Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of CSE
Tutorial-1

Programme: B.E
Course: Computer Organization

Term: Jan to May 2018
Course Code: CS45

Name:	PAVAN D. S	**Marks:** 19/10	**Date:**
USN:	MS18CS0P1	**Signature of the Faculty:**	

Activity I: Assembling and disassembling of a computer

Objective: To demonstrate the functional units of a system.

Assembling of a system: A PC computer is a modular type of computer, it can be assembled using hardware components made by different manufacturers, so as to have a custom built computer according to one's specific needs.

Disassembling of a system: When referring to hardware, **disassemble** is the process of breaking down a device into separate parts. A device may be disassembled to help determine a problem, to replace a part, or to take the parts and use them in another device or to sell them individually.

Activity to be performed by students: Identify the different parts of the system including its interconnection. Observe the assembly and disassembly procedure.

Answer the following questions.

1. Write down the detailed procedure to assemble a system.
2. Explain how troubleshooting a system helps to trace and correct the faults in a system
3. List out the procedure to install extra memory card to a system
4. With a diagram explain different cables used to connect function units in a system.
5. Discuss the safety precautions one should take while removing components of a system

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	PAVAN D G	Branch:	CSE
USN/Roll No. :	1MS18CS088	Sem/Sec:	IV 'B'
Subject :	COA	Subject Code:	

1. Write down the detail procedure to assemble the system.

A:- — Before starting assembling the computer system, make sure you have the screws and a screwdriver.

— The first step for assembling the computer system starts with mounting the processor on the processor socket of the motherboard. Once the processor is mounted, the heat sink will be attached on top of the processor. The CPU fan also attached on top of the heat sink.

— The motherboard is to be fixed vertically in the tower case and the screws are fixed from behind of the motherboard.

- Line up the power supply at the top back end of the cabinet and screw it.

- Install the CD/DVD drives at the top front end of the cabinet and screw it. Install the Hard disk drive and floppy disk drive below CD/DVD drive & screw it

- Select the appropriate data cable & connect one end of the cable to its drive socket and another end at its appropriate connector on the motherboard.

- For SATA hard disk drive (or) CD/DVD drives use SATA cable and its power cable else use IDE data cable.

- Mount the memory modules on the motherboard by aligning the RAM to its socket on motherboard and press it downward.

- Install the internal cards to its socket and attach the cables / power cable to it.

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	PAVAN D. G	Branch:	CSE
USN/Roll No. :	1MS18CS088	Sem/Sec:	IV / B
Subject :	COA	Subject Code:	

- Cover the tower by placing it and pressing towards front side & screw it.
- Connect the external devices with CPU at its appropriate socket. Monitor at the vedio output socket.
- Connect the power cable to the back of tower in SMPS. Plug in power cable to the electic board.

2) Explain how trouble shooting a system helps to trace and correct the failure in the system.

A:- Troubleshooting is a form of problem often applied to repair failed products or processes on a machine (or) a system. It is a logical, systematic search for the source of a problem in order to solve it and

make the product or process operational again troubleshooting is needed to identify the symptoms. Determing the most likely cause is a process of elimination, eliminating potential causes of a problem. Finally, troubleshooting requires confirmation that the solution restores the product (or) process to its working.

Step 1 :- Identify the problem.

Step 2 :- Establish a theory to determine cause.

Step 3 :- Test the theory to determine cause.

Step 4 :- Establish a plan of action to resolve the problem and implement the solution.

Step 5 :- Verify full system functionality and if applicable implement preventive measures.

Step 6 :- Document findings, actions & outcomes.

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

A :- Step 1 :- Disconnect the power cable from the system and if needed, unplug other back-panel

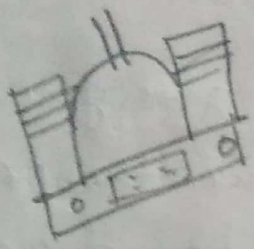
cables so that you can safely turn your system on to its side.

Step 2:- Remove the side panel to give you full access to the interior and locate the RAM slots. They are most commonly found next to the processor and its cooler. If there already RAM in your system, eject it by pressing firmly on the tabs on the motherboard at either end of the slots.

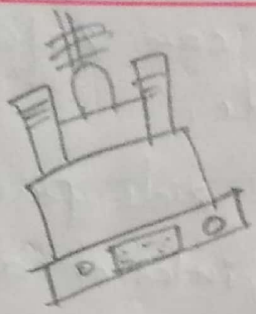
Step 3:- To install the new RAM, line up notches in the bottom of the sticks with gaps in the slot on the motherboard.

Step 4:- Once the sticks have clicked into confirm that the wing clips are locked into hold the stick firmly.

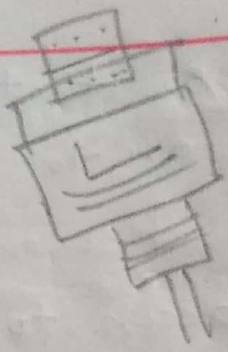
4) With a neat diagram explain different cables used to connect functional units in a system.



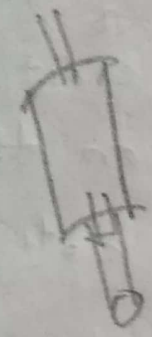
VGA cable



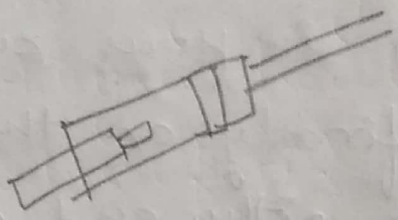
DVI cable



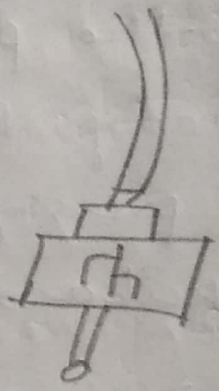
HDMI cable



PS/2 cable



Ethernet cable



USB cable

Computer power cord (Kettle plug.

VGA cable — connect one end to computer, monitor, TV connect other end to VGA.

DVI cable — connect one end to computer monitor other end to DVI port on Computer.

HDMI cable :- connect one end to computer monitor & the other end to HDMI port.

PS12 cable :- connect one end to PS12 keyboard PS12 mouse, other end to PS12 ports on computer.

Ethernet cable:- connect one end to router network switch other end to ethernal port on computer.

USB cable:- one end to USB device other end to USB ports on computer.

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

A :- 1. Fully shutdown and unplug the computer before disassemble the forces.

2. Take off any metal objects on your arms such as rings even if your unit is unplugged.

3. Make sure your hands are completely dry.

4. Work in a cool area to avoid perspiration.

5. Before touching any part within the tower put your hands against another metal surface.

6. Prepare a place to keep any screws you may remove.

7. If a component doesn't come out easily, do not forcefully remove it.

8. Be carefull when handling the motherboard.

9. Handel all parts with care. Place each piece you remove carefully down onto a stable surface.

10. Take note that the three most damaging things to a computer are moisture, shock & 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:	PAVAN D.G	**Marks:** 10 /10	**Date:**
USN:	1MS18CS088	**Signature of the Faculty:**	4/12/2020

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

3. Results and Snapshots:

RAMAIAH
Institute of Technology

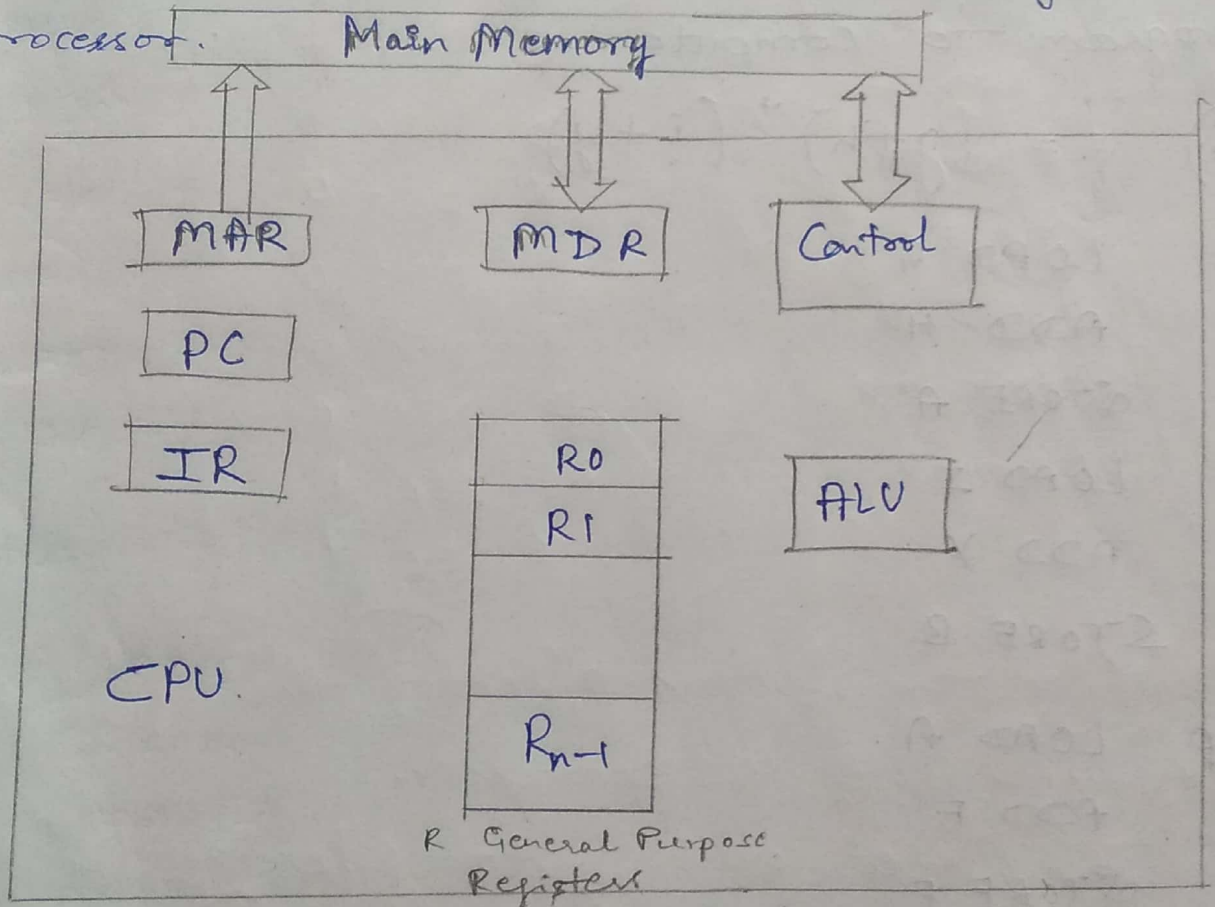
Data Sheet

MARKS :

Name :	PAVAN D.G	Branch:	CSE
USN/Roll No. :	1MS18CS088	Sem/Sec:	IV B
Subject :	COA	Subject Code:	

(1) Draw the interconnection b/w memory & processor.

A:-



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

A:- There are 6 steps:-

1) Fetch instruction
2) Decode information.

3) Perform ALU operation.

4) Access memory

5) Update register file

6) Update the program counter (PC)

(3) Write and execute the assembly language program to compute.

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

A:-

```
    LOAD G
    ADD H
    STORE A
    LOAD I
    ADD Y
    STORE B
loop LOAD A
    ADD F
    STORE F
    LOAD B
    SUBT one
    STORE B
```

```
    JUMP LOOP
    LOAD F
    Output
    HALT
Q   DEC 9
H   DEC 5
Y   DEC 8
I   DEC 2
A   DEC 0
B   DEC 0
F   DEC 0
one DEC 1
```

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

```
A:-  LOAD B
     STORE 0
     LOAD B
     ADD X
     STORE X
     SUBT one
     STORE 0
     JUMP first
```

```
Second  LOAD    Y
        ADD     X
        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     2
D       DEC     0
X       DEC     0
Y       DEC     0
Z       DEC     0
D       DEC     0
one     DEC     1
four    DEC     4
```

(4) Describe the factor affecting the performance of a processor.

A:- There are four factor affecting the performance of a processor.

i) Multiple cores :- Nowadays we have dual, quad and even octa core processors with its own fetch and execute cycles. However, the software should

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	PAVAN D.G	Branch:	CSE
USN/Roll No. :	1MS18CS088	Sem/Sec:	IV 'B'
Subject :	COA	Subject Code:	

make use of the multiple cores.

- clock speed - The processor requires a clock pulse in order to operate correctly one clock cycle = 1 Hr A PC clock speed is normally in GHz regia.

- Cache memory :- It is a small amount of high performance RAM that is built into the processor. This RAM stores the data which has to be repeatedly used by the processor if it doesn't require a request from memory.

- Word length :- The no of bits the CPU and procer simultaneously, for ex, a 32-bit processor because of the wides word length.

- Address Bus width :- It is the width of the address bus and determines the maximum amount of addressable locations. For example, an address bus of 8 bit means that you can have 256 address (0 to 255)

* Data Bus width :- It is the no. of bit that can be transferred simultaneously from one device to another. If the data bus is 16 bits and the address bus is 32 bits, so the data is fetched in 2 x 16 bit groups.

MARIE Assembler Code Editor

File Edit Assemble Help

```
LOAD X
ADD Y
STORE Z
HALT
X, DEC 5
Y, DEC 1
Z, DEC 0
```

D:\CO Lab\MARIE_Datapath_Simulators\add.mas Assembly successful

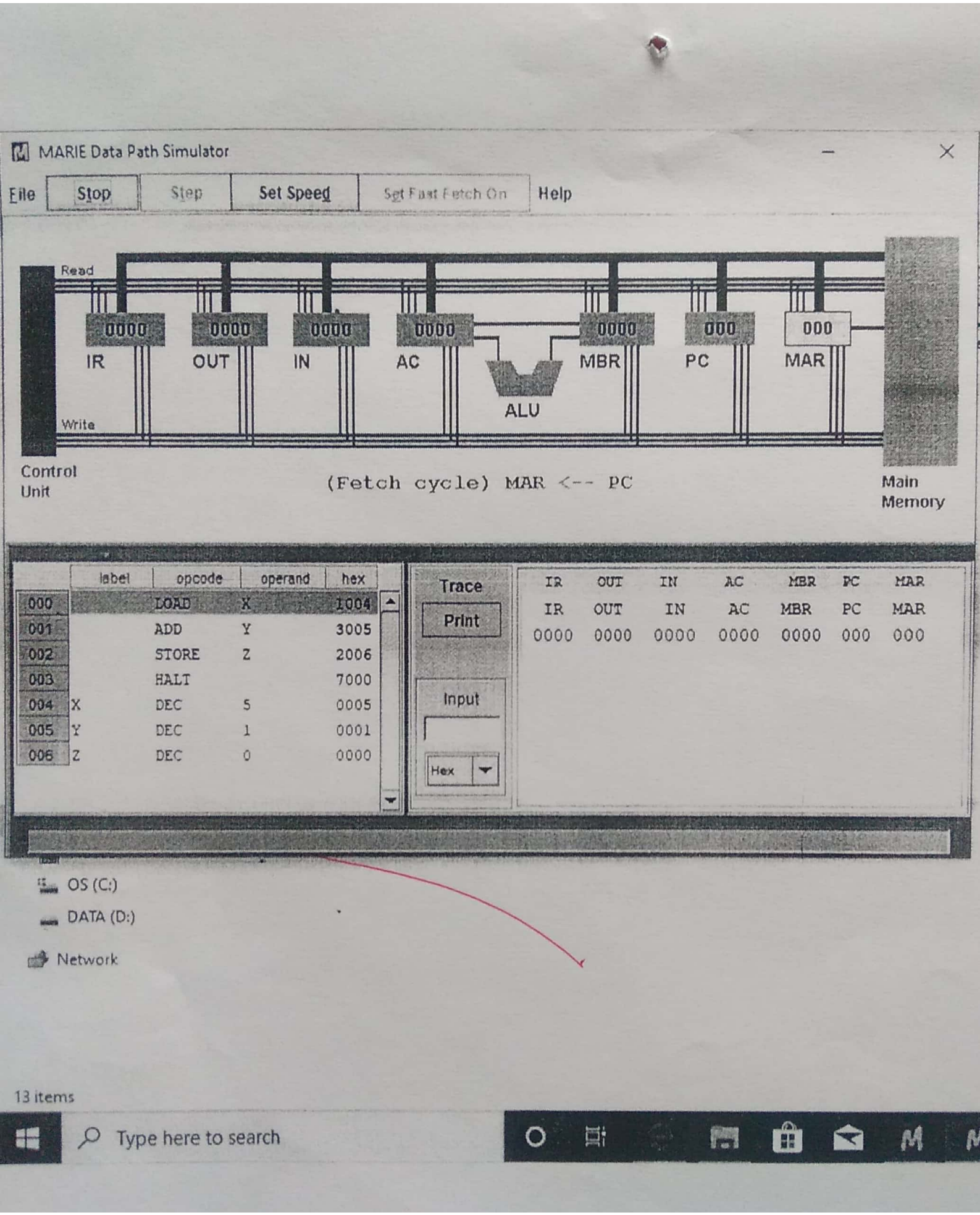
Type here to search

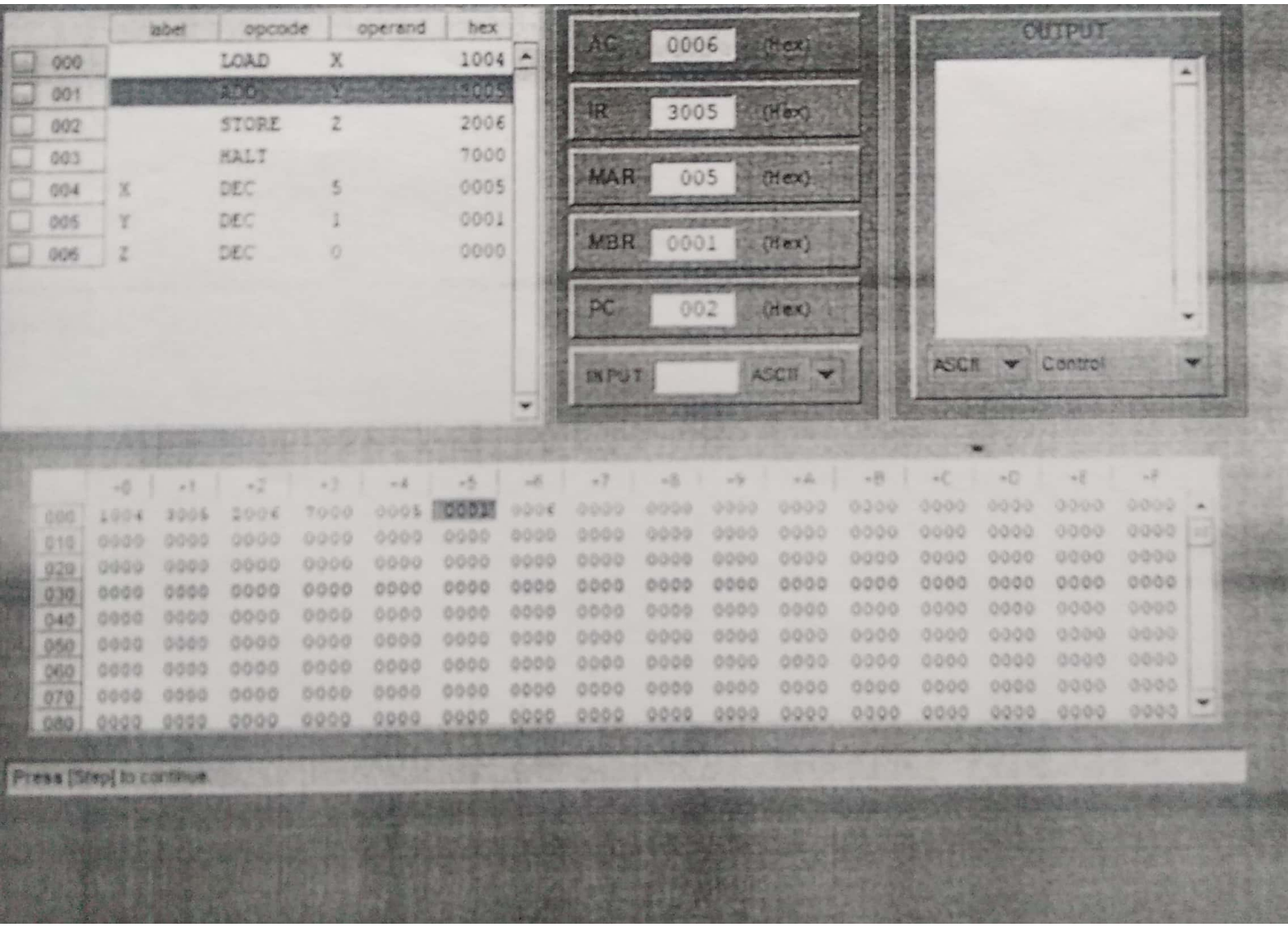
Assembly Listing for add.mas

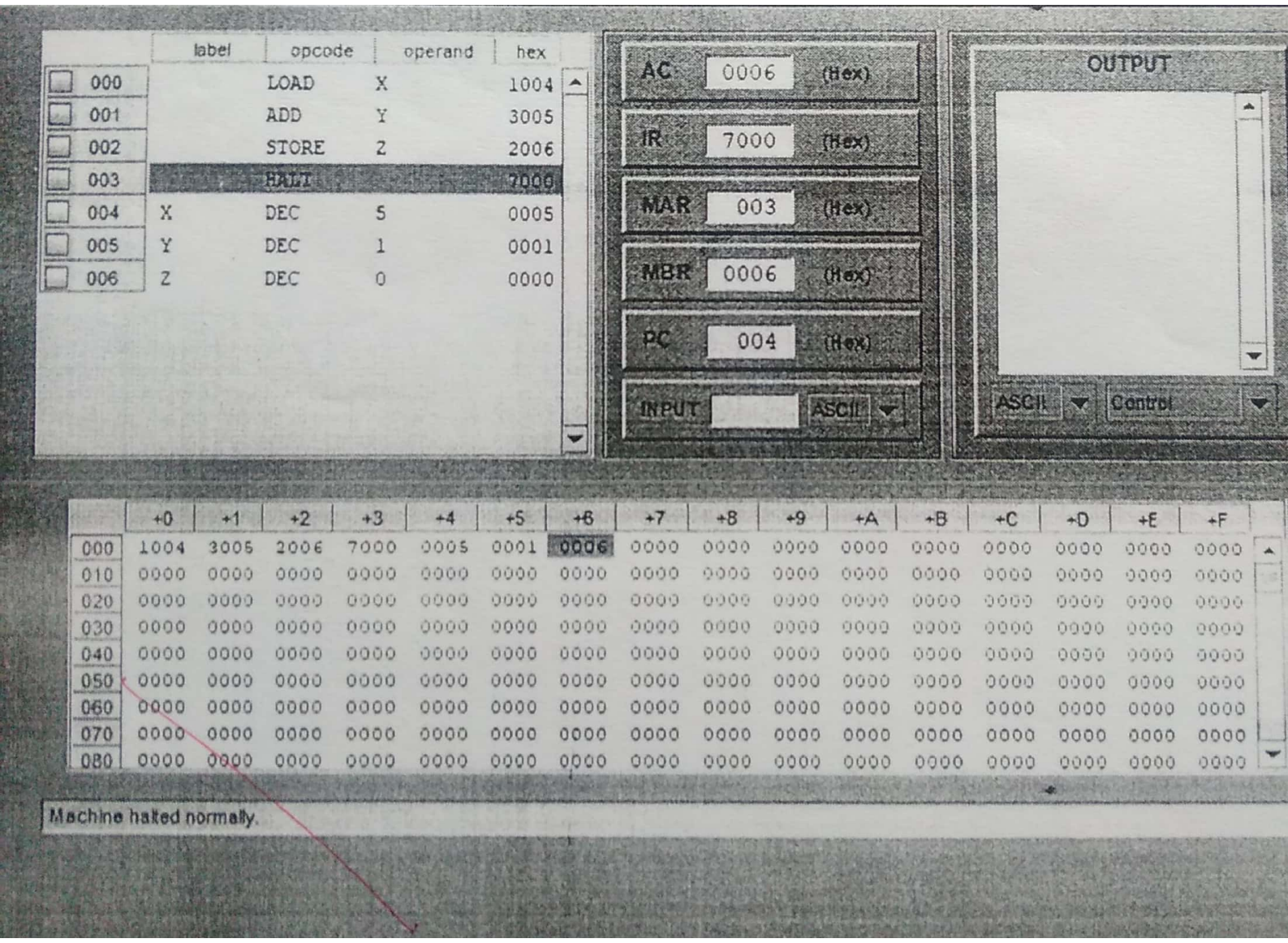
```
     Assembly listing for: add.mas
                Assembled: Tue Jan 28 06:57:27 IST 2020

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

Type here to search







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: PAVAN D.G	**Marks:** 10/10	**Date:** 04/02/20
USN: 1MS18CS088	**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 :	PAVAN D G	Branch:	CSE
USN/Roll No. :	1MS18CS088	Sem/Sec:	IV/B
Subject :	COA	Subject Code:	

1) Write a ARM program for arithmetic operation.

A:-
```
MOV   R5, #10
MOV   R6, #20
ADD   R7, R5, R6
SWI 0x11
```

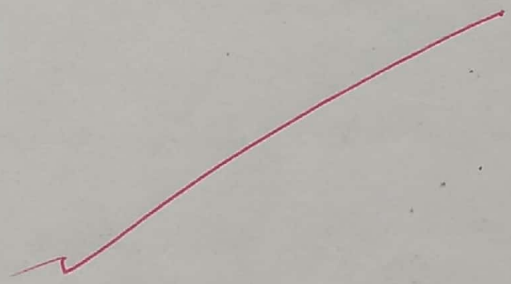
2) Write an ARM program to demonstrate the working of load & store instructions

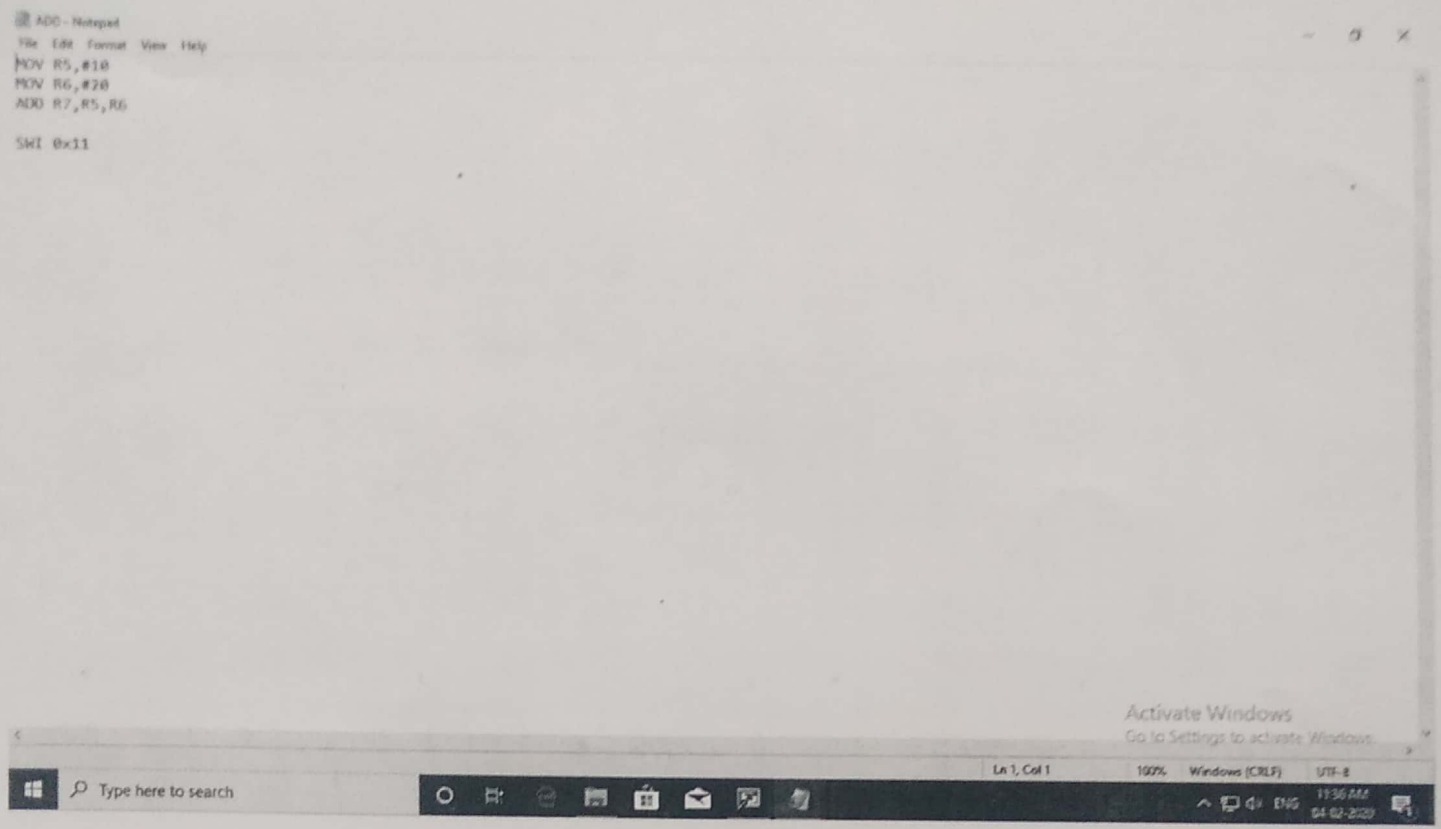
A:-
```
MOV   R1, #0x00000070
MOV   R1, #0
MOV   R4, #50
STR   R4, [R1, R1]
LDR   R6, [R1, R1]
SWI   0x11
```

3)
```
mov   R1, #20
mov   R2, #30
ADD   R3, R1, R2
mov   R4, #40
mov   R5, #50
ADD   R6, R4, R5
SUB   R7, R3, R6
SWI   0x11
```

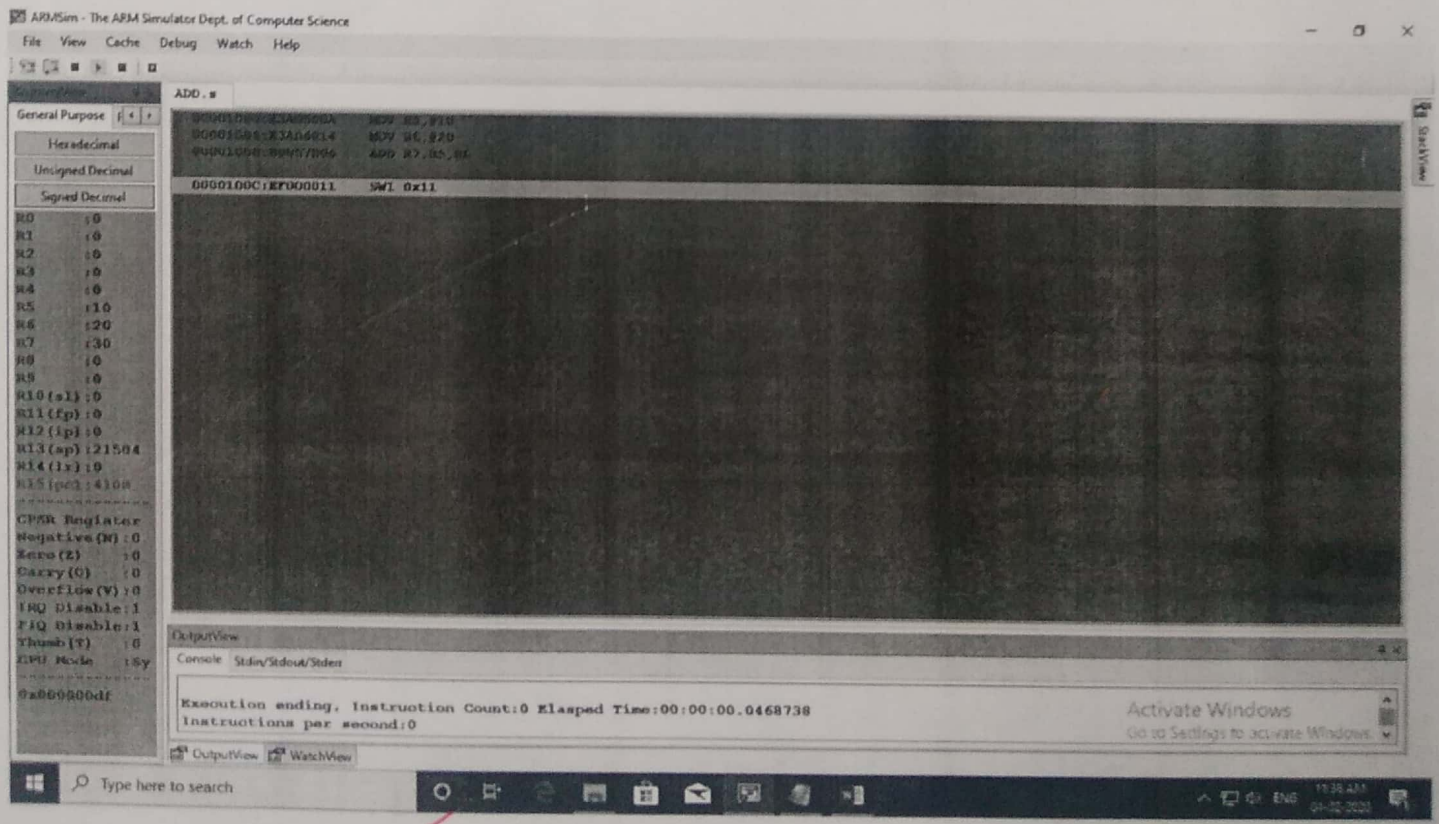
4)
```
mov   R0, #5
LDR   R1, =array
loop  LDR R2, [R1], #4
      ADD R3, R3, R2
      SUB R0, R0, #1
      CMP R0, #0
      BNE Loop
array DCD 0x00000001, 0x00000002, 0x00000003,
          0x00000004

SWI 0x11
```

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.
```

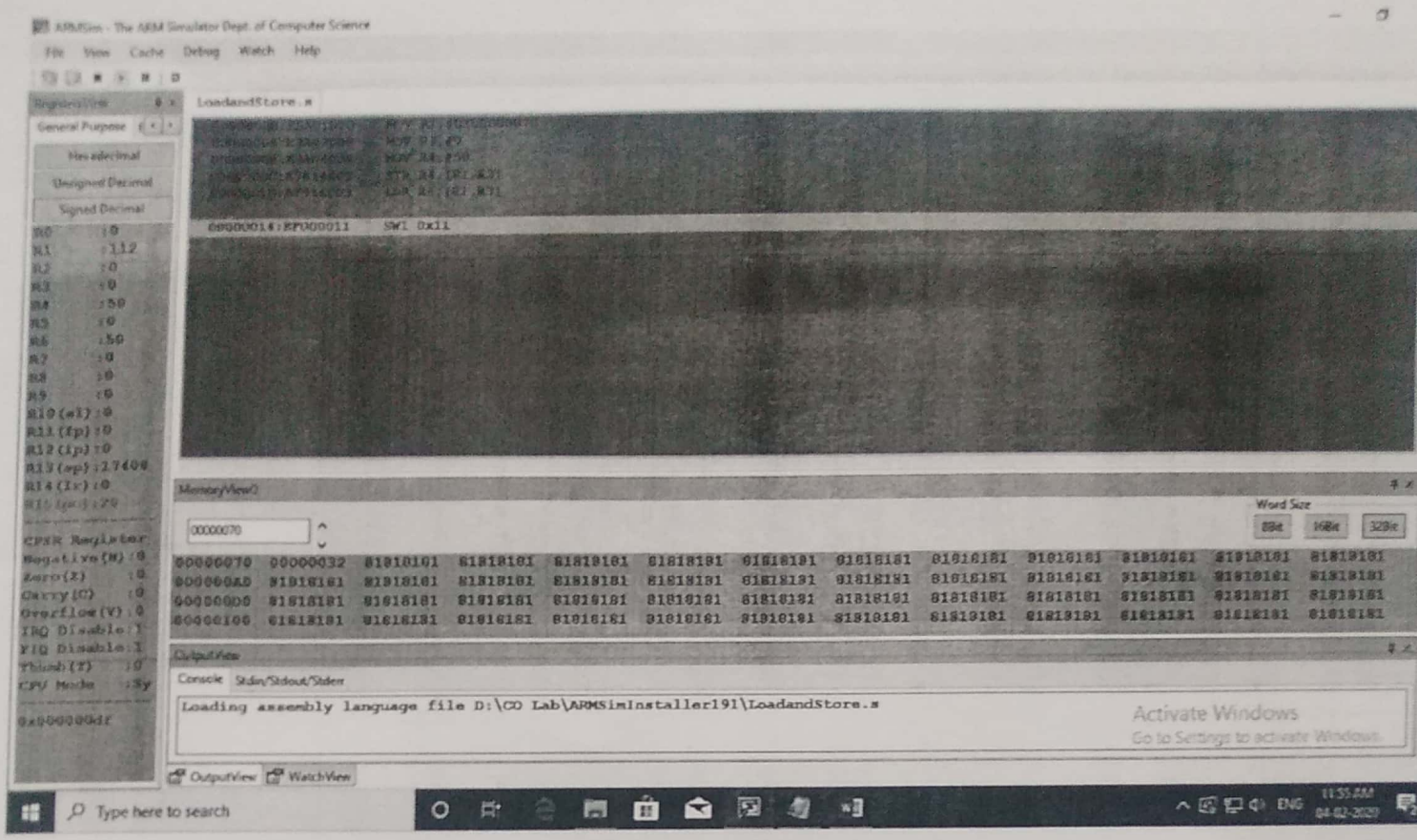


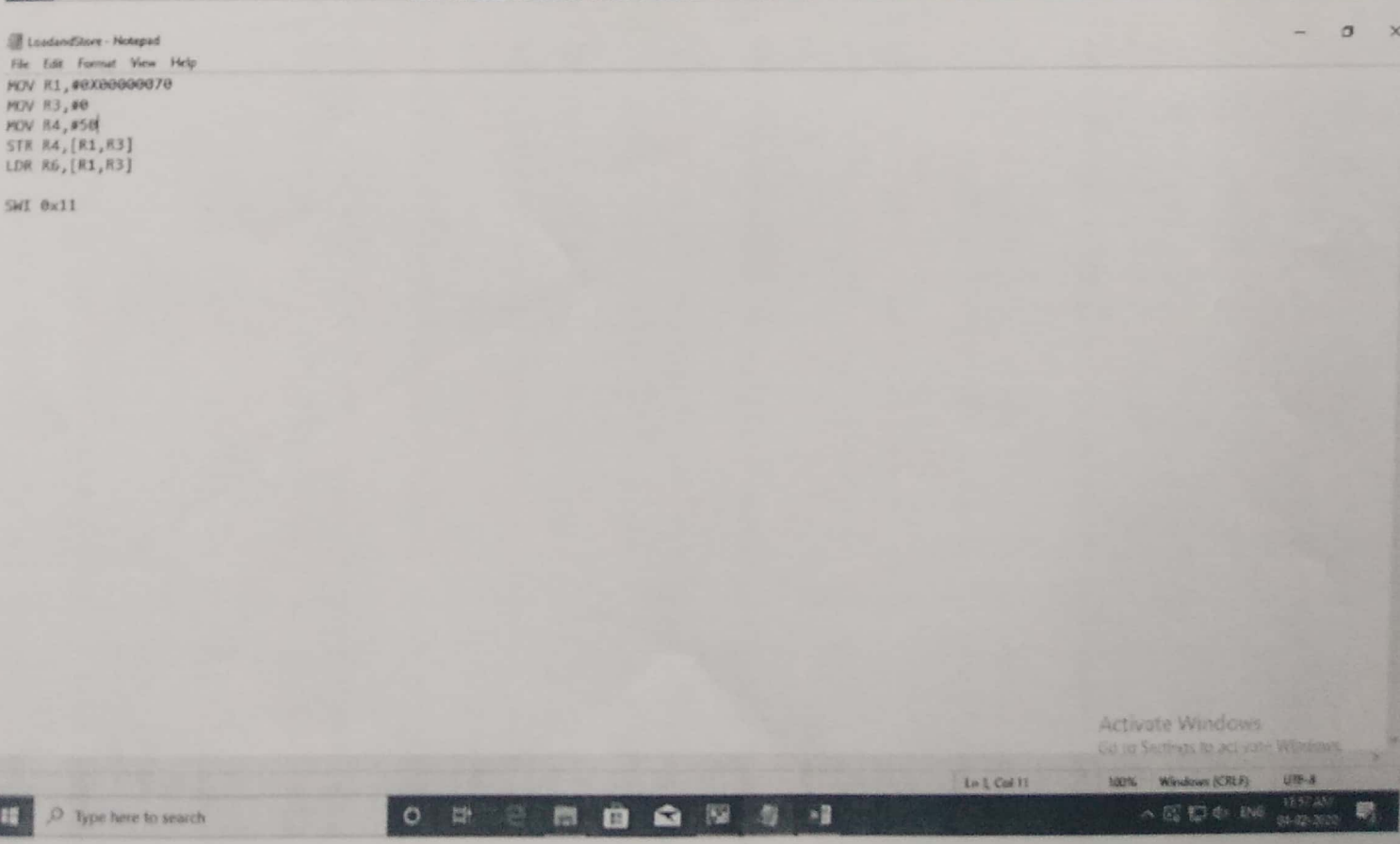


ADD.ASM

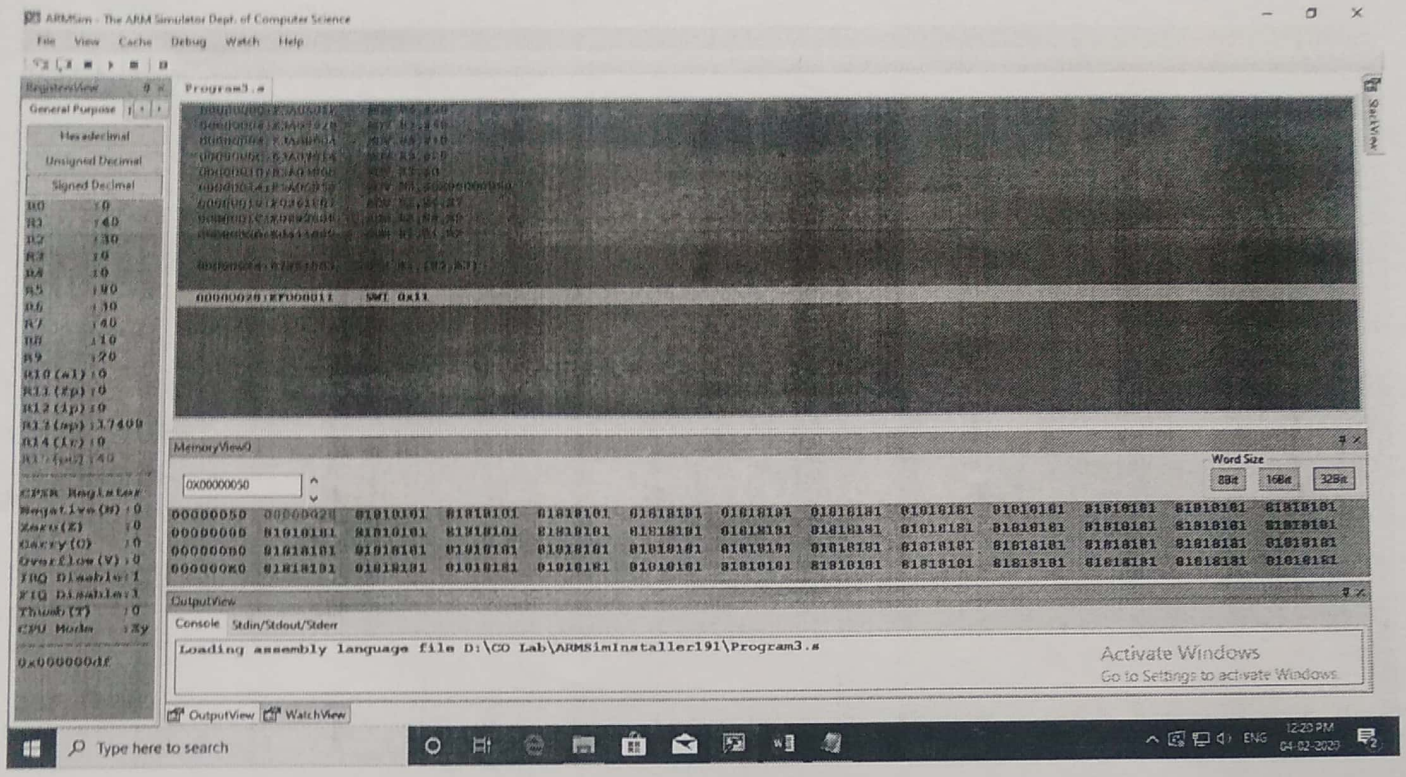


Program 2





Program 3



Program3 - Notepad

File Edit Format View Help

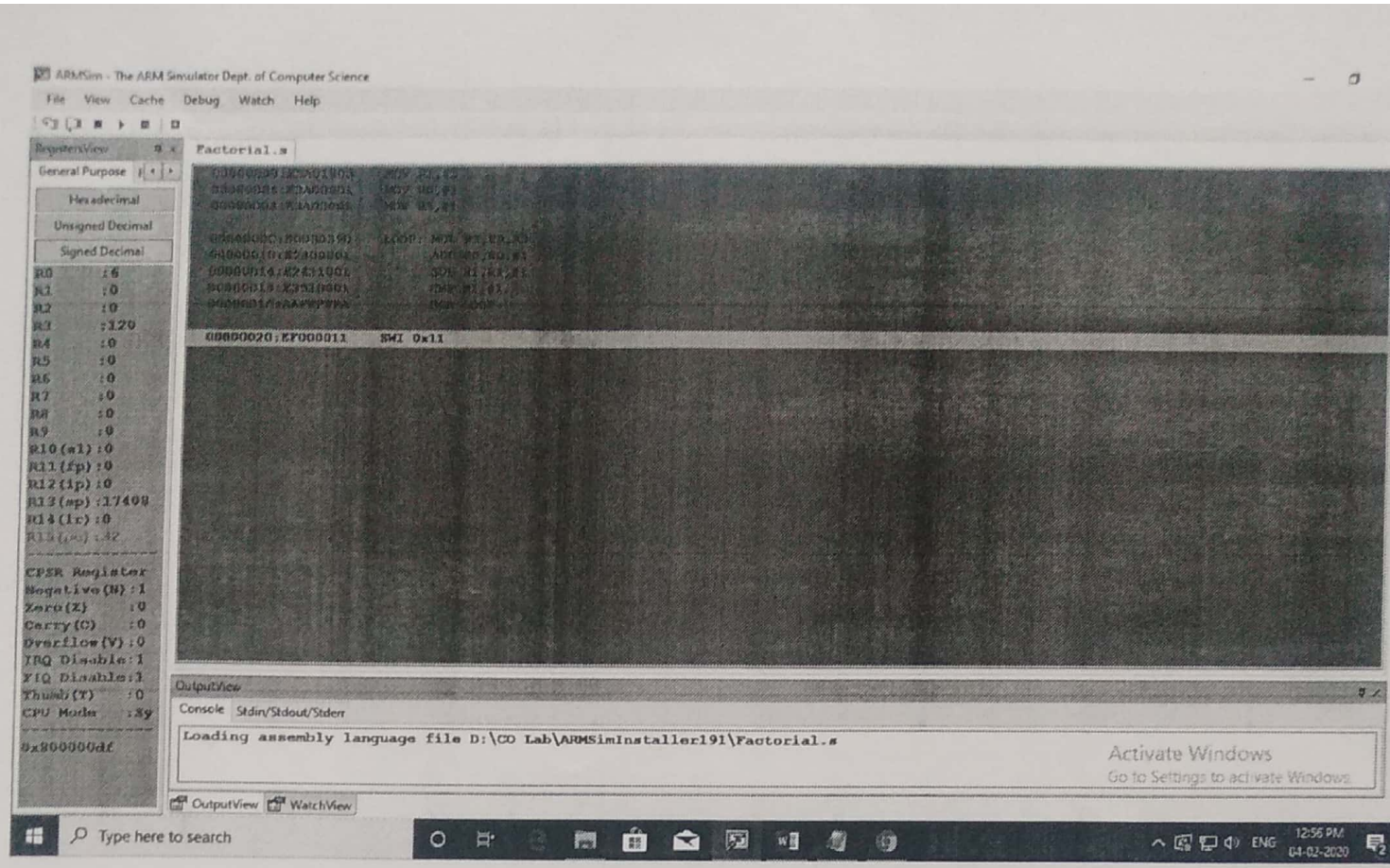
```
MOV R6,#30
MOV R7,#40
MOV R8,#10
MOV R9,#20
MOV R3,#0
MOV R5,#0X00000050
ADD R1,R6,R7
ADD R2,R8,R9
SUB R1,R1,R2

STR R1,[R5,R3]

SWI 0x11
```

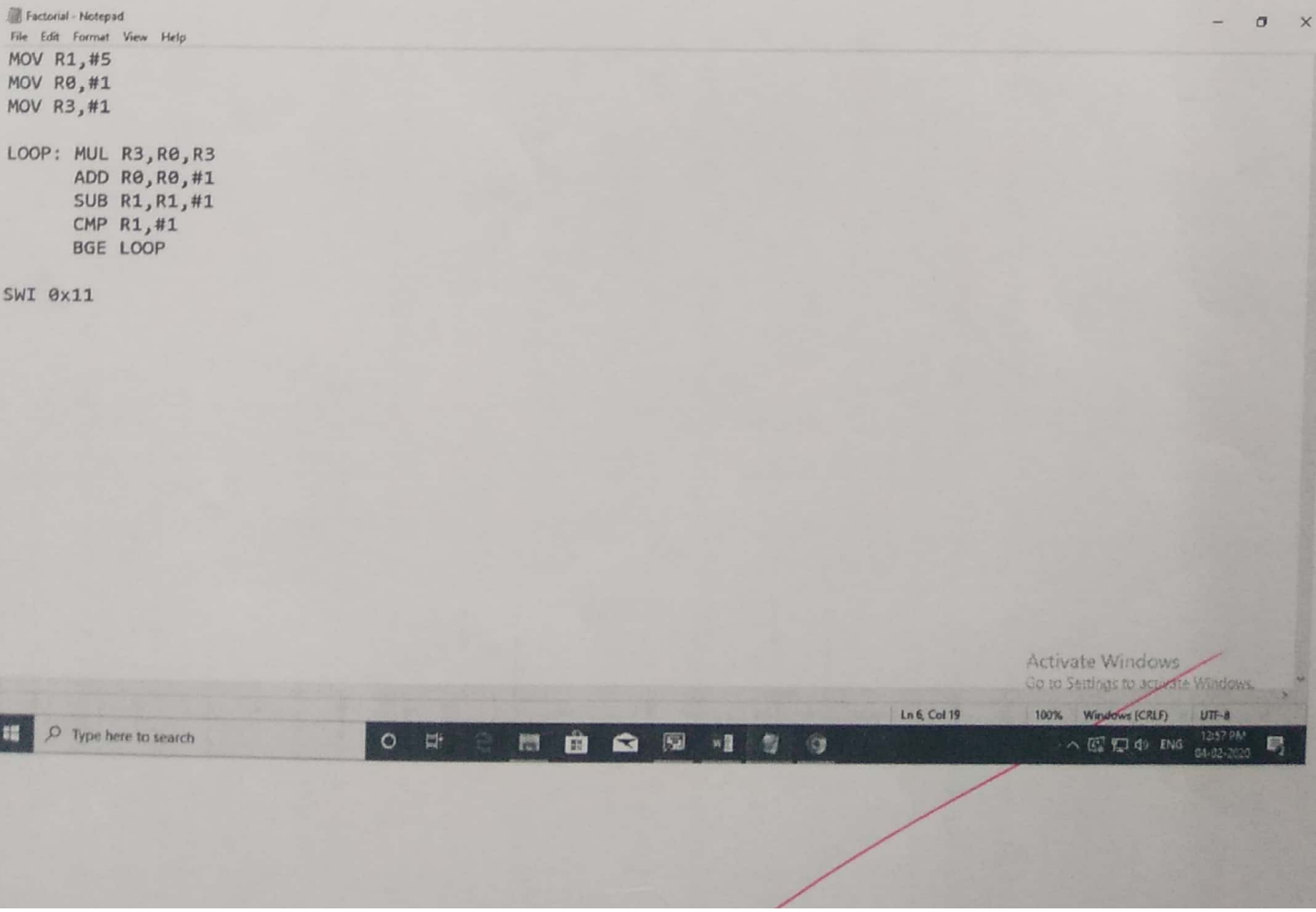
Activate Windows
Go to Settings to activate Windows.

Ln 4, Col 11 100% Windows (CRLF) UTF-8

Type here to search

Factorial





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

Department of CSE

Programme: B.E
Course: Computer Organization

Term: Jan to May 2019
Course Code: CS45

Activity IV: Executing ARM programs using ARMsim simulator.

Name: PAVAN D.G	Marks: /10	Date: 25/02/20
USN: 1MS18CS088	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.

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	PAVAN D-G	Branch:	CSE
USN/Roll No. :	1MS18CS082	Sem/Sec:	IV / 'B'
Subject :	COA	Subject Code:	CS45

1)
```
MOV   R0, #0
MOV   R1, #1
MOV   R2, #5
MOV   R3, #0
MOV   R4, =0X00002000
MOV   R5, #0
loop:
      STR   R0, [R4, R5]
      ADD   R6, [R0, R1]
      MOV   R0, R1
      MOV   R1, R6
      ADD   R5, R5, #4
      ADD   R3, R3, #1
      CMP   R3, R2
      BLT   loop
SWI   0x11
```

3)

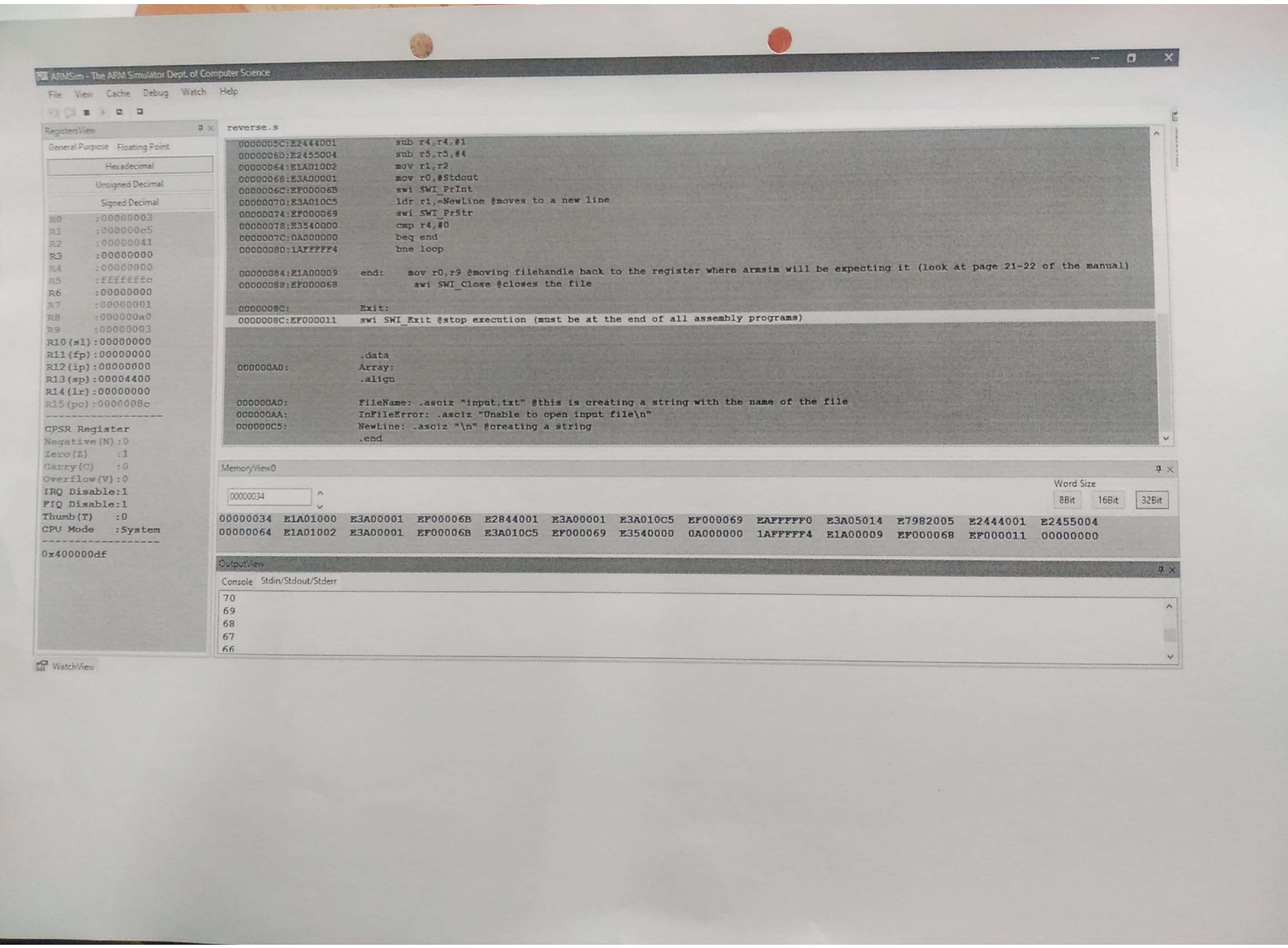
```
MOV  R0, #4
MOV  R1, #'x'
MOV  R4, #25
LDR  R0, = 0x 0000.2000
MOV  R5, #0x1
loop LDR, R2, [R0, R5]
     SUB  R0, R0, #1
     ADD  R5, R5, #4
     CMP  R1, R2
     BEQ  printy
     cmp  R0, #0
     BEQ  printx
     BNE  loop
printx: STR  R4, [R0]
     LDR  R0, [R0]
     SWI 0x00
     B END
```

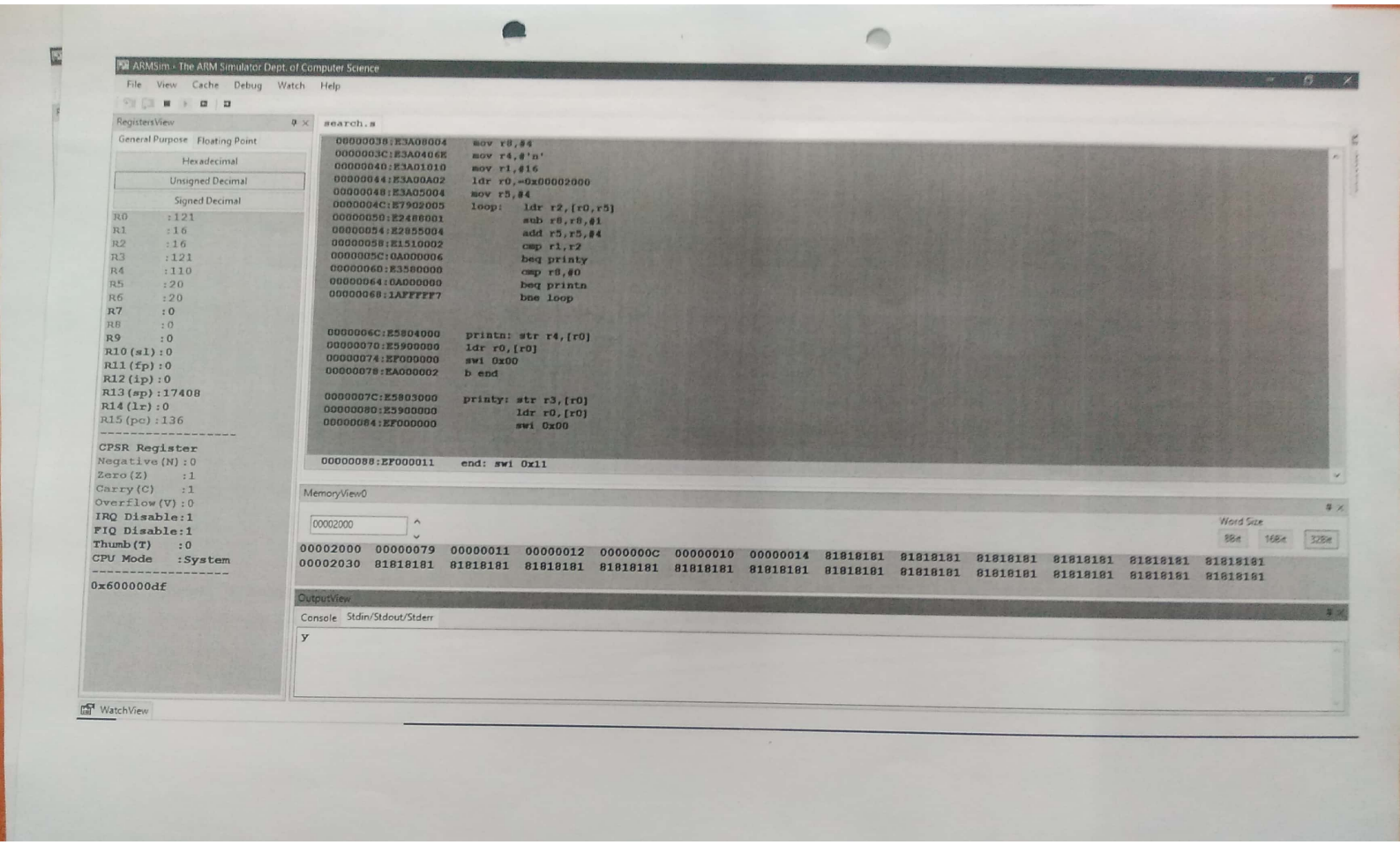
```
Printy :  STR  R7,[R9]
          LDR  R0,[R0]
          SWI 0x00
end       SWI 0x11
```

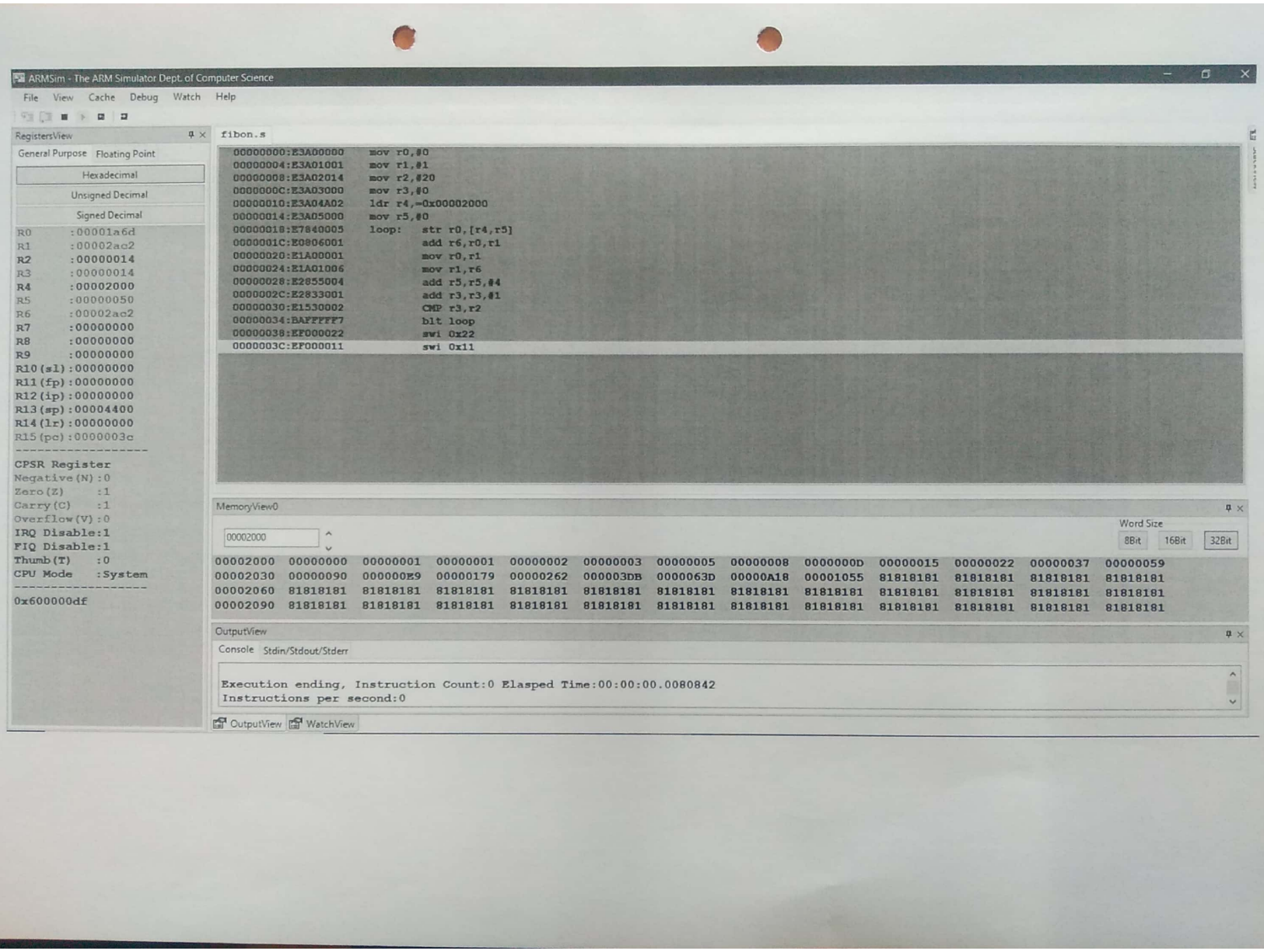
3)

```
.equ   SWI-open, 0x66
.equ   SWI-close, 0x68
.equ   SWI-print, 0x6b
.equ   SWI- RdInt, 0x6c
.equ   Stdout, 1
.equ   SWI-PrStr, 0x69
.equ   SWI-Exit, 0x11
global- start
text
LDR    r0, =Filename
mov    r1, #0
SWI    SWIopen
bcs    Exit
mov    r9, r0
mov    r5, #0
```

```
loop start:
mov r5, #stdout
mov r0, r9
LDR r8, = Array
bcs afterloop
str r0, [r8, r5]
add r5, r5, #4
mov r1, r0
mov r0, #stdout
str SWI-print
add r4, r4, #1
afterloop:
mov r5, #20
loop: LDR r2, [r8, r5]
      SUB r4, r4, #1
      SUB r5, r5, #4
      mov r1, r2
      beq end
      beq loop
end: mov r0, r9
     SWI _close
Exit:
   swi SWI Exit.
```







Computer organization and Architecture

PAVAN.G
1MS18CS088

Activity - V

Objective:- To simulate the working of Arithmetic and logic 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 other capacity to build larges circuits from smaller sub circuits and to draw bundles of wires with a single mouse click, logisim can be used to design and simulate entire cpu's for educational purpose.

Steps in designing ALU:-

- Add the two i/p pins name them A & B
- Add or, and, x-or, Nor gates and a 1-bit adder.
- Connect the A's & B's of all the gates to their respective pins.
- Add an output pin and name it result.

5. Add a 1-bit multiplexer with 3 select bits.

6. Connect outputs of all the gates to the mux.

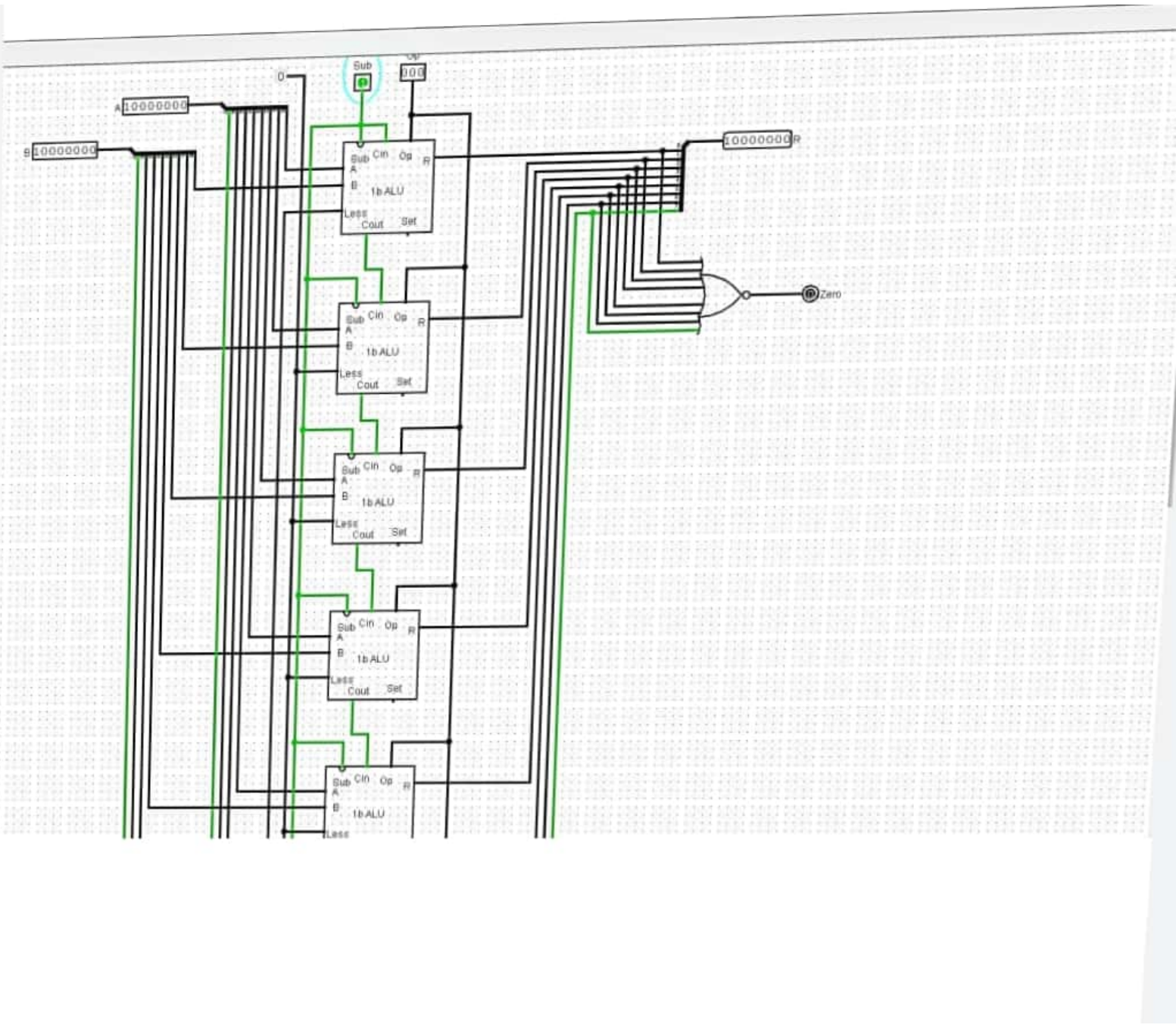
7. Connect 3-bit input pin to mux.

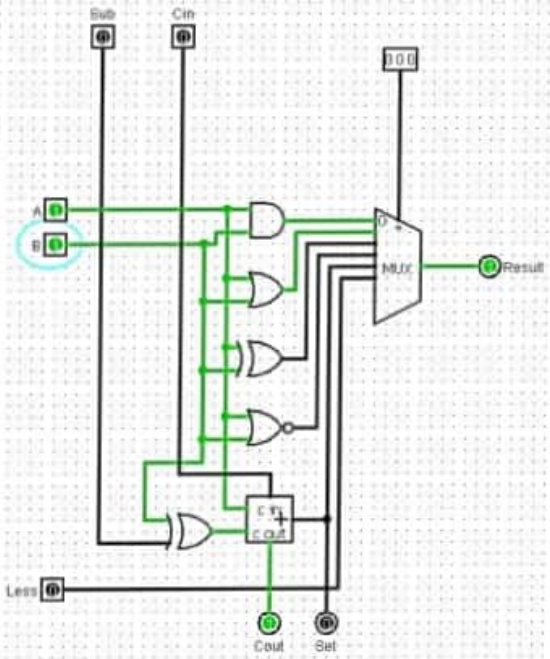
8. Add i/p pin to cin, and output pin to cout.

9. Add an ~~X~~-or gate connect its o/p to cout. The first i/p must to connected A and the second to another i/p pin sub.

10. Add another i/p and name it b. Connect it to the mux.

11. Add an output pin and name it set, connect it to the o/p of adder unit.





Computer organisation & Architecture

Activity VI

Param D.G
1MS18CS066

Objective:- To simulate the writing operation on memory.

Simulator description:- Logism is an educational tool for designing and simulating digital logic circuits with its simple toolbar 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, logism can be used to design and simulate entire CPU's for educational purposes.

The steps in designing memory system:-

1. Add a RAM with separate load and store selected.
2. Add a counter and connect Q to A of the RAM.
3. Add a controller buffer and connect its o/p to the RAM.

4. Add a clock and connect to the i/p of the buffer.
5. Add a TTH unit with 32 rows and columns. Make the connections with RAM.
6. Add a 7-bit random number generator, connect Q to D.
7. Add another connected buffer, connect it to TTH. Also add an i/p pin to the buffer.
8. Connect the o/p of the second buffer to the counter.
9. Connect a button to the counter.