A

PROJECT REPORT

On

KEYPAD INTERFACED GAME DESIGN USING 8051

Submitted in partial fulfilment of the

Requirements for the award of the Certification

By

APSSDC

In

Electronics & Communications Engineering

By

Ch. A V Manikanta kumar ***B. Guna Sai Vikas*** ***N. Kalpanath Reddy***

13004429 **13004541** **13004550**

E. MohanKrishna ***Ch. Prem Abhinav***

13004545 **13004467**

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS ENGINEERING

K L University

Green Fields, Vaddeswaram, Guntur District-522 502

2015-2016

K. L. UNIVERSITY

CERTIFICATE

This is to certify that the project entitled "**Keypad Interfaced Game Design using 8051"** is the bonafide work carried out by CH A V Manikanta Kumar (13004429), B. Guna Sai Vikas (13004541), N. Kalpanath Reddy (13004550), E. Mohan Krishna (13004545), Ch. Prem Abhinav (13004467) students of III year B-Tech, ECE department, College of Engineering, K. L. University, in the APPSDC Project for the academic year 2015-2016.

Signature of the Project guide

Head of the department

ACKNOWLEDGMENT

Our sincere thanks to **DR. A. S. C. S. SASTRY** for his outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **DR. A. S. C. S. SASTRY,** Head of the Department for Electronics and Communication Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project based lab work.

We would like to place on record the deep sense of gratitude to the honorable Vice Chancellor, K L University for providing the necessary facilities to carry the concluded project based lab work.

Last but not the least, we thank all Teaching and Non-Teaching Staff of our department and especially my classmates and my friends for their support in the completion of our Project based Report.

Place: K L University

Date: 16-05-2016

S.No	Name of the Students
1	CH A V Manikanta Kumar (13004429)
2	B. Guna Sai Vikas (13004541)
3	N. Kalpanath Reddy (13004550)
4.	E. Mohan krishna (13004545)
5.	Ch. Prem Abhinav (13004467)

K L UNIVERSITY

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS ENGINEERING

DECLARATION

We hereby declare that this project based lab report entitled **“Keypad Interfaced Game Design using 8051”** has been prepared by us in partial fulfilment of the requirement for the award of **“CERTIFICATION BY APSSDC”** during the academic year 2015-2016.

I also declare that this project based lab report is of our own effort and it has not been submitted to any other university for the award of any degree.

Date: 16 – 05 – 2016

Place: Vaddeswaram

S.No	Name of the Students
1	CH A V Manikanta Kumar (13004429)
2	B. Guna Sai Vikas (13004541)
3	N. Kalpanath Reddy (13004550)
4.	E. Mohan krishna (13004545)
5.	Ch. Prem Abhinav (13004467)

CONTENTS

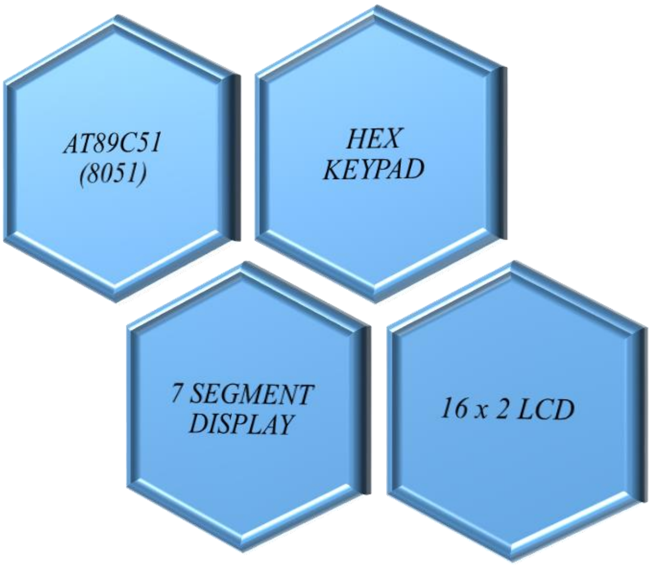
ABSTRACT

Games are mostly fascinating to play and there exists much engineering thoughts behind every logic in the game. The present day world deals with the games which are very precise in results in terms of speed, timing and placements. With this project we come with a new game where a person's timing and observational skills are tested and awarded with a score. In this project we use 8051 microcontroller which is interfaced with 7 segment display, a 7 x 16 LCD, and a Hex Keypad. The microcontroller generates some random number on the 7 segment. The conditions of the game is that we have to press the same number that is displayed in a stipulated amount of time say 1 second and send the same number using external keypad to 8051 before it displays the next random number. So if we fail to enter the number in the given time the game ends by displaying your result on the LCD. The delay time will be initiated with certain clock frequency and the frequency goes on increasing dynamically with increase in your score thus enabling more toughness in the game. Your result is the time that you successfully played the game. Your high score will always be updated in one of the registers. The project will be simulated in Protues.

CHAPTER-1

INTRODUCTION

Games are mostly fascinating to play and there exists much engineering thoughts behind every logic in the game. The present day world deals with the games which are very precise in results in terms of speed, timing and placements. With this project we come with a new game where a person’s timing and observational skills are tested and awarded with a score. In this project we use 8051 microcontroller which is interfaced with 7 segment display, a 7 x 16 LCD, and a Hex Keypad. The microcontroller generates some random number on the 7 segment. The conditions of the game is that we have to press the same number that is displayed in a stipulated amount of time say 1 second and send the same number using external keypad to 8051 before it displays the next random number. So if we fail to enter the number in the given time the game ends by displaying your result on the LCD. The delay time will be initiated with certain clock frequency and the frequency goes on increasing dynamically with increase in your score thus enabling more toughness in the game. Your result is the time that you successfully played the game. Your high score will always be updated in one of the registers. The project will be simulated in Protues.



CHAPTER-2

BLOCK DIAGRAM AND WORKING PRINCIPLE

2.1 BLOCK DIAGRAM:

In this we used 8051 microcontroller, 16x2 LCD display,7-segment display and Hex key pad. The number generated by AT89C51 (8051 Microcontroller) will be displayed on 7 segment display (common cathode type) and the same number is resent to 8051 through Hex keypad in the given time. The overall score and the high score will be displayed by the LCD (2 X 16 Matrix type).

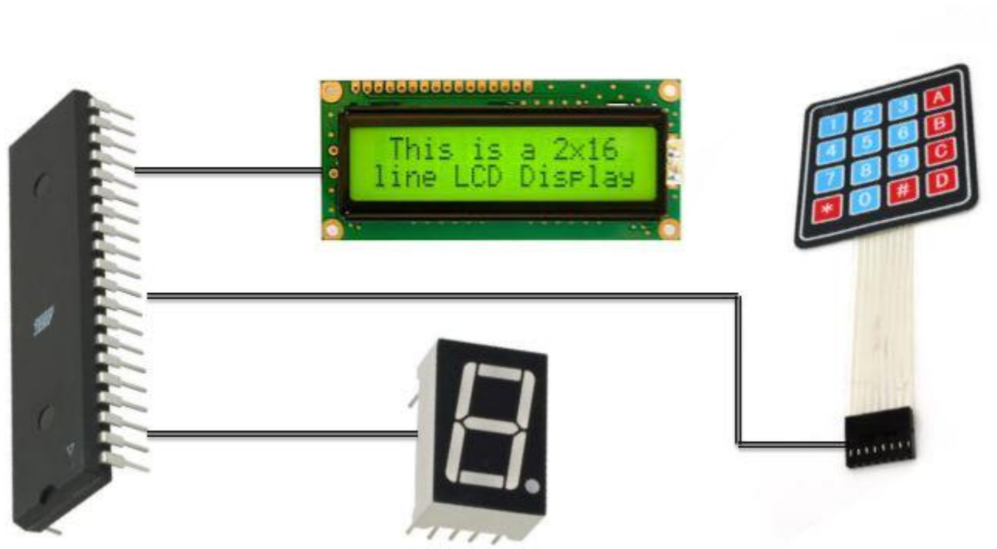


Fig 2.1: Block Diagram

2.2 WORKING PRINCIPLE:

There are 6 working modules in this project, they are as follows;

- Generation of Random Number between 0-9.
- Printing the Random number on 7 segment display.
- Entering the same number using Hex keypad.
- Comparing if the pressed number and generated number and also checking if the number pressed is before the stipulated time.
- Increasing the speed of timer dynamically with increase in score.
- Finally displaying the game results on LCD.

❖ GENERATION OF RANDOM NUMBERS 0 – 9:

- Theoretically there are 2 types of random numbers, they are:
- **True Random Numbers:** These are ideal random number sequences. The example is the noise in air which always changes randomly and give INFINITE random numbers. E.x.1,8,6,15,646,546468,0,154,1998,184,9,02,5991,942,564 etc.
- **Pseudo Random Numbers:** These are practically known as Deterministic random numbers are generally generated by computers. The major disadvantage they only generates N random numbers after which it again repeats the sequence.
- E.x.1,15,24,0,12,1,15,24,0,12, 1,15,24,0,12, 1,15,24,0,12, 1,15,24,0,12, etc.
- In this Project we generate Pseudo random numbers using any algorithms like LRG etc.
- But in this project we can write a simple c code to implement a mere true random number pattern by the logic of Moudulus operations.

Random Number (0-9) = (All set of Random Numbers) % 10

CHAPTER-3

HARDWARE MODULES

3.1 AT89C51 MICROCONTROLLER:

AT89C51 is an 8-bit microcontroller and belongs to Atmel's 8051 family. **ATMEL 89C51** has 4KB of Flash programmable and erasable read only memory (PEROM) and 128 bytes of RAM. It can be erased and program to a maximum of 1000 times.

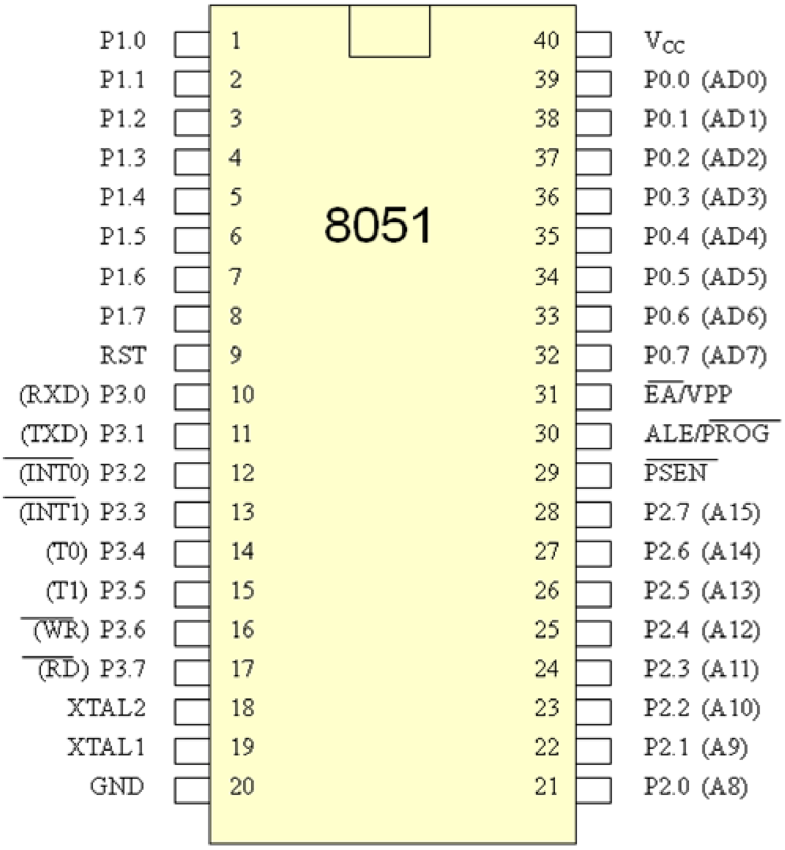


Figure.3.1: AT89C51 Microcontroller

In 40 pin AT89C51, there are four ports designated as P_1, P_2, P_3 and P_0. All these ports are 8-bit bi-directional ports, *i.e.*, they can be used as both input and output ports. Except P_0 which needs external pull-ups, rest of the ports have internal pull-ups. When 1s are written to these port pins, they are pulled high by the internal pull-ups and can be used as inputs. These ports are also bit addressable and so their bits can also be accessed individually. Port P_0 and P_2 are also used to provide low byte and high byte addresses, respectively, when connected to an external memory. Port 3 has multiplexed pins for special functions like serial communication, hardware interrupts, timer inputs and read/write operation from external memory. It can be programmed to operate at different baud rates. Including two timers & hardware interrupts, it has a total of six interrupts. AT89C51 is an 8-bit microcontroller and belongs to Atmel's 8051 family. ATMEL 89C51 has 4KB of Flash programmable and erasable read only memory (PEROM) and 128 bytes of RAM. It can be erased and program to a maximum of 1000 times. In 40 pin AT89C51, there are four ports designated as P1, P2, P3 and P0. All these ports are 8-bit bi-directional ports, i.e., they can be used as both input and output ports. Except P0 which needs external pull-ups, rest of the ports have internal pull-ups. When 1s are written to these port pins, they are pulled high by the internal pull-ups and can be used as inputs. These ports are also bit addressable and so their bits can also be accessed individually.

- Port P0 and P2 are also used to provide low byte and high byte addresses, respectively, when connected to an external memory. Port 3 has multiplexed pins for special functions like serial communication, hardware interrupts, timer inputs and read/write operation from external memory. AT89C51 has an inbuilt UART for serial communication. It can be programmed to operate at different baud rates. Including two timers & hardware interrupts, it has a total of six interrupts. Compatible with MCS-51 Products and has the following features.
- 4 Kbytes of In-System Reprogrammable Flash Memory. Endurance 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

3.2 16x2 LCD DISPLAY:

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs.

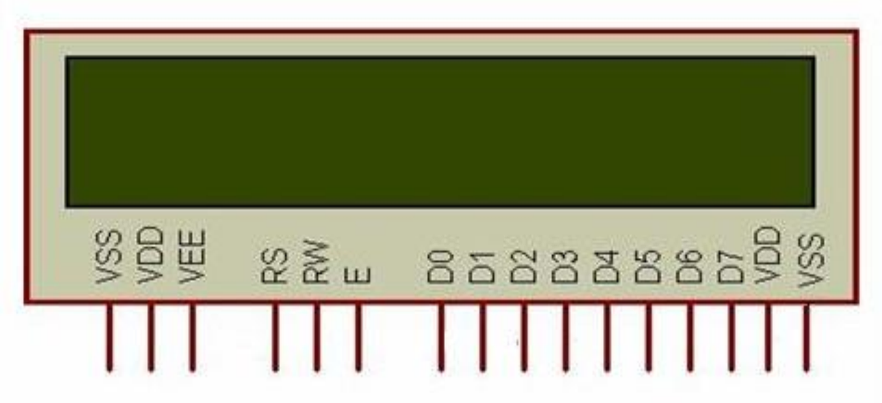


Figure.3.2: 16x2 LCD Display

The reasons being:

1. LCDs are economical, easily programmable, have no limitation of displaying in seven segment.
2. A 16x2 LCD means it can display 16 characters per line and there are 2 lines.
3. In this, each character is displayed in 5x7 pixel matrix. This LCD has 2 registers namely data and command.
4. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task.
5. The data register stores the data to be displayed on the LCD. The data is ASCII value of character to be displayed.
6. We can display messages, numbers, etc… on the LCD with simple commands.
7. The pin numbers, their names and corresponding functions are shown in Table.3.1

Pin N0:	Name	Function
1	VSS	This pin must be connected to the ground
2	VCC	Positive supply voltage pin (5V DC)
3	VEE	Contrast adjustment
4	RS	Register selection
5	R/W	Red or write
6	E	Enable
7	D0	Data
8	D1	Data
9	D2	Data
10	D3	Data
11	D4	Data
12	D5	Data
13	D6	Data
14	D7	Data
15	LED+	Back light LED+
16	LED-	Back light LED-

Table.3.2.1: LCD Pin description

3.2.1 16×2 LCD DISPLAY COMMAND CODES:

16×2 LCD module has a set of pre-set command instructions. Each command will make the module to do a particular task. The commonly used commands and their function are given in the table below.

Command	Function
1	Clear display screen
2	Return home
4	Decrement cursor
6	Increment cursor
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking

Command	Function
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift entire display left
1C	Shift entire display right
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	Two lines and 5x7 matrix

Table.3.2.2: LCD Commands

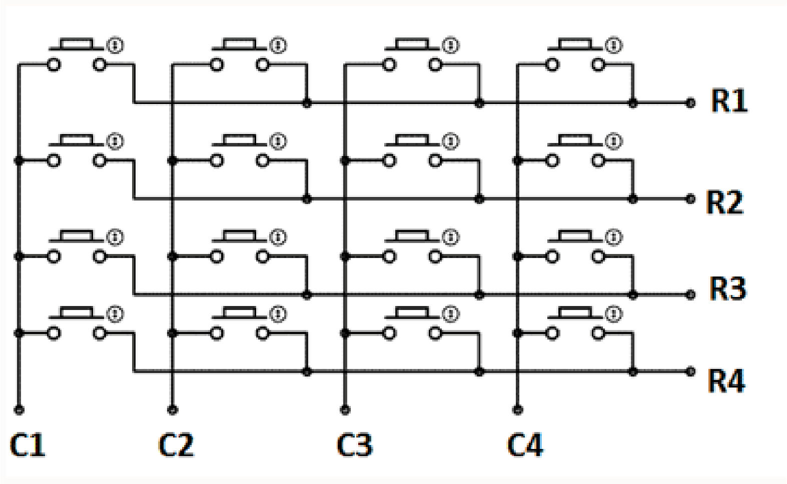
3.3 7-Segment Display:

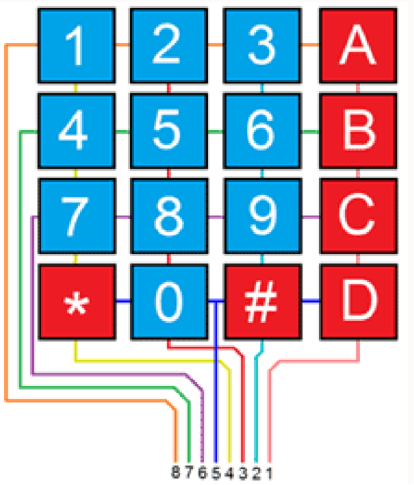
The seven elements of the display can be lit in different combinations to represent the a*rabic numerals*. Often the seven segments are arranged in an *oblique* (slanted) arrangement, which aids readability. In most applications, the seven segments are of nearly uniform shape and size (usually elongated hexagons, though trapezoids and rectangles can also be used), though in the case of adding machines, the vertical segments are longer and more oddly shaped at the ends in an effort to further enhance readability. The numerals 6, 7 and 9 may be represented by two or more different glyphs on seven-segment displays, with or without a 'tail'. The seven segments are arranged as a rectangle of two vertical segments on each side with one horizontal segment on the top, middle, and bottom. Additionally, the seventh segment bisects the rectangle horizontally. There are also fourteen-segment displays and sixteen-segment displays (for full alphanumerics); however, these have mostly been replaced by dot matrix displays. The segments of a 7-segment display are referred to by the letters A to G, where the optional decimal point (an "eighth segment", referred to as DP) is used for the display of non-integer numbers.

Fig:3.3.1 7-Segment Display

3.4 Hex Key Pad:

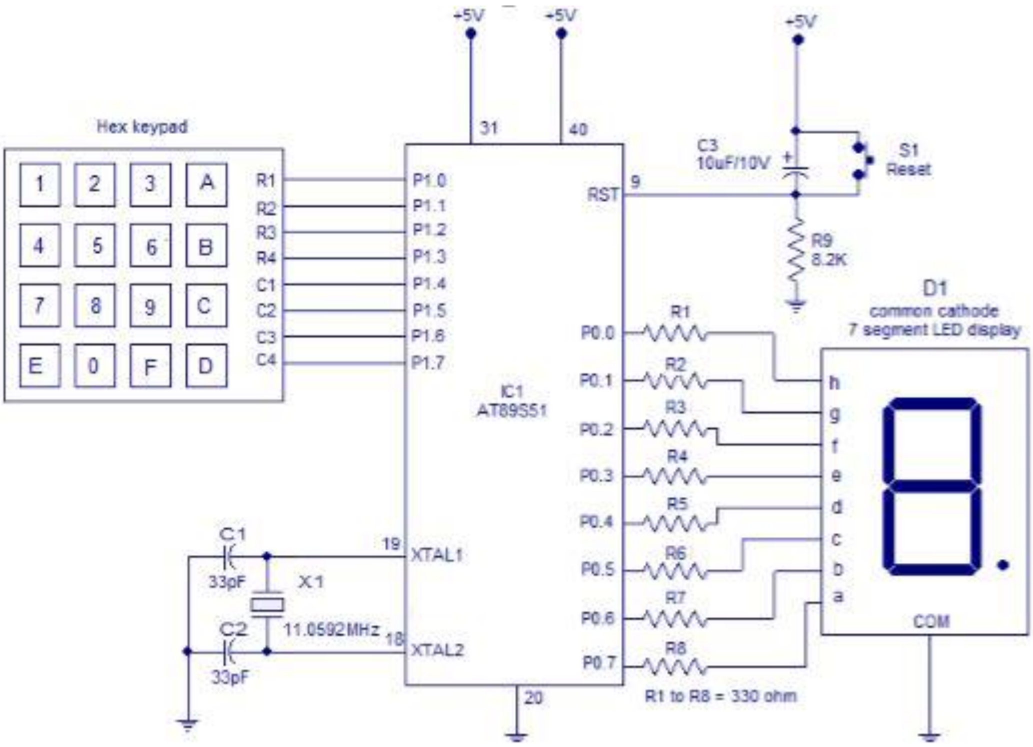
The objective of hex keypad is to input the number displayed on the seven segment display with in the stipulated time. It is interfaced with 8051 to enter the number.





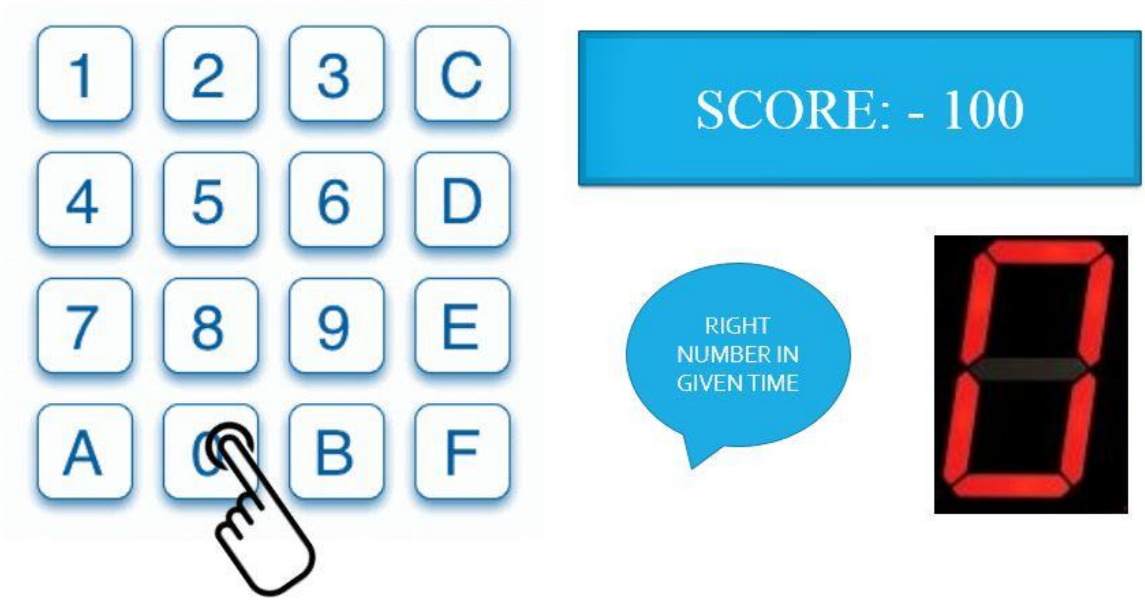
3.5 Interfacing hex keypad to 8051:

The circuit diagram for demonstrating interfacing hex keypad to 8051 is shown below. Like previous 8051 projects, AT89S51 is the microcontroller used here. The circuit will display the character/numeric pressed on a seven segment LED display. The circuit is very simple and it uses only two ports of the microcontroller, one for the hex keypad and the other for the seven segment LED display.

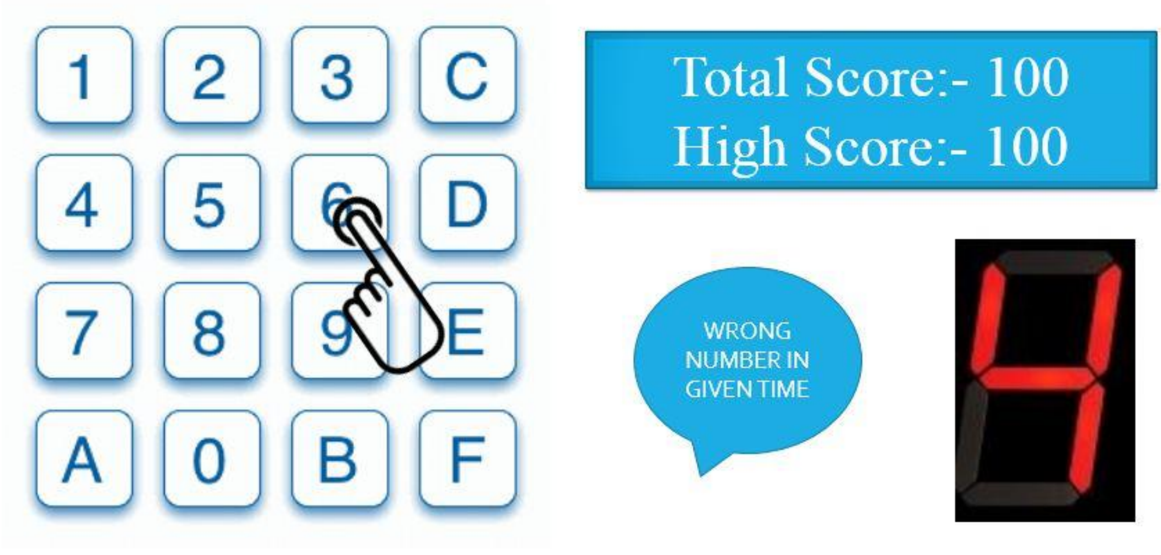


- **Working Example:**

Pass-1:



Pass-2:



CHAPTER-4

IMPLEMENTATION RESULT

4.1 ALGORITHM:

- Declare the PORT2 to LCD data pins and control pins to P0.0, P0.1 and P0.2.
- Now, connect P3.0 to P3.7 to 7 segment display.
- Now, connect the port P1.0 to P1.7 to Hex key pad data pins.
- Now start the circuit and display the random number on 7 segment display.
- Within given time press the same number in hex keypad.
- If the number matches increment the score and decrement the timer.
- If the number went wrong display the total score and updated high score on lcd.

4.2 CIRCUIT DIAGRAM:

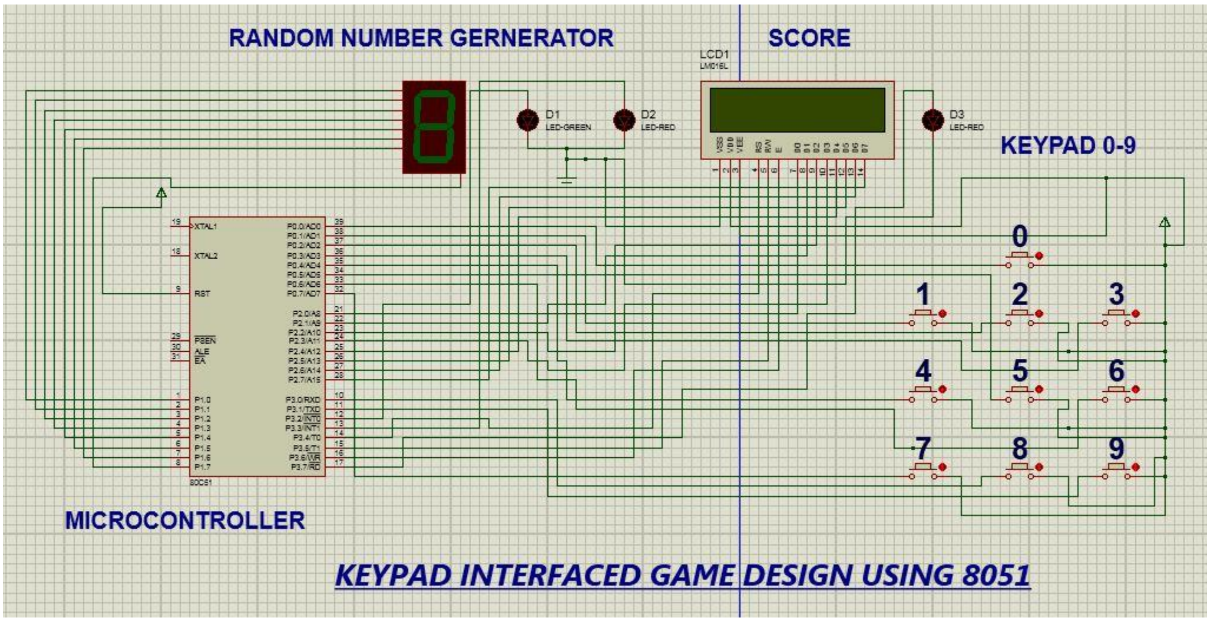
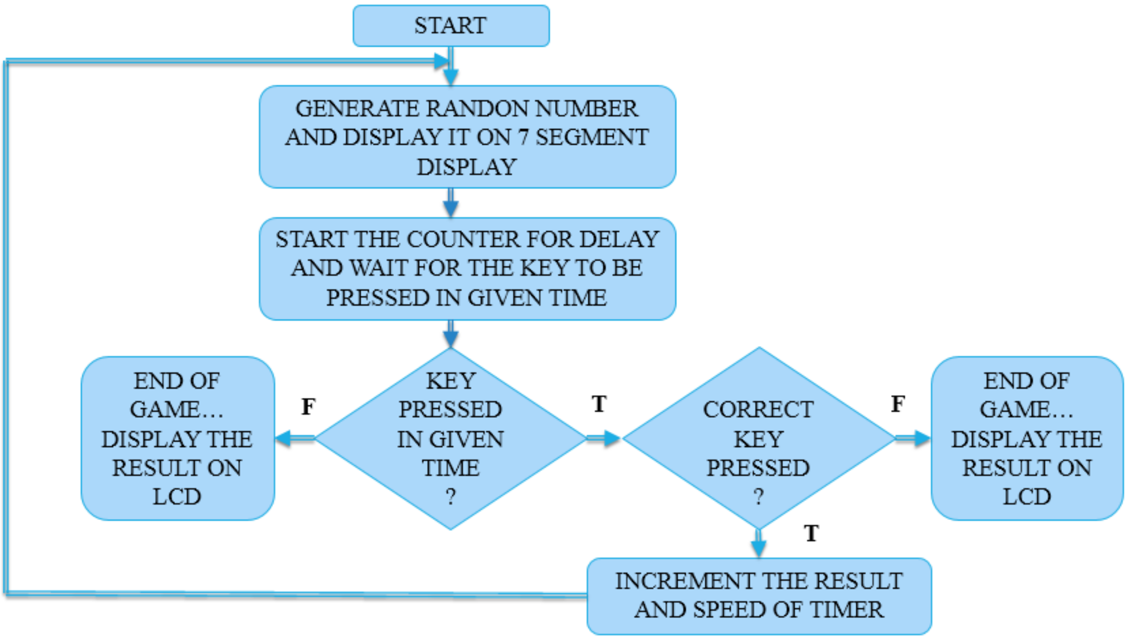


Figure.4.2.1: Circuit Diagram

4.3 OPERATION:

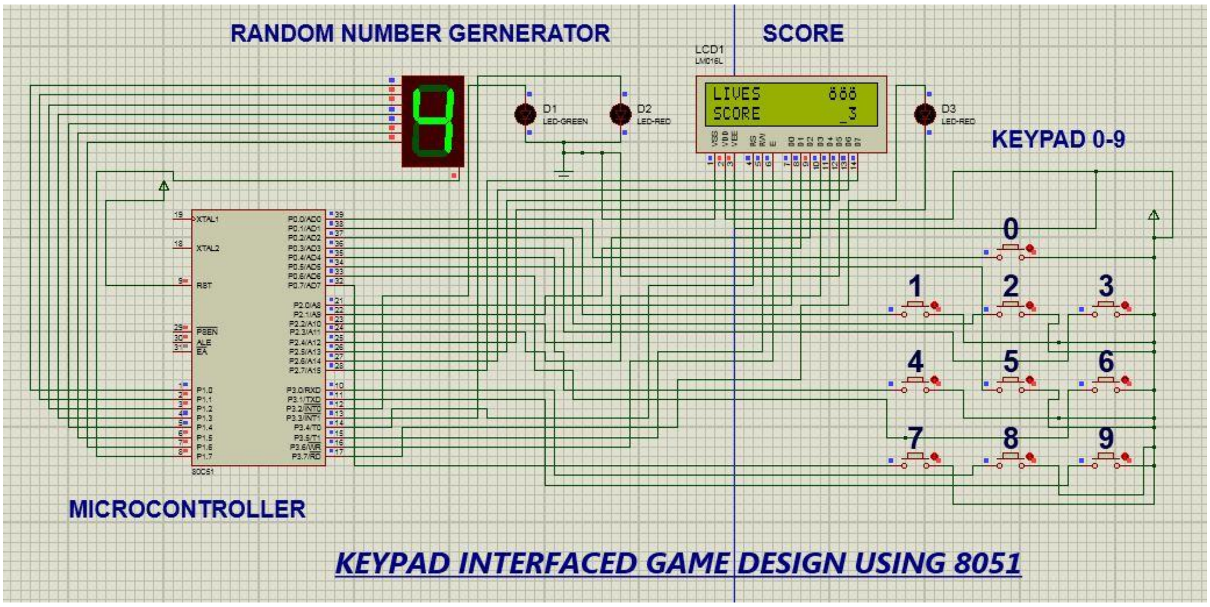
The microcontroller generates some random number on the 7 segment. The conditions of the game is that we have to press the same number that is displayed in a stipulated amount of time say 1 second and send the same number using external keypad to 8051 before it displays the next random number. So if we fail to enter the number in the given time the game ends by displaying your result on the LCD. The delay time will be initiated with certain clock frequency and the frequency goes on increasing dynamically with increase in your score thus enabling more toughness in the game. Your result is the time that you successfully played the game. Your high score will always be updated in one of the registers.

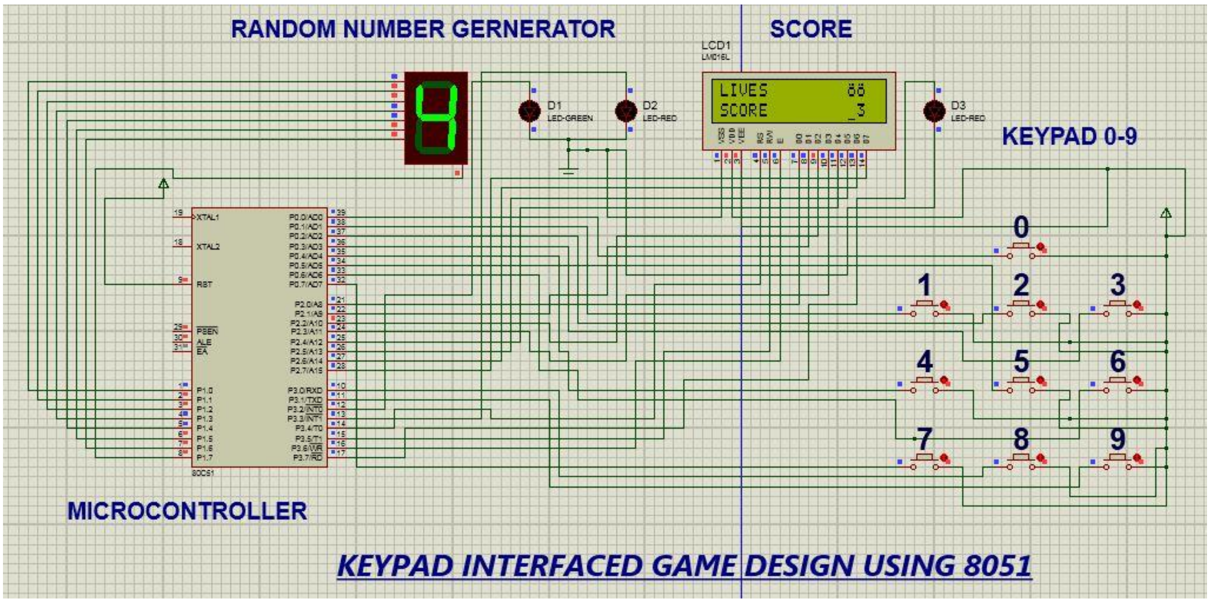
- **Flow Chart:**

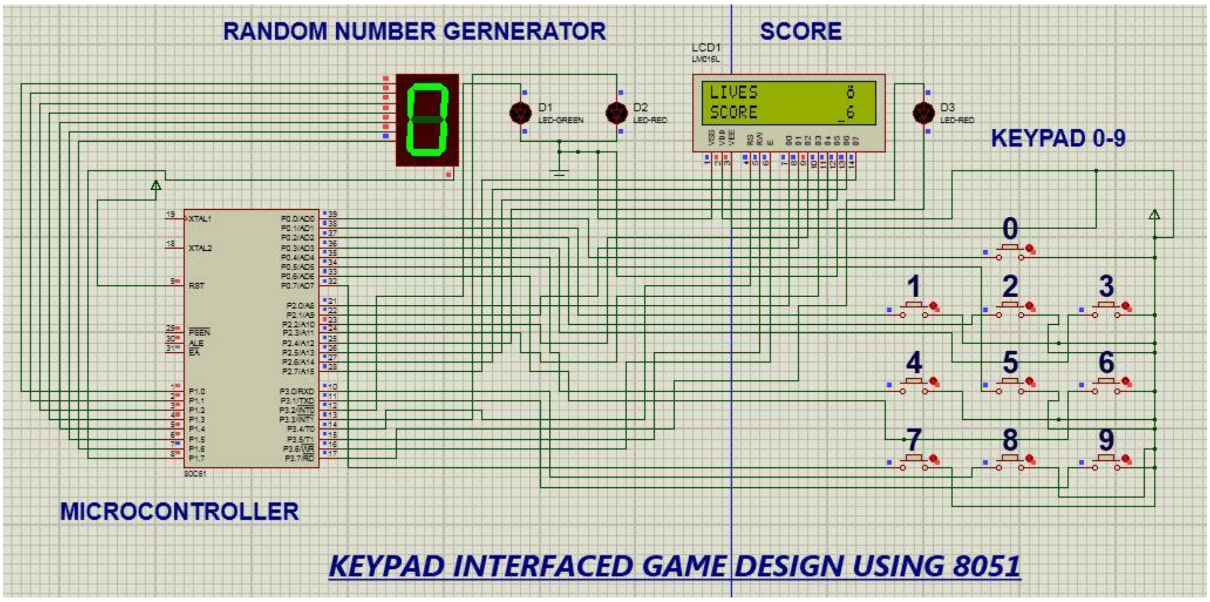

START
GENERATE RANDON NUMBER AND DISPLAY IT ON 7 SEGMENT DISPLAY
START THE COUNTER FOR DELAY AND WAIT FOR THE KEY TO BE PRESSED IN GIVEN TIME
KEY PRESSED IN GIVEN TIME ?
F
END OF GAME… DISPLAY THE RESULT ON LCD
T
CORRECT KEY PRESSED ?
F
END OF GAME… DISPLAY THE RESULT ON LCD
T
INCREMENT THE RESULT AND SPEED OF TIMER

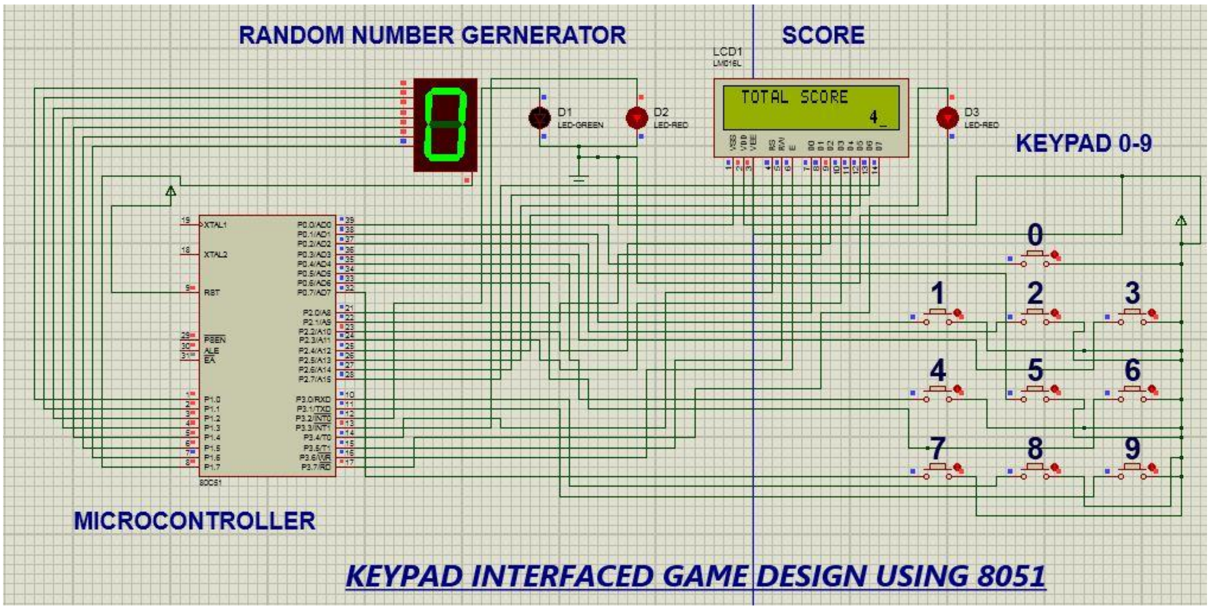
CHAPTER 5

RESULTS






RANDOM NUMBER GERNERATOR
SCORE
LCD1
LM016L
LIVES 6
SCORE _6
D1
LED-GREEN
D2
LED-RED
D3
LED-RED
KEYPAD 0-9
0
1
2
3
4
5
6
7
8
9
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
80C51
MICROCONTROLLER
KEYPAD INTERFACED GAME DESIGN USING 8051


RANDOM NUMBER GERNERATOR
SCORE
LCD1
LM016L
TOTAL SCORE
4_
D1
LED-GREEN
D2
LED-RED
D3
LED-RED
KEYPAD 0-9
0
1
2
3
4
5
6
7
8
9
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
80C51
MICROCONTROLLER
KEYPAD INTERFACED GAME DESIGN USING 8051

CHAPTER-6

ADVANTAGES AND DISADVANTAGES

6.1 ADVANTAGES:

- This circuit generates uniform random numbers between 0-9 which are very much helpful in applications like encryption of passwords etc.
- The circuit dynamically changes the delay which is very important in gaming applications.
- This circuit can be used to control the motion of vehicles dynamically by DTMF technique and properly adjusting the timer values.
- The circuit can further be expanded for various wireless applications, gaming applications and applications where timing is very important.

6.2 DISADVANTAGES:

- Another memory device is to be included into the system because if at all the power fails there is a chancing of losing the data.
- Cannot be used for very speed timers and special high capacity registers are needed to update the score.
- Complex logics are needed to implement the design.

CHAPTER-7

APPLICATIONS

- Can be extended to Robotics applications like controlling purposes.
- Encoding and Decoding of data is possible.
- Cryptography is possible with the same design.
- The circuit dynamically changes the delay which is very important in gaming applications.
- This circuit can be used to control the motion of vehicles dynamically by DTMF technique and properly adjusting the timer values.
- The circuit can further be expanded for various wireless applications, gaming applications and applications where timing is very important.

CHAPTER-8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION:

This circuit generates random numbers which seem to be a true random numbers because of no repetition of the sequence. The 7 segment display in common cathode mode helps in displaying the random number generated by the microcontroller. Here interfacing the hex keypad to 8051 enables us to send the data and this would lead us to applications like DTMF control and knowing the typing speed of the person etc. the dynamic increase in timer speed helps in many applications of gaming, machine control and communication aspects.

8.2 FUTURE SCOPE:

- Can be extended to Robotics applications like controlling purposes.
- DTMF can be included for controlling applications.
- The circuit dynamically changes the delay which is very important in gaming applications.
- This circuit can be used to control the motion of vehicles dynamically by DTMF technique and properly adjusting the timer values.
- The circuit can further be expanded for various wireless applications, gaming applications and applications where timing is very important.

CHAPTER 9

REFERENCES

- http://www.engineersgarage.com/
- http://www.projectguidance.com/
- http://www.electro-tech-online.com/
- Muhammad Ali Mazidi, Janice Mazidi,
- Electronic Voting Machines from Bharat Electronics Limited, India
- Facts About Electronic Elections: www.VotersUnite.Org
- Paper: On Voting Machine Design for Verification and Testability
- Report: Electronic voting – challenges and opportunities

APPENDIX-A

SOURCE CODE

A.1 CODE IN ASSEMBLY LANGUAGE:

```
ORG 00H

MOV P0,#00H
MOV P1,#00H
MOV P2,#00H
MOV P3,#00H
MOV R5,#00H
MOV R6,#00H
MOV R4,#00H
MOV R7,#03H

;LCD_PROGRAMME
lcd:
mov a,#0Ah
acall cr
acall DELAY1
mov a,#0eh
acall cr
mov a,r5
add a,r4
mov r4,a
mov r5,#00h
mov a,#38h
acall cr
mov a,#0eh
acall cr
```

```
mov a,#01h
acall cr
mov a,#06h
acall cr
mov a,#80h
acall cr
;acall delay_lcd
mov a,#'L'
acall dr
mov a,#'I'
acall dr
mov a,#'V'
acall dr
mov a,#'E'
acall dr
mov a,#'S'
acall dr
mov a,#8eh
acall cr
mov a,#04h
acall cr

mov a,r7
mov r6,a
dos: mov a,#0efh
        acall dr
        djnz r6,dos

acall delay_lcd
```

```
 mov a,#06h
acall cr

MOV A,#0C0H
ACALL CR
mov a,#'S'
acall dr
mov a,#'C'
acall dr
mov a,#'O'
acall dr
mov a,#'R'
acall dr
MOV A,#'E'
ACALL DR
acall delay_lcd




MOV DPTR,#NUM

GEN: MOV A,#0CEH
             ACALL CR
             INC R5
             mov a,r5
             mov r6,a
back: MOV A,R6
      mov b,#0ah
              div ab
              mov r6,a
              mov a,b
             ADD A,#30h
             MOV P2,A
             ACALL DR
             mov a,#04h
             acall cr
             CJNE r6,#00H,back
             CLR P3.3
             CLR P3.2
     CLR A
             MOVC A,@A+DPTR
             MOV P1,A
             ACALL DELAY
             LJMP ENDL



delay:  INC DPTR
        mov r2,#100d
        do: clr tr1
                        clr tf1
                                mov tmod,#10h
                                mov th1,#46h
                                mov tl1,#0fdh
                                setb tr1
            START1:             jnb tf1,START
                                clr tf1
                                clr tr1
                djnz r2,do
                        ret
```

```
DELAY1: mov r3,#02d
again: mov r0,#255d
        here1: mov r1,#255d
        here : djnz r1,here
                    djnz r0,here1
                    djnz r3,again
ret

cr: mov p2,a
clr p3.4
clr p3.5
setb p3.6
acall delay_lcd
clr p3.6
ret

dr: mov p2,a
setb p3.4
clr p3.5
setb p3.6
acall delay_lcd
clr p3.6
ret

delay_lcd: mov r0,#10h
herel:mov r1,#10h
herel1:djnz r1,herel1
djnz r0,herel
ret
```

```
START: JB P0.0,GO_0
        JB P0.1,GO_1
        JB P0.2,GO_2
        JB P0.3,GO_3
        JB P0.4,GO_4
        JB P0.5,GO_5
        JB P0.6,GO_6
        JB P0.7,GO_7
        JB P3.0,GO_8
        JB P3.1,GO_9
        LJMP START1

GO_0: CJNE A,#3FH,ENDL0
       SETB P3.2
       ACALL DELAY1
       LJMP GEN

ENDL0: LJMP ENDL

GO_1: CJNE A,#06H,ENDL1
       SETB P3.2
       ACALL DELAY1
       LJMP GEN

ENDL1: LJMP ENDL

GO_2: CJNE A,#5BH,ENDL2
       SETB P3.2
       ACALL DELAY1
```

```
        LJMP GEN

ENDL2: LJMP ENDL

GO_3: CJNE A,#4FH,ENDL3
        SETB P3.2
        ACALL DELAY1
        LJMP GEN

ENDL3: LJMP ENDL

GO_4: CJNE A,#66H,ENDL4
        SETB P3.2
        ACALL DELAY1
        LJMP GEN

ENDL4: LJMP ENDL

GO_5: CJNE A,#6DH,ENDL5
        SETB P3.2
        ACALL DELAY1
        LJMP GEN

ENDL5: LJMP ENDL

GO_6: CJNE A,#7DH,ENDL6
        SETB P3.2
        ACALL DELAY1
        LJMP GEN
```

```
ENDL6: LJMP ENDL

GO_7: CJNE A,#07H,ENDL7
        SETB P3.2
        ACALL DELAY1
        LJMP GEN

ENDL7: LJMP ENDL

GO_8: CJNE A,#7FH,ENDL8
        SETB P3.2
        ACALL DELAY1
        LJMP GEN

ENDL8: LJMP ENDL

GO_9: CJNE A,#6FH,ENDL9
        SETB P3.2
        ACALL DELAY1
        LJMP GEN

ENDL9: LJMP ENDL



ORG 300H
NUM: DB
3FH,7DH,66H,6FH,7DH,3FH,5BH,66H,06H,6DH,3FH,6FH,6DH,6FH,5BH,66H,06H,3FH,
5BH,7DH,6FH,3FH,7DH,66H,6FH,7DH,3FH,5BH,66H,06H,6DH,3FH,6FH,6DH,6FH,5BH,
66H,06H,3FH,5BH,7DH,6FH
```

```
e1:dec r7
   cjne r7,#00h,dol
   ljmp e2
dol:ljmp lcd
ENDL:
SETB P3.3
ACALL DELAY1
cjne r7,#00h,e1
e2:
acall delay_lcd
MOV A,#01H
ACALL CR
MOV A,#02H
ACALL CR
MOV A,#' '
ACALL DR
MOV A,#'T'
ACALL DR
MOV A,#'O'
ACALL DR
MOV A,#'T'
ACALL DR
MOV A,#'A'
ACALL DR
MOV A,#'L'
ACALL DR
MOV A,#' '
ACALL DR
mov a,#'S'
acall dr
```

```
mov a,#'C'
acall dr
mov a,#'O'
acall dr
mov a,#'R'
acall dr
mov a,#'E'
acall dr
acall delay_lcd
MOV A,#0CEH
ACALL CR
mov a,r4
subb a,#02h
             mov r6,a
back1: MOV A,R6
      mov b,#0ah
              div ab
              mov r6,a
              mov a,b
             ADD A,#30h
             MOV P2,A
             ACALL DR
             mov a,#04h
             acall cr
             CJNE r6,#00H,back1
BACKE: CPL P3.7
ACALL DELAY1
SJMP BACKE
;SJMP $
END
```