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: Isha Gupta	**Marks:** 10/10	**Date:**
USN: 1MS18CS049	**Signature of the Faculty:**	4/5/2020

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:

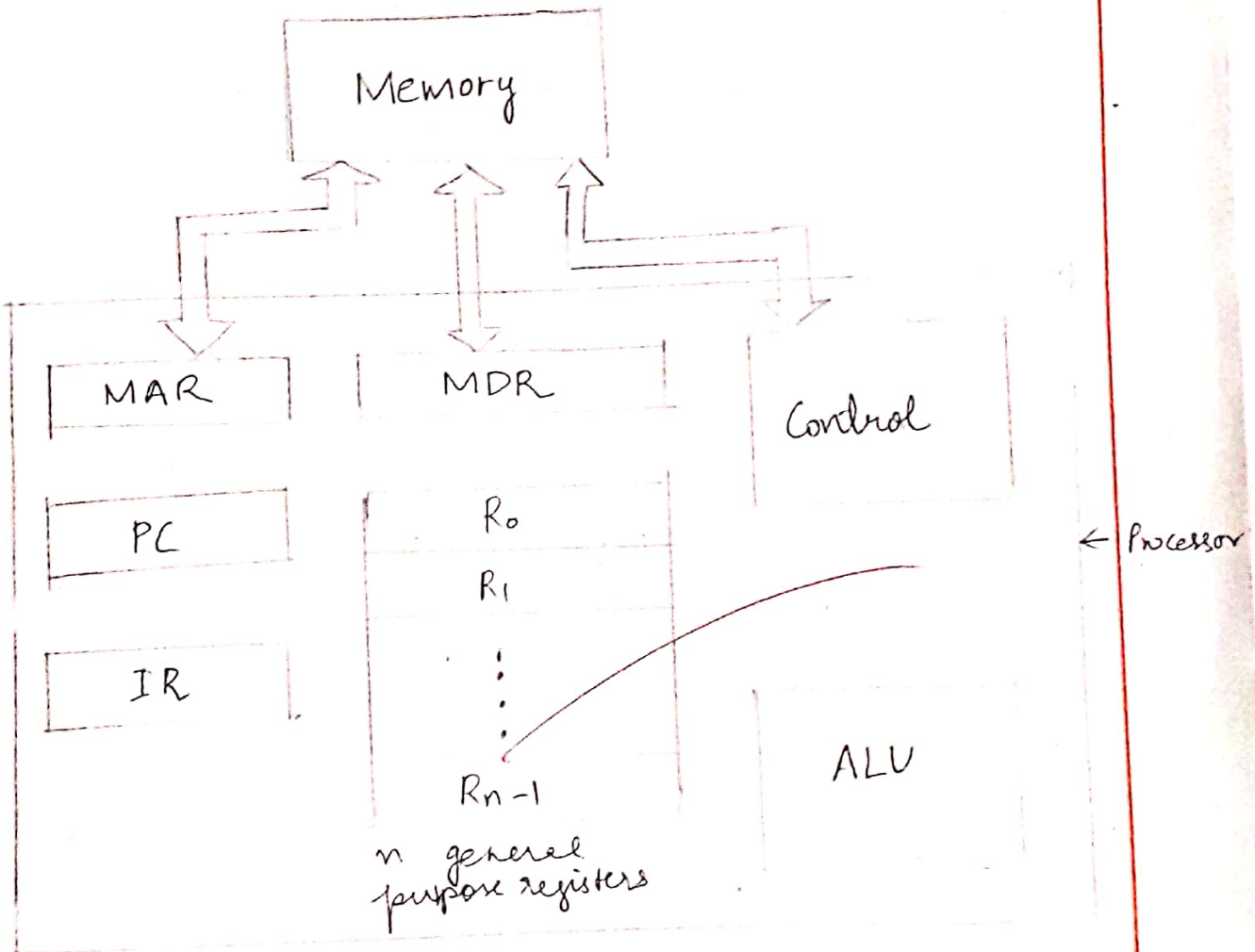
RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	Isha Gupta	Branch:	CSE
USN/Roll No. :	1MI18CS049	Sem/Sec:	IV / B
Subject :	CO lab	Subject Code:	

1) Interconnection between memory & a processor



2) Steps required to execute an instruction:

(i) Fetching ~~the~~ instruction ~~and operand~~
(ii) Fetching operands
(iii) Decoding the instructions
(iv) Performing ALU operation and executing instructions.
(v) Accessing memory.
(vi) Storing the output & updating the re file.
(vii) Updating the Program Counter (PC)

3) (i) LOAD G
ADD H
STORE A
LOAD I
ADD Y
STORE B
LOOP, LOAD H
ADD A
STORE N
LOAD B
SUBT O
STORE B
SKIPCOND 400
JUMP LOOP

```
LOAD  N
OUTPUT
HALT
G,   DEC  5
H,   DEC  10
I,   DEC  5
Y,   DEC  10
A,   DEC  0
N,   DEC  0
O,   DEC  1
B,   DEC  0
```

$d = b^2 - 4ac$

Ans:

```
LOAD  B
STORE D
```

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

second,
```
LOAD A
ADD  Y
STORE Y
LOAD  C
SUBT one
STORE  C
SKIPCOND  400
```

```
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
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
```

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :		Branch:	
USN/Roll No. :		Sem/Sec:	
Subject :		Subject Code:	

Factors affecting the performance of a processor are:

(i) Design of the hardware - The speed with which a compiler executes program is affected by the design of its hardware and its machine language instructions. For best performance, it is necessary to design the compiler, the machine instruction set, and the hardware in a coordinated way.

The processor time depends on the hardware involved in the execution of individual machine instructions. This Hardware comprises the processor and the memory, which are usually connected by a bus.

Basic performance Eqn. $T = \frac{N \times S}{R}$

where T = processor time
N = No. of machine language instructions
R = clock rate (in cycle/sec)
S = No. of basic steps to execute instruction

For high performance, reduce T (by reducing N & S, increasing R)

Pipelining

Pipelining & Superscalar operation :-

Pipelining is the overlapping execution of several instructions. This reduces no. of clock cycles. We can achieve a higher degree of concurrency. Parallel paths are created. Execution of several instructions in every clock cycle is called Superscalar execution.

CISC and RISC :-

RISC :- allows simple instructions which requires small no of basic steps to execute. A program will have more no. of instructions i.e. N = max, S = min.

CISC :- Instructions are complex, more no. of basic steps to execute. A program will have less no. of instructions i.e. N = min, S = max

Compiler :- Translates high-level language program into a sequence of machine instructions. Optimizing compiler reduces N × S, may rearrange program instructions, may 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 best results.

Processor clock :- Processor circuits are controlled by a clock. It is a timing signal clock defines regular intervals called clock cycles. To execute a machine instruction the processor divides the action to be performed into a series of steps each completed in 1 clock cycle. Length P of 1 clock cycle is important for processor performance, whose inverse is the clock rate. $R = 1/P$ Cycles/Sec is called Hertz, Mega - million, Giga - billion

Clock rate :- 2 possibilities to increase clock rate (R). Improving IC tech makes logic circuits faster, which reduces time to execute each step. P reduces, R is increased. Secondly, reducing amount of processing also works.

Performance measurement :-

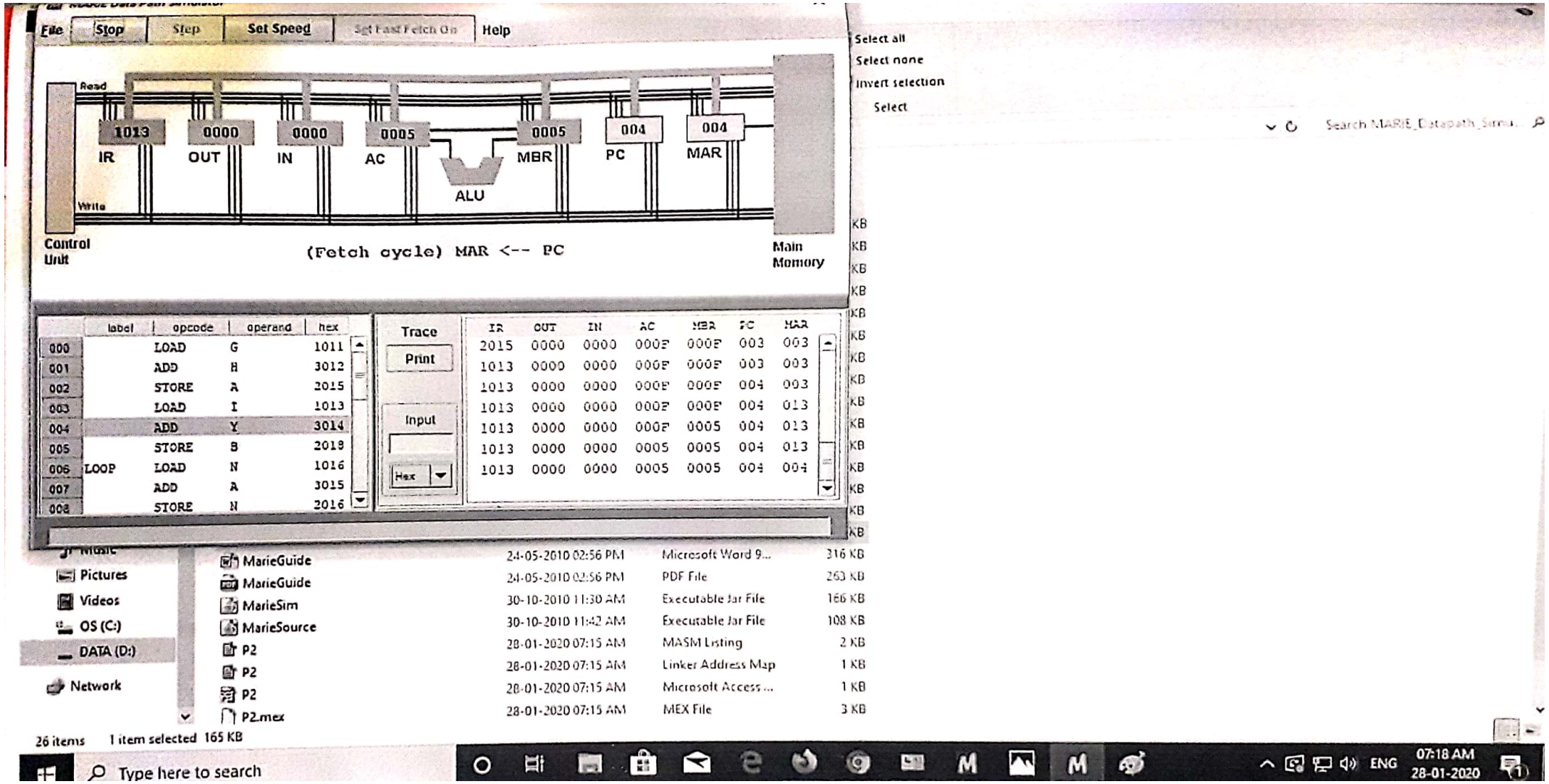
Used to evaluate effectiveness of new features, used in marketing process, to choose Computer models. It is done using time taken to execute Benchmark programs.

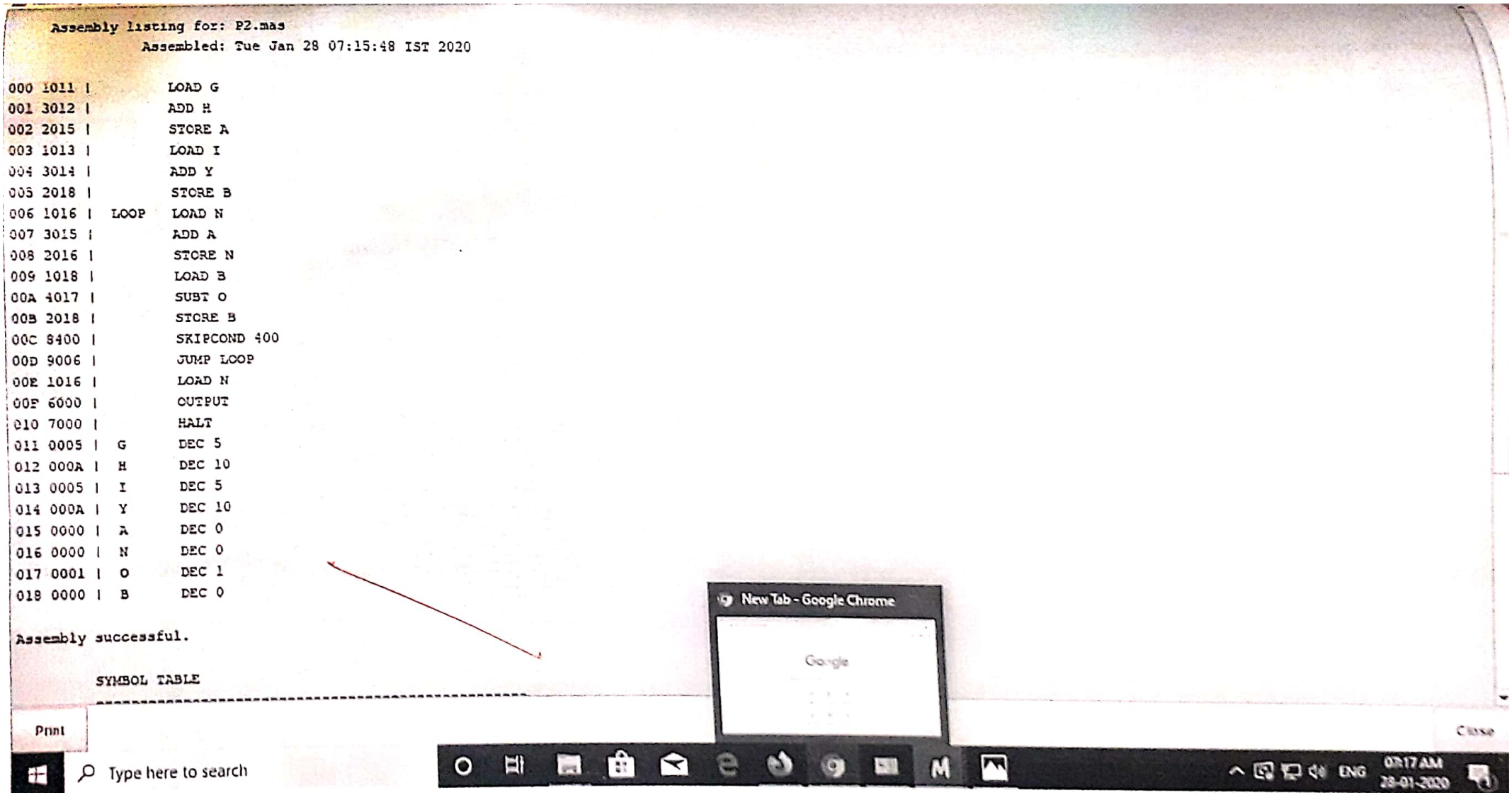
SPEC (Specific performance Evaluation Corporation) :- Running Time on reference computer / Running Time on test computer

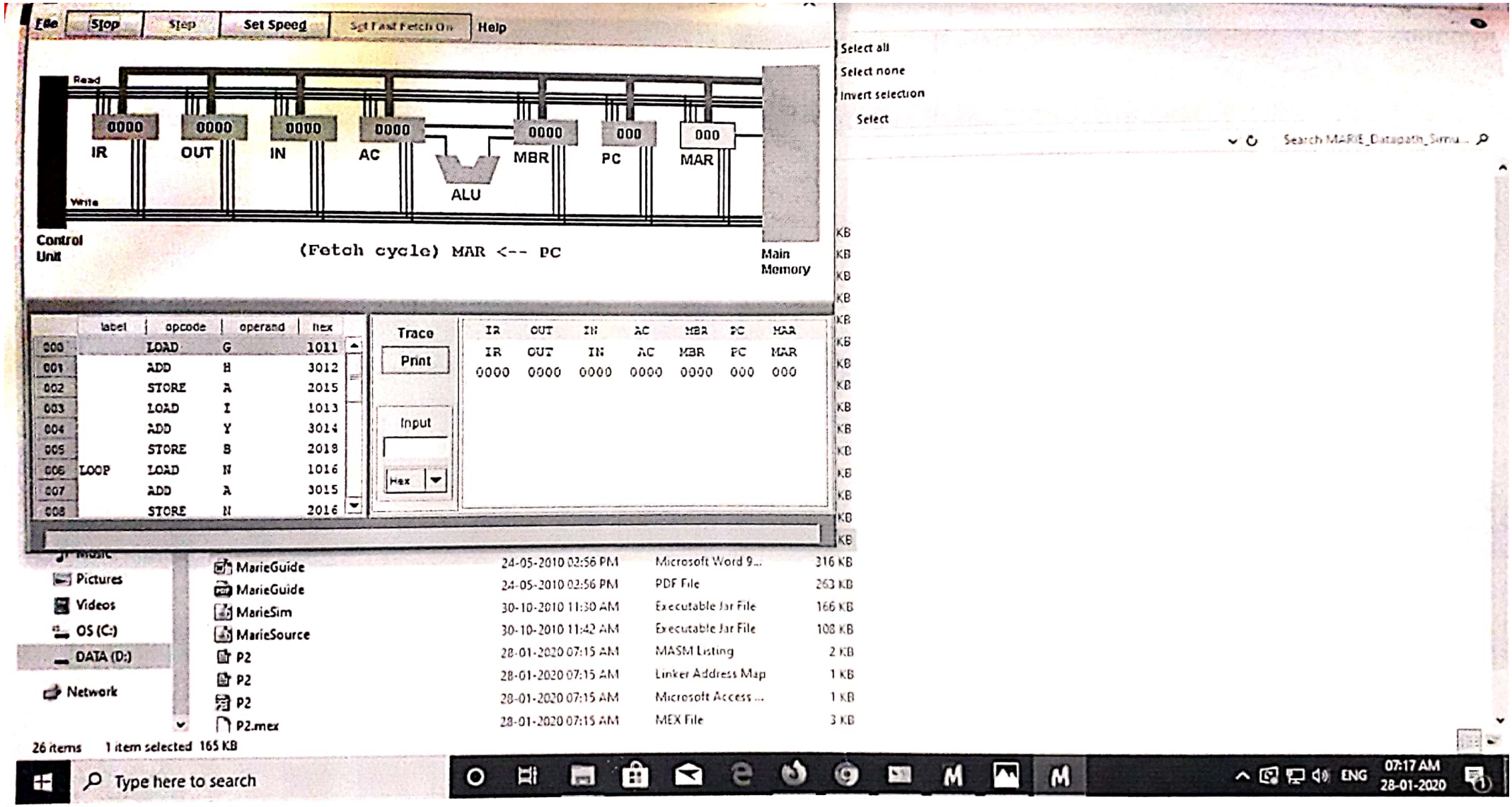
$$= \prod_{i=1}^{n} SPEC_i$$

```
LOAD G
ADD H
STORE A
LOAD I
ADD Y
STORE B
LOOP,  LOAD N
       ADD A
       STORE N
       LOAD B
       SUBT O
       STORE B
       SKIPCOND 400
       JUMP LOOP
LOAD N
OUTPUT
HALT
G, DEC 5
H, DEC 10
I, DEC 5
Y, DEC 10
A, DEC 0
N, DEC 0
O, DEC 1
B, DEC 0
```

Scanned by CamScanner







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: ISHA GUPTA	**Marks:** 10/10	**Date:**
USN: 1MS18CS049	**Signature of the Faculty:**	4/1/20

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 :	Isha Gupta	Branch:	CSE
USN/Roll No. :	1MS18CS049	Sem/Sec:	IV / B
Subject :	CO Laboratory	Subject Code:	

1) Write down the detailed procedure to assemble a system

Step 1: Remove side panels on case After removing the case from the panels box, the panels are removed from this case with thumb screws

Step 2: Insert motherboard - Before selling the board in, the I/O panel faceplate needs to be snapped into the location in the back of the case Once the board is resting in the case, line up the first hole

Step 3: check clearances Being that the computer includes high performance components, some of them are large enough that clearance can become an issue

Step 4: Front panel connections - Attach the connectors for the buttons, lights, USB ports and audio connections

Step 5: Install power supply. Cables that are needed are plugged into the unit

Step 6: Power motherboard - The largest motherboard power cable is to be connected

Step 7: - Installing optical drive. - The optical drive for the computer is a DVD/CD read/write combo.

Step 8: - Installing the hard drives. Components uses 4 drives, 2 in raid and rest for a main drive and miscellaneous storage

Step 9: Connect cables - Connect the cables to the hard drives and optical drives. The cables are

keyed so they only flow in one direction into the board

Step 10:- Installing RAM :- The slots are keyed as are the RAM sticks, so make sure the notch is lined up

Step 11:- Install graphic cards and expa cards. The network card and audio card for the comput are connected into the slots below the graphics ca

Step 12:- Cable management - Organising and hidin cables for high airflow and security / safety.

Q2) Explain how troubleshooting a system helps to trace and correct the false in the system.

Ans) Troubleshooting is a form of problem solving often applied to repair failed product or proces 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. Troublesho is needed to identify the symptoms. Determinin the most likely cause is a process of elimination eliminating potential causes of a problem. Finally, troubleshooting requires confirmation that the solution restored the product or process to its working.

Step 1:- Identify the problem.
Step 2: Establish a theory of probable 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.

Data Sheet

RAMAIAH
Institute of Technology

MARKS :

Name :		Branch:	
USN/Roll No. :		Sem/Sec:	
Subject :		Subject Code:	

Step 6:- Document findings, actions and outcomes
List out the procedure to install extra memory card to a system.
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 ~~access to the interior and~~ slots. They're most commonly found next to the processor and its cooler. If there's already RAM in your system, eject it by pressing firmly on the tabs on the motherboard at either end of the slots. Then memory sticks will pop out and you can remove them gently.
Step 3: To install the new RAM, line up the notches in the bottom of the sticks with gaps in the slot on the motherboard. Make sure the wings at either end of the slot are pushed back, so that they are tilted away from the RAM. As it does the wings will clamp in and hold the memory securely.
Step 4: Once the sticks have clicked into confirm that the wing clips are locked in to hold the sticks firmly in their slots and then close the PC back up. Plug all the cables back in and try to boot the system.

4) With a neat diagram, explain different cables 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 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 ~~sdd~~ port on computer.

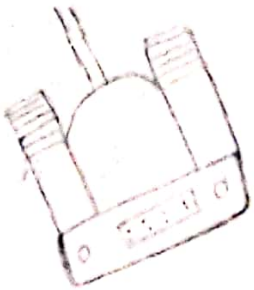
4. PS/2 cable:- Connect one end to PS/2 keyboard, PS/2 mouse. Connect other end to PS/2 ports on computer. Purple PS/2 port: keyboard. Green PS/2 port: mouse.

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 Cord (Kettle plug) connect one end to AC power socket. Connect other end to power supply unit, Computer monitor.

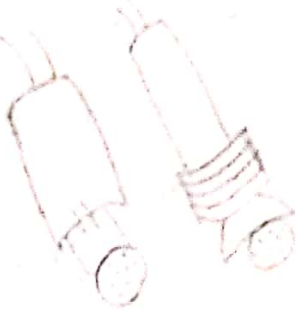
1) VGA cable



2) DVI cable

3) HDMI cable

4) PS/2 cable



5) Ethernet cable

6) USB cable

7) Computer Power cord (Kettle Plug)

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

Ans A few warnings and reminders before you start disassembling your computer tower keep both ur unit and urself safe.

1) Fully shut down and unplug the computer before you make any attempts to disassemble 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 with labels for each part is ideal to avoid confusion between the similar looking screws.

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

If a component does not come out easily, do not forcefully remove it.

Be careful when handling the motherboard.

Never attempt to remove the paper source, a box attached to the side or bottom of the unit to which all the cables are connected.

When removing any cables, wires or ribbons make sure to grasp the wire at its base or head to keep it from breaking.

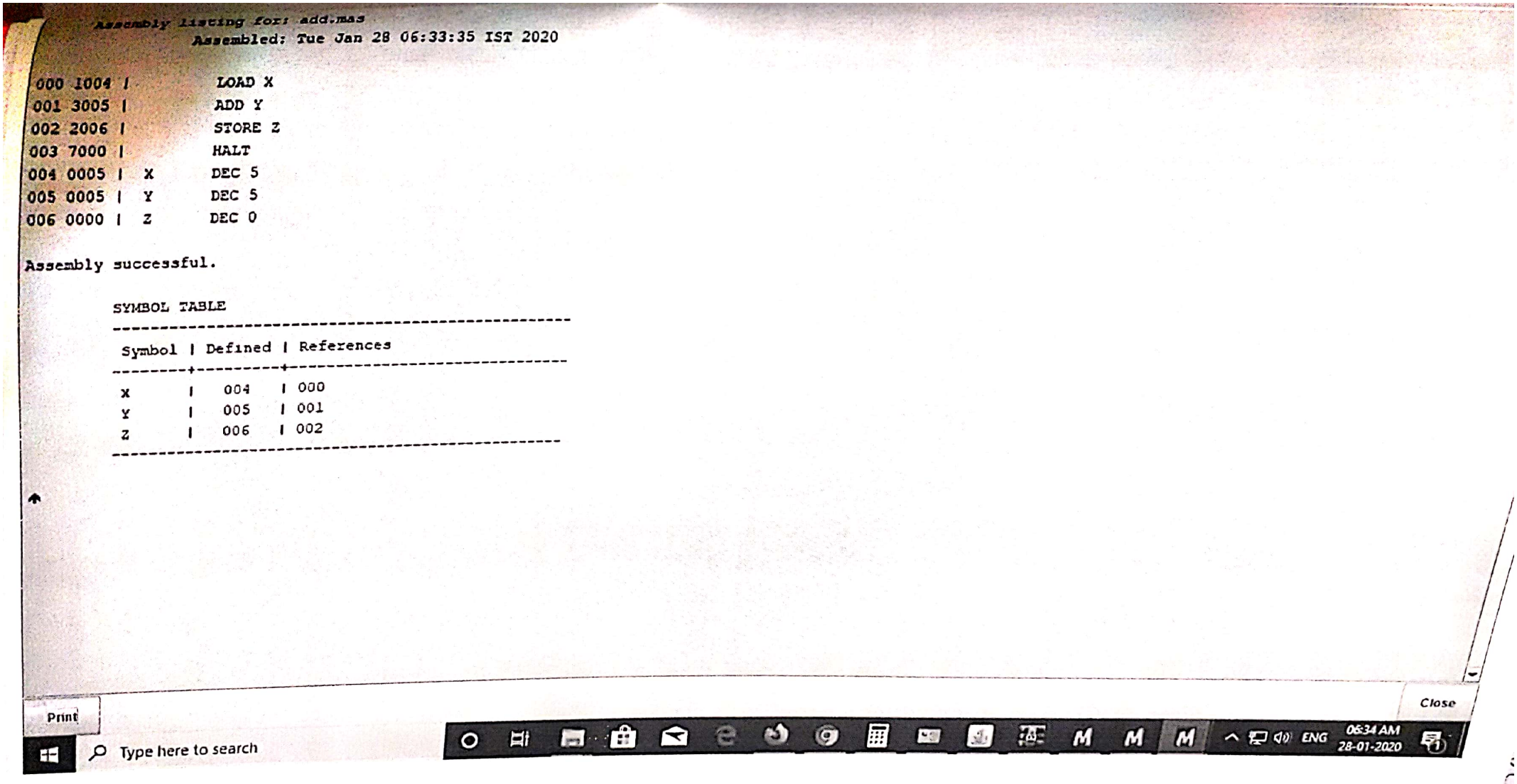
Be careful not to drop any small parts into unreachable areas such as into the computer fan or disk drive.

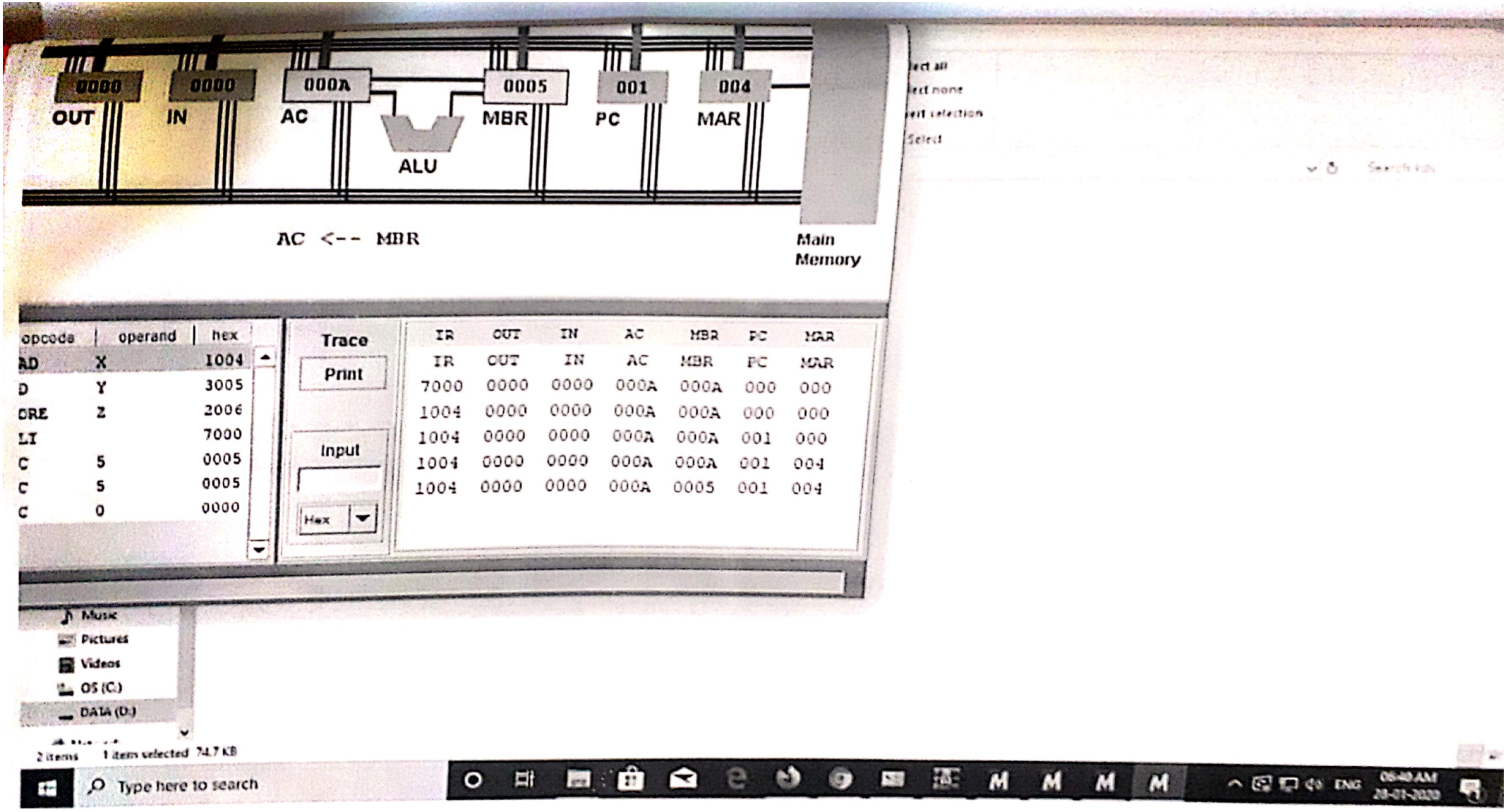
Take note that the three most damaging things to a computer are moisture, shock & dust.

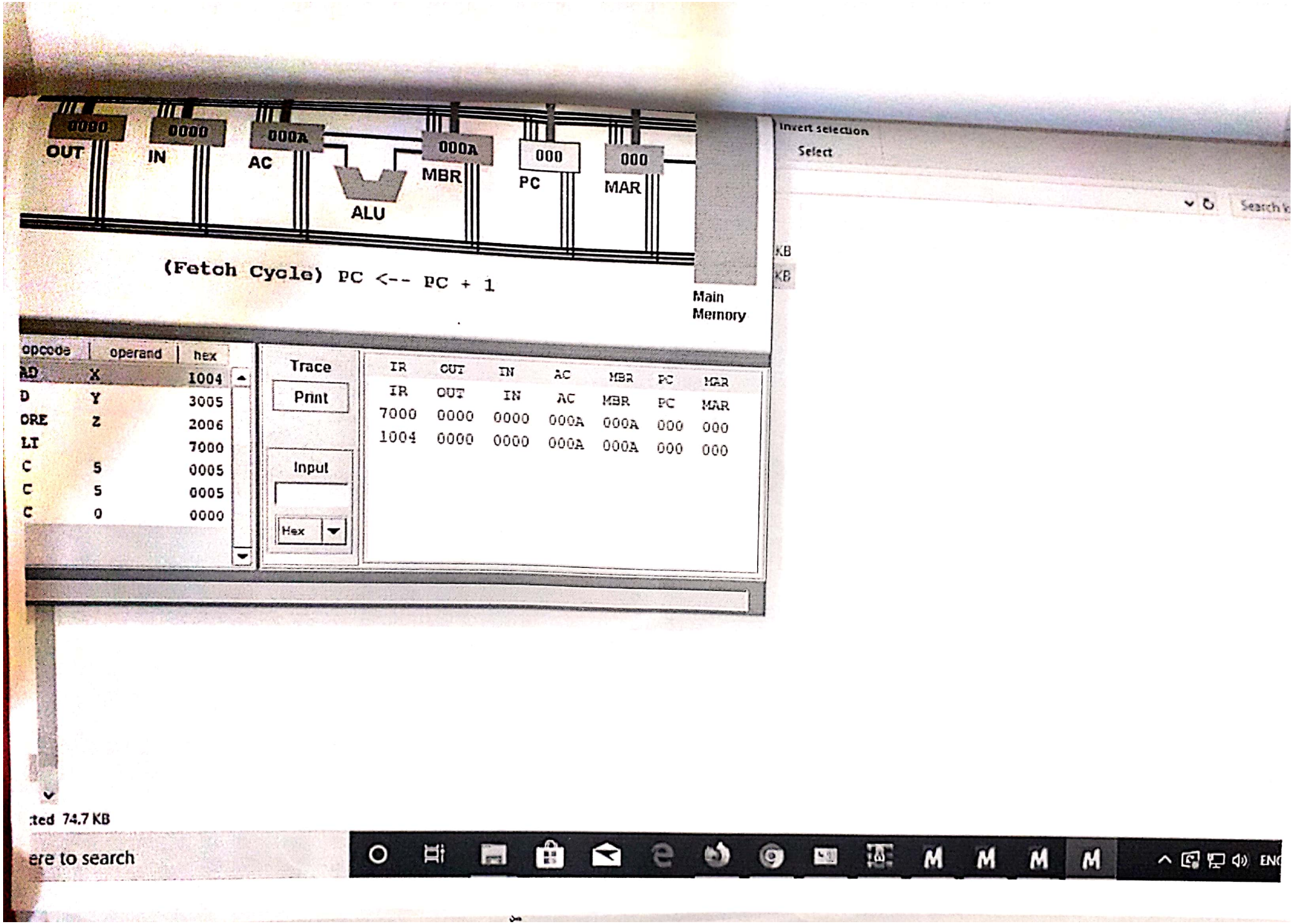
File Edit Assemble Help

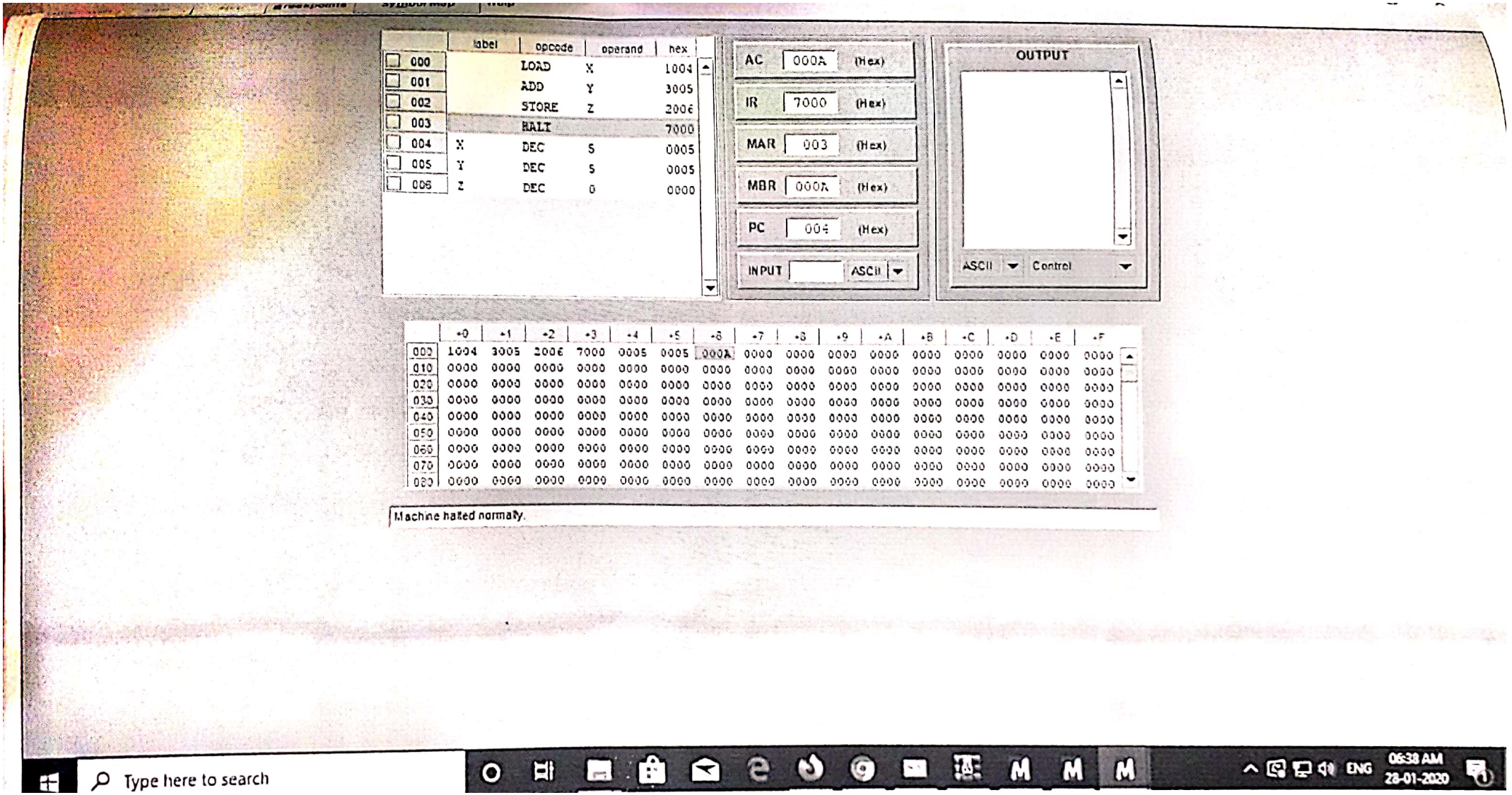
```
LOAD X
ADD Y
STORE Z
HALT
X, DEC 5
Y, DEC 5
Z, DEC 0
```

D:\CO Lab\MARIE Datapath Simulators\add.mas Assembly successful









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: Isha Gupta	**Marks:** 10 /10	**Date:** 11 Feb 2020
USN: 1MS18CS049	**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.

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	Isha Gupta	Branch:	CSE
USN/Roll No. :	1MS18CS049	Sem/Sec:	IV / B
Subject :	CO Lab	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,#0x00000030
MOV  R3,#0
MOV  R4,#50
STR  R4,[R1,R3]
LDR  R6,[R1,R3]
SWI  0x11
```

(3)

```
MOV  R1,#20
MOV  R2,#30
ADD  R3,R1,R2
```

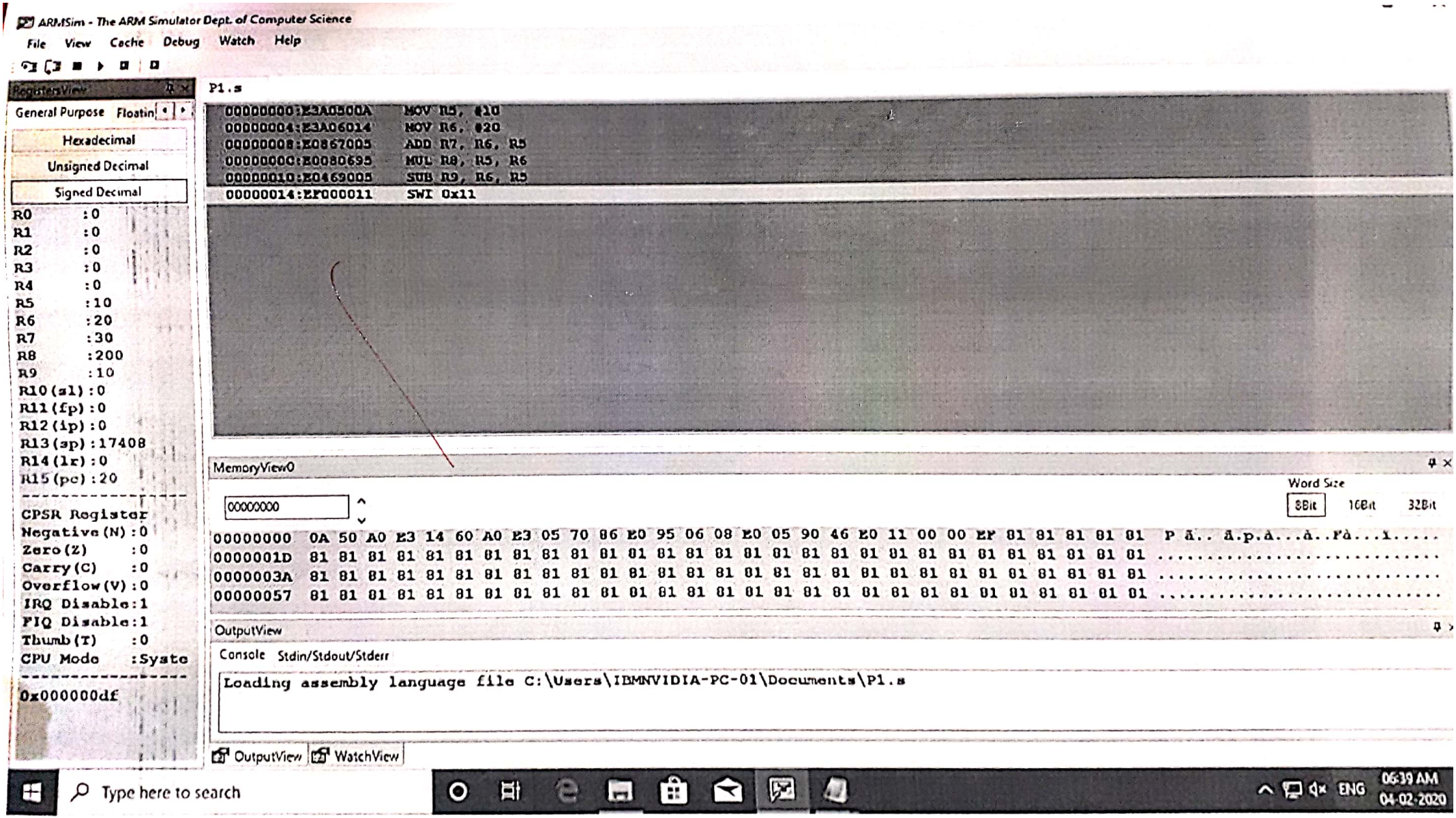
```
MOV   R4, #40
MOV   R5, #50
ADD   R6, R4, R5
SUB   R7, R3. R6
SWI   0x11
```

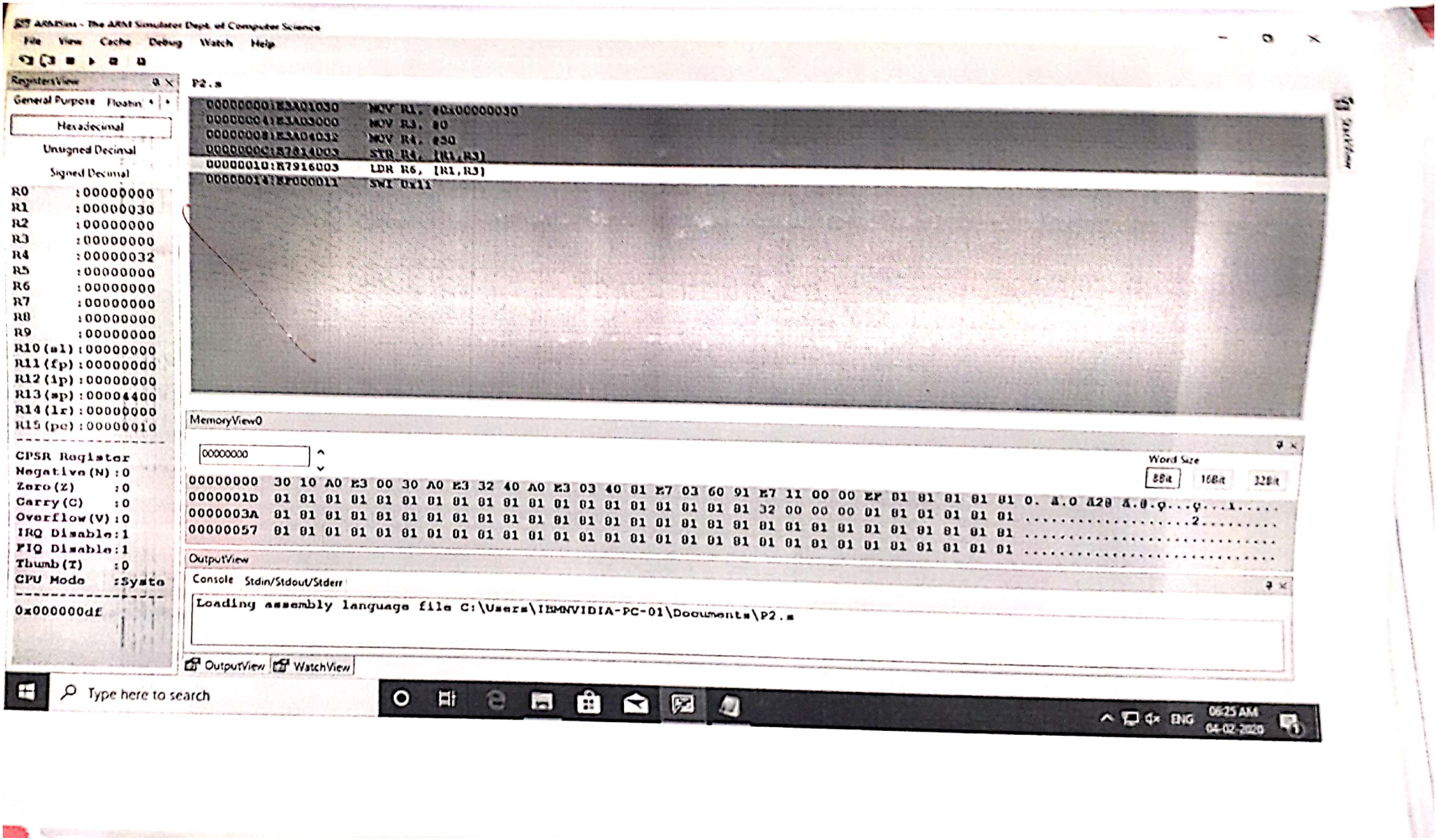
(5)

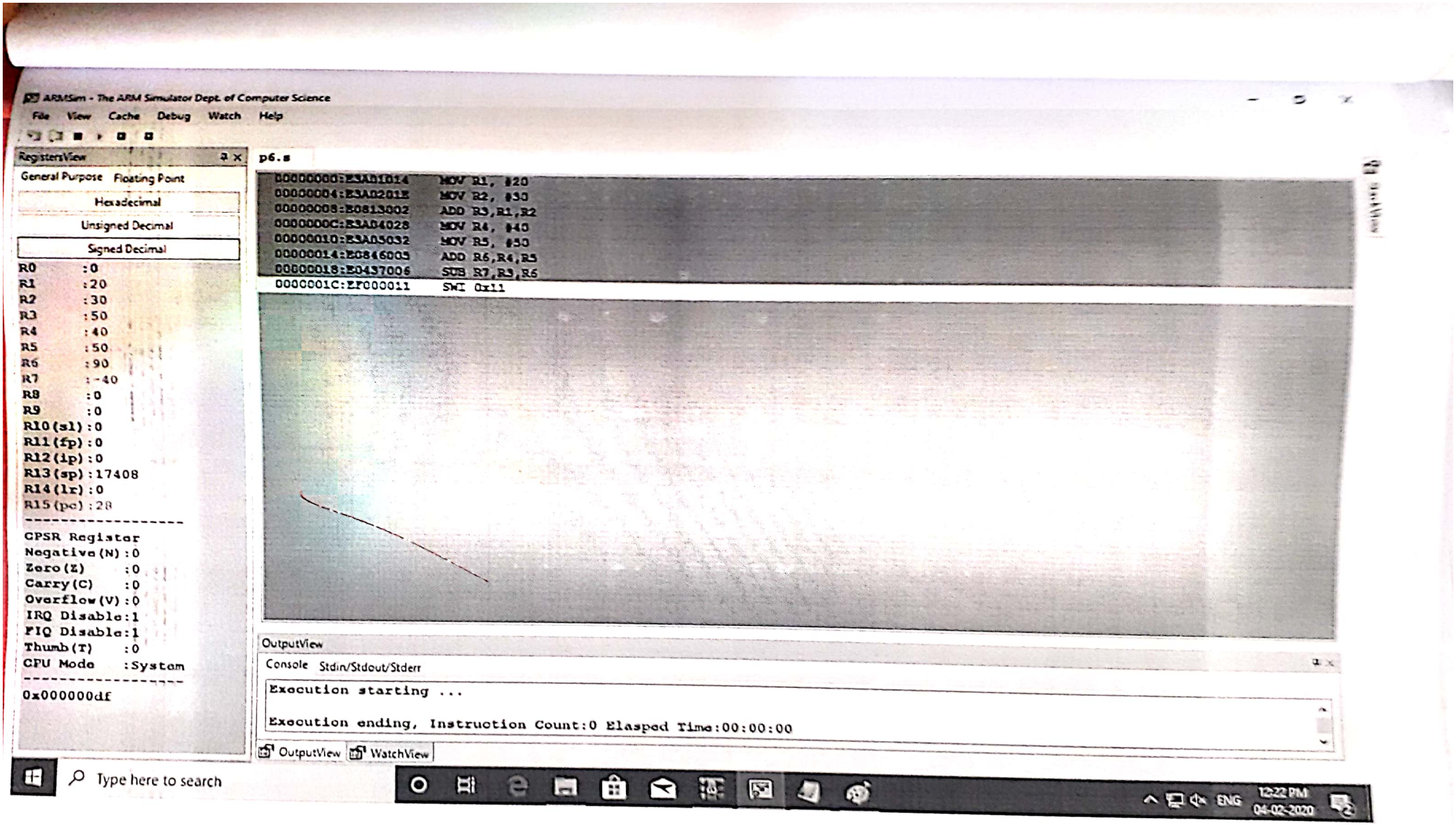
```
MOV   R0, #3
MOV   R1, R0
MOV   R2, #1
MOV   R3, #1
FACT
      MUL   R2, R1, R2
      SUB   R1, R1, R3
      CMP   R1, #1
      BGE   FACT
      SWI   0x11
```

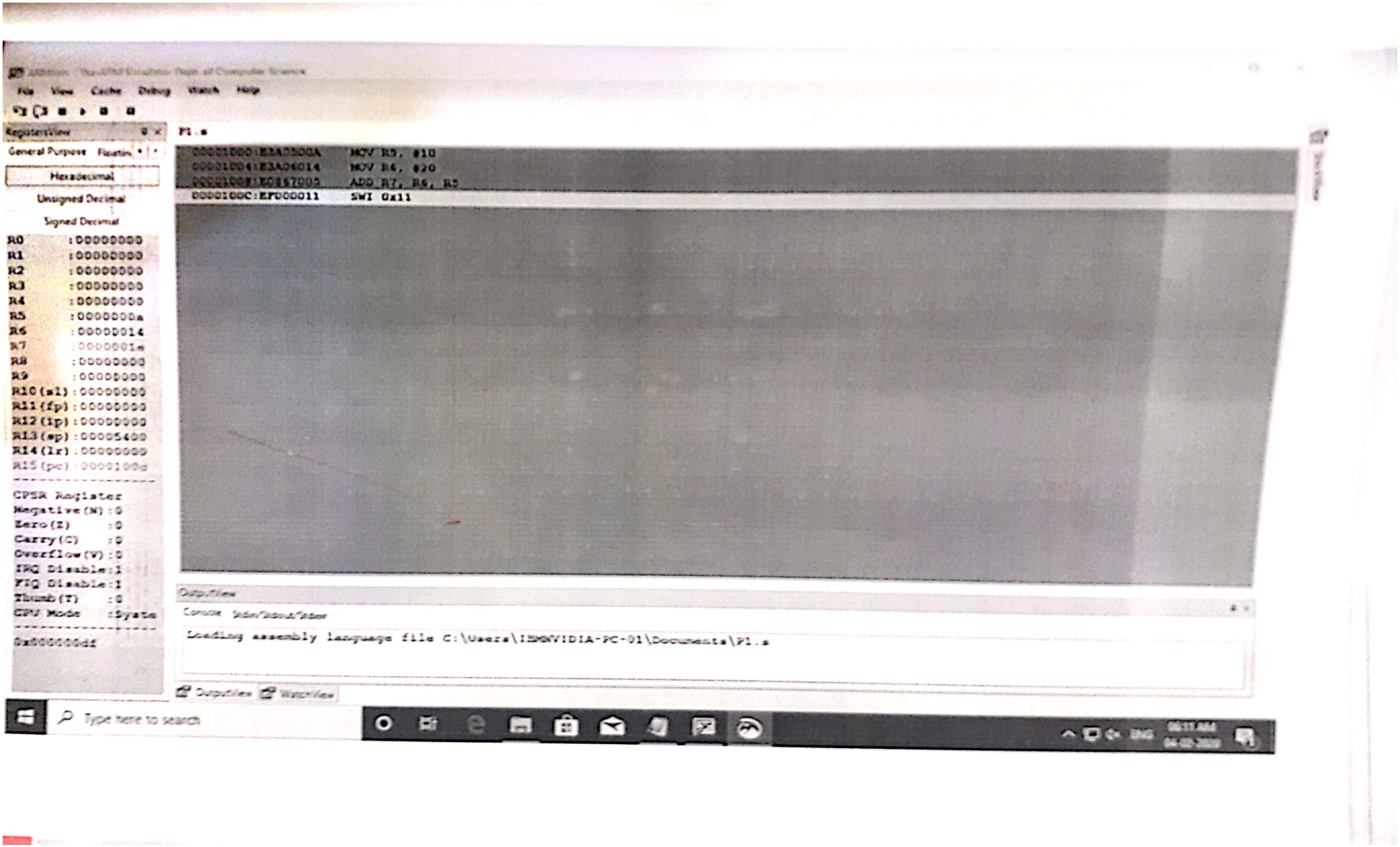
(6)

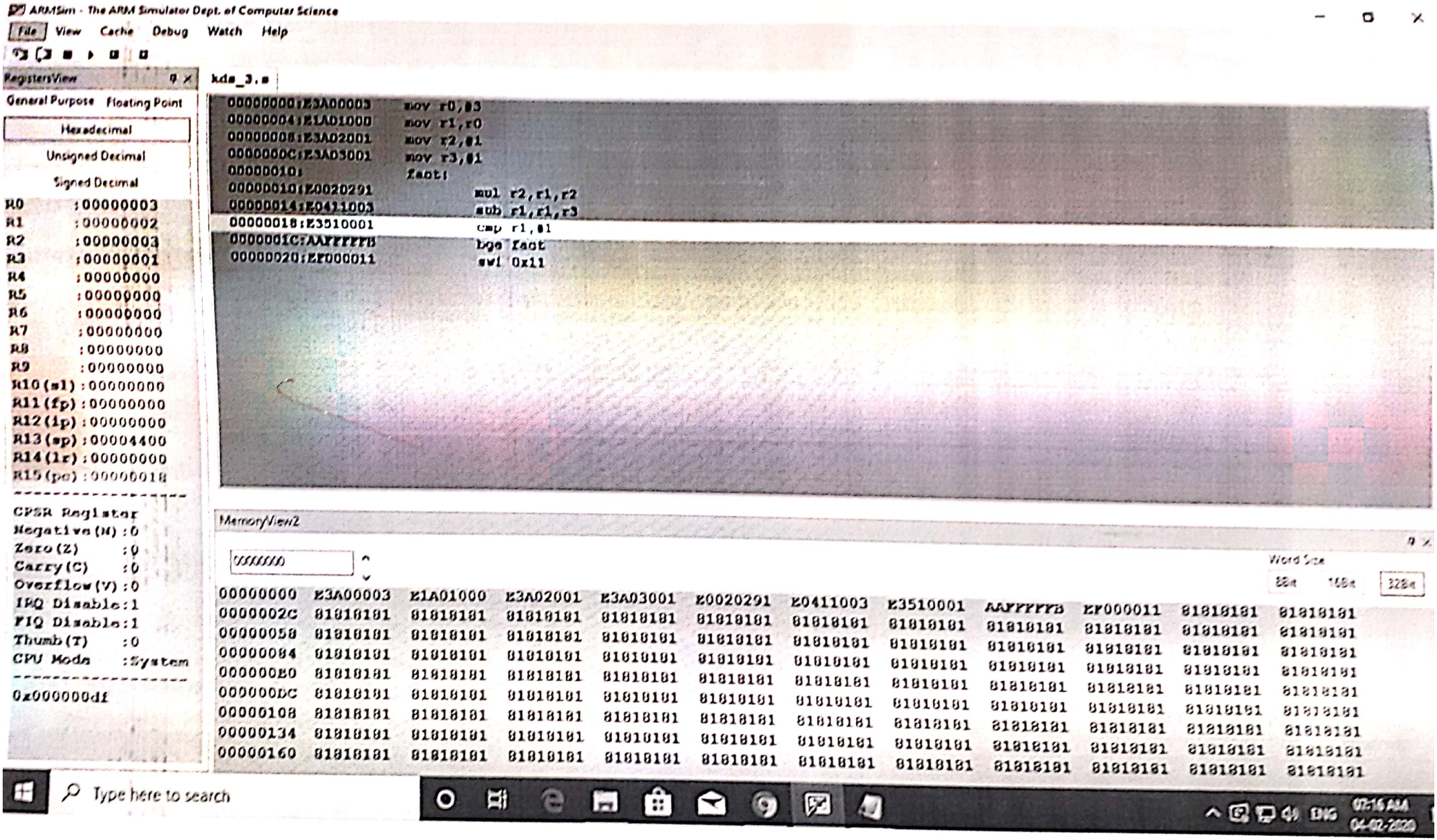
```
MOV   R0, #5
LDR   R1, =array
loop  LDR   R2, [R1], #4
      ADD   R3, R3, R2
      SUB   R0, R0, #1
      CMP   R0, #0
      BNE   loop
array DCD   0x0000 0001, 0x0000 0002,
            0x0000 0003, 0x0000 0004,
            0x0000 0005
      SWI 0x11
```











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: Isha Gupta	**Marks:** /10	**Date:** 18/02/2020	
USN: 1MS18CS049	**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 :	Ishra Gupta	Branch:	CSE
USN/Roll No. :	1MS18CS049	Sem/Sec:	IV / B
Subject :	CO Laboratory	Subject Code:	

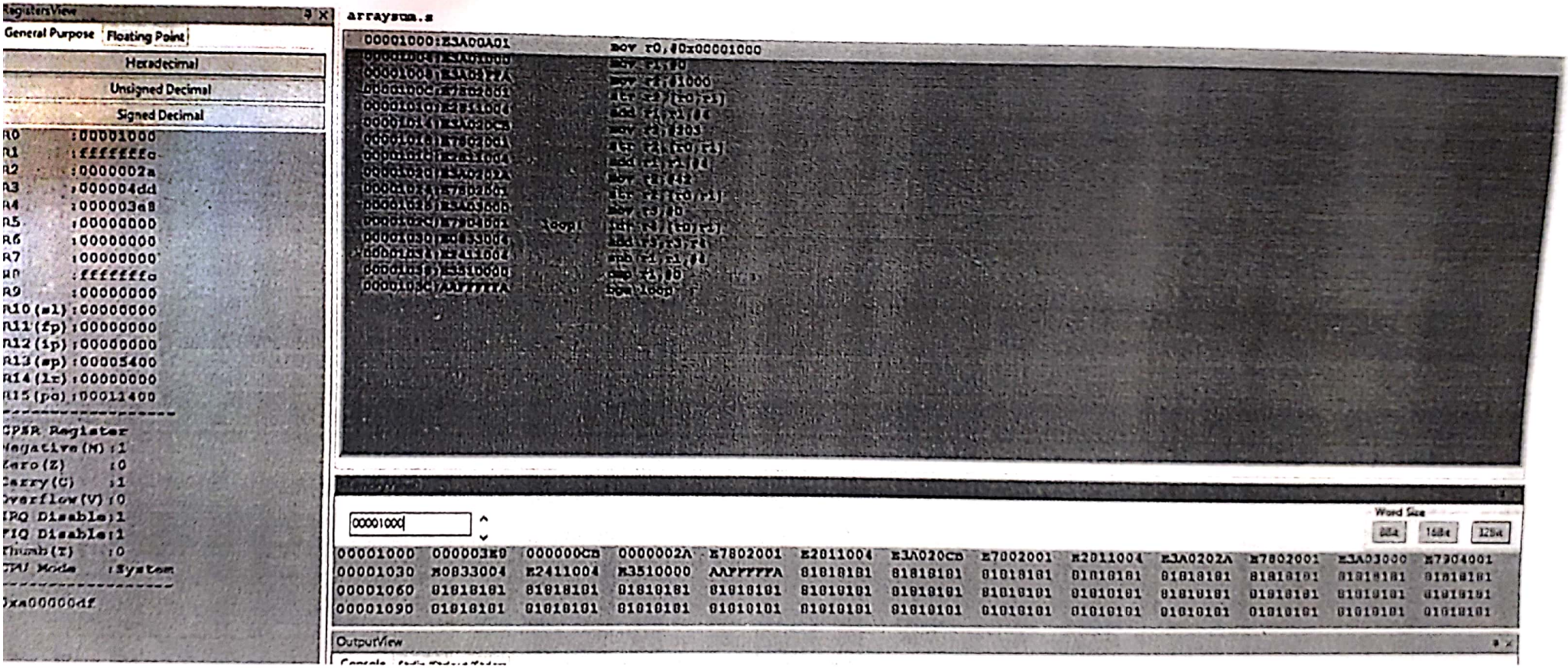
```
MOV   R0, #0
MOV   R1, #1
MOV   R2, #5
MOV   R3, #0
LDR   R4, =0x00002000
MOV   R5, #0
LOOP : STR  R0, [R4, R5]
       ADD  R6, R0, R1
       MOV  R0, R1
       MOV  R1, R6
       ADD  R5, R5, #4
       ADD  R3, R3, #1
       CMP  R3, R2
       BLT  LOOP
SWI   0x11
```

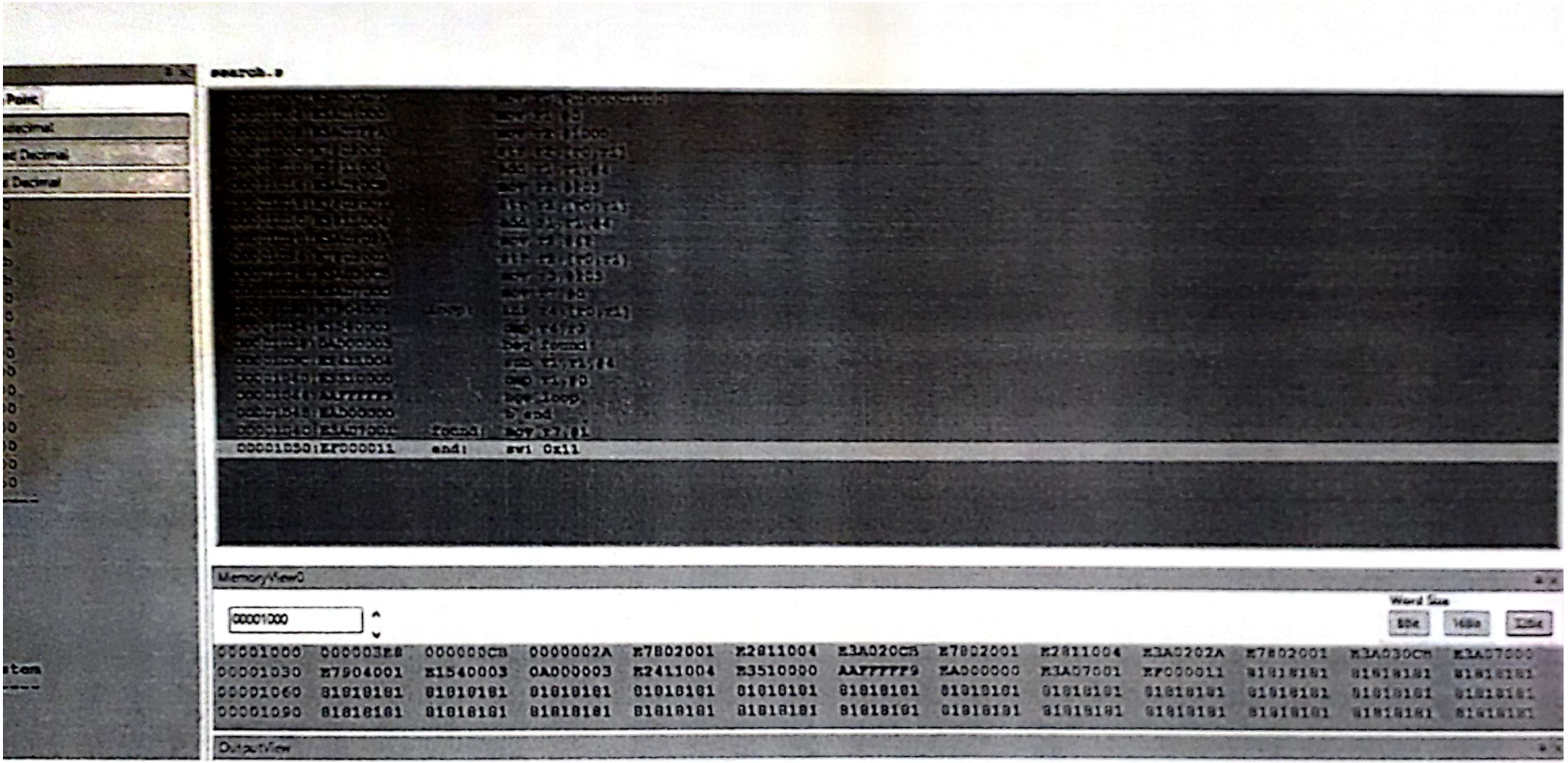
2)

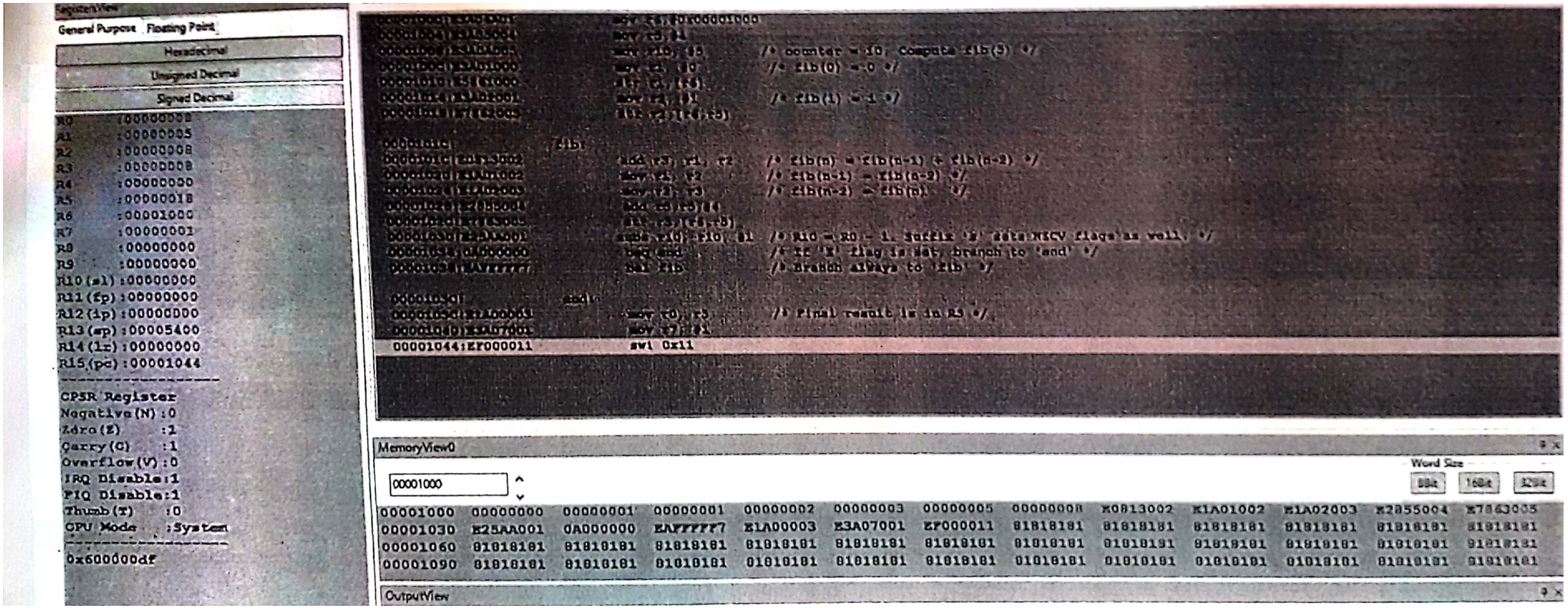
```
MOV   R0, #4
MOV   R4, #'n'
MOV   R1, #25
LDR   R0, =0x00002000
MOV   R5, #4
LOOP: LDR   R2, [R0, R5]
      SUB   R0, R0, #1
      ADD   R5, R5, #4
      CMP   R1, R2
      BEQ   printy
      CMP   R0, #0
      BEQ   printn
      BNE   LOOP
printn: STR   R4, [R0]
        LDR   R0, [R0]
        SWI   0x00
        B   END
printy: STR   R3, [R0]
        LDR   R0, [R0]
        SWI   0x00
end: SWI  0x11
```

```
.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

_start:
ldr r0,=FileName
mov r1, #0
swi SWI_Open
bcs Exit
mov r9, r0
mov r5, #0

loopstart:
mov r5, #Stdout
mov r0, r9
ldr r8, =Array
bcs afterloop
ldr r0, [r8, r5]
add r5, r5, #4
mov r1, r0
mov r0, #Stdout
swi SWI_PrInt
add r4, r4, #1
ldr r1, =Newline
```

```
    swi  SWI_PrStr
    bal  loopstart

afterloop:
    mov  r5, #20
loop:  ldr  r2, [r8, r5]
       Sub  r4, r4, #1
       Sub  r5, r5, #4
       mov  r1, r2
       mov  r0, #Stdout
       swi  SWI_PrInt
       Ldr  r1, =NewLine
       swi  SWI_PrStr
       Cmp  r4, #0
       beq  end
       bne  loop
end:   mov r0, r9
       swi  SWI_Close

Exit:
swi  SWI  Exit
```







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: Isha Gupta	Marks: /10	Date:
USN:1MS18CS049	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

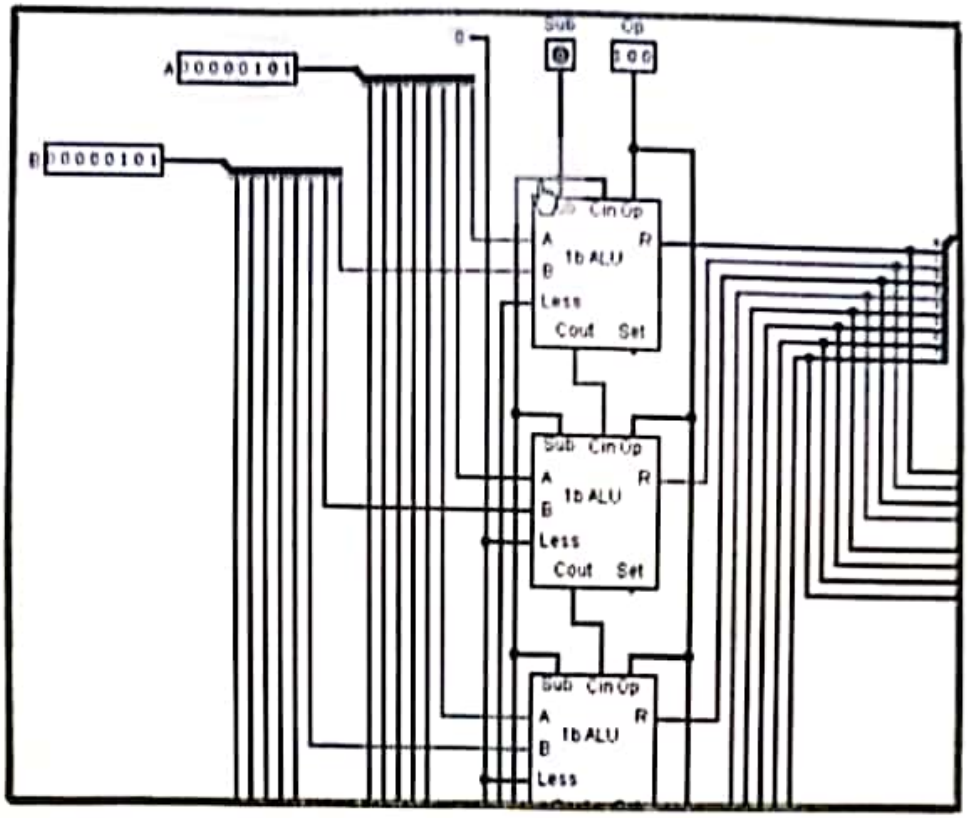
CO Lab - 5

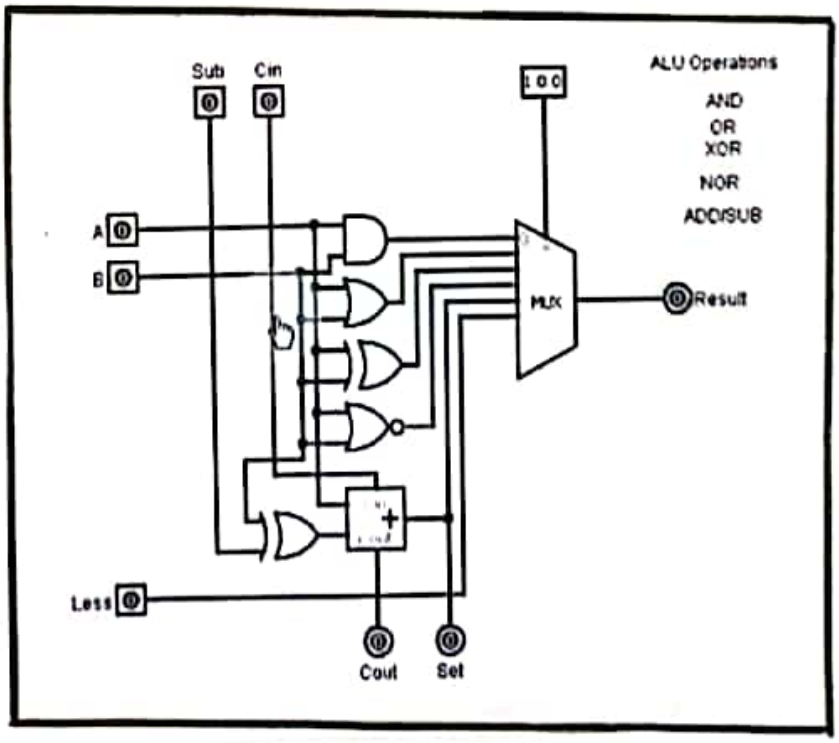
Name: Isha Gupta
USN: 1MS18CS049
Sem: 4 'B'.

1) List out the steps in designing ALU.

(i) Add the 2 i/p pins, name them A and B.
(ii) Add or, and, ex-or, nor gates and a 1-bit adder
(iii) Connect the A's and B's of all the gates to their respective pins.
(iv) Add an output pin and name it Result
(v) Add a 1-bit multiplier with 3-select bits.
(vi) Connect outputs of all the gates to the mux.
(vii) Connect 3-bit input pin to mux.
(viii) Add i/p pin to Cin, output pin to Cout.
(ix) Add an ex-or gate. Connect its o/p to Cout. The first i/p must be connected to B and second to another i/p pin sub.
(x) Add another i/p and name it less. Connect it to the mux.
(xi) Add an output pin and name it Set, connect it to the o/p of adder unit.

Snapshots.





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:Isha Gupta	Marks: /10	Date:
USN:1MS18CS049	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

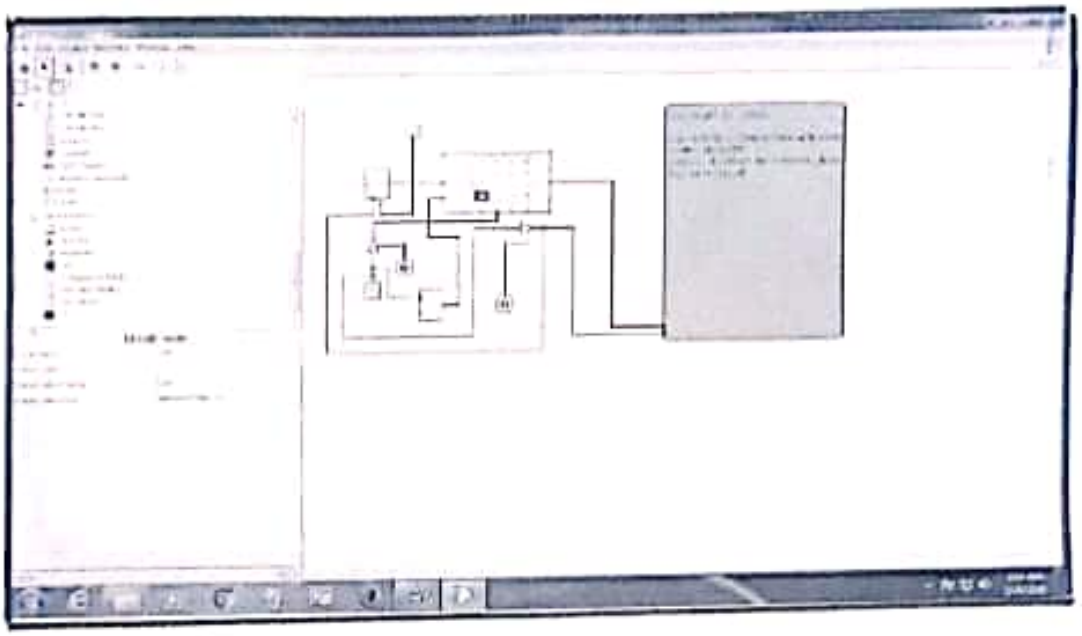
CO Lab-6

Name: Isha Gupta
USN: 1MS18CS049
Sem: 4 'B'

1) List out the steps in designing memory system

(i) Add a RAM with separate load and store selected
(ii) Add a counter and connect Q to A of the ~~Ram~~ RAM.
(iii) Add a controller buffer and connect its O/P to the RAM
(iv) Add a clock and connect to the i/p of the buffer.
(v) Add a TTY unit with 32 rows and columns. Make the connections with RAM.
(vi) Add a 7-bit random number generator, connect Q to D.
(vii) Add another controlled buffer, connect to TTY. Also add an i/p pin to the buffer
(viii) Connect the output of the second buffer to the counter.
(ix) Connect a button to the counter.

Snapshots :



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 VII: To simulate advantages of using pipeline technique in executing a program.

Name: Isha Gupta	Marks: /10	Date:
USN:1MS18CS049	Signature of the Faculty:	

Objective: To learn and analyze the performance of the CPU by overlapping of instructions using CPUOS-SIM simulator.

Simulator Used: CPUOS-SIM is a software development environment for the simulation of simple computers. It was developed by Dale Skrien to help users to understand computer architectures.

Modern CPU's contain several semi-independent circuits involved in decoding and executing each machine instruction. Separate circuit elements perform each of these typical steps:

- Fetch the next instruction from memory into an internal CPU register.
- Decode the instruction to determine which function sub-circuits it requires.
- Read any input operands required from high-speed registers or directly from memory.
- Execute the operation using the selected sub-circuits.
- Write any output results to high-speed registers or directly to memory.

Separate sections of the CPU circuitry are used for each of these steps. This allows these circuit sections to be arranged into a sequential pipeline, with the output of one step feeding into the next step.

Activity to be performed by students:

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

```
lw   $10,20($1)
sub  $11, 42, $3
add  $12, $3, $4
lw   $13, 24($1)
add  $14, $5, $6
```

CO-Lab:1

Name: Yeha Gupta
USN: 1MS18 CS049
Sec: 4 'B'

Program execution order (in instruction)

lw $10, 20($1)

subt $11, $2, $3

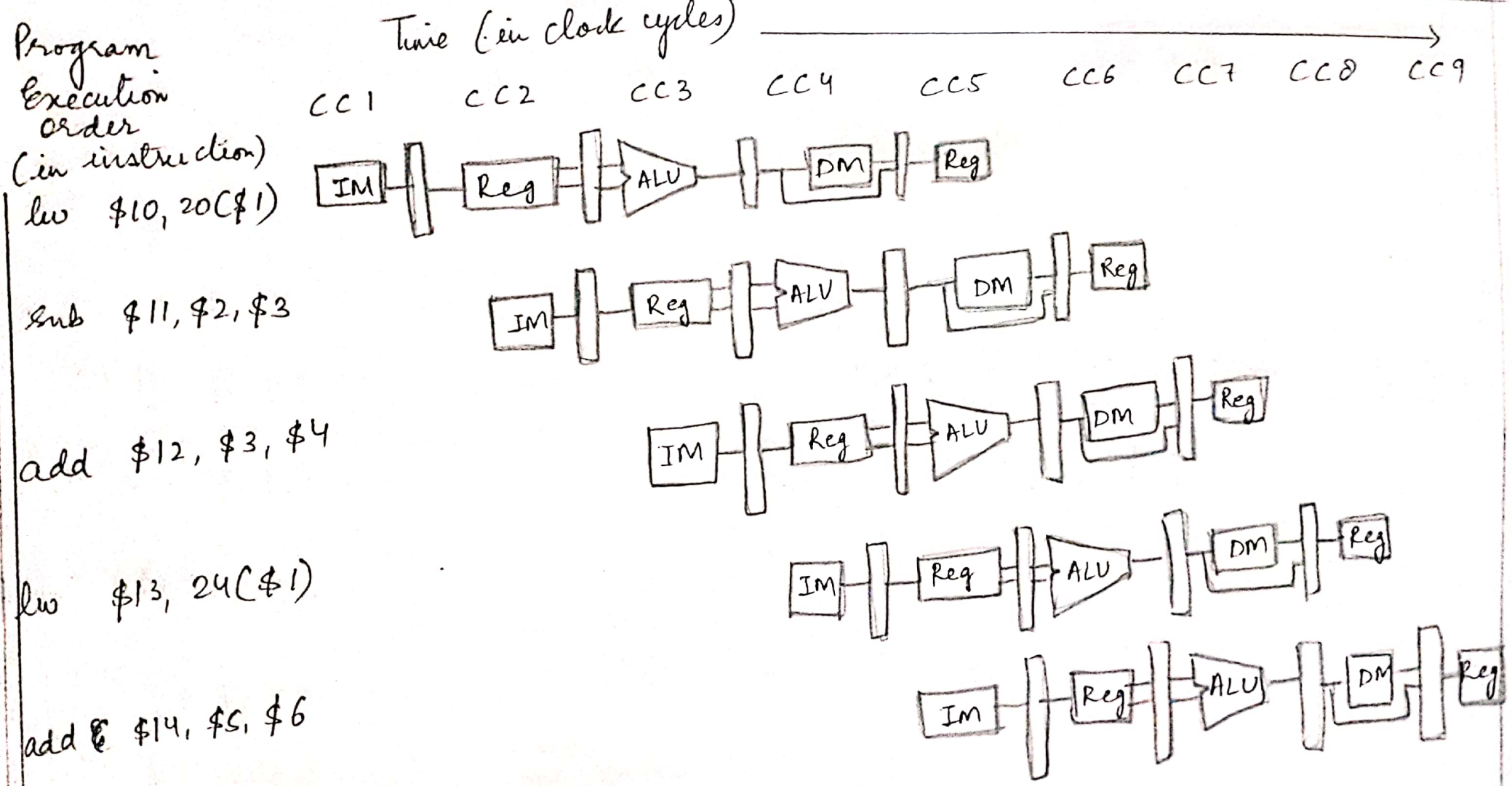
add $12, $3, $4

lw $13, 24($1)

add $14, $5, $6

Time (in clock cycles) →

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9
lw	Instruction fetch	Instruction decode	Execution	Data access	Write back				
subt		Instruction fetch	Instruction decode	Data access Execution	Write Data access	Write back			
add			Instruction fetch	Instruction decode	Execution	Data access	Write back		
lw				Instruction fetch	Instruction decode	Execution	Data access	Write back	
add					Instruction fetch	Instruction decode	Execution	Data access	Write back



Snapshots :