MICROPROCESSOR BASED SET POINT GENERATOR

A DISSERTATION

submitted in partial fulfilment of the requirements for the award of the degree

of

MASTER OF ENGINEERING

in

ELECTRICAL ENGINEERING

(System Engineering and Operations Research)

By

SHIV KUMAR SINGH

DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE-247 667 (INDIA)

February, 1985

DEDICATED TO MY BROTHER

BHUPENDRA

WHOSE FRAGRANT MEMORY

LINGERS.....

CERTIFICATE

Certified that the dissertation entitled 'Microprocessor Based Set Point Generator' which is being submitted by Shiv Kumar Singh in partial fulfilment for the award of the Degree of Master of Engineering in Electrical Engineering (System Engineering & Operation Research) of the University of Roorkee, Roorkee , is a record of student's own work carried out by him under my supervision and guidance. The matter embodied in this dissertation has not been submitted for the award of any other Degree or Diploma.

This is further to certify that he has worked for a period of about seven and half months from July 1984 to 15th February 1985 for preparing this dissertation at this University.

(M. K. Vasantha)
Reader,
Electrical Engineering Department,
University of Roorkee, Roorkee-247667 (U.P)

Dated:
Feb 18, 1985.

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to honourable and generous Mr. M.K. Vasantha, Reader, Department of Electrical Engineering, University of Roorkee, Roorkee for his unflagging, ungrudging and insightfull guidance at different stages of my work, specially when initiating me into the world of microprocessor. His unbiased support and gesture to handle the Micro Development System and other costly related equipments freely, led me to encounter and solve numberous problems related with the practical work.

Mr Vasantha's overwhelming personality has swept me for good. I will try to take leaf out of every aspect of his life which is so encouraging, so enlightening and so heartening.

I am also grateful to Prof. A.K. Pant, Dr J.D.Sharma and Dr Hari Om Gupta, Department of Electrical Engineering for their enthusiastic favour and immence help at various proceedings.

I am extremely grateful to Mr Jageesh, Rajeev and Naresh for making this dissertation possible. Special thanks are due to Mr. J.P.Sharma and Mr. Rajendra Singh for giving me ample support in building up the experimental set-up.

Further, I attribute to the encouragement, help and guidance received by my parents.

(SHIV KUMAR SINGH)

ABSTRACT

With the advancement of digital computer technology the microprocessor based systems are being used in nearly all the walks of life. Their usage is wide spread, from house-hold gadgets to sophisticated missiles. Advent of microprocessor based system in the field of operation research is not new. The main objectives of this dissertation is to acquire the complete understanding of the Micro-Development System and to develop the SET POINT GENERATOR.

Microprocessor based Set Point Generator was a natural choise of the practical system selected for development because of necessity of testing the reliability of equipments and components in Environmental Test Chamber before actually using them. The Department of Electrical Engineering has acquired the Environmental Test Chamber recently where the Set Point Generator are through hardwired logic.

The basic unit used was an Intel's 8085A based, AM02 Micro Development System, it brief description of Micro Development System is given in Chapter - 1. Chapter - 2 deals the digital to analog conversion along with DAC-08 Card.

Chapter - 3 deals with SET POINT GENERATOR characteristics and its field of application. Finally Soft Ware Development is given in Chapter - 4 along with all necessary subroutines. Experimentations, Description, Conclusion and Suggestions are given in last Chapter - 5.

...

CHAPTER	CONTENTS		Page No
1	GETTING ACQUAINTED WITH AMO2 MICRO DEVELOPMENT SYSTEM (VMDS-85):		
	1.1 Introduction	...	1-1
	1.2 General Description of VMDS-85	...	1-1
	1.3 System Specification	...	1-5
	1.4 System Capabilities	...	1-5
	1.5 Hardware Description	...	1-9
	1.6 System Software	...	1-19
	1.7 Data Terminal (VDT-85)	...	1-23
2	DIGITAL TO ANALOG CONVERTER		
	2.1 Introduction	...	2-1
	2.2 A Basic D/A Converter	...	2-2
	2.3 The Ladder Method	...	2-3
	2.4 The DAC Chips	...	2-7
	2.5 Digital to Analog Card	...	2-13
3	SET POINT GENERATOR		
	3.1 Introduction	...	3-1
	3.2 Application Field of Set Point Generator	...	3-1
	3.3 Duration of Cycle	...	3-1
	3.4 Getting Acquainted with Environmental Test Chamber	...	3-3
	3.5 Waveform of Set Point Generator	...	3-5

CHAPTER C O N T E N T S Page No

APPENDICES

...

/LIST OF FIGURES/

....

LIST OF TABLES

....

CHAPTER - 1

GETTING ACQUAINTED WITH AMO2 MICRO DEVELOPMENT SYSTEM (VMDS - 85)

1.1 INTRODUCTION

AMO2, (VMDS) marketed by VINYTICS, NEW DELHI, is a MICRO DEVELOPMENT SYSTEM based on the Intel's most popular microprocessor, the 8085A. This system formed the basis of all the work carried out in this dissertation. It is, therefore, necessary to gain an insight into the various functional aspects of the system before proceeding further with the discussion of the dissertation work.

The pin configuration and functional block diagram of the 8085A up is given at Appendix A. Fig.1.1 shows a block diagram representation of VMDS-85. The circuit diagrams for Micro Development System are shown in Fig.1.2 and Fig.1.3. VDT-85 is used as a Serial Communication Device. Its details are written in section 1.7.

1.2 GENERAL DESCRIPTION OF VMDS-85

The Microprocessor Training Kits available in India are sufficient to make the user aware of microprocessor. But these microprocessor training kits are too primitive for the development of any sizable application program. In order to develop any application program, the user requires a Development System. It can be very effectively used for developing any size of application program as well as a target system.

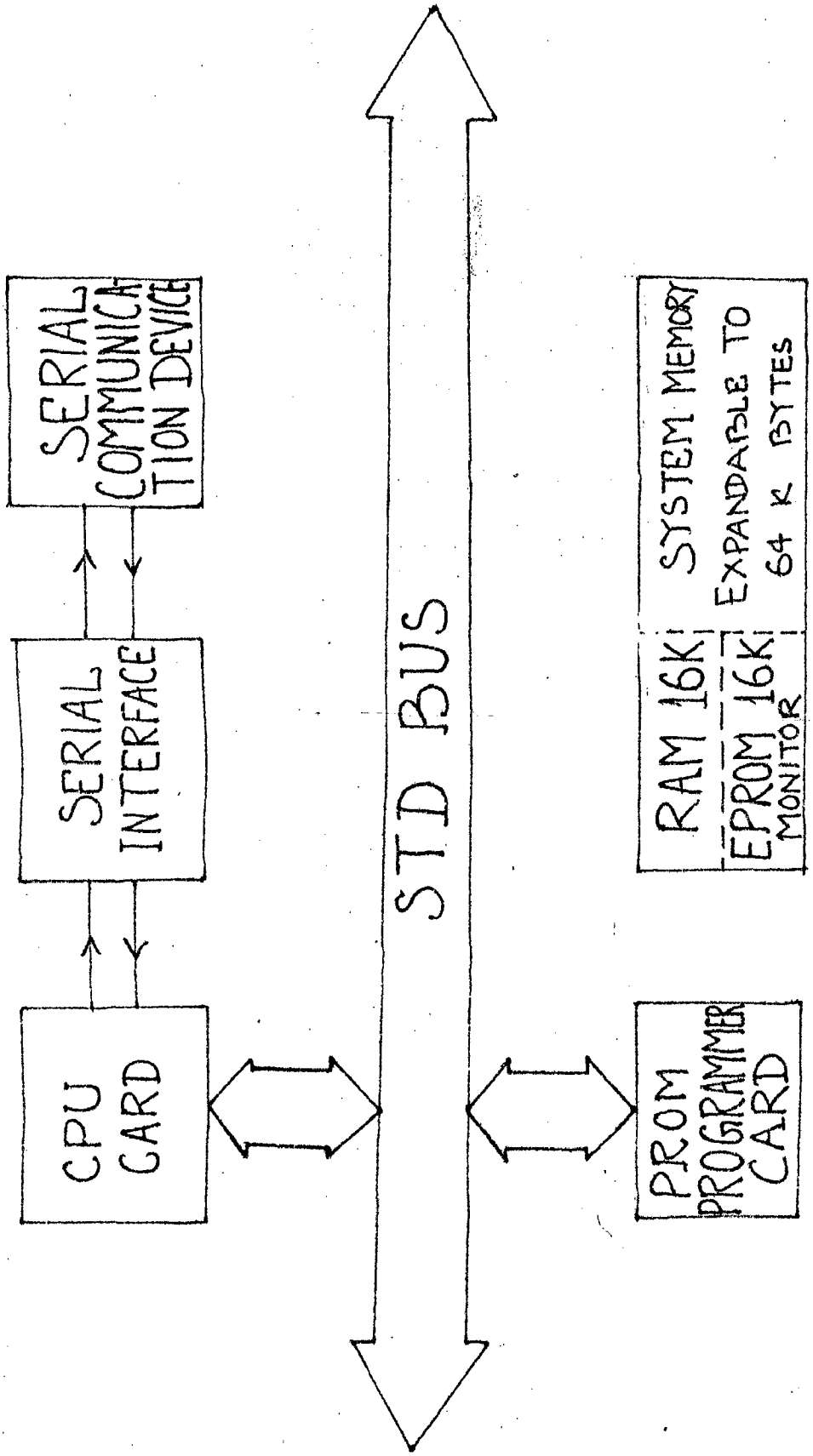


FIG. 1.1 BLOCK DIAGRAM OF VMDS-85

The VMDS based on 8085A, is housed in a 19'' card cage, having CPU card, RAM card, ERROM card, all based on INTERNATIONAL STANDARD BUS, The STD Bus. As the system is based on STD Bus, it can be easily expanded to support any of other cards like Memory card, Opto isolated input card and relay output card, Digital I/D and Timer card etc. It entirely depends upon the user's demands. Condition is that, card must be based on International Standard Bus so that it may easily be interfaced with CPU through STD Bus. In this dissertation, DIGITAL TO ANALOG CONVERTER based on DAC-08, card is used, which is designed also over STD Bus. Its details are placed in Chapter - 2.

AMO2 consists of Monitor Group, Assembler Group, De-assembler Group, Relocation Group, PROM program Group, Misc. and Menu Group. With this powerful system software, one can develop, debug, and modify the program for any target system.

The system can be used to develop any real time software and relocate the entire program to make it work for other memory. This relocated program can be directly burned into EPROM, which can be used with the required hardware to make a complete target system. The powerful Documentation Commands help the user to get well documented programs from the the development system.

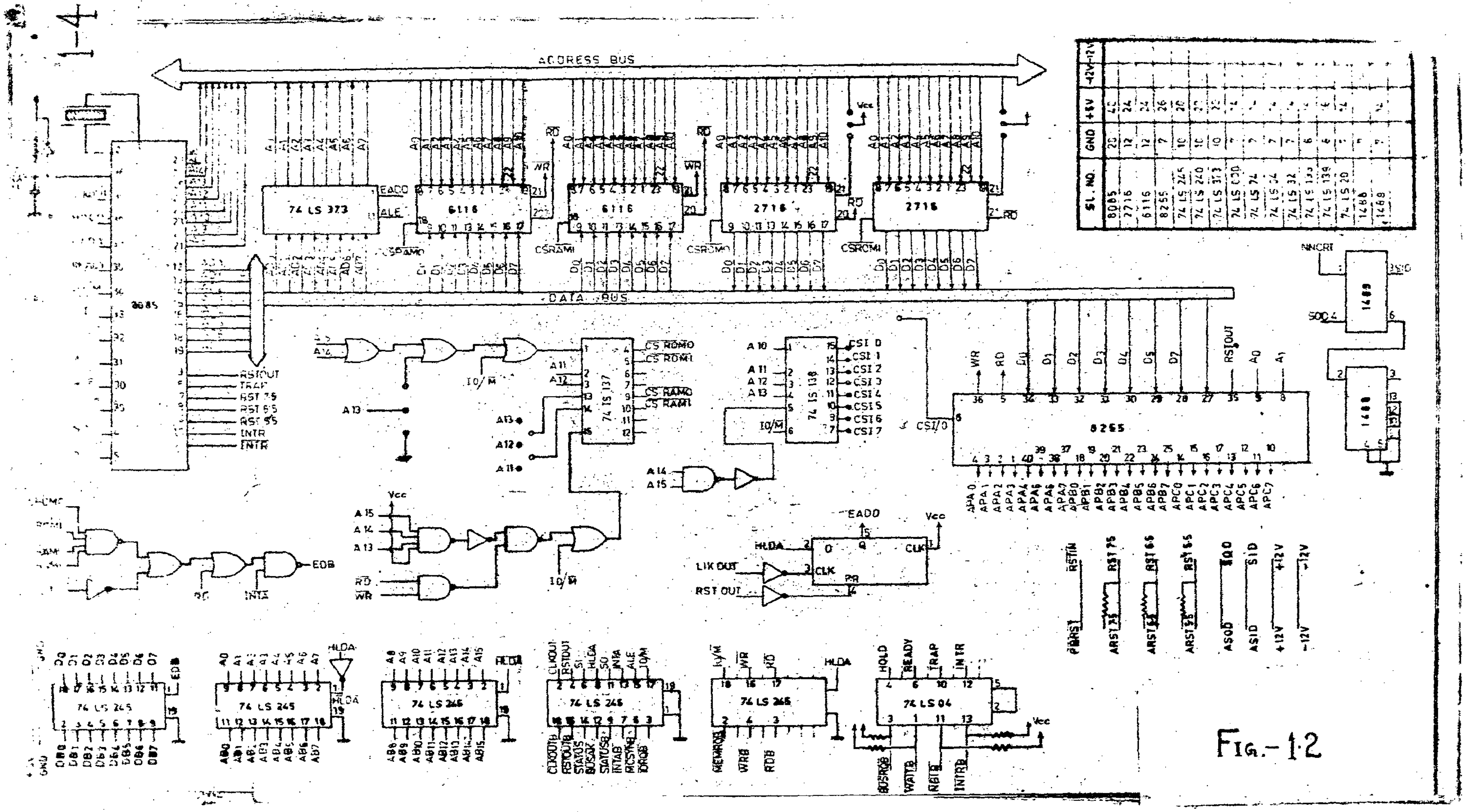


Fig.-1.2

SL. NO.	GND	+5V	-12V	+12V
8085	20	40		
2716	12	24		
6116	12	24		
8255	7	26		
74 LS 245	10	20		
74 LS 240	10	20		
74 LS 373	10	20		
74 LS C30	7	14		
74 LS 74	7	14		
74 LS 04	7	14		
74 LS 32	7	14		
74 LS 133	8	16		
74 LS 139	8	16		
74 LS 20	7	14		
1488	7			
1489	7	14		

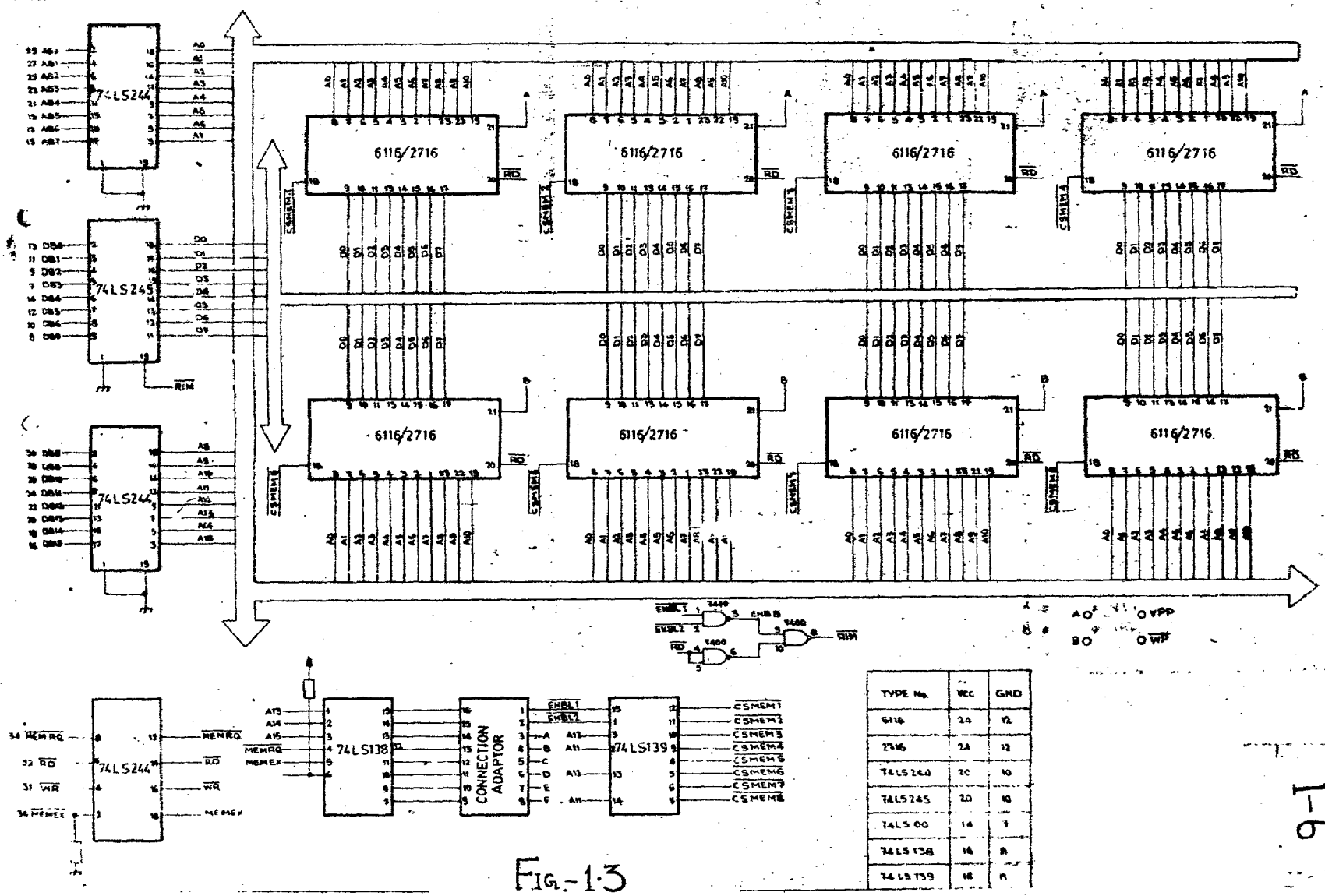
1.3 SYSTEM SPECIFICATION

CPU	8 Bit Microprocessor, the 8085A.
MEMORY	Can be expanded upto 64 K byte.
RAM	16 K byte (8 Nos. of CMOS RAM. 6116).
ROM	16 K byte loaded with powerful moniter.
I/O LINES	24 Programable input/output lines can be incorporated in the CPU card.
INTERFACE	TTY (20 mA current loop). CRT (20 mA or RS-232-C interface).
BUS	STD Bus. All addresses, data and control signals (TTL compatible) available at edge connector.
POWER SUPPLY REQUIRE-MENT	230 V, +/- 10%, 50 Hz.
OPERATING TEMP.	0 to 50 degree centigrade.
SLOTS	6 slots capacity, out of these three are reserved for CPU, RAM, and EPROM. And others three are left to expand the system facility.

1.4 SYSTEM CAPABILITIES

The capabilities of the system are listed as follows,

1. The moniter group can input/modify a program or data, introduce a Break point, restore a break point, Executive a program/subroutine with a symbolic name or address, Memory compare, Fill and block move the memory, Tape read/write etc.



TYPE No.	Vcc	GND
6116	24	12
2716	24	12
74LS240	20	10
74LS245	20	10
74LS00	14	7
74LS138	16	8
74LS139	16	8

FIG.-1.3

2. The relocate group can modify code/address with or without printout, Relocate and link the program, Relocate the multiprogram, Reverse relocate, insert and delete with relocation etc.

3. In the documentation group, the program can be documented by taking a listing of ASCII/Hexdata, or by listing the program in assembly language on VDU/TTY with or without comments.

4. The assembler group allows entry of a program in assembly. It allows relocation of program with or without labels. Labels can be entered, modified, removed and listed the independent of program entry.

5. Pseudo commands like Data ASCII, Data Byte, Data word, Data storage, Eng of assembly, End intermediate, Go back in program to cancle N number of instruction, help in the entry of an assembly language program.

6. The optional PROM programer can programme most popular 'NMOS (except 2708, 1708) PROM', list its data, verify Blank check etc.

7. The MENU helps the user to select a particular group and then enter a specific command within the group.

Note:- All the 69 commands of various groups are placed in Appendix-C.

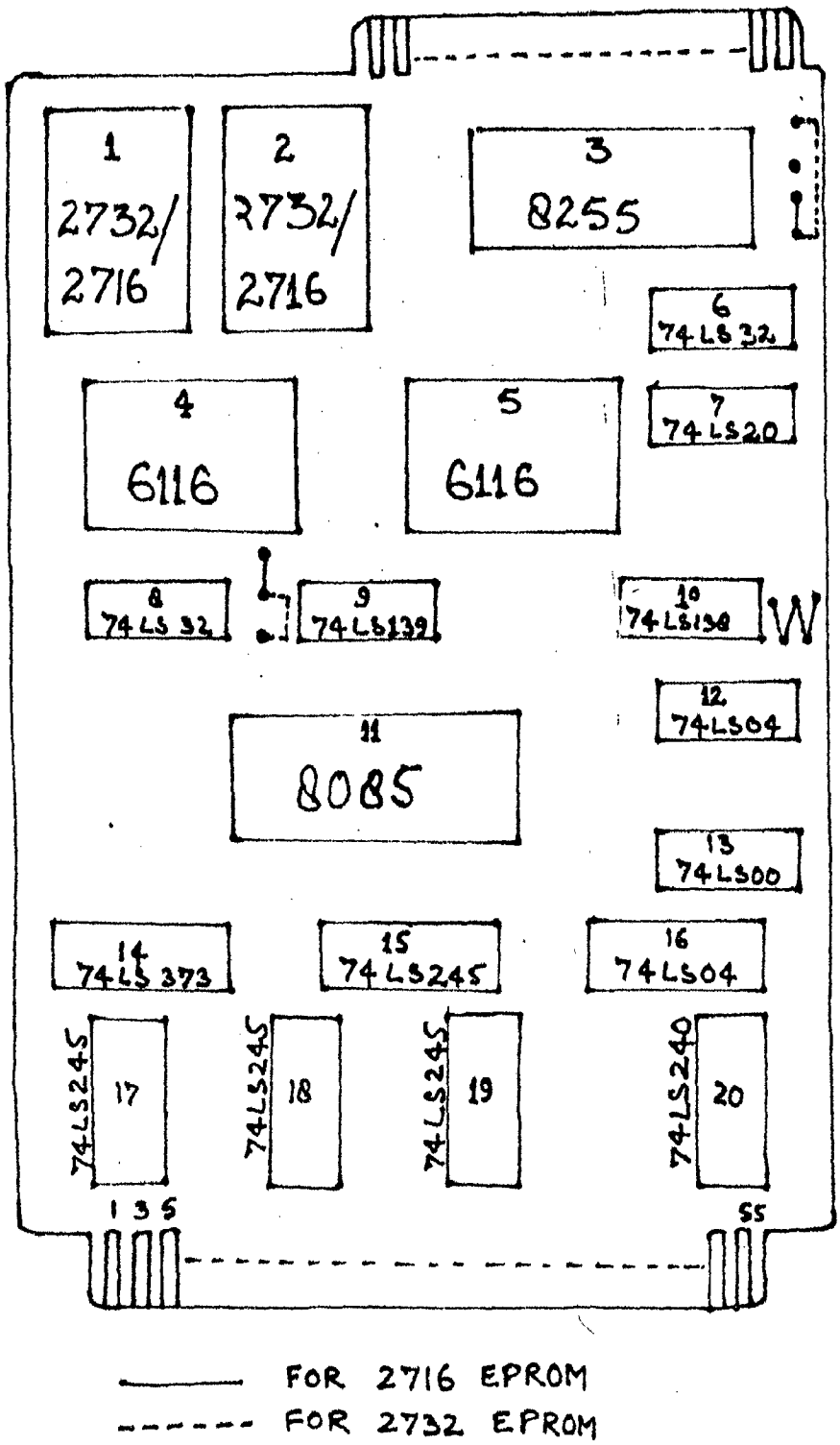


FIG.- 1.4, LAYOUT OF THE VCP-85 CARD

1.5 HARDWARE DESCRPTION

1.5.1 SYSTEM CONFIGURATION

The system is based on three cards and has a power supply module. All these are inhoused in a 19" card cage. All these cards and card cage are based on International STD Bus.

The three cards are namely a microcomputer card, a RAM card and an EPROM card. Three more slots are provided in the card cage to expand the entire system into the required configuration. The block diagram of the system is shown in Fig.1.1. In this dissertation one of these slots is being used by Digital to Analog converter card, which is designed on the international STD Bus so that it may easily be interfaced with the system.

1.5.2 SPECIFICATION OF THE CARDS

The hardware specification of each card is given below,

1.5.2.1 CPU CARD

Ckt. diagram of CPU card is shown in Fig.1.2. it mainly consists of -

1. 8085A at 6.144 MHz clock frequency, microprocessor.
2. 4 K byte EPROM (2716) starting location from 0000 onwards, The first three byte are reserved for monitor program i.e. 0000-00002. In place of 4 K byte (2732) of EPROM chip can also be placed, whose location would be from 0000 to 1FFF.

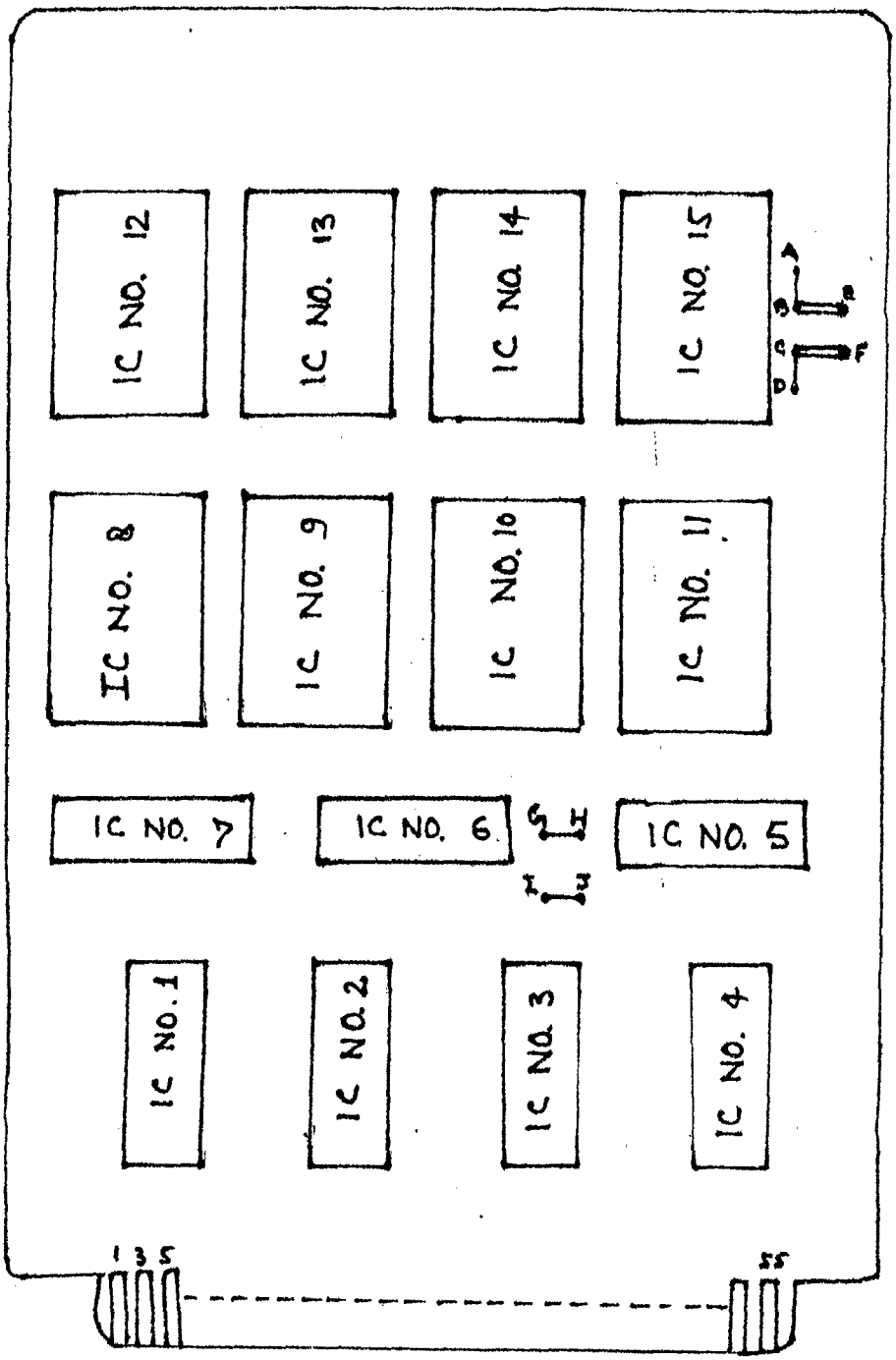


FIG.-1.5 LAYOUT OF THE VMR-85 CARD

NOTE 1: IC SOCKET NO. 8-15 ARE FOR EPROM/RAM (2716/6116)

IC SOCKET NO. 5 IS FOR CONNECTION ADAPTER.

NOTE 2: MEMORY SELECTION JUMPER DETAILS -

JUMPER CONNECTION	MEMORY BLOCK SELECTED	MEMORY TYPE SELECTED
B TO A	UPPER	8K EPROM, 2716
B TO E	UPPER	8K RAM, 6116
C TO D	LOWER	8K EPROM, 2716
C TO F	LOWER	8K RAM, 6116

NOTE 3: CONNECT G TO H AND I TO J FOR NORMAL WORKING. REMOVE THE SHORTINGS FOR USING THE CARD WITH MEMEX SIGNAL.

3. A socket is provided for 24 I/O lines through 8255. The user selectable area is from 00-1C.

4. 4 K byte of CMOS RAM based on 6116. Area allotted to RAM is from F000-FFFF.

5. All address, data and control signals are buffered and available at the STD bus controller.

6. The back edge connector has 2 programmable Input/ Output lines with RST 5.5, RST 6.5, RST 7.5 and SID, SOD available.

7. The decoding is done in such a fashion that no external memory will be selected if the addresses are common between internal and external memory.

Layout of the CPU card is shown in Fig.1.4.

1.5.2.1 RAM CARD

(It's ckt. diagram is shown in Fig.1.3)

1. Based on the STD bus. It can support either 6116 (2K byte CMOS RAM) or 2716 (2K byte EPROM) selectable through jumper in slot of 8K byte.

2. This card can be allotted any memory area from 0000-FFFF just by changing the connections in connection adopter. The connections are incorporated to RAM from 0000-FFFF.

3. All address, data and control bus buffered.

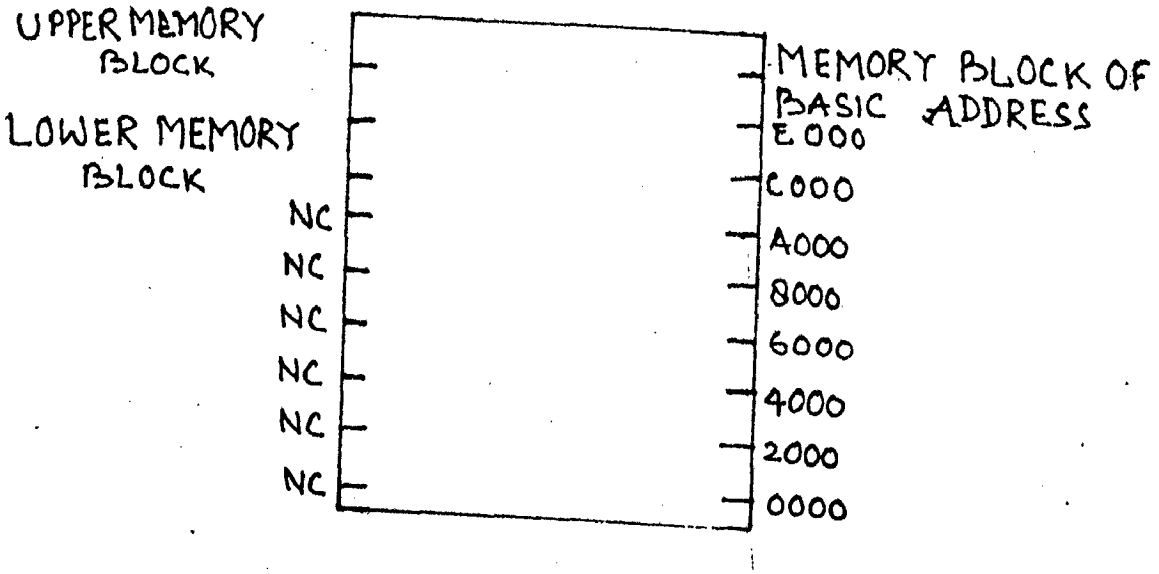
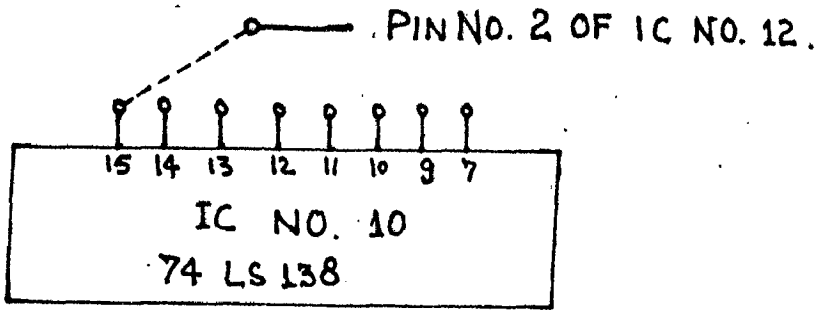


FIG.- 1·6A, BASE ADDRESS SELECTION ON THE CONNECTION ADOPTER



NOTE:------ INDICATES THAT THE OPTION 1 HAS BEEN SELECTED.

FIG.- 1·6B, JUMPER POISITION ON THE CARD.

S.No.	PORT ADDRESS	PORT NAME	REMARKS
1.	00	PORT A	
	01	PORT B	
	02	PORT C	
	03	CONTROL WORD	OPTION 1
2.	04	PORT A	
	05	PORT B	
	06	PORT C	
	07	CONTROL WORD.	OPTION 2
3.	08	PORT A	
	09	PORT B	
	0A	PORT C	
	0B	CONTROL WORD	OPTION 3
4.	0C	PORT A	
	0D	PORT B	
	0E	PORT C	
	0F	CONTROL WORD	OPTION 4
5.	10	PORT A	
	11	PORT B	
	12	PORT C	
	13	CONTROL WORD	OPTION 5
6.	14	PORT A	
	15	PORT B	
	16	PORT C	
	17	CONTROL WORD	OPTION 6
7.	18	PORT A	
	19	PORT B	
	1A	PORT C	
	1B	CONTROL WORD	OPTION 7
8.	1C	PORT A	
	1D	PORT B	
	1E	PORT C	
	1F	CONTROL WORD	OPTION 8

TABLE T-1.1, OPTION FOR 8255 PORT ADDRESS ON VCP-85 CARD

PIN NO.	DESCRIPTION	PIN NO	DESCRIPTION
1.	PA 3	23.	PC0
2.	PA 4	24.	PC1
3.	NOT USED	25.	Vcc.
4.	RST 7·5	26.	PC2
5.	NOT USED	27.	PB7
6.	SOD	28.	PC 3
7.	PA2	29.	PB 6
8.	PA5	30.	PB5
9.	PA1	31.	PB0
10.	PA6	32.	PB1
11.	PA 0	33.	PB2
12.	PA7	34.	PB4
13.	NOT USED	35.	PC 6
14.	NOT USED	36.	PB3
15	NOT USED	37.	PC7
16.	NOT USED	38.	RST 6·5
17.	NOT USED	39.	NOT USED
18.	NOT USED	40.	NOT USED
19.	NOT USED	41.	NOT USED
20	SID	42.	NOT USED
21.	PC 5	43.	NOT USED
22.	PC4	44.	RST. 5 5

PIN ASSIGNMENT OF THE 44 PIN EDGE CONNECTOR ON THE VCP-85 CARD.

TABLE, T-1.2.

4. The back PCD connector has output for TTY (20 mA current loop) and RS-233-C connection and SID, SOD comming from CPU card.

1.5.2.3. EPROM CARD

1. This card is identical to RAM card, except that the jumpers are connected for 2716 to be incorporated. The connection in connection adapter are connected to select the memory from OOOO-BFFF. Layout of the VMR is shown in Fig.1.5.

1.5.2.4 POWER PACK UNIT

1. Easily detachable unit.
2. Rated, for 5V/2.5A, +12V/250 mA, 30/25/100 mA load and line regulation better than 0.1%.
3. Power supply back pannel connections and power supply front panel connections are shown in Fig.1.7.
4. Circuit Diagram of power supply PS-III is shown in Fig.1.8.

1.5.3 CARD DISCRIPTION

1.5.3.1 CPU CARD (VCP-85)

VCP-85 is a microcomputer card based on the 8085 microprocessor. The card is configured around the Internationally accepted STD Bus.

This card provides (as already mentioned in section 1.5.2.1).

UPPER MEMORY BLOCK

IC NO.	STARTING ADDRESS
15.	BAS* + 0000
14.	BAS + 0800
13.	BAS + 1000
12.	BAS + 1800

Note : BAS* - Base Address Selected from Connection Adapter

LOWER MEMORY BLOCK

IC NO.	STARTING ADDRESS
11.	BAS + 0000
10.	BAS + 0800
9.	BAS + 1000
8.	BAS + 1800

TABLE, T-1.3, STARTING ADDRESS OF INDIVIDUAL ICs

1. 4K bytes of CMOS RAM using 6116.
2. 4K/8K bytes of EPROM using 2716/2732.
3. 24 programmable I/O lines using 8255.
4. Buffered data, address, and control lines.

The memory map for the VCP-85 is as follows -

S.NO.	ADDRESSES	RAM/ROM	REMARKS
1.	0000-0FFF	EPROM	
2.	1000-EFFF	RAM/ROM	Area for
3.	F000-FFFF.	RAM	expention

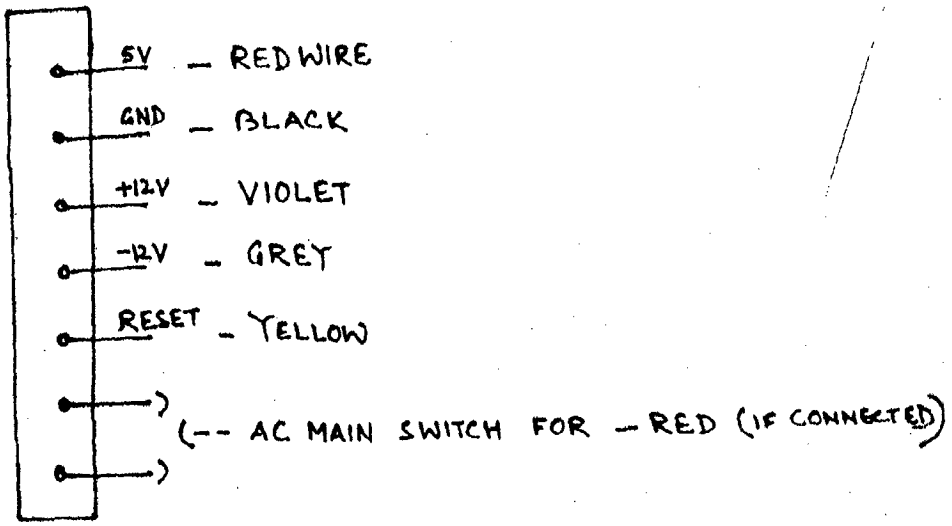
The programmable I/O's provided in the card are I/C mapped. The port addresses for the 8255 on VCP-85 are user selectable. Using jumper, the user can select any one of the eight possible port addresses sets given in Table, T-1.1.

The jumper position on the card are shown in Fig.1.6B. All the markable and unmarkable interrupts have been pulled down by 5.6 K resistances. The NMI (Trap) and INTR are buffered and brought out on the STD bus at pin no. and respectably as per the convention where as the 7.5, 6.5, and 5.5 interrupts are brought out on pin edge connector.

The details of the 44 pin dedge connector are given in Table T-1.2 Layout of the VCP-85 card is shown in Fig.1.4.

1.5.3.2 16K BYTE MEMORY CARD (VMR-85)

The VMR-85 card is also based on the international standard bus. This card can be configured as



POWER SUPPLY BACK PANEL CONNECTIONS

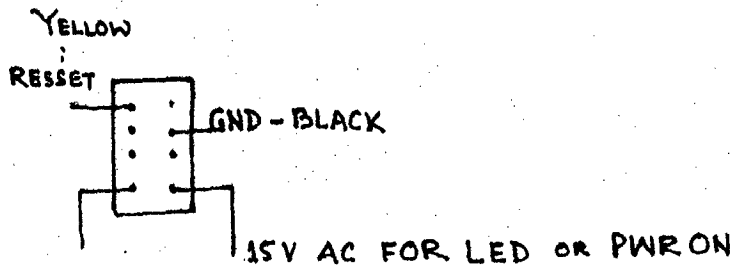
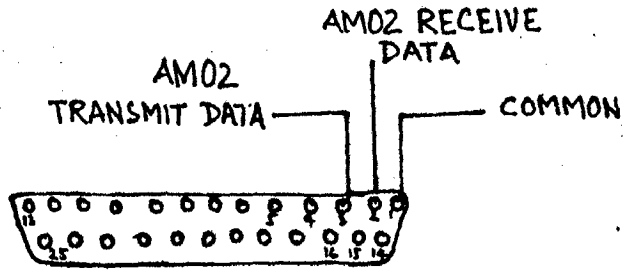


FIG 1.7A POWER SUPPLY FRONT PANEL CONNECTION



RS-232-C

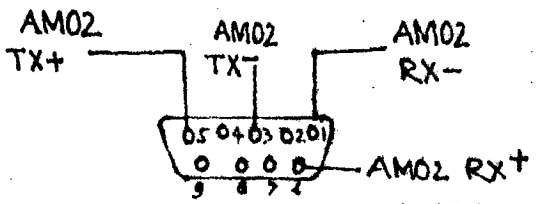


FIG-1.7B, 20 MA CURRENT LOOP

FIG – 1.7

16K byte ROM card (using 6116) or 16 K byte CMOSRAM (using 6116) or 8K byte of RAM plus 8K byte ROM by changing the jumper connection as shown in Fig.1.6A.

In VMR-85, the upper and lower memory block can be defined with different memory address of upper and lower memory blocks can be done by changing the jumper position in the connection adopter at the IC position 5 of Fig.1.5. The base addresses of various memory block selected by the connection adopter are shown in the Fig.1.6A.

The starting address of the individual ICs in a block would be as given in Table T-1.3.

The VMR-85 card can be easily utilised for secondary storage in a 8 bit system where memory to be used in more than 64K byte. The jumper position shown in the Fig.1.5 is to be cut if the VMR-85 card is to be through memory expansion lines.

1.6 SYSTEM SOFTWARE

The VMDS-85 has a powerful system software which includes system monitor commands, assembler, relocation, documentation and PROM Programming Software. The system is not based on floppy disc drive. The assembler and other system software is resident in the system memory. It is a single line double pass assembler i.e. it decodes each line and puts its proper machine code in the operating RAM area. User is allowed to give a symbolic name (address lebel)

to any instruction. Two separate labels are maintained in the system to store the symbolic names.

The system software communicates with the user through SID and SOD lines. A RS-232-C and 20 mA current loop interfaces are provided on the memory card. Its connectors diagrams are shown in Fig.1.9. The CRT terminal or Teletypr-writer is to be connected, in Asynchronous ASCII, one start bit two stop bit and parity off format.

The system can communicate with the terminal at a baud rate from 75 to 9600 bauds. The system itself finds out the baud rate of the device with which it is communicating. After pressing RESET, the baud rate recognizer of the system waits for a character to arrive from which it decides the baud rate (the least significant bit of this first character inputted should be high eg the character like A,C,E,ESC, Cr. etc.). This is required in order to measure the time period of the start bit, from which the VMDS calculates the baud rate. In order to change the baud rate RESET the system and enter the first character.

The system gives a message of AM02 with a prompt sign of '.' on a starting of a next line. Thus '.' signifies that the system is waiting for valid command to be inputted.

The command recogniser does comparing the entered commands alphabet with a stored table. Because all the EPROM available in the market (except 1702) have logic 1 in all bits

when unprogrammed. Comparison is done till a match or FF is encountered. It match is not found and FF is encounted than the system display an error indicated by * and a '.' (Prompt sign) on a next line to wait for a valid command.

This type of recogniser is not limited by any number, but by the memory area left for command. Each command occupies five memory location out of which two are required for command name and three location for jump to service-routine. The command table is alloted RAM area from 8600 to 8FFF, such that about 100 commandss can be incorporated in the system. Each command consists of two alphabets. The two characters are sufficient for the clarity. Command recogniser takes in last two alphebets before carriage return. This allows user to correct any error made in entry.

The system is interactive, that is it asks the type of information to be inputted. As you input a valid command the prompting message are displayed to ask for right type of input information. As the command is executing the appropriate information is displayed or a '.' is displayed on a fresh line signifying that the system is ready to accept a new command.

AMO2 has total of 69 commands, since it is very difficult to remember the symbolic names of all the commands, MENU command has been developed to display the command groups and the commands in a particular group. The user can enter

directly into a particular command through the MENU. All the commands are placed in Appendix-C.

The memory of the system can be expanded completely upto 64K bytes. The memory is configured into lower and upper slot of 32K bytes, each the upper 32K byte slot is used by the system in which the firmware lies from 8000 to BFFF and from C000 to FFFF is the area allotted to RAM. The monitor uses the RAM area of E000 to FFFF. The C000 to DFFF RAM area is left for further expansion but user can develop his program in this area. The F000 to F7FF RAM area is alloeated to operand labels and F800-FFFF area to address labels. As each label takes 8 location (Two location for label address and six location for label name). The total number of labels for each of these two groups can be a maximum of 255. The F000 and F800 have information of no. of operand and address label respectively in their label table. The maximum of 210 address label can be accommodated while entering the program. The remaining space is left for the undefined operand label after getting the definition.

The resident fireware is devided into seven groups, listed below:-

1. Monitor.
2. Relocate object program.
3. Documentation.
4. PROM

5. Assembler
6. Auxiliary storage (to be announced).
7. Miscellaneous.

VALID DELIMETERS:

The coma (,), carriage return (CR), Space (b) are the valid delimeters. The comma and blank act as an intermediate delimeter for a command. CR is a terminator for any entry in a program, whereas ESC acts as a terminator for any command. 'ESC' is a delimeter to make the system return back to the monitor and wait for a new command to be inputted.

Whenever any symbolic name is inputted, the name is searched for in the address label table for a match. If search is successful, the action required is take, else 'SEARCH FAILED' message is printed.

1.7 DATA TERMINAL (VDT-85):

1.7.1 SYSTEM INTRODUCTION

VDT-85 is a interactive terminal designed with operator convenience. It has got a full 12" diagonal non glare CRT screen with capability of displaying full 24 lines of 80 character. The characters are formulated using a TV raster scan technique with 5x7 dot matrix character window to provide a legible crisp and clear display.

VDT-85 provides a detachable keyboard which transmit ASCII standard code on a serial link. The keyboard

uses mechanical switches under software control to provide user's definable codes. It has numeric keypad, editkey pad, function key pad, cursor control keypad etc. The effective keyboard design provides a three wire communication between the keyboard and the display unit.

The terminal interact with computer systems through EIA-RS-232O 20mA current loop interface at baud rate that are programmable. Its connector diagram are shown in Fig.1.7(A). The baud rate, bit structure Half/full duplex, parity etc. are selectable by the keyboard. The top line (25th line) displays the mode of the terminal. The ESC command provides efficient management and editing such as cursor address, insert/delete characters or line, mode of transmission to character/line/page, intenity control and attributes.

The optional parallel printer port and Aux.Rs232-C-port facilities the interfacing of hard copy printers (In this dissertation, multi-channel recorder is used, its details are placed in Appendix-D), magnetic storage devices or chaining of terminals.

1.7.2 SYSTEM SPECIFICATION

Display Unit	12" nonglare screen
Screen Capacity	25 lines x 80 characters including status line).

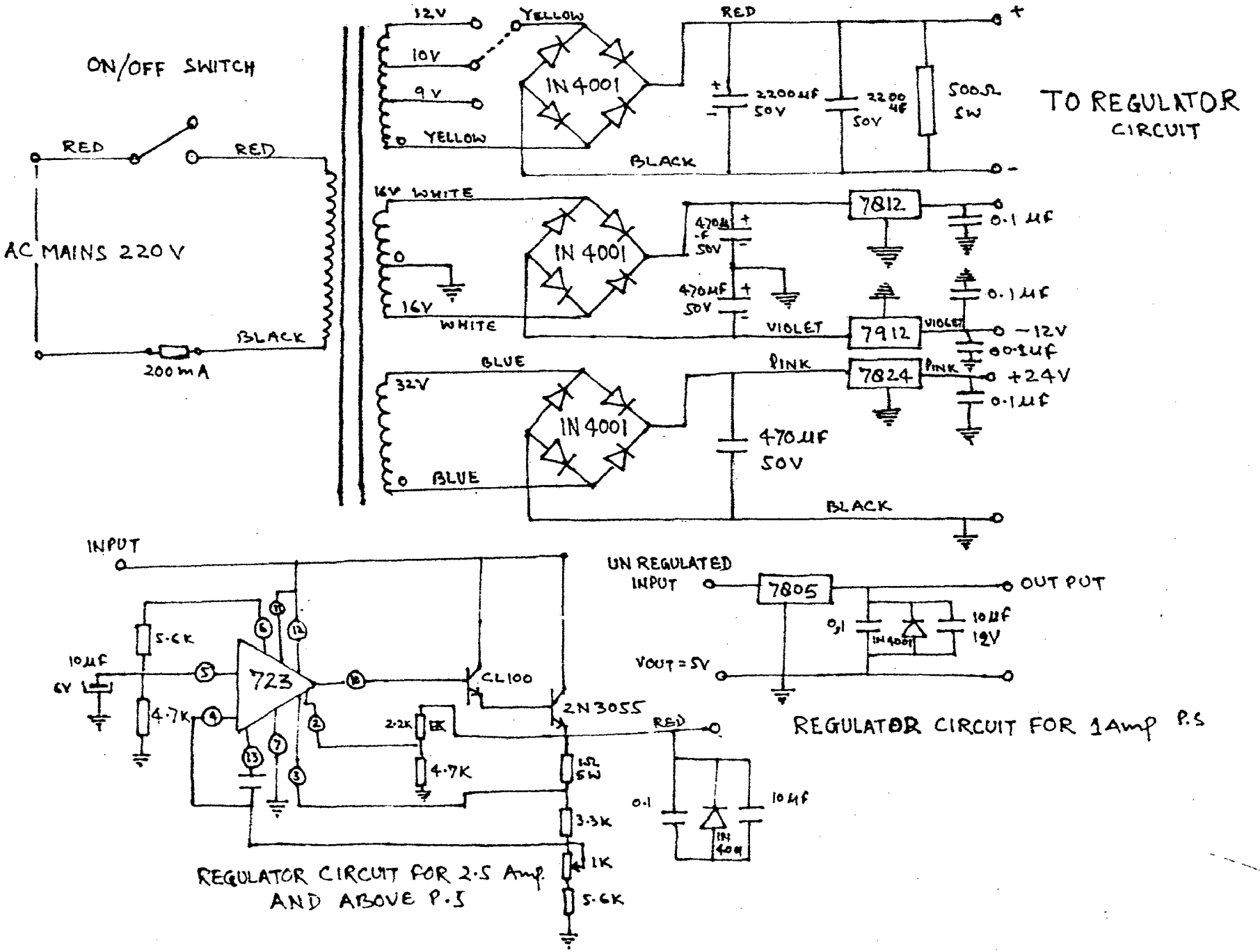


FIG-1.8 CIRCUIT DIAGRAM OF POWER SUPPLY – PS-III

Character Display	5x7 dot matrix light character with dark background or dark characters with light background.
Character Set	96 ASCII characters with lower case.
Refresh Rate	50 Hz.
Bit Structure	Data: 5,6,7, or 8 bits. Start: 1 bit. Stop: 1, 1.5 or 2 bits. Parity: 1 bit.
Parity	Odd/even or no parity.
Baud rate	75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19.2 K (Software selectable).
Transmit/Receive mode	Half/Full duplex.
Visual Attributes	Normal, Blinking, Reverse Video, Underline and Reduced intensity.
Memory	Multi page memory (upto 4 pages)
Keyboard	111 keys keyboard communicate with the display unit through three wi[illegible] transmission.
Cursor Movement	Cursor up, cursor down, cursor home cursor right, cursor left, cursor right up, cursor right down, curso[illegible] left up, cursor left down.
Cursor Addressing	Director cursor addressing.
Display Movements	Roll up/Roll down. Next page/Previous page. Row swap.

TRANSMIT/ERASE	a) Entire memory (all four pages). b) The current displayed page. c) The line on which the cursor is located. d) The character on which the cursor is located. e) From cursor position to end of page. f) From start of page to current cursor position. g) From cursor position to end of line. h) From start of line to current cursor position. i) P1 characters from current cursor position.
Tabulation	Tab set/Clear facility.
User Definable Keys	10 keys F0 to F9
Escape Sequences	Full cursor Control, Memory erase, Tab facility Editing functions, provided through escape sequence.
Interfaces	RS-232-C, 20mA current loop, Printer interface (Optional). Aux RS-232-C (Optional).
Power Input	230V, ± 10%, 50 Hz
Power Consumption	60 Watts, nominal
Operating Temperature	0 to 45°C.

1.7.3 FAMILIARISATION WITH VDT-85

Vinytics' VDT-85 has got a detachable keyboard. The keyboard is connected to the monitor through a cable at the back panel of the monitor. Fig.1.9 shows the details of the various connectors and switches at the back panel. In a normal mode, the terminal communicates with the computer either through RS-232-C (Port Main) or 20mA current loop. Either of this can be selected by a switch provided at the back panel.

On the right hand side of the monitor three controls are provided namely :-

1] Intensity Control.

2] Mains ON/OFF Switch.

3] Switch for selecting LINE or LOCAL mode of operation.

A reset switch has been provided on the left hand side of the keyboard. This switch resets the system to the power on stage.

CHAPTER - 2

DIGITAL TO ANALOG CONVERTER

2.1 INTRODUCTION

The data in a microprocessor is in digital form. This differs from the outside world where data is in analog (continuous) form. After a CPU has processed data, it is often necessary to convert the digital answer into an analog voltage or current. This conversion requires a digital to analog (D/A) converter.

2.2 A BASIC D/A CONVERTER

The op-amp summing circuit can be used to build a D/A converter by selecting input resistors that are weighted in binary progression. Figure 2.1 gives the idea. $V_{(REF)}$ is an accurate reference voltage and the resistors are precision resistors to get accurate input current. The switches can be open or closed. When all switches are open, all input currents are zero and the output current is zero.

2.2.1 ALL BITS HIGH

When all switches are closed, the input currents are

$$I_3 = \frac{V_{(REF)}}{R}, \quad I_2 = \frac{V_{(REF)}}{2R}, \quad I_1 = \frac{V_{(REF)}}{4R}, \quad I_o = \frac{V_{(REF)}}{8R}$$

The output current with all switches closed is the sum of all input currents and equals

$$I = \frac{V_{(REF)}}{R} (1 + .5 + .25 + .125) \qquad \ldots [2.1]$$

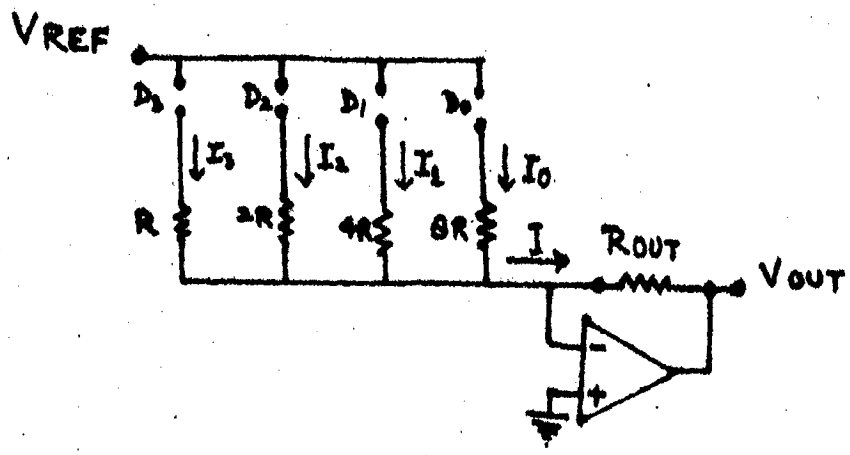


FIG.-2.1 D/A CONVERSION WITH BINARY-WEIGHTED RESISTORS.

D_3	D_2	D_1	D_0	OUTPUT CURRENT mA	FRACTION OF MAXIMUM
0	0	0	0	0	0
0	0	0	1	0·125	1/15
0	0	1	0	0·25	2/15
0	0	1	1	0·375	3/15
0	1	0	0	0·5	4/15
0	1	0	1	0·625	5/15
0	1	1	0	0·75	6/15
0	1	1	1	0·875	7/15
1	0	0	0	1	8/15
1	0	0	1	1·125	9/15
1	0	1	0	1·25	10/15
1	0	1	1	1·375	11/15
1	1	0	0	1·5	12/15
1	1	0	1	1·625	13/15
1	1	1	0	1·75	14/15
1	1	1	1	1·875	15/15

TABLE T2-1. WEIGHTED D/A CONVERTER.

$$I = 1.875 \frac{V_{(REF)}}{R}$$

By opening and closing switches we can produce 16 different output currents from 0 to 1.875 $V_{(REF)}/R$.

2.3 THE LADDER METHOD

One way to get around the problems of a binary weighted resistors is to use a ladder circuit Figure-2.2(a) is an example of R-2R ladder commonly used in integrated D/A convertors.

Only two resistance values are needed. This eliminates the range problem. Furthermore, since the resistors are on the same chip, they have almost identical characteristics, this minimizes the tolerance problem. In other words, as the number of bits increases, an integrated ladder can divide the current much more accurately than a binary weighted circuit.

2.3.1 LADDER PROPERTIES

An R-2R ladder does something interesting to the impedence at different points in the circuit. To begin with, the two resistors at node D in Fig.2.2(a) are in parallel and may be reduced to an equivalent resistance R, shown in Fig.2.2(b). Now, to the right of node C we have R in series with R, a total of 2R. Since node C has 2R is in parallel with 2R, the circuit reduces to Fig.2.2(c).

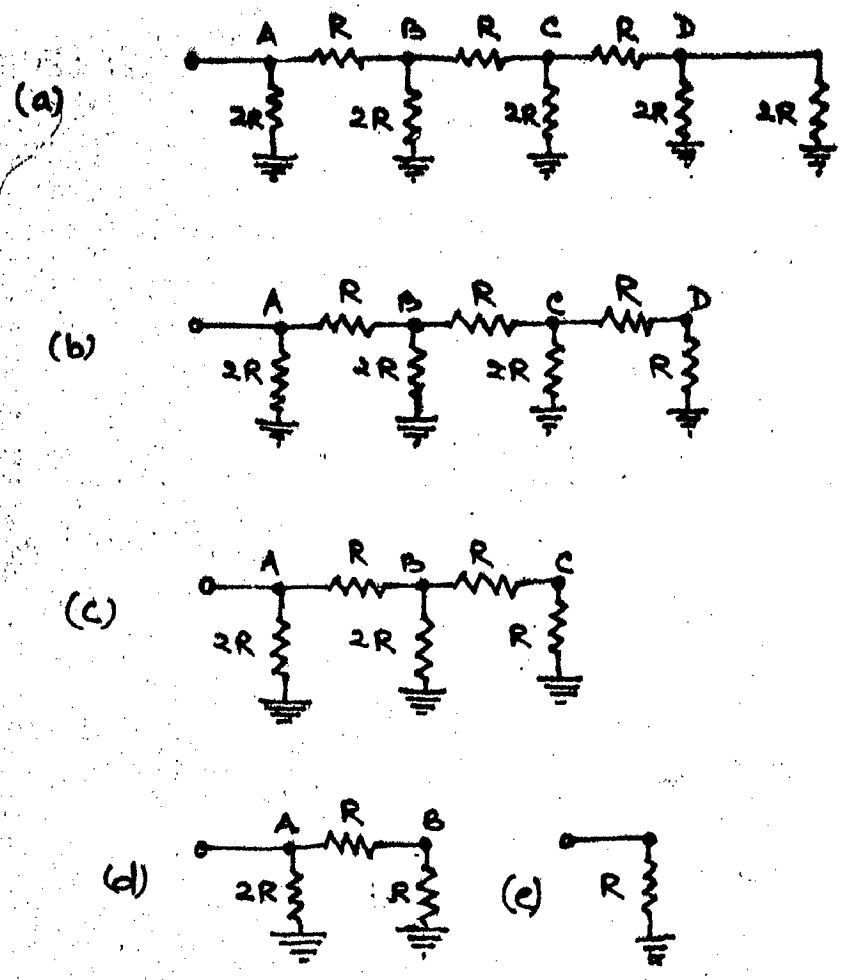


FIG-2.2 R-2R LADDER

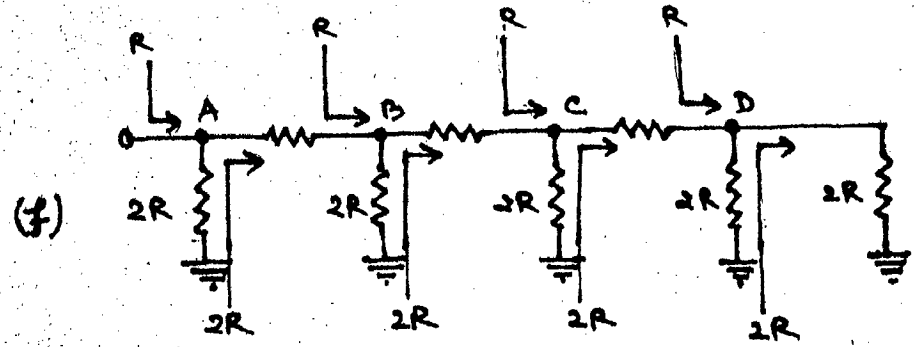


FIG-2.2(f) LADDER IMPEDANCES

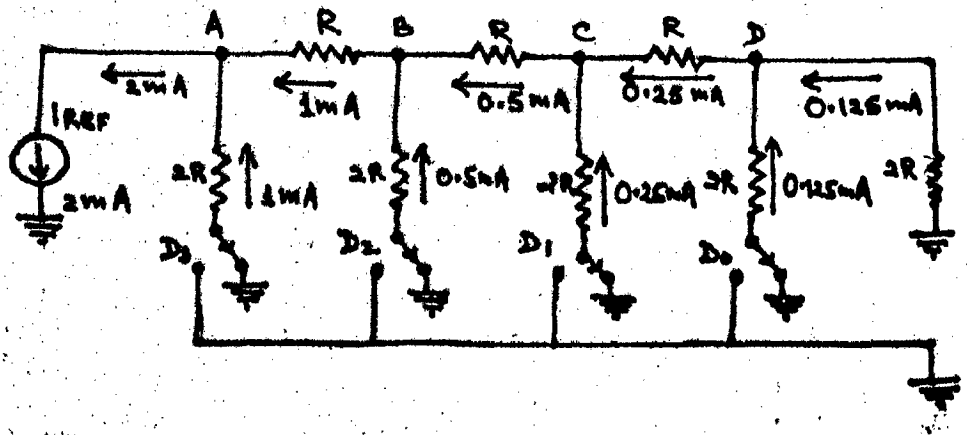


FIG.-2.3, D/A CONVERSION WITH R-2R LADDER.

Looking into the left side of node B (Fig.2.2(c), 2R is in parallel with 2R. Therefore, the circuit reduces to Fig. 2.2(d). Again, 2R is in parallel with 2R, so the circuit reduces to the single R shown in Fig. 2.2(e).

Fig.2.2(f) summarizes ladder impedances. Looking into the left side of a node, we always have an equivalent resistance of R. Just to the right of each node, we always see a resistance of 2R. This impedance phenomenon is the key to analyzing modern D/A converter because they use the ladders instead of weighted resistors.

2.3.2 BINARY DIVISION OF CURRENT

Figure 2.3 shows how a ladder can divide the current into binary levels. The typical D/A convertor has a reference current set by the user. Here, the reference current is 2 mA. The bottom of each 2R resistor is grounded in either switch position. When a switch is to the right, the current through a 2R resistor flows to the upper ground when a switch is to the left, the lower ground sinks the current. With all the switches to the right, as shown in Fig.2.3, IOUT is zero.

Here is how the ladder divides the 2 mA of reference current. Just to the right of node A, an equivalent resistance of 2R is seen. Therefore, the 2 mA of input current divides equally at node A. Similarly, at node B, 2R is in parallel with 2R, again, the current

divides equally into 0.5-mA branch currents. This process continues through the ladder, so that we wind up with the upper grounds sinking 1, 0.5, 0.25, and 0.125 mA.

2.3.3 OTHER SWITCH POSITION

When we move the switches, we do not change the way the current divides at the nodes. It still divides equally at each node. But when a switch is to the left, it steers the current into the lower ground. Bits D_3 to D_0 control the transistorized switches.

$$I_{OUT} = (D_3 + 2^{-1}D_2 + 2^{-2}D_1 + 2^{-3}D_0)\ \frac{I_{(REF)}}{2} \qquad \ldots [2.2]$$

Therefore, the output current of a 4-bit ladder is from 0 to 15/16 $I_{(REF)}$.

2.3.4 MORE BITS

A similar analysis is applied to longer ladders. The output current is

$$I_{OUT} = (D_{n-1} + 2^{-1}D_{n-2} + \ldots\ldots + 2^{1-n}D_0)\ \frac{I_{(REF)}}{2} \qquad \ldots [2.3]$$

For instance, an 8-bit ladder produces a maximum output current of 255/256 $I_{(REF)}$. The LSB increment is 1/255 $I_{(REF)}$.

2.3.5 STEER CURRENT

Current steering is seemed more complicated than necessary, but there is good reason for it. The currents throughout the ladder remain constant all that changes

are the ground points. Constant current implies constant voltage, which means that stray capacitance in the ladder has little effect. In other words, exponential charge and discharge associated with a change in voltage are not received. This reduces the settling time. For this reason, IC converters often use the current steering approach shown in Fig. 2.3.

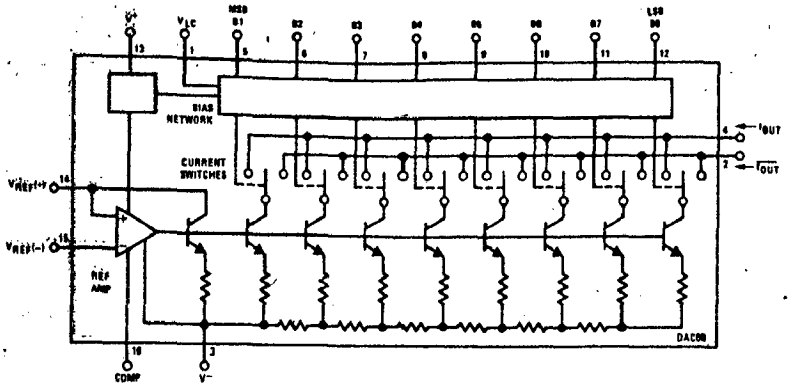
2.4 THE DAC CHIPS

There are many commercially available D/A converter. The least expensive have resolution of 8 to 12 bits. The most expensive have resolutions of 16 to 18 bits. Almost all are monotonic with less than $\pm$ 1/2 LSB error at each output level. Our card is based on DAC 0800 chip. This is 8-bit DAC. This inexpensive and widely used 8-bit D/A converter contains a reference current source, an R-2R ladder, and eight transistor switches to steer the binary currents. The typical applications of DAC-08 are given from Fig. 2.6 to Fig. 2.12. And its Block Diagram and Equivalent Circuit are given in Fig.2.4.

2.4.1 GENERAL DESCRIPTION OF DAC 0800

The DAC 08 is a monolithic 8-bit high speed current output digital-to-analog converter (DAC) featuring typical settling times of 100 ns. When used as a multiplying DAC monotonic performamce over a 40 to 1 references current range is possible. The DAC 08 also features high compliance

Block Diagram



Equivalent Circuit

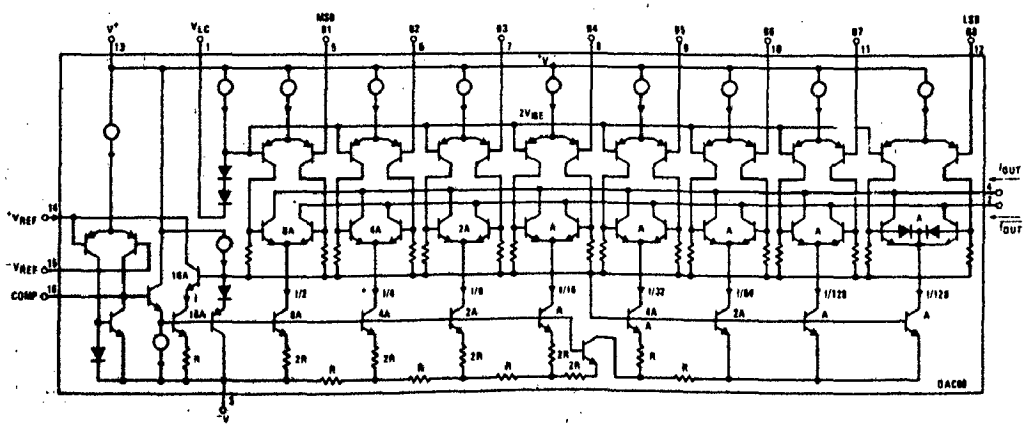


FIGURE - 2.4, DAC-08

complementary current outputs to allow differential output voltage of 20 V_{p-p} with simple resistor loads as shown in Fig.2.6. The reference to full scale current matching of better than ± LSB eliminates the need for full scale trims in most applications while the nonlinearities of better than ± 0.1% over temperature minimizes system error accummulations.

The noise immune inputs of the DAC 08 will accept TTL levels with the logic threshold pin, V_{LC}, pin 1 grounded. Simple adjustments of the V_{LC} potential allow direct interface to all logic families. The performance and characteristics of the device are essentially unchanged over the full ± 4.5 V to ± 18 V power supply range, power dissipation is only 33 mW with ± 5 V supplies and is independent of the logic input states. The DAC 0800L is directly replacement for the DAC 08.

The top view of DAC-08 is shown in Fig.2.5.

2.4.2. FEATURES

1.	Fast settling output current	100 ns
2.	Full scale error	± 1 LSB
3.	Nonlinearity over temperature	± 0.1%
4.	Full scale current drift	± 10 ppm/°C.
5.	High output compliance	-10V to +18V
6.	Complementary current outputs	
7.	Interface directly with TTL, CMOS, PMOS and others	
8.	2 quadrant wide range multiplying capability.	

CONNECTION DIAGRAM

DUAL-IN-LINE PACKAGE

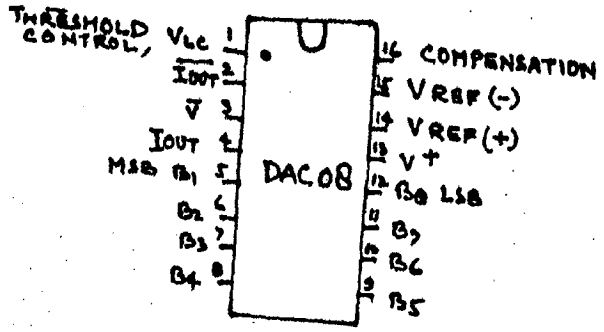


FIG-2.5 TOP VIEW

TYPICAL APPLICATIONS -

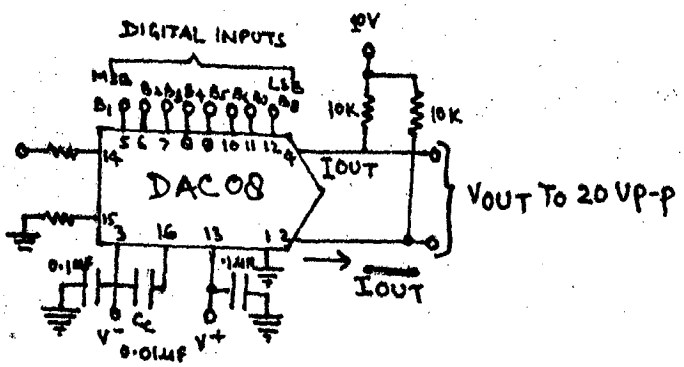


FIG.-2.6 20 ± VP-P OUTPUT DIGITAL-TO-ANALOG CONVERTER.

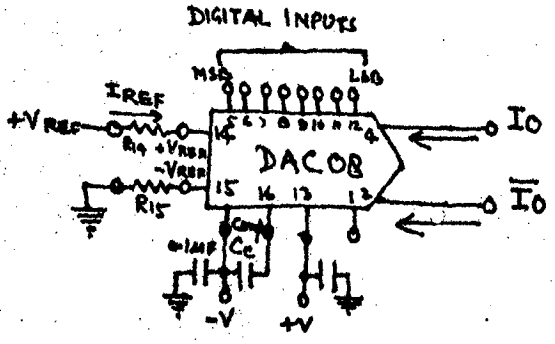


FIG.-2.7 BASIC POSITIVE REFERENCE OPERATION

$$I_{FS} \approx \frac{+V_{REF}}{R_{REF}} \times \frac{255}{256}$$

$I_O + \overline{I_O} = I_{FS}$ for all logic states.

For fixed reference, TTL OPERATION

TYPICAL VALUES ARE:

$V_{REF} = 10.000V$

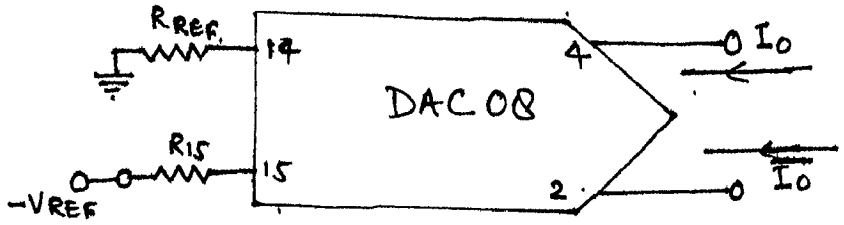
$R_{REF} = 5.000K$

$R_{15} \approx R_{REF} (R_{14})$

$C_C = 0.01\,\mu F$

$V_{LC} = 0V$ (GROUND)

TYPICAL APPLICATIONS (CONTINUED) —



$I_{FS} \cong \frac{-V_{REF}}{R_{REF}} \times \frac{255}{256}$

Note - R_{REF} sets I_{FS}; R for bias current cancellation.

FIG-2.8 BASIC NEGATIVE REFERENCE OPERATION

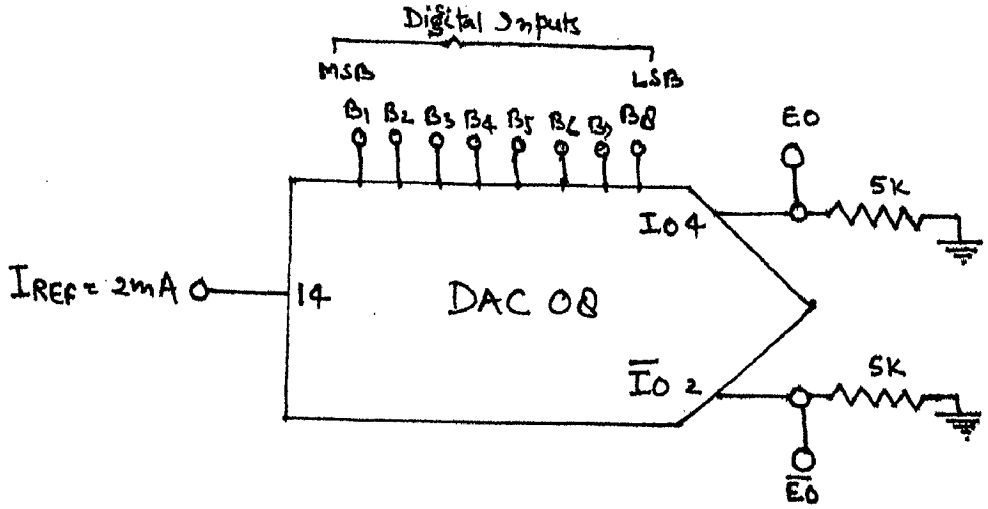


FIG-2.9 BASIC UNIPOLAR NEGATIVE OPERATION.

	B1	B2	B3	B4	B5	B6	B7	B8	Io mA	$\overline{Io}$ mA	eo	$\overline{eo}$
FULL SCALE	1	1	1	1	1	1	1	1	1.992	0.000	-9.960	0.000
FULL SCALE -LSB	1	1	1	1	1	1	1	0	1.984	0.008	-9.920	-0.040
HALF SCALE +LSB	1	0	0	0	0	0	0	1	1.008	0.984	-5.040	-4.920
HALF SCALE	1	0	0	0	0	0	0	0	1.000	0.992	-5.000	-4.960
HALF SCALE -LSB	0	1	1	1	1	1	1	1	0.992	1.000	-4.960	-5.000
ZEROSCALE +LSB	0	0	0	0	0	0	0	1	0.008	1.984	-0.040	-9.920
ZERO SCALE	0	0	0	0	0	0	0	0	0.000	1.992	0.000	-9.960

TABLE-T2-2 , BASIC UNIPOLAR NEGATIVE OPERATION.

TYPICAL APPLICATION (CONTINUED) –

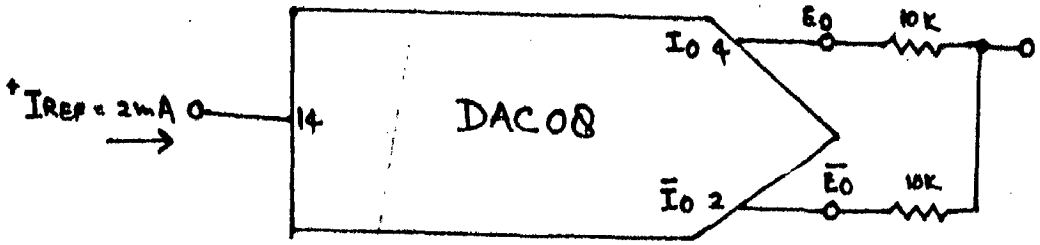


FIG.- 2·10, BASIC BIPOLAR OUTPUT OPERATION

	B1	B2	B3	B4	B5	B6	B7	B8	EO	ĒO
POSITIVE FULL SCALE	1	1	1	1	1	1	1	1	-9·920	+10·000
POSITIVE FULL SCALE -LSB	1	1	1	1	1	1	1	0	-9·840	+9·920
ZERO SCALE +LSB	1	0	0	0	0	0	0	1	-0·080	+0·160
ZERO SCALE	1	0	0	0	0	0	0	0	0·000	+0·080
ZERO SCALE -LSB	0	1	1	1	1	1	1	1	+0·080	0·000
NEGATIVE FULL SCALE +LSB	0	0	0	0	0	0	0	1	+9·920	-9·840
NEGATIVE FULL SCALE	0	0	0	0	0	0	0	0	+10·000	-9·920

TABLE – T2·3, BASIC BIPOLAR OUTPUT OPERATION.

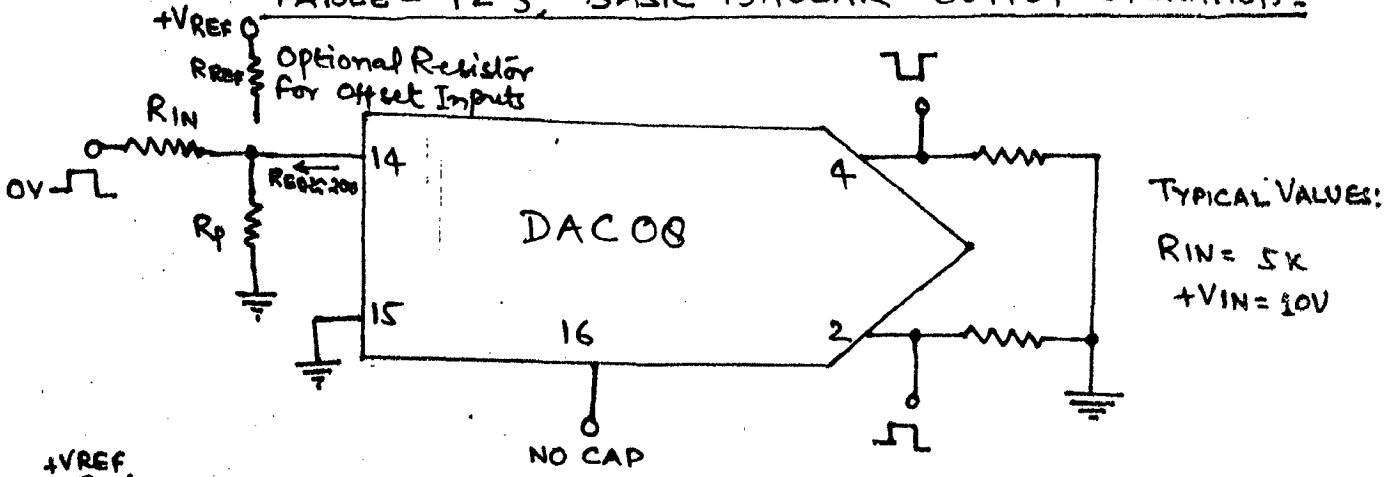
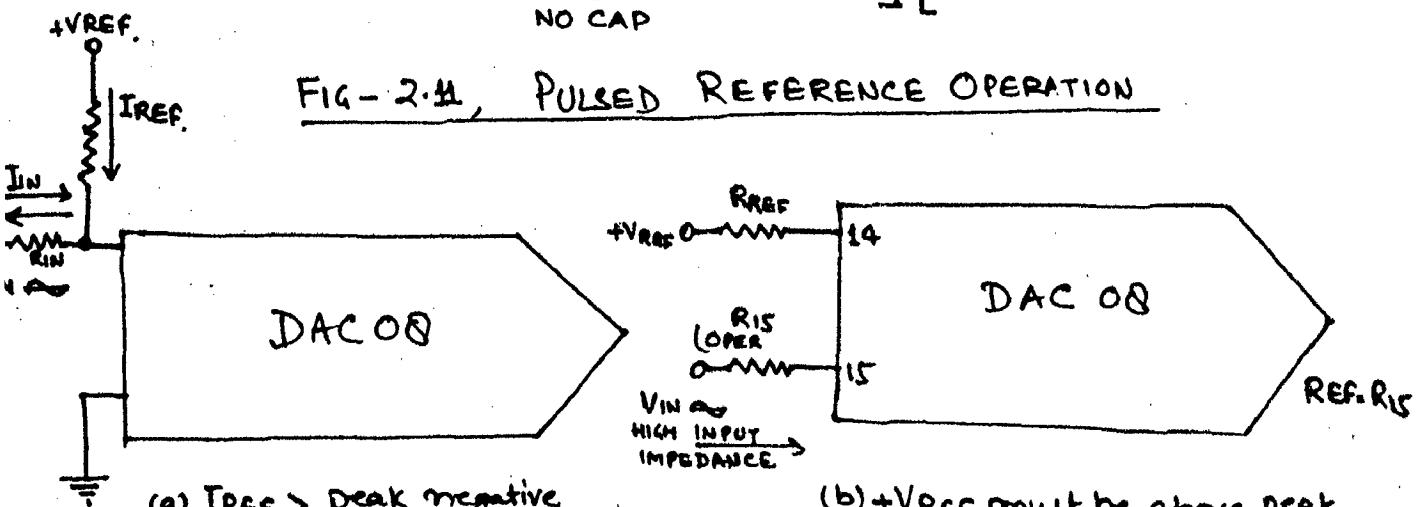


FIG – 2·11, PULSED REFERENCE OPERATION



(a) $I_{REF} \geq$ peak negative swing of I_{IN}

(b) $+V_{REF}$ must be above peak positive swing of V_{IN}

FIG. 2·12 ACCOMMODATING BIPOLAR REFERENCE

9. Wide power supply range	±4.5V to ±18V
10. Low power consumption	33 mW at ± 5v
11. I ow Cost	

2.4.3 ABSOLUTE MAXIMUM RATINGS

1. Supply voltage	±18V or 36
2. Power Dissipation	500 mW
3. Reference Input Differential Voltage (V_{14} to V_{15})	V^- to V^+
4. Reference Input Common-Mode Range (V_{14} to V_{15})	V^- to V^+
5. Reference Input current	5 mA
6. Logic Inputs	V^- to V^- plus 36V
7. Storage Temperature	-65°C to +150°C
8. Lead Temperature (Soldering,10 seconds)	300°C

2.4.4. OERATING CONDITIONS

TEMPERATURE (TA)	MIN	MAX	UNITS
DAC 0800L, LMDAC08	-55	+125	°C

2.5 DIGITAL TO ANALOG CARD

The card is based on Digital to Analog converter chip DAC-0800 whose connection diagram is shown in Fig.2.5 and its block diagram and equivalent circuit are shown in Fig.2.6. The salient feature of the card are

1] Dual channel latched Digital to Analog Converter

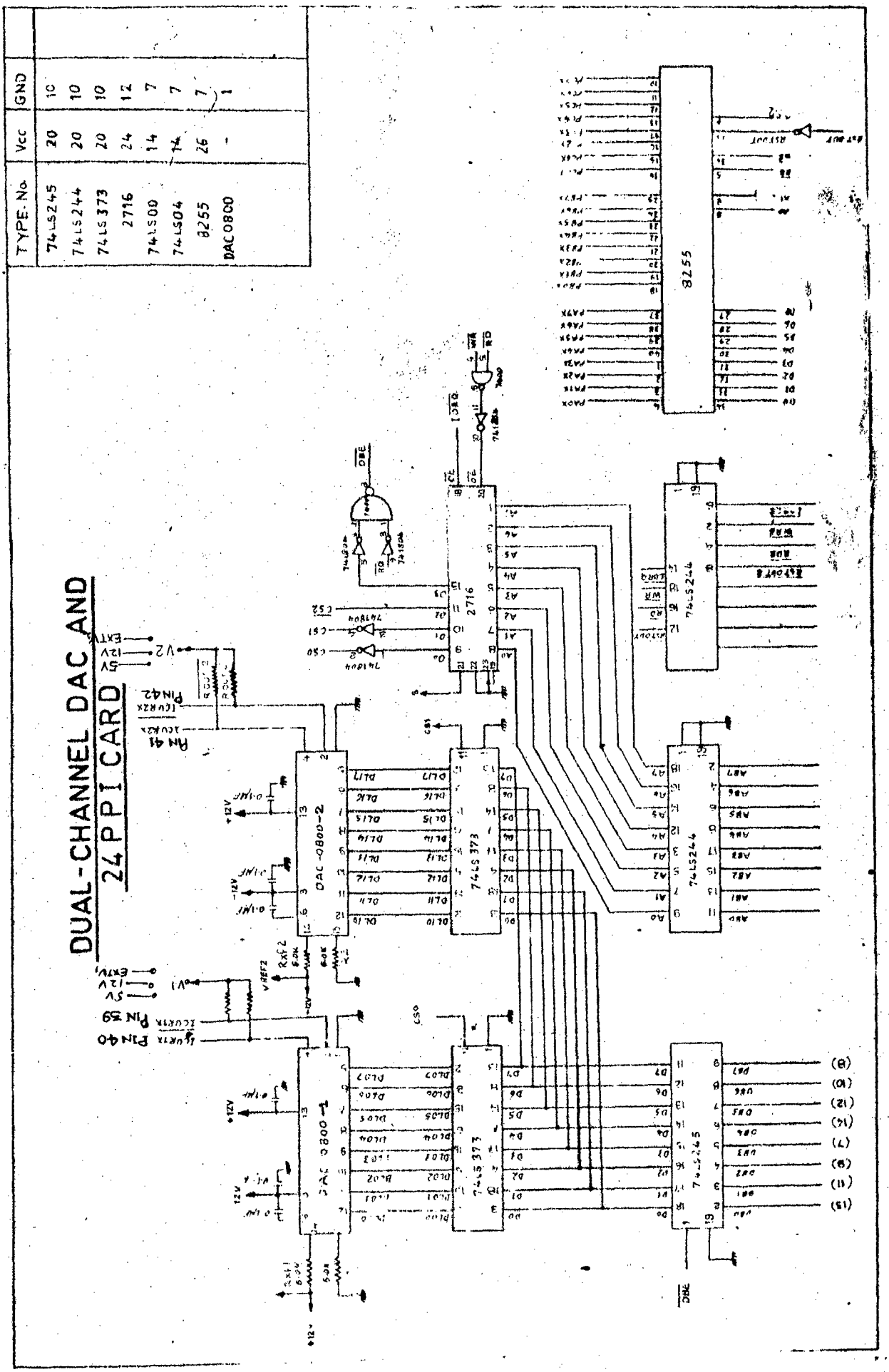
2] 24 Programmable Input/Output lines

3] I/O Mapped

4] Programmed address decoding through 2716 EPROM.

TYPE No.	Vcc	GND
74LS245	20	10
74LS244	20	10
74LS373	20	10
2716	24	12
74LS00	14	7
74LS04	14	7
8255	26	7
DAC0800	-	1

DUAL-CHANNEL DAC AND 24 PPI CARD



2.5.1 CIRCUIT DESCRIPTION

The card consists of bidirectional data bus buffer 74LS245. The direction of bus is controlled through DBE (Data Bus Enable). The address and control signals are buffered through 74LS244. The data for a particular channel is latched through octal transparent latch, the 74LS373 the latch signals for both the channels generated through PROM based decoding technique.

The data lines of DAC-0800 are connected to 74LS373. To give the output current of the range 0-2 mA, the reference current should be 2 mA.

$$I_{FS} = \frac{+V_{(REF)}}{R_{(REF)}} \times \frac{255}{256}$$

$$I_o + \bar{I}_o = I_{FS} \text{ for all logic states.}$$

For fixed references:

$$V_{(REF)} = +12V, \quad R_{(REF)} = 6\ K\ , R = R_{(REF)}$$

The output is available in the form of the current which can be converted into the equivalent voltage by connecting the ROUT and R_{OUT} resistance. The maximum range depends upon V_1 and V_2 voltages. The V_1 and V_2 voltages can be connected through jumpers to +5V, +12V or external voltage. (In our card V_1 and V_2 voltage are connected through jumpers to +5V) Refer Fig.2.13 for position of jumpers and placing of $R_{OUT\ 1}$, $R_{\overline{OUT\ 1}}$, $R_{OUT\ 2}$ and $R_{\overline{OUT\ 2}}$.

PIN NO.	DETAIL	PIN NO.	DETAIL
1.		21	PC5
2.		22	PC6
3.		23	PC7
4.		24	PA7
5.		25	PA6
6.		26	PA5
7.		27	PA4
8.	PB7	28	PA3
9.	PB6	29	PA2
10.	PB5	30	PA1
11.	PB4	31	PA0
12.	PB3	32-38	NOT USED
13	PB2	39	I CUR 2X
14	PB1	40	$\overline{\text{I CUR 2X}}$
15	PB0	41	ICUR 1X
16	PC3	42	$\overline{\text{ICUR 1X}}$
17	PC2	43	V_1
18	PC1	44	V_2
19	PC0	45	NOT USED
20	PC4	46-64	NOTUSED

TABLE-T2.3, DETAILS OF THE 64-PIN CONNECTOR.

The card is I/O mapped and can be selected from any port address defined from 00-FF. This is achieved by PROM decoding technique. The port address lines A0 to A7 are connected to EPROM 2716 and the output lines, O0, O1, O2, O3 are connected to chip selects of DAC-0800-1, DAC-0800-2, 8255 and card select line respectively. To select any I/O device the particular output line should be low alongwith the card select line (O3) at the desired port address location. The following table shows how to arrive at a code no. to select the desired I/O devices.

O7	O6	O5	O4	O3	O2	O1	O0	Code	Device Selected
1	1	1	1	0	1	1	0 (CS0)	F6	DAC-0800-1 with CS0
1	1	1	1	0	1	0 (CS1)	1	F5	DAC-0800-2 with CS1
1	1	1	1	0	0 (CS2)	1	1	F3	8255 with CS2

The desired port address can be any number from 00-FF. These codes are programmed in EPROM 2716. The four port addresses are required for programming 8255, so the four port addresses should have the code F3. The programmed EPROM is for.

$\overline{CS0}$	at	48	(DAC-0800-1)
$\overline{CS1}$	at	49	(DAC-0800-2)
$\overline{CS2}$	at	4A	(Port C of 8255)
$\overline{CS2}$	at	4B	(Control line)

CS2	at	4C	(Port A of 8255)
CS2	at	4D	(Port B of 8255)

Refer Fig. 2.13 for selecting $V_{(REF)}$ (For both the channel) and $R_{(REF).1}$, R1, $R_{(REF).2}$, R2.

The I/O lines of the PPI and the output of the DAC channels are available at the 64 pins connector. The details of the connector is given in Table 2.4.

...

CHAPTER - 3

SET POINT GENERATOR

3.1 INTRODUCTION

Set Point Generator generates the cycle as shown in Fig.3.1. This cycle may be of temperature, pressure, humidity, etc. as per test. To find out the reliability of equipments and components, environmental tests are necessary. The object of this is to determine the effects (thermal, mechanical, chemical, electrical etc.) produced on equipments and components as a result of exposure to environment conditions experienced at the surface of earth.

3.2 APPLICATION FIELD OF SET POINT GENERATOR

1. Environmental Test Chamber (Temperature, Humidity, Pressure, Vacuum, Cooling etc.).
2. Control of Furnace and oven operation.
3. Heat treatment of metals (Annealing, Hardening, Tempering etc.).
4. Cyclic change in voltage and current.
5. And where the parameter are being changed in a cyclic order (with the help of respective sensor or transducer).

3.3 DURATION OF CYCLE

Generally tests are done over 24 hours duration of cycle (complete). This cycle may be taken less or

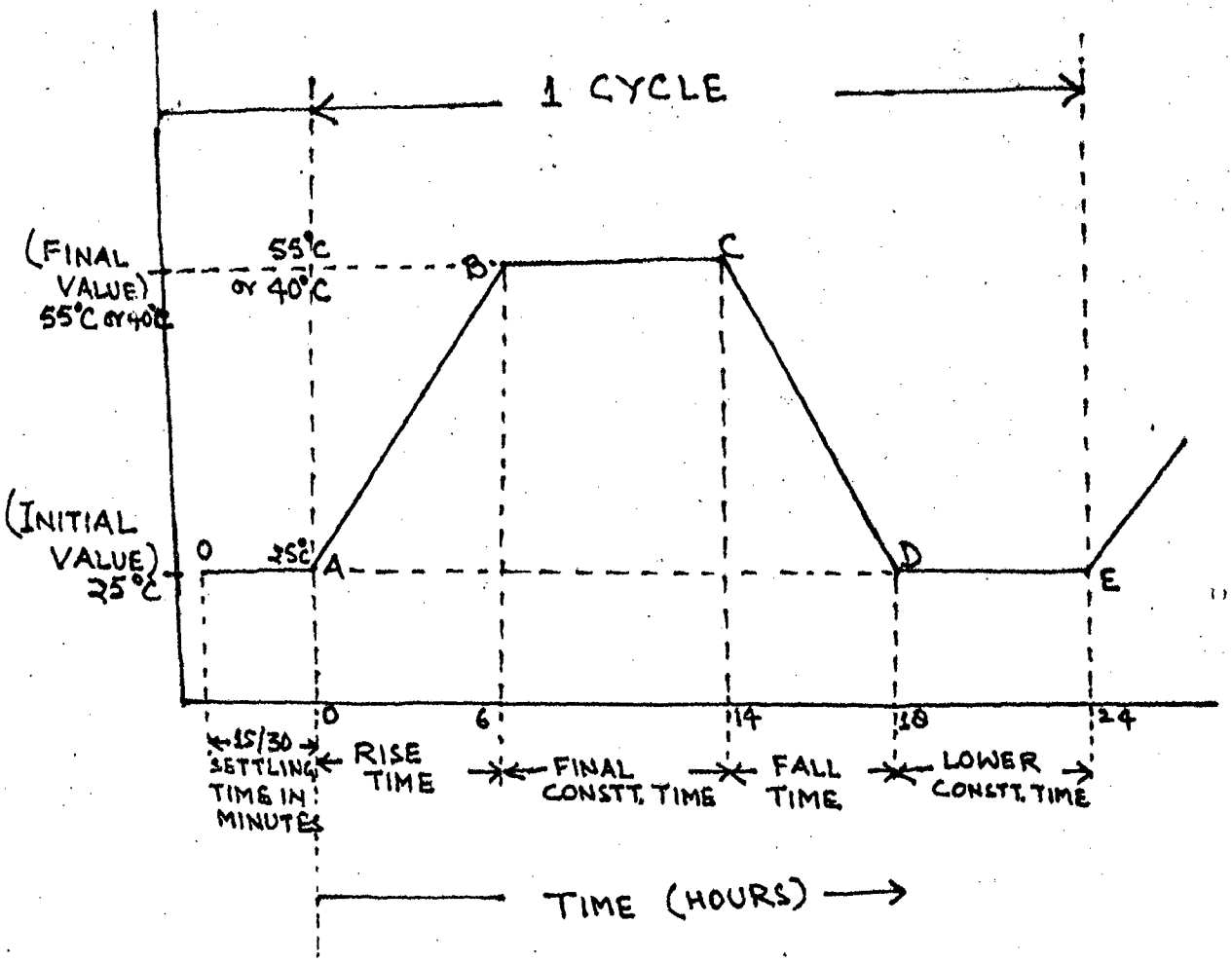


FIG.-3.1, WAVEFORM OF SET POINT GENERATOR

more duration of time. This Set Point Generator may generate one complete cycle in minimum of 4 minutes to maximum of many days (16 days approximately). Here this wave form is taken for Environmental Test Chamber.

3.4 GETTING ACQUAINTED WITH ENVIRONMENTAL TEST CHAMBER

The Environmental Test Chamber is designed and fabricated to conduct the dry heat test, accelerated damp heat test, cool test and pressure tests.

The chamber is doubled walled constructed with 200 mm thick glass wool insulated with vapour barrier all over the chamber. The interior work space is argon arc welded to with stand all temperature.

To avoid the any damage during the pressure test the whole work space is reinforced by M.S. flats. A single front opening doubled walled door with interior lined (stainless steel) is provided. At the centre of the door a 30 cms dia 2 pane toughened glass with low voltage heater is provided. Heavy duty stainless steel wheel type 8 Nos. of hinges are provided to the door for positive locking and to maintain the specified vacuum inside the work space. Double sylicone rubber gasket i.e. one on the chamber and other on the door are provided for leak proofness. At the left hand side of the chamber one 25 mm dia post hole is provided for external connection. For uniform conditions of temperature and humidity inside the work

space 12" dia 17 Nos. of blades stainless steel fan with 1425 RPM, 3 phase induction motor is provided.

All the refrigeration system is housed below the working space on trolley with grills for air ventilation. Over temperature safety thermostat is provided to shut off the chamber in case of overshoots of temperature more than set value. The work space temperature is monitored by PT-100 sensor and maintained by rate programmers.

3.4.1 SPECIFICATION.

1.	Temperature Range	: -40°C TO 200°C
2.	Temperature Tolerance	: ± 3°C
3.	Humidity Range	: 20% RH TO 25% RH
4.	Humidity Tolerance	: ± 3% RH
5.	Work Space	: 500 MW x 750 MMH x 650 M
6.	Temperature Control	: By solid state Digital Rate Programmer for dry bulb and wet bulb temperature.
7.	Temperature Sensors	: PT-100 Duplex Probes
8.	Temperature Gradient	: ± 2°C At any set poi inside the work space.
9.	Temperature Recorder	: i) Pressure below atmospheric by vacuum pump and switch and indicating gauge.
		ii) Pressure above atmosphere by pressure switch which is connected to air compressor

10.	Temperature Recorders	: By solid state strip chart recorder.
11.	Power Requirement	: 415 Volts, 50 C/S, 3 phase neutral, 15 amps/phase A.C. supply.

3.5 WAVEFORM OF SET POINT GENERATOR

3.5.1 SETTLING TIME

As shown in Fig. 3.1, OA is represented as settling time. This time is set for 15 minutes or 30 minutes in our particular wave shape. Initial value is taken as a room temperature. If given initial value is slightly different from room temperature than in settling time of 15 minutes or 30 minutes, temperature is adjusted.

NOTE:- Value given to the terminal by the user for settling time is such that, it (Set Point Generator) would accept either 15 or 30 minutes. If value given is 30 or more than 30, would be taken as 30 minutes by the generator. If the value given is less than 30, would be taken as 15 minutes by the Set Point Generator.

3.5.2 INITIAL VALUE

This value is room temperature value. After taking room temperature, this value is given. If this value is slightly different from room temperature than it would be adjusted in settling time of 15 minutes or 30 minutes.

3.5.3 FINAL VALUE

This value is taken higher than initial value. Generally as 40°C or 55°C, otherwise Set Point Generator does not accept and 'Not acceptable enter again' error message comes up over the terminal. This value may be taken upto 64°C as our requirement is upto 55°C. If we give more than it, error message will be printed up.

3.5.4 RISE TIME

This value is given in hours and minutes separately So rise time may be in fraction of hours also. Terminal asks separately the hours and minutes values in decimal. It raises the temperature from initial value to final value at the rate of as user have given through the ASCII Keyboard of system. It is represented by AB in Fig.3.1.

-3.5.5 FINAL CONSTANT TIME

This is the duration over which equipments or components are kept constantly at final value temperature. This may be given in hours and minutes separately. This is shown in Fig.3.1. as BC.

3.5.6 FALL TIME

After attaining final constant duration, temperature is reduced upto the initial value (room temperature). The rate of falling may be taken different from rate of rising. This value is also given in hours and minutes separately according to the user requirement. This duration is shown as CD in Fig. 3.1.

3.5.7 LOWER CONSTANT TIME

This constant time is taken same as it have been taken in the FINAL CONSTANT TIME. Value may be changed in hours and minutes separately. This period is shown in Fig.3.1 as DE.

One complete cycle time is the sum of Rise time + Final constant time + Fall time + Lower constant time. It may be of 24 hours or less (or more) than 24 hours in duration. After completing one cycle, it will generate another cycle of the same shape till the system resets.

This wave shape can be realized over multimeter or multichannel recorders.

CHAPTER - 4

SOFTWARE DEVELOPMENT

4.1 SUBROUTINES

4.1.1 FUNCTION NAME : CI (CHARACTER IN)

INPUT : NONE

OUTPUT : 8 BIT CHARACTER RECEIVED IN REG.A, PARITY UNMASKED

CALLS : DELAY

DESTROYS : A, FLAGS

DESCRIPTION : Interrupt one disabled in this routine. CI waits until a character has been entered at the terminal and then return the character via the Reg. A to the calling routine.

CALLING ADDRESS:814F

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	814F	F3	DI		DISABLED INTERRUPT
	8150	E5	PUSH H		SAVE (HL)
	8151	D5	PUSH D		SAVE (DE)
	8152	C5	PUSH B		SAVE (BC)
	8153	2A 85 E0	LHLD	E085	LOAD (DE) FOR DELAY
	8156	EB	XCHG		DELAY
	8157	20	RIM		CHECK SID
	8158	17	RAL		IF SID IS HIGH
	8159	DA 57 81	JC	8157	GO TO CHECK NEXT
	815C	CD C0 81	CALL	81C0	CALL DELAY
	815F	01 08 00	LXI B	0008	(C)=NO.OF BITS
	8162	2A 87 E0	LHLD	E087	LOAD (DE) FOR
	8165	EB	XCHG		DELAY
	8166	CD C0 81	CALL	81C0	CALL DELAY
	8169	20	RIM		GET SID
	816A	17	RAL		CY = SID

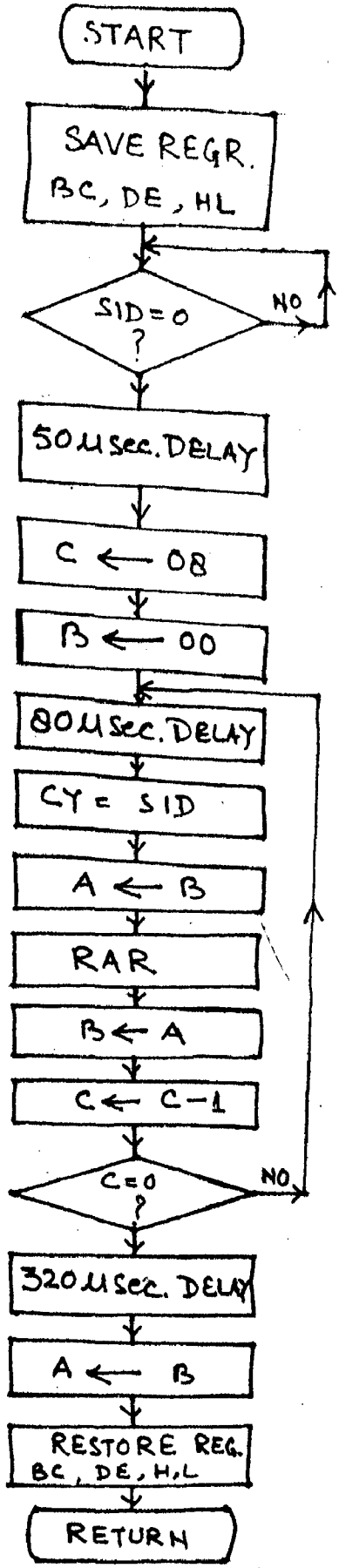


FIG-4.2, FLOW CHART FOR CHARACTER IN (CI)

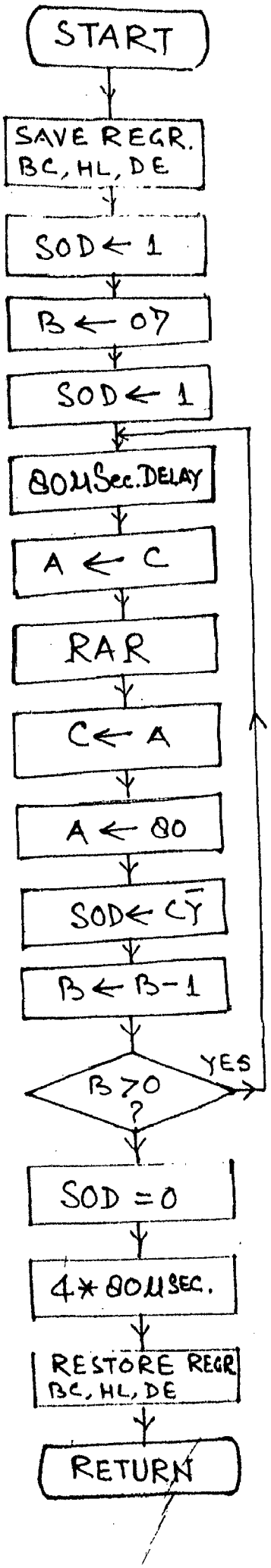


FIG-4.2 FLOW CHART FOR CHARACTER OUT [CO]

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	816B	78	MOV A,B		TRANSFERRED (B) TO (A
	816C	1F	RAR		MOV CY TO MSB OF (A)
	816D	47	MOV B,A		SAVE (A) IN (B)
	816E	0D	DCR C		DECREMENT COUNT
	816F	C2 62 81	JNZ	8162	IF NOT ZERO,CONTINUE
	8172	2A 87 E0	LHLD	E087	LOAD (DE) FOR
	8175	EB	XCHG		DELAY
	8176	CD C0 81	CALL	81C0	CALL DELAY
	8179	78	MOV A,B		TRANSFER INTO (A) FROM (B)
	817A	C1	POP B		RESTORE (BC)
	817B	D1	POP D		RESTORE (DE)
	817C	E1	POP H		RESTORE (HL)
	817D	C9	RET		RETURN

.2. FUNCTION NAME : CO (CHARACTER OUT)

INPUT : REG.C HAS CHARACTER TO OUTPUT ON IO DEVICE

OUTPUT : REG.C CONTAIN THE CHARACTER IS OUTPUT TO I/O DEVICE.

CALLS : DELAY

DESTROYS : A, FLAGS

DESCRIPTION : CO sends its input argument to I/O device. Interrupt are disabled.

CALLING ADDRESS:8187

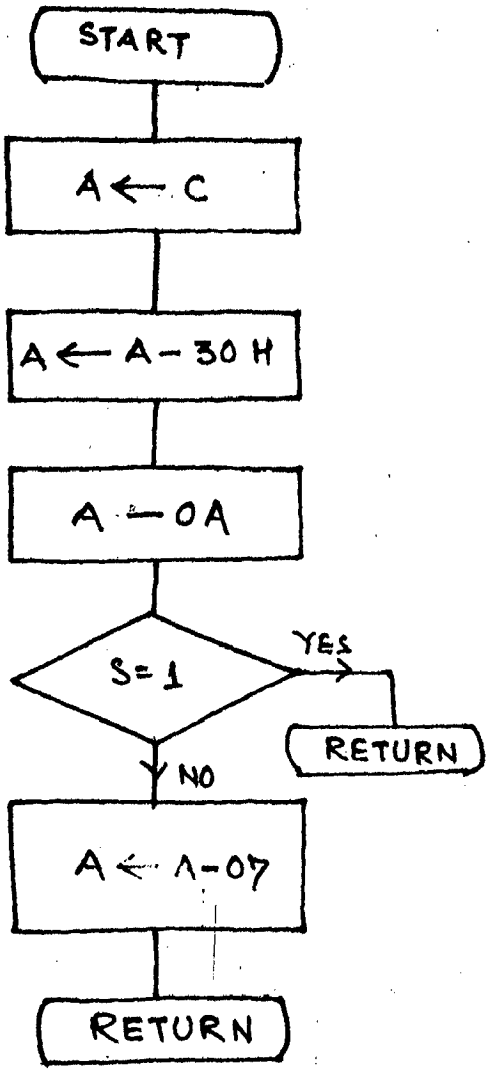
LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
	8187	F3	DI	DISABLE INTERRUPT
	8188	F5	PUSH H	SAVE (H,L)
	8189	D5	PUSH D	SAVE (D,E)

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	818A	C5	PUSH B		SAVE (B,C)
	818B	3E CO	MVI A	CO	KEEP CO H INTO (A)
	818D	06 07	MVI B	07	KEEP 07 H INTO (B)
	818F	30	SIM		SOD = 1
	8190	2A 87 E0	LHLD	E087	LOAD (DE) FOR
	8193	EB	XCHG		DELAY
	8194	CD CO 81	CALL	81CO	CALL DELAY
	8197	79	MOV A,C		GET (C) IN (A)
	8198	1F	RAR		CY = LSB
	8199	4F	MOV C,A		SAVE (A) IN (C)
	819A	3E 80	MVI A	80	MSB = CY
	819C	1F	RAR		BIT A6 = 1
	819D	EE 80	XRI	80	COMPLEMENT MSB
	819F	05	DCR B		DECREMENT COUNT
	81A0	F2 8F 81	JP		IF(B) > 0, CONTINUE
	81A3	3E 40	MVI A	40	SOD = 0
	81A5	30	SIM		
	81A6	2A 87 E0	LHLD	E087	LOAD (HL) WITH COUNT DELAY
	81A9	29	DAD H		MULTIPLY (HL) BY 2
	81AA	29	DAD H		MULTIPLY (HL) BY 2
	81AB	EB	XCHG		DE = HL
	81AC	CD CO 81	CALL	81CO	CALL DELAY
	81AF	C1	POP B		RESTORE B,C
	81B0	D1	POP D		RESTORE D,E
	81B1	E1	POP H		RESTORE H,L
	81B2	C9	RET		RETURN

1.3. FUNCTION NAME : DELAY

INPUT : REG. DE-SIXTEEN BIT INTEGER DENOTING NUMBER OF TIMES TO LOOP

OUTPUT : NONE



FLOW CHART FOR CNVBN

FLOW CHART FOR MLPNT

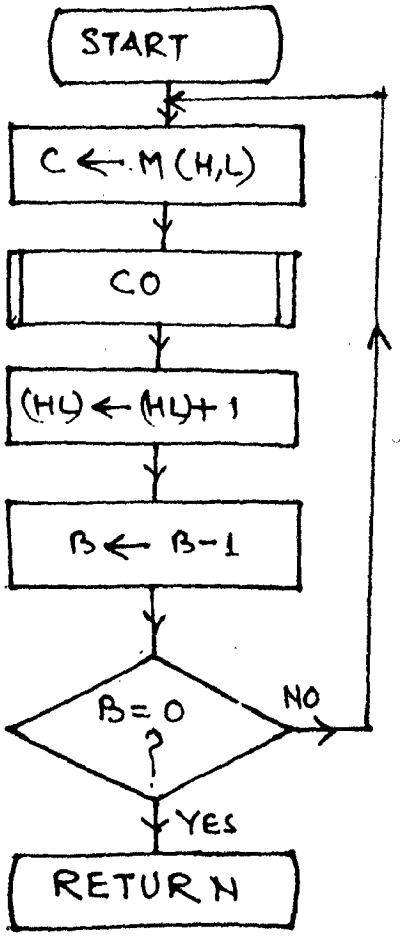


FIG-4.3

CALLS : NONE

DESTROYS : A, D, E FLAGS

DESCRIPTION : Delay does not return to caller until input argument is counted down to zero. For one count the delay is 7.680 μ sec. The delay equation is (24N+7)x320 n sec. Where N is the count in Reg. DE. This does not include the time for initializing the count and calling the routine.

CALLING ADDRESS: 81CO

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	81CO	1B	DCX D		DECREMENT COUNT
	81C1	7A	MOV A,D		TRANSFERRED (D) CONTENT INTO A
	81C2	B3	ORA E		ORED WITH (E)
	81C3	C2 CO 81	JNZ	81CO	NOT ZERO JUMP TO DECREMENT
	81C6	C9	RET		OTHERWISE RETURN

.1.4. FUNCTION NAME : CNVBN (CONVERT BINARY)

INPUT : INPUT IN REG.C. THE ASCII CHARACTER FOR O-9 OR A-F.

OUTPUT : OUTPUT IN REG.A THE HEX FROM O-F

CALLS : NONE

DESTROYS : A, FLAGS

DESCRIPTION : Converts the ASCII representation of hex into its corresponding Binary Value. It does not checks for validity of its input.

CALLING ADDRESS: 817E

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	817E	79	MOV A,C		TRANSFERRE ASCII INTO (A)
	817F	D6 30	SUI	30	SUBTRACT 30 FROM ( )
	8181	FE OA	CPI	OA	COMPARE WITH OA H
	8183	F8	RM		RETURN IF MINUS
	8184	D6 07	SUI	07	OTHERWISE SUBTRACT 07
	8186	C9	RET		AND THAN RETURN

4.1.5. FUNCTION NAME : MLPNT

INPUT : THE POINTER OF THE BLANK TO BE PRINTED IN REG. HL THE COUNTS IS DEFINED BY REG.B

OUTPUT : CHARACTER IS OUTPUT TO I/O DEVICE

CALLS : CO

DESTROYS : A,B,C,H,L FLAGS

DESCRIPTION : The block of ASCII masked parity character pointed by HL is printed upto the nth position. Where as n is defined by Reg.B.

CALLING ADDRESS:82F5

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	82F5	4E	MOV C,M		CONTENTS OF MEMORY POINTER TRANSFERRED INTO (C)
	82F6	CD 87 81	CALL	8187	CALL CHARACTER OUT SUBROUTINE
	82F9	23	INX H		INCREMENT MEMORY POINTER
	82FA	05	DCR B		DECREMENT COUNT
	82FB	C2 F5 82	JNZ	82F5	IF COUNT IS NOT ZERO. 'CO' ANOTHER
	82FE	C9	RET		OTHERWISE RETURN

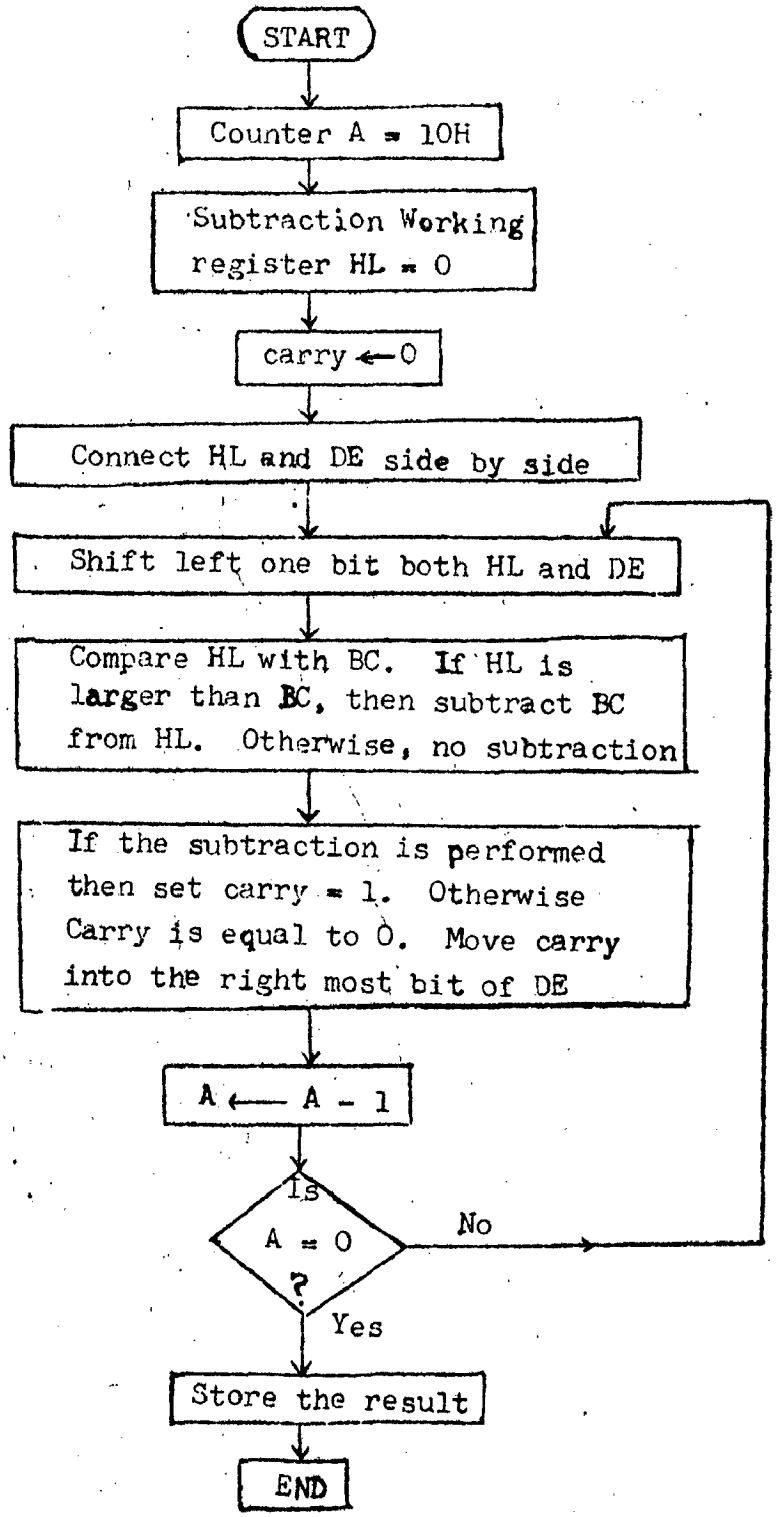


FIG-4.4 FLOW CHART FOR DIVISION (16 BIT BINARY NUMBER)

.1.6. FUNCTION NAME : DIVSN (DIVISION 16 BIT/16 BIT)

INPUT : DE REGISTER PAIR CONTAINS 16-BIT DIVIDEND
BC REGISTER PAIR CONTAINS 16-BIT DIVISOR
REGISTER A IS USED AS 16 BIT COUNTER

OUTPUT : QUOTIENT IN HL PAIR
REMAINDER IN DE PAIR

CALLS : NONE

DESTROYS : AF, D, E, H, L

DESCRIPTION : Address D100 H stores the counter

CALLING ADDRESS: C350

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
DIVSN:	C350	D5	PUSH D	SAVE (DE)
	C351	C5	PUSH B	SAVE (BC)
	C352	EB	XCHG	(DE) = (HL)
	C353	48	MOV C,B	
	C354	06 00	MVI B 00	CLEAR (B)
	C356	AF	XRA A	CLEAR (A) AND FLAGS
	C357	67	MOV H,A	(H) = 00
	C358	6F	MOV L,A	(L) = 00
	C359	3E 10	MVI A 10 H	ACCUMULATOR CONTAINS 10 H COUNTER
LP1:	C35B	32 00 D1	STA D100	M(ADDR) = COUNTER
	C35E	7B	MOV A,E	SHIFT THE 16 BIT
	C35F	17	RAL	DATA IN (DE) PAIR
	C360	5F	MOV E,A	LEFT ONE BIT
	C361	7A	MOV A,D	THROUGH CARRY
	C362	17	RAL	
	C363	57	MOV D,A	
	C364	7D	MOV A,L	HL ← HL+HL+CY
	C365	8D	ADC L	
	C366	6F	MOV L,A	

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
	C367	7C	MOV A,H	
	C368	8C	ADC H	
	C369	67	MOV H,A	
	C36A	7D	MOV A,L	HL ← HL - BC
	C36B	91	SUB C	
	C36C	6F	MOV L,A	
	C36D	7C	MOV A,H	
	C36E	98	SBB B	
	C36F	67	MOV H,A	
	C370	D2 7A C3	JNC LP2	
	C373	7D	MOV A,L	CY = 1, HL ← HL+BC
	C374	81	ADD C	
	C375	6F	MOV L,A	
	C376	7C	MOV A,H	
	C377	88	ADC B	
	C378	67	MOV H,A	
	C379	37	STC	SET CY = 1,
LP2:	C37A	3F	CMC	
	C37B	3A 00 D1	LDA D100	A ← M(ADDR)
	C37E	3D	DCR A	A ← A - 1
	C37F	C2 5B C3	JNZ LP1	A ≠ 0 , JUMP TO LP
	C382	EB	XCHG	A = 0, DE ⇌ HL
	C383	7D	MOV A,L	
	C384	8D	ADC L	
	C385	6F	MOV L,A	
	C386	7C	MOV A,H	
	C387	8C	ADC H	
	C388	67	MOV H,A	
	C389	7D	MOV A,L	
	C38A	C1	POP B	RESTORE (BC)
	C38B	D1	POP D	RESTORE (DE)
	C38C	C9	RET	RETURN

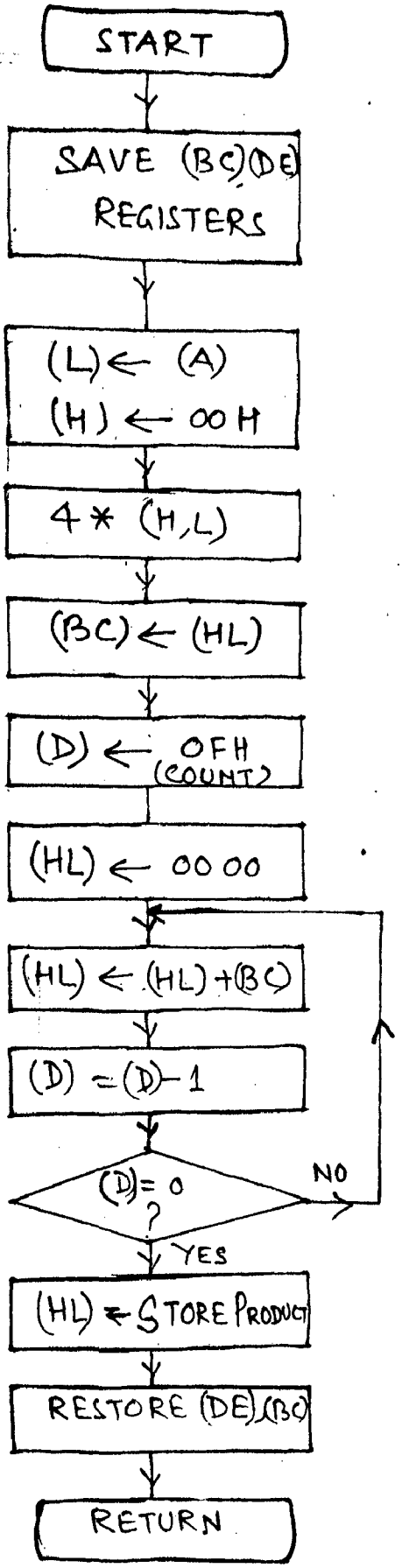


FIG-4.5 ; FLOW CHART FOR MULTIPLICATION BY 60

1.1.7. FUNCTION NAME : MULT6O (MULTIPLICATION BY 6O)

INPUT : MULTIPLICAND INTO (A), COUNT INTO (D)

OUTPUT : HL HOLD ACCUMULATED PRODUCT

CALLS : NONE

DESTROYS : H, L, A

DESCRIPTION : Hours are multiplied by 6O to convert into minutes.

CALLING ADDRESS : C1AO

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
MULT6O:	C1AO	C5	PUSH B	SAVE (BC)
	C1A1	D5	PUSH D	SAVE (DE)
	C1A2	6F	MOV L,A	SAVE MULTIPLICAND IN (L)
	C1A3	AF	XRA A	CLEAR (H)
	C1A4	67	MOV H,A	
	C1A5	29	DAD H	(HL) = (HL)*4
	C1A6	29	DAD H	
	C1A7	44	MOV B,H	(BC) = (HL)
	C1A8	4D	MOV C,L	
	C1A9	16 OF	MVI D OFH	COUNTER FOR 15D OPERATIONS
	C1AB	AF	XRA A	CLEAR (H)
	C1AC	67	MOV H,A	
	C1AD	6F	MOV L,A	AND (L)
LOOP:	C1AE	09	DAD B	(HL) = (HL)+(BC)
	C1AF	15	DCR D	DECREMENT COUNTER
	C1BO	C2 AE C1	JNZ LOOP	IF NOT ZERO; CONTIN
	C1B3	D1	POP D	RESTORE (DE)
	C1B4	C1	POP B	
	C1B5	C9	RET	

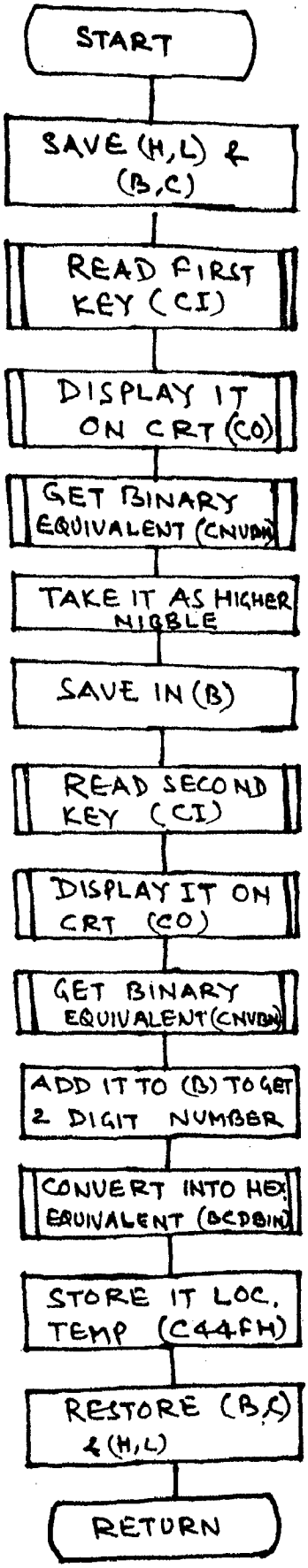


FIG-4.6 ; FLOWCHART FOR CHARACTER IN 2 DIGIT [C12D]

.1.8 FUNCTION NAME : CI2D (CHARACTER IN TWO DIGIT)

INPUT : NONE

OUTPUT : BINARY NO. IN LOC. C44F CORRESPONDING TO TWO KEYS PRESSED BY USER

CALLS : BCDBIN, CO, CI, CNVBN

DESTROY : A,

DESCRIPTION : Two Digits are given thrcugh ASCII Key board as information. And are sent to required location after conversion.

CALLSING ADDRESS: C1BE

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
CI2D:	C1BE	E5	PUSH H	SAVE (HL)
	C1BF	C5	PUSH B	SAVE (BC)
	C1CO	CD 4F 81	CALL 814FH	GET ASCII CODE OF FIRST PRESSED KEY
	C1C3	4F	MOV C,A	DISPLAY CHARACTER
	C1C4	CD 87 81	CALL 8187H	ON CRT
	C1C7	CD 7E 81	CALL 817EH	GET BINARY EQUIVALENT
	C1CA	07	RLC	EXCHANGE THE POSI-
	C1CB	07	RLC	TION OF LOWER AND
	C1CC	07	RLC	HIGHER NIBBLE
	C1CD	07	RLC	
	C1CE	47	MOV B,A	SAVE IN (B)
	C1CF	CD 4F 81	CALL 814FH	GET ASCII CODE OF SECOND PRESSED KEY
	C1D2	4F	MOV C,A	DISPLAY ON CRT
	C1D3	CD 87 81	CALL 8187H	
	C1D6	CD 7E 81	CALL 817EH	GET BINARY EQUIVALENT
	C1D9	BO	ORA B	ORED (A) WITH (B)
	C1DA	5F	MOV E,A	SAVE IN (E)
	C1DB	AF	XRA A	CLEAR D
	C1DC	57	MOV D,A	

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
	C1DD	00		
	C1DE	CD E9 C1	CALL C1E9H	CONVERT INTO HEX
	C1E1	C1	POP B	RESTORE (BC)
	C1E2	7D	MOV A,L	STORE HEXA EQUIVA
	C1E3	32 4F C4	STA C44FH	IN LOC C44FH
	C1E6	E1	POP H	RESTORE (H)
	C1E7	C9	RET	RETURN

1.9 FUNCTION NAME : BCDBIN

INPUT : BCD DATA IN D.E. REGISTER PAIR

OUTPUT : BIN DATA IN H.L. REGISTER PAIR

CALLS : NONE

DESTROYS : H,L,A,B,D,E,

DESCRIPTION : In converts 4 digit BCD NO. in (DE) into corresponding binary NO. in (HL).

CALLING ADDRESS : C1E9

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
BCDBIN:	C1E9	06 10	MVI B 10H	SET COUNTER FOR OPERATIONS
LOP:	C1EB	7A	MOV A,D	
	C1EC	1F	RAR	
	C1ED	57	MOV D,A	
	C1EE	7B	MOV A,E	DIVIDE BY 2
	C1EF	1F	RAR	
	C1F0	5F	MOV E,A	
	C1F1	7C	MOV A,H	DIVIDE (HL) BY 2
	C1F2	1F	RAR	AND TAKE REMAINDER
	C1F3	67	MOV H,A	OF (DE)/2.

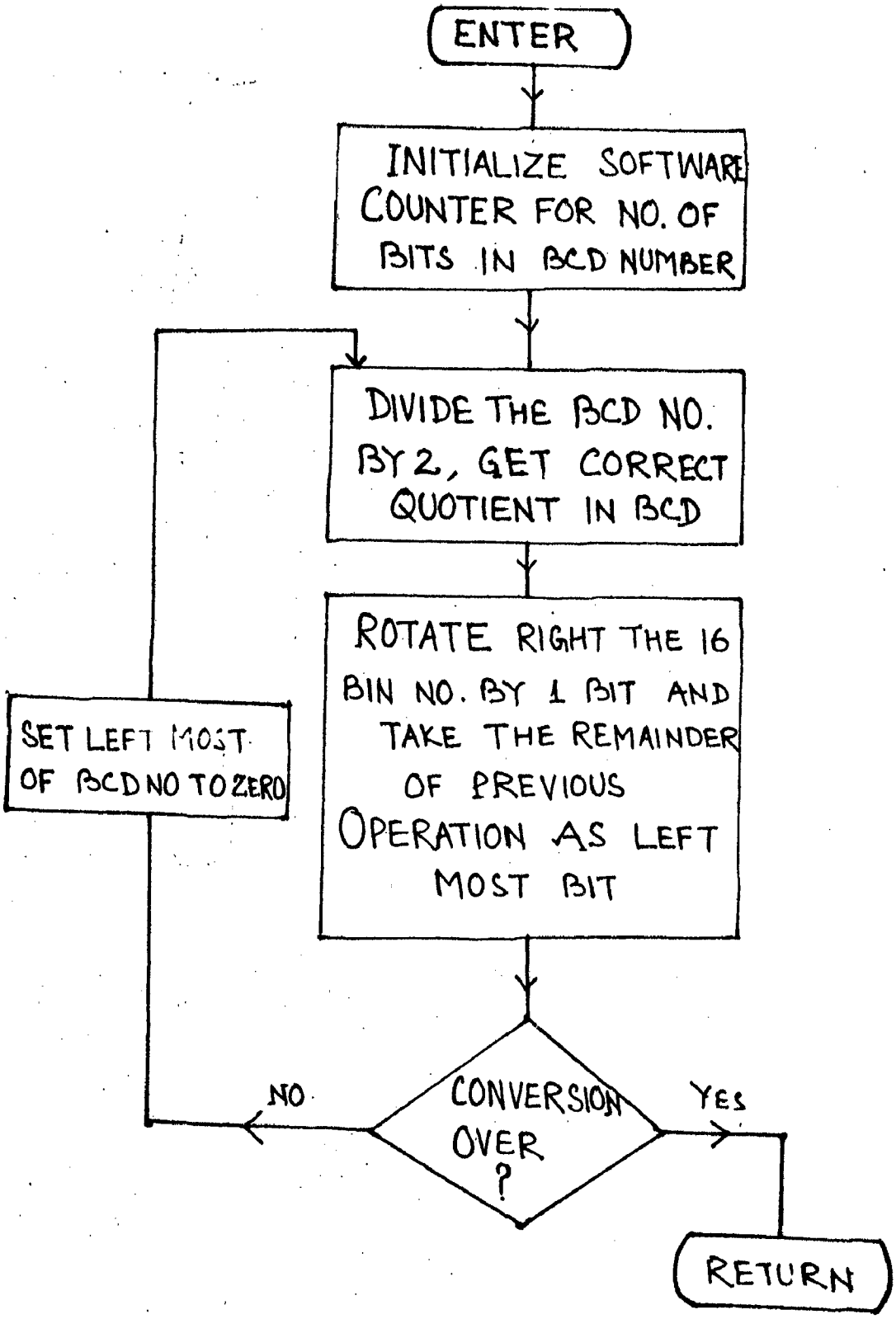


FIG-4.7 . FLOWCHART FOR BCD TO BINARY CONVERSION

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
	C1F4	7D	MOV A,L	DIVIDE (HL) BY 2
	C1F5	1F	RAR	AND TAKE REMAINDE
	C1F6	6F	MOV L,A	OF (DE)/2
	C1F7	05	DCR B	DECREMENT COUNTER
	C1F8	C2 FC C1	JNZ CONT	IF ZERO; RETURN TO
	C1FB	C9	RET	CALLING PROGRAM
CONT:	C1FC	7A	MOV A,D	
	C1FD	E6 7F	ANI 7FH	
	C1FF	57	MOV D,A	
	C200	E6 08	ANI 08H	
	C202	CA 09 C2	JZ NOCHG1	IF (D) CONTAINS
	C205	7A	MOV A,D	ILLEGAL BCD CODE
	C256	DE 03	SBI 03H	THEN CORRECT IT
	C258	57	MOV D,A	
NOCHG1:	C209	7B	MOV A,E	
	C20A	E6 80	ANI 80H	
	C20C	CA 13 C2	JZ NOCHG2	IF (E) CONTAINS
	C20F	7B	MOV A,E	ILLEGAL BCD CODE;
	C210	DE 30	SBI 30H	THEN CORRECT IT.
	C212	5F	MOV E,A	
NOCHG2:	C213	7B	MOV A,E	
	C214	E6 08	ANI 08H	
	C216	CA EB C1	JZ LOP	
	C219	7B	MOV A,E	
	C21A	DE 03	SBI 03H	
	C2AC	5F	MOV E,A	
	C21D	C3 EB C1	JMP LOP	CONTINUE

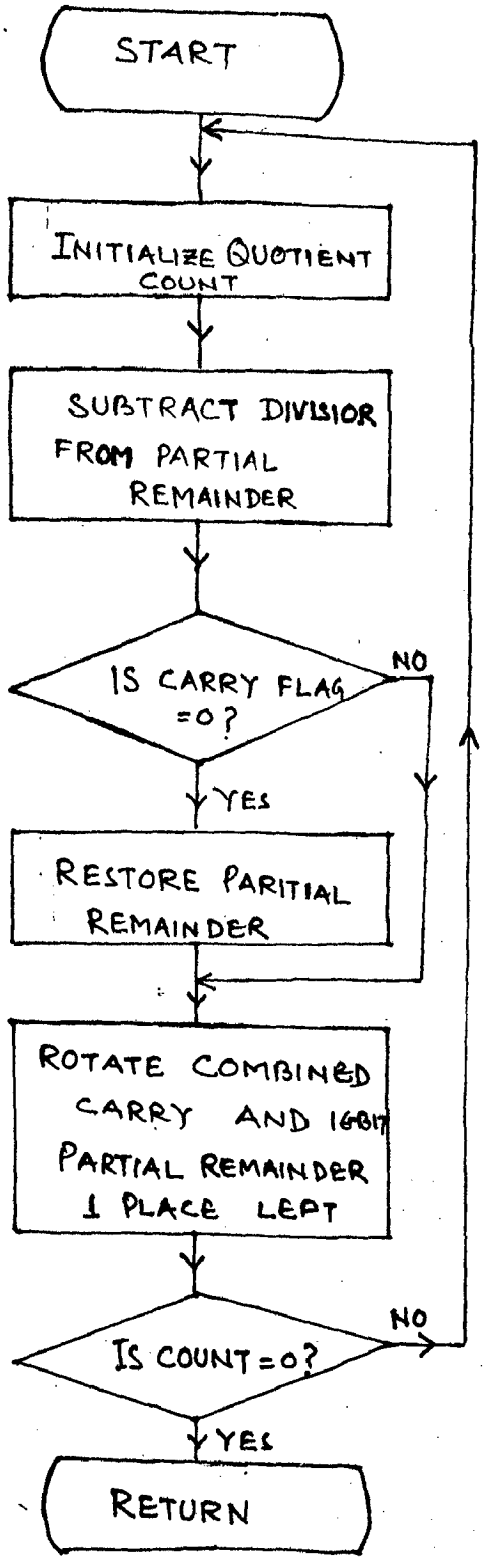


FIG.- 4.8 ; FLOW CHART FOR DIVISION [16 BIT/8 BIT]
NINTH BIT IS IN CARRY

.1.10. FUNCTION NAME : DIVSN (DIVISION 16 BIT/8 BIT)

INPUT : AB HOLD THE DIVIDEND - M.S. BYTE IN A
C HOLD THE DIVISOR. D IS USED AS TEMPORARY REGISTER
E HOLDS A COUNT OF THE NUMBER

OUTPUT : THE QUOTIENT IS HELD IN THE B REGISTER (AND HENCE REPLACES THE L.S. BYTE OF THE DIVIDEND) WITH THE M.S.(NINTH) BIT IN THE CARRY.

CALLS : NONE

DESTROYS : A,B,C,H,L

DESCRIPTION : This subroutine could be used as it is very short in size. But some time result may overflow because quotient is held in 8 bit (M.S. Ninth bit is in the carry) only.

CALLING ADDRESS : C450

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
DIVSN:	C450	D5	PUSH D	
	C451	7C	MOV A,H	
	C452	48	MOV C,B	
	C453	45	MOV B,L	
	C454	1E 09	MVI E 09H	(E) - 9
LOOP1:	C456	91	SUB C	(A) ← (A) - (C)
	C457	3F	CMC	COMPLEMENT CY
	C458	DA 5E C3	JC LOOP2	CY = 1 ?
	C45B	81	ADD C	NO:(A) ← (A)+(C)
	C45C	37	STC	SET CARRY
	C45D	3F	CMC	COMPLEMENT CY
LOOP2:	C45E	57	MOV D,A	SAVE A
	C45F	78	MOV A,B	MOVE CY(QUOTIENT BIT) INTO L.S.BIT POSITION OF B AND M.S.BIT OF B INTO CY.
	C460	17	RAL	
	C461	47	MOV B,A	

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
	C462	7A	MOV A,D	RESTORE A
	C463	17	RAL	MOVE CY INTO A
	C464	1D	DCR E	(E) ← (E)- L
	C465	C2 56 C3	JNZ LOOP1	COUNT ZERO? NO: REPEAT
	C468	78	MOV A,B	GET INTO A
	C469	D1	POP D	RESTORE
	C46A	C9	RET	

1.11. FUNCTION NAME : DEL15S AND DLP25S

INPUT : NONE

OUTPUT : NONE

CALLS : DELAY

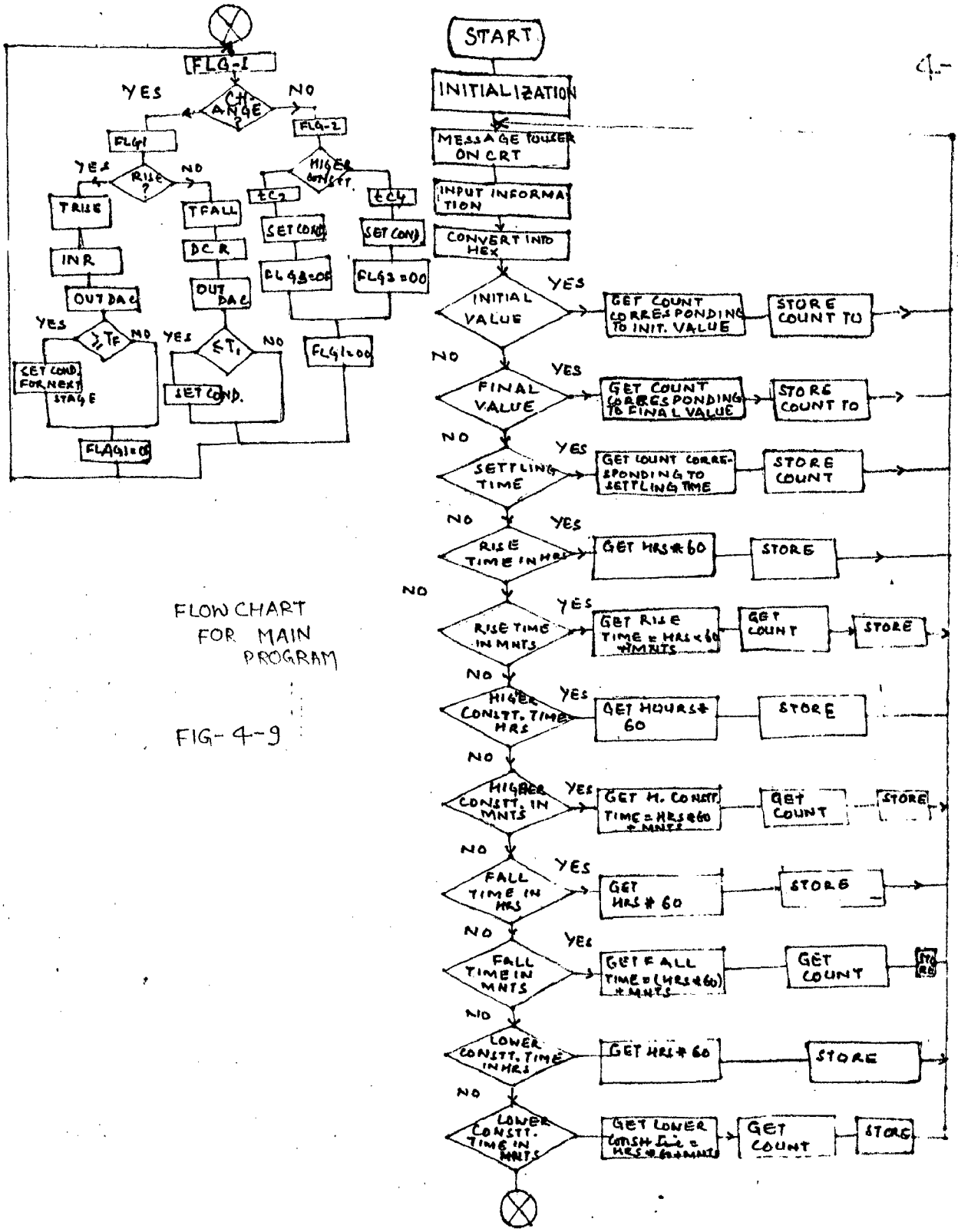
DESTROYS : H,D,E

DESCRIPTION : Subroutine of 15 secs Delay and subroutine of 0.25 sec delay.

CALLING ADDRESS: C330 and C340 respectively

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
DEL15S:	C330	26 01	MVI H 1EH	GET COUNT 30D IN (H)
RPT:	C332	11 4C FE	LXI D EF4CH	FOR 0.5 SEC. DELAY [FE4CH = 65100]
	C335	CD CO 81	CALL 81CO	CALL DELAY
	C338	25	DCR H	DECREMENT COUNT

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
	C339	C2 32 C3	JNZ RPT	IF COUNT NOT ZERO, JUMP TO RPT
	C33C	C9	RET	OTHERWISE RETURN
DLP25S:	C340	11 26 7F	LXI D 7F26H	COUNT FOR 0.25 SEC DELAY
	C343	CD C0 81	CALL 81C0	CALL DELAY
	C346	C9	RET	
MNTSND:	C347	11 1E 02	LXID 021EH	COUNT FOR DEMONSTRATING IN SECONDS AND MINUTES.
	C34A	CD C0 81	CALL 81C0	CALL DELAY
	C34D	C9	RET	



FLOW CHART FOR MAIN PROGRAM

FIG-4-9

4.2 INITIALIZATION

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
	C015	31 FF CF	LXI SP CFFFH	INITIALIZE STACK POINTER
	C018	21 00 C5	LXI H C500H	STARTING ADDRESS OF MESSAGE
	C01B	0E 0C	MVI C 0CH	NO.OF INFORMATION REQUIRED
GETINF:	C01D	06 18	MVI B 18H	NO.OF CHARACTER TO BE OUTPUTED
NEXT:	C01F	C5	PUSH B	SAVE (BC)
	C020	4E	MOV C,M	DISPLAY CHARACTER
	C021	CD 87 81	CALL 8187H	AS POINTED BY (HL)
	C024	C1	POP B	RESTORE (BC)
	C025	23	INX H	(HL) = (HL)+1
	C026	05	DCR B	DECREMENT COUNTER
	C027	C2 1F C0	JNZ NEXT	IF NOT ZERO;CONTINUE
	C02A	CD BE C1	CALL C1BEH	READ TWO KEYS PRESSED(C12D)
	C02D	79	MOV A,C	IF (C) = 0CH
	C02E	FE 0C	CPI 0CH	THEN GO TO TEMP.
	C030	CA 6D C0	JZ TIN	INITIAL (TIN)
	C033	FE 0B	CPI 0BH	IF (C)= 0BH; THEN
	C035	CA 84 C0	JZ TFNL	GO TO TEMP.FINAL (TFNL)
	C038	FE 09	CPI 09H	IF (C)= 09H; THEN
	C03A	CA 70 C4	JZ SETLE	GO TO SETTLING TIME
	C03D	00 00 00	NOP NOP NOP	
	C040	00 00	NOP NOP	
	C042	FE 08	CPI 08H	IF (C)= 08H THEN
	C044	CA A8 C0	JZ RTMH	GO TO RTMH
	C047	FE 07	CPI 07H	IF (C)= 07H THEN GO
	C049	CA B7 C0	JZ RTMM	TO RISE TIME IN MINUTES

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	C04C	FE 06	CPI	06H	IF (C)= 06H; GO TO FINAL VALUE TIME I HOURS (FVTMH)
	C04E	CA E2 C0	JZ	FVTMH	
	C051	FE 05	CPI	05H	IF (C)= 05H; GO TO FINAL VALUE TINE IN MINUTES (FVTMM)
	C053	CA F4 C0	JZ	FVTMM	
	C056	FE 04	CPI	04H	IF (C)= 04H; GO TO FALL TIME IN HOURS
	C058	CA 0C C1	JZ	FTMH	
	C05B	FE 03	CPI	03H	IF (C)= 03H; GO TO
	C05D	CA 1B C1	JZ	FTMM	FALL TIME IN MINUTES
	C060	FE 02	CPI	02H	IF (C)= 02H; GO TO
	C062	CA 46 C1	JZ	LVTMH	LOWER VALUE TIME IN HOURS
	C065	FE 01	CPI	01	IF (C)= 01H; GO TO
	C067	CA 58 C1	JZ	LVTMM	LOWER VALUE TIME IN MINUTES OTHERWISE
	C06A	C3 25 C2	JMP		GO TO WAVEFROM PROGRAM
TIN:	C06D	0D	DCR C		
	C06E	3A 4F C4	LDA	C44FH	GET TWO KEYS PRESSED VALUES IN (A)
	C071	FE 40	CPI	40 H	COMPARE WITH 64 DECIMAL
	C073	CA 74 C1	JZ	ERR	> 64, THEN GO TO
	C076	D2 74 C1	JNC	ERR	ERROR SUBROUTINE
	C079	07	RLC		MULTIPLY BY 2
	C07A	07	RLC		MULTIPLY BY 2
	C07B	32 40 C4	STA	C440	STORE TO LOC C440H
	C07E	C3 1D C0	JMP	GETINF	GO TO TAKE ANOTHER VALUE
	C081	00	NOP		
	C082	00	NOP		
	C083	00	NOP		

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
TFNL:	C084	OD	DCR C	DECREMENT NO.OF INFORMATION REQUIRED
	CC85	3A 4F C4	LDA C44FH	TEMPORARY LOCATION
	C088	FE 41	CPI 41 H	COMPARE WITH 65D
	C08A	CA 74 C1	JZ ERR	IF > 65D,GO TO ERROR
	C08D	D2 74 C1	JNC ERR	SUBROUTINE
	C090	07	RLC	MULTIPLY BY 2
	C091	07	RLC	MULTIPLY BY 2
	C092	57	MOV D,A	SAVE IN(D) REGR.
	C093	3A 40 C4	LDA C440H	LOAD INITIAL VALUE
	C096	57	MOV E,A	INITIAL VALUE IN E REGR.
	C097	7A	MOV A,D	FINAL VALUE IN ACCUMULATOR
	C098	BB	CMP E	COMPARE FINAL VALUE WITH INITIAL
	C099	CA 74 C1	JZ ERR	IF FINAL IS LESS, ERROR MESSAGE
	C09C	DA 74 C1	JC ERR	IF FINAL EQUALS, ERROR MESSAGE
	C09F	32 41 C4	STA C441H	OTHERWISE STORE FINAL VALUE
	C0A2	00	NOP	
	C0A3	OD	DCR C	REDUCE NO.OF INFOR MATION
	C0A4	C3 1D C0	JMP C01DH	GO TO TAKE ANOTHER INFORMATION
	C0A7	00	NOP	
RTMH:	C0A8	OD	DCR C	DECREMENT NO.OF INFORMATION REQUIRED
	C0A9	3A 4F C4	LDA C44FH	LOAD FROM TEMP.
	C0AC	E5	PUSH H	SAVE (H,L)
	C0AD	CD A0 C1	CALL C1A0H	MULTIPLY BY 60
	C0B0	22 44 C4	SHLD C444H	STORE TEMPORARY
	C0B3	E1	POP H	RESTORE H,L

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	COB4	C3 1D CO	JMP	GETINF	GET NEXT INFORMATION
RTMM:	COB7	OD	DCR C		REDUCE ONE INFORM-ATION
	COB8	3A 4F C4	LDA	C44FH	LOAD FROM TEMP.
	COBB	E5	PUSH H		SAVE (H,L).REGR.PAIR
	COBC	C5	PUSH B		SAVE (B,C) REGR.PAIR
	COBD	2A 44 C4	LHLD	C444H	LOAD TEMPORARY CON-TENT OF MINUTES FROM H,L
	COCO	4F	MOV C,A		TRANSFER MINUTES INTO C FROM ACCUMU-LATOR
	COC1	AF	XRA A		ZERO IN ACCUMULATOR
	COC2	47	MOV B,A		TRANSFER INTO B
	COC3	O9	DAD B		ADD BC WITH HL. HL CONTENTS TOTAL MINUTE
	COC4	CD 80 C4	CALL	GUNA6O	MULTIPLY WITH 6O TO MAKE SECOND
	COC7	3A 40 C4	LDA	C44OH	LOAD INITIAL VALUE
	COCA	47	MOV B,A		TRANSFER INTO B
	COCB	3A 41 C4	LDA	C441H	LOAD FINAL VALUE (TF)
	COCE	90	SUB B		A = TF-TO
	COCF	4F	MOV C,A		TRANSFER INTO C FROM A
	CODO	AF	XRA A		CLEAR B
	COD1	47	MOV B,A		
	COD2	CD 50 C3	CALL	DIVSN	N^r IN DE; D^r IN BC; RESULT IN H,L
	COD5	00	NOP		
	COD6	29	DAD H		MULTIPLY BY 2
	COD7	29	DAD H		MULTIPLY BY 2
	COD8	22 44 C4	SHLD	C444H	STORE (H,L) CONTENTS
	CODC	C1	POP B		RESTORE B,C
	CODD	E1	POP H		RESTORE H,L

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	CODE	C3 1D C0	JMP	GETINF	GO TO GET ANOTHER
	C0E1	00	NOP		INFORMATION
FVTMH:	C0E2	0D	DCR C		REDUCE NO.OF INFORM-ATION REQUIRED
	C0E3	3A 4F C4	LDA	C44FH	LOAD HOUR VALUE TEMPORARILY
	C0E6	E5	PUSH H		SAVE (H,L)
	C0E7	CD A0 C1	CALL	C1A0H	CONVERT HOURS INTO MINUTE MULTIPLYING BY 60.
	C0EA	22 46 C4	SHLD	C446H	STORE MINUTES TEM-PORARILY.
	C0ED	E1	POP H		RESTORE H
	C0EE	C3 1D C0	JMP	C01DH	TAKE ANOTHER INFOR-MATION.
	C0F1	00	NOP		
	C0F2	00	NOP		
	C0F3	00	NOP		
FVTMM:	C0F4	0D	DCR C		REDUCE NO OF INFORM-ATION REQUIRED
	C0F5	3A 4F C4	LDA	C44FH	LOAD MINUTES INTO ACCUMULATOR
	C0F8	E5	PUSH H		SAVE (H,L)
	C0F9	C5	PUSH B		SAVE (B,C)
	C0FA	2A 46 C4	LHLD	C446H	LOAD HRS*60 INTO (H,L)
	C0FD	4F	MOV C,A		TRANSFER INTO C REG
	C0FE	AF	XRA A		CLEAR B
	C0FF	47	MOV B,A		
	C100	09	DAD B		(H,L) = (H,L)+(B,C)
	C101	29	DAD H		MULTIPLY BY 2
	C102	29	DAD H		MULTIPLY BY 2
	C103	22 46 C4	SHLD	C446H	STORE THE (HL)CONTENTS
	C106	C1	POP B		RESTORE (B)
	C107	E1	POP H		RESTORE (H)
	C108	C3 1D C0	JMP	C01DH	GET ANOTHER INFORMA-TION

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	C10B	00	NOP		
FTMH:	C10C	0D	DCR C		DECREMENT NO OF I - FORMATIONS REQUIRED
	C10D	3A 4F C4	LDA	C44FH	LOAD FROM TEMP.
	C110	E5	PUSH H		SAVE (H,L)
	C111	CD A0 C1	CALL	C1A0H	MULTIPLY BY 60D
	C114	22 48 C4	SHLD	C448H	STORE (H,L)
	C117	C1	POP H		RESTORE (H,L)
	C118	C3 1D C0	JMP	C01DH	GO TO GET NEXT INFORMATION
FTMM:	C11B	0D	DCR C		DECREMENT NO OF IN-FORMATION REQUIRED
	C11C	3A 4F C4	LDA	C44FH	LOAD FROM TEMP.
	C11F	E5	PUSH H		SAVE (H,L)
	C120	C5	PUSH B		SAVE (B,C)
	C121	2A 48 C4	LHLD	C448H	GET HRS*60 IN (H,L)
	C124	4F	MOV C,A		(H,L) = (H,L)+(B,C)
	C125	AF	XRA A		
	C126	47	MOV B,A		
	C127	09	DAD B		
	C128	CD 80 C4	CALL	GUNA60	MULTIPLY BY 60D
	C12B	3A 40 C4	LDA	C440H	(C) = (T_F-T_O)
	C12E	47	MOV B,A		
	C12F	3A 41 C4	LDA	C441H	
	C132	90	SUB B		
	C133	4F	MOV C,A		
	C134	AF	XRA A		CLEAR (B)
	C135	47	MOV B,A		
	C136	CD 50 C3	CALL	C350H	GET DURATION OF EACH STEP IN MINS
	C139	00	NOP		

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	C13A	29	DAD H		MULTIPLY BY 4
	C13B	29	DAD H		
	C13C	22 48 C4	SHLD	C448H	STORE (H,L)
	C13F	00	NOP		
	C140	C1	POP B		RESTORE REGS.
	C141	E1	POP H		
	C142	C3 1D C0	JMP	C01DH	GO TO GET NEXT INFORMATION
	C145	00	NOP		
	C146	00 00	NOP NOP		
LVTMH:	C148	0D	DCR C		DECREMENT NO OF INFORMATION REQUIRED
	C149	3A 4F C4	LDA	C44FH	(A)=HRS IN LOWER VALUE TIME
	C14C	E5	PUSH H		SAVE (H,L)
	C14D	CD A0 C1	CALL	C1A0H	MULTIPLY BY 60D
	C150	22 4A C4	SHLD	C44AH	STORE HOURS*60
	C153	E1	POP H		RESTORE (H,L)
	C154	C3 1D C0	JMP	C01DH	GO TO GET NEXT INFORMATION
	C157	00 00 00	NOP NOP NOP		
LVTMM:	C15A	0D	DCR C		
	C15B	3A 4F C4	LDA	C44FH	(A)=MINS IN LOWER VALUE TIME
	C15E	E5	PUSH H		SAVE REGS
	C15F	C5	PUSH B		
	C160	2A 4A C4	LHLD	C44AH	(H,L)= HRS*60
	C163	4F	MOV C,A		(H,L)=HRS*60+MINS
	C164	AF	XRA A		
	C165	47	MOV , B,A		
	C166	09	DAD B		
	C167	29	DAD H		MULTIPLY BY 4
	C168	29	DAD H		

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
	C169	22 4A C4	SHLD C44AH	STORE (H,L)
	C16C	C1	POP B	RESTORE REGS.
	C16D	E1	POP H	
	C16E	C3 25 C2	JMP C225H	GO TO NEXT INFORMATION
	C171	00 00 00	NOP NOP NOP	
ERR:	C174	E5	PUSH H	SAVE REGS.
	C175	C5	PUSH B	
	C176	21 0A C6	LXI H C60AH	
	C179	06 1C	MVI B 1C	
NEC:	C17B	00	NOP	
	C17C	4E	MOV C,M	DISPLAY ERROR
	C17D	CD 87 81	CALL 8187H	MESSAGE ON CRT
	C120	23	INX H	
	C121	05	DCR B	
	C122	C2 7B C1	JNZ C17BH	
	C125	C1	POP B	RESTORE REGS.
	C126	E1	POP H	
	C127	0C	INR C	
	C128	7D	MOV A,L	
	C129	DE 18	SBI 18H	MAKE NECESSARY CORRECTION TO GET CORRECT INFORMATION
	C12B	6F	MOV L,A	
	C12C	D2 1D C0	JNC C01DH	
	C12F	C6 FF	ADI FF	
	C131	6F	MOV L,A	
	C132	25	DCR H	
	C133	C3 1D C0	JMP C01DH	GO TO GET INFORMATION

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
SETLE:	C470	OD	DCR C		DECREMENT NO OF INFORMATION REQUIRED
	C471	CD BE C1	CALL CI2D (C1B6H)		(A)=SETTLING TIME
	C474	32 42 C4	STA	C442H	STORE SETTLING TIME
	C477	C3 1D C0	JMP	C01DH	GO TO GET NEXT INFORMATION
GUNA60:	C480	00	NOP		
	C481	C0	NOP		
	C482	29	DAD H		MULTIPLY BY 4
	C483	29	DAD H		
	C484	06 0E	MVI B, 0E		COPY (H,L) INTO (D,E)
	C486	54	MOV D,H		ADD (D,E) 14 TIMES
	C487	6B	MOV E,L		TO (H,L) TO GET
	C488	19	DAD D		(H,L)* 15D IN (H,L)
	C489	05	DCR B		
	C48A	C2 88 C4	JNZ	C488	
	C48D	00	NOP		
	C48E	00	NOP		
	C48F	C9	RET		RETURN TO CALLING PROGRAM

WAVEFORM GENERATION PROGRAM

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
WAVFOM:	C225	AF	XRA A		CLEAR (A)
	C226	32 4C C4	STA	C44CH	CLEAR FLG1
	C229	32 4D C4	STA	C44DH	CLEAR FLG2

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	C22C	32 4E C4	STA	C44EH	CLEAR FLG3
	C22F	3A 40 C4	LDA	C440H	LOAD INITIAL VALUF IN (A)
	C232	D3 48	OUT	48H	OUTPUT TO DAC
STLTM:	C234	OE 3C	MVI C	3CH	(C)= 60D
	C236	3A 42 C4	LDA	C442H	GET SETTLING TIME
	C239	FE 1E	CPI	1EH	< 30, GO TO DELP
	C23B	DA 41 C2	JM/JC	DELP	
	C23E	79	MOV A,C		(C)= (C)*2
	C23F	07	RLC		
	C240	4F	MOV C,A		
DELP:	C241	CD 30 C3	CALL	DEL15S	CALL 15 SECOND DELAY
	C244	OD	DCR C		DECREMENT COUNT
	C245	C2 41 C2	JNZ	DELP	SETTLING TIME OVER?
	C248	3A 40 C4	LDA	C440H	YES:(A)=INITIAL VALUE
VALOUT:	C24B	D3 48	OUT	48H	OUTPUT TO DAC
	C24D	F5	PUSH	PSW	SAVE VALUE
	C24E	3A 4C C4	LDA	C44CH	IF FLG1 IS ZERO,
	C251	B7	ORA A		GO TO CHANG
	C252	CA 5B C2	JZ	CHANG	
	C255	C3 CA C2	JMP	CONST	OTHERWISE CONSTANT
	C258	00 00 00	NOP NOP	NOP	
CHANG:	C25B	3A 4D C4	LDA	C44DH	CHECK FLAG2
	C25E	B7	ORA A		IF IT IS ZERO
	C25F	CA 65 C2	JZ	RISE	JUMP TO RISE
	C262	C3 97 C2	JMP	FALL	OTHERWISE FALL
RISE:	C265	2A 44 C4	LHLD	C444H	COUNT CORRESPONDING TO RISE TIME
	C268	44	MOV B,H		SAVE (H) IN (B)
	C269	4D	MOV C,L		SAVE (L) IN (C)
	C26A	79	MOV A,C		GET (C) IN (A)

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
	C26B	B7	ORA A		IF IT IS ZERO, JUMP
	C26C	CA 76 C2	JZ	CRT2	TO CRT2
CRT1:	C26F	CD 40 C3	CALL	DLP25S	CALL DELAY OF 0.25 SECONDS
	C272	0D	DCR C		DECREMENT COUNT
	C273	C2 6F C2	JNZ	CRT1	IF COUNT >0, JUMP TO CRT1
CRT2:	C276	78	MOV A,B		GET (B) IN (A)
	C277	B7	ORA A		IF (A) IS ZERO JUMP
	C278	CA 81 C2	JZ	CRTCMP	TO CRTCMP
	C27B	0E FF	MVI C	FFH	(BC) = BC - 1
	C27D	05	DCR B		
	C27E	C3 6F C2	JMP	CRT1	JUMP TO CRT1
CRTCMP:	C281	F1	POP	PSW	RESTORE VALUE
	C282	3C	INR A		INCREMENT STEP
	C283	21 41 C4	LXI H	C441H	(A)< TF, JUMP TO
	C286	BE	CMP M		VALOUT
	C287	C2 4B C2	JNZ	VALOUT	
	C28A	F5	PUSH	PSW	SAVE VALUE
	C28B	3E 0F	MVI A,	0FH	SET FLG2
	C28D	32 4D C4	STA	C44DH	
	C290	32 4C C4	STA	C44CH	SET FLG1
	C293	F1	POP	PSW	RESTORE VALUE
	C294	C3 4B C2	JMP	VALOUT	JUMP TO VALOUT
FALL:	C297	2A 48 C4	LHLD	C448H	GET COUNT CORRESPON-
	C29A	44	MOV B,H		DING TO FALL TIME
	C29B	4D	MOV C,L		IN (BC)
	C29C	79	MOV A,C		GET LOWER BYTE IN (A)
	C29D	B7	ORA A		IF ZERO, JUMP TO
	C29E	CA A8 C2	JZ	CFT2	CFT2

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
CFT1:	C2A1	CD 40 C3	CALL	DLP25S	CALL 0.25 SECOND DELAY
	C2A4	0D	DCR C		DECREMENT LOWER BYTE
	C2A5	C2 A1 C2	JNZ	CFT1	IF NOT ZERO,CONTINU
CFT2:	C2A8	78	MOV A,B		CHECK HIGHER BYTE
	C2A9	B7	ORA A		OF FALL TIME
	C2AA	CA B3 C2	JZ	CFTCMP	IF ZERO, JUMP TO CFTCMP;FALL TIME COMPLETE.
	C2AD	0E FF	MVI C	FFH	(BC)= (BC)-1
	C2AF	05	DCR B		
	C2B0	C3 A1 C2	JMP	CFT1	CONTINUE
CFTCMP:	C2B3	F1	POP	PSW	RESTORE VALUE
	C2B4	3D	DCR A		DECREMENT IT BY 1
	C2B5	21 40 C4	LXI H	C440H	IF A>TO, JUMP TO
	C2B8	BE	CMP M		VALOUT
	C2B9	C2 4B C2	JNZ	VALOUT	
	C2BC	F5	PUSH	PSW	SAVE (A)
	C2BD	AF	XRA A		CLEAR
	C2BE	32 4D C4	STA	C44DH	FLG2
	C2C1	3E 0F	MVI A	0FH	SET FLG1
	C2C3	32 4C C4	STA	C44CH	
	C2C6	F1	POP	PSW	RESTORE VALUE
	C2C7	C3 4B C2	JMP	VALOUT	CONTINUE
CONST:	C2CA	3A 4E C4	LDA	C44EH	TEST FLG3
	C2CD	B7	ORA A		IF IT IS ZERO
	C2CE	CA D4 C2	JZ	CFVL	JUMP CORRESPONDING FINAL VALUE CONSTANT
	C2D1	C3 00 C3	JMP	CLVL	OTHERWISE CORRESPONDING LOWER VALUE CONSTANT

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS		COMMENTS
CFVL:	C2D4	2A 46 C4	LHLD	C446H	GET COUNT CORRES-
	C2D7	44	MOV B,H		PONDING TO FINAL
	C2D8	4D	MOV C,L		VALUE IN (BC)
	C2D9	79	MOV A,C		GET LOWER BYTE OF COUNT IN (A)
	C2DA	B7	ORA A		IF ZERO,GO TO CFV2
	C2DB	CA E5 C2	JZ	CFV2	
CFV1:	C2DE	CD 30 C3	CALL	DEL15S	CALL 0.25S DELAY
	C2E1	0D	DCR C		DECREMENT LOWER BYTE,
	C2E2	C2 DE C2	JNZ	CFV1	IF NOT ZERO,CONTINUE
CFV2:	C2E5	78	MOV A,B		CHECK HIGHER BYTE
	C2E6	B7	ORA A		OF FINAL VALUE COUNT,
	C2E7	CA F0 C2	JZ	CFCMP	IF ZERO, GO TO CFCMP, FINAL VALUE TIME COMPLETED
	C2EA	0E FF	MVI C	FFH	(BC)= (BC)-1
	C2EC	05	DCR B		
	C2ED	C3 DE C2	JMP	CFV1	CONTINUE
CFCMP:	C2F0	AF	XRA A		CLEAR FLG1
	C2F1	32 4C C4	STA	C44CH	
	C2F4	3E 0F	MVI A	0FH	SET FLG3
	C2F6	32 4E C4	STA	C44EH	
	C2F9	F1	POP	PSW	RESTORE VALUE
	C2FA	C3 4B C2	JMP	VALOUT	JUMP TO VALOUT
	C2FD	00 00 00	NOP	NOP NOP	
CLVL:	C300	2A 4A C4	LHLD	C44AH	GET COUNT CORRES-
	C303	44	MOV B,H		PONDING TO LOWER
	C304	4D	MOV C,L		VALUE DURATION IN (BC)
	C305	79	MOV A,C		GET LOWER BYTE IN (A)
	C306	B7	ORA A		IF ZERO, JUMP TO
	C307	CA 11 C3	JZ	CLV2	CLV2

LEBEL	ADDRESS	CONTENTS	MNEMONICS AND OPERANDS	COMMENTS
CLV1:	C30A	CD 30 C3	CALL DEL15S	CALL 15 SECONDS DELAY
	C30D	0D	DCR C	DECREMENT LOWER BYTE
	C30E	C2 0A C3	JNZ CLV1	IF NOT ZERO,CONTINUE
CLV2:	C311	78	MOV A,B	CHECK HIGHER BYTE OF
	C312	B7	ORA A	LOWER VALUE TIME,IF
	C313	CA 1C C3	JZ CLCMP	ZERO JUMP TO CLCMP,
				TIME COMPLETED
	C316	0E FF	MVI C FFH	(BC)= (BC)-1
	C318	05	DCR B	
	C319	C3 0A C3	JMP CLV1	CONTINUE
CLCMP:	C31C	AF	XRA A	CLEAR FLG1
	C31D	32 4C C4	STA C44CH	
	C320	32 4E C4	STA C44EH	CLEAR FLG3
	C323	F1	POP PSW	RESTORE VALUE
	C324	C3 4B C2	JMP VALOUT	CONTINUE
	C327	00 00 00	NOP NOP NOP	

CHAPTER - 5

EXPERIMENTATION DISCUSSIONS CONCLUSIONS AND SUGGESTIONS

5.1 EXPERIMENTATION DISCUSSIONS AND CONCLUSION:

The details software developed for the SET POINT GENERATOR has been discussed in Chapter -4. The hardware software interrection is then tested using AMO2 MICRO DEVELOPMENT SYSTEM. The block diagram of a experimental set up is shown in Fig.5.1.

Necessary data inputs are then inputed as discussed in Chapter - 3. One sample inputs given are as follows:-

INITIAL VALUE (A) =

FINAL VALUE (B) =

SET TIME OA IN 15/30 MINUTES =

RISE TIME (AB) IN HOURS =

AND IN MINUTES =

FINAL CONST. TIME (BC) IN HOURS=

AND IN MINUTES =

FALL TIME (CD) IN HOURS =

AND IN MINUTES =

LOWER TIME CONST.(DE) IN HOURS =

AND IN MINUTES =

As soon as last two digits are given, program starts to execute.

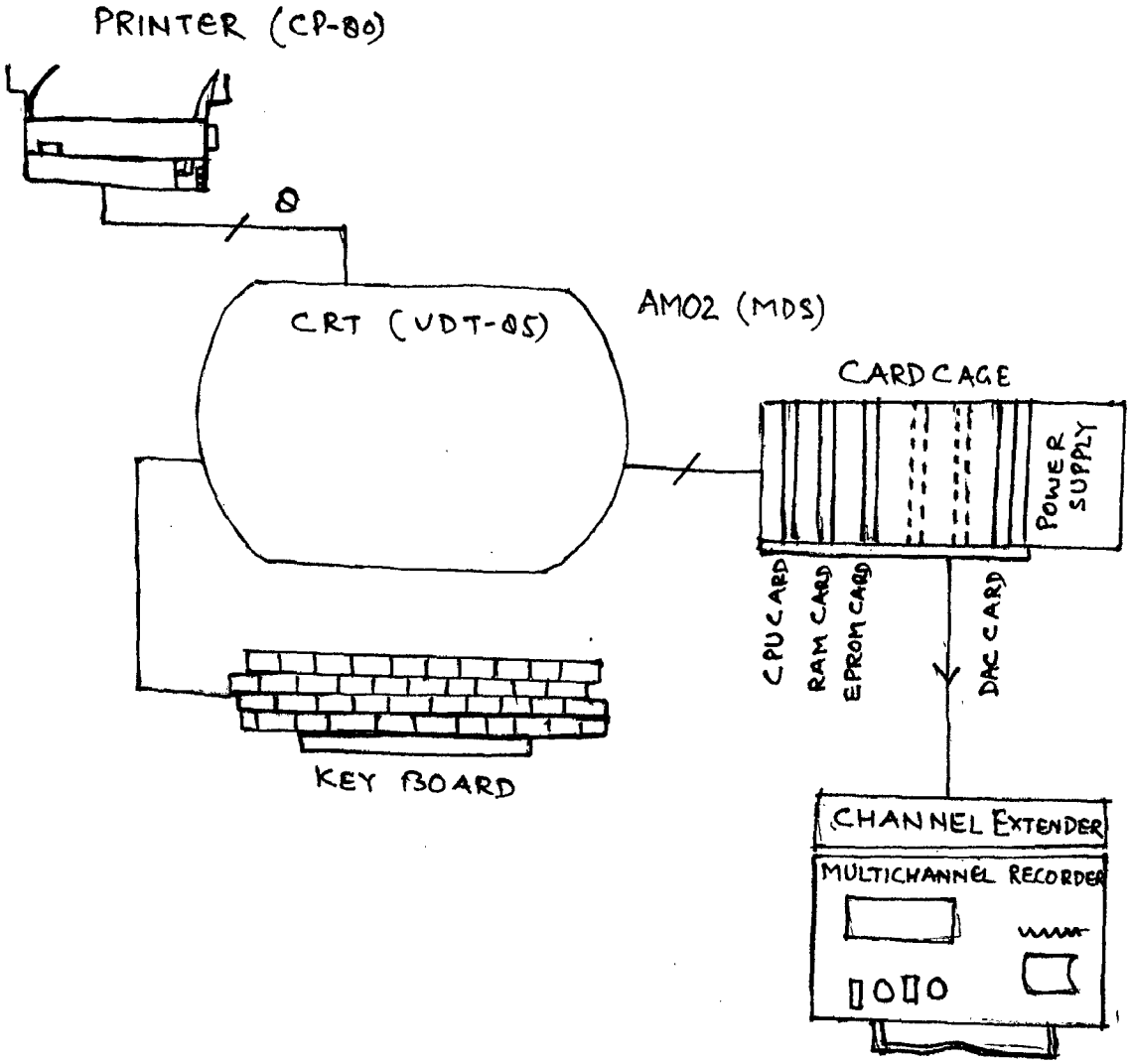


FIG-5-1, EXPERIMENTAL SETUP

Sample output printed out in digital multichannel recorder is showing in the Appendix-D.

In this duration we have inputed the necessary parameters through ASCII Key board of CRT (VDT-85); however in an industrial unit these parameters should be inputed by means of BCD thumbwheel switches through 8255 PPI Chips and multiplexer. Because of the nonavailability of thumbwheel switches this portion could not be implemented.

5.2 FURTHER SUGGESTION:

This Set Point Generator is for single channel. It may be made of multichannel also. Besides, many other shapes of cycle may be generated by SET POINT GENERATOR.

Time delay has been simulated through software, however if microprocessor has some useful work to perform during the time delay period like multi loop feed back control of the Environmental Test Chamber then hardware interrupt control for time delay using 8253 timer chip should be incorporated.

....

APPENDIX - A

INTEL 8085A : A BRIEF DESCRIPTION

A.1 SPECIAL FEATURES

8085A is a single chip, N-Channel Central processing unit having following features.

1. Single +5V supply.
2. 100% software compatibility with 8080A.
3. 1.3 us instruction cycle.
4. On chip clock generator (with external crystal, LC or RC network).
5. On chip system controller, advanced cycle status information available for large system control.
6. Four vectored interrupt inputs (one is non-maskable) plus an 8080A compatible interrupt.
7. Serial IN/OUT ports.
8. Decimal, binary and double precision arithematic.
9. Direct addressing capability to 64K bytes of memory.

A.2 FUNCTIONAL BLOCK DIAGRAM AND PINOUT DIAGRAM

Functional block diagram of internal architecture of 8080A is shown in Fig.A-1. The pin configuration of 8080A and logic representation are shown in Fig.A-2 and Fig.A-3 respectively.

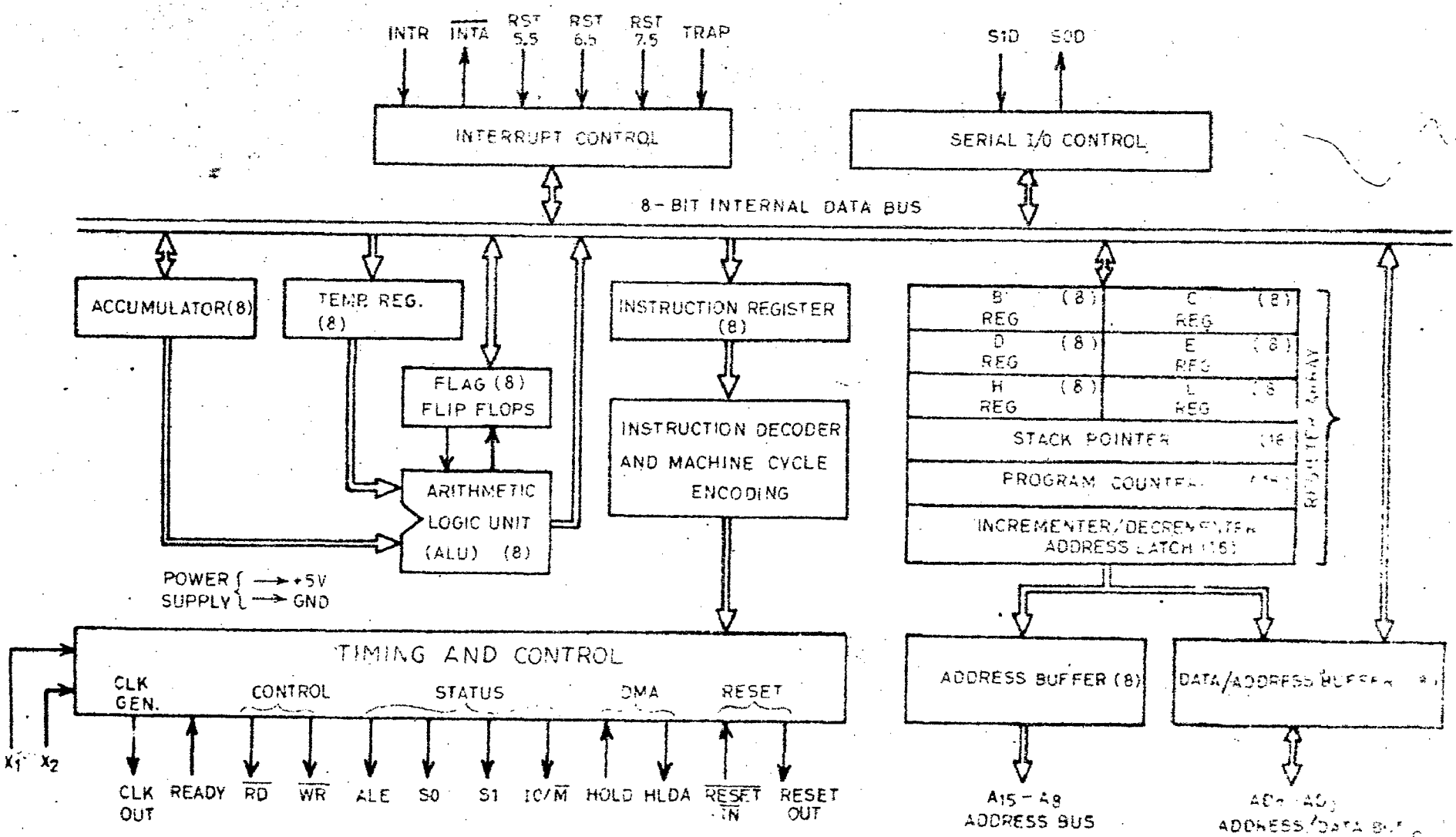


Fig. 1: 8085A CPU Architecture.

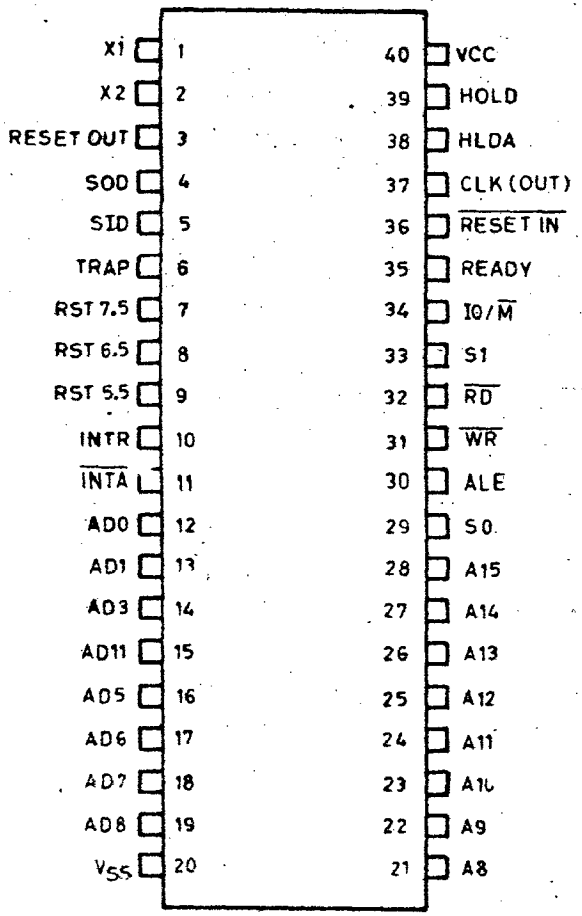


Fig A2 8085A Pin out diagram.

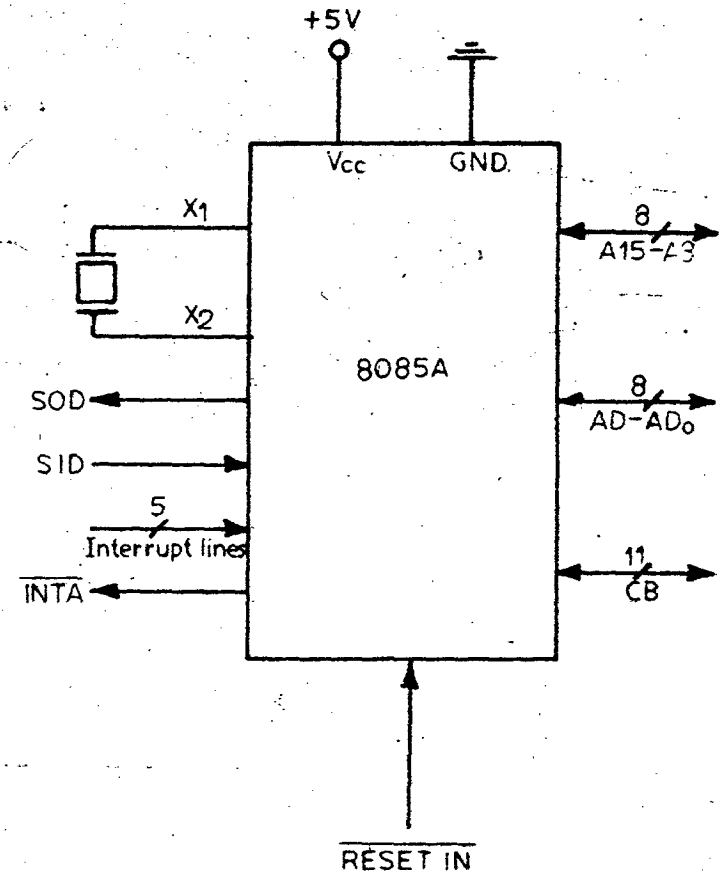


Fig A3 Logic Representation.

APPENDIX-B

PCB EDGE CONNECTOR CONNECTIONS

The PCB edge connector is as per STD BUS.

The STD BUS pin out is organised into four functional groups :-

* Dual Power Buses : Pins 1-6 and 53-56.
* Data Bus : Pins 7-14
* Address Bus : Pins 15-30
* Control Bus : Pins 31-52.

The organisation and pinouts are shown in Table B-1 and B-2. This table lists the mnemonic function and signal flow direction (referenced to the processor card in control of the Bus) for each pin of the STD BUS. The STD BUS is further defined as requiring a 56-pin (dual 28) card edge connector, with 0.125 in. Pin center. Connectors are on a spacing interval of 0-5 in centers minimum, and they accept the standard 4.5x6.5x0,062 in card.

TABLE B-1: STD BUS PINOUTS WITH SIGNAL FLOW REFERENCED TO THE PROCESSOR CARD

COMPONENT SIDE

	PIN	MNEMONIC	SIGNAL	DESCRIPTION
LOGIC	1	+5VDC	In	Logic Power (bussed)
POWER	3	GND	In	Logic Ground (bussed)
BUS	5	VBB	In	Logic Bias 1 (-5V)
DATA	7	D3	In/Out	Low-Order Data Bus
BUS	9	D2	In/Out	Low-Order Data Bus
	11	D1	In/Out	Low-Order Data Bus
	13	DO	In/Out	Low-Order Data Bus

	PIN	MNEMONIC	SIGNAL	DESCRIPTION
ADDRESS	15	A7	Out	Low-Order Address Bus
BUS	17	A6	Out	Low-Order Address Bus
	19	A5	Out	Low-Order Address Bus
	21	A4	Out	Low-Order Address Bus
	23	A3	Out	Low-Order Address Bus
	25	A2	Out	Low-Order Address Bus
	27	A1	Out	Low-Order Address Bus
	29	A0	Out	Low-Order Address Bus
	31	WR*	Out	Write to Memory or I/O
	33	IORQ*	Out	I/O Address Select
	35	IOEXP	In/Out	I/O Expansion
	37	REFRESH*	Out	Refresh Timing
	39	STATUSI*	Out	CPU Status
CONTROL	41	BUSAK*	Out	Bus Acknowledge
BUS	43	INTAK*	Out	Interrupt Acknowledge
	45	WAITRQ*	In	Wait Request
	47	SYSRESET*	Out	System Reset
	49	CLOCK*	Out	Clock Processor
	51	PCO	Out	Priority Chain Out
AUXILI-	53	ADX	In	AUX Ground (Bussed)
ARY POWER BUS	55	GND AUX+V	In	AUX Positive (+12V DC)

TABLE B-2: STD BUS PINOUTS WITH SIGNAL FLOW REFERENCED TO THE PROCESSOR CARD

C I R C U I T S I D E

	PIN	MNEMONIC	SIGNAL FLOW	DESCRIPTION
LOGIC POWER BUS	2	+5VDC	In	Logic Power (Bussed)
	4	GND	In	Logic Ground (Bussed)
	6	VBB≠2	In	Logic Bias 2(-5V)
DATA BUS	8	D7	In/Out	High-Order Data Bus
	10	D6	In/Out	High-Order Data Bus
	12	D5	In/Out	High-Order Data Bus
	14	D4	In/Out	High-Order Data Bus
ADDRESS BUS	16	A15	Out	High-Order Address Bus
	18	A14	Out	High-Order Address Bus
	20	A13	Out	High-Order Address Bus
	22	A12	Out	High-Order Address Bus
	24	A11	Out	High-Order Address Bus
	26	A10	Out	High-Order Address Bus
	28	A9	Out	High-Order Address Bus
	30	A8	Out	High-Order Address Bus
CONTROL BUS	32	RD*	Out	Read Memory or I/O
	34	MEMRQ*	Out	Memory Address Select
	36	MEMEX*	In/Out	Memory expansion
	38	MCSYNC*	Out	CPU Machine Cycle Sync.
	40	STATUSO*	Out	CPU Status
	42	BUSRQ*	In	BUS request
	44	INTRQ*	In	Interrupt request
	46	NMIRQ	In	Nonmaskable interrupt.
	48	PERESET*	In	Push-Button Reset
	50	CNTRL*	In	AUX Timing
	52	PCI	In	Priority Chain in

	PIN	MNEMONIC	SIGNAL FLOW	DESCRIPTION
AUXILI-	54	AUXGND	In	AUX Ground (Bussed)
ARY	56	AUX-V	In	AUX Negative (-12V DC)
POWER				
BUS				

B.1 STD BUS :

STD means simple-to-debug, simple-to-develop and swift-to-deliver. It is also an abbreviation for the word standard, as it is the microprocessor system bus adopted as a standard by Pro-Log, MOSTEC and others.

The bus has four sections : power distribution, an eight-bit bidirectional Data Bus, a 16-bit Address Bus and a Control Bus. All data and address lines are tri-state buffered. Control lines include controls for memory and I/O bus access, interrupts, and lines for CPU actions. Power consists of +5 volts for logic and +12 volts for auxiliary power.

B.2 POWER DISTRIBUTION :

The STD Bus contains a logic power Bus, Data Bus, Address Bus, Control Bus and Analog Power Bus. The principle logic Power Supply is +5 volts. Two bused pin pairs are provided at one card edge to allow both ample contact area and sufficient width. At the opposite end of the connector, single opposed pins supply +12 volts. This

supplies the twenty-four volts likely to be needed for analog interface and industrial controls. Next in from the card edge, at each end, are bused ground lines. Pins three and four are bused to provide logic ground and pins fifty-three and fifty-four are bused to provide auxiliary ground. A second logic voltage (-5 volts DC) is provided at pins five and six. This source could provide a substrate voltage for dynamic memories.

B.3 DATA BUS :

All compatible functions are grouped together on the Bus connectors. The data bus is bidirectional and uses pins 7 and 14. Data lines are interfaced using tri-state buffer drivers.

B.4 ADDRESS BUS :

The next sixteen pins, pins 15 through 30, provide a sixteen lines address bus. The lower order sight address lines are odd numbered pins and the higher address lines are on even numbered pins. A with the Data Bus, all connections to tethe Address Bus are through tri-state buffer drivers.

B.5 CONTROL BUS :

Pins 31 and 32 provide write and read commands respectively. These signals are ANDed with the memory request on pin 34 to address memory or the I/O request

on pin 33 to address Input or Output. Pins 35 and 36 provide and I/O and memory expansion capability. Expansion may be implemented in line as a 17th bit to be decoded or each memory card. A processor card on board port with an output latch could be used to select the upper or lower bank of memory. The I/O expansion line allows a subroutine using absolute I/O port addresses to be applied to more than one set of I/O devices by simply swapping banks. Pin 37 provides refresh timing information for dynamix memories. The Z80 microprocessor has a refresh register on the CPU chip; in other systems, the refresh address counter would be implemented separately on the processor card or other auxiliary card.

Pins 42 and 41 provide a bus request and bus acknowledge allowing secondary controllers access to data and address buses. The set of signals is necessary for implementation of direct memory access (DMA) operations and facilities multi-processing.

Pins 44 and 43, interrupt request and interrupt acknowledge, provide a means for implementing a fixed priority interrupt scheme such as the 8080 microprocessor set.

Used with the non-masking interrupt signals (Pin 46). One level of priority interrupt is provided. Vectoring (e.g., by placing an 8080 Restart instructions

on pin 33 to address Input or Output. Pins 35 and 36 provide and I/O and memory expansion capability. Expansion may be implemented in line as a 17th bit to be decoded on each memory card. A processor card on board port with an output latch could be used to select the upper or lower bank of memory. The I/O expansion line allows a subroutine using absolute I/O port addresses to be applied to more than one set of I/O devices by simply swapping banks. Pin 37 provides refresh timing information for dynamix memories. The Z80 microprocessor has a refresh register on the CPU chip; in other systems, the refresh address counter would be implemented separately on the processor card or other auxiliary card.

Pins 42 and 41 provide a bus request and bus acknowledge allowing secondary controllers access to data and address buses. The set of signals is necessary for implementation of direct memory access (DMA) operations and facilities multi-processing.

Pins 44 and 43, interrupt request and interrupt acknowledge, provide a means for implementing a fixed priority interrupt scheme such as the 8080 microprocessor set.

Used with the non-masking interrupt signals (Pin 46). One level of priority interrupt is provided. Vectoring (e.g., by placing an 8080 Restart instructions

on the data bus at interrupt acknowledge time) would be required for further interrupt expansion. The non-maskable interrupt also provides for a power failure interrupt. Pins 51 and 52 used in conjunction, provide for a daisy-chain. Priority is established by physically positioning the highest priority card to the right of the next highest and so on. Priority I/O operations, interrupt driven applications and possible implementation of multi-processors are served by these two lines.

Machine Cycle Syno. pin 38 is a three state active low processor output signal that occurs once during each processor machine cycle. A machine cycle is defined as the sequence that involves addressing, data transfer and execution. Machine Cycle sync. defines the beginning of the machine cycle. The exact nature and timing of this signal is microprocessor dependent. Status 1, pin 39, and status 0, pin 40, provide timing information related to special machine cycle operations. Where specifically available, status is considered to be a signal to identify instruction fetch. Wait Request, pin 45, is an active low input line to the processor that suspends processor operations as long as it remains during the wait. The wait request allows aslow acess time devices to be used on the same bus as a faster processor.

System Reset, Pin 47, provides a debounced reset

System Reset, Pin 47, provides a debounced reset signal to all cards in the system requiring initial system re-setting. Pin 48, Push Button Reset, allows the source of the reset condition to be other than the processor card; it provides an active low input line to the system reset circuit. Pin 49 provides a buffered processor clock signal used for system synchornisation or as a general clock source. Pin 50 provides a circuit for auxiliary clock timing. It may be a multiple of the processor clock signal or a real time clock signal. It allows slow speed devices to use a common clock.

A P P E N D I X - C

SOFTWARE DEMAND DESCRIPTION

1. MONITOR COMMAND GROUP :

1.1 Break Point Command - BP

FORMAT : BP CR
BREAK ADR/NAME XXXX CR
PROG ADR/NAME XXXX ∅ AF ∅ BC ∅ DE ∅ HL ∅
or PROG ADR/NAME XXXX CR.

1.2 Break Point Restore - BR :

FORMAT : 'BR' CR

1.3 Move Memory Commands - MM :

FORMAT : MM CR
MEM ADR 'STARTING', 'ENDING' CR
TOP ADR X X X X CR
BOT ADR X X X X CR
MEM ADR N N N N N N N N

1.4 Memory Compare Command MC

FORMAT : MC CR
MEM ADR
01 'STARTING', 'ENDING', CR
02 'STARTING' CR
DONE if successfuly
ADR CONTENT ADR CONTENT
(First Block) (Second Block) CR/ESC.

1.5 Memory Fill - MF :

FORMAT : MF CR
MEM ADR XXXX, XXXX CR
FILL WITH XX CR.
DONE IF SUCCESSFUL
ADR CONTENT (if unsuccessful) CR/ESC

1.6 Go to user program command - GO

FORMAT : GO CR
PROG ADR/NAME XXXX ∅ / CR

1.7 Input and substitute command - IP

FORMAT : IP CR
RAM ADR 'STARTING ADR' CR
NNNN XXXXXX ∅ NN - XX ∅ NN-XXXX
∅ NNNN - XXXXXX CR
NNNN ∅ NN-XXXXXX ∅ NNNN-XX ∅ NNNNNN - XX C

1.8 Input Data - ID

FORMAT : ID CR
RAM ADR 'STARTING ADR' CR
NNNN NN - ∅ NN - ∅ NN - ∅ NN - CR
NNNN NN - ∅ NN - ∅ NN - ∅ NN - ∅ NN - ∅
NN - ∅ NN - CR
NNNN ∅ NN - XX ∅ NN - XX ESC

1.9 Go to subroutine command - GS :

FORMAT : GS CR
SUBR ADR/NAME 'XXXX' ∅ AF-XXXX ∅ BC-XXXX ∅
DE-XXXX ∅ HL-XXXX CR
or SUBR ADR/NAME 'XXXX' CR.

1.10 Tape write - TW

FORMAT : TW CR
NO OF BLOCKS X CR
BLOCK ADR
01 XXXX, XXXX CR
02 XXXX, XXXX CR

1.11 Read from Tape - TR

FORMAT : TR CR
Offset adr XXXX CR

1.12 Display Program - DP

FORMAT : DP CR
PROG ADR 'Starting Adr', 'End Adr' CR
NNNN XX CR
NNNN XX XX XX CR | Binary expansion
NNNN XX XX ESC

2. PROGRAM RELOCATION COMMANDS :

2.1 RL - Relocate the Program :

FORMAT : RL CR
PROG ADR ' STARTING ADR', 'ENDING ADR' CR
RELOC ADR 'XXXX' CR
RELOC ADR 'NNNN', 'NNNN'

2.2 RE - Relocate external program commands :

FORMAT : RE CR
PROG 'STARTING', 'ENDING' CR
RAM ADR 'XXXX' CR
REV RELOC TO NNNN, NNNN.

2.3 RR - Relocate Reverse Command :

FORMAT : RR CR
PROG ADR 'STARTING', 'ENDING' CR
REV RELOC TO 'XXXX' CR
REV RELOC TO NNNN, NNNN

2.4 ST - Shift Command :

FORMAT : ST CR
PROG ADR 'STARTING ADR', 'ENDING ADR' CR
SHIFT ADR 'FIRST ADR', 'SECOND ADR' CR
PROG ADR NNNN, NNNN

2.5 RK - Relocate and link command :

FORMAT : RK CR
NO OF BLOCK X CR
BLOCK ADR
01 ϕ 'START', 'END' CR
02 ϕ 'START', 'END' CR
RAM ADR XXXX CR
PROG ADR NNNN, NNNN
BLOCK ADR
X ϕ 'NNNN', NNNN

2.6 RM - Relocate Multiprogram command

FORMAT : RM CR
NO OF BLOCKS 'X' CR
BLOCK ADR
01 'START', 'END' CR
02 'START','END', CR
RELOC ADR
01 'XXXX' CR
02 'XXXX' CR
BLOCK ADR
X b ϕ NNNN, NNNN
X ϕ NNNN, NNNN

2.7 CA - Code modification with address print command, and

2.8 CM - Code modification

FORMAT : CA CR (or CM CR)
PROG ADR 'STARTING', 'ENDING' CR
OLD CODE XX ⌀ XX ⌀ XX CR.
NEW CODE XX ⌀ XX ⌀ XX CR.
NNNN (Adr print for CA Only)

2.9 MA - Modify adr. command OR

2.10 MP - Modify adr. and Print Command

FORMAT : MA CR (or MP CR)
PROG ADR 'STARTING', 'ENDING' CR
CODE XX, XX, -----, XX CR
OLD ADR XXXX, XXXX CR
NEW ADR XXXX, CR
NEW ADR NNNN NNNN
ADR
ADR | FOR MP ONLY

3. DE ASSEMBLER/PROGRAM DOCUMENTATION GROUP :

3.1 LS - List Command (without comments) :

FORMAT : LS CR
USER NAME 'Programmer's name' CR
PROG NAME 'Program name' CR
IN 'Inputs to program' CR
OUT 'Outputs from program' CR

3.2 LA - List ASC-II data :

FORMAT : LA CR PAGE NO. XXCR
DATA/LINE 'X' CR
ASCII DATA ADR 'STARTING', 'ENDING' CR
PROG NAME 'PROG.NAME'
LIST ADR 'STARTING'
ADR CODE OF CHARACTER CHARACTER

3.3 LD - List Data Command :

FORMAT : LD CR
DATA/LINE 'X' CR
DATA ADR 'STARTING', 'ENDING' CR
PROG NAME 'PROGRAM NAME' CR
LIST ADR 'STARTING' CR

3.4 CV-Comment Entry on VDU :

FORMAT : CV CR
PROG ADR 'STARTING ', 'ENDING CR
LIST ADR XXXX CR

3.5 CT - Comment Entry on TTY :

FORMAT : CT CR
PROG ADR 'STARTING','ENDING' CR
LIST ADR XXXX CR

The information is displayed in following format :

ADR * CODE * COMMENT ENTRY

3.6 LC - List with Comments :

This command is similar to LS command in all respect, except that it prints the comments which were entered through CV/CT command.

3.7 LV - List PROG ON VDU :

FORMAT : LV CR
PROG ADR 'STARTING', 'ENDING' CR
LIST ADR 'STARTING' CR

3.8 AV - List ASCII Data Command :

FORMAT : AV CR
DATA/LINE 'X' CR
ASCII DATA ADR 'STARTING', 'ENDING' CR
ADR CHARACTER - CODES CHARACTER

3.9 DV - List Data on VDU :

FORMAT : DV CR
DATA/LINE 'X', CR
DATA ADR 'STARTING', 'ENDING CR

4. PROM PROGRAMMING GROUP :

4.1 PE - PROM Empty Test Command :

FORMAT : PR CR
DONE if successful
ADR CONTENT (if unsuccessfuly/waits for further check)

4.2 PL - List PROM Command :

FORMAT : PL CR
ROM ADR 'STARTING', 'ENDING' CR

4.3 PM - PROM and Memory Compare Command :

FORMAT : PM CR
MEM ADR 'STARTING', 'ENDING', CR
ROM ADR 'XXXX' CR
DONE if successful
ADR CONTENT (of the PROM) ADR CONTENT (of the memory)

4.4 PR - PROM read into RAM Command :

FORMAT : PR CR
ROM ADR 'STARTING', 'ENDING' CR
RAM ADR 'XXXX' CR
END ADR NNNN

4.5 PT - PROM Test Command :

FORMAT : PT CR
ROM ADR 'STARTING','ENDING' CR
CMP WITH 'XX' CR
DONE if successful
ADR CONTENT if unsuccessful CR

4.6 PP - PROM Programming

FORMAT : PP CR
PROGRAMMING NNNN CR
MEM ADR 'STARTING', 'ENDING' CR
ROM ADR 'XXXX' CR
DONE if successful
ADR CONTENT (of PROM) ADR CONTENT (of MEM)

5. ASSEMBLER GROUP :

5.1 AS Assemble PROG (NEW) :

ASSEMBLER :

FORMAT : AS CR
RAM ADR XXXX ∅
NNNN Mnemonic ∅ O-Operand label A-adr label C
NNNN

NNNN END ∅

PROG ADR NNNN NNNN

DEFINE OPERAND LABEL

(LABEL) XXXX CR

(LABEL) XXXX CR

LABEL ADR NNNN, NNNN

NO OF LABEL NN

5.2 RG - Restart Assembly :

FORMAT : RG CR

RAM ADR XXXX ∅

NNNN Mnemonic ∅ O- operand label ∅ Δ-adr.label ∅

5.3 ΔI - Assembler Intermediate

While entering a long program, if you want to intermediately stop the entry of program this can be achieved by giving ENDI pseudo command. The system will not ask for any definition of the operand labels. When you want to restart the entry of program, from the place (address) you have left, the ΔI Command should be used.

5.4 LD - Label Display :

FORMAT : LD CR

LABEL NO XX CR

The following is the format displayed

NO LABEL ADR

NN NNNNNN NNNN CR

NN NNNNNN NNNN ESC

5.5 Label Information-LX :

FORMAT : LX CR
LABEL ADR NNNN NNNN
NO OF LABEL NN

5.6 LN - Label Number and Address

FORMAT : LN CR
LABEL NAME - XXXXXX CR
Adr No.
NNNN NN.
LABEL NAME .ESC.

5.7 LM - Label Modify :

FORMAT : LM CR
LABEL NO -- XX CR
NO LABEL ADR
XX XXXXXX XXXX CR
LABEL NO -- ESC

5.8 LE : Label Equate :

FORMAT : LE CR
LABEL NO XX CR
NO LABEL ADR
NN XXXXXX XXXX CR
NN XXXXXX XXXX

5.9 LR - Label Remove :

FORMAT : LR CR
LABEL NO XX CR

5.10 LL - List Label :

FORMAT : LL CR
Label No XX CR PAGE XXCR
NO. OF LABEL XX CR
CR
LABEL NO XX CR
NO OF LABEL CR
LABEL NO XX CR

5.11 LP - Label Program :

FORMAT : LP CR
PROG ADR 'STARTING', 'ENDING' 'ADR' CR
PROG NAME XXXX CR
SUBR NAME XXXX CR

5.12 XR - Cross Ref. :

FORMAT : XR CR
PROG ADR 'STARTING', 'ENDING' CR
LABEL NAME/ADR XXXX CR
NNNNNN NNNN

5.13 XL - Label Cross Ref. List :

FORMAT : XL CR
* PAGE XXCR
PROG ADR 'STARTING', 'END' CR
NNNNNN NNNN
NNNN NNNN ------ NNNN
.
.
NNNNNN NNNN
NNNN NNNN ------ NNNN

```
.                .
.                .
.
NNNN             NNNN
                      NNNN NNNN - - - NNNN
                      NNNN NNNN - - - NNNN.
```

(till the end of the address label table).

5.14 RA - Relocate the Assembled Program :

FORMAT : RA CR
PROG ADR (STARTING) , (ENDING) CR.
RELOC ADR XXXX CR
RELOC ADR NNNN, NNNN

5.15 AR - Reverse Relocate Assembled Program :

FORMAT : AR CR
PROG ADR (STARTING), (ENDING) CR
REV RELOC TO XXXX CR
REV RELOC TO NNNN, NNNN

5.16 SA - Shift the Assembled Program :

FORMAT : SA CR
PROG ADR (STARTING),(ENDING) CR
SHIFT ADR (FROM) (TO) CR
PROG ADR NNNN, NNNN

5.17 ER - Relocate External Assembled Program :

FORMAT : ER CR
PROG ADR 'STARTING', 'ENDING' CR
RELOC ADR XXXX CR
REV RELOC TO NNNN, NNNN

6. PSEUDO COMMANDS :

6.1 DA - Data ASCII :

FORMAT : NNNN DA ∅ STRING OF ASCII CHARACTERS
esc. SA - Address Label ∅

6.2 DB - Data Byte :

FORMAT : NNNN DB ∅ XX, XX, XX, XX, ∅ Δ - Address label ∅

6.3 DS - Data Storage :

FORMAT : DS ∅ XX ∅ Δ - Address Label ∅

6.4 DW - Data Word :

FORMAT : NNNN DW ∅ XXXX, XXXX, XXXX, ∅ Δ - Address label ∅

6.5 END - End of Assembly

FORMAT : NNNN, END ∅
PROG ADR (STARTING), (ENDING)
DEFNINING OPERAND LABEL
(OPLABEL)XXXX CR
(OPLABEL)XXXX CR.

(OPLABEL) XXXX CR
LABEL ADR NNNN, NNNN
NO OF LABEL NN

6.6 GB - Back in the Program :

FORMAT : NNNN GB ↵
RAM ADR XXXX ↵
NNNN

6.7 ENDI - End of Assembly Intermediate :

FORMAT : NNNN ENDI ↵
PROG ADR NNNN, NNNN
OPERAND LABEL AREA NNNN, NNNN
ADR LABEL NNNN, NNNN
NO OF LABEL NN.

7. MISCELLANEOUS COMMANDS :

7.1 DH - Decimal to hex conversion :

FORMAT : D - XXXXX H - NNNN D -

7.2 HA - Hex arithmatic command

FORMAT : A XXXX ↵ B XXXX ↵ A+B = NNNN ↵
A-B = NNNN.

7.3 HD - Hex to decimal conversion command :

FORMAT : H - XXXX D- NNNN H -

7.4 OH - Octal to hex conversion :

FORMAT : A - XXXX ↵ H - NNNN O -

APPENDIX-D

MULTI-CHANNEL RECORDER

D-1 INTRODUCTION

The TR2721A Multi-Channel Digital Recorder is a compact, multi-purpose, high performance data recorder.

It measures temperatures, DC voltages, or converter outputs of up to 12 channels (24 channels with channel extender) at a preset time interval and records the measured results on a built-in thermal printer. The measurement time and channel number are printed along with the measured result.

D-2 FEATURES

[1] Easy-to-use handy type

The TR2721A is a handy type that can be easily carried anywhere. Simple panel design permits immediate use even by beginners.

[2] Any desired channel can be monitored.

Monitor function permits repeated measurement of any desired channel without regard to the last channel. This function is extremely effective in temperature distribution measurement; especially temperature monitoring of important points.

[3] Ten standard ranges, including percentage (%) scaling JIS T(CC), J(IC), E(CRC), K(CA), AND

R(PR) thermocouple temperature measurement, plus ±200 mV, ±2 V, and 1 V to 5 V DC voltage measurement are available.

Scaling function capable of displaying and printing 1 V to 5 V converter outputs (4 mA to 20 mA by inserting a 250 resistance) as percentage (0 to 100%) is convenient in measuring process amount in the industrial instrumentation field.

[4] Temperature and voltage difference measuring function

Difference function measures the temperature of DC voltage difference between each measurement channel and input channel 01. This function is extremely effective when performed for a temperature rise test (temperature difference from room temperature) of electrical home appliances such as refrigerators, televisions, stereo sets, etc., temperature difference measurement between the inlet and outlet of air conditioners and thermal converters, and in voltage difference measurement of electronic parts and circuits relative to a reference value.

Simultaneous displaying and printing of the result eliminates troublesome data recording.

[5] Trend printing mode indicates changes in the data of each channel with time at a glance

The TR2721A has a NORMAL printing mode convenient in checking the value and distribution of temperatures and voltages from channel 01 to a preset last channel and a TREND printing mode that indicates the changes in each measurement point with time.

The printed result can be directly used as supplemental data.

REFERENCES

[1] 'USER'S MANUAL FOR AMO2 MICRO DEVELOPMENT SYSTEM.

[2] USER'S MANUAL FOR VINYTICS DATA TERMINAL.

[3] LINEAR DATA BOOK, National Semi-conductor.

[4] SHRIVASTAVA VAGEESH; Microprocessor Based Inventory Control and Cash Register - Dissertation, 1984.

[5] MALVINO A.P. - Digital Computer Electronic - An Introduction to Microcomputer (Second Edition).

[6] Instruction Manual - TR2721A, Multi-Channel Digital Recorder.

[7] Operational Instruction Manual of Environmental Test Chamber.

[8] HALSALL F./LISTER P.F.; Microprocessor Fundamentals

[9] INDIAN STANDARD, IS:2106 Part (XVII) - 1973

[10] Hindustan Instruments Limited - Microprocessor Trainer, HIL-2961 User's Manual.

[11] TOCCI, Ronald J. and Laskowski, Lester. 'Microprocessors and Microcomputers: Hardware and Software' Englewood Cliffs: Printice Hall Inc., 1979

[12] TOCCI, Ronald J. 'Digital System: Principles and Applications', Englewood Cliffs: Printice Hall Inc., 1977.

[13] Rafiquzzaman, Mohamed 'Microcomputer Theory and Applications with the INTEL SDK-85', New York: John Wiley and Sons, Inc., 1982.

[14] Garland, Harry, 'Introduction to Microprocessor System Design', Tokyo: McGraw Hill Kogakusha Ltd., 1979.

[15] 'Microprocessors and Programming Logic: Printice Hall, 1981.

[16] Capt. LOHUMI H.C. - Microprocessor Based IC Testers - Dissertation, 1982.

...