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 20
Course Code: CS45

Name: Kaushik Kamplí	**Marks:** 10/10	**Date:**
USN: 1MS18CS057	**Signature of the Faculty:**	4/2/20

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 :	Kaushik Kamplí	Branch:	CSE
USN/Roll No. :	1MS18CS057	Sem/Sec:	III / B
Subject :	Computer Organisation	Subject Code:	

1) Write down detailed procedure to assemble a system.

The procedure is :-

1) Make sure to have all the parts ready.

2) Install the power supply : some cases come with power supply pre installed, if not follow The power supply will go near the top or bottom rear of the case

3) Add components to the motherboard :

- Attach the processor to the motherboard by finding the processor slot and placed it in the right orientation which is indicated by an arrow
- Attach RAM to the motherboards by snapping them into the slots, a little pressure may be necessary.
- Apply thermal paste to the processor if necessary

- Attach the heat sink : This varies from heat sink to heat sink.
- Secure the motherboard into place by the screws provided and make necessary connections
- Install your hard drive : Plug the hard drive's sata cable into the sata slot on the motherboard
- Install any peripheral hardware such as
  graphic card
  network card
  etc
- Plug the monitor to the cpu and install the OS of your choice and you are good to go.

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

- We first describe the symptoms of malfunction and trouble shooting is the process of identifying the causes of these symptoms

• It is the process of elimination and usually requires confirmation that the solution restores the product to its working state.

• It demands critical thinking. A basic principle is to start from simplest and most probable problems first.

• Troubleshooting can also take the form of systematic checklist, procedure or flowchart made before one problem occurs.

• One of the core principles of troubleshooting is that reproducible problems can be reliably isolated and resolved.

• When talking about replacement, one must focus on 'adjustment or other modification" & not literally replace components.

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

1) Determine the mode and amount of RAM, the clock speed and memory.

- Disconnect cables : Unplug the Ac power card and disconnect all peripheral devices.
- Ground yourself : To discharge the built up static electricity.
- Check existing expansion slots : You will have to remove the old card, if no new slots are available.
- Uninstall old RAM : Remove by pushing outward on white ejector tabs
- Insert new memory : Insert the stick such that it is perpendicular to the motherboard
- Lock the memory stick into its place.
- Check the system properties and verify if everything is proper
- Close the cover properly.

RAMAIAH
Institute of Technology

Data Sheet

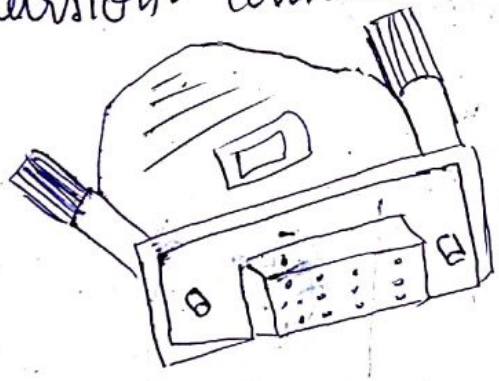
MARKS :

Name :	Kaushik Kampli	Branch:	CSE
USN/Roll No. :	1MS18CS057	Sem/Sec:	IV / B
Subject :	Computer Organization	Subject Code:	

3) Write a neat labelled diagram; explain different cables used to connect to the function units in a system.

Different cables are :-

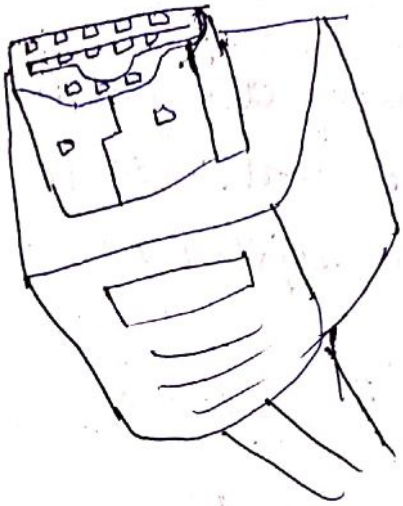
1) VGA cable :
Also known as D-sub cable, analog video cable. Connect one end to computer monitor, television. Connect other end to VGA port



(a)

(b)

2) HDMI cable : Used to transmit display and sound to TV saves to make TV as an external monitor with sound. checkers on TV.

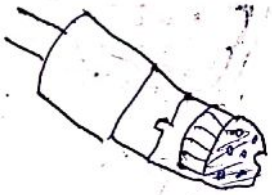


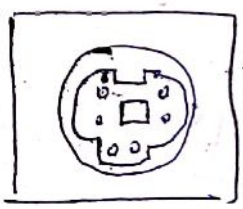
HDMI

3) PS/2 cable

Purple ps/2 cable : key board

Green PS/2 port : Mouse

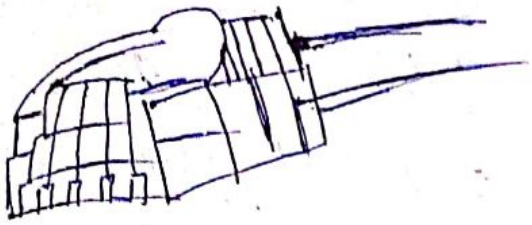




4) Ethernet (RJ45 cable)

One end to : router, network switch

other end to : ethernet port on computer.



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

1) Wear proper apparel: Avoid acrylic or wool sweaters when working with electronic parts.

2) Unplug all computer equipment and peripherals before opening any case (only exception is if you were working w/o any anti-static mat)

3) Retain all screws during disassembly

4) Do not forcefully remove components, as you may end up damaging the parts

5) Power supplies produce usual voltage levels.

6) Do not damage any cam or cover during disassembly, may cause problems during refitting of the system and its components.

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

Tutorial -II

Programme: B.E
Course: Computer Organization

Term: Jan to May 2020
Course Code: CS45

Name: Kaustuk Kamili	**Marks:** 10/10	**Date:** 4/1/2020
USN: 1MS18CS057	**Signature of the Faculty:**	

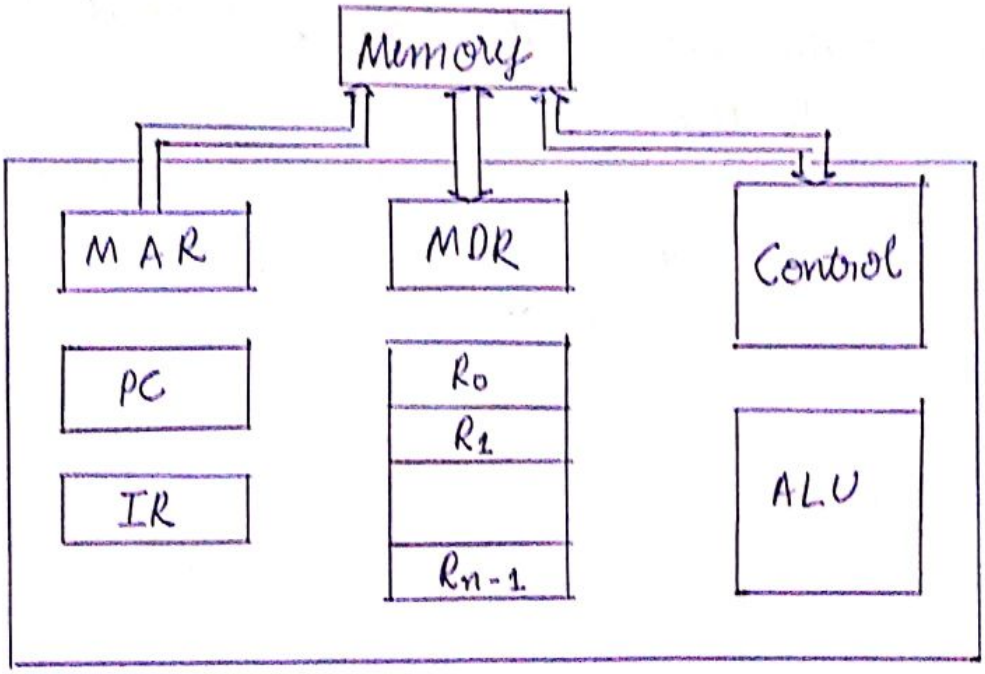
Activity II: Demonstrating Datapath and Instruction execution stages using MarieSim Simulator

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) The steps required to execute an instruction are:

1) Fetching an instruction : The instruction is fetched from main memory. It is fetched and stored in IR.

2) Decode instruction : During this cycle the encoded instruction is interpreted by the decoder in the IR

3) ALU operation: The operations will be carried out based on the operator and operands

4) Access Memory : Only 2 kinds of instructions access memory, LOAD copies a value from memory to AC and store copies a register value to memory.

5) Update Register File : Result of ALU operation is written back to the register file to update it.

6) Update the PC : The PC must be updated to the address of the next instruction, so that the process can be repeated.

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=b²-4ac

4. Describe the factors affecting the performance of a processor

The major factors affecting performance are:-

1) Design of the compiler: Speed is improved by pipelining, as simultaneous instructions can be executed.

2) The machine instruction set: The instructions used to write machine code also affect performance

3) Design of hardware: The speed at which memory is recalled by the processor through a bus reduces drastically if the copy of the memory is stored in cache. It is read directly from cache

Minor Factors are

1) Elapsed Time: Time taken for whole operation

2) Processor Time: Time during which the processor is active ($T = N \times S / R$)

3) Processor clock: Higher the clk rate, higher the no of basic steps per clock cycle.

3. Results and Snapshots:

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

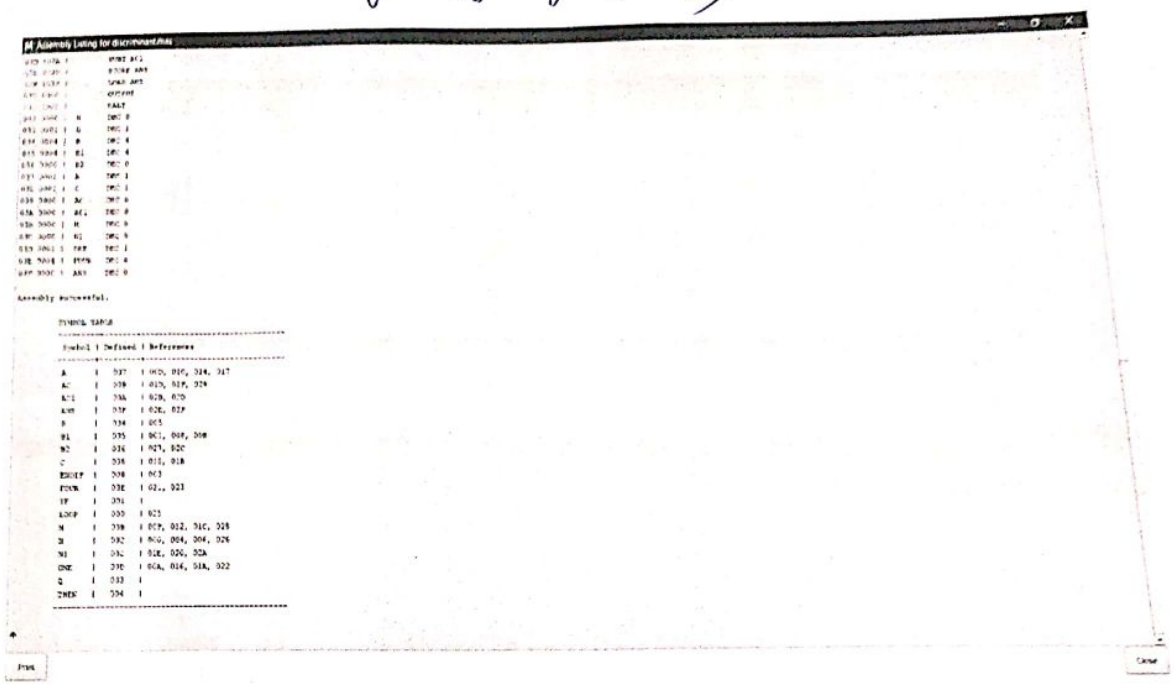
```
LOAD  G1
ADD   H
STORE A
LOAD  I
ADD   Y
STORE B
loop, LOAD  A
      ADD   F
      STORE F
      LOAD  B
      SUBT  one
      STORE B
      SKIPCOND 400
      JUMP  loop
LOAD  F
OUTPUT
HALT

G, DEC 9
H, DEC 5
Y, DEC 8
I, DEC 2
A, DEC 0
B, DEC 0
F, DEC 0
one DEC 1
```

ii) $b^2 - 4*a*c$

```
LOAD  B
STORE O
first, LOAD  B
       ADD   X
       STORE X
       LOAD  O
       SUBT  one
       STORE O
       SKIPCOND 400
       JUMP  first

record, LOAD  A
        ADD   Y
        STORE Y
        LOAD  C
        SUBT  one
        STORE C
        SKIPCOND 400
        JUMP  record

third, LOAD  four
       ADD   Z
       STORE Z
       LOAD  Y
       SUBT  one
       STORE Y
       SKIPCOND 400
       JUMP  third
LOAD   D
ADD    X
SUBT   Z
OUTPUT
HALT
```

1

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

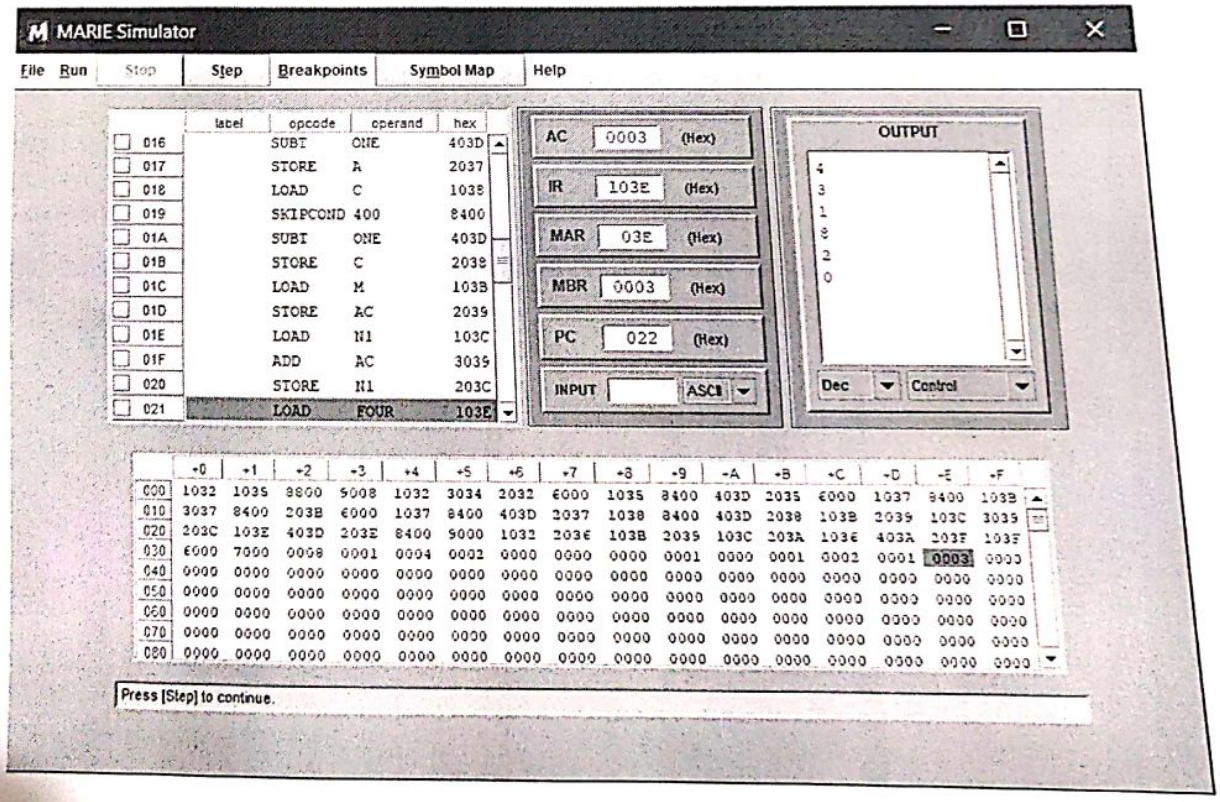


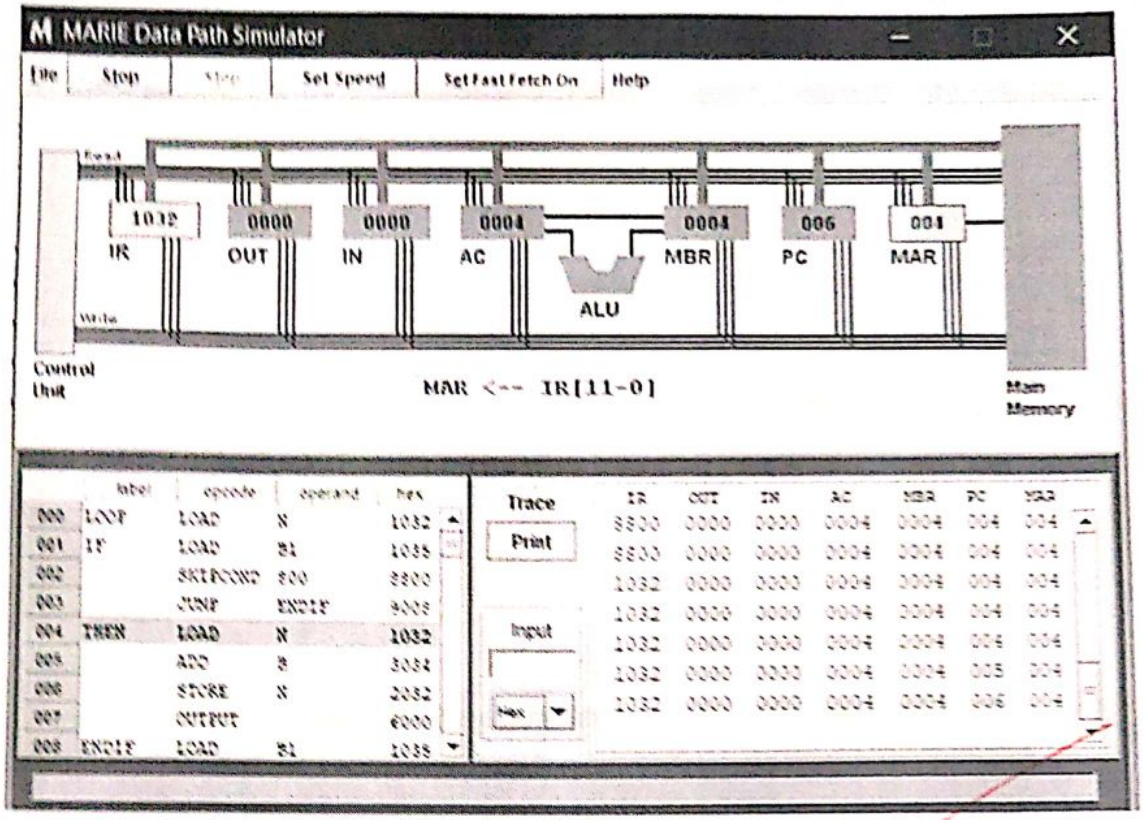
Assembly successful.

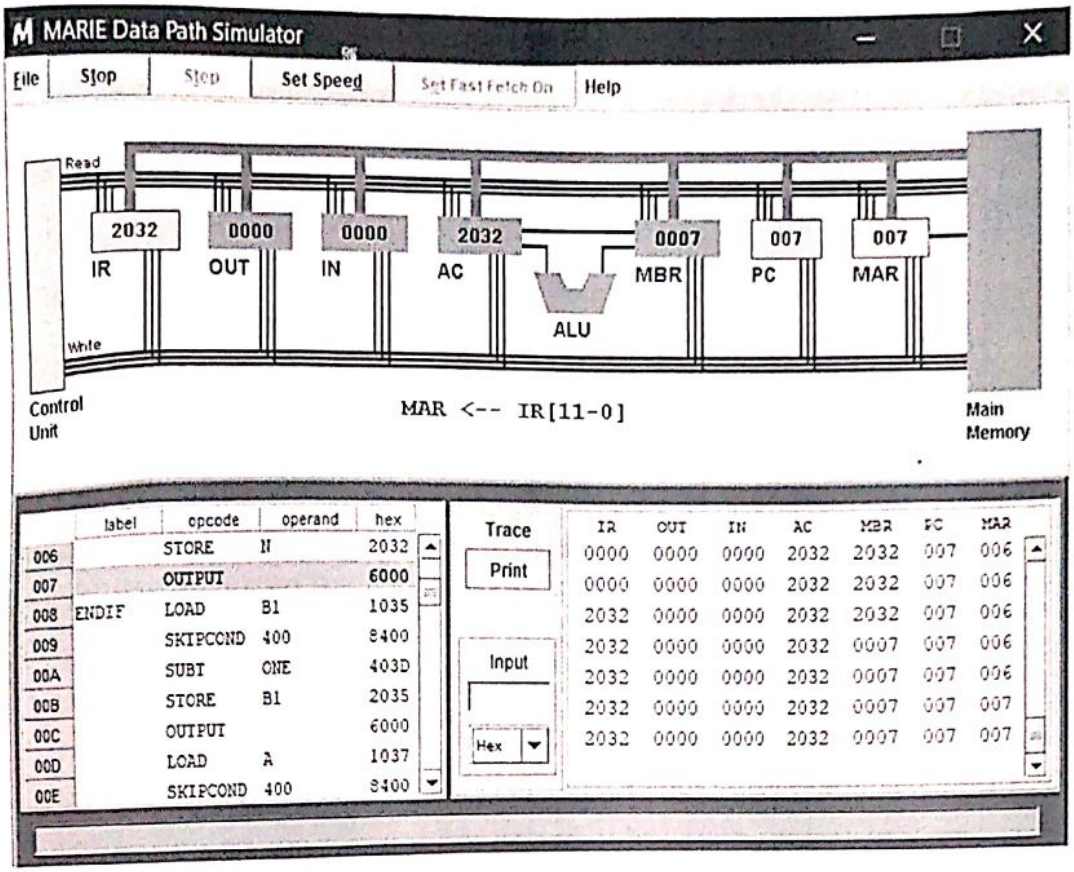
SYMBOL TABLE

Symbol	Defined	References
A	037	00D, 010, 014, 017
AC	039	015, 01F, 029
ADD1	03A	02B, 02D
ADR	03F	02E, 02F
B	034	005
B1	035	001, 008, 00B
B2	036	007, 02C
C	038	011, 018
ENDIF	00B	003
FOUR	03E	021, 023
IF	001	1
LOOP	000	025
M	03B	00F, 012, 01C, 028
N	032	00G, 004, 006, 026
N1	03C	02E, 02G, 02A
ONE	03D	00A, 016, 01A, 022
Q	033	1
TWEN	004	1

Print... Close







$d = b^2 - 4*a*c$

MARIE Assembler Code Editor

File Edit Assemble Help

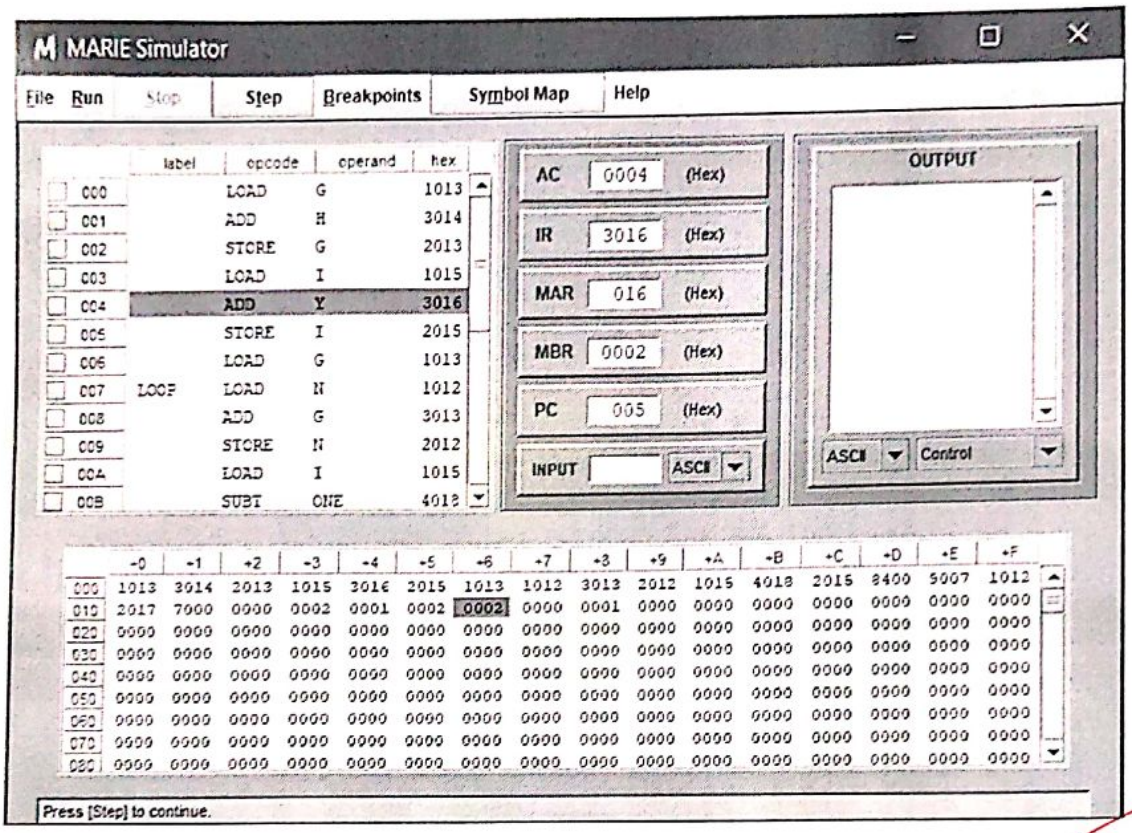
```
LOAD G
ADD H
STORE G
LOAD I
ADD Y
STORE I
LOAD G
LOOP,    LOAD N
         ADD G
         STORE N

         LOAD I
         SUBT ONE
         STORE I

         SKIPCOND 400
         JUMP LOOP

LOAD N
STORE F
HALT
N,       DEC 0
G,       DEC 1
H,       DEC 1
I,       DEC 2
Y,       DEC 2
F,       DEC 0
ONE,     DEC 1
```

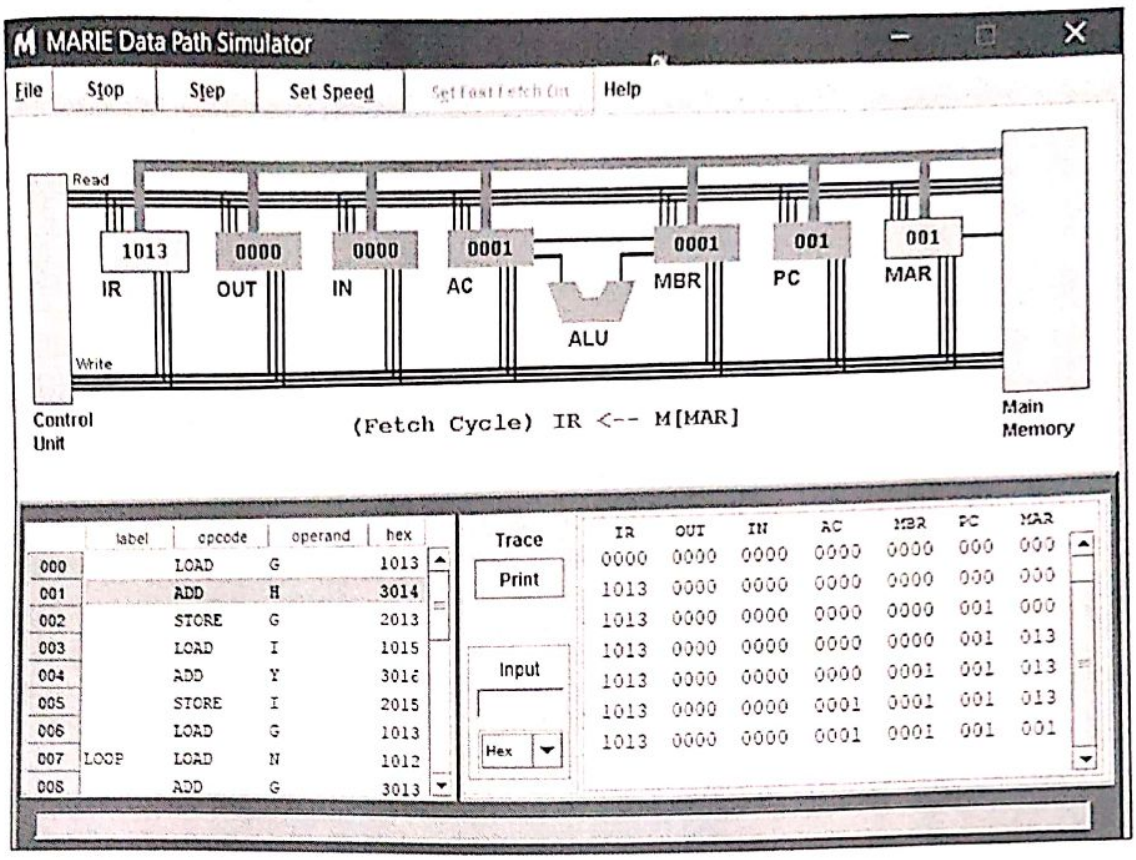
C:\Users\Manas Shankar\Desktop\MarieSim\multAdd\multAdd.mas Assembly successful.

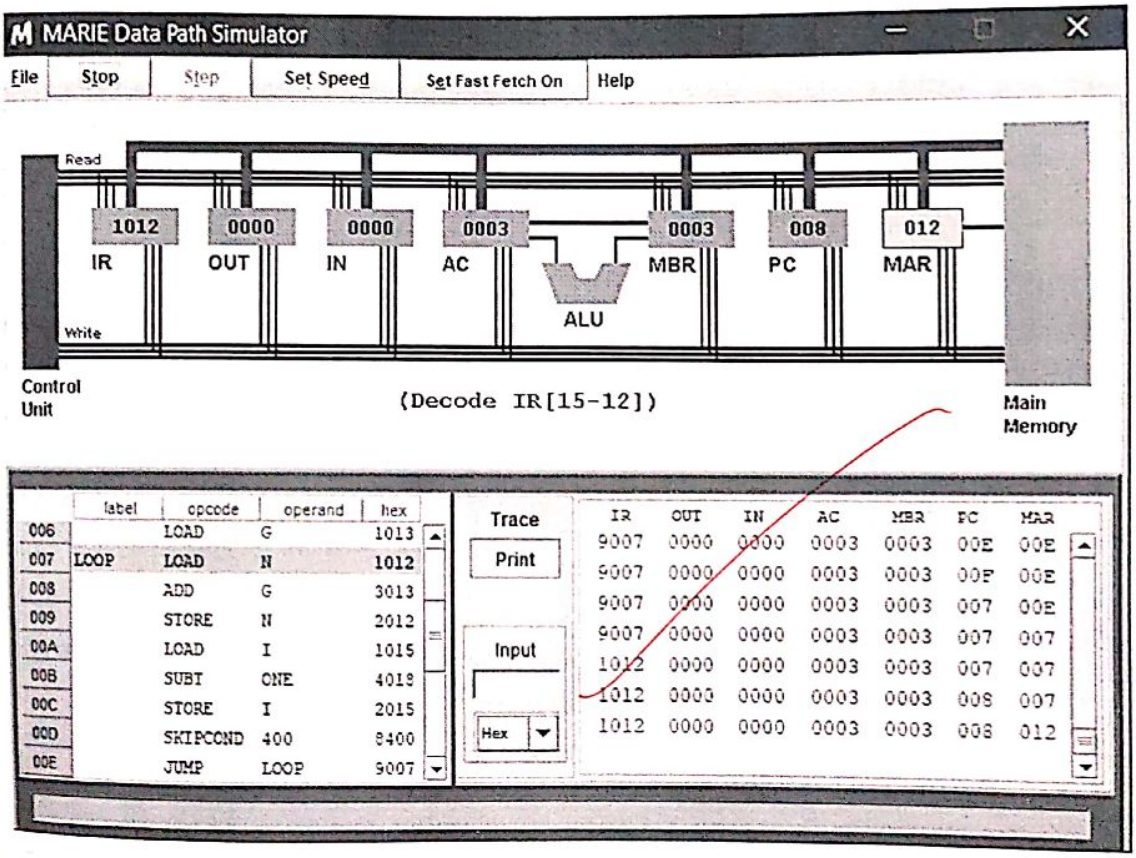
Assembly Listing for multAdd.mas

```
000 1013 |        LOAD G
001 3014 |        ADD H
002 2013 |        STORE G
003 1015 |        LOAD I
004 3016 |        ADD Y
005 2015 |        STORE I
006 1013 |        LOAD G
007 1012 | LOOP   LOAD N
008 3013 |        ADD G
009 2012 |        STORE N
         |
00A 1015 |        LOAD I
00B 4018 |        SUBT ONE
00C 2015 |        STORE I
         |
00D 8400 |        SKIPCOND 400
00E 9007 |        JUMP LOOP
         |
00F 1012 |        LOAD N
010 2017 |        STORE F
011 7000 |        HALT
012 0000 | N      DEC 0
013 0001 | G      DEC 1
014 0001 | H      DEC 1
015 0002 | I      DEC 2
016 0002 | Y      DEC 2
017 0000 | F      DEC 0
018 0001 | ONE    DEC 1

Assembly successful.

      SYMBOL TABLE
      ------------------------------------------------------
      Symbol | Defined | References
      -------+---------+-----------------------------------
      F      |  017    | 010
      G      |  013    | 000, 002, 006, 008
      H      |  014    | 001
      I      |  015    | 003, 005, 00A, 00C
      LOOP   |  007    | 00E
      N      |  012    | 007, 009, 00F
      ONE    |  018    | 00B
      Y      |  016    | 004
      -------+---------+-----------------------------------
```

Print Close







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

Tutorial -III

Programme: B.E
Course: Computer Organization Course Code: CS45

Term: Jan to May 2020

Name: Kaushik Kamali	**Marks:** 10 /10	**Date:**
USN: 1MS18CS057	**Signature of the Faculty:**	19/2/2020

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 :	Kaushik Kamphli	Branch:	CSB
USN/Roll No. :	1MS18CS057	Sem/Sec:	IV / B
Subject :	Computer Organization	Subject Code:	

Tutorial - III

1) ARM program to perform basic arithmetic operations

```
MOV  R1, #0x00000020
MOV  R2, #2
MOV  R3, #1
MOV  R4, #1
ADD  R1, R1, R2
ADD  R3, R3, R4
SUB  R1, R1, R3

SWI  0x11
```

2) ARM program to demonstrate load and store instructions

```
MOV R1, #0x00000030
MOV R3, #0
MOV R4, #5
```

```
STR  R4, [R1, R3]
LDR  R5, [R1, R3]

SWI  0x11
```

3) Write an ARM program to evaluate expression
$f = (g+h) - (i+j)$

```
MOV  R3, #41
MOV  R4, #45
MOV  R7, #35
MOV  R8, #38

LDR  R1, =0x00000064
LDR  R2, =0x00000068
LDR  R9, =0x00000074
LDR  R10, =0x00000078

STR  R3, [R1]
STR  R4, [R2]
STR  R7, [R9]
STR  R8, [R10]

LDR  R5, [R1]
LDR  R6, [R2]
LDR  R11, [R9]
LDR  R12, [R10]

ADD  R5, R5, R6
ADD  R11, R11, R12
SUB  R0, R5, R11

SWI  0x11
```

4) ARM program to find the sum of all elements in an array

```
    MOV R0, #5
    LDR R1, =array
loop: LDR R2, [R1], #4
    ADD R3, R3, R2
    SUB R0, R0, #1
    CMP R0, #00
    BNE loop

    SWI 0x11
    .data
array: word 0x00000001, 0x0000000A, 0x000000AB
       0x0000000B, 0x00000008
```

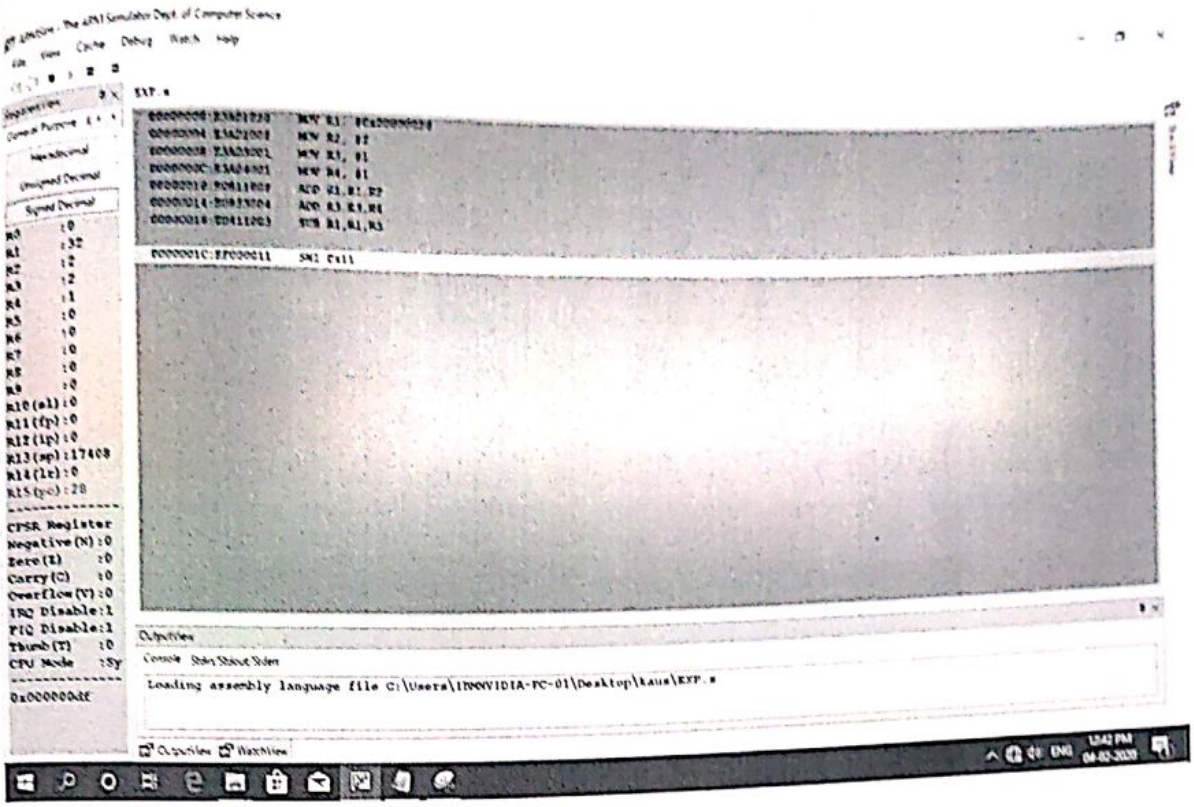
5) write an ARM program to find the factorial of a number.

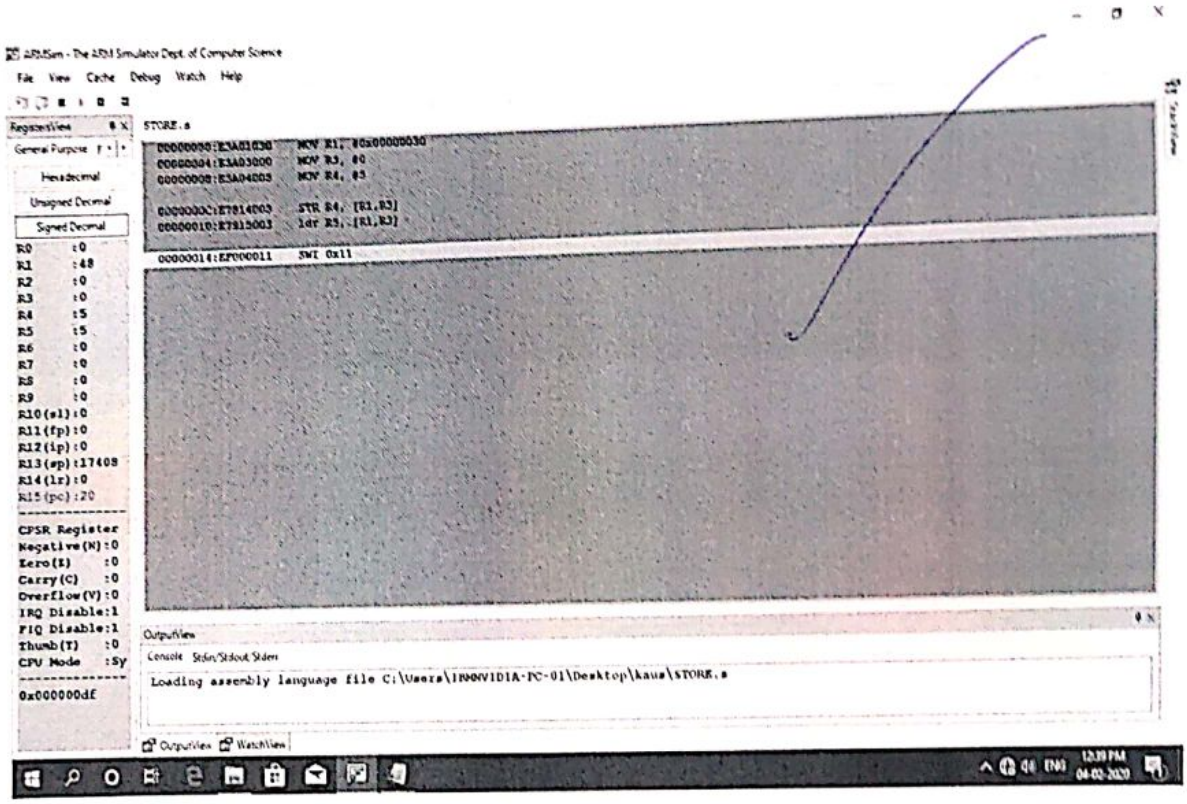
```
    LDR R1, =array
    LDR R2, [R1]
    LDR R3, [R1]
loop: SUB R2, R2, #1
    CMP R2, #0
    BEQ end
    MUL R0, R2, R3
    MOV R3, R0
```

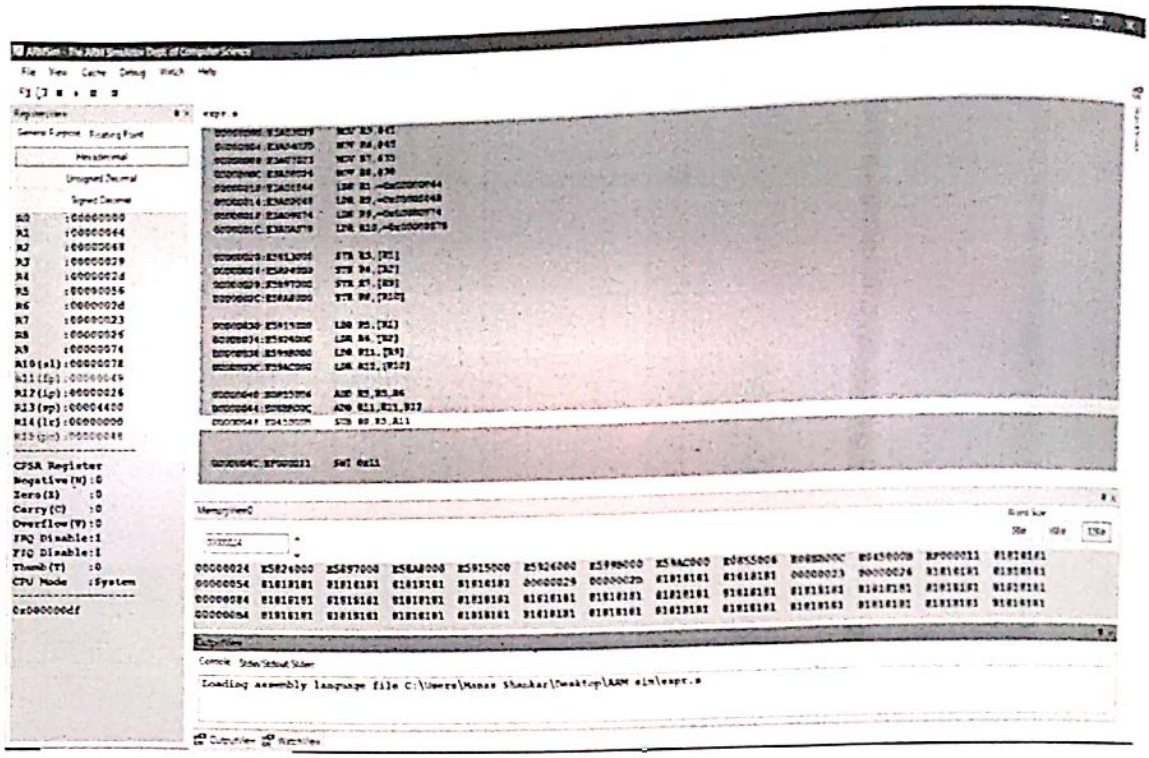
```
CMP R2, #0
BNE loop

SWI 0x11
```

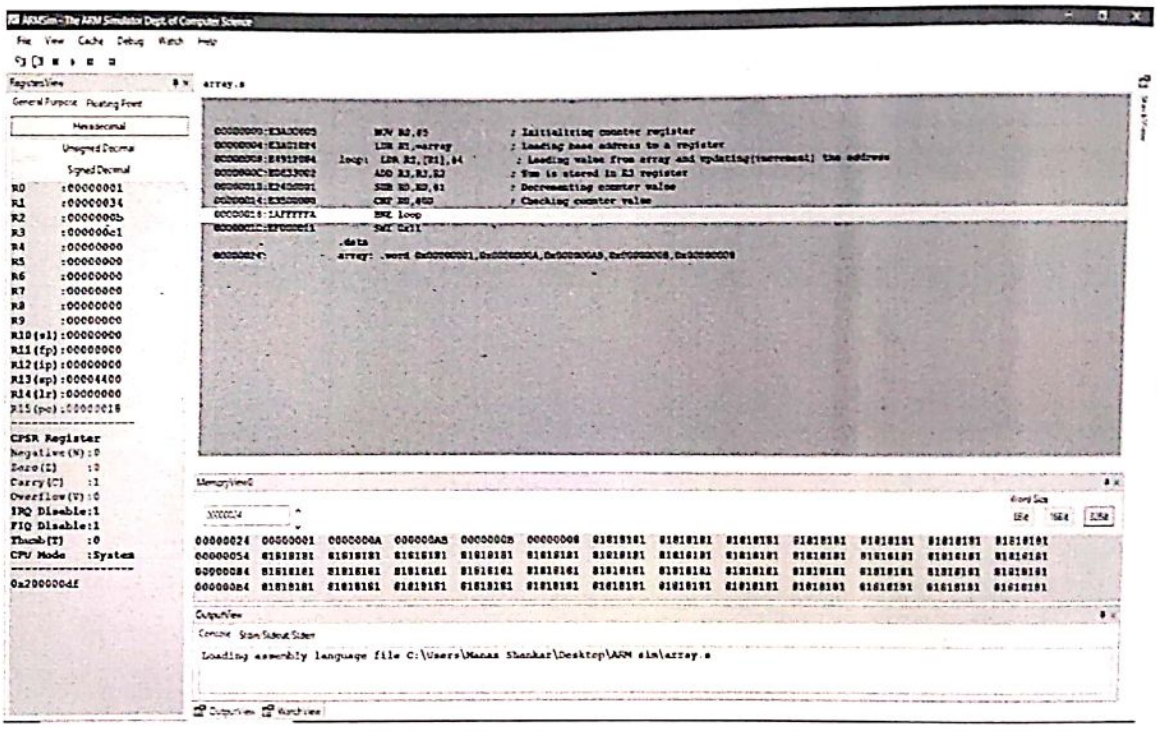
- data

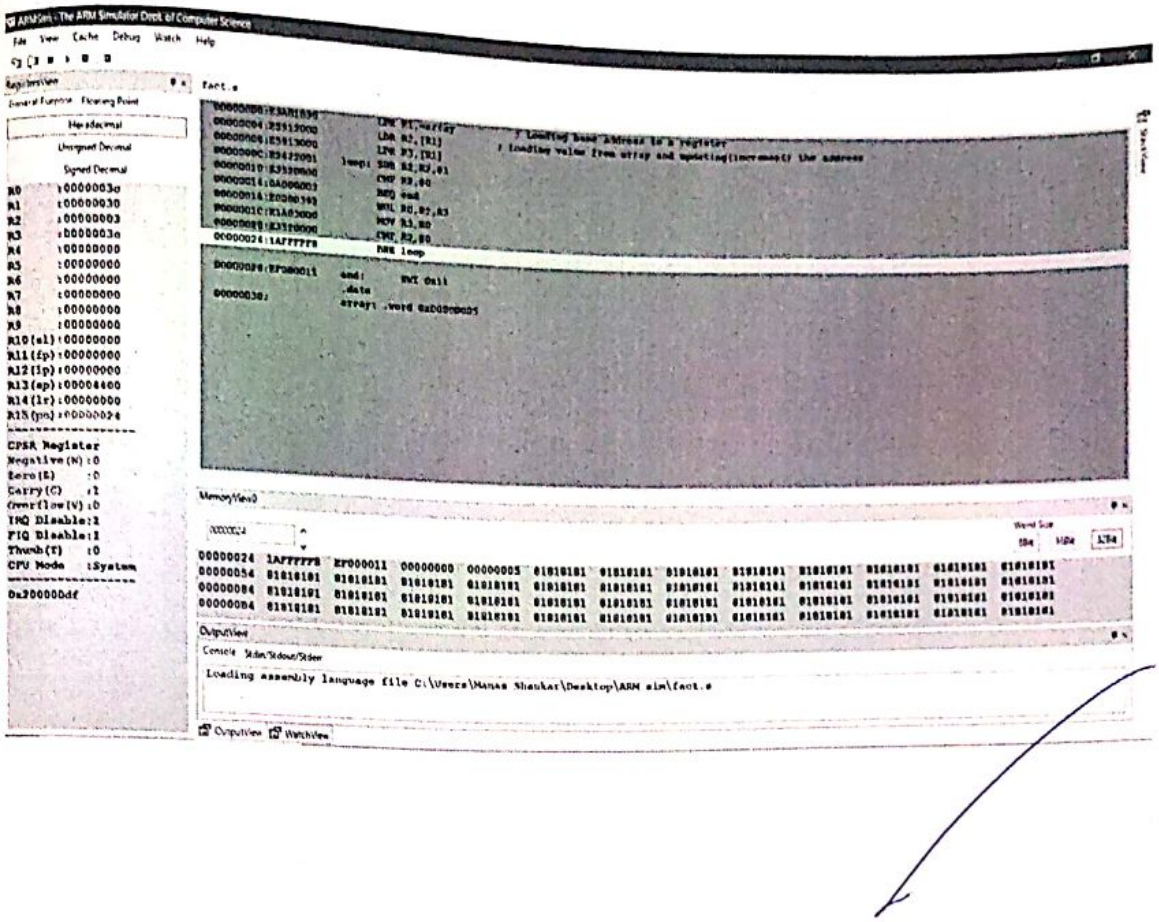
```
fact: .word 0x00000005
```











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

Department of CSE

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

Activity IV: Executing ARM programs using ARMsim simulator.

Name: Kaushik Kamath	**Marks: /10**	**Date:** 11/02/2020
USN: 1MS18CS057	**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 :	Kaushik Kamplí	Branch:	CSE
USN/Roll No. :	1MS18CS057	Sem/Sec:	IV / B
Subject :	CO Lab IV	Subject Code:	

ARM Program to generate Fibonacci sequence

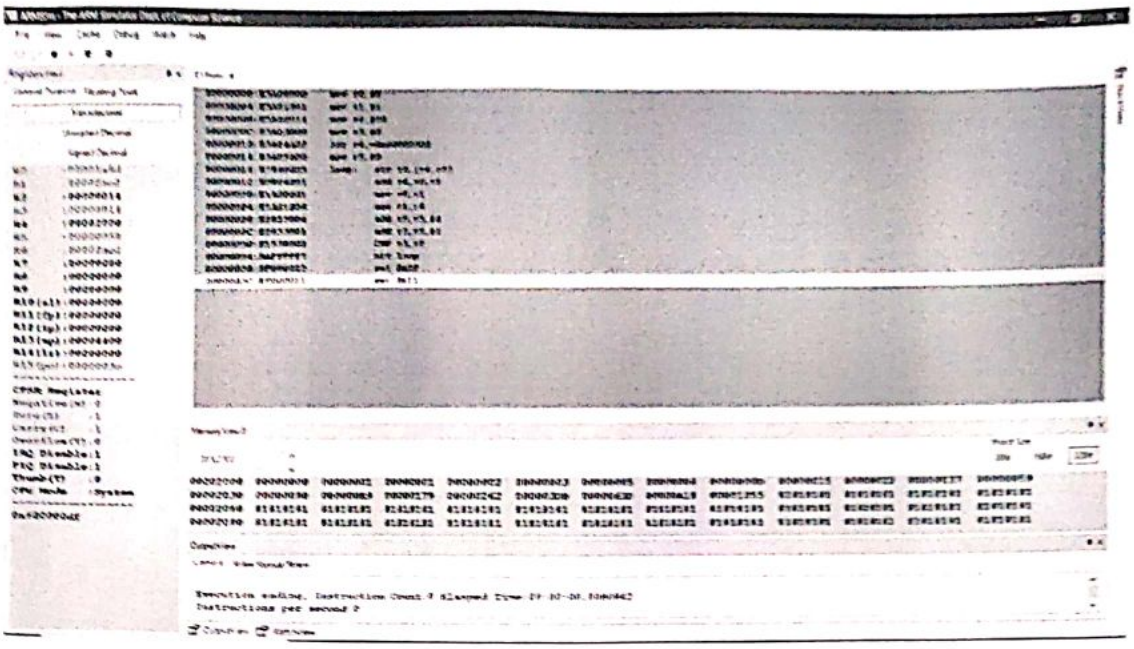
```
mov  r0,#0
mov  r1,#1
mov  r2,#20
mov  r3,#0
ldr  r4,=0x00002000
mov  r5,#0
loop: str  r0,[r4,r5]
      add  r6,r0,r1
      mov  r0,r1
      mov  r1,r6
      add  r5,r5,#4
      add  r3,r3,#1
      cmp  r3,r2
      blt  loop
      swi  0x22
      swi  0x11
```

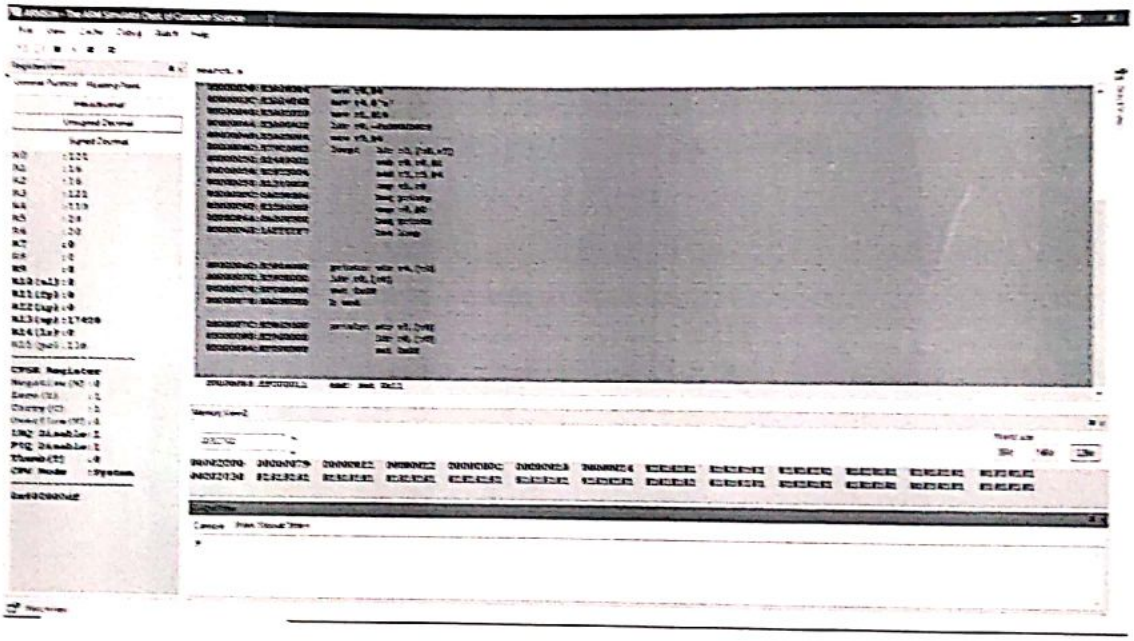
Program to match an element in an array

```
mov  r8, #4
mov  r4, #'n'
mov  r1, #16
ldr  r0, = 0x00002000
mov  r5, #4
loop :   ldr  r2, [r0, r5]
         sub  r8, r8, #1
         add  r5, r5, #4
         cmp  r1, r2
         beq  print1
         cmp  r8, #0
         beq  printn
         bne  loop
print1 : str  r4, [r0]
         ldr  r0, [r0]
         swi  0x00
         b end
printn : str  r3, [r0]
         ldr  r0, [r0]
         swi  0x00
```

Program to copy a string

```
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

li:
      ldrb r6, [r0, r7]
      strb r6, [r5, r8]
      add r7, r7, #1
      add r8, r8, #1
add r9, r9, #1
cmp r9, #5
bne li
swi 0x11
```





Lab 5

Kaushik Kamplí
1MS18CS057
IV 'B'

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

Objective : To simulate working of ALU using simulator.

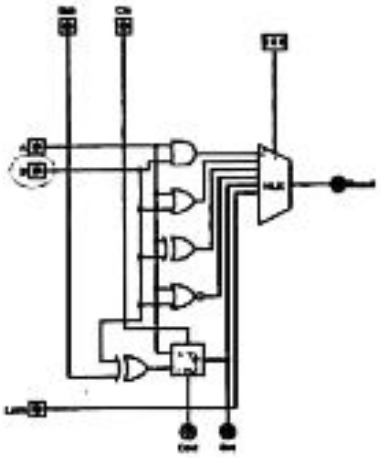
Activity to be performed by students :

List out the steps in designing ALU :

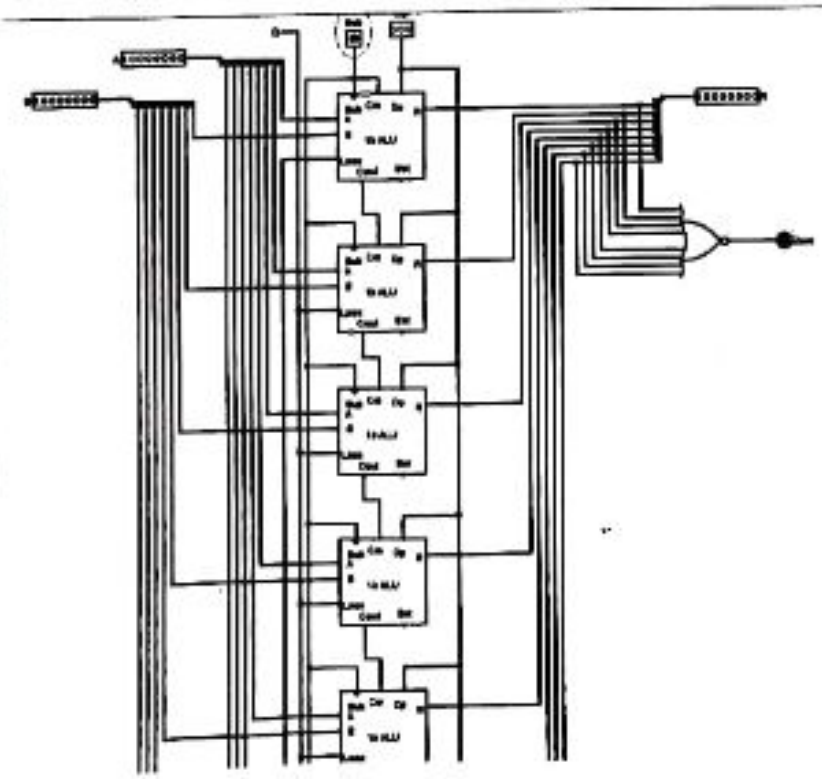
1. Add the two i/p pins. Name them A and B.
2. Add OR, AND, XOR, NOR gates and a 1-bit adder.
3. Connect the A's and B's of all the gates to their respective pins
4. Add an output pins and name it Result.
5. Add a 1-bit multiplier with 3 select bits
6. Connect outputs of all gates to the MUX
7. Connect 3-bit i/p pin to MUX
8. Add i/p pin to C_{in} and o/p pin to Cout.
9. Add an XOR gate, connect its o/p to Cout. The first i/p must be connected to B and second to another i/p pin sub.
10. Add another i/p and name it dec. Connect it to the MUX
11. Add an output pin and name it ut, connect it to the o/p of adder unit.

Snapshots :

1-bit Arithmetic Logic Unit (ALU)



8-bit Arithmetic Logic Unit (ALU).



Scanned with CamScanner

Lab - 6

Kaushik Kamplí
1MS18CS057
IV 'B'

Activity VI : To stimulate writing operation on memory, and hence to design a memory system using logisim simulator.

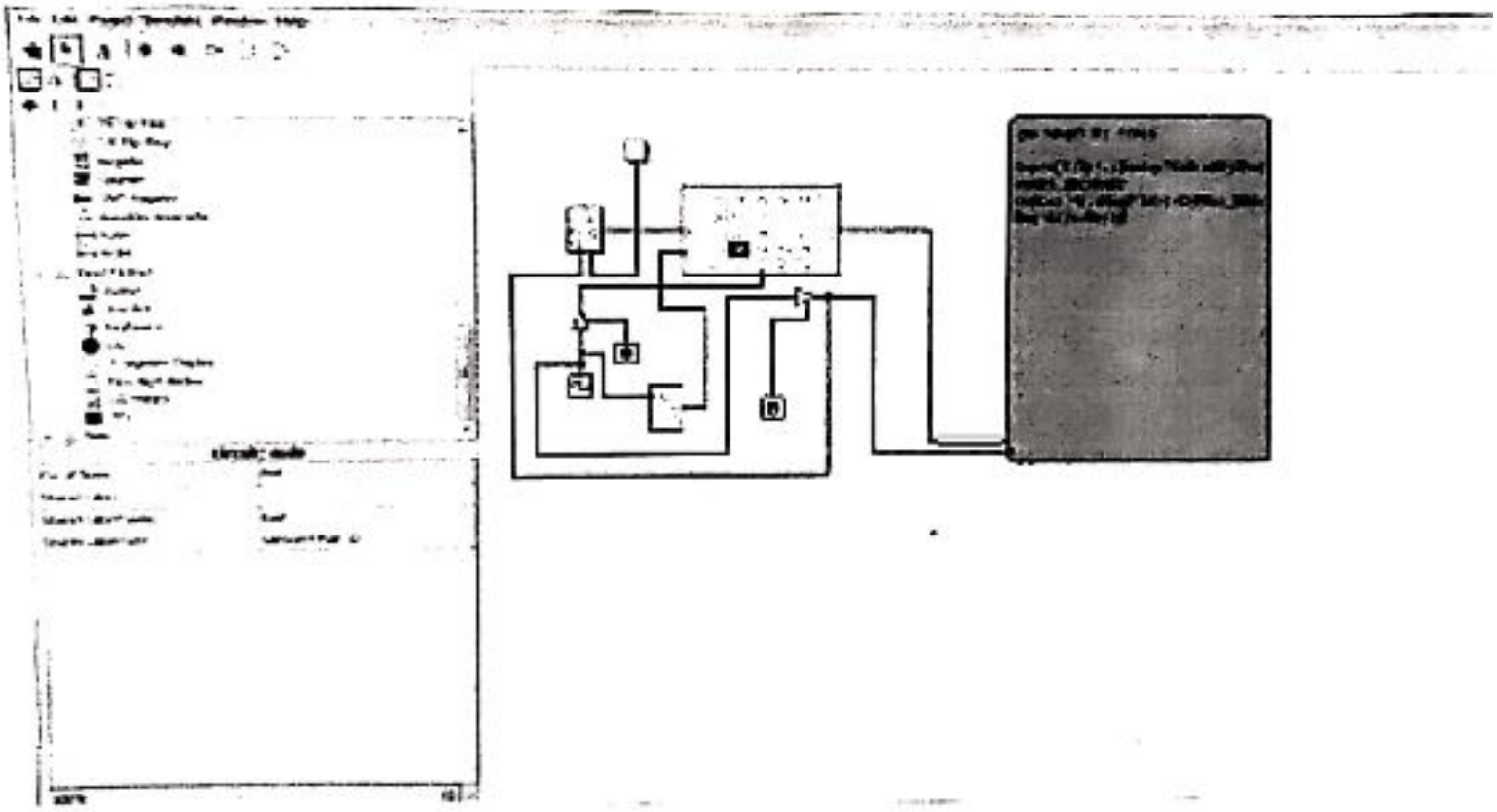
Objective : To stimulate writing operation on memory.

Activity to be performed by students :

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

Observations and Snapshots:



Scanned with CamScanner