*	10	Double Add BC to HL	HL=HL+BC
DAD D	19	----*	10	Double Add DE to HL	HL=HL+DE
DAD H	29	----*	10	Double Add HL to HL	HL=HL+HL
DAD SP	39	----*	10	Double Add SP to HL	HL=HL+SP
DCR r	3D	****-	4	Decrement	r=r-1 (0X5)
DCR M	35	****-	10	Decrement Memory	[HL]=[HL]-1
DCX B	0B	-----	6	Decrement BC	BC=BC-1
DCX D	1B	-----	6	Decrement DE	DE=DE-1
DCX H	2B	-----	6	Decrement HL	HL=HL-1
DCX SP	3B	-----	6	Decrement Stack Pointer	SP=SP-1
DI	F3	-----	4	Disable Interrupts	
EI	FB	-----	4	Enable Interrupts	
HLT	76	-----	5	Halt	
IN p	DB	-----	10	Input	A=[p]
INR r	3C	****-	4	Increment	r=r+1 (0X4)
INR M	3C	****-	10	Increment Memory	[HL]=[HL]+1
INX B	03	-----	6	Increment BC	BC=BC+1
INX D	13	-----	6	Increment DE	DE=DE+1
INX H	23	-----	6	Increment HL	HL=HL+1
INX SP	33	-----	6	Increment Stack Pointer	SP=SP+1
JMP a	C3	-----	7	Jump unconditional	PC=a

Mnemonic	Op	SZAPC	~s	Description	Notes
JC a	DA	-----	7	Jump on Carry	If CY=1(10~s)
JM a	FA	-----	7	Jump on Minus	If S=1 (10~s)
JNC a	D2	-----	7	Jump on No Carry	If CY=0(10~s)
JNZ a	C2	-----	7	Jump on No Zero	If Z=0 (10~s)
JP a	F2	-----	7	Jump on Plus	If S=0 (10~s)
JPE a	EA	-----	7	Jump on Parity Even	If P=1 (10~s)
JPO a	E2	-----	7	Jump on Parity Odd	If P=0 (10~s)
JZ a	CA	-----	7	Jump on Zero	If Z=1 (10~s)
LDA a	3A	-----	13	Load Accumulator direct	A=[a]
LDAX B	0A	-----	7	Load Accumulator indirect	A=[BC]
LDAX D	1A	-----	7	Load Accumulator indirect	A=[DE]
LHLD a	2A	-----	16	Load HL Direct	HL=[a]
LXI B,nn	01	-----	10	Load Immediate BC	BC=nn
LXI D,nn	11	-----	10	Load Immediate DE	DE=nn
LXI H,nn	21	-----	10	Load Immediate HL	HL=nn
LXI SP,nn	31	-----	10	Load Immediate Stack Ptr	SP=nn
MOV r1,r2	7F	-----	4	Move register to register	r1=r2 (1XX)
MOV M,r	77	-----	7	Move register to Memory	[HL]=r (16X)
MOV r,M	7E	-----	7	Move Memory to register	r=[HL] (1X6)
MVI r,n	3E	-----	7	Move Immediate	r=n (0X6)
MVI M,n	36	-----	10	Move Immediate to Memory	[HL]=n
NOP	00	-----	4	No Operation	
ORA r	B7	**0*0	4	Inclusive OR Accumulator	A=Avr (26X)
ORA M	B6	**0*0	7	Inclusive OR Accumulator	A=Av[HL]
ORI n	F6	**0*0	7	Inclusive OR Immediate	A=Avn
OUT p	D3	-----	10	Output	[p]=A
PCHL	E9	-----	6	Jump HL indirect	PC=[HL]
POP B	C1	-----	10	Pop BC	BC=[SP]+
POP D	D1	-----	10	Pop DE	DE=[SP]+
POP H	E1	-----	10	Pop HL	HL=[SP]+
POP PSW	F1	-----	10	Pop Processor Status Word	{PSW,A}=[SP]+
PUSH B	C5	-----	12	Push BC	-[SP]=BC
PUSH D	D5	-----	12	Push DE	-[SP]=DE
PUSH H	E5	-----	12	Push HL	-[SP]=HL
PUSH PSW	F5	-----	12	Push Processor Status Word	-[SP]={PSW,A}
RAL	17	----*	4	Rotate Accumulator Left	A={CY,A}<-
RAR	1F	----*	4	Rotate Accumulator Righ	A=->{CY,A}
RET	C9	-----	10	Return	PC=[SP]+
RC	D8	-----	6	Return on Carry	If CY=1(12~s)
RIM	20	-----	4	Read Interrupt Mask	A=mask
RM	F8	-----	6	Return on Minus	If S=1 (12~s)
RNC	D0	-----	6	Return on No Carry	If CY=0(12~s)
RNZ	C0	-----	6	Return on No Zero	If Z=0 (12~s)
RP	F0	-----	6	Return on Plus	If S=0 (12~s)
RPE	E8	-----	6	Return on Parity Even	If P=1 (12~s)
RPO	E0	-----	6	Return on Parity Odd	If P=0 (12~s)
RZ	C8	-----	6	Return on Zero	If Z=1 (12~s)
RLC	07	----*	4	Rotate Left Circular	A=A<-
RRC	0F	----*	4	Rotate Right Circular	A=->A
RST z	C7	-----	12	Restart (3X7)	-[SP]=PC,PC=z
SBB r	9F	*****	4	Subtract with Borrow	A=A-r-CY
SBB M	9E	*****	7	Subtract with Borrow	A=A-[HL]-CY
SBI n	DE	*****	7	Subtract with Borrow Immed	A=A-n-CY

```
|SHLD a     |22|-----|16|Store HL Direct           |[a]=HL         |
|SIM        |30|-----| 4|Set Interrupt Mask        |mask=A         |
|SPHL       |F9|-----| 6|Move HL to SP             |SP=HL          |
|STA a      |32|-----|13|Store Accumulator         |[a]=A          |
|STAX B     |02|-----| 7|Store Accumulator indirect|[BC]=A         |
|STAX D     |12|-----| 7|Store Accumulator indirect|[DE]=A         |
|STC        |37|----1| 4|Set Carry                 |CY=1           |
|SUB r      |97|*****| 4|Subtract                  |A=A-r     (22X)|
|SUB M      |96|*****| 7|Subtract Memory           |A=A-[HL]       |
|SUI n      |D6|*****| 7|Subtract Immediate        |A=A-n          |
|XCHG       |EB|-----| 4|Exchange HL with DE       |HL<->DE        |
|XRA r      |AF|**0*0| 4|Exclusive OR Accumulator  |A=Axr     (25X)|
|XRA M      |AE|**0*0| 7|Exclusive OR Accumulator  |A=Ax[HL]       |
|XRI n      |EE|**0*0| 7|Exclusive OR Immediate    |A=Axn          |
|XTHL       |E3|-----|16|Exchange stack Top with HL|[SP]<->HL      |
|-------------+-----+--+------------------------------------------|
| PSW         |-*01 |  |Flag unaffected/affected/reset/set        |
| S           |S    |  |Sign (Bit 7)                              |
| Z           | Z   |  |Zero (Bit 6)                              |
| AC          |  A  |  |Auxilary Carry (Bit 4)                    |
| P           |    P|  |Parity (Bit 2)                            |
| CY          |    C|  |Carry (Bit 0)                             |
|----------------------+------------------------------------------|
| a p                  |Direct addressing                         |
| M z                  |Register indirect addressing              |
| n nn                 |Immediate addressing                      |
| r                    |Register addressing                       |
|----------------------+------------------------------------------|
|DB n(,n)              |Define Byte(s)                            |
|DB 'string'           |Define Byte ASCII character string        |
|DS nn                 |Define Storage Block                      |
|DW nn(,nn)            |Define Word(s)                            |
|----------------------+------------------------------------------|
| A  B  C  D  E  H  L  |Registers (8-bit)                         |
| BC  DE  HL           |Register pairs (16-bit)                   |
| PC                   |Program Counter register (16-bit)         |
| PSW                  |Processor Status Word (8-bit)             |
| SP                   |Stack Pointer register (16-bit)           |
|----------------------+------------------------------------------|
| a   nn               |16-bit address/data (0 to 65535)          |
| n   p                |8-bit data/port (0 to 255)                |
| r                    |Register (X=B,C,D,E,H,L,M,A)              |
| z                    |Vector (X=0H,8H,10H,18H,20H,28H,30H,38H)  |
|----------------------+------------------------------------------|
| +   -                |Arithmetic addition/subtraction           |
| &   ~                |Logical AND/NOT                           |
| v   x                |Logical inclusive/exclusive OR            |
| <-   ->              |Rotate left/right                         |
| <->                  |Exchange                                  |
| [ ]                  |Indirect addressing                       |
| [ ]+   -[ ]          |Indirect address auto-inc/decrement       |
| { }                  |Combination operands                      |
| ( X )                |Octal op code where X is a 3-bit code     |
| If ( ~s)             |Number of cycles if condition true        |
-------------------------------------------------------------------
```

APPENDIX - D

COMPONENT LAYOUT



LIST OF COMPONENT's :-

IC's :-

U1 = 8085 (40 PIN BASE)
U2,U3 = ICL232 (16 PIN BASE)
U4 = 8253 (24 PIN BASE)
U5,U6,U45 = 8255 (40 PIN BASE)
U7 = 7408 (14 PIN BASE)
U8,U47 = 74LS245 (20 PIN BASE)
U9,U10 = 74LS373 (20 PIN BASE)
U11,U21,U48 = 74LS174 (16 PIN BASE)
U12 = 2764 (28 PIN BASE)
U13,U14,U15 = 6264 (28 PIN BASE)
U23,U24,U25,U26 = 6264 (28 PIN BASE)
U16,U27 = 8251 (28 PIN BASE)
U17,U22 = 74LS244 (20 PIN BASE)
U18,U20 = CD4015 (16 PIN BASE)
U19,U44,U46 = 74LS273 (20 PIN BASE)
U28,U29,U30,U31 = PAL16L8 (20 PIN BASE)
U32,U39 = 74LS138 (16 PIN BASE)
U33 = 7404 (14 PIN BASE)
U34,U35 = 74125 (14 PIN BASE)
U36,U43,U49,U53 = 74LS74 (14 PIN BASE)
U37 = LM741 (8 PIN BASE)
U38 = 8279 (40 PIN BASE)
U40 = DAC0800 (18 PIN BASE)
U41,U42 = 7407 (14 PIN BASE)
U50 = 7432 (14 PIN BASE)
U51 = ADC0809 (28 PIN BASE)
U52 = 8259A (28 PIN BASE)
U54,U55 = PC817 (4 PIN MACHINE SOCKET)
U56 = 7400 (14 PIN BASE)

CONNECTOR's :-

P1 = 5PIN UNICON MALE
P2 = CASHIO SOCKET
P3,P4 = 9 PIN 'D' MALE CONNECTOR
P5,P18 = 10 PIN RMC CONNECTOR
P6,P7 = 26 PIN FRC CONNECTOR
P8,P9 = 5PIN RMC CONNECTOR
P10 = 50 PIN FRC CONNECTOR
P11,P12 = EP SOCKET
P13 = 48 PIN EURO CONNECTOR (MALE)
P14 = 25 PIN 'D' FEMALE CONNECTOR
P15 = PS2 KEYBOARD CONNECTOR
P16 = J801 12 PIN CONNECTOR
P17 = 16 PIN FRC CONNECTOR

RESISTOR's :-

R1,R12,R14,R16,R18 = 330E
R20,R22,R24,R26,R42 = 330E
R68,R71 = 330E
R2,R28,R29,R41,R46-R60,R60A = 10K
R3,R6 = 1.5K
R4 = 51K
R5,R31,R32,R34,R38,R43,R75 = 100E
R7,R11,R30,R33,R60B,R66,R67,R69,R79 = 3.3K
R8,R36 = 47K
R9,R10,R39,R70,R78,R80,R83 = 1K
R13,R15,R17,R19,R21,R23,R25,R27 = 22K
R35 = 0E
R37 = 15E
R40 = 470E
R44 = 220E/1W
R45 = 36E/2W
R61,R63,R72,R77 = 5.1K
R62 = 100E/0.5W
R64,R65,R81,R82 = 2.2K
R73,R76 = 100K
R74 = 120E
SIP1,SIP2,SIP4 = 4.7K 10 PIN SIPRESISTOR
SIP3 = 3.3K 10 PIN SIPRESISTOR

CAPACITOR's :-

C1,C3,C5 = 100MF/25V/35V/E
C2,C4,C6-C14,C23-C26 = 0.1M/C
ALL DC = 0.1MF/C
C15 = 10MF/E
C18,C21,C22 = 0.01MF/C
C27-C32 = 10MF/40V/E
C33 = 470PF

DIODE's :-

D1,D3,D4,D5,D6 = 1N4148 DIODE
D19-D30 = 1N4148 DIODE
D2,D11,D14,D33 = 1N4001 DIODE
D7 = 3.6V ZENER DIODE
D8,D9,D12,D32 = 3.9V ZENER DIODE
D10,D31 = 5.6V ZENER DIODE
D13 = 8.2V ZENER DIODE
D15,D16,D17,D18 = 5.6V/0.5W ZENER DIODE

OTHER COMPONENT's :-

RRE KEY 33NO's
BT1 = 3.6V NICD BATTERY
DISP1-DISP8 = TIL313 7 SEGMENT DISPLAY
J1-J4,J6,J9,J10,J11,J12 = 2 PIN JUMPER
J5,J7,J8 = 3 PIN JUMPER
LCD = 16x2 (Or) 20x4 LCD DISPLAY
L1 - L7 = 3MM RED LED
POT = 1K POTMETER
Q1,Q19-Q32 = 2N2222A TRANSISTOR
Q2-Q18 = BC158 TRANSISTOR
RL1,RL2,RL3 = 5V SIP RELAY
TP1 = 10K TRIMPOT (V or H)
SW1 = NCNO SWITCH
SW2-SW13 = TINNY SWITCH
T1 - T8 = P8000 CONNECTOR
X1 = 6.144MHz CRYSTAL
Z1 = 28 PIN ZIF SOCKET

Vi Microsystems Pvt.Ltd.,

MICROIPOWER - 1

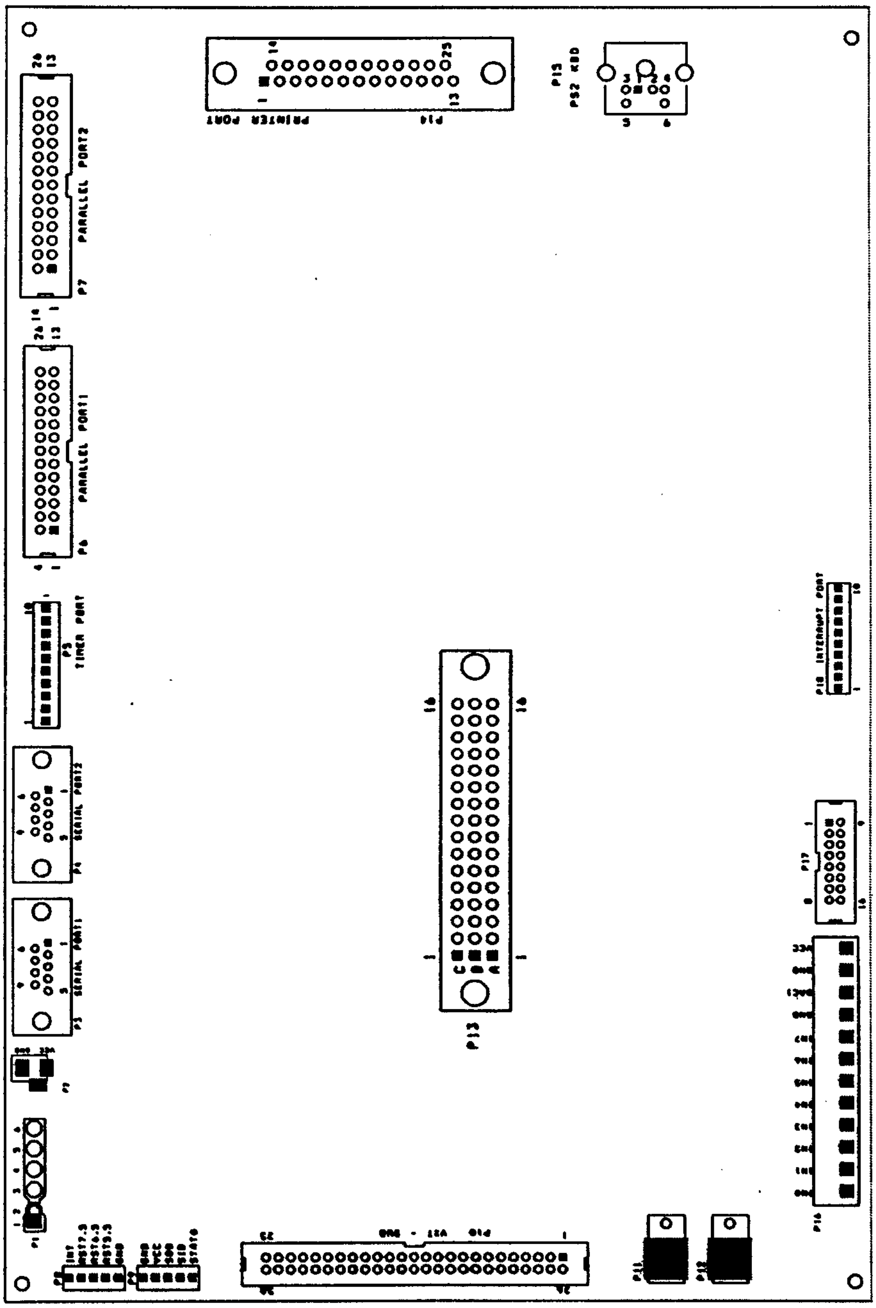
DATE : 28/12/04
REV : 20
SHEET 1 OF 1

COMPONENT LAYOUT

DRN BY:Mrs.R.V CHK BY:Ms.S.J

APPENDIX - E

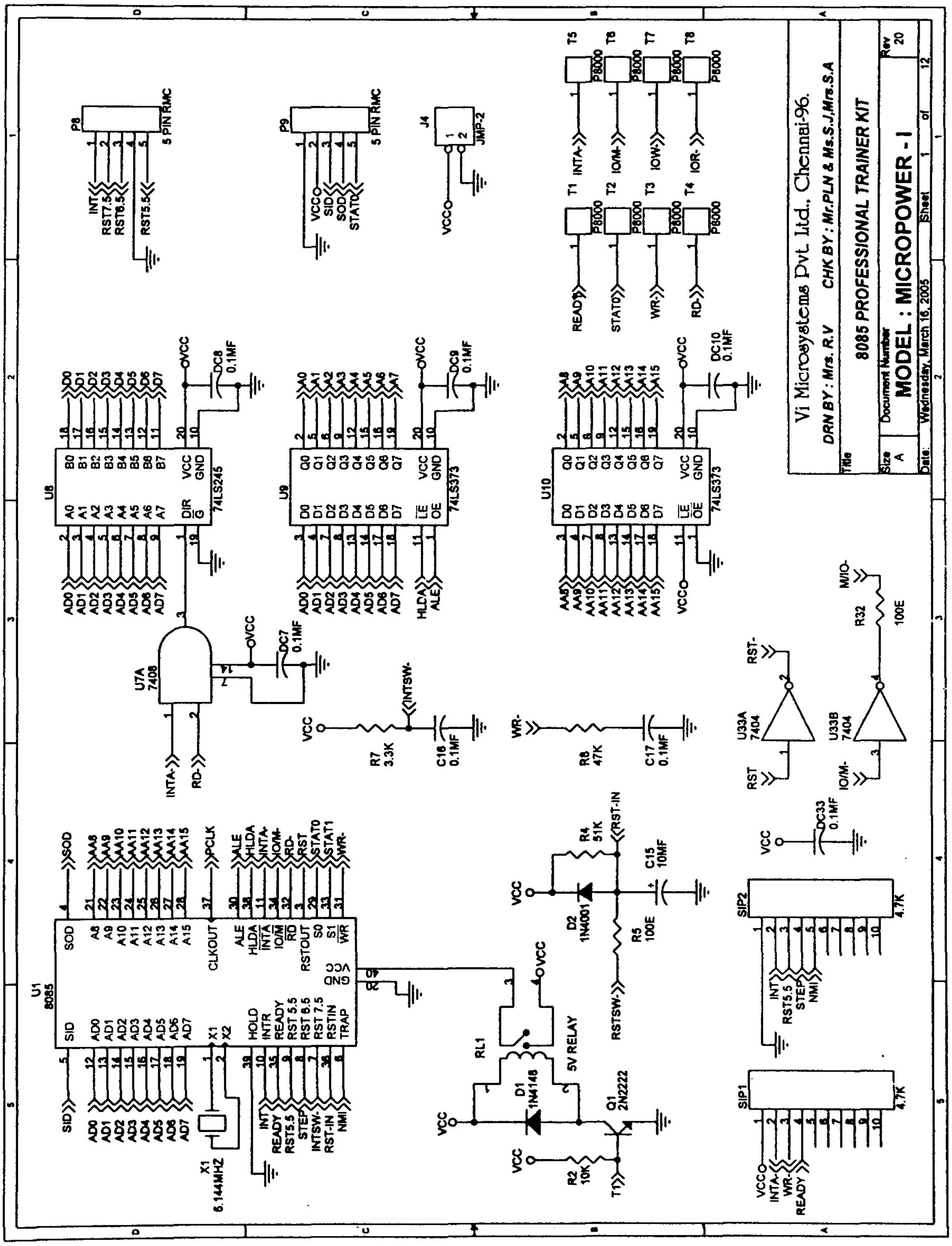
CONNECTOR LAYOUT

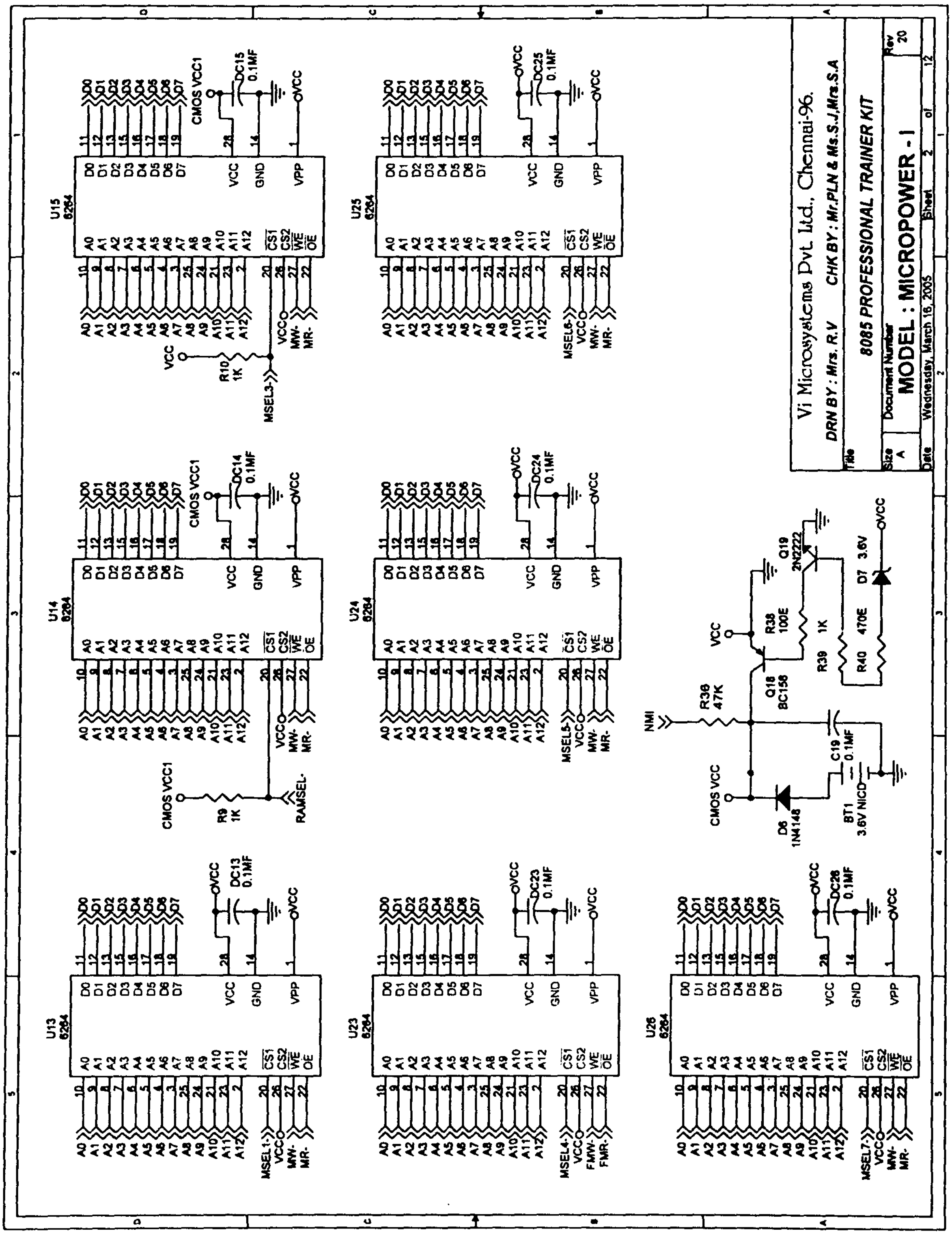

PRINTER PORT
P14
P15
PS2 KBD
PARALLEL PORT2
P7
PARALLEL PORT1
P6
P5
TIMER PORT
SERIAL PORT2
P4
SERIAL PORT1
P3
P2
P1
P13
P18 INTERRUPT PORT
P17
P16
P10
P11
P12
P8
P9

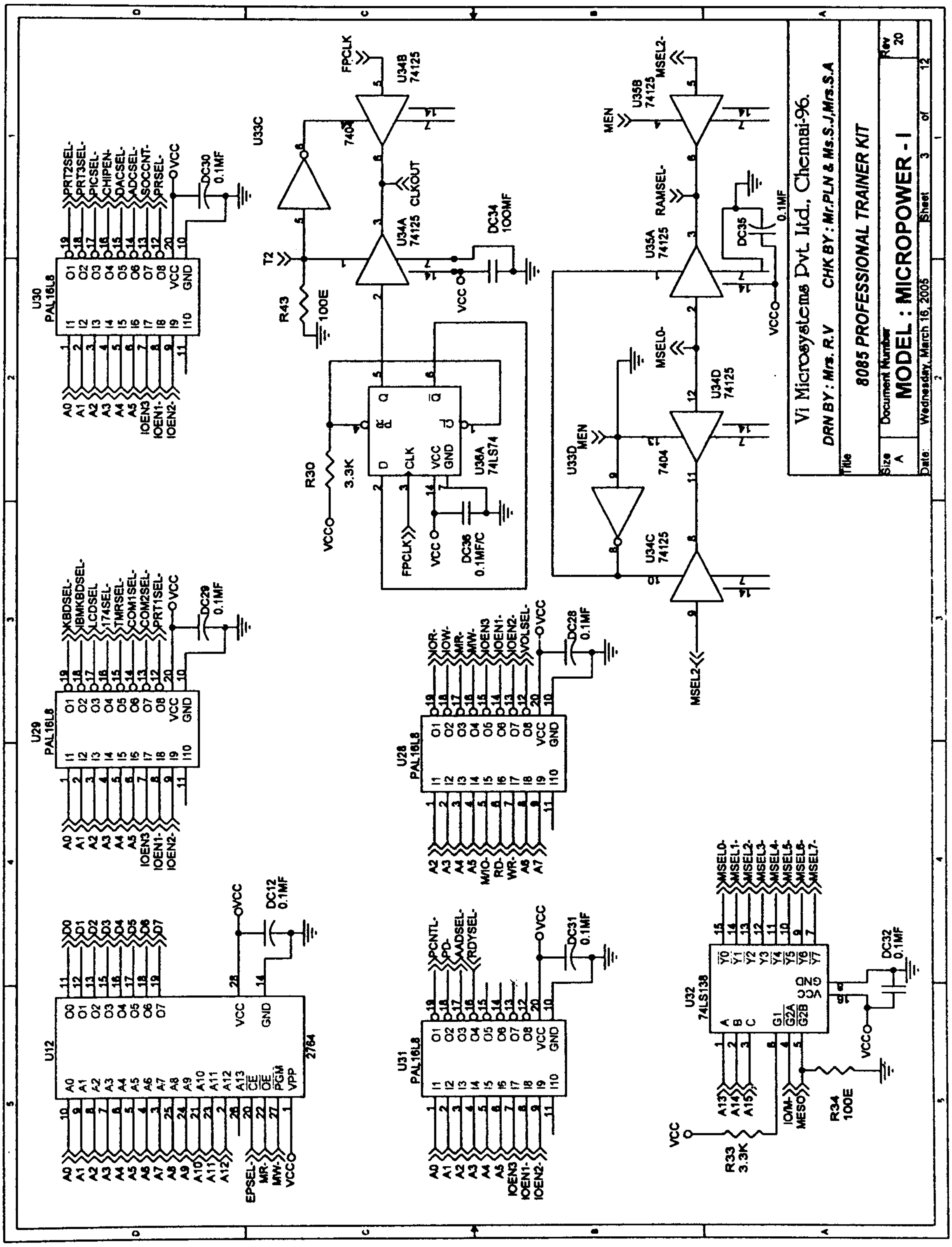
APPENDIX - F

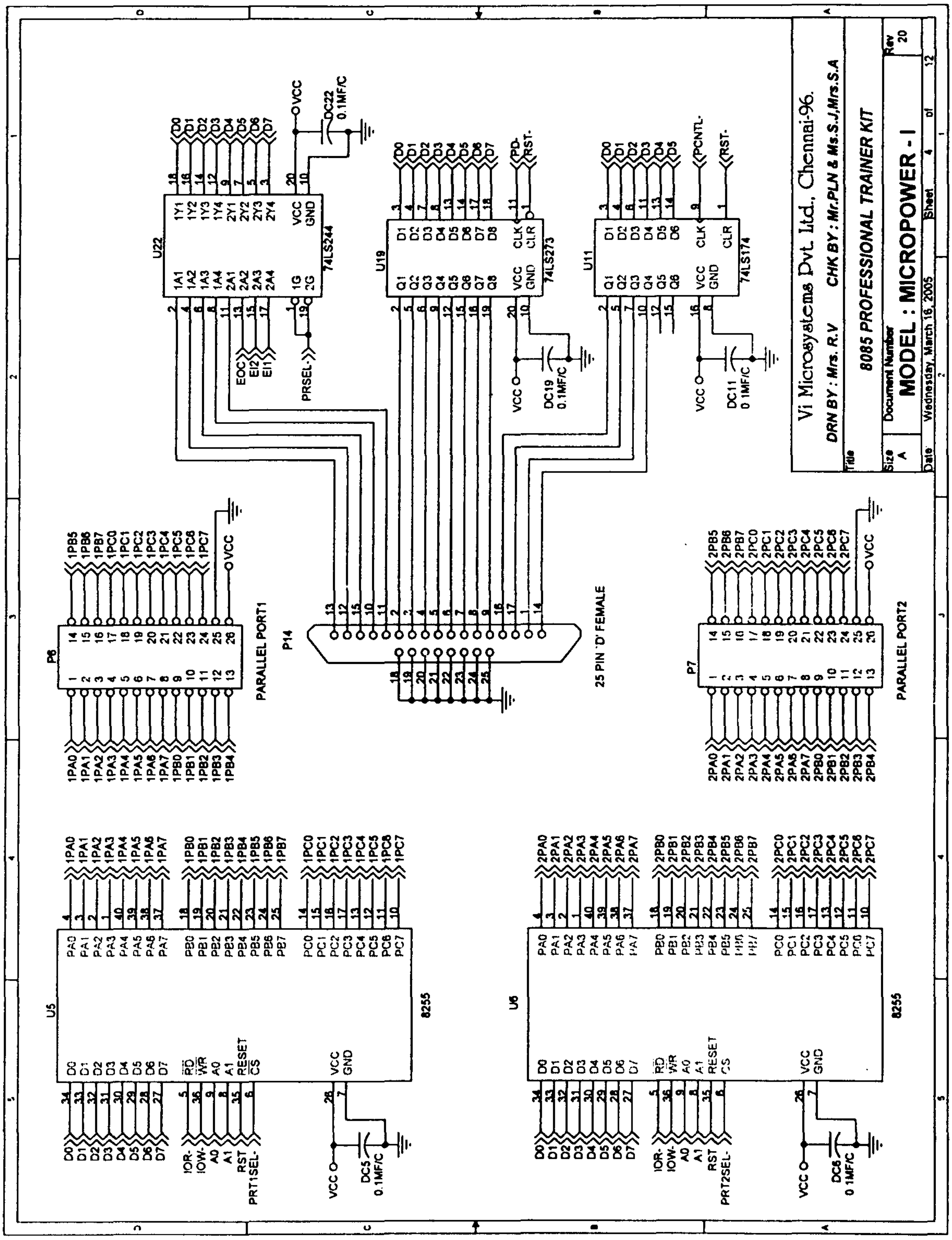
CIRCUIT DIAGRAM

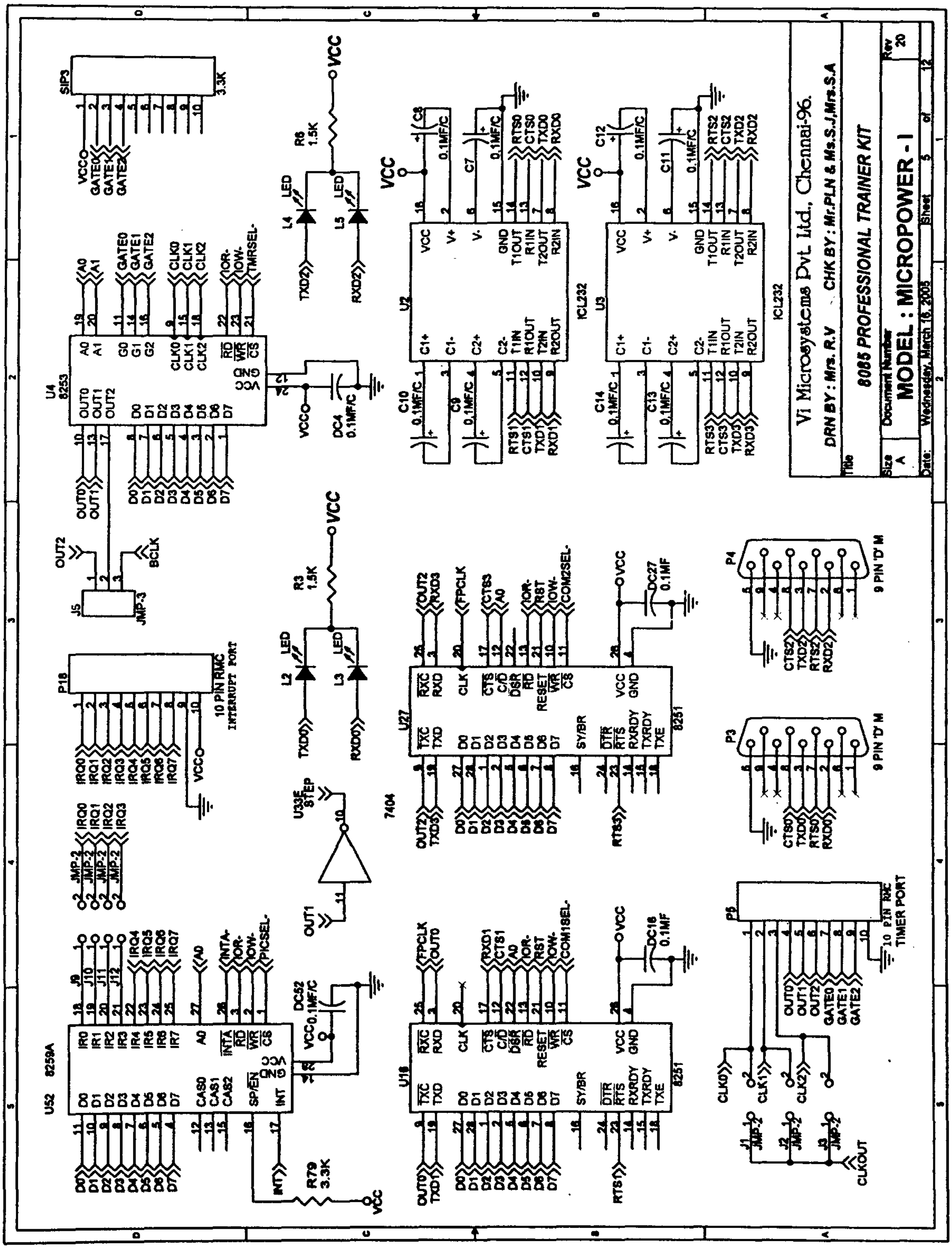
CIRCUIT DIAGARM

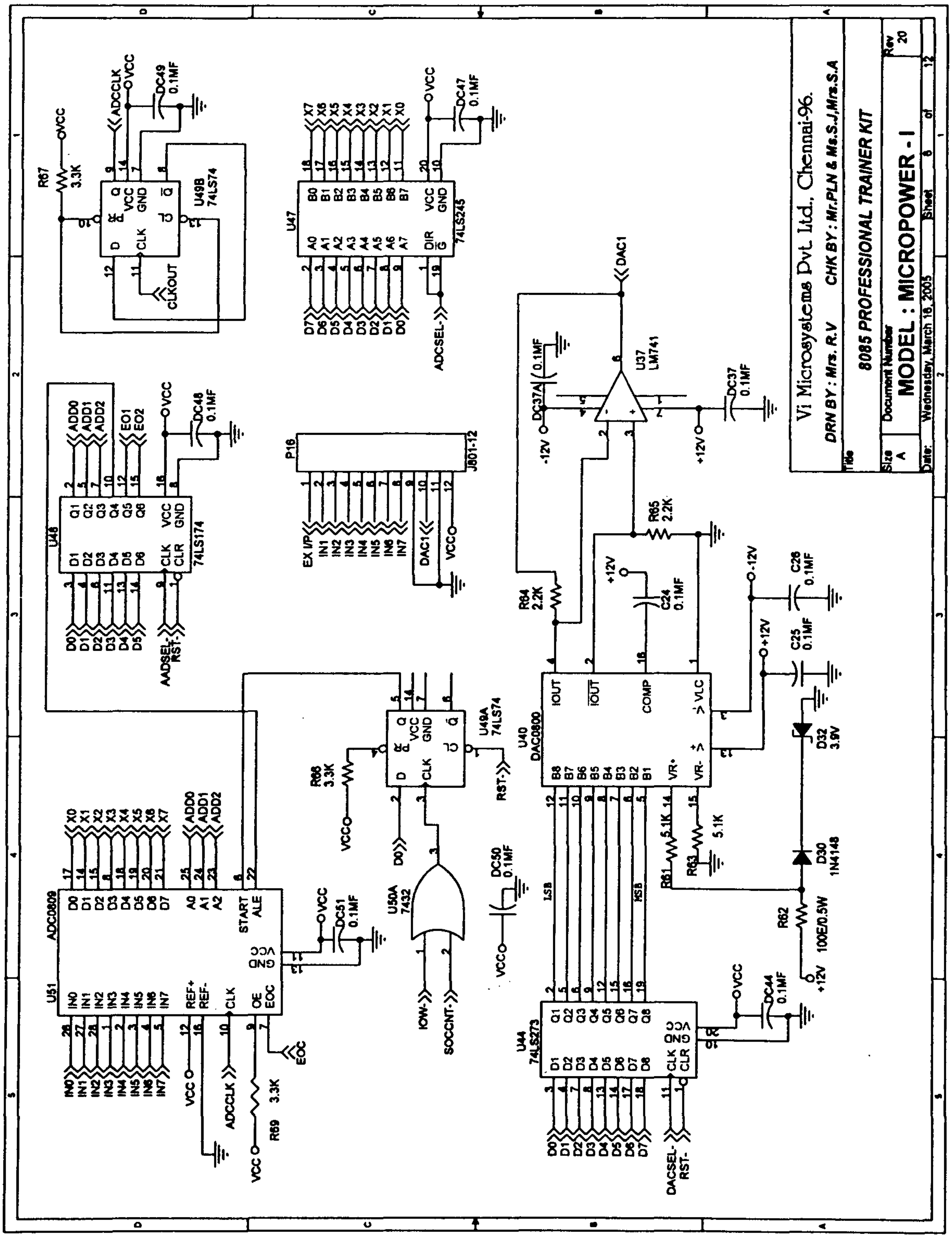


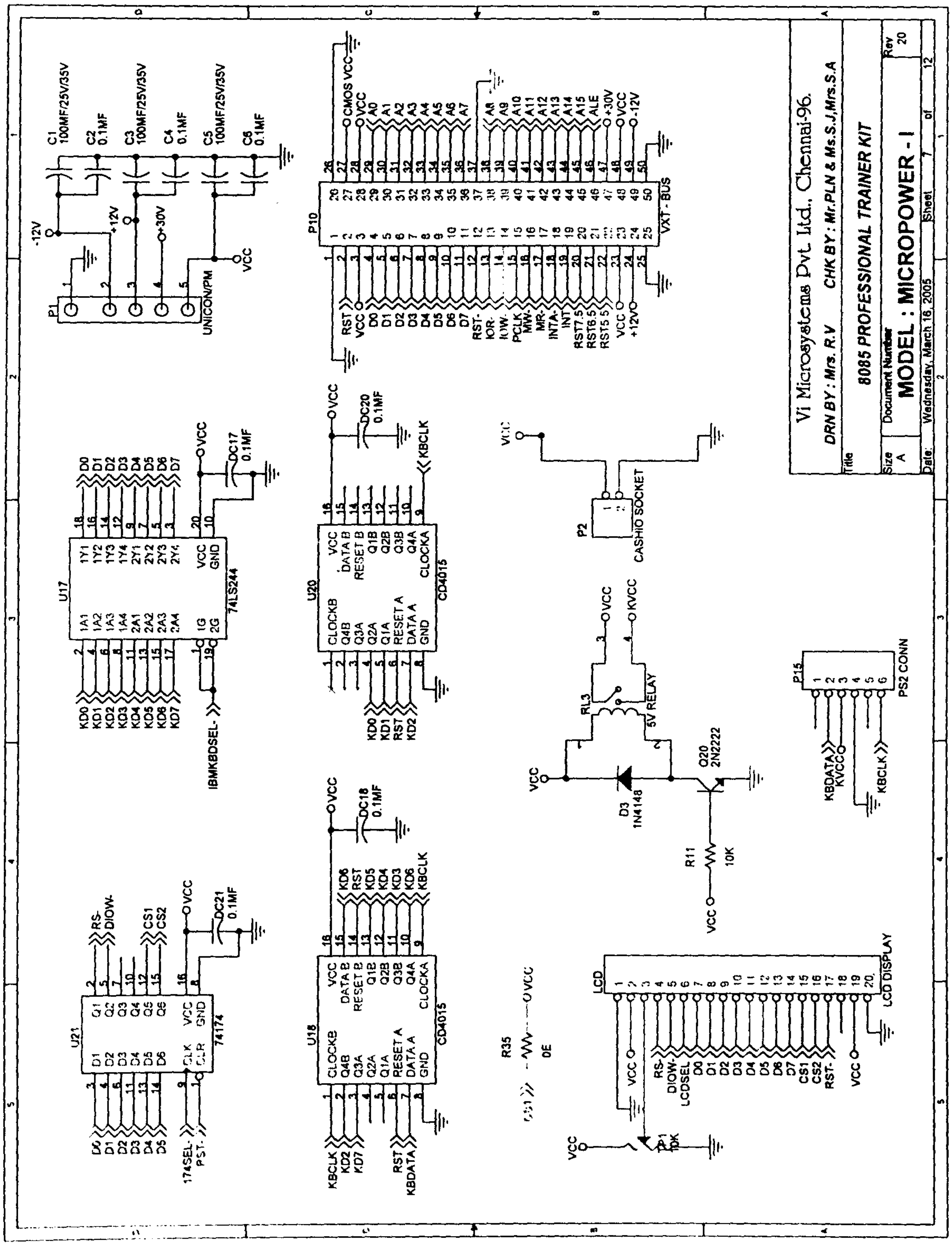

U15
6264
U25
6264
U14
6264
U24
6264
U13
6264
U23
6264
U26
6264
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
CS1
CS2
WE
OE
D0
D1
D2
D3
D4
D5
D6
D7
VCC
GND
VPP
MSEL1-
MSEL3-
MSEL4-
MSEL5-
MSEL7-
RAMSEL-
MW-
MR-
FMW-
FMR-
CMOS VCC1
CMOS VCC
DC13 0.1MF
DC14 0.1MF
DC15 0.1MF
DC23 0.1MF
DC24 0.1MF
DC25 0.1MF
DC26 0.1MF
R9 1K
R10 1K
NMI
R36 47K
Q18 BC158
R38 100E
Q19 2N2222
R39 1K
R40 470E
D7 3.6V
C19 0.1MF
BT1 3.6V NICD
D6 1N4148
Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V
CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
Title
8085 PROFESSIONAL TRAINER KIT
Size
A
Document Number
MODEL : MICROPOWER - I
Rev
20
Date
Wednesday, March 16, 2005
Sheet
2
of
12

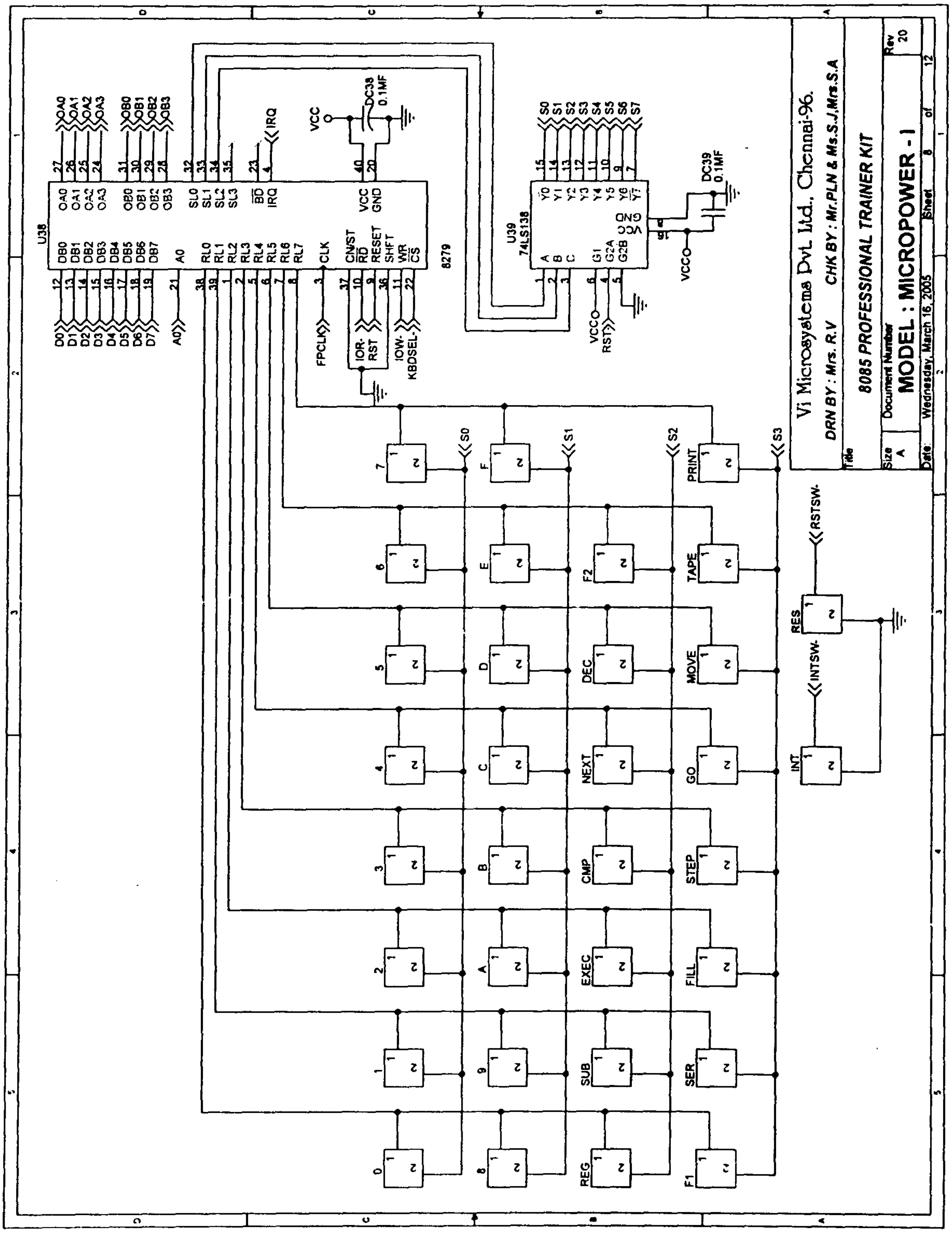

Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V
CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
Title
8085 PROFESSIONAL TRAINER KIT
Size
A
Document Number
MODEL : MICROPOWER - I
Rev
20
Date: Wednesday, March 16, 2005
Sheet 3 of 12
U30
PAL16L8
U29
PAL16L8
U28
PAL16L8
U31
PAL16L8
U12
2764
U32
74LS138
U36A
74LS74
U34A
74125
U34B
74125
U34C
74125
U34D
74125
U35A
74125
U35B
74125
U33C
U33D
7404
R43
100E
R30
3.3K
R33
3.3K
R34
100E
DC30
0.1MF
DC29
0.1MF
DC28
0.1MF
DC31
0.1MF
DC12
0.1MF
DC32
0.1MF
DC34
100MF
DC35
0.1MF
DC36
0.1MF/C
PRT2SEL-
PRT3SEL-
PICSEL-
CHIPEN-
DACSEL-
ADCSEL-
SOCCNT-
PRSEL-
KBDSEL-
IBMKBDSEL-
LCDSEL
174SEL-
TMRSEL-
COM1SEL-
COM2SEL-
PRT1SEL-
IOR-
IOW-
MR-
MW-
IOEN3
IOEN1-
IOEN2-
WOLSEL-
PCNTL-
PD-
AADSEL-
RDYSEL-
M/IO-
RD-
WR-
EPSEL-
MSEL0-
MSEL1-
MSEL2-
MSEL3-
MSEL4-
MSEL5-
MSEL6-
MSEL7-
IO/M-
MESO
FPCLK
CLKOUT
T2
MEN
RAMSEL-
VCC
GND

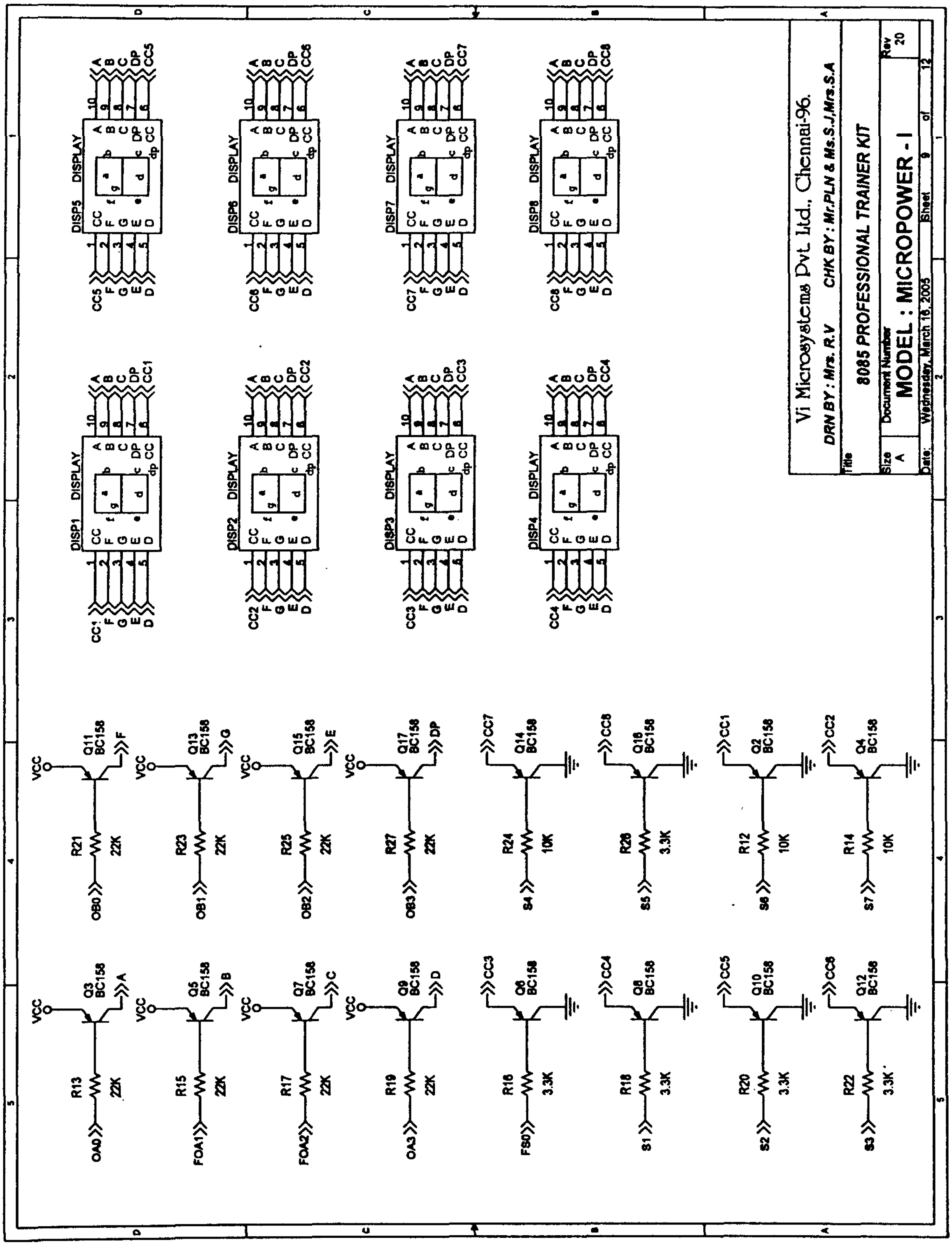

Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V
CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
Title
8085 PROFESSIONAL TRAINER KIT
Size A
Document Number
MODEL : MICROPOWER - I
Rev 20
Date Wednesday, March 16, 2005
Sheet 4 of 12
U22
74LS244
U19
74LS273
U11
74LS174
DC22 0.1MF/C
DC19 0.1MF/C
DC11 0.1MF/C
DC5 0.1MF/C
DC6 0.1MF/C
VCC
PRSEL-
EOC
EI2
EI1
PD-
RST-
PCNTL-
P14
25 PIN 'D' FEMALE
P6
PARALLEL PORT1
P7
PARALLEL PORT2
U5
U6
8255
IOR-
IOW-
A0
A1
RST
PRT1SEL-
PRT2SEL-

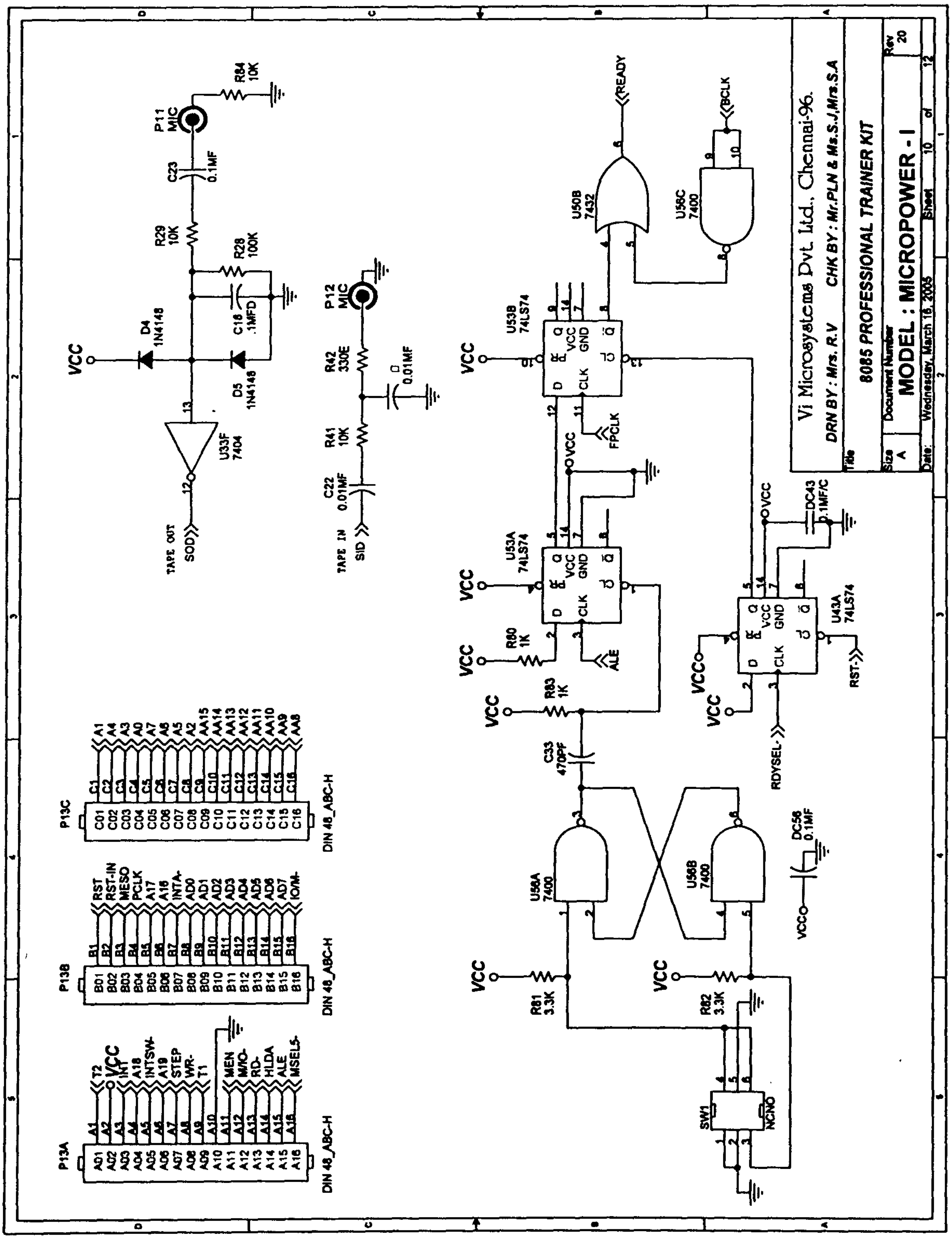

Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V
CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
Title
8085 PROFESSIONAL TRAINER KIT
Size A
Document Number
MODEL : MICROPOWER - I
Rev 2.0
Date: Wednesday, March 16, 2005
Sheet 5 of 12
U4 8253
U52 8259A
U27 8251
U16 8251
U2 ICL232
U3 ICL232
U33E 7404
SIP3 3.3K
P18 10 PIN RMC INTERRUPT PORT
P5 10 PIN RMC TIMER PORT
P3 9 PIN 'D' M
P4 9 PIN 'D' M
J5 JMP-3
R6 1.5K
R3 1.5K
R79 3.3K
L2 LED
L3 LED
L4 LED
L5 LED
STEP
VCC
CLKOUT

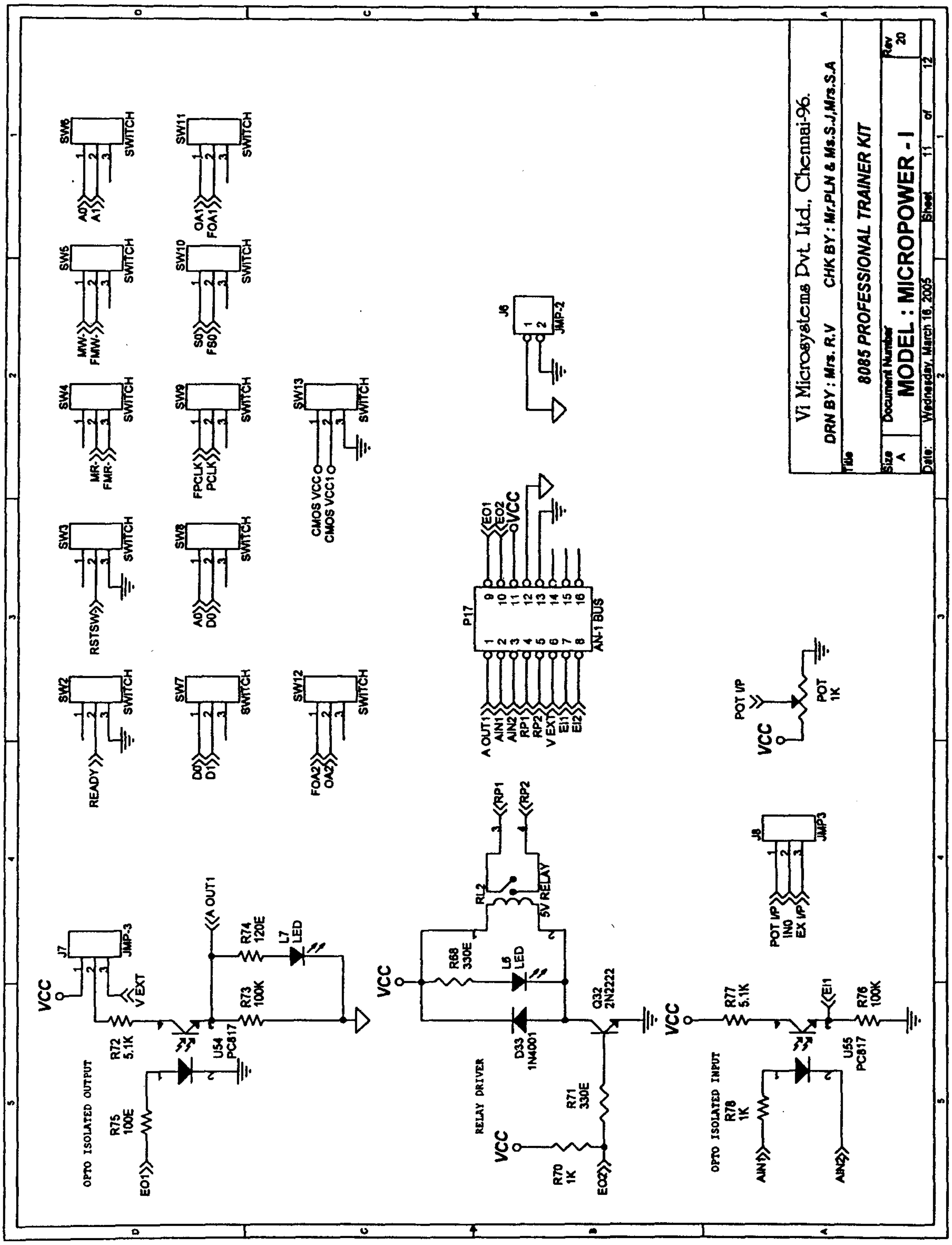

Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V
CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
8085 PROFESSIONAL TRAINER KIT
Title
Size A
Document Number
MODEL : MICROPOWER - I
Rev 20
Date: Wednesday, March 16, 2005
Sheet 6 of 12
U49B 74LS74
R67 3.3K
ADCCLK
CLKOUT
DC49 0.1MF
U47 74LS245
ADCSEL-
DC47 0.1MF
U48 74LS174
ADD0 ADD1 ADD2
EO1 EO2
DC48 0.1MF
AADSEL-
RST-
P16
J801-12
EX I/P
IN1 IN2 IN3 IN4 IN5 IN6 IN7
DAC1
VCC
U37 LM741
DC37A 0.1MF
DC37 0.1MF
-12V
+12V
R65 2.2K
R64 2.2K
C24 0.1MF
C28 0.1MF
C25 0.1MF
U49A 74LS74
R68 3.3K
U40 DAC0800
IOUT
COMP
VLC
VR+
VR-
D32 3.9V
D30 1N4148
R61 5.1K
R63 5.1K
R62 100E/0.5W
LSB
MSB
U50A 7432
IOW-
SOCCNT-
DC50 0.1MF
U51 ADC0809
START
ALE
OE
EOC
REF+
REF-
DC51 0.1MF
R69 3.3K
U44 74LS273
DC44 0.1MF
DACSEL-
RST-

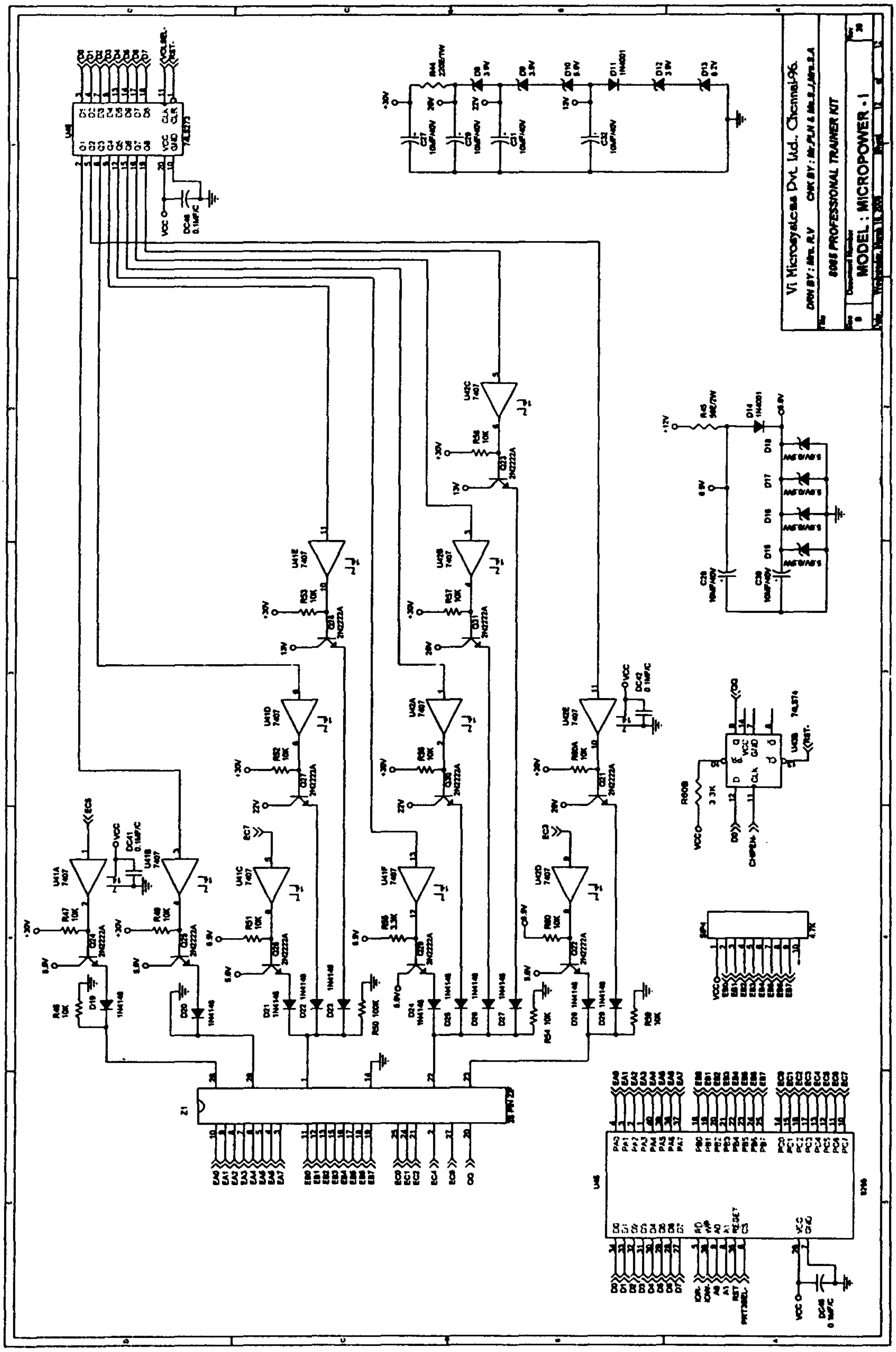

Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V
CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
Title
8085 PROFESSIONAL TRAINER KIT
Size
A
Document Number
MODEL : MICROPOWER - I
Rev
20
Date: Wednesday, March 16, 2005
Sheet 7 of 12
P1
UNICON/PM
-12V
+12V
+30V
VCC
C1 100MF/25V/35V
C2 0.1MF
C3 100MF/25V/35V
C4 0.1MF
C5 100MF/25V/35V
C6 0.1MF
P10
VXT - BUS
RST
VCC
D0
D1
D2
D3
D4
D5
D6
D7
RST-
IOR-
IOW-
PCLK
MW-
MR-
INTA-
INT
RST7.5
RST6.5
RST5.5
+12V
CMOS VCC
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
ALE
+30V
-12V
U17
74LS244
1A1
1A2
1A3
1A4
2A1
2A2
2A3
2A4
1G
2G
1Y1
1Y2
1Y3
1Y4
2Y1
2Y2
2Y3
2Y4
GND
KD0
KD1
KD2
KD3
KD4
KD5
KD6
KD7
IBMKBDSEL-
DC17 0.1MF
U20
CD4015
CLOCKB
Q4B
Q3A
Q2A
Q1A
RESET A
DATA A
DATA B
RESET B
Q1B
Q2B
Q3B
Q4A
CLOCKA
KBCLK
DC20 0.1MF
P2
CASHIO SOCKET
U21
74174
CLK
CLR
Q1
Q2
Q3
Q4
Q5
Q6
RS-
DIOW-
CS1
CS2
174SEL-
PST-
DC21 0.1MF
U18
CD4015
KBDATA
DC18 0.1MF
RL3
5V RELAY
KVCC
Q20
2N2222
D3
1N4148
R11
10K
P15
PS2 CONN
R35
0E
LCD
LCD DISPLAY
LCDSEL
R1
10K


Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V
CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
Title
8085 PROFESSIONAL TRAINER KIT
Size
A
Document Number
MODEL : MICROPOWER - I
Rev
20
Date: Wednesday, March 16, 2005
Sheet 8 of 12
U38
8279
U39
74LS138
DC38
0.1MF
DC39
0.1MF
VCC
GND
VCCO
RST
D0
D1
D2
D3
D4
D5
D6
D7
A0
FPCLK
IOR-
RST
IOW-
KBDSEL-
OA0
OA1
OA2
OA3
OB0
OB1
OB2
OB3
IRQ
S0
S1
S2
S3
S4
S5
S6
S7
RSTSW-
INTSW-
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
REG
SUB
EXEC
CMP
NEXT
DEC
F2
F1
SER
FILL
STEP
GO
MOVE
TAPE
PRINT
RES
INT


DISP1 DISPLAY
DISP2 DISPLAY
DISP3 DISPLAY
DISP4 DISPLAY
DISP5 DISPLAY
DISP6 DISPLAY
DISP7 DISPLAY
DISP8 DISPLAY
CC F G E D
A B C DP CC
a b c d e f g dp
CC1 CC2 CC3 CC4 CC5 CC6 CC7 CC8
VCC
Q3 BC158 A
Q5 BC158 B
Q7 BC158 C
Q9 BC158 D
Q11 BC158 F
Q13 BC158 G
Q15 BC158 E
Q17 BC158 DP
Q6 BC158 CC3
Q8 BC158 CC4
Q10 BC158 CC5
Q12 BC158 CC6
Q14 BC158 CC7
Q16 BC158 CC8
Q2 BC158 CC1
Q4 BC158 CC2
R13 22K OA0
R15 22K FOA1
R17 22K FOA2
R19 22K OA3
R21 22K OB0
R23 22K OB1
R25 22K OB2
R27 22K OB3
R16 3.3K FS0
R18 3.3K S1
R20 3.3K S2
R22 3.3K S3
R24 10K S4
R26 3.3K S5
R12 10K S6
R14 10K S7
Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
Title
8085 PROFESSIONAL TRAINER KIT
Size A
Document Number
MODEL : MICROPOWER - I
Rev 20
Date: Wednesday, March 16, 2005 Sheet 9 of 12
A B C D
1 2 3 4 5


Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V
CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
Title
8085 PROFESSIONAL TRAINER KIT
Size
A
Document Number
MODEL : MICROPOWER - I
Rev
20
Date: Wednesday, March 16, 2005
Sheet 10 of 12
P11 MIC
R84 10K
C23 0.1MF
R29 10K
R28 100K
C16 .1MFD
D4 1N4148
D5 1N4148
VCC
U33F 7404
TAPE OUT SOD
P12 MIC
R42 330E
D 0.01MF
R41 10K
C22 0.01MF
TAPE IN SID
READY
BCLK
U50B 7432
U56C 7400
U53B 74LS74
FPCLK
U53A 74LS74
ALE
R80 1K
R83 1K
C33 470PF
U43A 74LS74
DC43 0.1MF/C
RST-
RDYSEL-
U56A 7400
U56B 7400
DC56 0.1MF
R81 3.3K
R82 3.3K
SW1
NCNO
P13A
DIN 48_ABC-H
T2
VCC
INT
A18
INTSW-
A19
STEP
WR-
T1
MEN
M/IO-
RD-
HLDA
ALE
MSEL5-
P13B
DIN 48_ABC-H
RST
RST-IN
MESO
PCLK
A17
A16
INTA-
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
IO/M-
P13C
DIN 48_ABC-H
A1
A4
A3
A0
A7
A6
A5
A2
AA15
AA14
AA13
AA12
AA11
AA10
AA9
AA8


Vi Microsystems Pvt. Ltd., Chennai-96.
DRN BY : Mrs. R.V
CHK BY : Mr.PLN & Ms.S.J,Mrs.S.A
Title
8085 PROFESSIONAL TRAINER KIT
Size
A
Document Number
MODEL : MICROPOWER - I
Rev
20
Date: Wednesday, March 16, 2005
Sheet 11 of 12
SW2 SWITCH READY
SW3 SWITCH RSTSW
SW4 SWITCH MR- FMR-
SW5 SWITCH MW- FMW-
SW6 SWITCH A0 A1
SW7 SWITCH D0 D1
SW8 SWITCH A0 D0
SW9 SWITCH FPCLK PCLK
SW10 SWITCH S0 F80
SW11 SWITCH OA1 FOA1
SW12 SWITCH FOA2 OA2
SW13 SWITCH CMOS VCC CMOS VCC1
J6 JMP-2
J7 JMP-3
J8 JMP3
POT I/P IN0 EX I/P
P17 AN-1 BUS
A OUT1 AIN1 AIN2 RP1 RP2 V EXT EI1 EI2
EO1 EO2 VCC
POT I/P POT 1K
VCC
OPTO ISOLATED OUTPUT
R75 100E
R72 5.1K
U54 PC817
R73 100K
R74 120E
L7 LED
A OUT1
V EXT
EO1
RELAY DRIVER
R68 330E
L6 LED
D33 1N4001
Q32 2N2222
R71 330E
R70 1K
EO2
RL2 5V RELAY
RP1 RP2
OPTO ISOLATED INPUT
R77 5.1K
R78 1K
R76 100K
U55 PC817
EI1
AIN1 AIN2


Vi Microsystems Pvt. Ltd., Chennai-96.
8085 PROFESSIONAL TRAINER KIT
MODEL : MICROPOWER - I
74LS273
7407
2N2222A
1N4148
1N4001
74LS74
8255
SIP4
4.7K
VCC
GND
CLK
CLR
+30V
+12V
Z1
28 PIN ZIF