RAMAIAH
Institute of Technology

Data Sheet

MARKS : 10

Name :	SUKRUTH. S	Branch:	
USN/Roll No. :	1MS18CS142	Sem/Sec:	4th 'B'
Subject :	computer Organisation	Subject Code:	

1] Write down the detailed procedure to assemble a system.

Ans. Step 1: Remove Side panels on case

After removing the case from the box, the panels are removed from this case with thumb screws. Our specific model's manual will have more information.

Step 2: Insert motherboard

Depending on the motherboard, case, CPU and CPU fan, we need to install the CPU cooler must be installed before or once in place.

Step 3: Check clearances

Being that the computer includes high performance components, some of them are large enough that the clearance can become an issue.

Step 4: Front Panel connections

It is time to attach the connections for the buttons, lights, USB ports and audio connections.

Step 5: Install power supply

The supply is screwed into the back panel by 4 screws.

Step 6: Power Motherboard

Motherboard power is the largest cable, so run the cable first and plug it into the board.

Step 7: Installing optical drive

The optical drive is a DVD/CD read/write combo.

Step 8: Installing the hard drives

The size and number of hard drives is dependent on our use and storage needs.

Step 9: Connect cables

We need to connect cables for hard drives and optical drives.

Step 10: Install RAM

We need to now insert the RAM. The slots are keyed as are the RAM sticks and ensure that the notch is lined up.

Step 11: Install Graphics card and exparsion cards

We need to add additional cards if the computer does not come with a graphics card integrated into motherboard

Step 12: Cable Management

Hiding cables and organising them will help in the future if we are looking for airflow.

Step 13: Final product

The assembly of brand new system is completed, it is time to fire it up and enjoy our creation.

2] Explain how troubleshooting a system helps to trace and correct the failure in the system.

Ans. Troubleshooting means error rectification by diagnosing the source or sources of the error or problem. It involves both identification and fixing of mostly software problems on failures in a system, by following procedure step by step. Troubleshooting approach requires one to be simplistic in their approach and not go all critical suddenly since

troubleshooting is applied when a system suddenly stops working, so the best approach is to start from the basics and then slowly climb up to more complex problem solving strategies.

Step 1: for troubleshooting involves to detect what exactly the problem is. Getting to the bottom of a computer issue can sometimes feel like dealing with lot of questions at a time, so we need to ask the right questions first if we want to discover the root problem quickly.

Step 2: It is important to gather more details and to eliminate variables. In many instances, what was reported as a general issue (eg: the internet is down) is actually something very particular, such as a specific website being offline. We need to ask the pertinent questions and then dig up more information from various sources such as error messages, error logs, providing screenshots, diagnostics results.

Step 3: It is important to reproduce the problem, develop hypothesis of root cause. After gathering basic background information, it's time to get hands-on with the problem.

Step 4: Attempt to find the solution based on the findings. The evidence we have already gathered should have narrowed down possible root causes and positioned to fix the issue.

By gaining in depth knowledge of a system's working and the working of individual components one can establish procedure to nail the source of problem and it's fixation

3] List out the procedure to install extra memory card to the system.

Ans. Step 1: Select the RAM that is compatible with the computer. Consult the owner's manual for your computer or motherboard to determine the correct RAM type.

Step 2: Remove the cover or access panel from your computer. Most computers will have thumb screws, Philip screws or push buttons to open the case. Locate the screws or buttons and remove the access panel or door.

Step 3: Locate the RAM slots

Most standard desktop computers will have two, four or six slots. They are often grouped together and atleast one of the available slots will already contain a memory module, as you must have RAM fitted to your computer for it to boot. In newer machines the RAM slots will be coloured. It is best to fill each colour first rather than skipping the ram.

Step 4: Insert the RAM

Apply the necessary pressure and keep pushing until you hear a click or the holding tabs on either side snugly into the indentations in either side of the module

RAMAIAH
Institute of Technology

Data Sheet

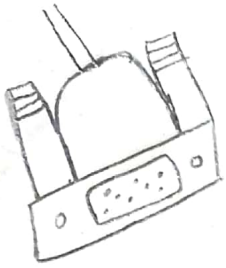
MARKS :

Name :	SUKRUTH-S	Branch:	
USN/Roll No. :	1MS18CS142	Sem/Sec:	
Subject :		Subject Code:	

4] With a neat diagram, explain different cable used to connect functional units in a system.

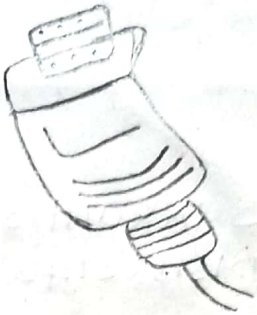
Ans.

1. VGA Cable: Also known as D-sub cable, analog video cable. Connect one end to computer monitor, television. Connect the other end to: VGA port on computer.
2. DVI Cable: Connect one end to computer monitor. Connect other end to DVI port on computer.
3. HDMI Cable: Connect one end to computer, monitor, television. Connect other end to HDMI port on computer.
4. PS/2 Cable: Connect one end to PS/2 keyboard, PS/2 mouse. Connect other end to PS/2 ports on computer. Purple PS/2 port: keyboard. Green PS/2 port: mouse.
5. Ethernet Cable: Connect one end to router, network switch. Connect other end to ethernet port on computer.
6. USB Cable: Connect one end to USB device. Connect other end to USB ports on computer.
7. Computer Power (AC) Connect one end to AC power socket. Connect other end to power supply unit, computer monitor.

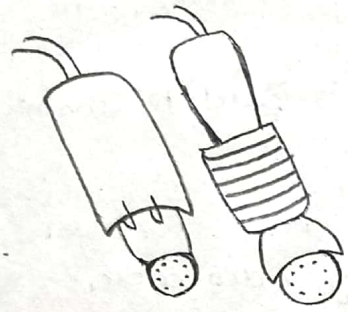


1. VGA Cable

2. DVI Cable

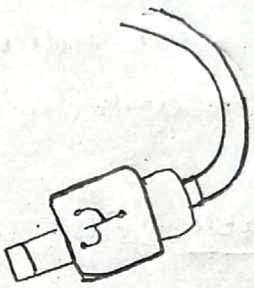


3. HDMI Cable

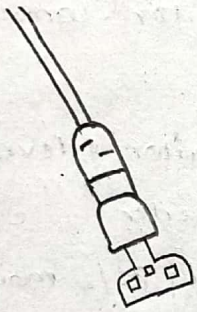


4. PS/2 Cable

5. Ethernet Cable



6. USB Cable



7. Computer Power Cord

5] Discuss the safety precautions one should take while removing the components of a system.

Ans.

1. Fully shutdown and unplug the computer before you make any attempts to dissamble the tower
2. Take off any metal objects on your arms or fingers such as bracelets, rings or watches. Even if your unit is unplugged, there may still be some remaining electric charge.

3. Make sure your hands are completely dry to avoid damaging any mechanical parts as well as to avoid electrocution.
4. Work in a cool area to avoid perspiration for the same person as sun in the previous number
5. Before touching any part within the tower, put your hands against another metal surface to remove static charge, which may damage sensitive devices.
6. Prepare a place to keep any screws you may remove. A container or piece of paper may remove which labels for each part is ideal to avoid confusion between the similar looking screws.
7. Handle all parts with care. Place each piece you remove carefully down onto a stable surface.
8. If a component does not come out easily, do not remove it forcefully.
9. Be careful when handling the motherboard.
10. Never attempt to remove the power source, a box attached to the side or bottom of the unit to which all cables are connected.
11. When removing any cables, wires or ribbons, make sure to grasp the wire at the base or head to keep it from breaking.
12. Be careful not to drop any small parts into unreachable areas such as into the computer fan or disc drive.
13. Take note that the three most damaging things to a computer are moisture, shock and dust.

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of CSE

Tutorial -II

Programme: B.E
Course: Computer Organization Course Code: CS45 Term: Jan to May 2020

Name: SUKRUTH.S	**Marks:** 10 /10	**Date:** 28/01/2020
USN: 1MS18CS142	**Signature of the Faculty:**	

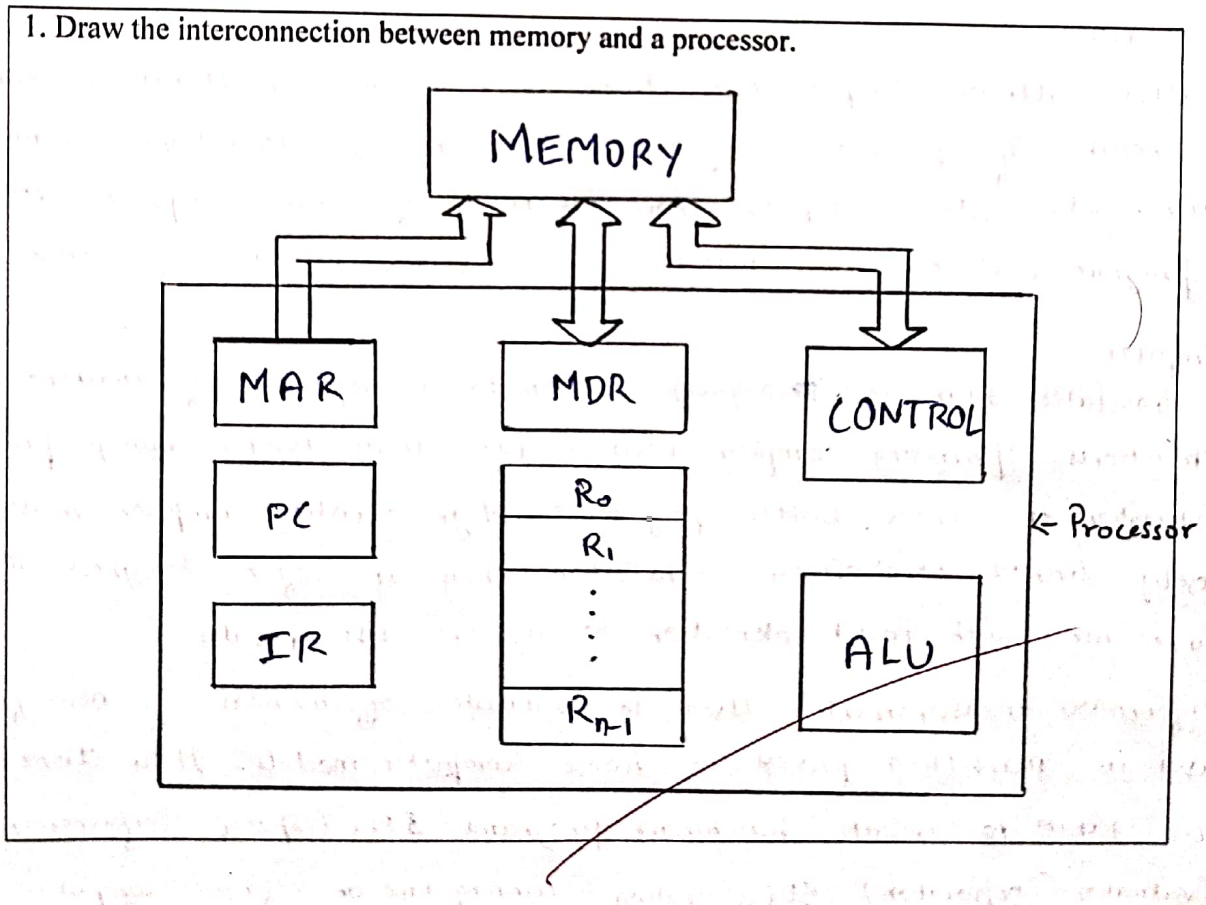
Activity II: Demonstrating Datapath and instruction execution stages using MarieSim Simulator

Objective: To simulate inter communication between CPU and memory.

Simulator Description: MarieSim is a computer architecture simulator based on the MARIE architecture. It provides users with interactive tools and simulations to help them deepen their understanding of the operation of a simple computer. One can observe how assembly language statements affect the registers and memory of a computer system.

Activity to be performed by students:

1. Draw the interconnection between memory and a processor.



2. List out the steps required to execute an instruction.

3. Write and execute assembly language program to compute

i) f=(g+h)*(i+y)

ii) $d=b^2-4ac$

4. Describe the factors affecting the performance of a processor

2. 6 steps : (i) Fetch instruction
(ii) Decode instruction
(iii) Perform ALU operation
(iv) Access memory
(v) Update register file
(vi) Update program counter

3. (i)
```
LOAD  G
ADD   H
STORE A
LOAD  I
ADD   Y
STORE B
LOOP, LOAD A
      ADD  B
      STORE F
      HALT

G, DEC 5
H, DEC 4
I, DEC 3
Y, DEC 2
A, DEC 0
B, DEC 0
```

3. Results and Snapshots:

4.(i) Processor clock: Processor circuits are controlled by a clock - it is a timing signal called clock. Clock defines regular time intervals - clock cycles. To execute a machine instruction, the processor divides the action to be performed into a series of basic steps, each completed in 1 clock cycle. Length P of one clock cycle is important for processor performance, whose inverse is the clock rate, $R = 1/P$ cycles/sec calculated in Hertz.

(ii) Basic Performance equation $T = \frac{N \times S}{R}$, where T = Processor time
N = No. of machine language instructions
R = clock rate
S = No. of basic steps to execute instruction

For high performance, reduce T by reducing N & S and increasing R.

(iii) Pipelining and Superscalar operation: Pipelining is the overlapping execution of several instructions. This reduces number of clock cycles, we can achieve a higher degree of concurrence. Parallel paths are created. Execution of several instructions in every clock cycle is called super scalar execution.

(iv) Clock rate: 2 possibilities to increase clock rate R is improving the Integrated Circuit technology which makes logic circuits faster, which reduces the time needed to complete a basic step. Reducing the amount of processing done in one basic step also makes it possible to reduce the clock period P.

(v) CISC and RISC

RISC allows simple instructions which require small no. of basic steps to execute. If program will have more no. of instructions i.e., N = max, S = min. CISC are complex, more number of basic steps to execute. A program will have less number of instructions i.e., N = min, S = max.

(vi) Compiler

Translates high level language program to a sequence of machine instructions. Optimising compiler reduces N×S, may reorg rearrange program instructions to achieve better performance. High quality compiler must be closely linked to processor architecture, they are often designed at the same time, with much interaction to achieve test results.

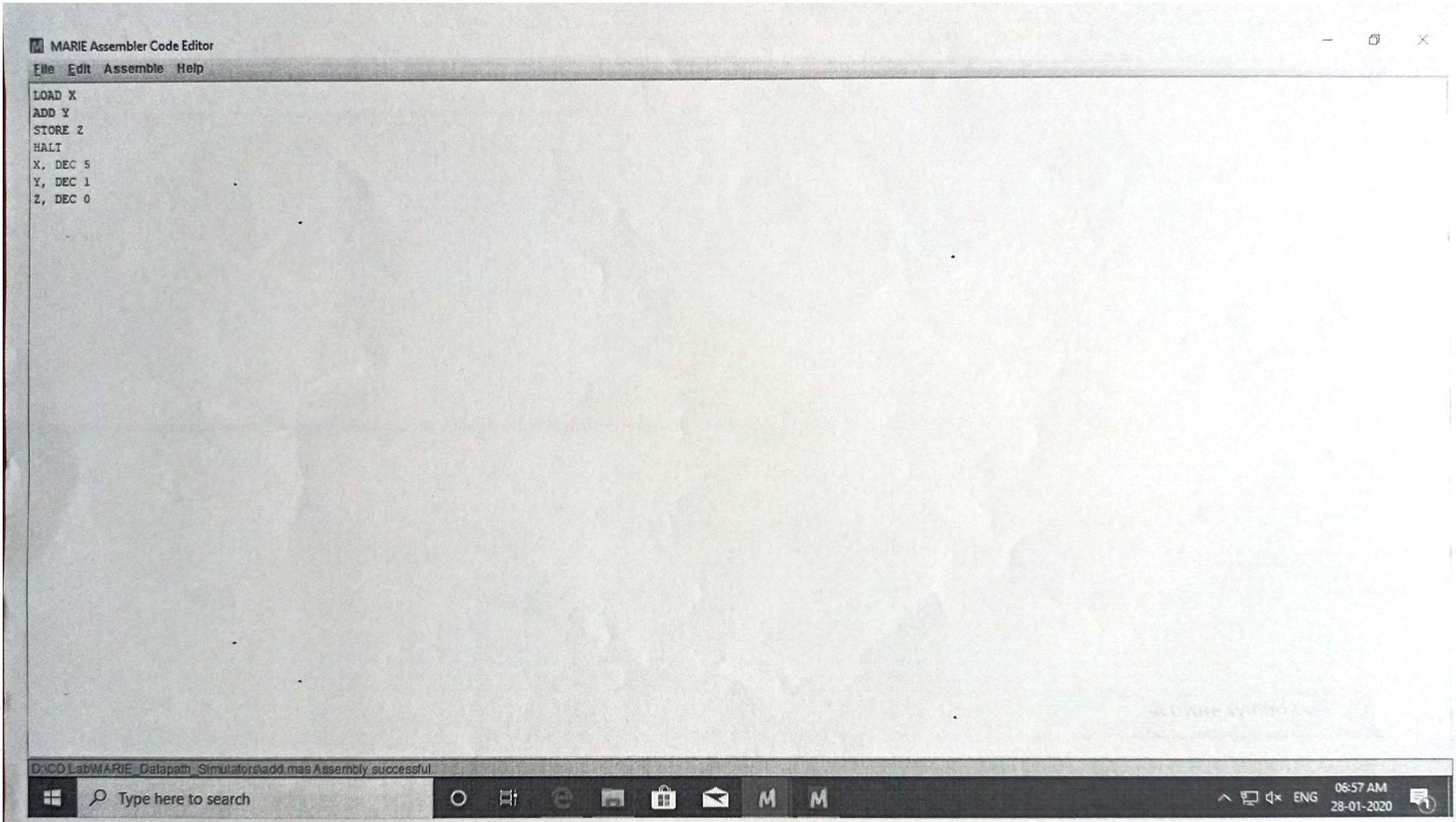
(vii) Performance measurement: Used to evaluate effectiveness of new features, used in marketing process, to choose computer models. It is done using time taken to execute benchmark programs. SPEC (Specific Performance Evaluation Corporation) SPEC rating = $\frac{\text{running time on reference computer}}{\text{running time on computer under test}}$

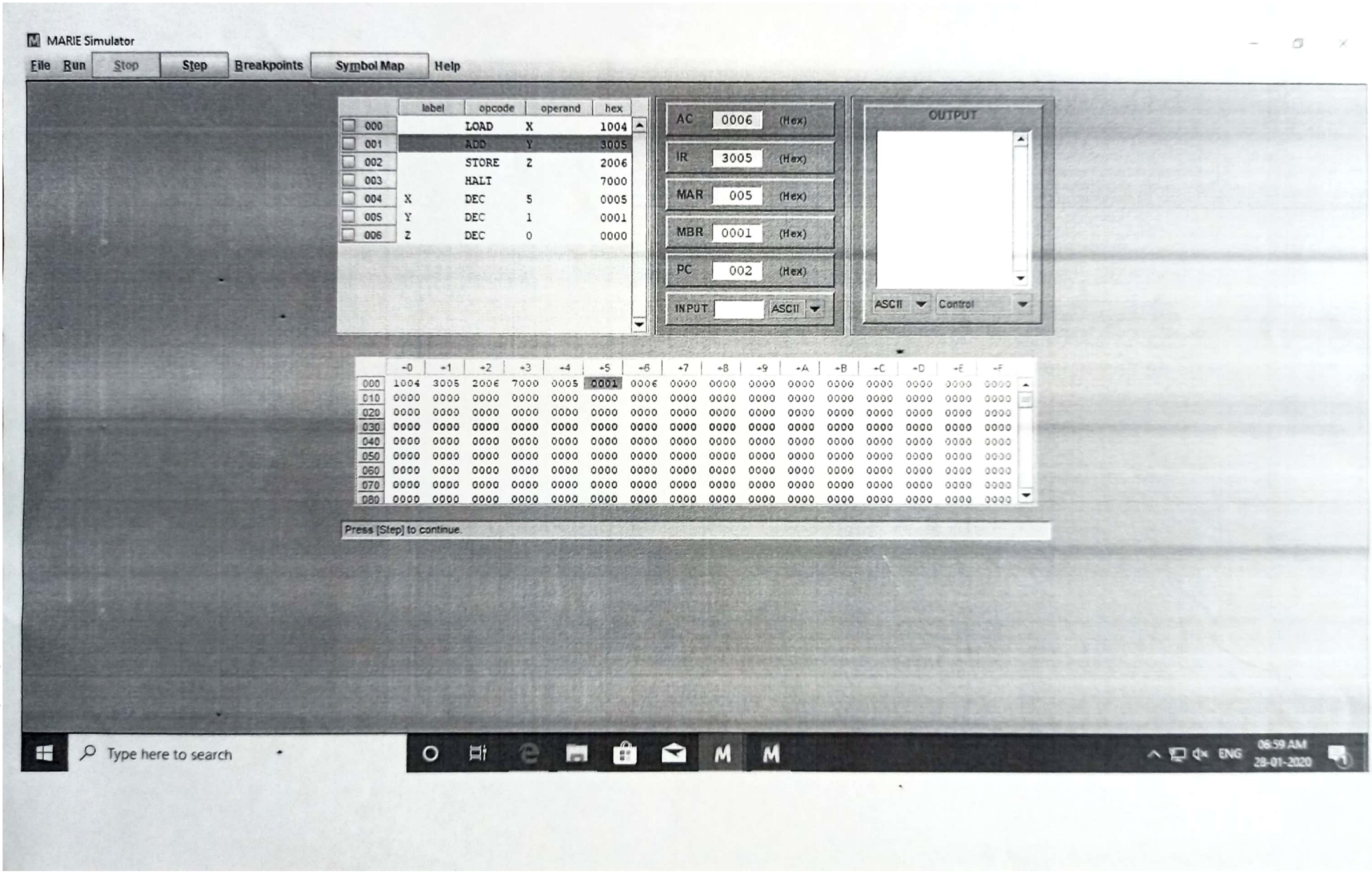
$$\text{SPEC rating} = \left(\prod_{i=1}^{n} \text{SPEC}_i\right)$$

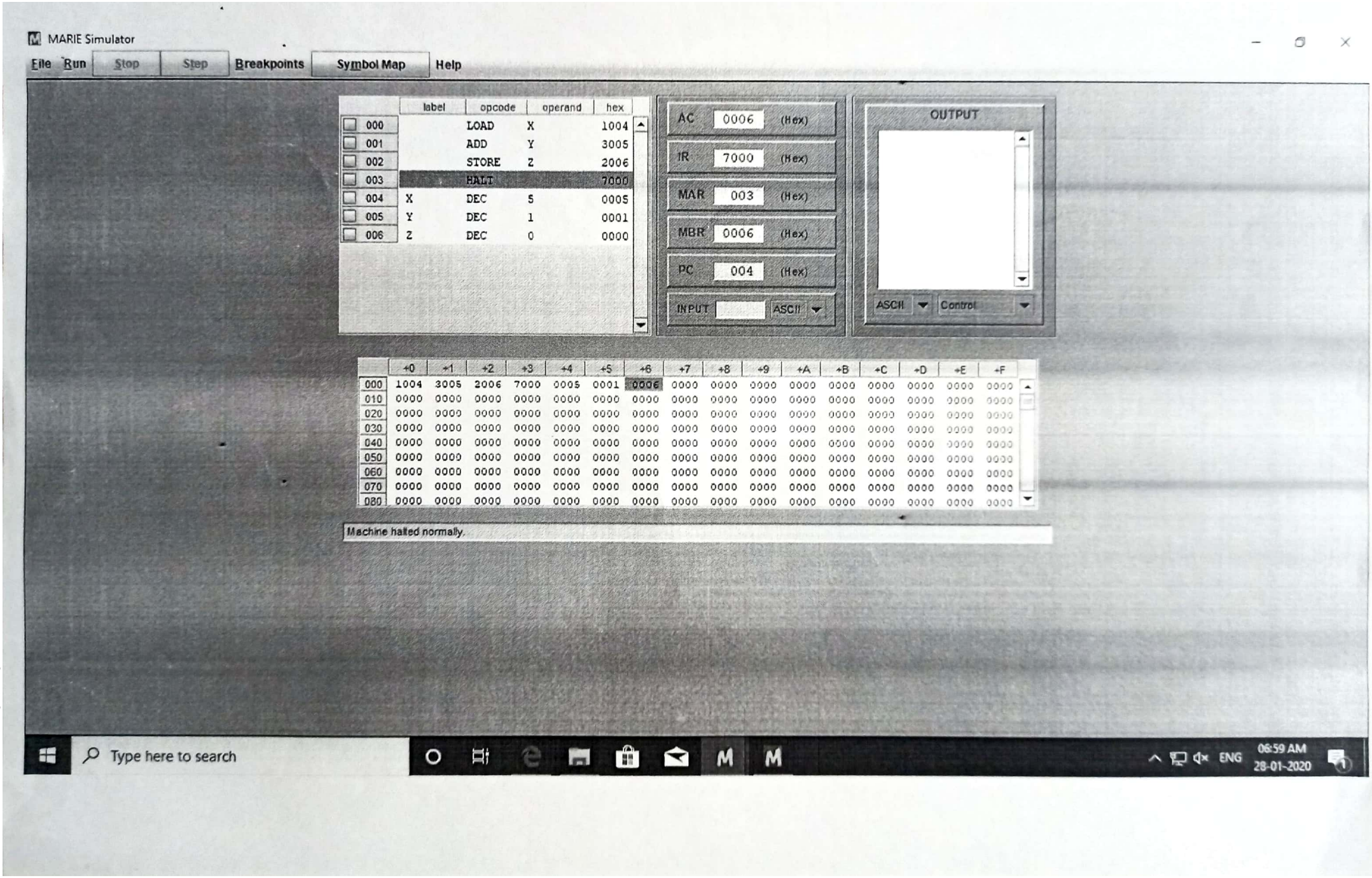
3] ii. $d = b^2 - 4ac$

```
        LOAD  B
        STORE D
        LOAD  B
        ADD   X
        STORE X
        SUB   one
        STORE 0
        jump  FIRST

Second  LOAD
        ADD   Y
        STORE Y
        LOAD  C
        SUBT  one
        STORE C
        jump  second

third   LOAD  four
        ADD   Z
        STORE Z
        LOAD  Y
        SUBT  one
        STORE Y
        jump  third
```

```
        LOAD  D
        ADD   X
        SUBT  Z
        OUTPUT
        HALT

A, DEC  6
B, DEC  3
C, DEC  1
D, DEC  0
X, DEC  0
Y, DEC  0
Z, DEC  0
D, DEC  0
one, DEC 1
four, DEC 4
```







Assembly Listing for add.mas

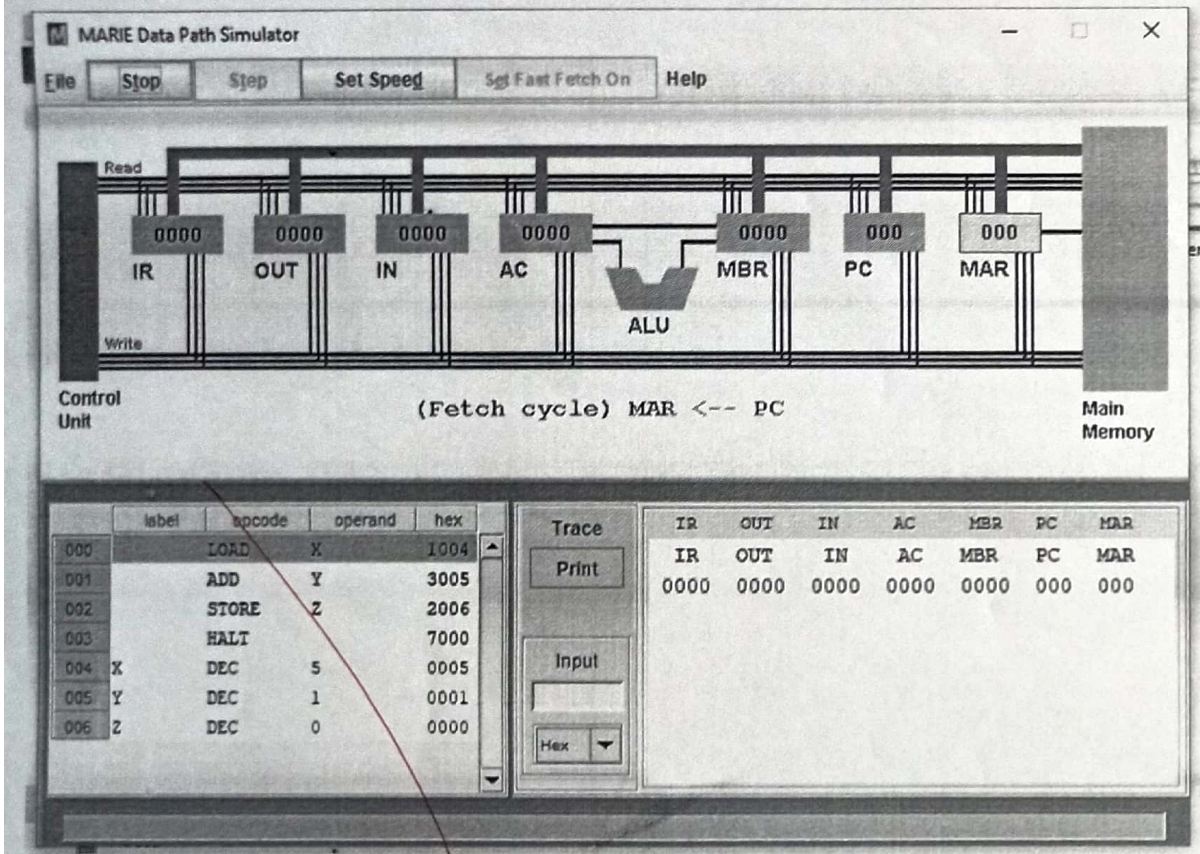
```
    Assembly listing for: add.mas
              Assembled: Tue Jan 28 06:57:27 IST 2020

000 1004 |         LOAD X
001 3005 |         ADD Y
002 2006 |         STORE Z
003 7000 |         HALT
004 0005 |  X      DEC 5
005 0001 |  Y      DEC 1
006 0000 |  Z      DEC 0

Assembly successful.

        SYMBOL TABLE
        ------------------------------------------------
         Symbol | Defined | References
        --------+---------+--------------------------------
         X      |   004   | 000
         Y      |   005   | 001
         Z      |   006   | 002
        ------------------------------------------------
```

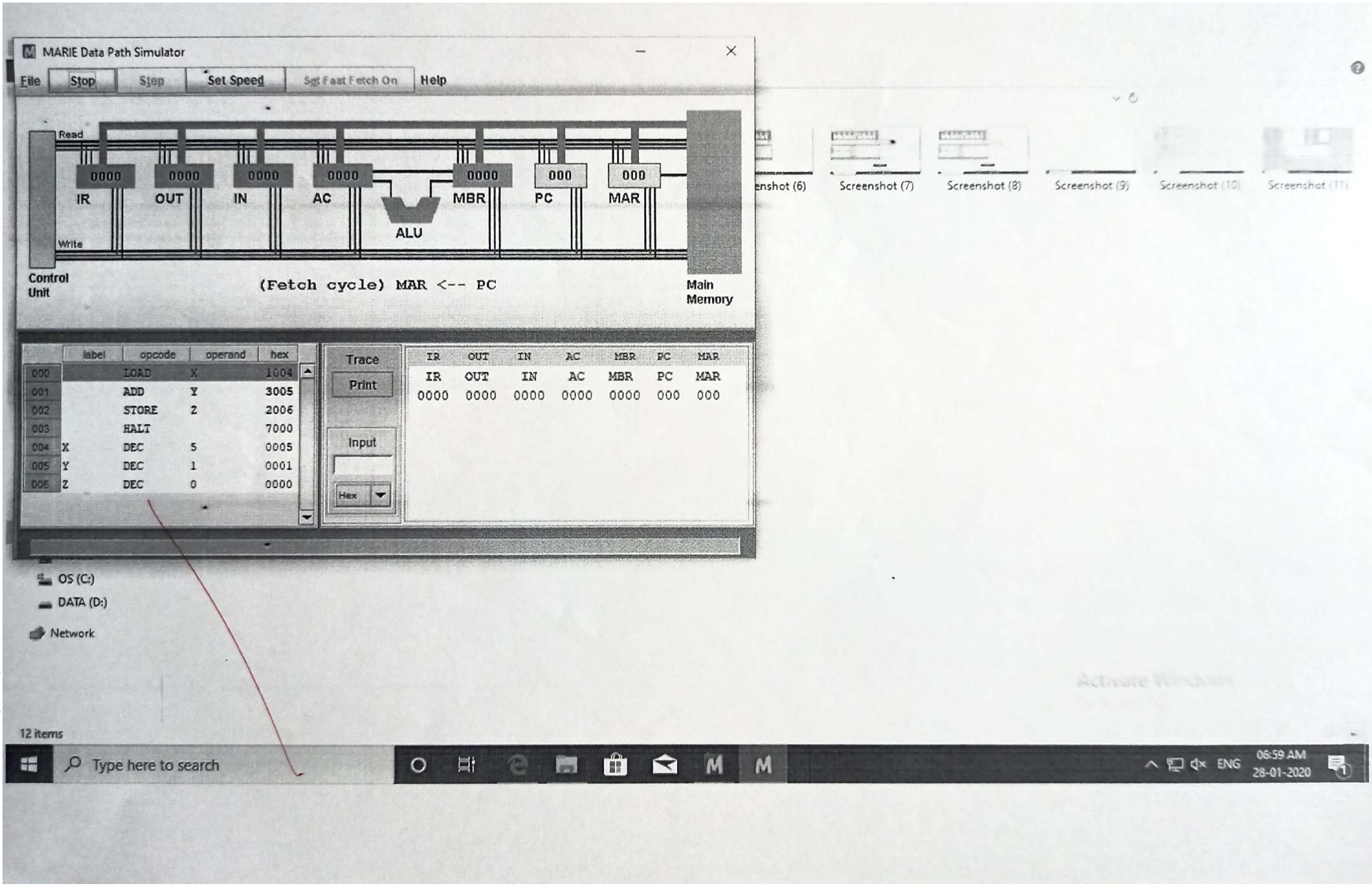
Print

Close

Type here to search

ENG 06:58 AM 28-01-2020





Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of CSE

Tutorial -III

Programme: B.E Term: Jan to May 2020
Course: Computer Organization Course Code: CS45

Name: SUKRUTH.S	**Marks:** 10 /10	**Date:** 04/02/2020
USN: 1MS18CS142	**Signature of the Faculty:**	

Objective: To simulate ARM Instruction set using ARMsim simulator.

Simulator Used: ARMSim 1.91 is a desktop application running in a Windows environment. It allows users to simulate the execution of ARM assembly language programs on a system based on the ARM7TDMI processor.

ARM enables the users both to debug ARM assembly programs and to monitor the state of the system while a program executes.

Activity to be performed by students:

1) Write an ARM program to perform basic arithmetic operations.
2) Write an ARM program to demonstrate the working of load and store instructions.
3) Write an ARM program to evaluate expression f=(g+h)-(i+j)
4) Write an ARM program to find the sum of all elements of an array.
5) Write an ARM program to find the factorial of a number.

Programs and the snapshots:

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	SUKRUTH·S	Branch:	C·S·E
USN/Roll No. :	1MS18CS142	Sem/Sec:	IV B
Subject :	Computer Organisation	Subject Code:	

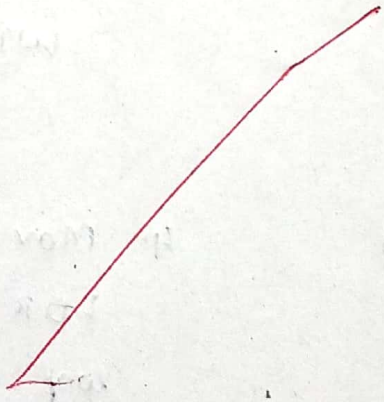
1.
```
MOV   R5,#10
MOV   R6,#20
ADD   R7,R6,R5
MUL   R8,R5,R6
SUB   R9,R6,R5
SWI   0x11
```

2.
```
MOV   R1,#0x00000010
MOV   R2,#4
MOV   R4,#20
STR   R4,[R1,R2]
LDR   R6,[R1,R2]
SWI   0x11
```

3.
```
MOV   R6,#30
MOV   R7,#40
MOV   R8,#10
MOV   R9,#20
MOV   R3,#0
MOV   R5,#0x00000050
ADD   R1,R6,R7
ADD   R2,R8,R9
SUB   R1,R1,R2
STR   R1,[R5,R3]
SWI   0x11
```

4.
```
MOV   R0,#5
LDR   R1,=array
loop  LDR R2,[R1],#4
      ADD R3,R3,R2
      SUB R0,R0,#1
      CMP R0,#0
      BNE loop
```

5.
```
MOV   R0, #3
MOV   R1, R0
MOV   R2, #1
MOV   R3, #1
fact:
      MUL   R2, R1, R2
      SUB   R1, R1, R3
      CMP   R1, #1
      BGE   fact
      SWI   0X11
```



ADD - Notepad

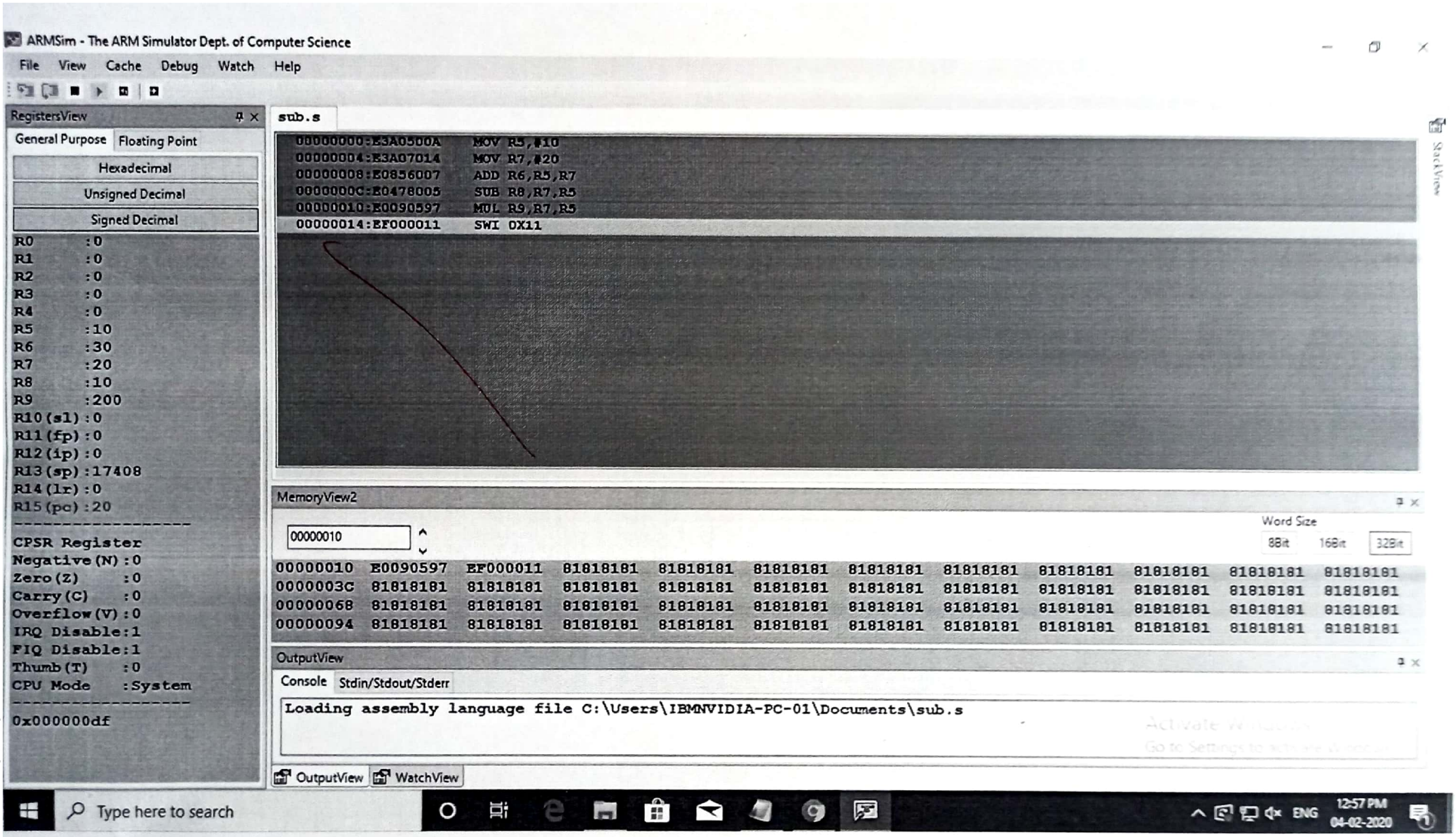
File Edit Format View Help

```
MOV R5,#10
MOV R7,#20
ADD R6,R5,R7
SUB R8,R7,R5
MUL R9,R7,R5
SWI 0X11
```


Ln 5, Col 13 100% Windows (CRLF) UTF-8

Type here to search

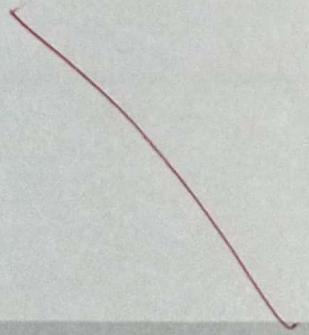
ENG 12:55 PM 04-02-2020



p2 - Notepad

File Edit Format View Help

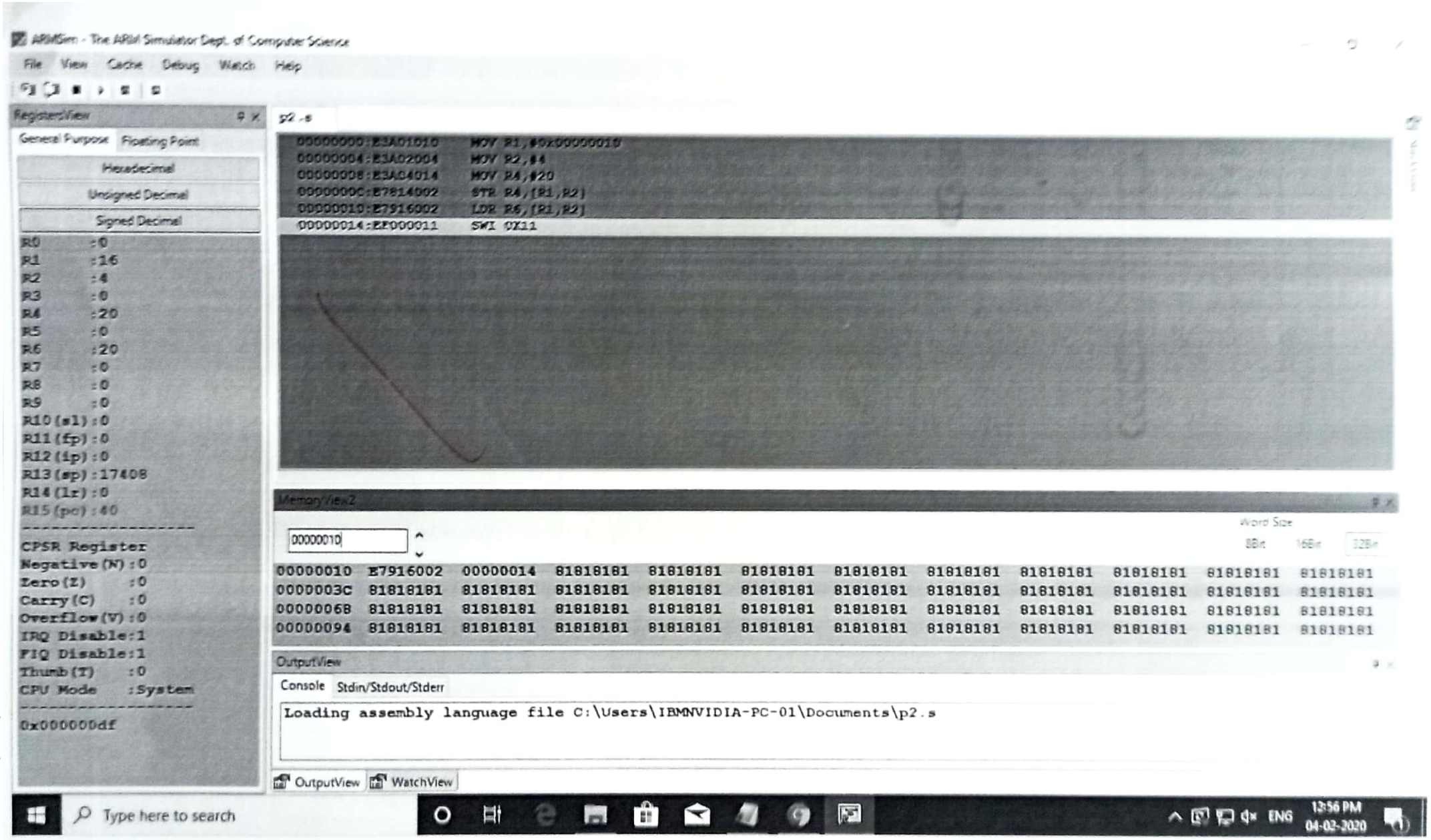
```
MOV R1,#0x00000010
MOV R2,#4
MOV R4,#20
STR R4,[R1,R2]
LDR R6,[R1,R2]
SWI 0X11
```



Ln 1, Col 1 100% Windows (CRLF) UTF-8

Type here to search

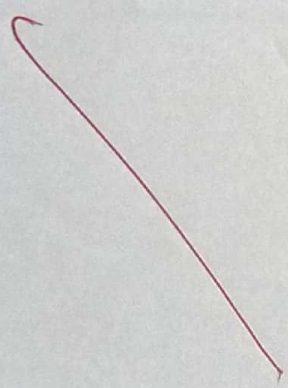
ENG 12:55 PM 04-02-2020



Untitled - Notepad

File Edit Format View Help

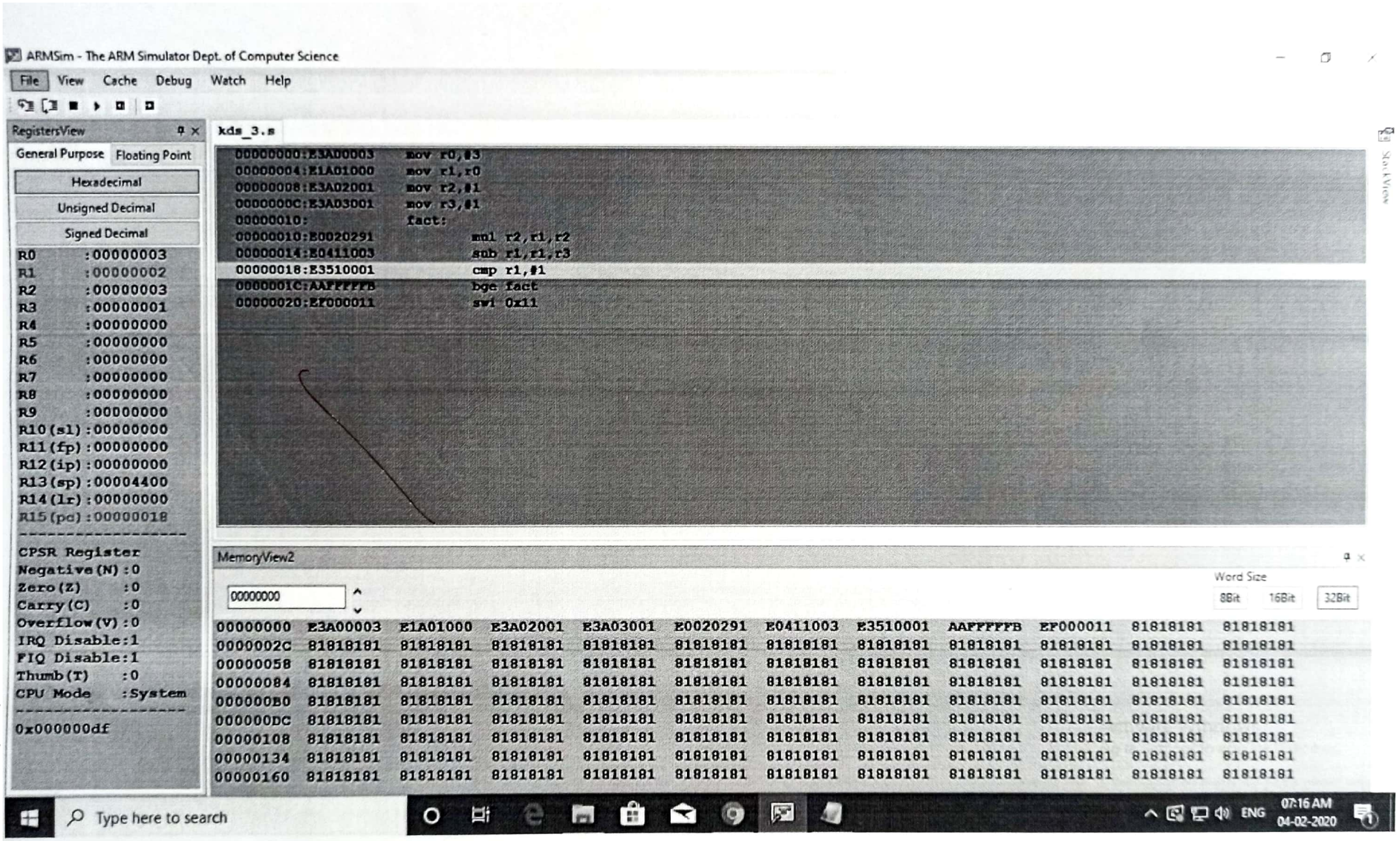
```
mov r0,#5
mov r1,r0
mov r2,#1
fact:
        mul r2,r2,r1
        sub r1,r1,#1
        cmp r1,#0
        bge fact
swi 0x11
```



Windows (CRLF) Ln 9, Col 9 100%

Type here to search

ENG 8:03 PM 2/10/2020



Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Department of CSE

Programme: B.E | **Term: Jan to May 2019**
Course: Computer Organization | **Course Code: CS45**

Activity IV: Executing ARM programs using ARMsim simulator.

Name: SUKRUTH.S	Marks: /10	Date: 11/02/19
USN: 1MS18CS142	Signature of the Faculty:	

Objective: To simulate ARM Instruction set using ARMsim simulator.

Simulator Used: ARMSim 1.91 is a desktop application running in a Windows environment. It allows users to simulate the execution of ARM assembly language programs on a system based on the ARM7TDMI processor.

ARM enables the users both to debug ARM assembly programs and to monitor the state of the system while a program executes.

Activity to be performed by students:

1) Write an ARM program to generate Fibonacci Series.
2) Write an ARM to search an element in an array and print Y if found and print N if not found.
3) Write an ARM program to find the length of a string and copying one string to another.

1)
```
MOV   R0, #0
MOV   R1, #1
MOV   R2, #6
MOV   R4, #0X00001000
MOV   R5, #0
loop: ADD   R3, R0, R1
      MOV   R0, R1
      MOV   R1, R3
      STR   R3, [R4, R5]
      ADD   R5, R2, #1
```

Results/Conclusions and Snapshots: Take the snap shot of registers file and memory view

```
        SUB   R2, R2, #1
        CMP   R2, #2
        BGT   loop
    SWI   0x11
```

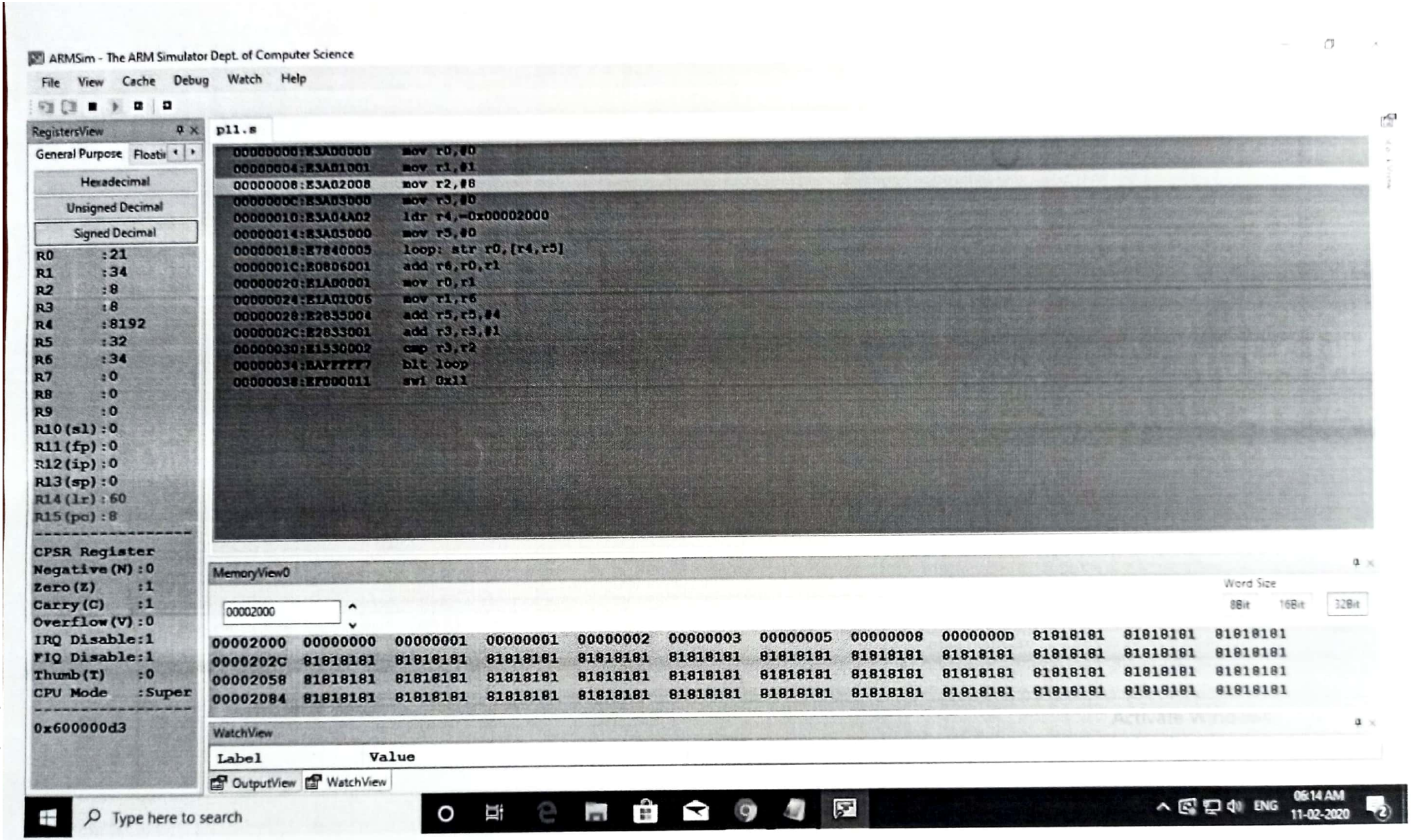
2]

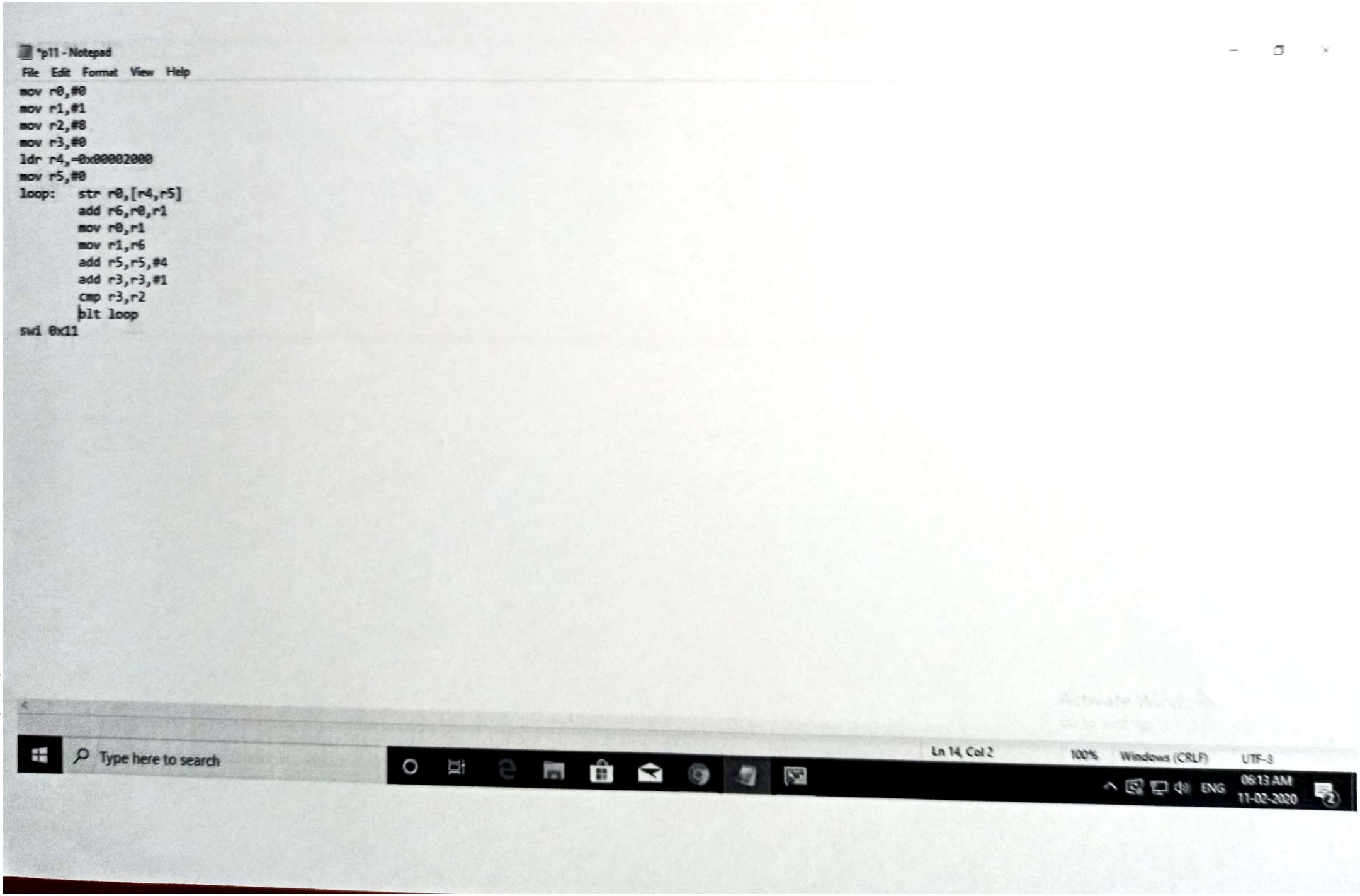
```
mov   R0, #0x00001000
mov   R1, #0
mov   R2, #5
mov   R5, #1
mov   R6, #0
mov   R4, #9
loop: ADD   R5, R5, #2
      STR   R5, [R0, R1]
      ADD   R1, R1, #4
      SUB   R2, R2, #1
      CMP   R2, #0
      BGT   loop
mov   R0, #0x00001000
mov   R1, #0
mov   R2, #5
search:
      LDR   R4, [R0, R1]
      ADD   R1, R1, #4
      CMP   R4, R6
      BEQ   Result
      SUB   R2, R2, #1
      CMP   R2, #0
      BGT   search
```

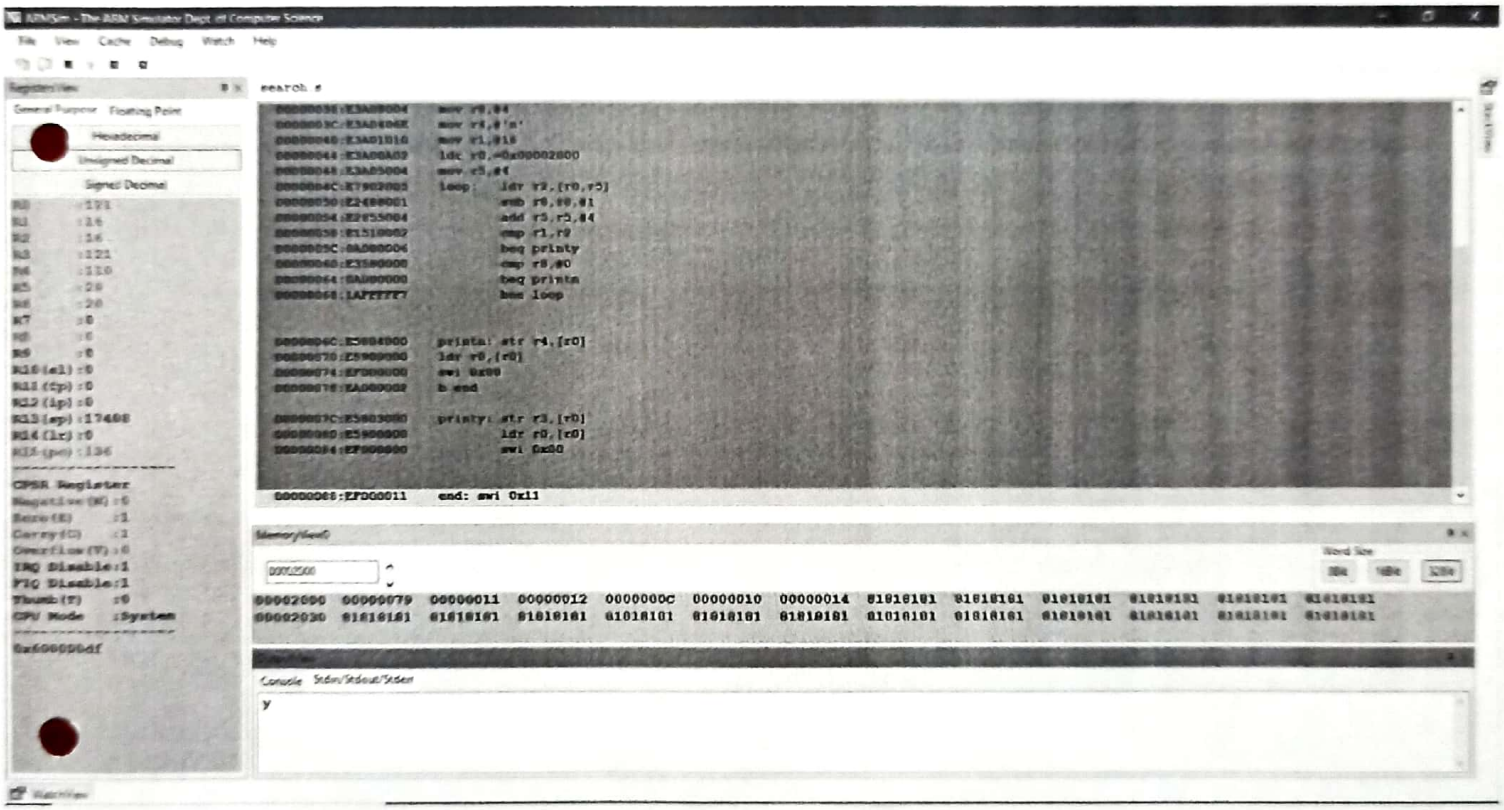
```
result : CMP   R4, R6
         BNG   end
         MOV   R7, #1

end:
         SWI   0X11
```

3]
```
.equ SWI  -open, 0x66
.equ SWI  - close, 0X68
.equ SWI  -print, 0x 66
.equ SWI  - RdInt, 0X6C
.equ SWI , -exit, 0X11
```

```
global_start
text
LDR  r0, =Filename
mov  r1, #0
SWI  SWI_open
bcs  Exit
mov  r9, r0
mov  r5, #0
loop test
mov  r5, #stdout
Mov  r0, r9
 LDR r8, =Array
bcs  afterloop
str  r0, [r6, r5]
add  r5, r5, #4
 mov r1, r0
 mov r0, #Shdout
Shi  SWI-pri
cmp  r4, #0
beq  end
bne  loop
excl : Mov  r0, r9
       swi  SWI_clot

Exit:
 swi  SWI Exit
```





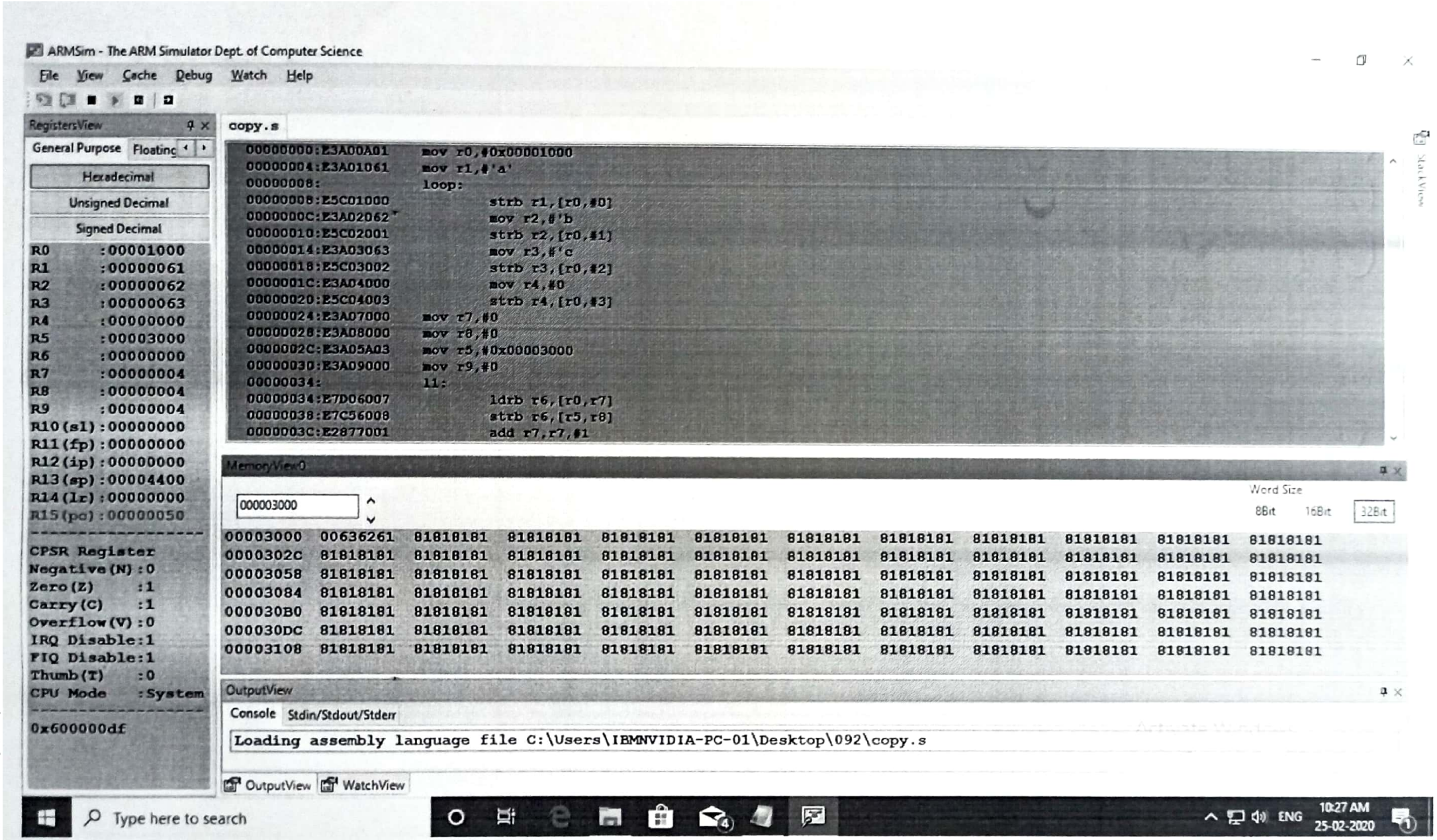


copy - Notepad

File Edit Format View Help

```
mov r0,#0x00001000
mov r1,#'a'
loop:
        strb r1,[r0,#0]
        mov r2,#'b
        strb r2,[r0,#1]
        mov r3,#'c
        strb r3,[r0,#2]
        mov r4,#0
        strb r4,[r0,#3]
mov r7,#0
mov r8,#0
mov r5,#0x00003000
mov r9,#0
l1:
        ldrb r6,[r0,r7]
        strb r6,[r5,r8]
        add r7,r7,#1
        add r8,r8,#1
        add r9,r9,#1
        cmp r9,#4
        bne l1
swi 0x11
```


Ln 19, Col 14 100% Windows (CRLF) UTF-8



Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Department of CSE

Programme: B.E **Term: Jan to May 2019**
Course: Computer Organization **Course Code: CS45**

Activity V: Designing an ALU to perform arithmetic and logical functions using Logisim simulator.

Name: SUKRUTH S	Marks: /10	Date: 22/05/2020
USN: 1MS18CS142	Signature of the Faculty:	

Objective: To simulate the working of Arithmetic and Logical Unit using simulator.

Simulator Description: Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to build larger circuits from smaller sub circuits, and to draw bundles of wires with a single mouse drag, Logisim can be used (and is used) to design and simulate entire CPUs for educational purposes.

Activity to be performed by students:

List out the steps in designing ALU

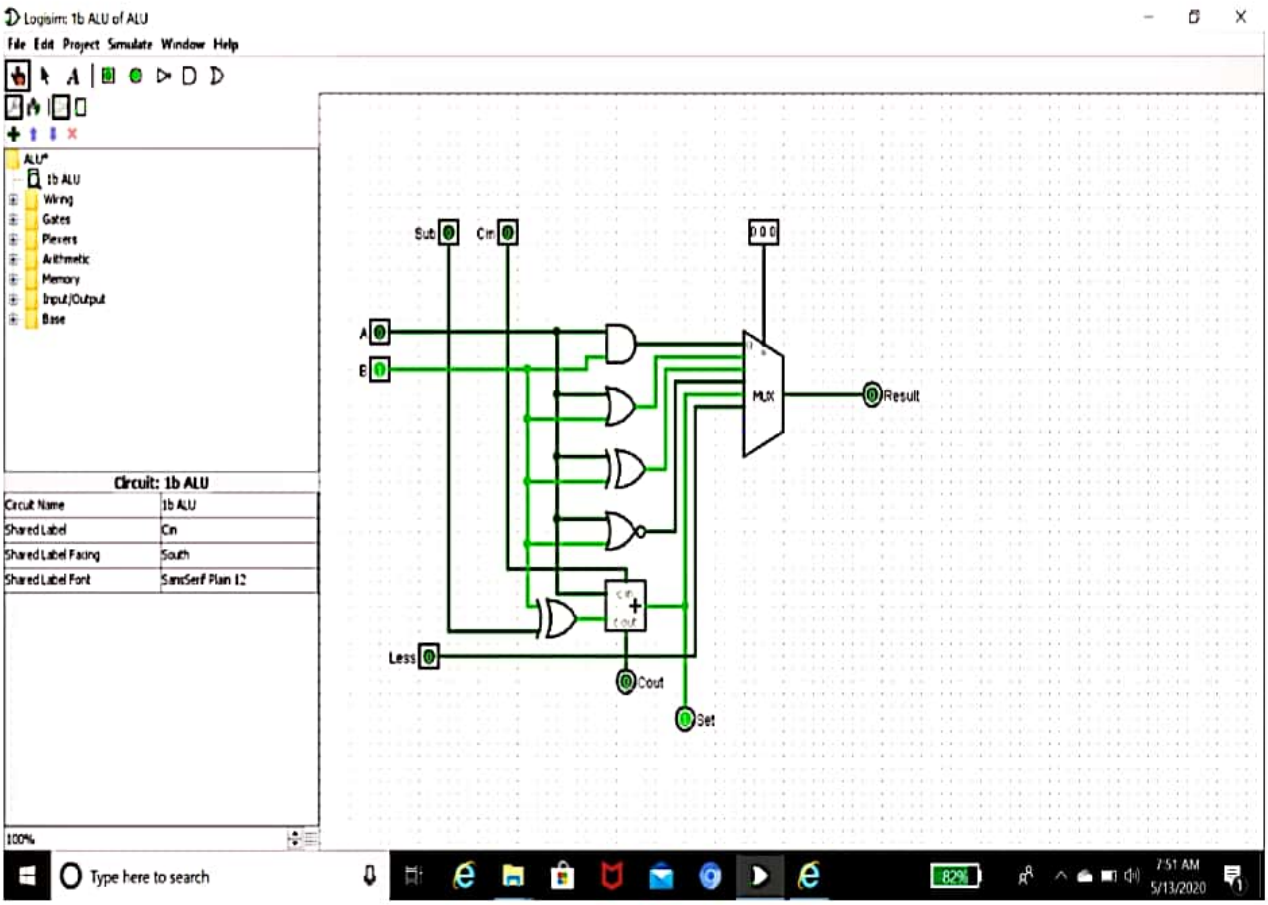
SUKRUTH. S
1MS18CS142
4th "B"

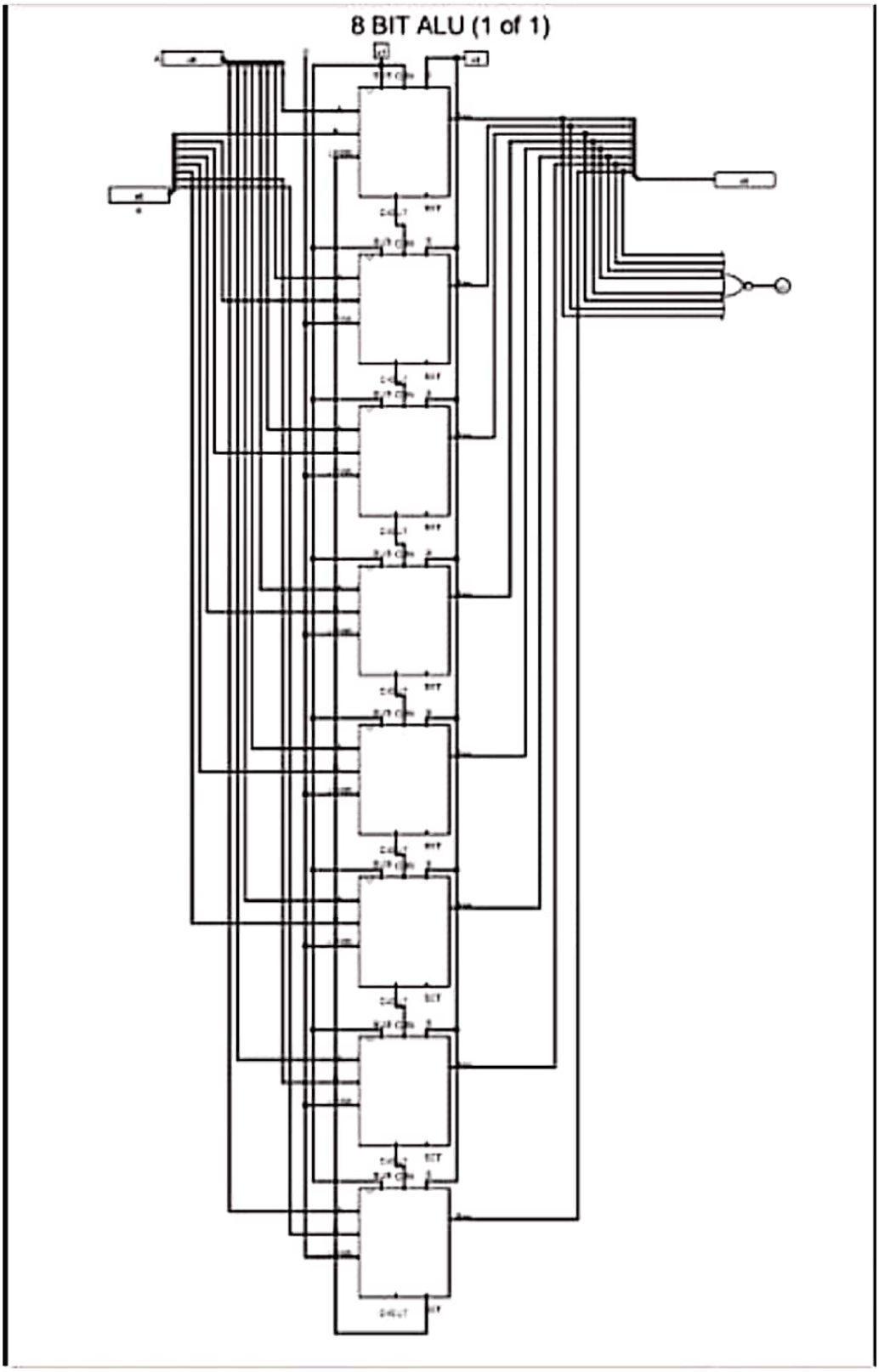
COMPUTER ORGANISATION [CS45]

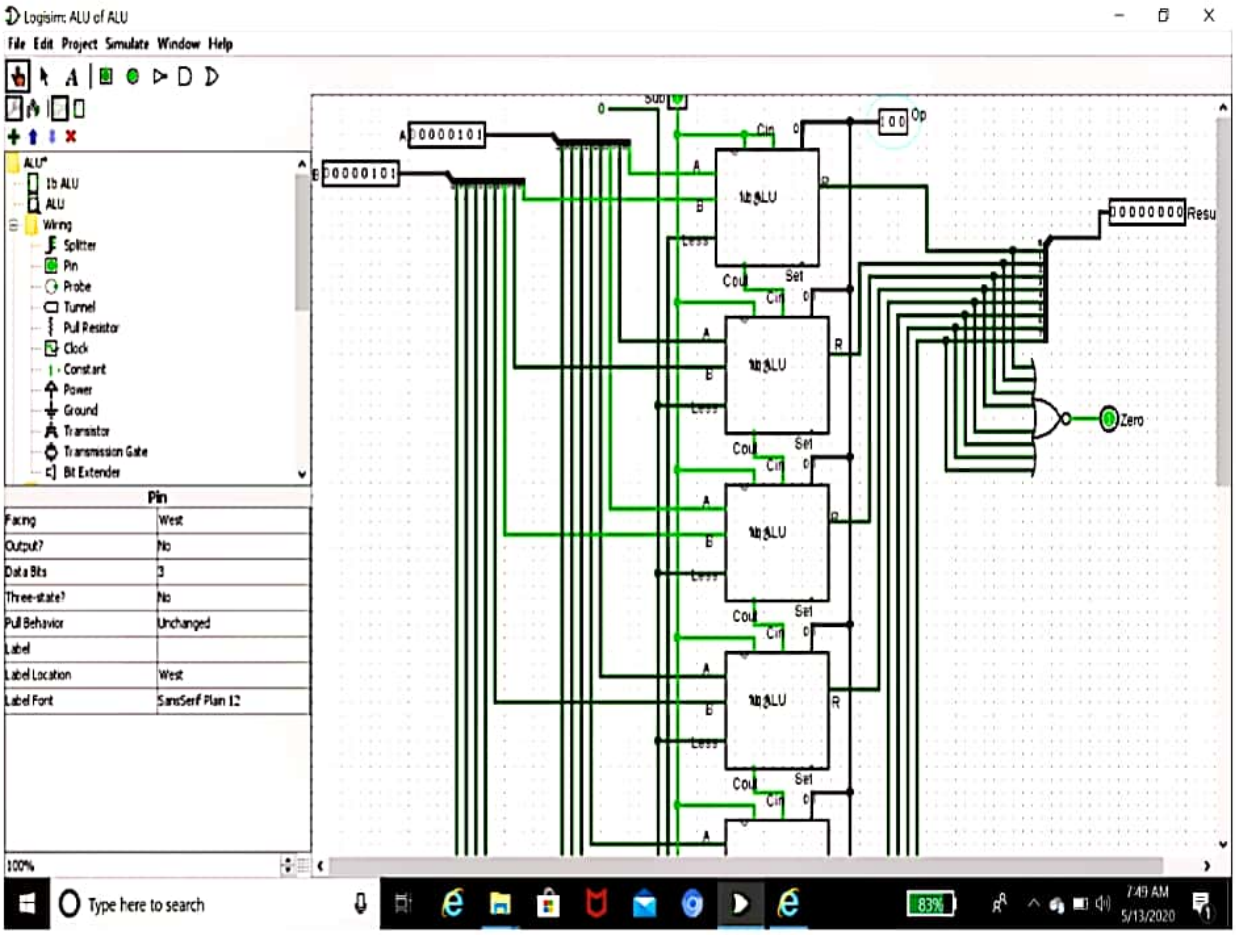
LAB - 5

List out the steps in designing the ALU

1. Place 2 pins A and B (input pins)
2. Connect the input pins to an AND gate
3. Similarly connect A and B to an 'OR' gate, 'XOR' gate and 'NOR' gate and add a 1-bit adder
4. Place a multiplexer with 3 select bits.
5. Connect the outputs of all the gates to the MUX and a 3 bit input pin to the MUX.
6. Connect the output of the MUX to the output pin and name the output pin as 'Result'
7. Connect an input pin (Cin) to the 1-bit adder and an output pin (Cout) to the 1-bit adder
8. Connect another input pin ('sub') to an EXOR gate and the output of it to the Cout
9. Connect an input pin (Less) to the MUX for comparisons
10. Connect an input pin (set) to the output of the 1-bit adder which is in-turn connected to the MUX.







Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Department of CSE

Programme: B.E | Term: Jan to May 2019
Course: Computer Organization | Course Code: CS45

Activity VI: Designing memory system using Logisim simulator.

Name: SUKRUTH S	Marks: /10	Date: 22/05/2020
USN: 1MS18CS142	Signature of the Faculty:	

Objective: To simulate the writing operation on memory.

Simulator Description: Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to build larger circuits from smaller sub circuits, and to draw bundles of wires with a single mouse drag, Logisim can be used (and is used) to design and simulate entire CPUs for educational purposes.

Activity to be performed by students:

List out the steps in designing memory system

SUKRUTH.S
IMS18CS142
4th 'B'

COMPUTER ORGANISATION [CS45]

LAB - 6

List out the steps in designing memory system

1] Add a RAM with seperate load and store selected

2] Place a counter and connect it to the RAM

3] Connect a clock to a controlled buffer and the output of it to the RAM

4] Connect an input pin to that buffer.

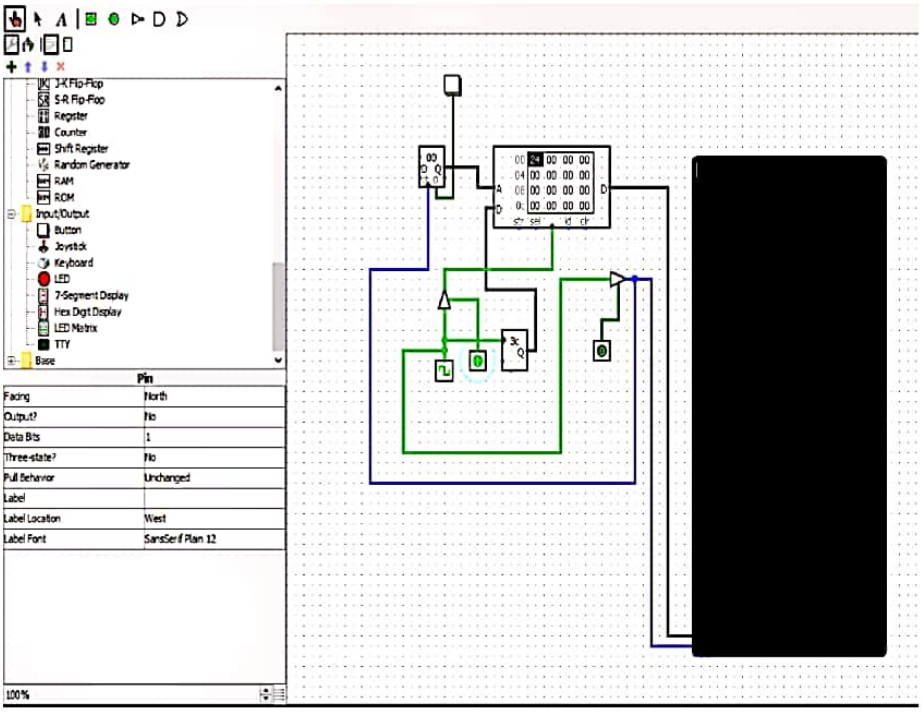
5] Place another controlled buffer and give an input from the input pin and also from the clock.

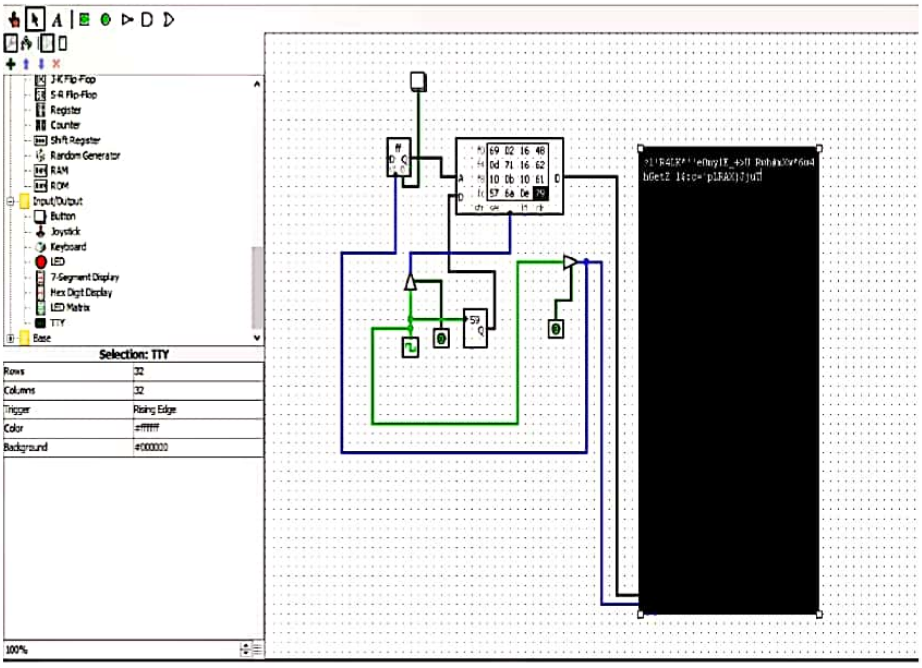
6] Now add a 7-bit random generator and connect Q to D of the RAM.

7] Connect the 'D' from RAM to a TTY unit to display

8] Connect another input from the second controlled buffer to the TTY

9] Connect a RESET button to the counter.





SUKRUTH.S
1MS18CS142
4th "B"

COMPUTER ORGANISATION [CS45]

LAB - 7

With diagram demonstrate the execution of the following using pipelining technique.