International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 7 Issue 2|| February 2018 || PP. 58-65

Interfacing A 3x8 Matrix Keyboard with the 8086 Microprocessor

Md. Moyeed Abrar
Department of Computer science & Engineering, Khaja Banda Nawaz College of Engineering, Kalaburagi,Visvesvaraya Technological University, Belagavi, Karnataka, India.
Corresponding Author: Md. Moyeed Abrar

ABSTRACT : *All keyboards are available in a variety of sizes. The standard keyboards are the 101 key QWERTY keyboards interfaced to the microprocessor. The small specialized keyboards are also available which contains only 4, 16 and 24 keys. Such small keyboards can be purchased, preassembled or may be constructed from individual key switches. .The state of the art presented in this paper is the Interfacing of a 3x8 Matrix keyboard with the 8086 Microprocessor. The Assembler used in the system is the MASM assembler. Microprocessor is an integrated circuit, which is the heart of the microprocessor based computer system. In other words a Microprocessor is a finite state automation machine that executes instructions held in a memory. Microprocessor has inputs, outputs and a process. The inputs and outputs of a microprocessor are a series of voltages that can be used to control external devices.*

KEYWORDS *-Assembler, external devices, interfacing, integrated circuit, keyboard, key switches, MASM, Microprocessors.*

Date of Submission: 25-01-2018 Date of acceptance: 17-02-2018

I. INTRODUCTION

The microprocessor sometimes referred to as the CPU (central processing unit), is the controlling element in a computer system. The CPU or processor acts as a controller of all actions or services provided by the system. The operations of the CPU can be reduced to three basic steps namely fetch, decode and execute. The microprocessor controls the memory and I/O (input/output) through a series of connections called buses. The buses select an I/O or memory device; transfer the data between I/O or memory and the microprocessor and controls the I/O and memory system. Memory and I/O are controlled through instructions that are stored in the memory and executed by the microprocessor [1].

The microprocessor performs three main tasks for the computer system. Firstly, it transfers the data between itself and the memory or I/O systems. Secondly, it performs simple arithmetic and logic operations. Lastly it helps program flow via simple decisions. The power of the microprocessor is its ability to execute billions of millions of instructions per second from a program or software (group of instructions) stored in the memory system. This concept of stored program has made the microprocessor and computer system very powerful devices [1], [2].

The8086 is a microprocessor with 16-bit data bus internally and externally, which means all registers are 16 bits wide and there is a 16-bit data bus to transfer data in and out of the CPU.

In this paper the interfacing of 3x8 Matrix keyboard with the 8086 microprocessor is presented. The rest of the paper is organized into sections as follows: section II illustrates the overview of 8086 Microprocessor. Section III focuses on the system design. Section IV includes the software description of the system. Results and discussion are reported in section V. Finally section VI summarizes the paper and presents the concluding remark.

II. 8086 MICROPROCESSOR OVERVIEW

The Intel 8086 is a 16 bit microprocessor that is intended to be used as a CPU in a microcomputer. 16 bit indicates the word length of the microprocessor. It is also referred to as the data size of the microprocessor. A bus is a group of wires organized to provide a means of communication among different elements in a Microprocessor system. There are three types of buses namely Address bus, Data bus and control bus. All these buses are effectively used by the microprocessor for performing various tasks.

2.1 Important features of 8086 Microprocessor

- The 8086 microprocessor has a 16 bit data bus hence its ALU and internal registers are able to process 16 bit information at a time.
- It has 20 address lines, so it can address 2^{20} = 1MB memory locations.

- It has multiplexed address and data bus which reduces the number of pins needed.
- It has 16 control lines for providing handshaking signals during bus transfer and for permitting at least some external control of the CPU.
- It requires only one +5 Volt supply voltage.
- It supports both multiprogramming and multiprocessing.
- Its instruction stream byte queue speeds up its execution of instruction.
- It operates in two modes namely Minimum mode and Maximum mode. In minimum mode of operation 8086 microprocessor works only as the CPU of the system whereas in the maximum mode of operation issue of I/O peripherals and memory is included.

III. SYSTEM DESIGN

3.1 Hardware Design

The block diagram of the system is depicted in fig.1

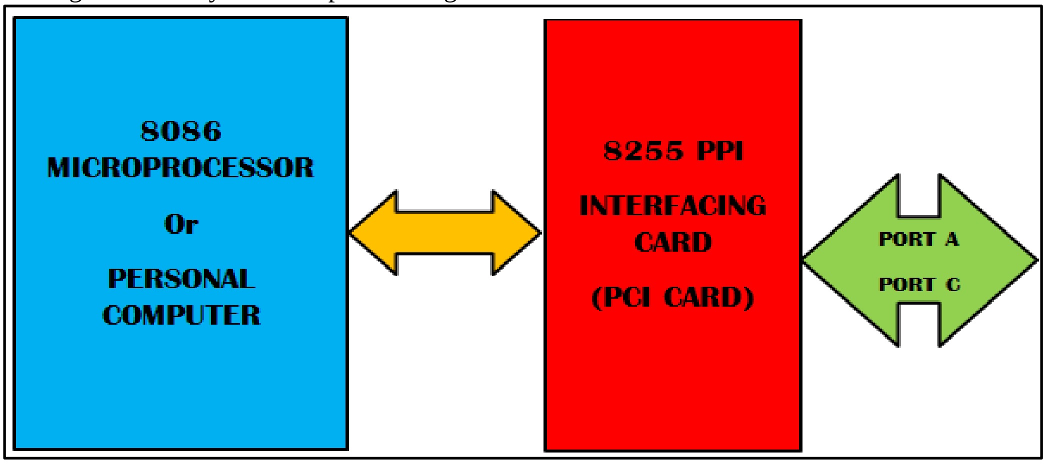

Fig.1 System block diagram

3.2 System Specifications

The System specifications are illustrated in TABLE 1.

TABLE 1 System specifications

SL.NO	SPECIFICATIONS
1.	Domain: Microprocessors, Assembly language Programming
2.	Microprocessor: 8086 version 16-bit.
3.	Keyboard: 3x8 Matrix Keyboard (24 keys)
4.	Programmable Peripheral interface: PPI 82C55
5.	Desktop computer : Dual core processor, 1GB RAM, processor speed 2.5 GHz.
6.	Software Assembler: MASM
7.	Interfacing device: NIFC-39
8.	Applications: To display the key pressed on the screen as well as to display the row and column numbers of the key pressed.

3.3 Keyboard Interface module overview

The keyboard interface module is equipped with 24 keys which are placed on a single euro card size (160x100mm) glass epoxy PCB. The keyboard possesses 24 keys which are having calculator type keycaps. The keys are organized as three rows of eight keys each in a sense of three rows and eight columns. Hence this keyboard is also referred to as 3x8 matrix keyboard. The keyboard consists of numbers 0,1,2,3,4,5,6,7,8,9, special characters +, -, x, /, =, %, C, CE and five blank keys. The configuration of the keyboard module is illustrated in fig2 and the photographic view is illustrated in fig 3.

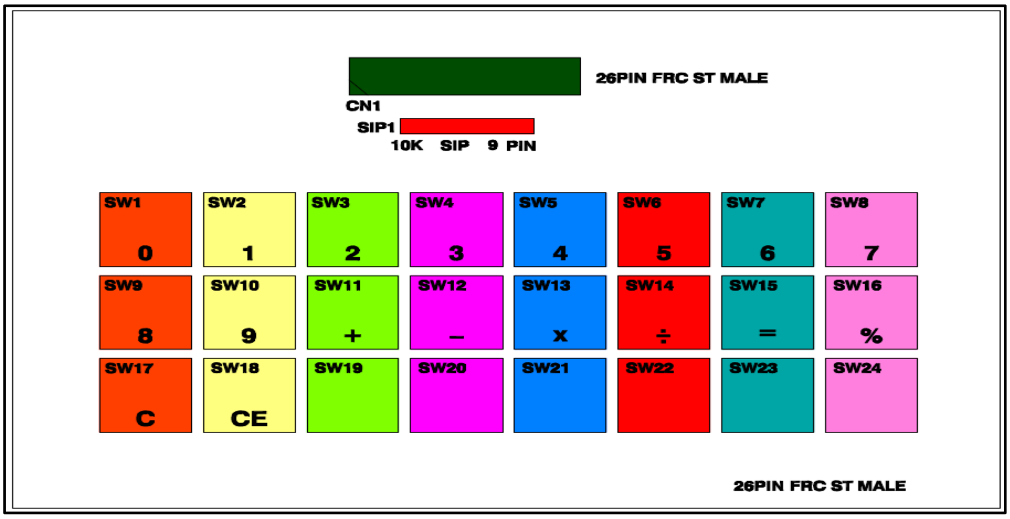

Fig.2 configuration of the keyboard module

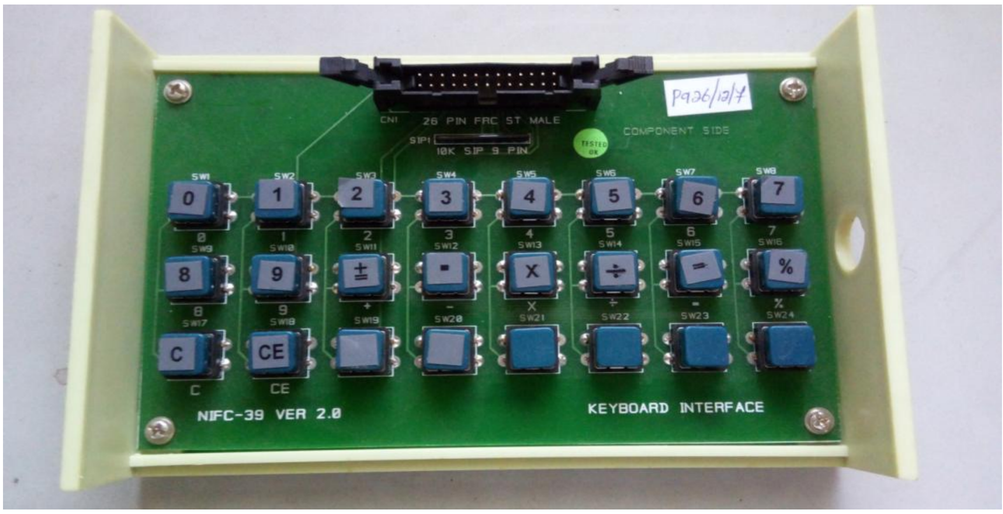

Fig. 3 Photographic view of the keyboard module.

The three rows (scan lines) are connected to bits 0, 1 and 2 of port C which are represented as PC1, PC2 and PC3 while the eight columns are connected to bits 0 to 7 of port A of the Programmable peripheral interface 8255 represented as PA0 to PA7. The complete interfacing is depicted in fig 3.

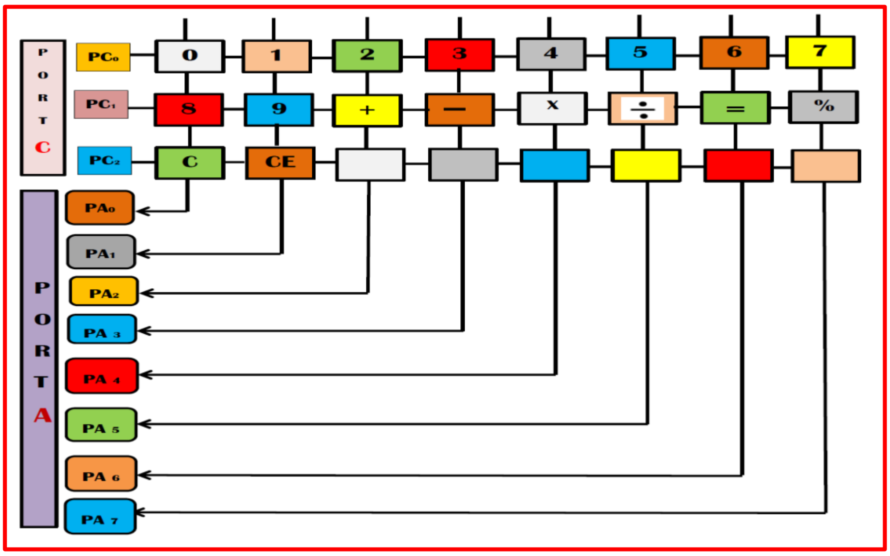

Fig.4 Complete interfacing of the system

The internal circuit schematic for the interface is shown in fig 5

3.4 Programmable Peripheral interface

All the 82C55 Programmable Peripheral Interface (PPI) is a very popular, low cost interfacing component used in a various applications. The PPI has 24 pins for I/O which can be programmed in groups of 12 pins. It has groups which can operate in three distinct modes of operation. The 82C55 possess the capacity to interface any TTL-compatible I/O devices to the microprocessor. The 82C55 PPI (CMOS) version needs the insertion of wait states when operated with a microprocessor using higher than an 8 MHz clock. It also provides at least 2.5mA of sink current (logic 0) at each output, with a maximum of 4 mA. As I/O devices are inherently slow, wait states that are used during I/O transfers does not have a significant impact on the speed of the system. The 82C55 still finds application even in the latest core 2 based computer system. The modern computers uses a few 82C55s located inside the chip set for various features on the personal computer. The 82C55 is used for interfacing to the keyboard as well as to the parallel printer port in many personal computers, but it is found as a function within an interfacing chip set. Moreover, the chip also controls the timer and reads data from the keyboard interface.

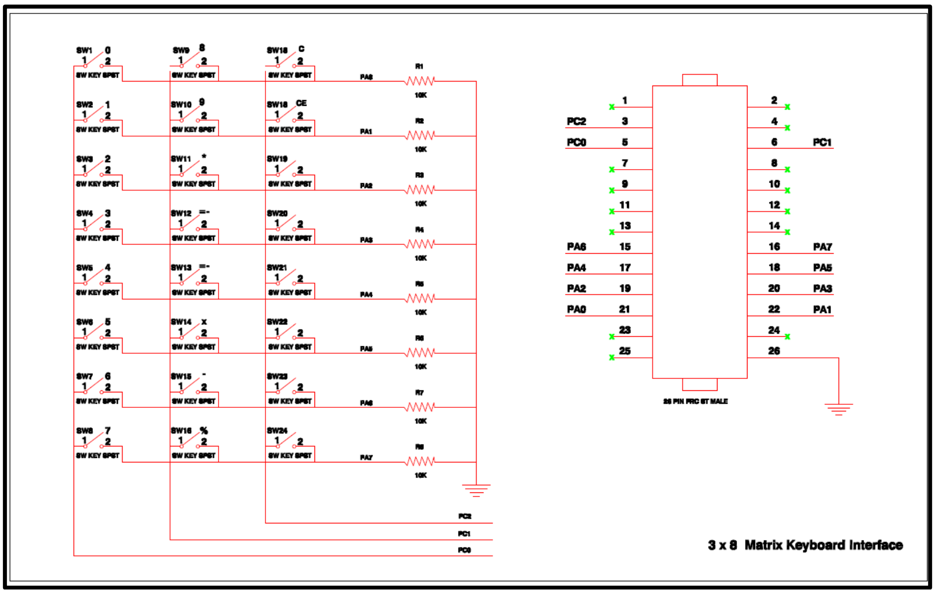

Fig.5 Circuit schematic for the interface.

3.5 Key Debounce concept

Whenever a mechanical push-button is pressed or released once, the mechanical components of the key do not change the position smoothly, rather it generates a transient response. These transient variations may be interpreted as the multiple key presses and responds accordingly by the microprocessor system. To overcome this problem two different schemes are suggested. The first one utilizes a bi-stable multivibrator at the output of the key to debounce and the second one suggests that the microprocessor should be made to wait for the transient period (usually 10ms or 20ms) so that the transient response settles down and reaches a steady state. Logic 0 will be read by the microprocessor when the key is pressed. [2]

IV. SOFTWARE DESCRIPTION

4.1 MASM Assembler overview

The microprocessor needs an Assembler program which generates machine language, because machine language instructions are too complex to efficiently generate by hand. The Assemblers which are commonly used are Microsoft MACRO assembler (MASM) assembler, Intel Assembler (ASM), Borland Turbo assembler (TASM). The MASM assembler is recommended to be used as a development tool and the Intel Assembler (ASM) and Borland Turbo assembler (TASM) functions equally as well. The most recent versions of TASM assembler completely emulate the MASM assembler. The proposed system uses MASM assembler to perform the task of converting the Assembly language program into machine language program.

The microprocessor uses the MASM assembler in two ways.

1. With models that are unique to a particular assembler.

2. with full segment definitions that allow complete control over the assembly process and are universal to all the assemblers. [1]

In the proposed system, the Assembly language program for the interfacing of a 3x8 keyboard is written using Full segment definition.

4.2 Assemble, Link and Run a Program

There are three steps to create an executable Assembly language program. These steps are illustrated in table 3.

TABLE 3 Steps to create executable Assembly language programming.

SL. NO	STEP	INPUT	PROGRAM	OUTPUT
1.	Edit the program	Keyboard	Editor	Myfile.asm
2.	Assemble the program	Myfile.asm	MASM or TASM	Myfile.obj
3.	Link the program	Myfile.obj	LINK or TLINK	Myfile.exe

The MASM and LINK programs are the assembler and linker programs for the Microsoft's MASM assembler. Many excellent editors or word processors are available that can be used to create and/or edit the program. The editor must be able to produce an ASCII file. Although filenames follow the usual operating system conventions, the source file must end in ".asm" for the assembler. This ".asm" source file is assembled by an assembler, such as the Microsoft's MASM or Borland's TASM. The assembler will produce an object file and a list file, along with other files that may be useful to the programmer. The extension for the object file must be ".obj". This object file is the input to the LINK program, which produces the executable program that ends in ".exe". The ".exe" file can be executed by the microprocessor. Before feeding the ".obj" file into LINK, all syntax errors produced by the assembler must be corrected. Figure 6 shows how an executable program is created by the steps outlined above. [2]

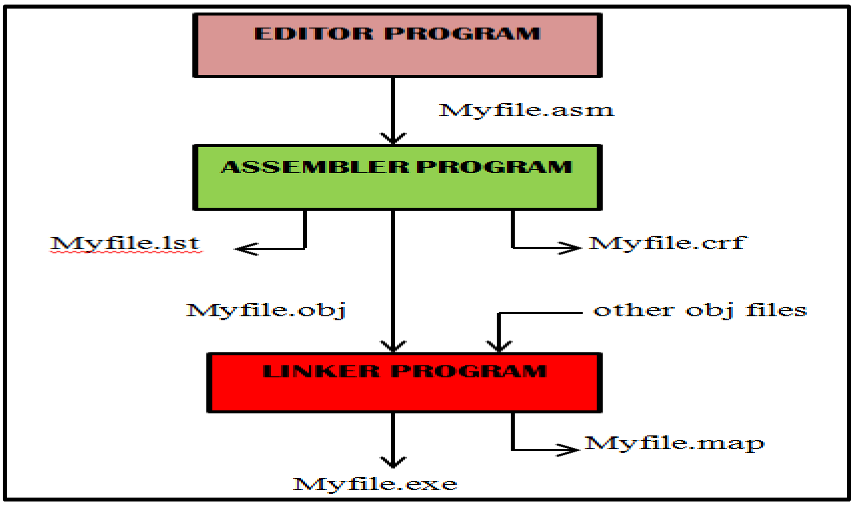

Fig.6 creation of executable Assembly language program

4.3 Software program description

The Assembly language program (ALP) written for the keyboard interface with the microprocessor illustrates the manner in which Keyboard is scanned, detects the key closure, debouncing of the key, and displaying the corresponding hex code in the display. The program is written in such a manner that it will scan the keyboard row by row and checks whether any key is closed. If a key closure is found it will find the code of the closed key and displays it on the monitor screen. The program makes use of two key lockouts. If two keys are pressed simultaneously, key code corresponding to the key released last is displayed. The key code to each key is shown in table 2.

TABLE 2 Key code for each key

SL. NO	KEY	CODE
1.	0	00
2.	1	01
3.	2	02
4.	3	03
5.	4	04
6.	5	05
7.	6	06
8.	7	07
9.	8	08

10.	9	09
11.	+	0A
12.	-	0B
13.	X	0C
14.	÷	0D
15.	=	0E
16.	●	0F
17.	%	10
18.	C	11
19.	CE	12
20.	BLANK	13
21.	BLANK	14
22.	BLANK	15
23.	BLANK	16
24	BLANK	17

The flowchart for the system is illustrated in figure 7.

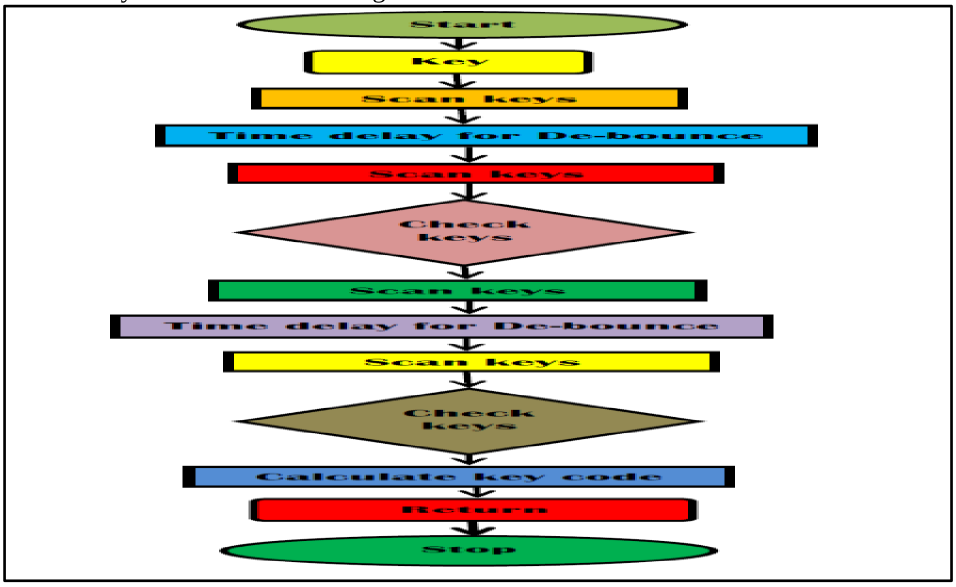

Fig.7 Flowchart for the designed system

V. Experimental Results And Discussion

The experimental set up and its conduction was done in Microprocessors laboratory[3]. The system consists of a Desktop computer, 3x8 matrix keyboard interface module and interfacing PPI card.The photographic view of the system is shown in figure 8.

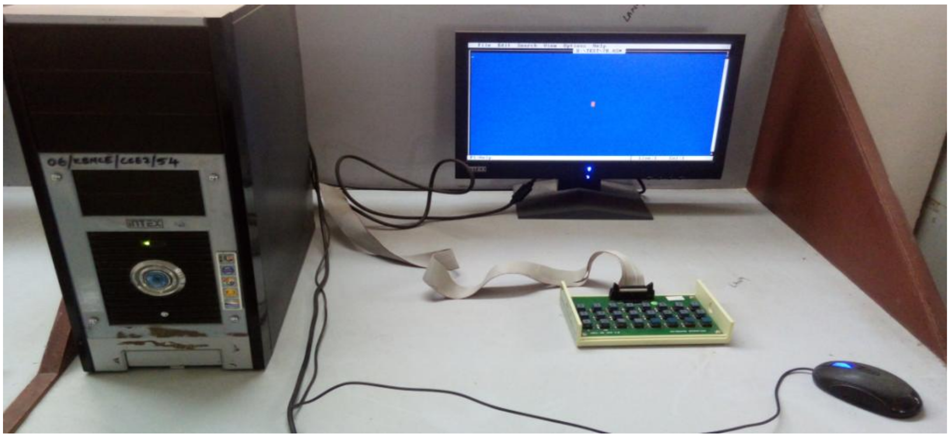

Fig.8 photographic view of the designed system

The desktop computer was turned ON and start option was clicked and CMD (command prompt) was selected. In order to enter the editor window for writing the program the following steps were performed.

i. D:
ii. CD TEST
iii. PROGRAM NAME.ASM

The Blue color editor window screen appeared. The required program was written, saved, compiled and executed. The execution steps for getting the output are as follows
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;
As soon as the last step is performed, enter any key option appeared in the output window. The 0 key on the 3x8 matrix keyboard was pressed. The column and row numbers were displayed as COL=1, ROW=1. Once again step 4 was repeated. Again Enter any key option appeared in the output window. This time the + sign on the 3x8 matrix keyboard was pressed. The column and row numbers were displayed as COL=3, ROW=2.
The photographic view of the output is as shown in figure in 9.

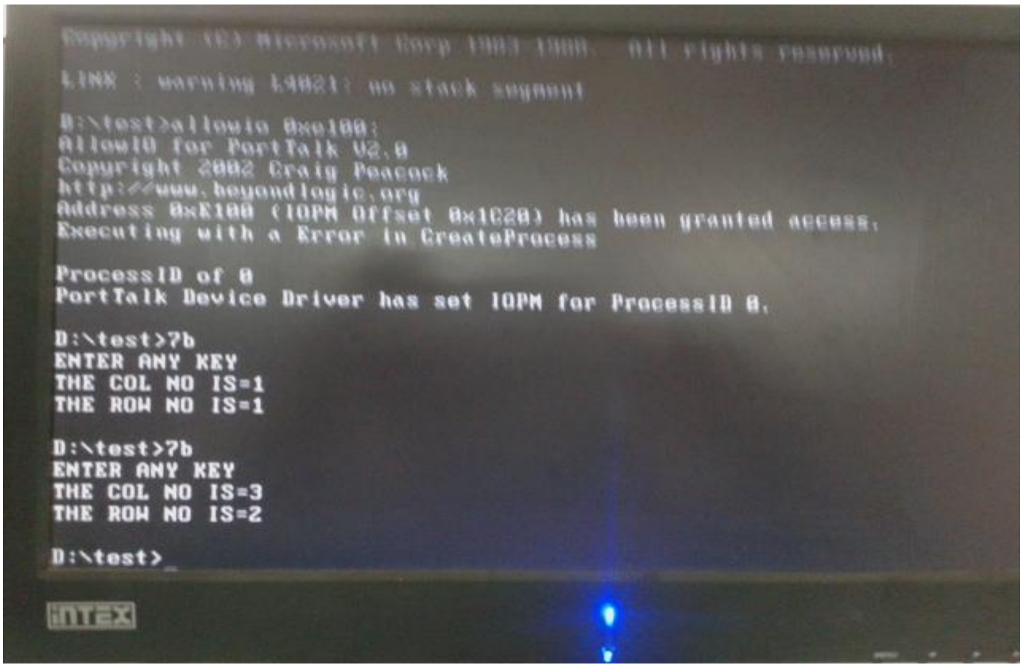

Fig.9Photographic view of the output obtained.

VI. CONCLUSION

In this paper the interfacing of a 3x8 matrix keyboard with the 16-bit microprocessor 8086 is presented. The program was written in such a way that it scans a 3x8 keypad for key closure and displays the key pressed on the screen as well as displays the row and column numbers of the key pressed. The keyboard interface module with many important features is easy for the programmer to install and use and is user friendly. Care was taken in properly connecting the interfacing card to the 26 pin male connector to 3x8 matrix keyboard interfacing module. The output obtained was very clear and precise. The entire system is very stable, reliable and cost effective.

Acknowledgements

First of all I would like to thank Almighty Allah by the grace of whom I reached the stage of completion of this work. This avenue has been a turning point in my career to mold me into a thorough and dynamic Professional. My sincere thanks to the Principal Dr. S Kamal Mohd. Azam, Vice Principal Dr. Ruksar Fatima and Dr. Asma Parveen H.O.D Computer Science and Engineering department of my esteemed institution for their inspiration and support. Lastly I am also thankful to my beloved Parents who have helped me pave this path to success.

REFERENCES

[1]. Barry B. Brey, Introduction to the Microprocessor and Computer,*The Intel Microprocessors – Architecture, Programming and Interfacing*, (India: Dorling Kindersley Pvt.ltd, Pearson-Prentice Hall Eighth Edition 2009) 25-29.

[2]. Muhammad Ali Mazidi, Janice Gillespie Mazidi and Danny Causey, Keyboard and Mouse Programming, *The x86 PC Assembly Language, Design and Interfacing*,(India: Dorling Kindersley Pvt.ltd, fifth edition 2011) 161-165.

[3]. SaskoRistov, NevenaAckovska, VesnaKirandziska, DarkoMartinovikj, The Significant progress of the Microprocessors and Microcontrollers coursse for Computer Science Students, *37th International Convention on Information and CommunicationTechnology, Electronics and Microelectronics (MIPRO), IEEE*,2014,Opatija, Croatia, 818-823.

[4]. NurIdawati, Md. Enzai, Norhayati Ahmad, Syazilawati Mohamed, Siti Sara Rais, Mohd Amir Hamzah Ab. Ghani, Nuraiza Ismail, CDIO implementation in microprocessor course through mini project assignment, *9th IEEE International Conference on Engineering Education (ICEED)*, Kanazawa, Japan, 2018, 7-11.

[5]. Jorge A. Martinez, Juan Antonio Maestro, Pedro Reviriego, Evaluating the impact of the instruction set on Microprocessor Reliability to Soft Errors, *IEEE transactions on Device and Materials Reliability, PP(99)*, 2018, 1-1.

[6]. D.L Smith, Exercises, Projects, and Solutions for a Microprocessor electronic interfacing laboratory course sequence, *proceedings of the 26th IEEE Annual conference, Frontiers in Education conference*, Salt lake city, UT, USA, 1996.

[7]. Lynne A. Slivovsky, Albert A. Liddicoat, Work in progress: future Pedagogical Trends in the Microprocessor Course- The Soft Core Processor, *Proc. 36th Annual IEEE conference, Frontiers in Education*, San Diego, CA, USA, 2006, 1-2.

[8]. Scott Mc Mahon, New and future 29K family product Microprocessors and Microcontrollers, *Digest of papers COMPCON spring, IEEE conferences*, San Francisco, CA, USA, 1992, 350-353.

[9]. Doniursutiu; Victor Nascov; Cornel Samoila; Marius Moga, Microcontroller technologies in low power applications, *15th IEEE International conference on interactive collaborative learning(ICL)*, Villach, Austria, 2012, 1-5.

[10]. Chris Herring, Microprocessors, Microcontrollers and Systems in the New Millennium, *IEEE Micro, vol.20, issue 6*, 2000, 45-51.

International Journal of Engineering Science Invention (IJESI) is UGC approved Journal with Sl. No. 3822, Journal no. 43302.

Md. Moyeed Abrar "Interfacing A 3x8 Matrix Keyboard With The 8086 Microprocessor" International Journal of Engineering Science Invention (IJESI), vol. 07, no. 02, 2018, pp. 58–65.