ACADEMY OF TECHNOLOGY

DEPARTMENT: ECE

SEMESTER: 4TH

PAPER NAME:

MICROPROCESSOR

&

MICROCONTROLLER

VIRTUAL LAB

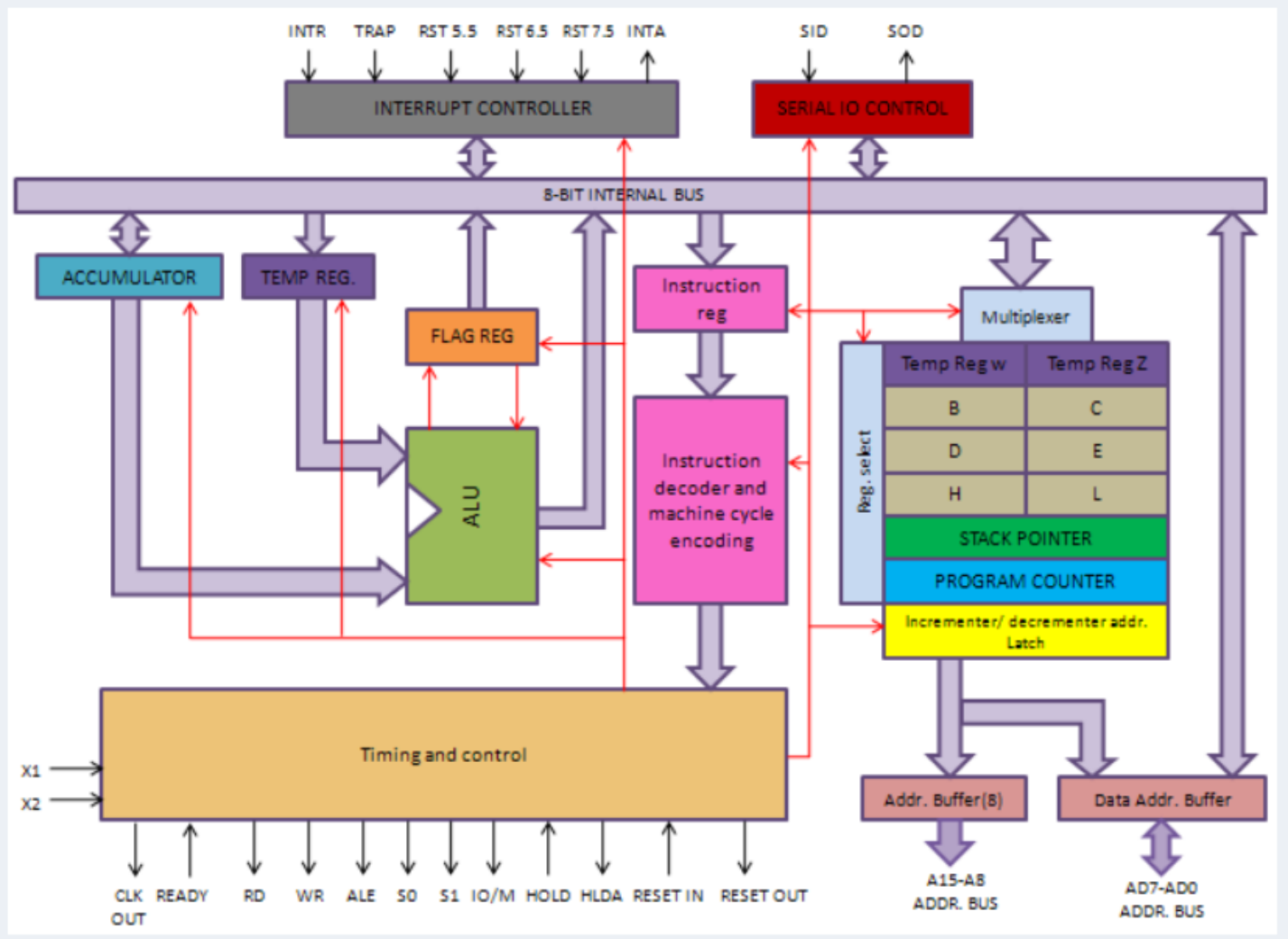
PAPER CODE: EC - 493

CONTENTS

INTRODUCTION

The 8085 Microprocessor

A microprocessor is a multipurpose, programmable logic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions and provide result as output. The power supply of 8085 is +5V and clock frequency is 3MHz.



8085 Architecture Diagram

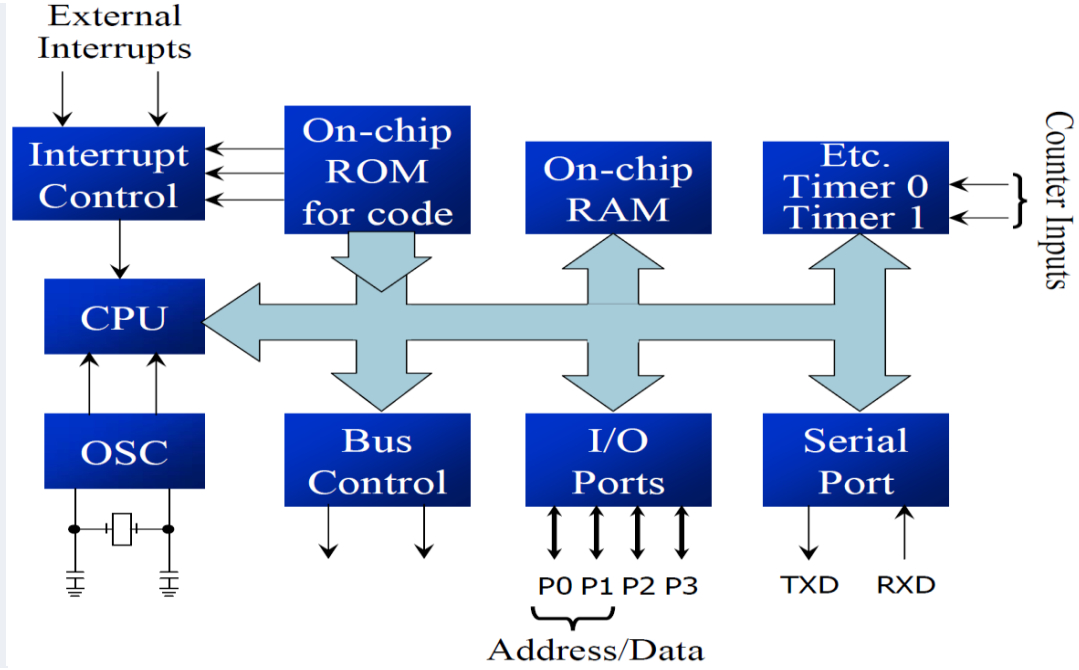
8085 Microprocessor Trainer Model MPT-85LCD:

8085 Microprocessor Trainer Model MPT-85LCD Features

- This is an LCD unit, with built in Line-by-line assembler and disassembler facility
- Programs can be assembled or Dissembled using built-in assembler OR by entering HEX codes.
- 8085 Microprocessor operating at 6.144 MHZ
- 32KB powerful software monitor
- 27C128 EPROM
- 32KB RAM
- Three 16-bit programmable timers from 8253/8254
- 48 Programmable I/O lines from two nos. of 8255
- Serial interface using 8251
- 50 Pin FRC connector for system bus expansion
- 50 pin FRC connectors for user interface from 8255
- 9 Pin D type connector for RS 232G interface
- 9 Pin D type connector for connecting to Power supply
- Six different selectable baud rates from 150 to 9600
- 101 PC type keyboard for entering user address/data and for commands
- Built in line-by-line assembler and disassembler
- Built in power supply
- User friendly software monitor for loading and executing programs with break point facility
- 50 Pin Flat Ribbon Cable (FRC)
- Connects to PC at RS232C Serial port

The 8051 Microcontroller

The 8051 is an 8-bit controller. The CPU can work on only 8 bits of data at a time. The 8051 has 128 bytes of RAM, 4K bytes of on-chip ROM, Two timers, One serial port, Four I/O ports, each 8 bits wide, 6 interrupt sources. Following diagram shows the architecture of the 8051 Microcontroller



8051 Architecture Diagram

8051 Microcontroller Trainer Model MPT-51LCD:

8051 Microcontroller Trainer Model MPT-51LCD Features:

- This is an LCD unit, with built in assembler and disassembler facility
- Programs can be assembled or Dissembled using built-in assembler OR by entering HEX codes.

- 8051 Microcontroller operating at 10 MHz
- 32KB powerful software monitor 27C256
- EPROM 32KB external RAM for user applications 256 bytes on chip RAM (partially used by Firmware) 1 Two 16-bit Programmable Timer
- 50 Pin FRC connector for system bus expansion 50 pin FRC connector for user interface from 8255
- 9 Pin D type connector for RS 232C interface I 9 Pin D type connector for connecting to Power supply
- Six different selectable baud rates from 150 to 9600 One timer and external interrupt 101 PC type keyboard for entering user address/data and for commands Built in line-by-line assembler and disassembler
- Built in power supply
- User-friendly software monitor for loading and executing programs with break point facility
- 50 Pin Flat Ribbon Cable (FRC)
- Connects to PC at RS232C Serial port.

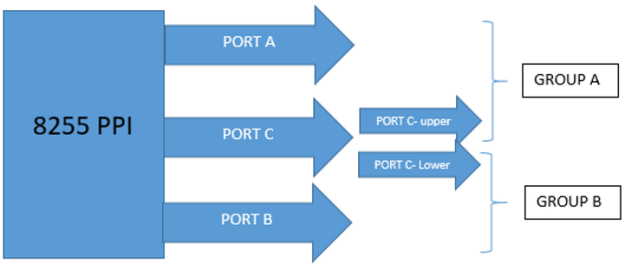
The 8255 PPI (Programmable peripheral interface)

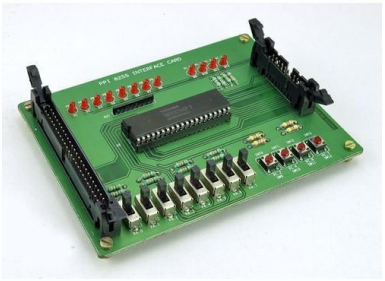
The 8255 PPI (Programmable peripheral interface) is a device through which we can achieve the basic method of communication between humans or machines and the microprocessor. Here, the PPI stands for "Programmable Peripheral Interface". As the name itself suggests, it is a peripheral device for a machine which is programmed to act as an interface, and the 8255 PPI is an interface between the microprocessors (basically 80- microprocessor series) and the input-output devices, ADC (Analog to Digital Converters), DAC (Digital to Analog Converters), etc.

Characteristics of the 8255 PPI:

- The **8255 IC is a Programmable Peripheral Interface (PPI)** which is used to interface the microprocessors with the various devices.
- The **8255 PPI** chip is a 40 pin IC.
- In the 8255 IC, there are 24 input/ output lines.
- In this IC, the input/ output lines are divided into e different ports: Port A, Port B, and Port C. Each port contains 8 data lines. The Port C is further divided into two different parts as C- upper and C- lower. The Port A and the Port C- upper together form the Group A and the Port B and the Port C- Lower Together form the Group B.
- There are two modes in which the 8255 PPI operates. The one is the BSR mode (which stands for Bit Set Reset) and the other one is the IO mode (which stands for Input- Output).
- There are further 3 modes for the IO mode: mod- 0, mod- 1 and mod- 2.
- It operates on +5V of the voltage supply.
- The 8255 contains 1 pin for voltage supply and 1 pin is used for ground voltage.
- The operations of the 8255 PPI are controlled via the 8-bit binary code which is stored in the Control word. The control word refers to 8 pins (from 27 to 34) which store the 8-bit binary code in it.
- The contents of the control word decide which mode will be used, which lines will be used and what action will be performed. Hence, the control word depends upon the mode which is being used.
- The handshaking procedure can be followed using the 8255 PPI.

The following diagram illustrates the division of 8255 I/O lines into different ports,





8255 INTERFACING with SWITCH LED MODULE

LIST OF PROGRAMS:

1. Familiarization with 8085 & 8051simulator on PC.

2. Study of prewritten programs using basic instruction set (data transfer, Load/Store, Arithmetic, Logical) on the KIT. Assignments based on above

3. Programming using kit and simulator for:
   i) Table look up
   ii) Copying a block of memory
   iii) Shifting a block of memory
   iv) Packing and unpacking of BCD numbers
   v) Addition of BCD numbers
   vi) Binary to ASCII conversion
   vii) String Matching, Multiplication using shift and add method and Booth's Algorithm

4. Program using subroutine calls and IN/OUT instructions using 8255 PPI on the trainer kit e.g., subroutine for delay, reading switch state and glowing LEDs accordingly.

5. Study of timing diagram of an instruction on oscilloscope.

6. Interfacing of 8255: Keyboard and Multi-digit Display with multiplexing using 8255

7. Study of 8051 Micro controller kit and writing programs as mentioned in S/L3. Write programsto interface of Keyboard, DAC and ADC using the kit.

8. Serial communication between two trainer kits

VIRTUAL LAB PLATFORM/ SIMULATOR:

8085 MICROPROCESSOR & IT'S INTERFACING:

1. http://vlabs.iitb.ac.in/vlabs-dev/labs_local/microprocessor/labs/explist.php
2. https://www.sim8085.com

8051 MICROCONTROLLER & IT'S INTERFACING:

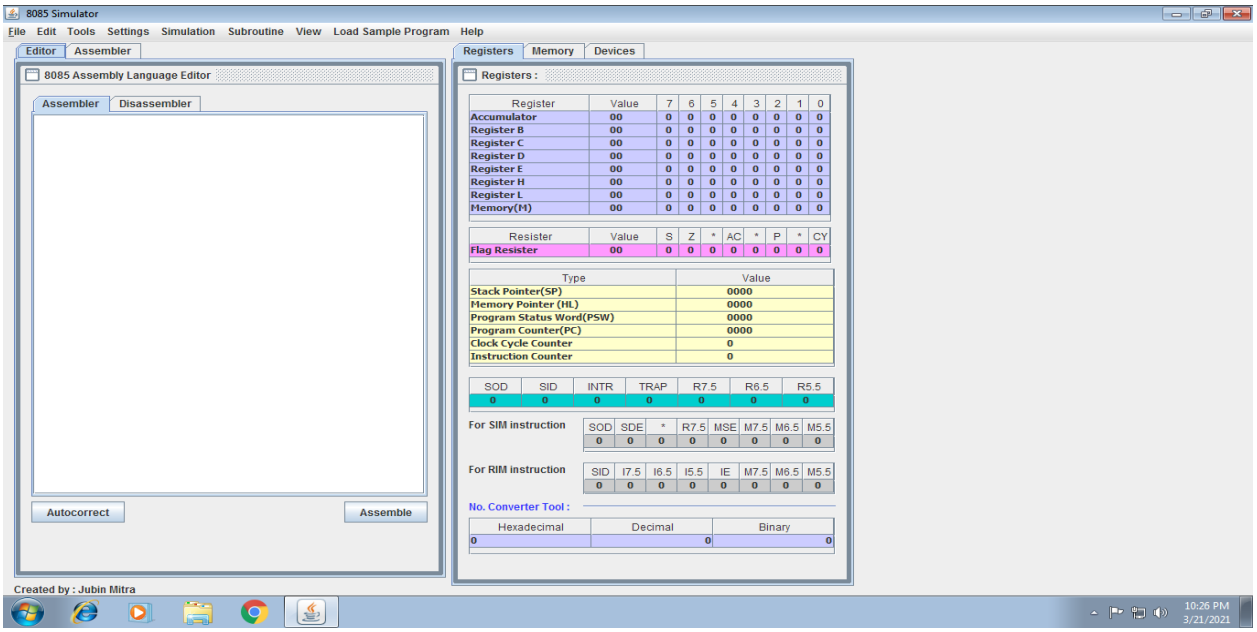
1. https://www.edsim51.com/installationInstructions.html
2. http://vlabs.iitb.ac.in/vlabs-dev/labs/8051-Microcontroller-Lab/labs/index.php

PROCEDURE:

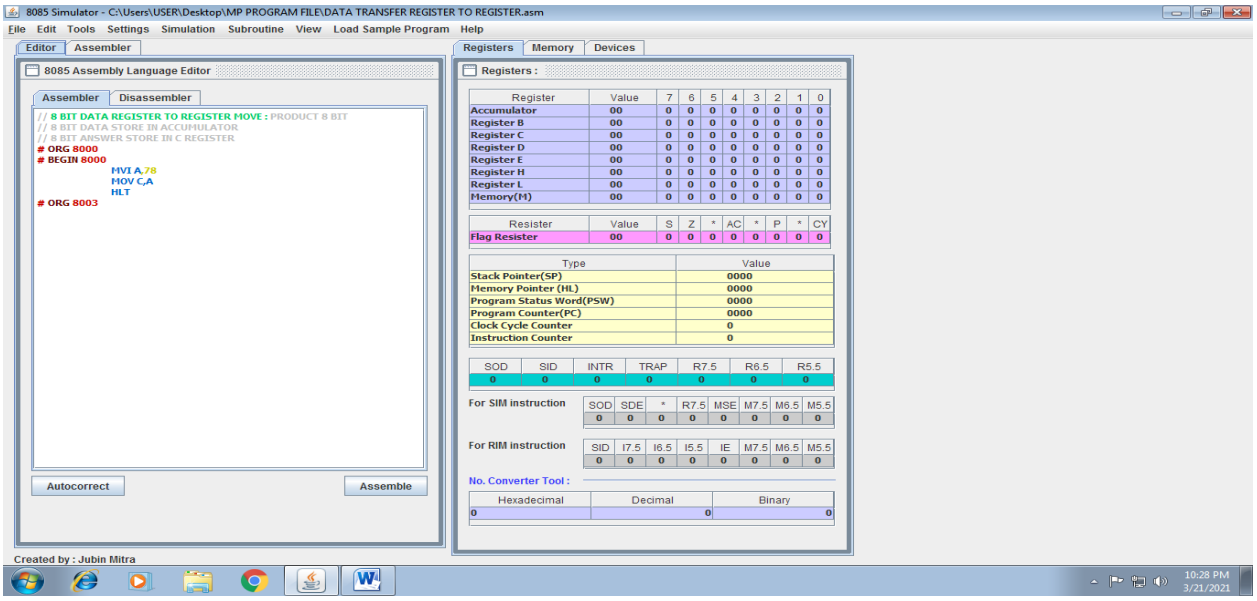
8085 PROGRAMMING

1. Click on **8085 simulator version 2** : **Microprocessor Simulator** and **Java** download to install it.
2. Write the program code on editor file window.
3. Store some input data at specified Memory Array.
4. Click on the option Assemble for simulation.
5. Write the starting address of the program to simulate option.
6. Simulation of entire code could be run at a time or step by step.
7. Execution results can be checked through the Register, Memory and Device environment window.
8. Note down updated register and memory content value for verification of result.
9. For instruction mnemonic details, click the Help option.

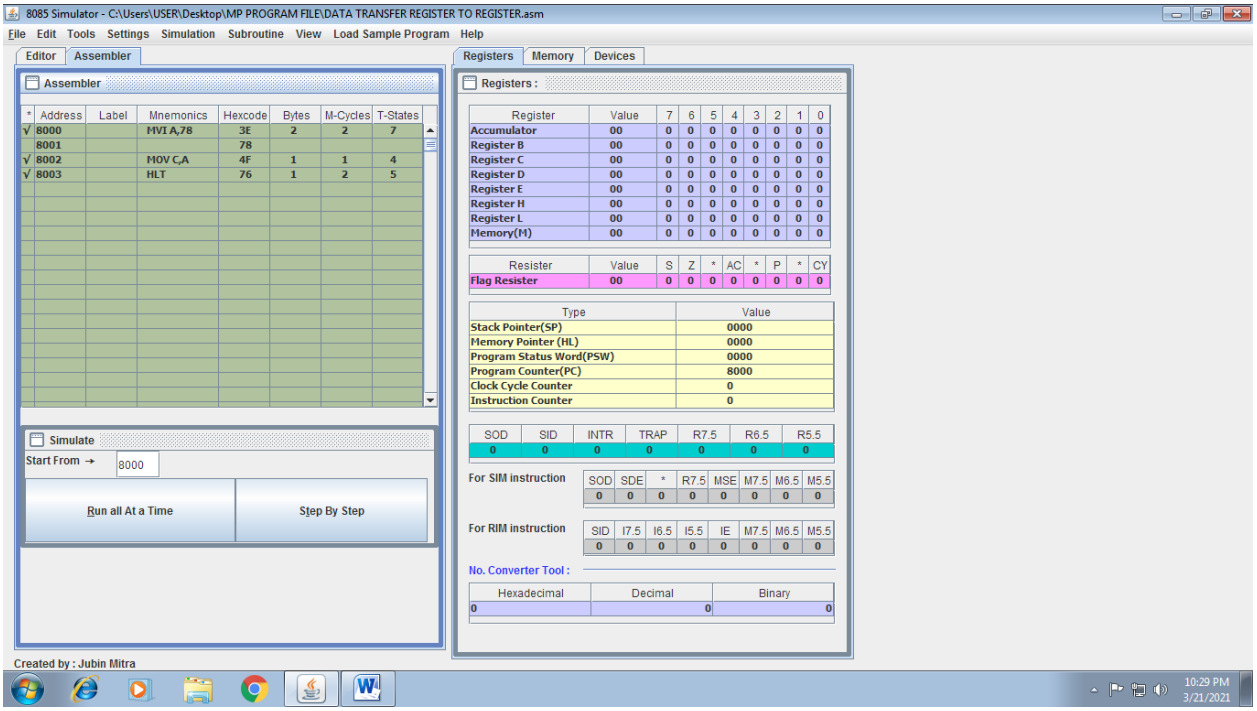
STEP 1: OPEN THE 8085 SIMULATOR ICON



STEP 2: WRITE PROGRAM IN WHITE BLANK SPACE / LOAD PREVIOUS PROGRAM FROM 'FILE' OPTION.

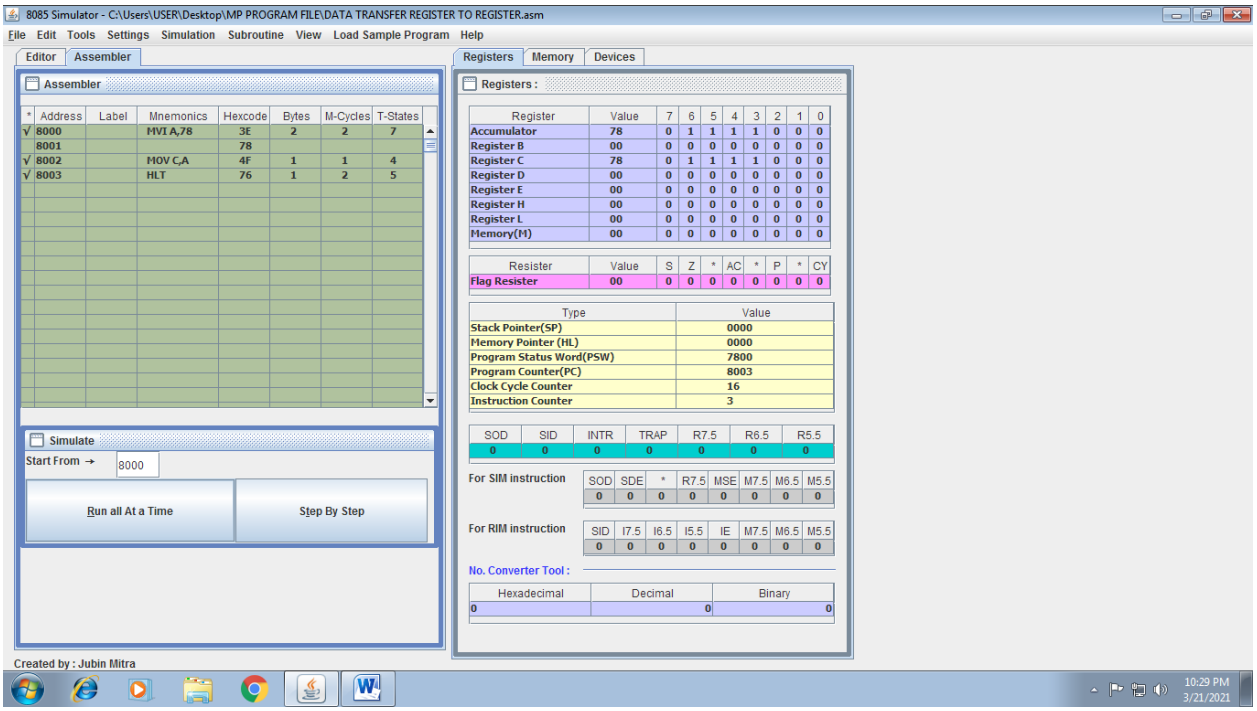


STEP 3: CLICK ON 'AUTOCORRECT' OPTION AND THEN CLICK ON 'ASSEMBLE' OPTION.

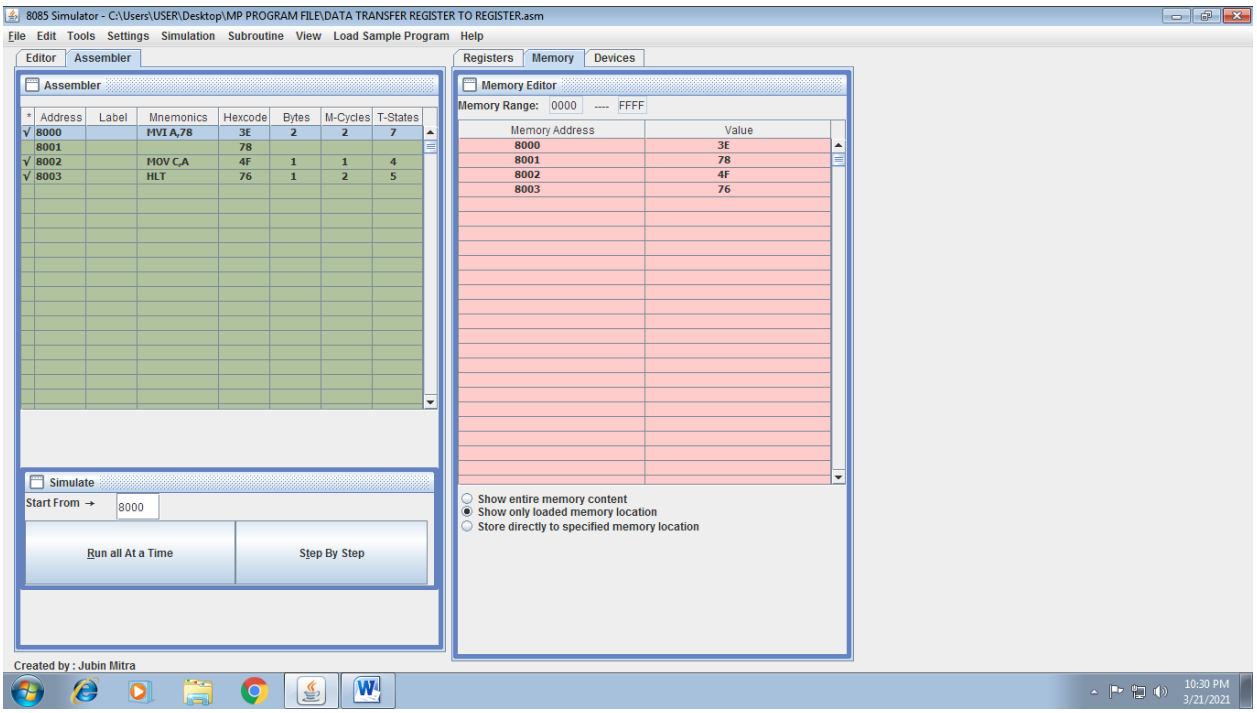


STEP 4: CLICK ON 'Step By Step' OPTION TO EXECUTE INSTRUCTIONS OF THE PROGRAM ONE BY ONE OR CLICK ON OPTION 'Run all a time' TO EXECUTE ALL INSTRUCTIONS OF THE PROGRAM AT A TIME.

OUTPUTS WILL BE DISPLAYED VIA REGISTER AND MEMORY LOCATION.



STEP 5: SELECT ‘MEMORY’ TO DISPLAY OUTPUT DATA STORED IN MEMORY.



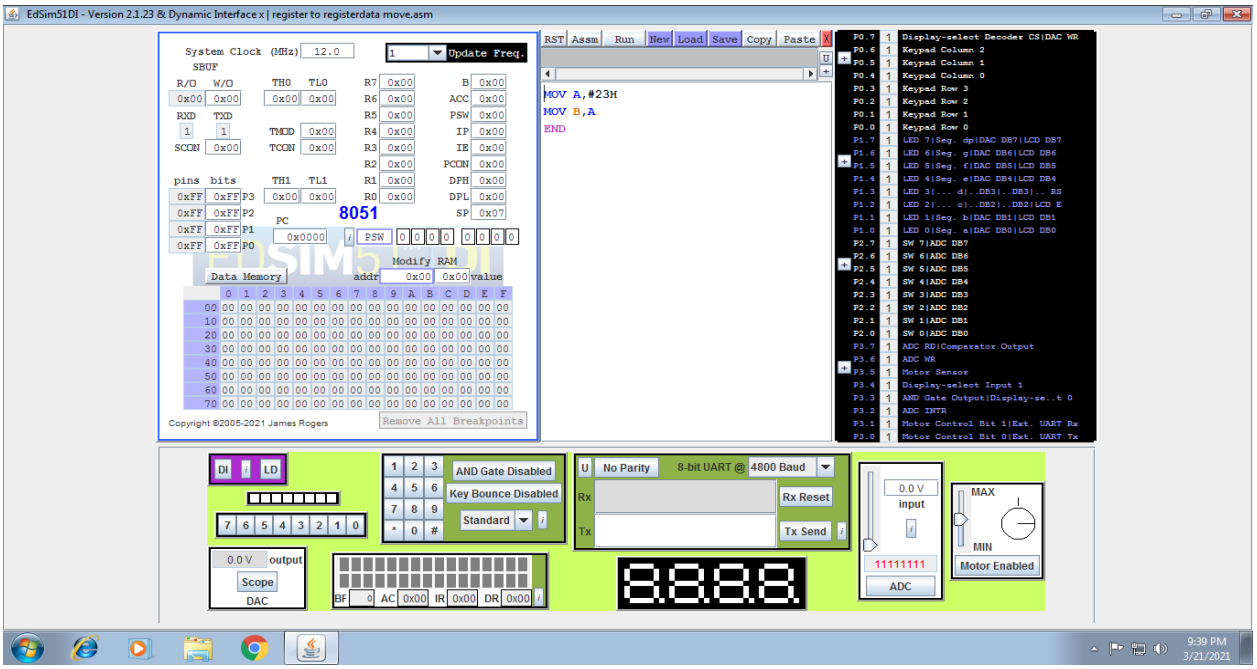
TO CHANGE/EDIT ANY INSTRUCTION, SELECT ‘EDITOR’ OPTION. AFTER EDITING THE PROGRAM SHOULD BE RUN AGAIN.

TO WRITE A NEW PROGRAM SELECT THE ‘EDIT’ OR ‘DISASSEMBLER’ OPTION.

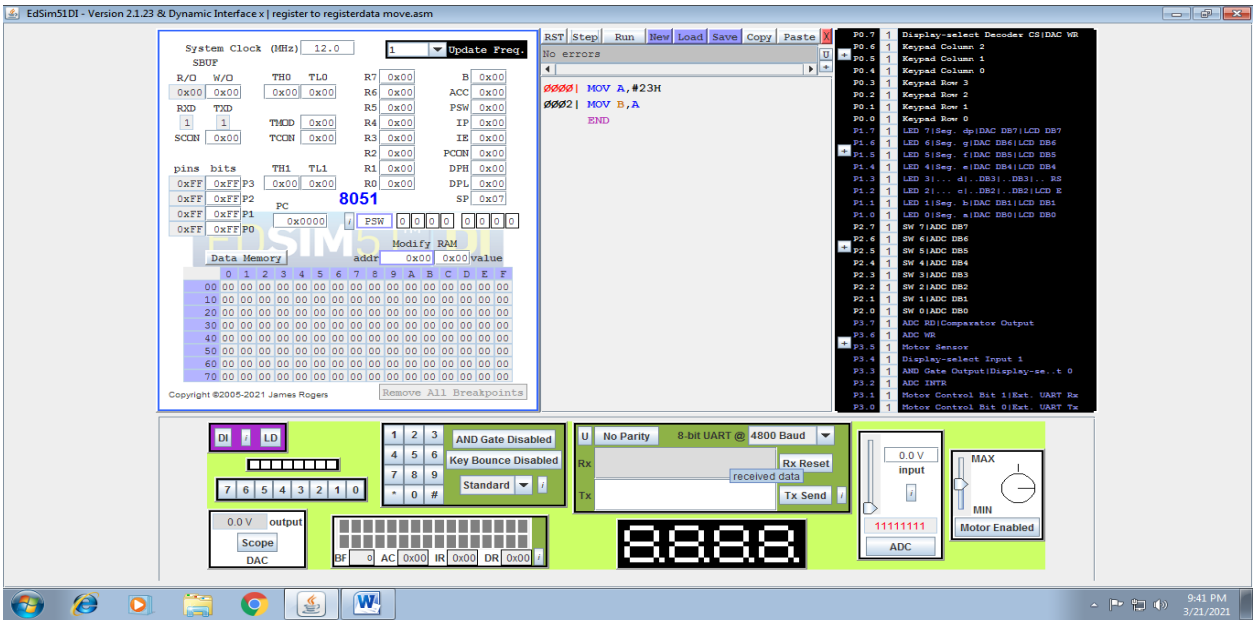
8051 PROGRAMMING

1. At first install the EdSim 51 software.
2. Then open the EdSim 51 icon.
3. Then write the program in edit place (white blank place).
4. Next choose the ‘Assemble’ option.
5. Next choose the ‘Save’ option.
6. Then select the ‘RUN’ option.
7. Then want from the run program to pause the program, then use the ‘PAUSE’ option.
8. Next want to Run the any type of previous program, then choose ‘LOAD’ option.
9. Next write another new program, then select the ‘NEW’ option.

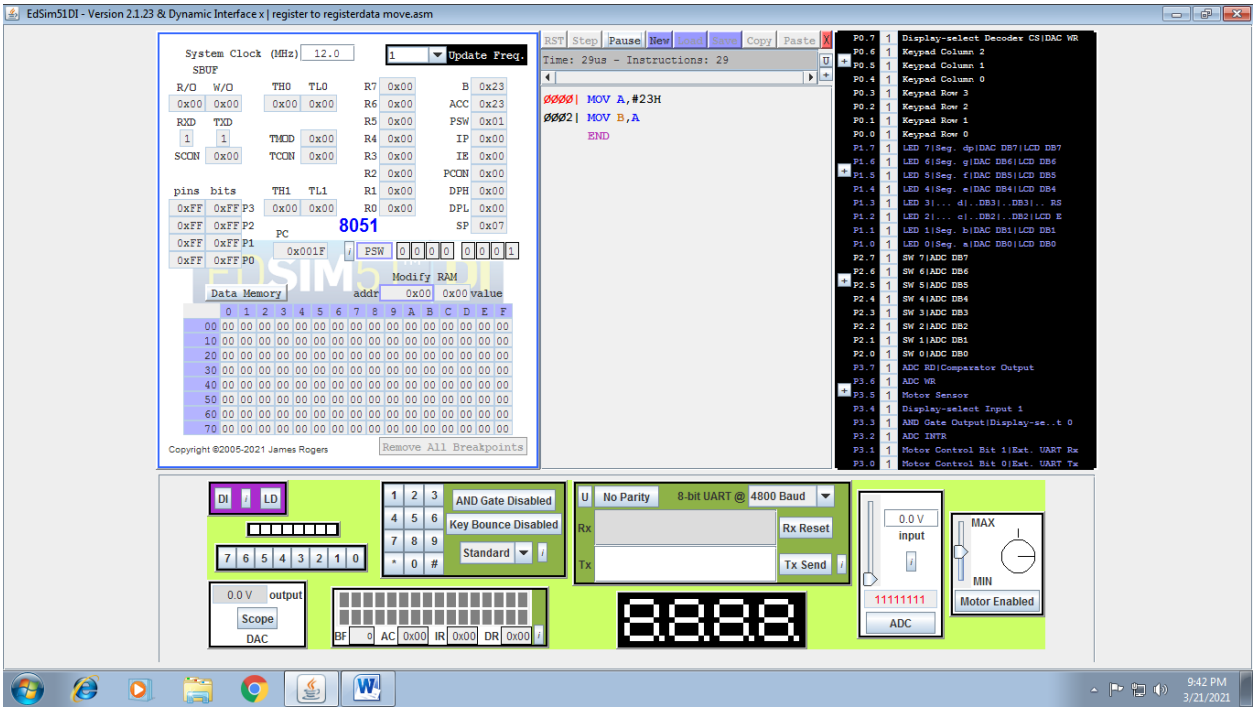
STEP 1: WRITE PROGRAM IN WHITE BLANK SPACE OR LOAD PREVIOUS PROGRAM FROM 'LOAD' OPTION.



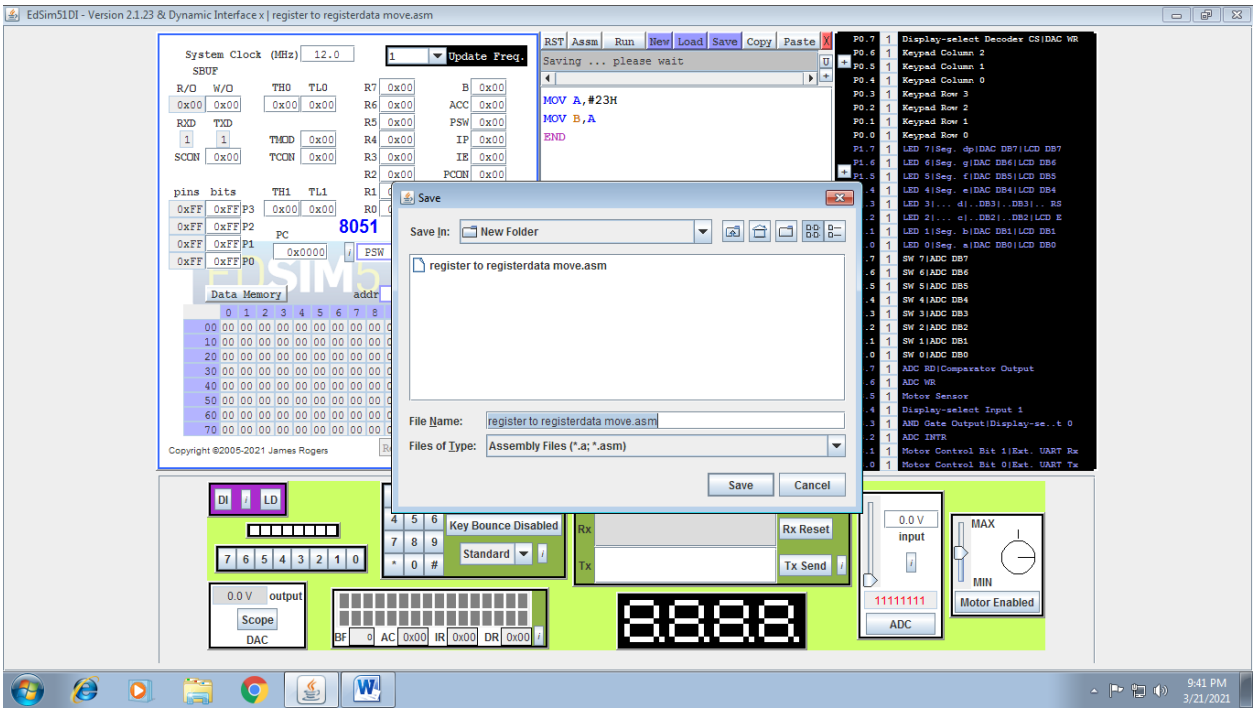
STEP 2: CLICK ON ‘ASSM’ TO ASSEMBLE THE PROGRAM



STEP 3: CLICK ON ‘RUN’ TO EXECUTE THE PROGRAM



STEP 4: CLICK ON ‘SAVE’ TO SAVE THE PROGRAM



OUTPUTS WILL BE DISPLAYED VIA REGISTER AND MEMORY LOCATION.

TO CHANGE/EDIT ANY INSTRUCTION, CLICK ON ‘ASSM’ OPTION. AFTER EDITING THE PROGRAM SHOULD BE RUN AGAIN.

TO WRITE A NEW PROGRAM CLICK ON ‘NEW’.

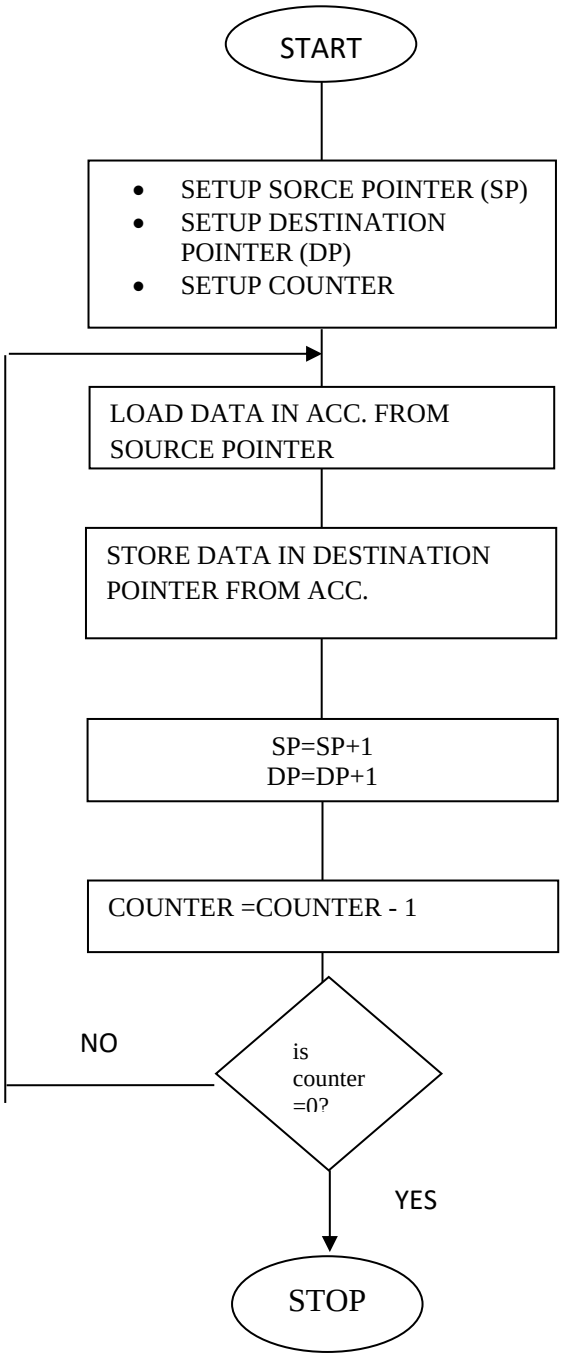
COPY WRITING PROCEDURE:

EXAMPLE 1

Q. Write an Assembly Language Program for the 8085 microprocessor to transfer the block of **sixteen data bytes** which are stored starting from memory location **8500**H into the new memory locations starting from **9600H.**

Data: 01H,02H,03H,04H,05H,06H,07H,08H,09H,0AH,0BH,0CH,0DH,0EH,0FH,10H

FLOW CHART: a block of data bytes transfer from source memory location to another memory location



PROGRAM CODE:

LEBEL	ADDRESS	MNEMONICS
	8000H	LXI B, 8500H
	8003H	LXI D, 9500H
	8006H	MVI H, 10H
LOOP	8008H	LDAX B
	8009H	LDAX D
	800AH	INX D
	800BH	INX B
	800CH	DCR H
	800DH	JNZ LOOP
	8010H	RST 1

INPUT DATA (H): **OUTPUT DATA (H):**

INPUT ADDRESS	INPUT DATA		OUTPUT ADDRESS	OUTPUT DATA
8500	01		9500	01
8501	02		9501	02
8502	03		9502	03
8503	04		9503	04
8504	05		9504	05
8505	06		9505	06
8506	07		9506	07
8507	08		9507	08
8508	09		9508	09
8509	0A		9509	0A
850A	0B		950A	0B
850B	0C		950B	0C
850C	0D		950C	0D
850D	0E		950D	0E
850E	0F		950E	0F

EXAMPLE 2

Q. Write an Assembly Language Program for the 8051 microcontroller to transfer the block of sixteen data bytes which are stored starting from memory location 40H into the new memory locations starting from 50H.
Data: 01H,02H,03H,04H,05H,06H,07H,08H,09H,0AH,0BH,0CH,0DH,0EH,0FH,10H.

PROGRAM CODE:

```
        ORG 0000H
        MOV R0, # 40H
        MOV R1, # 50H
        MOV R2, # 10H
LOOP:   MOV A, @R0
        MOV @ R1, A
        INC R0
        INC R1

        DEC R2
        JNZ LOOP
        END
```

INPUT DATA (H) : **OUTPUT DATA (H):**

INPUT ADDRESS	INPUT DATA		OUTPUT ADDRESS	OUTPUT DATA
40	01		50	01
41	02		51	02
42	03		52	03
43	04		53	04
44	05		54	05
45	06		55	06
46	07		56	07
47	08		57	08
48	09		58	09
49	0A		59	0A
4A	0B		5A	0B
4B	0C		5B	0C
4C	0D		5C	0D
4D	0E		5D	0E
4E	0F		5E	0F

8255 INTERFACING

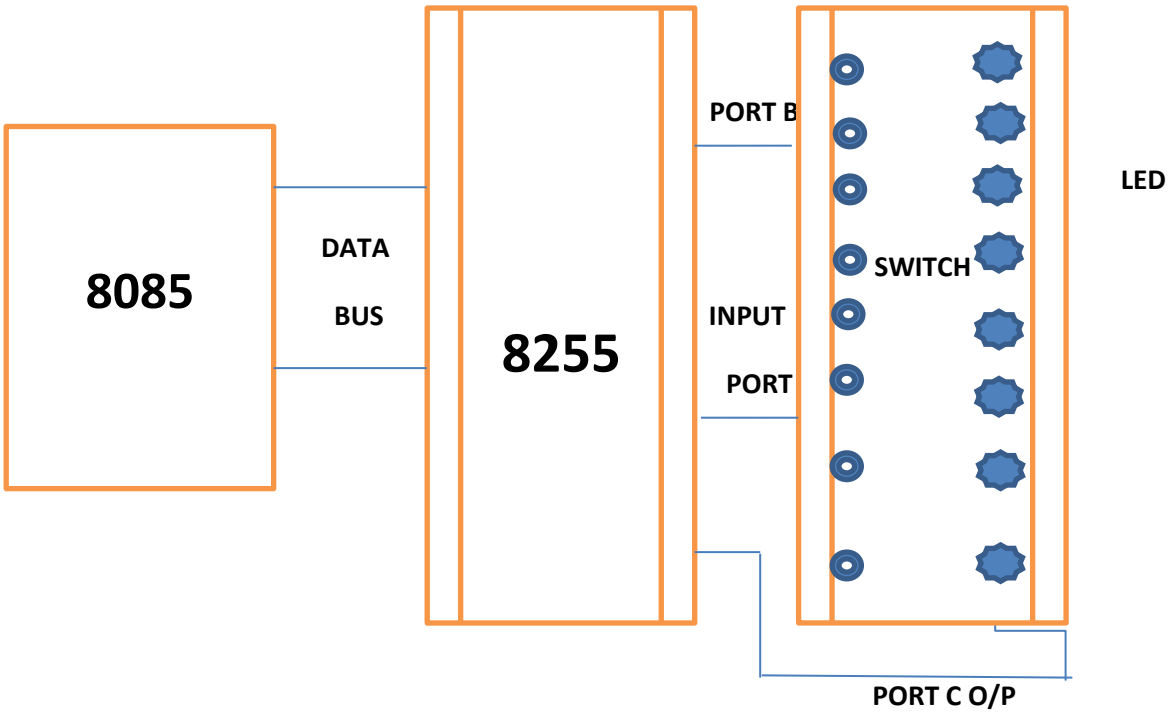
EXAMPLE:

Q. Connect SWITCH & LED module to MPT-85 through 8255 PPI to interface 8 numbers of SPDT switches as inputs and LEDs as output. Write an ALP to read any change in the switch status and the send it to LED'S for displaying the corresponding status of the switches.

PROGRAM CODE:

```
        MVI A, 82H
        OUT CONTROL WORD
LOOP:   IN   PORT B
        OUT PORT C
        CALL ADDRESS
        JMP   LOOP
        HLT
```

SWITCH & LED MODULE INTERFACING WITH 8085 USING 8255:



I/O INTERFACING WITH THE 8051

Various input and output devices can be connected to the four different ports of the 8051. The following figure (1) illustrates the entire connection of I/O devices with the 8051 available in EdSim 51DI.

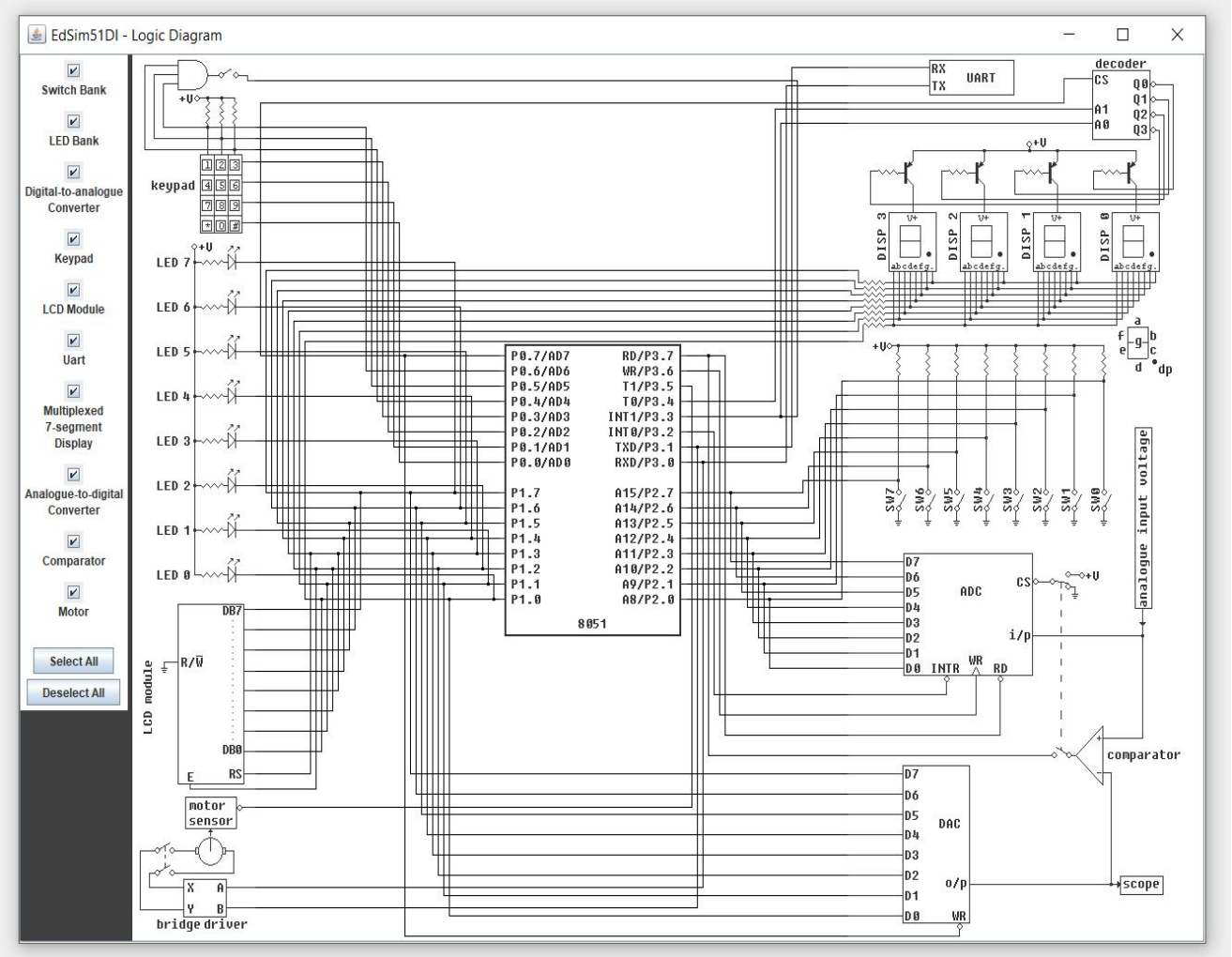


Fig. (1)

Connections of port pins of the four ports are shown in figure (2):

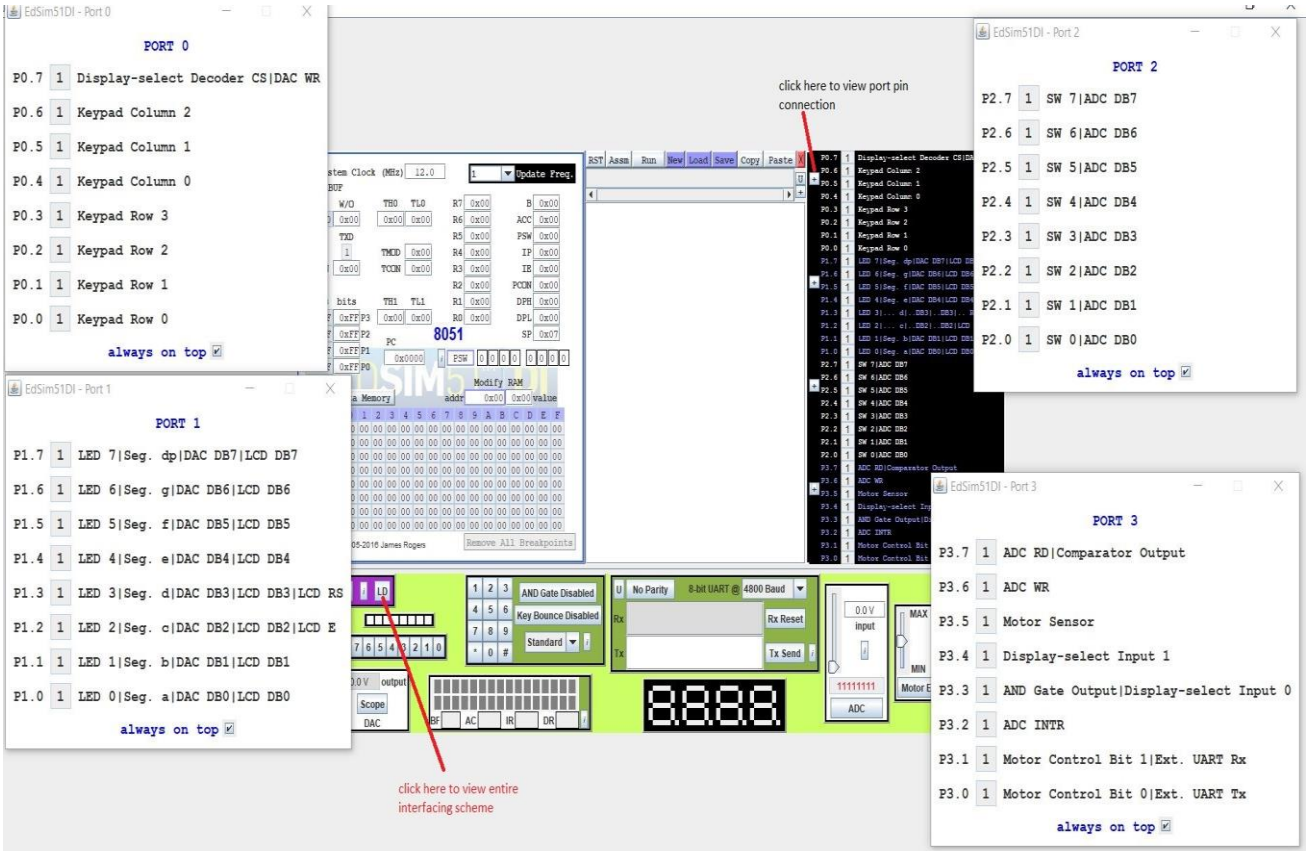


Fig. (2)

SAMPLE QUESTIONS:

1. Assume that switch 2 (bit P2.2) is used to control an outdoor light and switch 5 (bit P2.5) a light inside a building. Write assembly language program to turn on the outside light and turn off the inside one.

SOLUTION:

```
SETB C          ; CY = 1
ORL C, P2.2     ; CY = P2.2 ORed with CY
MOV P2.2, C     ; turn switch 2 on if not on
CLR C           ; CY = 0
ANL C, P2.5     ; CY = P2.5 ANDed with CY
MOV P2.5, C     ; turn switch 5 off if not off
```

The input and output of the above program is shown in figure 3(a) and (b), respectively.

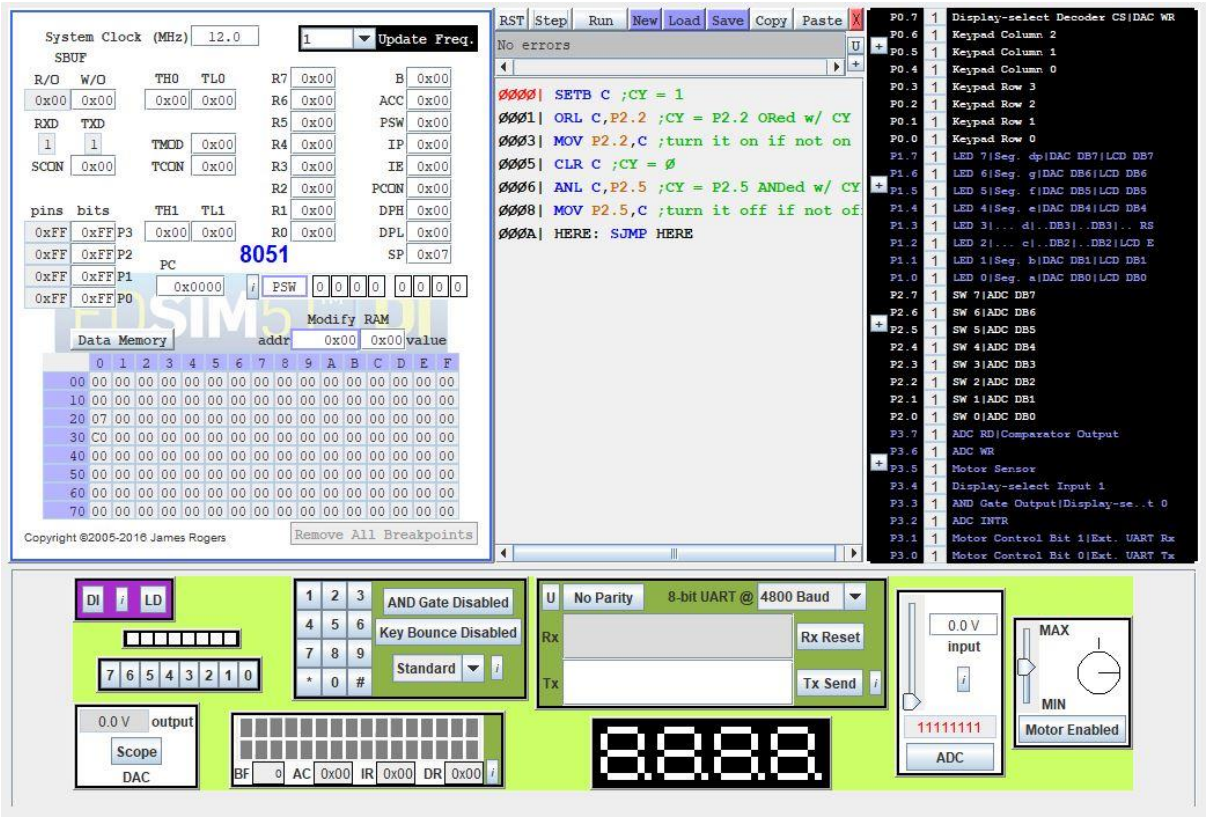

System Clock (MHz) 12.0
Update Freq.
8051
PC
Modify RAM
Data Memory
Copyright ©2005-2016 James Rogers
Remove All Breakpoints
RST Step Run New Load Save Copy Paste
No errors
ØØØØ| SETB C ;CY = 1
ØØØ1| ORL C,P2.2 ;CY = P2.2 ORed w/ CY
ØØØ3| MOV P2.2,C ;turn it on if not on
ØØØ5| CLR C ;CY = Ø
ØØØ6| ANL C,P2.5 ;CY = P2.5 ANDed w/ CY
ØØØ8| MOV P2.5,C ;turn it off if not of
ØØØA| HERE: SJMP HERE
P0.7 Display-select Decoder CS|DAC WR
P0.6 Keypad Column 2
P0.5 Keypad Column 1
P0.4 Keypad Column 0
P0.3 Keypad Row 3
P0.2 Keypad Row 2
P0.1 Keypad Row 1
P0.0 Keypad Row 0
P1.7 LED 7|Seg. dp|DAC DB7|LCD DB7
P1.6 LED 6|Seg. g|DAC DB6|LCD DB6
P1.5 LED 5|Seg. f|DAC DB5|LCD DB5
P1.4 LED 4|Seg. e|DAC DB4|LCD DB4
P1.3 LED 3|... d|..DB3|..DB3|.. RS
P1.2 LED 2|... c|..DB2|..DB2|LCD E
P1.1 LED 1|Seg. b|DAC DB1|LCD DB1
P1.0 LED 0|Seg. a|DAC DB0|LCD DB0
P2.7 SW 7|ADC DB7
P2.6 SW 6|ADC DB6
P2.5 SW 5|ADC DB5
P2.4 SW 4|ADC DB4
P2.3 SW 3|ADC DB3
P2.2 SW 2|ADC DB2
P2.1 SW 1|ADC DB1
P2.0 SW 0|ADC DB0
P3.7 ADC RD|Comparator Output
P3.6 ADC WR
P3.5 Motor Sensor
P3.4 Display-select Input 1
P3.3 AND Gate Output|Display-se..t 0
P3.2 ADC INTR
P3.1 Motor Control Bit 1|Ext. UART Rx
P3.0 Motor Control Bit 0|Ext. UART Tx
AND Gate Disabled
Key Bounce Disabled
Standard
No Parity
8-bit UART @ 4800 Baud
Rx Reset
Tx Send
0.0 V input
11111111
ADC
MAX
MIN
Motor Enabled
0.0 V output
Scope
DAC

Fig. 3(a)

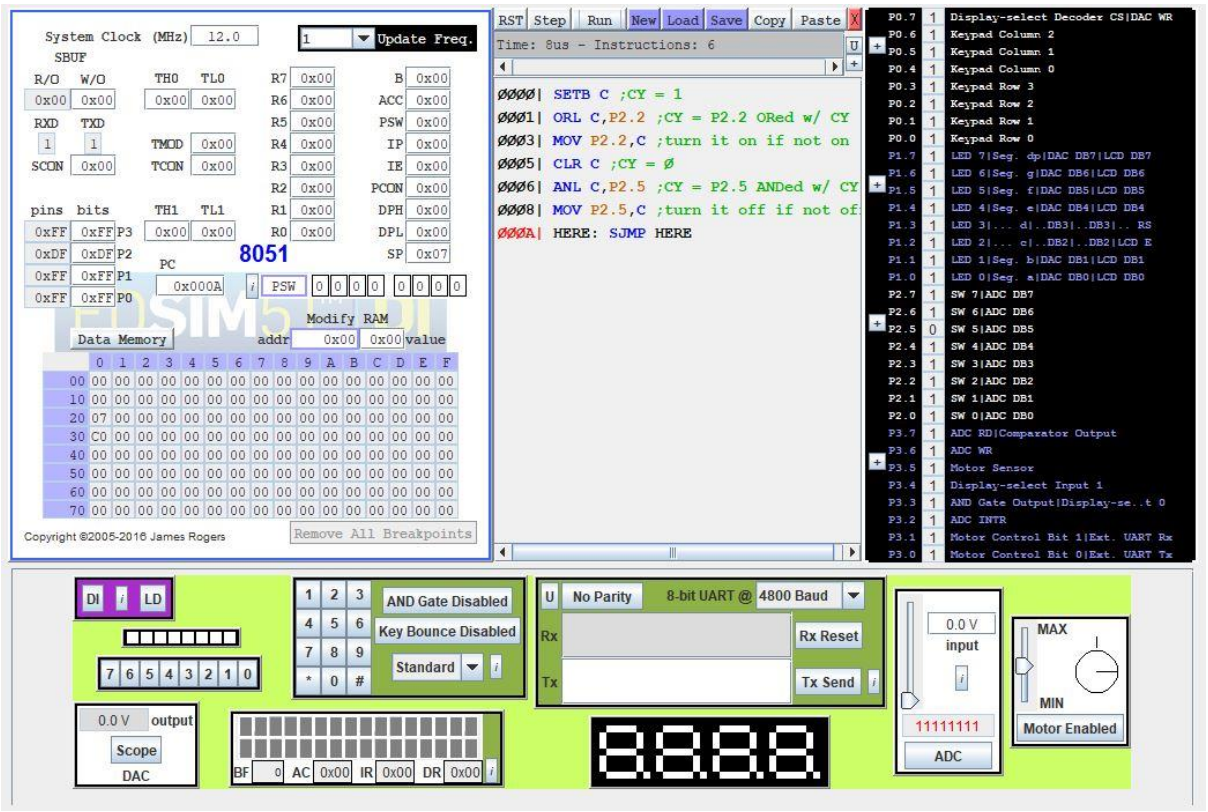

System Clock (MHz) 12.0
Update Freq.
8051
PC
Modify RAM
Data Memory
Copyright ©2005-2016 James Rogers
Remove All Breakpoints
RST Step Run New Load Save Copy Paste
Time: 8us - Instructions: 6
ØØØØ| SETB C ;CY = 1
ØØØ1| ORL C,P2.2 ;CY = P2.2 ORed w/ CY
ØØØ3| MOV P2.2,C ;turn it on if not on
ØØØ5| CLR C ;CY = Ø
ØØØ6| ANL C,P2.5 ;CY = P2.5 ANDed w/ CY
ØØØ8| MOV P2.5,C ;turn it off if not of
ØØØA| HERE: SJMP HERE
AND Gate Disabled
Key Bounce Disabled
Standard
No Parity
8-bit UART @ 4800 Baud
Rx Reset
Tx Send
0.0 V input
11111111
ADC
MAX
MIN
Motor Enabled
0.0 V output
Scope
DAC

Fig. 3(b)

2. Write an assembly language program to display a digit on the first 7-segment unit. Common anode type 7-segment display unit is used here i.e., logic 1 is required at the common terminal to enable it.

SOLUTION:

```
MOV P3, #00H          ;7-SEGMENT selection out of 4 segments
MOV P0, #80H          ; to enable the selected 7-Segment
MOV A, #0F8H          ; to display desire pattern, suppose 7
MOV P1, A
HERE:   SJMP HERE
```

The input and output of the above program is shown in figure 4(a) and (b), respectively.

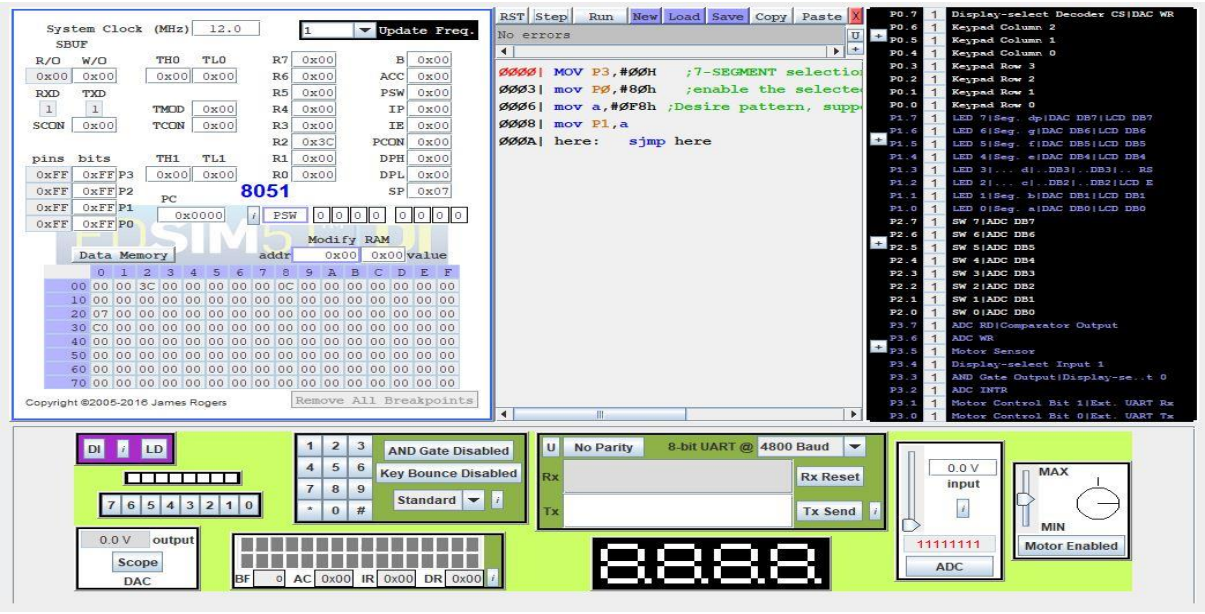


Fig.4(a)

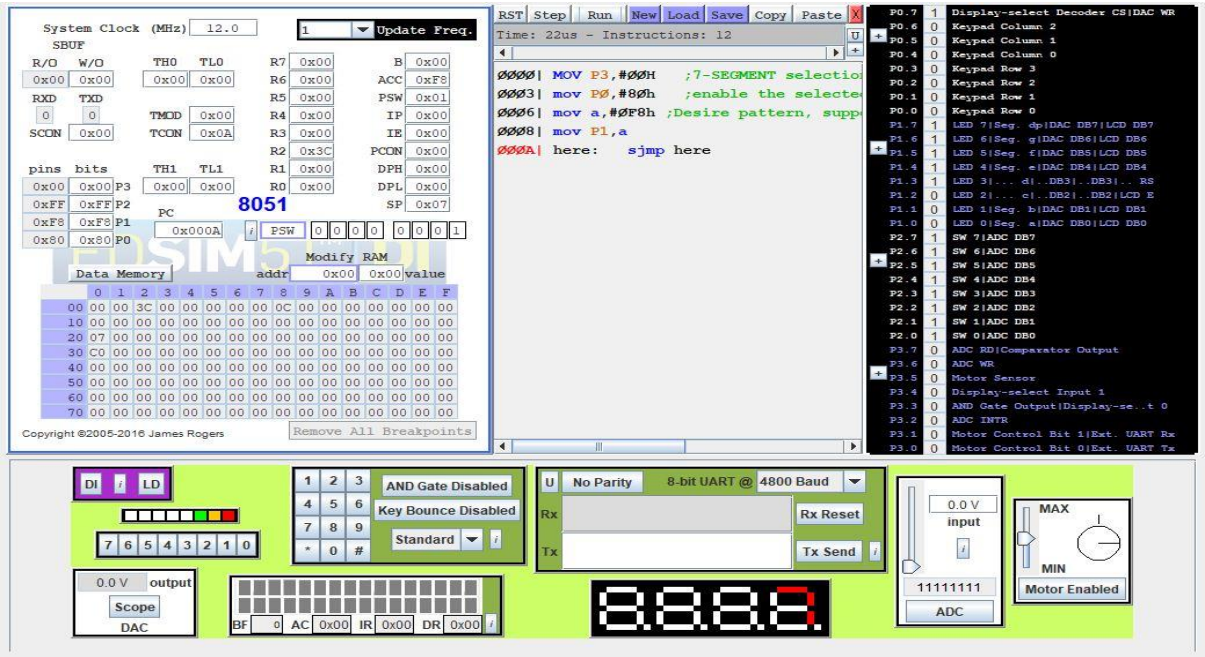


Fig.4(b)

Note:

1. Here last 7- segment unit is selected. Any other unit can be chosen by changing the combination at P3.3 & P3.4 port pins. For displaying at any 7-segment chip select line (P0.7 PIN) must be high (refer fig. (1)).

2. The following figure (5) illustrates the position and configuration of 8-bit switch module:

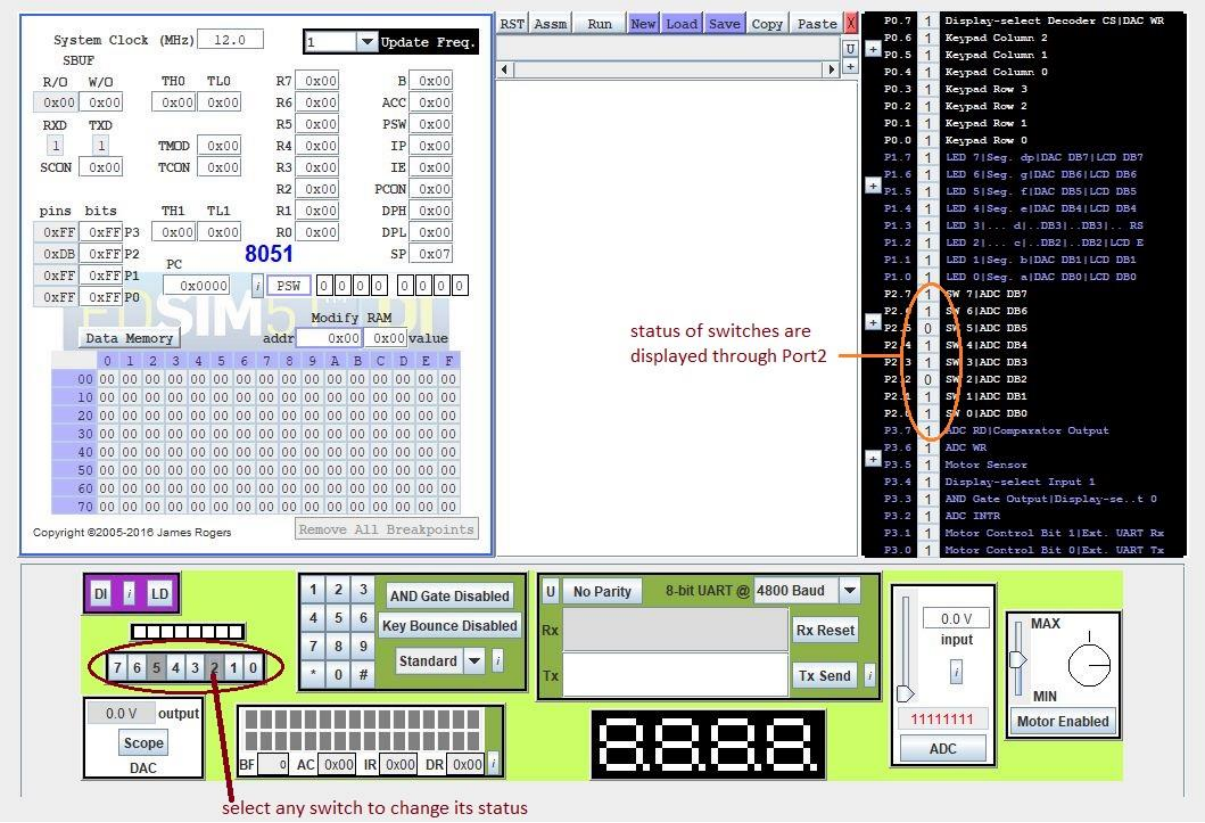


Fig.5

General Guidelines:

1. **Bring one binding Laboratory note book.**

2. **Write down date, page number, assignment number, questions, assembly language program with address and input output data with address, in ruled page.**

3. **Draw the flowchart in white page of practical copy.**

4. **Write name, roll no. and question no. as command line in the coding section at the top of each program.**

5. **Screen shoot of the programming code with input (if any) and executed code with the output (status of register, flag and memory) should be given as result for each given program.**

6. **Write your name and roll no. in right top corner of every page in your lab copy, and take snaps of those pages, make it a single pdf file along with the screenshots taken from the simulator and submit it as your assignment. the name of your file will be - roll no._your name_assignment no.**