Department of Electronics & Communication Engineering

LAB MANUAL

SUBJECT: Microprocessor and Interface **[ECE-331]**

B.Tech Year: 3rd Semester: V
(Branch: ECE)

Version: 1(Dec, 2017)

The LNM Institute of Information Technology
Jaipur, Rajasthan-302031

The LNM Institute Of Information Technology
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

Microprocessor Laboratory

Semester: ODD (2018-19)

S No.	Name of the Experiment	Page No.
1.	Write Program for addition and subtraction of two 8-bit numbers using GNU SIM software.	
2.	Write a program for addition and subtraction of two 16-bit numbers using GNU SIM software.	
3.	To write an assembly language for multiplying and division two 8-bit numbers using GNU SIM software.	
4.	To find the largest and smallest element in an array of size ‘n’ using GNU SIM software.	
5.	Write a program to make an LED chaser following the pattern given using EDSIM51 Software. Each transition will take place after a prescribed delay.	
6.	Write a program to display series of numbers on seven segment display Use EDsim51 software for simulation.	
7.	Write a program to interface Digital to Analog Converter for generate the waveform given using EDsim51 software.	
8.	Write a program to interface 4x3 Keypad with 8051. Write a program to display HELO in the seven-segment display when ‘0’ key is pressed in the keypad. Use EDsim51 software to test and verify your program.	

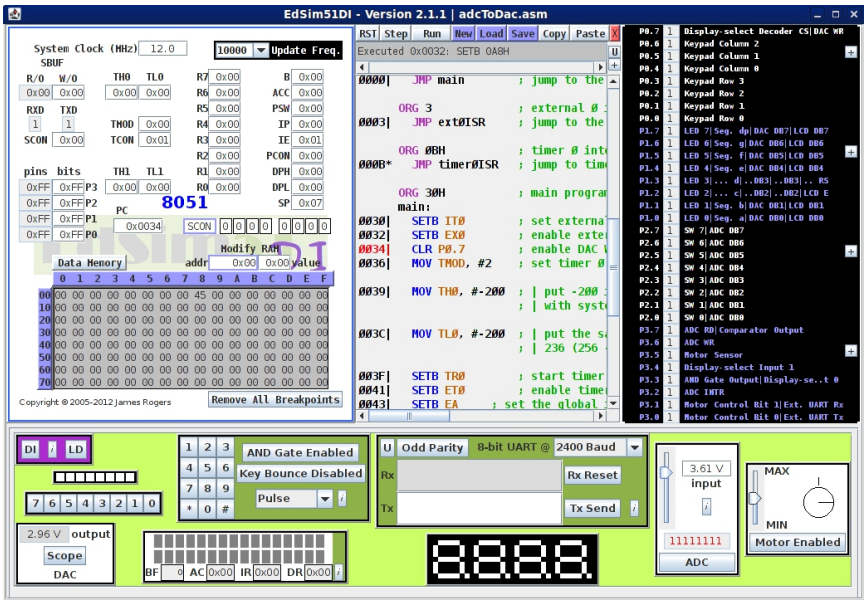
ECE-331: MICROPROCESSOR LAB
EXPERIMENT No. : 01

A. Aim: - Write Program for addition and subtraction of two 8-bit numbers using GNU SIM software.

B. Apparatus Used: - 1. GNU SIM software.

C. Theory:-

EdSim51 Simulator



The top left box gives the user access to all the 8051's registers, data memory and code memory. In the centre is a textbox where the user either loads an assembly program or writes the code directly. Shown above is a program being single-stepped (execution is currently at location 0034H in code memory - hence that line is highlighted).
On the right is a list of the 32 port pins and what each one is connected to. The current value of the port pin is displayed here.
The bottom panel shows all the peripherals that are connected to the 8051.
I hope this simulator will help students learn to write programs to exercise the 8051. Students can begin by ignoring the peripherals. In this way they can first become accustomed to the many different move, arithmetic and branch instructions that make up the 8051 instruction set.

As the students gain experience and confidence they can then write code to scan a keypad, or count the motor's revolutions, or multiplex 7-segment displays, etc.

Some of the peripherals share the same port pins. For example, the 7-segment displays' data lines and the DAC are connected to port 1. This means that, if the student wishes to use the display and the DAC together, he/she must learn how to disable one to access the other.

D. Program for addition of two 8-bit numbers:-

Consider the first number 26H is stored in memory location 8500H and the second number 62H is stored in memory location 8501H. The result after addition of two numbers is to be stored in the memory location 8502H. Assume program starts from memory location 8000H.

Algorithm:

1. Initialize the memory location of first number in HL register pair.
2. Move first number/data into accumulator.
3. Increment the content of HL register pair to initialize the memory location of second data.
4. Add the second data with accumulator.
5. Store the result in memory location 8003H.

Procedure:-

Memory address	Machine Codes	Labels	Mnemonics	Operands	Comments
8000	21, 00, 85		LXI	H, 8500 H	Address of first number in H-L register pair.
8003	7E		MOV	A,M	Transfer first number in accumulator.
8004	23		INX	H	Increment content of H-L register pair
8005	86		ADD	M	Add first number and second number
8006	32, 02, 85		STA	8502H	Store sum in 8502 H
8009	76		HLT		Halt

E. Observation:- (Include your own Table relevant to the Experiment)

Input DATA		RESULT	
Memory	Data	Memory location	Data
8500	26H	8502	88H
8501	62H		

F. Analysis of Results: (Write your own)

G. Conclusions:

The addition of two 8-bit numbers is performed using 8085 microprocessor where sum is 8 – bit.

H. **Precautions:**

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Handle the Trainer kit carefully.

ECE-331: MICROPROCESSOR LAB
EXPERIMENT No. : 02

A. Aim: Write a program for addition and subtraction of two 16-bit numbers using GNU SIM software.

B. Apparatus Used: 1. GNU SIM software.

C. Theory:

Program to Add Two 16 Bit Numbers:-

Assume four 8 bit numbers are stored in the memory location 9000H, 9001H, 9002H and 9003H respectively. Using the data stored in a pair of two memory addresses, we can define one 16-bit number. Hence using two pairs of such memory addresses, we can define two 16-bit numbers. Given below is a program to add these two 16 bit numbers, and store the results in memory location 9004H and 9005H. Check if any carry is generated, and if the carry bit is HIGH then then stored 01H in memory location 9006H, else if carry bit generated is LOW, then store 00H in the memory location 9006H.

Algorithm:

1. Initialize register C for using it as a counter for storing carry value.
2. Load data into HL register pair from the memory address 9000H and 9001H.
3. Exchange contents of register pair HL with DE.
4. Load second data into HL register pair from 9002H and 9003H.
5. Add register pair DE with HL, and store the result in HL.
6. If carry is not generated then go to Loop, else increment register C by 1.
8. Store value present in register pair HL to memory address 9004H and 9005H.
9. Move content of register C to accumulator A, and store the value present in accumulator (i.e., carry) into 9006H.
11. Terminate the program.

D. Program:

Memory address	Machine Codes	Labels	Mnemonics	Operands	Comments
8000	0E,00		MVI	C, 00H	
8002	2A,00,90		LHLD	9000H	Get first 16-bit number in HL
8005	EB		XCHG		Save first 16-bit number in DE
8006	2A,02,90		LHLD	9002H	Get second 16-bit number in HL
8009	19		DAD	D	Add register pair DE with HL and store the result in HL.
800A	D2,0E,80		JNC	LOOP	If carry is present, go to LOOP
800D	0C		INR C		Increment register C by 1.
800E	22,04,90	LOOP:	SHLD	9004H	Store value present in register pair HL to 9004H.

8011	79		MOV	A, C	Move content of register C to accumulator A.
8012	32,06,90		STA	9006H	Store value present in accumulator (carry) into 9006H.
8015	76		HLT		Terminate the program.

Subtract two 16-bit numbers

Form two 16-bit numbers using the data present in memory locations 8500H, 8501H, and 8502H, 8503H respectively. Most significant eight bits of the two numbers are stored in memory location 8501H and 8503H. Subtract the two 16 bit numbers and store the result in memory locations 8504H and 8505H, where, the most significant byte should be stored in memory location 8505H.

E. Observation: (Include your own Table relevant to the Experiment)

F. Analysis of Results: (Write your own)

Example: 1

Address Data

9000H = 34H
9001H = 12H
9002H = 34H
9003H = 12H
9004H = 68H
9005H = 24H
9006H = 00H

1234
+1234

2468

Example: 2

Address Data

(8500H) = 07H
(8501H) = 08H
(8502H) = 05H
(8503H) = 06H
(8504H) = 02H
(8505H) = 02H

Result =
0807H
- 0605H

= 0202H

G. Conclusions:

The addition and subtraction of two 16-bit numbers is performed using 8085 microprocessor where sum and sub is 16 – bit.

H. **Precautions:**

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Handle the Trainer kit carefully.

ECE-331 : MICROPROCESSOR LAB
EXPERIMENT No. : 03

A. Aim: To write an assembly language for multiplying and division two 8-bit numbers using GNU SIM software.

B. Apparatus Used: **1.** 8085 microprocessor kit.
2. +5V power supply.

.

C. Theory:

Consider the first number 25H is stored in register C and the second number 05H is stored in Register E. The result after multiplication of two numbers is to be stored in the HL pair. Assume program starts from memory location 8000H.

Algorithm:

1. Initialize the first number in C register.
2. Initialize the second number in E register.
3. Multiplication the second data with register C and store the result HL pair.

D. Procedure:(Program)

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
8000		0E,25	MVI	C,25	Move the no. in reg. C
8002		1E,05	MVI	E,05	Move the no. in reg. E
8004		06,00	MVI	B,00	Clear reg. B
8006		21,00,00	LXI	H,0000	Initial Product=0000
8009	UP1:	09	DAD	B	HL+BC=>HL
800A		1D	DCR	E	Decrement reg. E
800B		C2,09,70	JNZ	UP1(8009)	Jump ifnot zero tolocation up1
800E		22,00,75	SHLD	8500	Store HL at8500
8011		CF	RST 1		Terminate

E. Observation: (Include your own Table relevant to the Experiment)

INPUT DATA		OUTPUT DATA	
Reg. C	25H	HL Pair	00B9H
Reg. E	05H		
Reg. B	00H		

F. Analysis of Results: (Write your own)
(Include sample calculations/Display/plot/typical graph)

G. Conclusions:

The multiplication of two 8-bit numbers is performed using 8085 microprocessor where multiplication is 8 – bit.

H. Precautions:

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Handle the Trainer kit carefully.

ECE-331: MICROPROCESSOR LAB
EXPERIMENT No. : 04

A. Aim: To write an Assembly Language Program to find the largest number in an array of data using 8085 Microprocessor kit.

B. Apparatus Used: **1.** 8085 microprocessor kit.
2. +5V power supply.

C. Theory:

Algorithm

1) Load the address of the first element of the array in HL pair.
2) Copy the count to register B.
3) Increment the pointer.
4) Get the first data in accumulator.
5) Decrement the count.
6) Increment the pointer.
7) Compare the content of memory addressed by HL pair with that of Accumulator.
8) If Carry = 0, go to step 10 or if Carry = 1 go to step 9.
9) Copy the content of the memory addressed by HL to Accumulator.
10) Decrement the count.
11) Check for Zero of the count. If Zero Flag (ZF) = 0, go to step 6, or if ZF = 1 go to next step.
12) Store the largest data in memory.
13) Terminate the program.

D. Program:

Memory Address	Label	Machine Code	Mnemonics with Operands	Comments
8000		21,00,81	LXI H, 8200H	Set pointer the memory address 8200H to the HL pair
8003		46	MOV B,M	Copy the array size to B register
8004		23	INX H	Increment the memory
8005		7E	MOV A,M	Copy the first data to the Accumulator
8006		05	DCR B	Decrement the array size by 1
8007	loop	23	INX H	Increment the memory
8008		BE	CMP M	Compare number with memory
8009			JNC Next	Jump on no carry to label Next

800C			MOV A,M	Copy the memory content to the Accumulator
800D	Next		DCR B	Decrement register B by 1
800E		C2,07,80	JNZ loop	Jump on no carry to label loop
8011		32,00,82	STA 8300	Store accumulator content to 8200
8014			HLT	

E. Observation: (Include your own Table relevant to the Experiment)

Input at 8200 : 05 H ---------- Array Size
8201 : 0A H
8202 : F1 H
8203 : 1F H
8204 : 26 H
8205 : FE H

Output at 8300 : FE H

F. Analysis of Results:(Write your own)
(Include sample calculations/Display/plot/typical graph)

G. Conclusions:

H. Precautions:

1. Properly connect the 8085 microprocessor kit with power supply terminals.
2. Switch on the power supply after checking connections.
3. Handle the Trainer kit carefully.

ECE-331: MICROPROCESSOR LAB
EXPERIMENT No. : 05

A. Aim: Write a program to make an LED chaser following the pattern given using EDSIM51 Software. Each transition will take place after a prescribed delay.

B. Apparatus Used: **1.** EDSIM51 Software.

Theory:

This program displays the binary pattern from 0 to 255 (and back to 0) on the LEDs Interfaced with port 1.A 1 in the pattern is represented by the LED on, while a 0 in the pattern is represented by the LED off. However, logic 0 on a port 1 pin turns on the LED, therefore it is necessary to write the inverse of the pattern to the LEDs. The easiest way to do this is to send the data FFH to 0 (and back to FFH) to the LEDs. Since port 1 is initially at FFH all we need to do is

C. Observation: (Include your own Table relevant to the Experiment)

D. Analysis of Results:(Write your own)
(Include sample calculations/Display/plot/typical graph)

E. Conclusions:

F. Precautions:

4. Properly connect the 8085 microprocessor kit with power supply terminals.
5. Switch on the power supply after checking connections.
6. Handle the Trainer kit carefully.

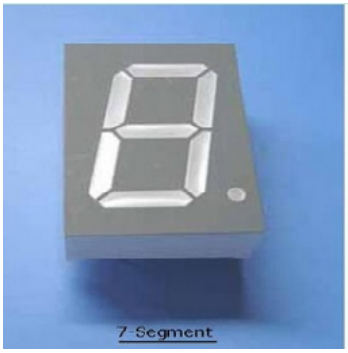
ECE-331: MICROPROCESSOR LAB
EXPERIMENT No. : 06

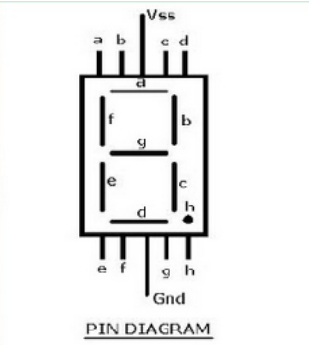
A. AIM: Write a program to display series of numbers on seven segment display with AT8051 Microcontroller.

B. Apparatus Used: EDSIM51 Software, AT89S51 Microcontroller etc.

C. Theory:-

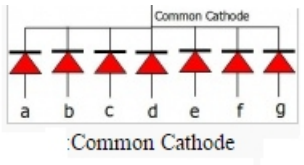
A 7-Segment display is a useful electronic component use to produce numeric, alphabetic and some non-alphabetic symbols using a specific arrangement of LEDs as shown in figure here.



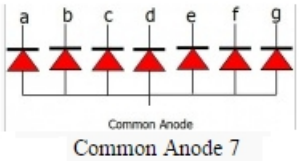


PIN DIAGRAM

A 7-Segment display has 10-Pins representing each **a**, **b**, **c**, **b**, **e**, **f**, **g** and **h** LEDs respectively along with two extra pins for **GND** and **VSS**. following shown an original 7-Segment display device along with its pin diagram. LED **h** is also denoted by symbol **dp**.
Since these are basically LEDs arranged as a group they can either have anode in common or cathode. Seven Segment are available in two configuration - **(1)** Common Anode **(2)** Common Cathode. Here common anode seven segment display is used because the output current of the microcontroller is not sufficient enough to drive the LED's, similar to the case of driving an LED. The circuit diagram shows the connections of seven segment to the controller. The pins a to g of the Seven Segment are connected to the Port P1 of the microcontroller.

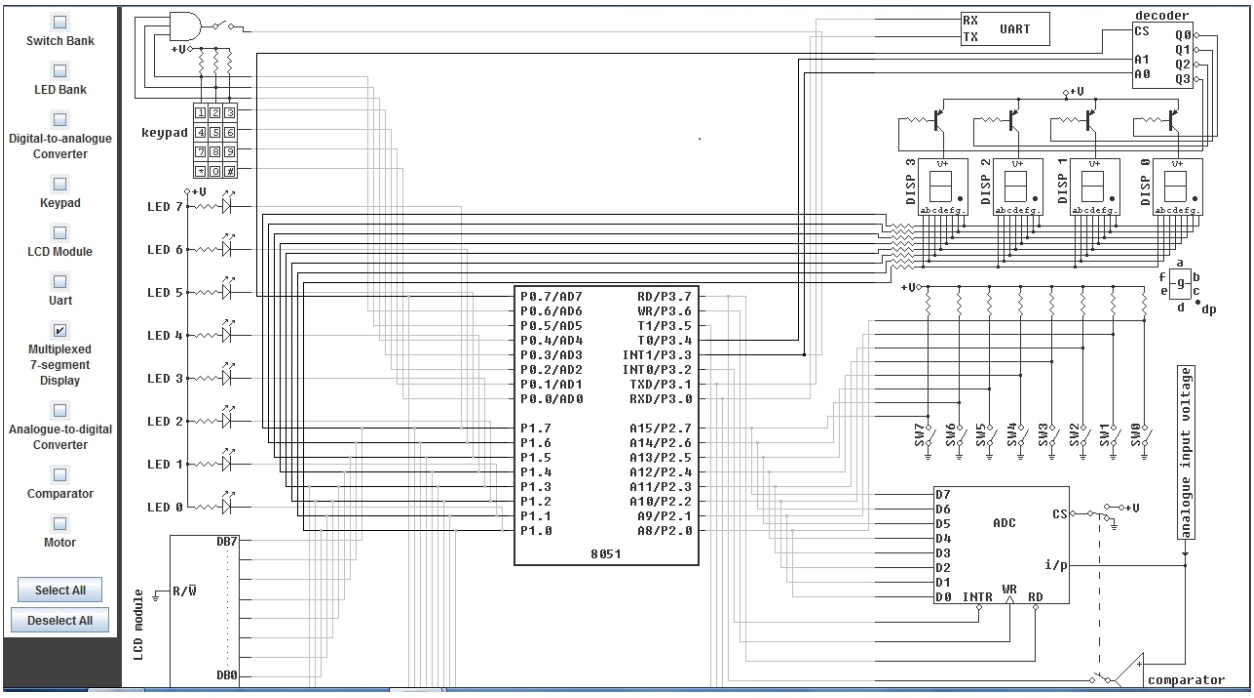


:Common Cathode



Common Anode 7

Digit	H	G	F	E	D	C	B	A
1	1	1	1	1	1	0	0	1
2	1	0	1	0	0	1	0	0
3	1	0	1	1	0	0	0	0
4	1	0	0	1	1	0	0	1



When running this program, best viewed with **Update Freq.** set to **100**

Program to interface seven-segment display:-

This program multiplexes the number 1234 on the four 7-segment displays.

```
start:
        SETB P3.3               ; |
        SETB P3.4               ; | enable display 3
        MOV P1, #11111001B      ; put pattern for 1 on display
        CALL delay
        CLR P3.3                ; enable display 2
        MOV P1, #10100100B      ; put pattern for 2 on display
        CALL delay
        CLR P3.4                ; |
        SETB P3.3               ; | enable display 1
        MOV P1, #10110000B      ; put pattern for 3 on display
        CALL delay
        CLR P3.3                ; enable display 0
        MOV P1, #10011001B      ; put pattern for 4 on display
        CALL delay
        JMP start               ; jump back to start

    ; a crude delay
        delay:
        MOV R0, #200
        DJNZ R0, $
        RET
```

D. Observation: Write/ Plot Your Own With Observation Table (If Required).

E. Analysis of Results: (Write your own)

F. Conclusions:

G. Precautions:

1. Properly connect the AT89S52 Ultra Development Kit with USB Cable.
2. Handle the Trainer kit carefully.

ECE-331: MICROPROCESSOR LAB
EXPERIMENT No. : 7

A. Aim: Write a program to interface Digital to Analog Converter use EDsim51 software for positive ramp generator.

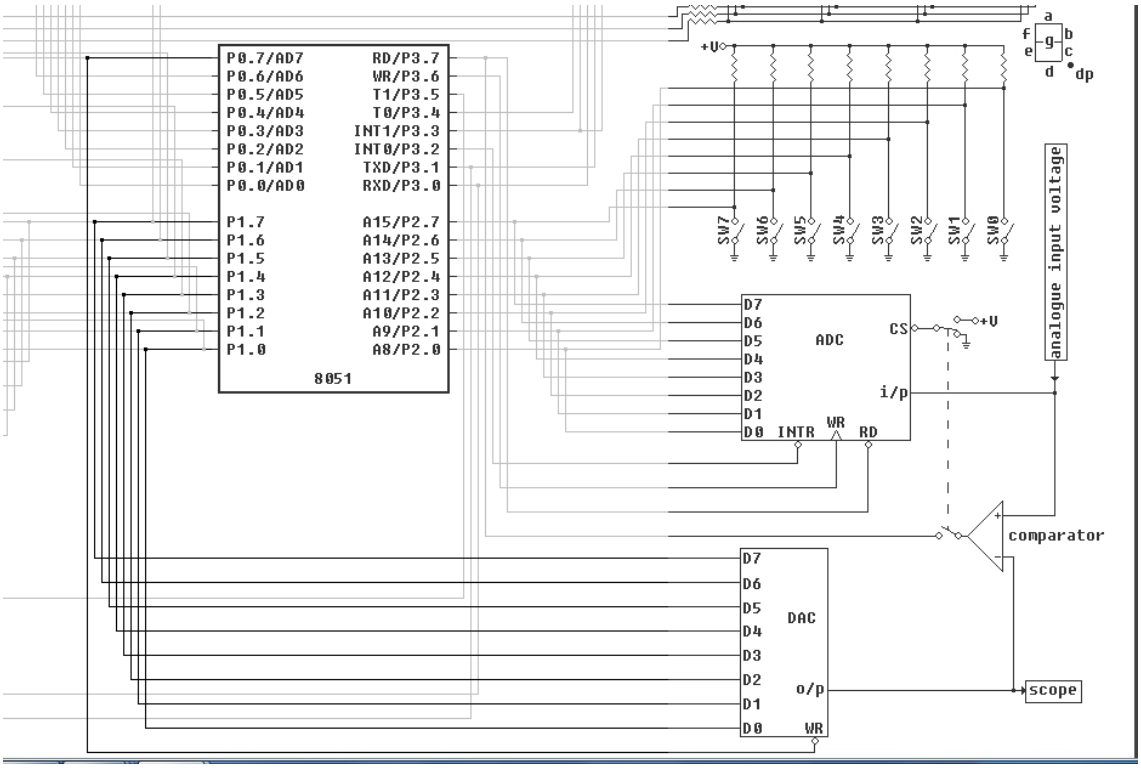
B. Apparatus Used:

1.Use EDsim51 software
2.Digital to Analog Converter

C. Theory:

DAC-0800 is a monolithic 8-bit digital to analog converter. The output voltage of this DAC is programmable in the range of 0 to approx.5V DC. The effect of this circuit is to produce a DAC output that ramps up to whatever level the analog input signal is at, output the binary number corresponding to that level, and start over again.

Block Diagram



D. Program:

```
CLR P0.7        ; enable the DAC WR line
loop:
        MOV P1, A    ; move data in the accumulator to the ADC inputs (on P1)
        ADD A, #4    ; increase accumulator by 4
        JMP loop     ; jump back to loop
```

Note: When running this program, best viewed with Update Freq. set to 1.

Functional Description:-

E. Observation:

F. Analysis of Results:(Write your own)
(Include sample calculations/Display/plot/typical graph)

G. Conclusions:

H. Precautions:

7. Properly connect the 8085 microprocessor kit with power supply terminals.
8. Switch on the power supply after checking connections.
9. Handle the Trainer kit carefully.

ECE-331: MICROPROCESSOR LAB
EXPERIMENT No. : 8

G. Aim: Write a program to interface 4x3 Keypad with 8051. Use EDsim51 software to test and verify the code.

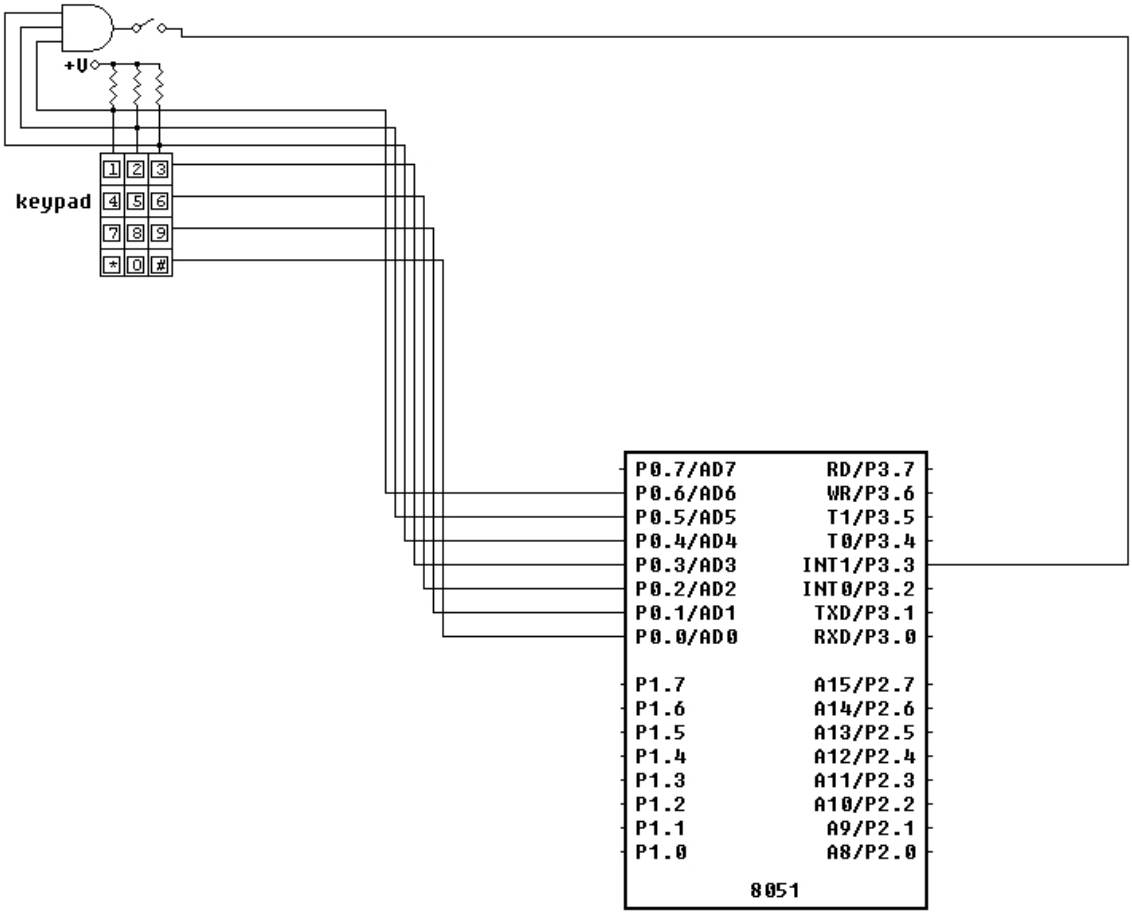
H. Apparatus Used:

3.Use EDsim51 software
4.4x3 Keypad
5.LEDs and Seven Segment Displays

I. Theory:

DAC-0800 is a monolithic 8-bit digital to analog converter. The output voltage of this DAC is programmable in the range of 0 to approx.5V DC. The effect of this circuit is to produce a DAC output that ramps up to whatever level the analog input signal is at, output the binary number corresponding to that level, and start over again.

Schematic:



Logic diagram showing the keypad connections

J. Program:

```
start:
        MOV R0, #0              ; clear R0 - the first key is key0

        ; scan row0
        SETB P0.3               ; set row3
        CLR P0.0                ; clear row0
        CALL colScan            ; call column-scan subroutine
        JB F0, finish       ; | if F0 is set, jump to end of program
                            ; | (because the pressed key was found and its number is in  R0)

        ; scan row1
        SETB P0.0               ; set row0
        CLR P0.1                ; clear row1
        CALL colScan            ; call column-scan subroutine
        JB F0, finish       ; | if F0 is set, jump to end of program
                            ; | (because the pressed key was found and its number is in  R0)

        ; scan row2
        SETB P0.1               ; set row1
        CLR P0.2                ; clear row2
        CALL colScan            ; call column-scan subroutine
        JB F0, finish       ; | if F0 is set, jump to end of program
                            ; | (because the pressed key was found and its number is in  R0)

        ; scan row3
        SETB P0.2               ; set row2
        CLR P0.3                ; clear row3
        CALL colScan            ; call column-scan subroutine
        JB F0, finish       ; | if F0 is set, jump to end of program
                            ; | (because the pressed key was found and its number is in  R0)


        JMP start               ; | go back to scan row 0
                    ; | (this is why row3 is set at the start of the program
                    ; | - when the program jumps back to start, row3 has just been scanned)

finish:
        JMP $       ; program execution arrives here when key is found - do nothing


; column-scan subroutine
```

```
colScan:
        JNB P0.4, gotKey    ; if col0 is cleared - key found
        INC R0                          ; otherwise move to next key
        JNB P0.5, gotKey    ; if col1 is cleared - key found
        INC R0                          ; otherwise move to next key
        JNB P0.6, gotKey    ; if col2 is cleared - key found
        INC R0                          ; otherwise move to next key
        RET                             ; return from subroutine - key not found
gotKey:
        SETB F0                         ; key found - set F0
        RET                             ; and return from subroutine
```

Note: When running this program, best viewed with Update Freq. set to 1.

Functional Description:-

K. Observation:

L. Analysis of Results:(Write your own)
(Include sample calculations/Display/plot/typical graph)

M. Conclusions:

N. Precautions:

10. Properly connect the 8085 microprocessor kit with power supply terminals.
11. Switch on the power supply after checking connections.
12. Handle the Trainer kit carefully.