Experiment 11

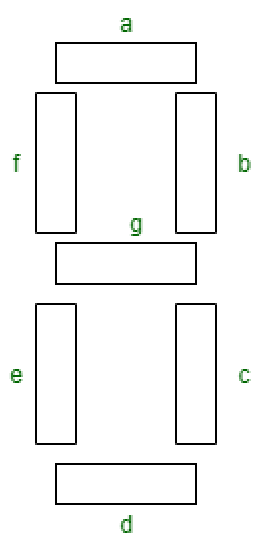
Experiment no. 11: Case Study on Interfacing Seven Segment Display with 8086

Roll No.	17
Name	Manav Jawrani
Class	D10-A
Subject	Microprocessor Lab
LO Mapped	LO6: Design interfacing of peripheral devices with 8086 microprocessor

Aim: Case Study on Interfacing Seven Segment Display with 8086.

Introduction:

Seven Segment Displays: Seven segment displays are the output display device that provide a way to display information in the form of image or text or decimal numbers which is an alternative to the more complex dot matrix displays. It is widely used in digital clocks, basic calculators, electronic meters, and other electronic devices that display numerical information. It consists of seven segments of light emitting diodes (LEDs) which are assembled like numerical 8.



Theory:

Abstract—

A wide variety of digital and analog devices can be interfaced with the microprocessor. Microprocessor is a complete CPU on a single chip and is compact and powerful and plays a vital role in various applications. The desired circuitry and software is needed to perform the interfacing. Before a circuit or device is interfaced to the microprocessor, the terminal characteristics of the microprocessor and its associated interfacing components must be known. Interfacing in a microprocessor is nothing but an integrated circuit that performs the fundamental functions of the central processing unit. It allows the user to

communicate with the computer. There are various versions of the Microprocessor available. A common way of categorizing the microprocessors is by the number of bits their ALU can work with at a time. A microprocessor with a 16-bit ALU is referred to as a 16-bit microprocessor. The 8086 Microprocessor is a classic example of a 16-bit microprocessor. The state of the art presented in this paper is the interfacing of a multiplexed seven segment display with the 8086 Microprocessor. Assembly language program was designed and developed to display text messages “COOL” and “DUDE” alternately with flickering effects on a multiplexed 7-segment display interface for a suitable period of time.

Keywords—
Digital devices, analog devices, interfacing, assembly language program, multiplexed seven segment display, 8086 microprocessor, text messages.

I. INTRODUCTION

Information processed by the microprocessor, in many microprocessor based instruments, has to be displayed in intelligible form to the user. A number of schemes have been evolved which fall broadly into two categories- multiplexed and non-multiplexed displays. In the case of multiplexed displays each digit is turned ON for a short interval of time after latching the data corresponding to that digit and this is repeated at least 70 to 80 Hz to ensure that the display is stationary. In non multiplexed displays each digit, the character of the display is always ON and separate data/latched inputs have to be provided for each digit/character. Alphanumeric LED displays are available in three common formats. For displaying only numbers, letters and hexadecimal letters, simple 7 segment displays are used. To display numbers and the entire alphabet 18 segment displays or 5 by 7 dot matrix displays can be used. The 7 segment type is the least expensive, most commonly used and easiest to interface. Figure 1 shows a seven segment indicator comprising seven LEDs labeled a through g and h is the DP. By forward biasing different LEDs, the digits 0 through 9 as well as some of the alphabets can be displayed. For instance, to display 0 segments a, b, c, d, e and f should be lighted up and to display 5 segments a, c, d, f and g should be lighted up. Similarly in the case of alphabets display such as U segments b, c, d, e and f should be lighted up and to display letter C segments a, d, e and f should be lighted up [1].

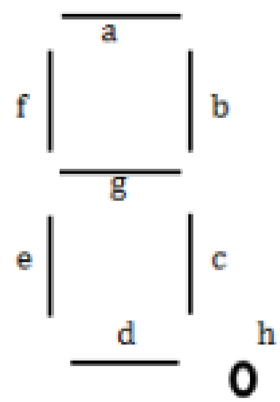
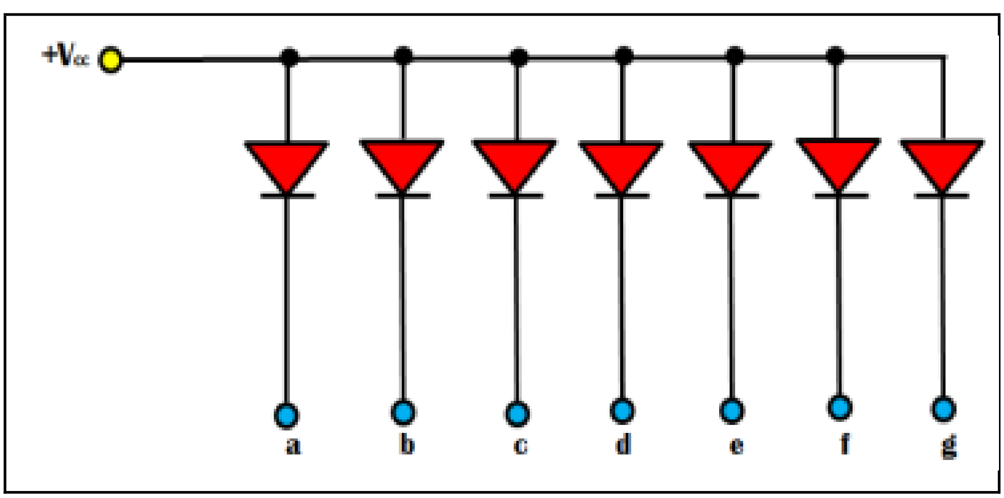


Fig.1 Seven segment indicator

Seven segment indicators are of two type's namely common anode type and common cathode type. In a common anode type all the anodes are connected together as illustrated in figure 2, whereas in a common cathode type all the cathodes are connected together as depicted in figure



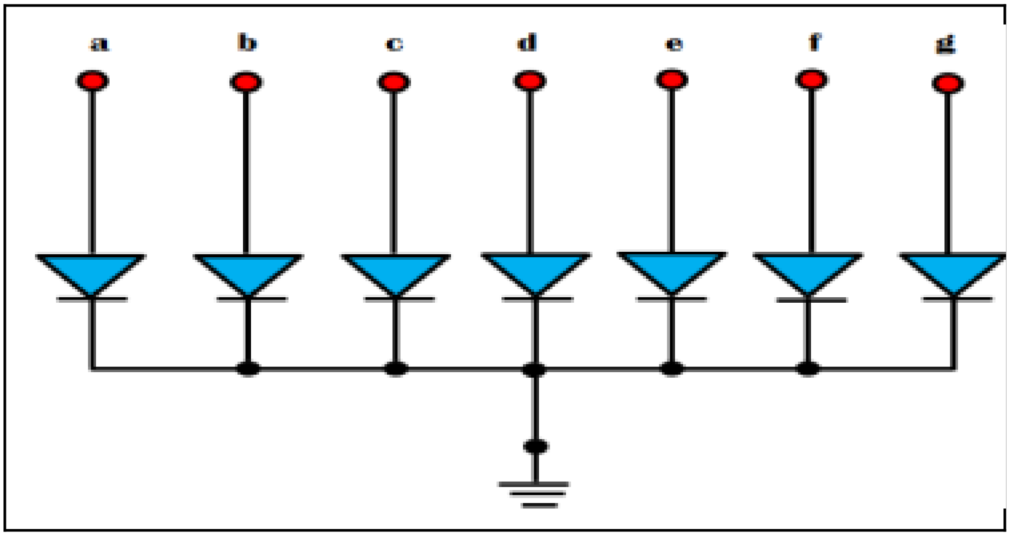


Fig.3 common cathode type seven segment indicator

In this paper the interfacing of the multiplexed seven segment display with the 8086 microprocessor is presented. The rest of the paper is organized into sections as follows: section II illustrates the fundamentals of the 8086 Microprocessor. Section III focuses on the system design. Results and discussion are reported in section IV. Finally section V summarizes the paper and presents the concluding remark.

II. 8086 MICROPROCESSOR FUNDAMENTALS

Microprocessor also known as the CPU is the heart of the computer system. Microprocessor accomplishes three vital tasks for the computer system. First, it is involved in the data transfer between itself and the memory or Input/output systems. Second, it performs arithmetic and logic tasks. Third, it aids program flow via simple decisions. The Intel 8086 is a 16 bit microprocessor that is intended to be used as the central processing unit (CPU) in a microcomputer. The term 16 bit implies that its internal registers, Arithmetic and Logic unit (ALU) and many of its instructions are designed to function with 16 bit binary data termed as word. The 8086 microprocessor mainly consists of a CPU, memory and ports. These parts are connected together by three buses namely data bus, address bus and control bus. The 8086 microprocessor possesses a 16 bit data bus, so it can read data from or write data to memory or ports that are either 16 bits or 8 bits at a time. It has a 20 bit address bus, so it can address any one of the 220 or 1048576 memory locations. The basic control bus consists of the signals labeled M/IO, RD and WR. If the 8086 is doing a read from the memory or port, the RD will be asserted. If the 8086 is doing a write to memory or to a port, the WR signal will be asserted. During a memory read or memory write, the M/IO signal will be high and during port operations M/IO signal will be low. The salient feature of the 8086 CPU is that it is divided into two independent functional parts namely the Bus interface unit (BIU) and the Execution unit (EU). The work is divided between these two units as a result of which there is an increase in processing speed of the processor.

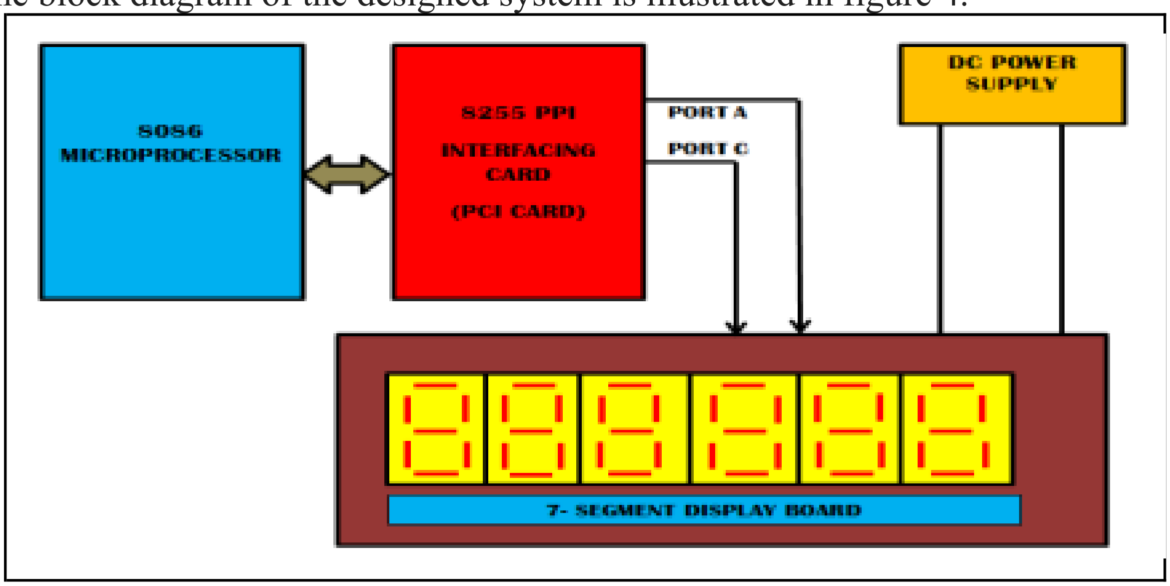
The 8086 microprocessor executes instructions in as little as 400ns (2.5 MIPS or 2.5

millions of instructions per second). In addition the 8086 microprocessor addressed 1 Mega Byte (MB) of memory. A 1MB of memory contains 1024 Kilobytes sized memory locations or 1048576 bytes. This higher execution speed and larger memory compared to earlier versions of the microprocessor allowed the 8086 microprocessor to replace smaller minicomputers in many applications. A special feature found in the 8086 microprocessor was a small 4 or 6 byte instruction cache or queue that pre fetched a few instructions before they were executed. The queue speeded the operation of many sequences of instructions and proved to be the basis for much larger instruction caches found in modern microprocessors. The increased memory size and additional instructions in the 8086 have led to many sophisticated applications for the microprocessor. 8086 is also called CISC (complex instruction set computers) because of the number and complexity of instructions. Additional instructions eased the task of developing efficient and much sophisticated applications. Furthermore, the 8086 microprocessor also provided much internal register storage space and the additional registers allowed software to be written more efficiently.

III.SYSTEM DESIGN

A. Hardware Design

The block diagram of the designed system is illustrated in figure 4.



B. System Specifications

The system specifications are illustrated in table 1.

TABLE I
SYSTEM SPECIFICATIONS

SL. NO	SPECIFICATIONS
1.	Domain: Microprocessors, Assembly language Programming
2.	Microprocessor: 8086 version 16-bit.
3.	Seven segment display type: Multiplexed seven segment display
4.	Programmable Peripheral interface: PPI 82C55
5.	Desktop computer : Dual core processor, 1GB RAM, processor speed 2.5 GHz.
6.	Software Assembler: MASM
7.	Applications: To display any suitable message on the multiplexed seven segment display.
8.	Power supply: External power supply +5V, 1A.

C. *Multiplexed seven segment display module overview*

The ALS-NIFC 38 is a multiplexed 6 digit display and is interfaced to the microprocessor and PCI I/O add-on card using an 8255 segment buffer and a digit decoder driver. The photographic view of the multiplexed seven segment display is shown in figure 5**.**

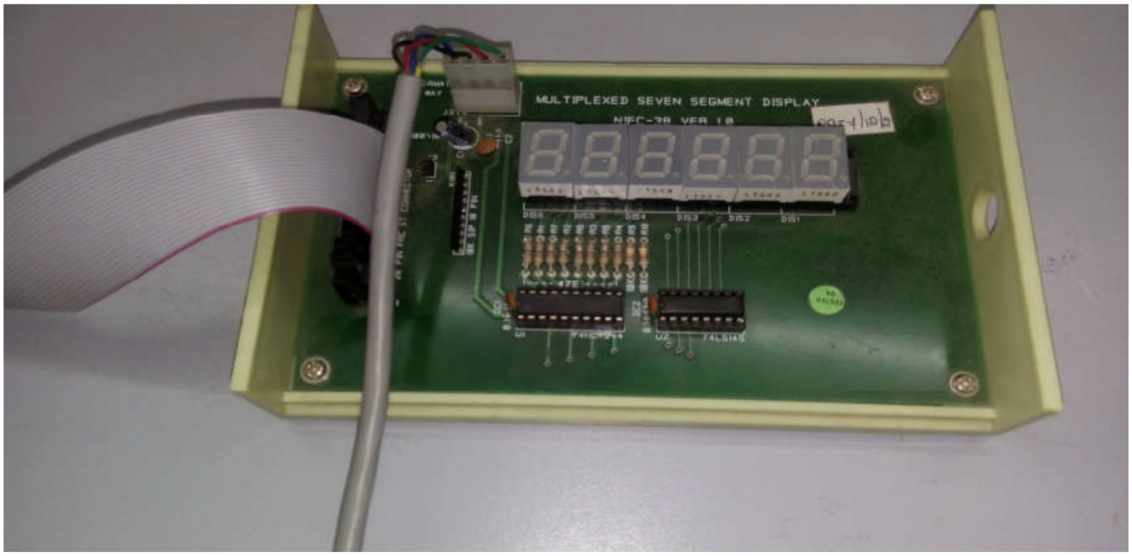


Fig. 5 ALS-NIFC multiplexed seven segment display module

Port A line of the 8255 is buffered and connected to the segments of the seven segment displays, as shown in the table 2.

TABLE III
PORT A CONFIGURATION

SL.NO	PORT-LINE	SEGMENT
1.	PA0	A
2.	PA1	B
3.	PA2	C
4.	PA3	D
5.	PA4	E
6.	PA5	F
7.	PA6	G
8.	PA7	DECIMAL POINT

Port lines PC2, PC1 and PC0 are connected to the inputs of a 74LS145 decoder driver, the outputs of which are connected to the common cathode of the six, Seven-segment displays. The port line combination and corresponding digits enabled are listed in table 3.

TABLE IIIII
PORT C CONFIGURATION

SL. NO	PC2	PC1	PC0	DIGIT
1.	0	0	0	D1
2.	0	0	1	D2
3.	0	1	0	D3
4.	0	1	1	D4
5.	1	0	0	D5
6.	1	0	1	D6

Based on the application the backup table is used for storing the seven segment code for each digit as illustrated in table 4. The seven segment code for each digit is output to port A while turning on the corresponding digit using port C lines. Normally for a 6-digit display, this would be done for about 2 msec. For the next 2 msec, the data for the next digit is sent to port A and the corresponding code to port C. this is repeated till all the digits are serviced, at this time the whole sequence is again restarted. Normally converted seven segment data is stored in connective locations in memory and the decimal point information is OR'ed or AND'ed before transferring to port A

CHARACTER	h	g	f	e	d	c	b	a	In Hex
C	0	0	1	1	1	0	0	1	39H
O	0	0	1	1	1	1	1	1	3FH
O	0	0	1	1	1	1	1	1	3FH
L	0	0	1	1	1	0	0	0	38H
D	0	1	0	1	1	1	1	0	5EH
U	0	0	1	1	1	1	1	0	3EH
D	0	1	0	1	1	1	1	0	5EH

E	0	1	1	1	1	0	0	1	79H

D. System set up

The experimental set up and its conduction was carried out in Microprocessors laboratory [10], [11], [12]. The system set up consisted of a desktop computer, external power supply and the ALS Multiplexed seven segment display Kit and a flat ribbon cable (FRC). The power supply connections were done as follows: the 5V and GND connections to the interface were provided using the 4-way power mate. The color codes of the 4 way power connector (power mate) on the interface are +5V (Blue), GND (Black). The 26 core flat ribbon cable from the PCI ADD-ON cards is connected to the 26 pin connector on the multiplexed seven segment display interface. The photographic view of the power supply and the complete system are illustrated in figures 6 and 7 respectively.

To display a word on a display unit Di (i= 0 to 5), the seven segment code of the character to be displayed on the unit is to be output through port A, while the display unit is selected by the combination of PC2 to PC0 bits of port C. This is depicted by the circuit schematic in fig 8.

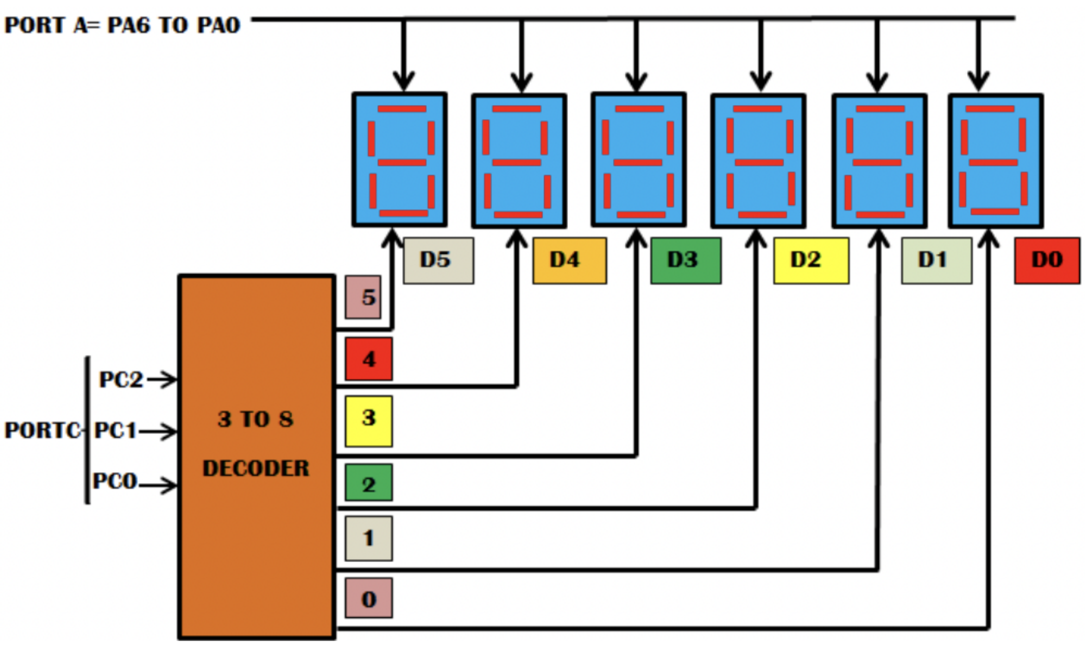


Fig. 8 Circuit schematic for displaying a Word

IV. RESULTS AND DISCUSSIONS

The desktop computer was switched ON and the start option was chosen by clicking. After this CMD (command prompt) was selected as a result of which a black colored window appeared on the screen. To enter the editor window for typing the program the following steps were done sequentially

i. cd\
ii. d:
iii. cd test
Iv. edit program nam.asm

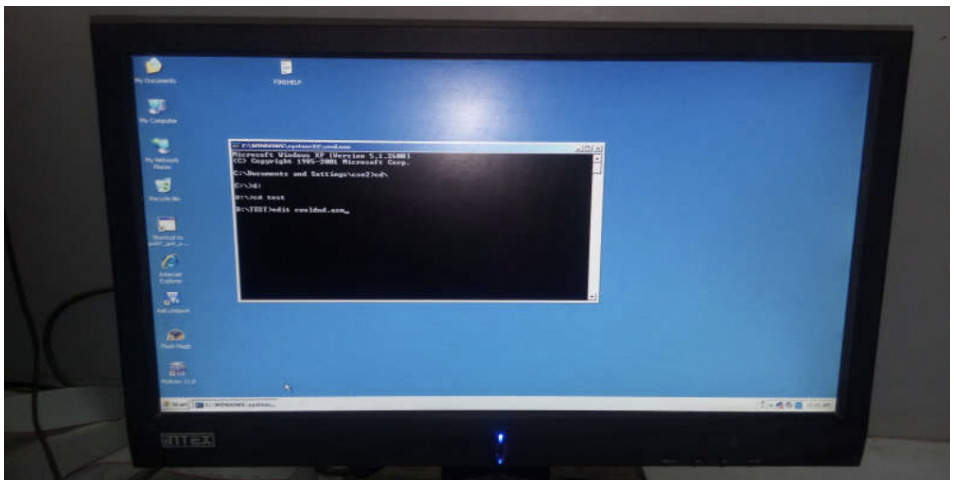


Fig. 9 Steps for editor window screen

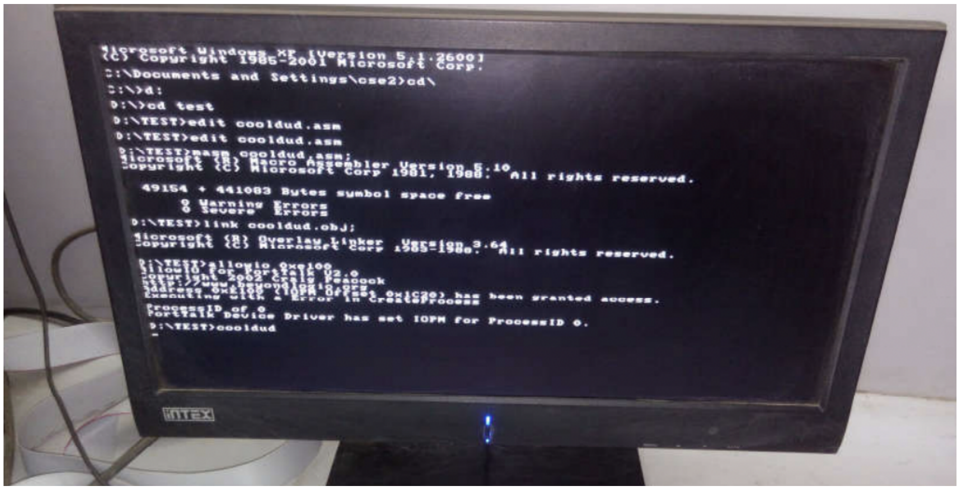
As soon as the above steps were carried out a blue colored editor window screen appeared. The program for the seven segment display interface for the display of messages Cool and dude was written, saved, compiled and executed. The output was obtained using the following execution steps

Step 1 D:\>TEST>MASM FILENAME.ASM;
Step 2 D:\>TEST>LINK FILENAME.OBJ;
Step 3 D:\>TEST>ALLOWIO 0XE100
Step 4 D :\> TEST>FILENAME

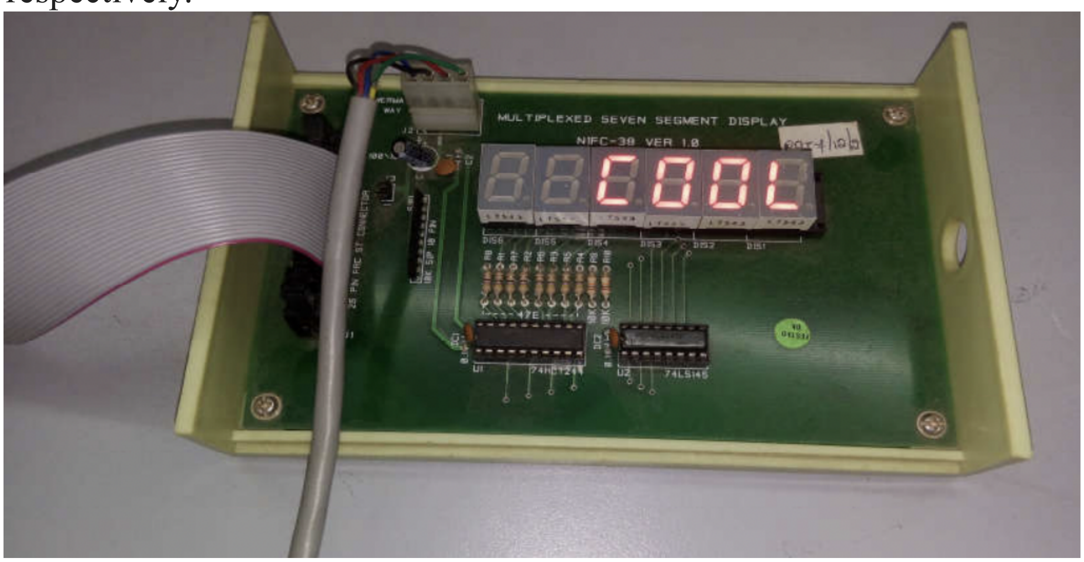
In the proposed system the file name was chosen as cooldud. Accordingly the above steps are

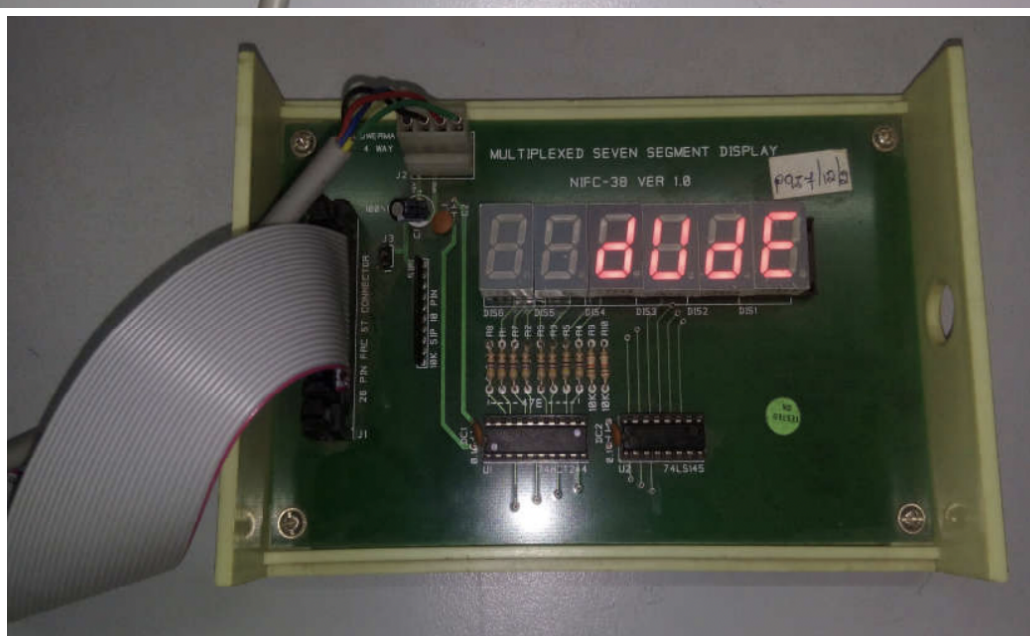
Step 1 d:\>TEST>masm cooldud.asm;
Step 2 d :\> TEST>link cooldud.obj;
Step 3 d :\> TEST>allowio 0xe100
Step 4 d :\> TEST>cooldud

All the above steps are depicted in figure 10.



As the last step was completed, the required output in the form of text messages "COOL" and "dUdE" were displayed alternately on the seven segment display interface for a suitable period of time as depicted by the photographic view in fig. 11 and 12 respectively.





V. CONCLUSIONS

The interfacing of a multiplexed seven segment display with the 8086 microprocessor is presented in this paper. The program was written in Assembly language for the display of text messages Cool and dude alternately with flickering on the multiplexed seven segment display interface module. A MASM assembler was used to assemble the program. Care was taken in properly making the required connections between the computer system and the multiplexed seven segment display interface module .The output obtained was very precise and clear. The entire system is very stable, cost effective and easy to design and use. To sum up it can also be concluded that Multiplexed displays are worth using compare to static displays or the non

multiplexed displays, owing to its many advantageous features. Furthermore, any desired text messages or digits can be displayed on the multiplexed seven segment displays by making suitable changes in the Assembly language program.