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: Manish V	**Marks:** /10	**Date:**
USN: 1MS18CS067	**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: 10

Name :	Manish.V	Branch:	CSE
USN/Roll No. :	1MS18CS067	Sem/Sec:	IV 'B'
Subject :	Computer Organization.	Subject Code:	

Lab-I (Tutorial-I).

1) Write down the detailed procedure to assemble the system.

Ans: To assemble the system,

1) Getting the computer case ready: We have to analyze our needs & requirements & we have to get our computer case ready. If you plan on installing an internal adapter card, say for example a PCI express wireless adapter card, remove the metal bracket from the case rear corresponding to the PCI Express x1 slot on your motherboard.

2) Install power supply unit (PSU): Make sure Voltage switch on PSU is set to right voltage for your country. Fasten 4 screws that came with the case through rear of case into PSU to hold PSU in place. Tighten all screws with finger tip force only to avoid over tighting and potential damage.

3) Install DVD drive: Remove from the front panel of the case the top external 5.25 inch drive bay cover for the DVD drive.

4) Install Motherboard: Lay case on its side with open

side facing up, back of case (rear outputs side) closest to you. Drape all case wires including PSU wires outside case, making sure not to scratch case. Install shield by snapping along edges till small points take hold.

5) Install processor (CPU): Follow any installation instructions that came with CPU. Handle CPU with care touch only by the sides! Don't touch the socket on the motherboard! Note location of a triangle marked on one corner of the CPU plastic cap that covers the CPU socket on the motherboard.

6) Install CPU heat sink & CPU fan: Follow any installation instructions that came with CPU heat sink and CPU fan. Make sure the thermal paste on the bottom of the CPU heat sink is not wiped off. Connect the 4 CPU heat sink connectors to the motherboard per instructions from the CPU manual.

7) Install memory: Install one memory stick in memory slot 1. Install other memory stick in memory slot 2.

8) Install hard disk drive (HDD) or M.2 SSD: Check which empty slot in 3.5 inch internal drive bay in case would be best for hard drive.

9) Clear CMOS to reset BIOS & UEFI settings.

10) Tuck and tie PSU wires.

11) Connect power to motherboard: Plug in 20+4 pin main power connector from the PSU.

12) Connect power to CPU: Plug in the 4 pin CPU connector from the PSU to the CPU power

header on motherboard.

13) Connect case front panel wires to motherboard.
14) Connect case front panel USB 3 wire to motherboard.
15) Connect case front panel HD wire audio wire to motherboard.
16) Connect case front panel HD audio wire to motherboard.
17) Connect case fan to motherboard.
18) Connect powrh to HDD/SSD & DVD.

Q) Explain how trouble shooting a system helps to trace & correct the failure in the system.

Ans: Troubleshooting is a form of problem solving, often applied to repair failed products or processes on a machine or a system. It is logical, systematic search for the source of a problem in order to solve it, and make the product or process operational again. A basic principle in troubleshooting is to start from the simplest & most probable possible problems first. The principle resuets in the comman complaint about help desks or manuals, that they sometimes first ask: "Is it plugged in and dos that receptable have power?. It also helps to start from a known good state, the

best example being a computer reboot. A cognitive walk through is also a good thing to try. It provides theory of operation for the subject device or system. Troubleshooting can also take form of a systematic checklist, troubleshooting procedure, flowchart or table. Troubleshooting tables can be computerized to make them more efficient for users.

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

Ans:- 1) Check how much RAM your computer currently has installed: It will be helpful to know how much RAM you have already installed in your computer.

2) Check how much RAM your computer and operating system can support:

There are several factors that will dictate how much RAM your system can support, including your operating system and motherboard limits.

3) Check what RAM format your motherboard supports: The standard these days are DDR4 RAM, there are DDR3, DDR2 & even DDR

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	Manish V		
USN/Roll No. :	1MS18CS067	Branch:	CSE
Subject :	Computer Organization	Sem/Sec:	IV 'B'
		Subject Code:	

4) Determine the clock speed: The clock speed of the RAM is measured in megahertz (MHz). Motherboards typically support a range of clock speeds.

5) Purchase RAM modules in pairs: Nearly all RAM should be installed in pairs. The total value of each module should be within limits of your motherboard.

6) Understand the difference between desktop & laptop memory: Most desktop computers use DIMM RAM while most laptops use SO-DIMM which is smaller.

7) Install your RAM: Push each module directly into the slot, ensuring the notches at the bottom line up.

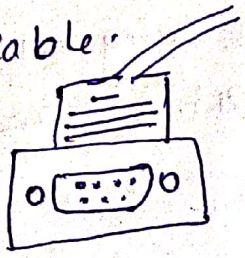
8) Boot up your operating system

9) Verify that the RAM is recognized.

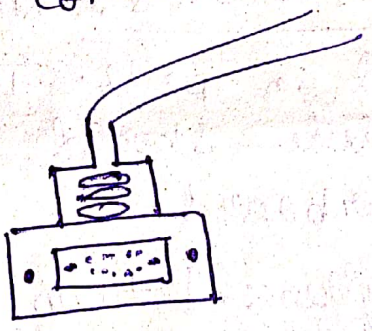
4) With a neat diagram explain different cable used to connect functional units in a system.

Ans: A computer port is also called as Communication Port as it is responsible for communication blw the computer & its peripheral device.

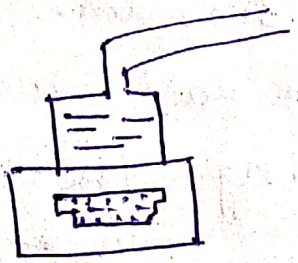
1) VGA cable: Also known as D-sub cable, analog video cable.



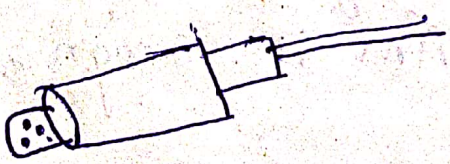
2) DVI Cable: Connect one end to computer monitor.



3) HDMI cable: Connect one end to computer monitor, television.
Another end to HDMI port on computer.



4) PS/2 Cable

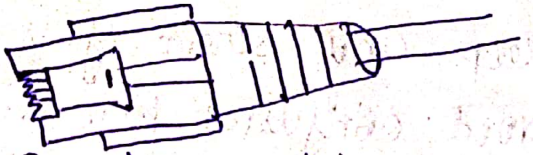


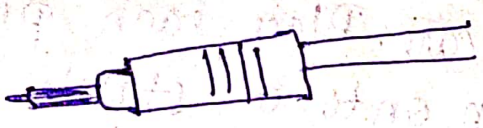
Connect one end to PS/2 keyboard, PS/2 mouse: other end to ports on computer.

5) Ethernet cable: Also known as RJ-45 cable.

Connect one end to: router, network switch.
Another end to ethernet port on computer.

6) 3.5mm audio cable:





Connect one end to computer speakers.

Other end to audio ports on Computer.

Discuss the safety precautions that needed to be taken care while removing the components of the system.

A few warnings and reminders before you start disassembling your computer to keep both the unit and yourself safe are:

1) Fully shut down and unplug the computer before you make any attempts to disassemble.
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 be still some remaining electric charge.

3) Make sure your hands are completely dry to a avoid damaging any mechanical parts as well as to avoid electrocution.

4) Work in a cool area to avoid perspiration for the same reason as seen in previous number.

5) Before touching any part, 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 it is ideal to avoid confusion b(w similar looking screws.

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

8) If a component dos not come out easily, do not forcefully remove it.

9) Be carefully when holding the motherboard, it's underside actually quite pointy and able to hurt you.

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

11) Take a note that the three of the 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: Manish V	Marks: 10/10	Date:
USN: 1MS18CS067	Signature of the Faculty:	4/21/20

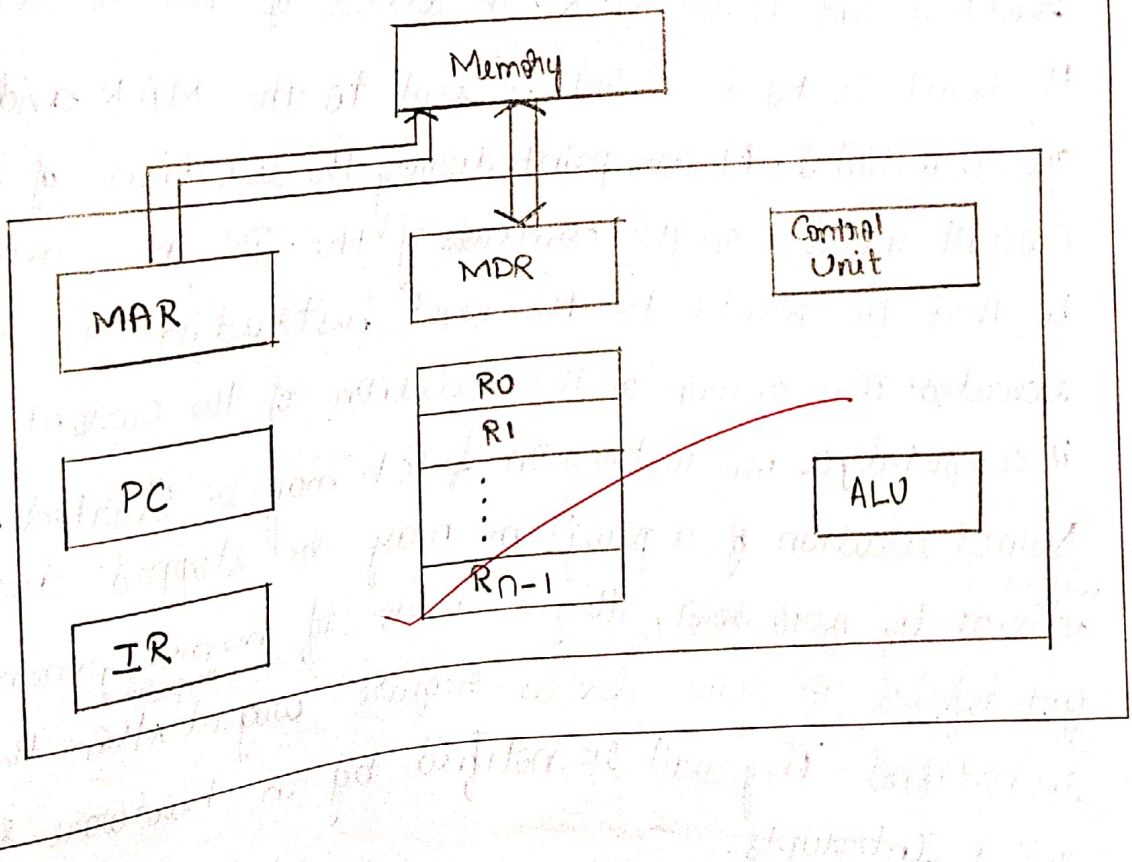
Activity II: Demonstrating Datapath and instruction execution stages using MarieSim Simulator

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.

Ans: Initially PC will have the address of the starting instruction. The same address is copied down to MAR. After that CU generates the read signal & data is copied down to MDR. Next, the content of MDR is transferred to the IR. At this point, the instruction is ready to be decoded & executed. If the instruction involves an operation to be performed by the ALU, it is necessary to obtain the required operands. If an operand resides in the memory, it has to be fetched by sending its address to the MAR and initiating a Read Cycle. When the operand has been read from the memory to MDR, it is transferred from the MDR to ALU. After one or more operands are fetched in this way, the ALU can perform the desired operation. If the result of this operation is to be stored in the memory, then result is sent to the MDR. The address of the location where the result is to be stored is sent to the MAR and a write cycle is initiated. At some point during the execution of the current instruction the contents of the PC are incremented so that PC points to the next instruction to be executed. Thus, as soon as the execution of the current instruction is completed, a new instruction fetch may be started.

Normal execution of a program may be stopped due to (e.g. such as) divide by zero error, illegal access of memory, memory unit (repair) got defected or some devices require urgent attention then this will be notified. They will be notified by an hardware signal called Interrupts.

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

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

```
LOAD g
ADD h
STORE X
LOAD i
ADD y
STORE Y
loop, LOAD f
Add X
Store f
Load Y
SUBT one
Store Y
Skipcond 400
Jumploop
LOAD f
Output
Halt.

X, DEC 0
Y, DEC 0
one, DEC 1
Num, DEC 0
g, DEC 10
h, DEC 5
i, DEC 10
y, DEC 20
```

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

```
LOAD b
loop, LOAD X
ADD b
STORE X
Store b
SUBT one
Store b
Skipcond 400
Jumploop
LOAD X
Output
STORE X
loop, LOAD Y
ADD b
STORE Y
STORE a
SUBT one
STORE b STORE 4
Skipcond 400
Jumploop
LOADY
STOREY
loop, LOAD Z
ADD Y
STORE Y
STORE C
SUBT one
STORE Y
Skipcond 400
Jumploop.
SUBT X, Z
LOAD d
Output
Halt.

X, DEC 0
Y, DEC 0
Z, DEC 0
d, DEC 0
b, DEC 10
a, DEC 40
b, DEC 3
```

3. Results and Snapshots:

4) Describe the factors affecting the performance of a processor.

Ans: a) i) Design of Hardware.
ii) Instruction set.
iii) Compiler.

b) i) Elapsed time : The turn around time blw execution of a program.

ii) Processor time : The time spent by processor in executing the program.

iii) Processor clock : Processor ckts are controlled by a timing clock (measured in terms of clock cycles).

iv) Basic steps → Steps which are required to execute a single instruction.

v) Clock Rate ; $R = \frac{1}{P}$ cycles/sec Hz.

P → length of clock.

Design of Hardware: Performance eqn : $T = \frac{N \times S}{R}$ T → processor time required to execute a program.

N → total no. of ins in a program. S → avg no. of Basic steps needed to execute one machine instruction

R → clock rate.

To achieve high performance, the designer must reduce the value of T.

∴ we can use pipeline to increase value of S.

Instruction set : There are 2 types of computers

1) RISC (Reduced Instruction set computer)
2) CISC (Complex Instruction set computer).

When we compare these two by implementing pipeline technique then CISC gives best performance.

Compiler : Unnecessary instructions must be resolved.

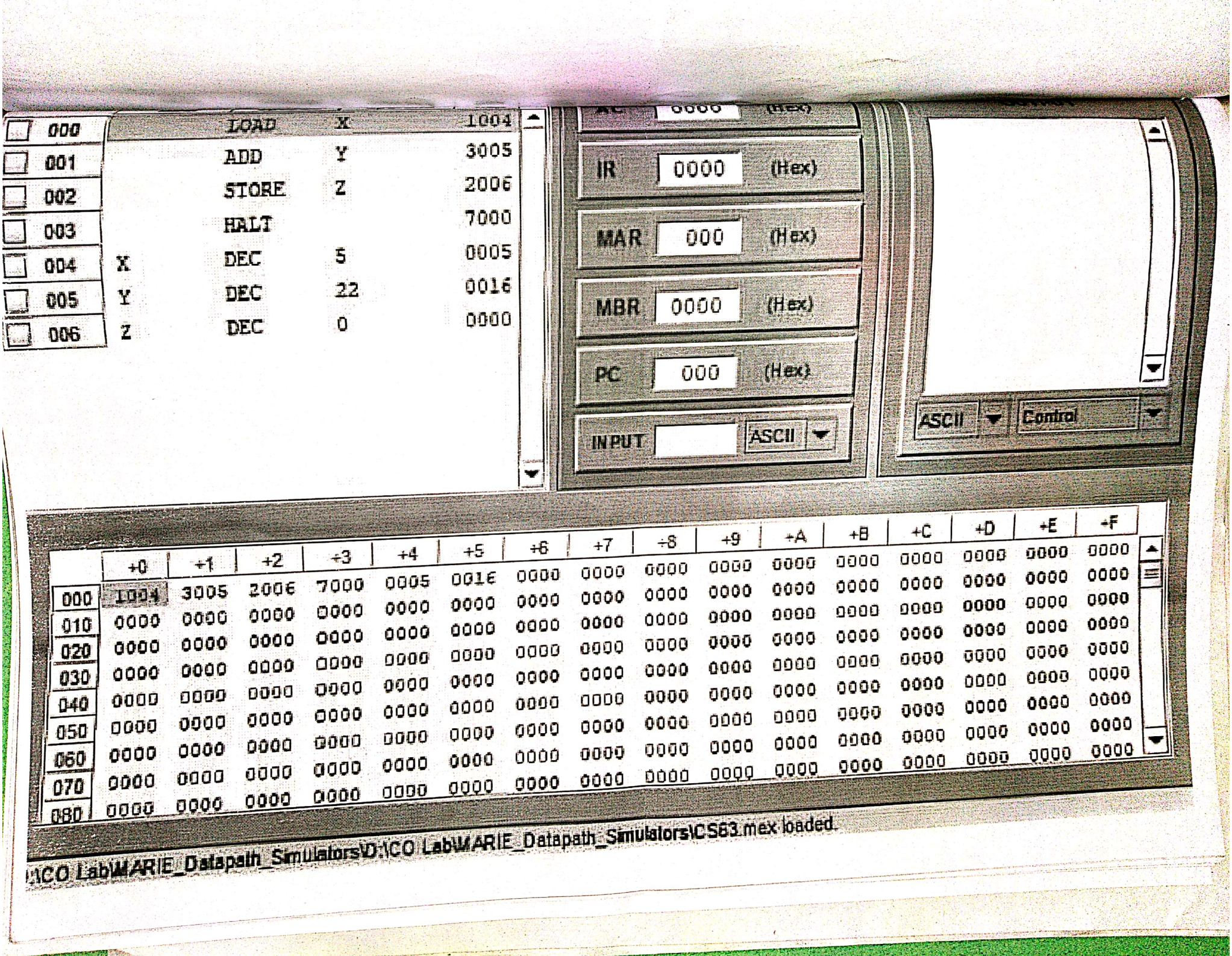
Compiler may do reording of instructions to achieve better performance.

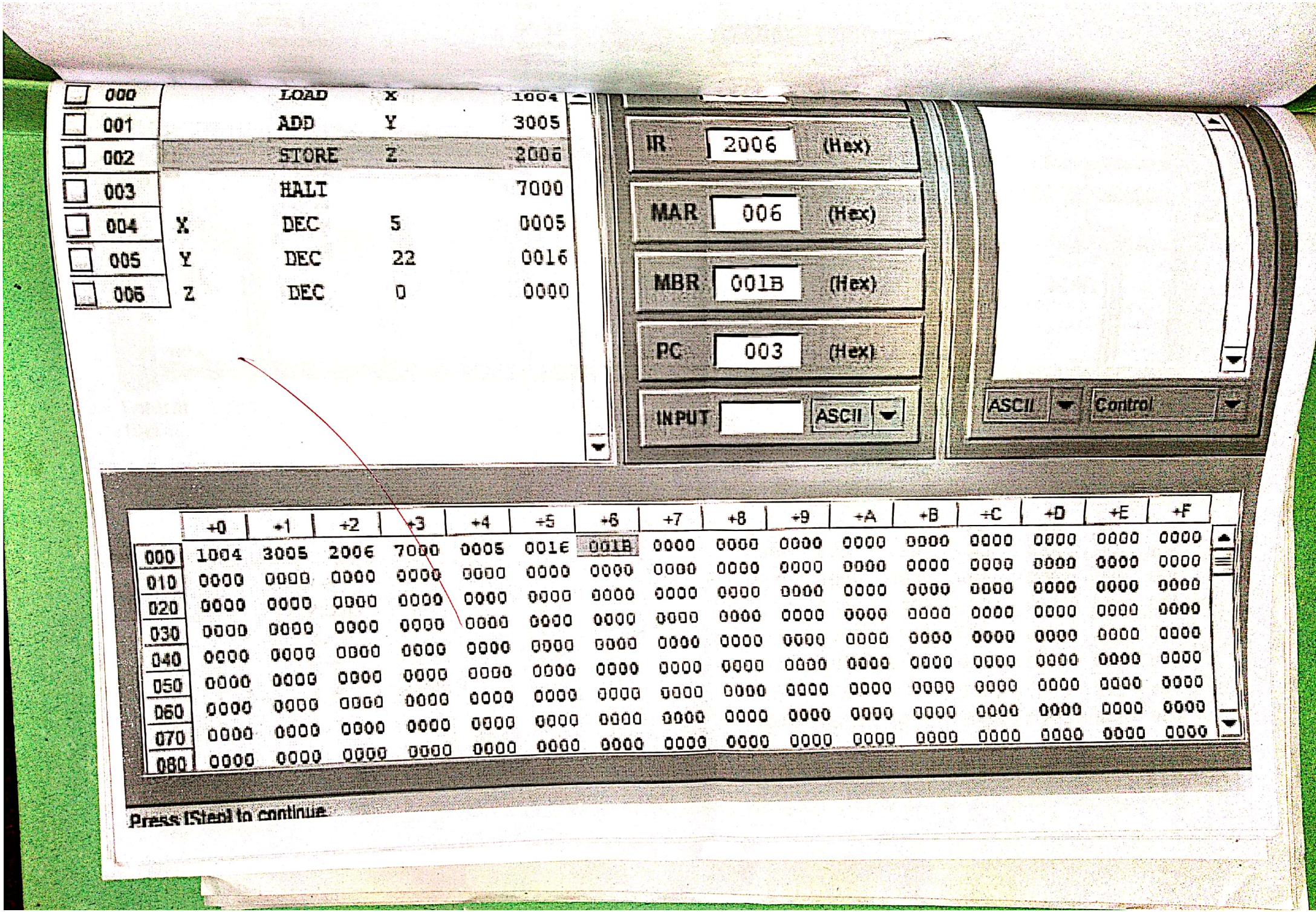
It should reduce N*S.

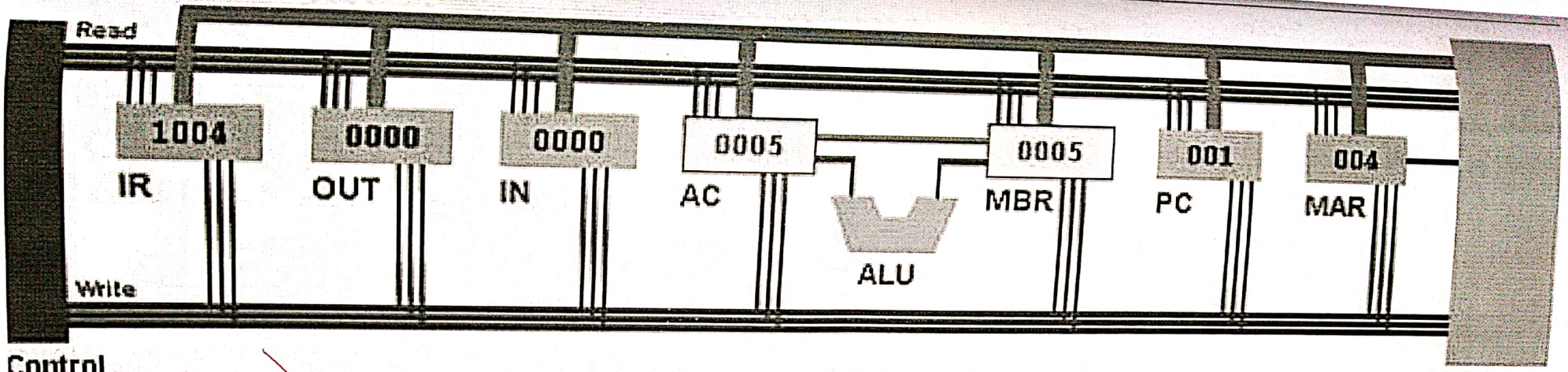
```
Assembly listing for: CS63.mas
          Assembled: Tue Jan 28 06:26:37 IST 2020

000 1004 |          LOAD X
001 3005 |          ADD Y
002 2006 |          STORE Z
003 7000 |          HALT
004 0005 |  X       DEC 5
005 0016 |  Y       DEC 22
006 0000 |  Z       DEC 0

Assembly successful.
```

SYMBOL TABLE

Symbol	Defined	References
X	004	000
Y	005	001
Z	006	002







	label	opcode	operand	hex
000		LOAD	X	1004
001		ADD	Y	3005
002		STORE	Z	2006
003		HALT		7000
004	X	DEC	5	0005
005	Y	DEC	22	0016
006	Z	DEC	0	0000

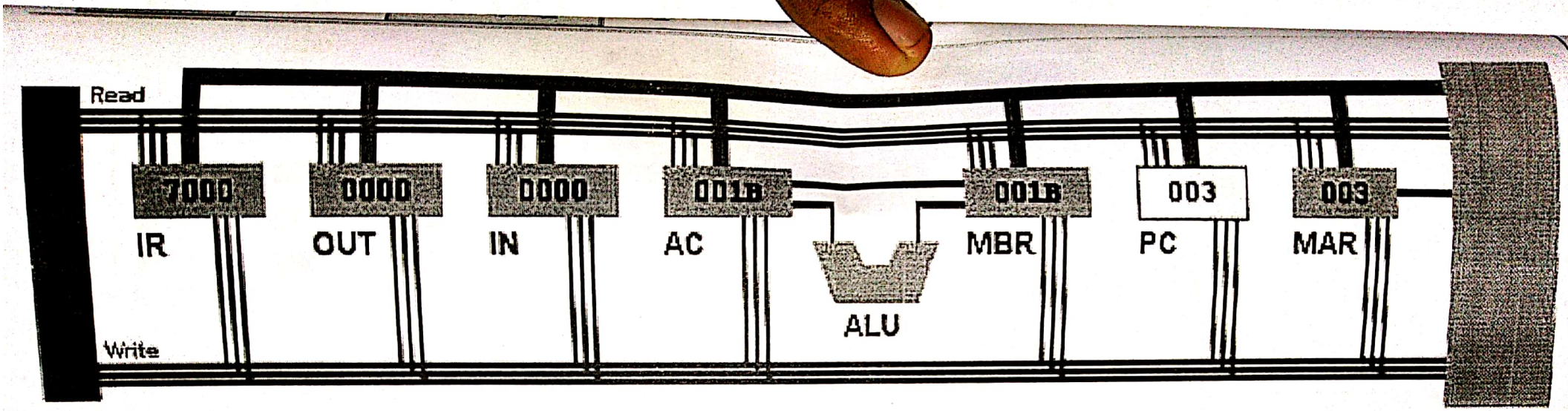
Trace

Print

Input

Hex

IR	OUT	IN	AC	MBR	PC	MAR
IR	OUT	IN	AC	MBR	PC	MAR
0000	0000	0000	0000	0000	000	000
1004	0000	0000	0000	0000	000	000
1004	0000	0000	0000	0000	001	000
1004	0000	0000	0000	0000	001	004
1004	0000	0000	0000	0005	001	004
1004	0000	0000	0005	0005	001	004



Control Unit

(Fetch Cycle) PC <-- PC + 1

Main Memory

	label	opcode	operand	hex
000		LOAD	X	1004
001		ADD	Y	3005
002		STORE	Z	2006
003		HALT		7000
004	X	DEC	5	0005
005	Y	DEC	22	0016
006	Z	DEC	0	0000

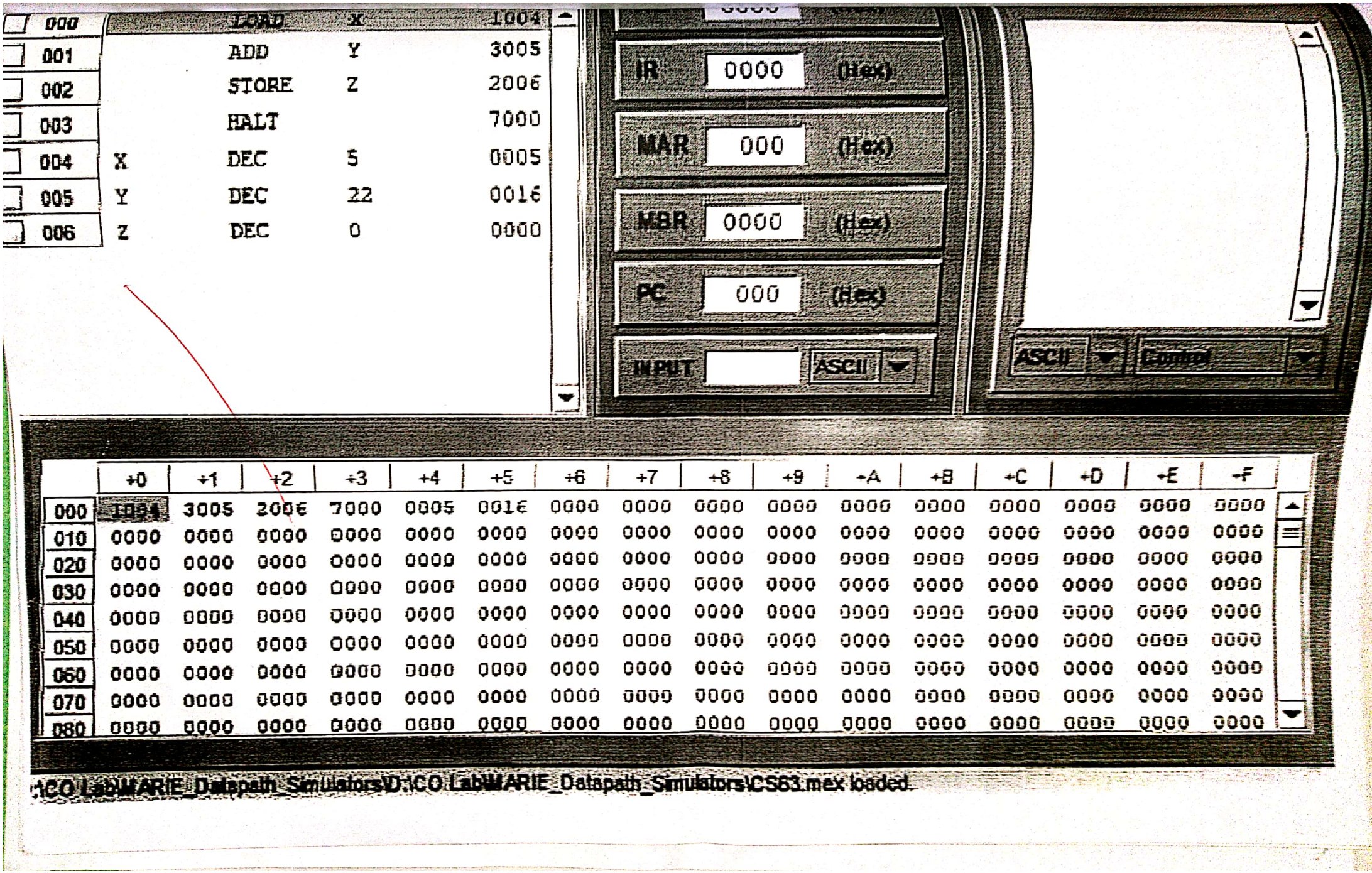
Trace

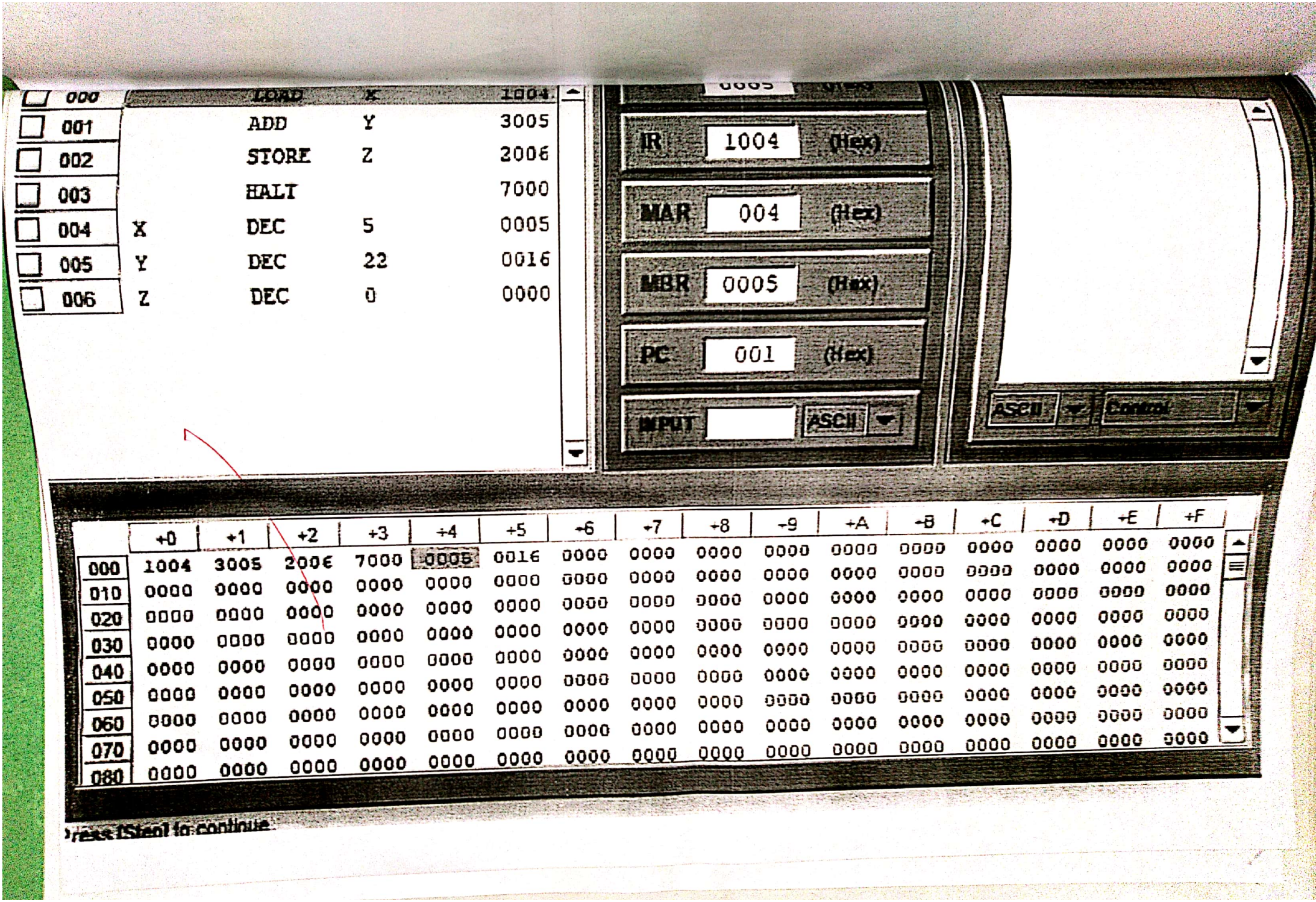
Print

Input

Hex

IR	OUT	IN	AC	MBR	PC	MAR
3005	0000	0000	001B	0016	002	002
2006	0000	0000	001B	0016	002	002
2006	0000	0000	001B	0016	003	002
2006	0000	0000	001B	0016	003	006
2006	0000	0000	001B	001B	003	006
2006	0000	0000	001B	001B	003	003
7000	0000	0000	001B	001B	003	003





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: Manish V		
USN: 1MS18CS067	Marks: 10/10	Date:
	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 :	Manish V	Branch:	CSE
USN/Roll No. :	1MS18CS067	Sem/Sec:	IV 'B'
Subject :	Computer Organization.	Subject Code:	CS45

1) Write an ARM program to perform basic arithmetic operations.

Ans:
```
MOV R0, #10
MOV R1, #20
ADD R2, R0, R1
SWI 0x11
```

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

Ans:
```
MOV R1, #0x00000070
MOV R3, #0
MOV R4, #50
STR R4, [R1, R3]
LDR R6, [R1, R3]
SWI 0x11
```

3) Write an ARM program to evaluate expression

$f = (g+h) - (i+j)$

Ans:-
```
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) Write an ARM program to find sum of all elements in the array.

Ans:-
```
      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 BCD 0x00000001, 0x00000002, 0x00000003, 0x00000004,
          0x00000005

      SWI 0x11
```

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

Ans:
```
MOV R1, #5
MOV R0, #1
MOV R3, #1
```

```
Loop: MUL R3, R0, R3
      ADD R0, R0, #1
      SUB R1, R1, #1
      CMP R1, #1
      BGE Loop

SWI 0x11
```

ADD - Notepad

File Edit Format View Help

```
MOV R5,#10
MOV R6,#20
ADD R7,R5,R6

SWI 0x11
```

Type here to search

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

ENG 11:36 AM 04-02-2020

ADD.ASM

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

ADD.s

```
00001000:E3A0500A    MOV R5,#10
00001004:E3A06014    MOV R6,#20
00001008:E0857006    ADD R7,R5,R6

0000100C:EF000011    SWI 0x11
```

RegisterView

General Purpose

Hexadecimal

Unsigned Decimal

Signed Decimal

```
R0      :0
R1      :0
R2      :0
R3      :0
R4      :0
R5      :10
R6      :20
R7      :30
R8      :0
R9      :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4108

CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :Sy

0x000000df
```

OutputView

Console Stdin/Stdout/Stderr

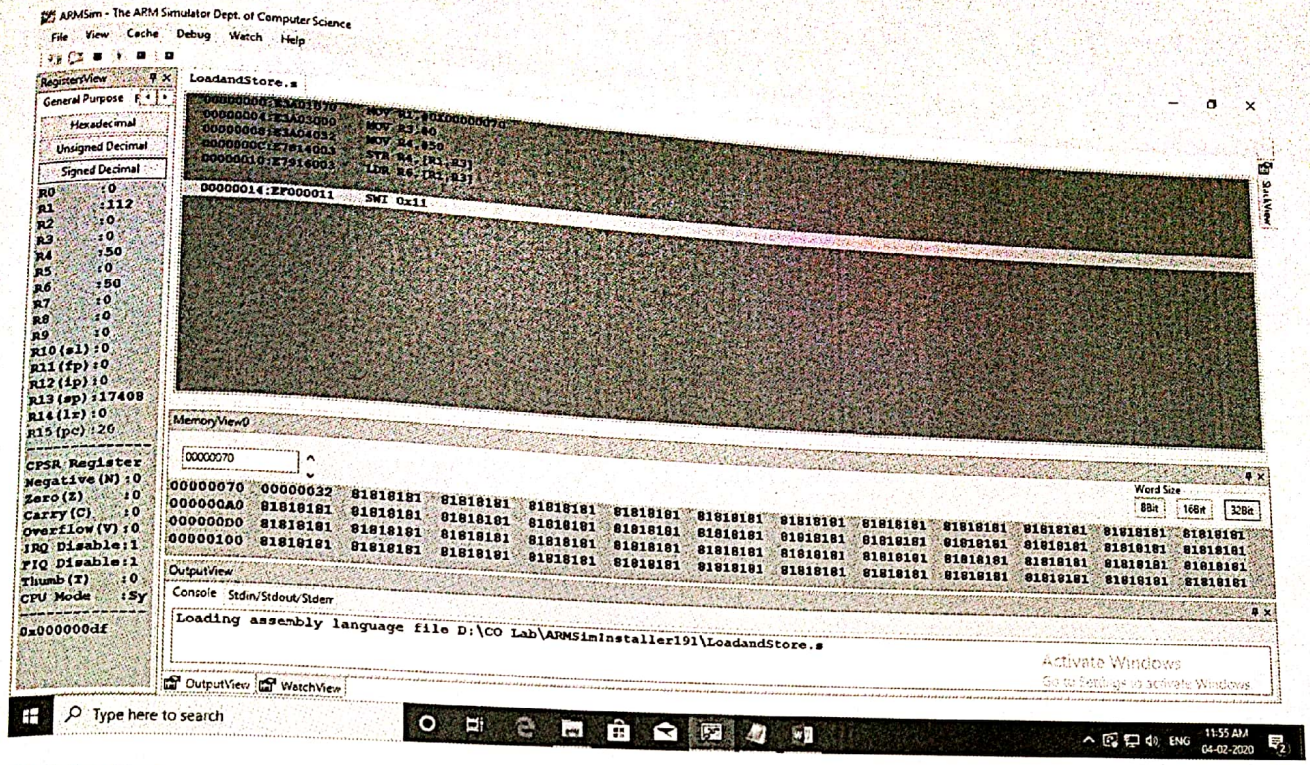
```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0468738
Instructions per second:0
```

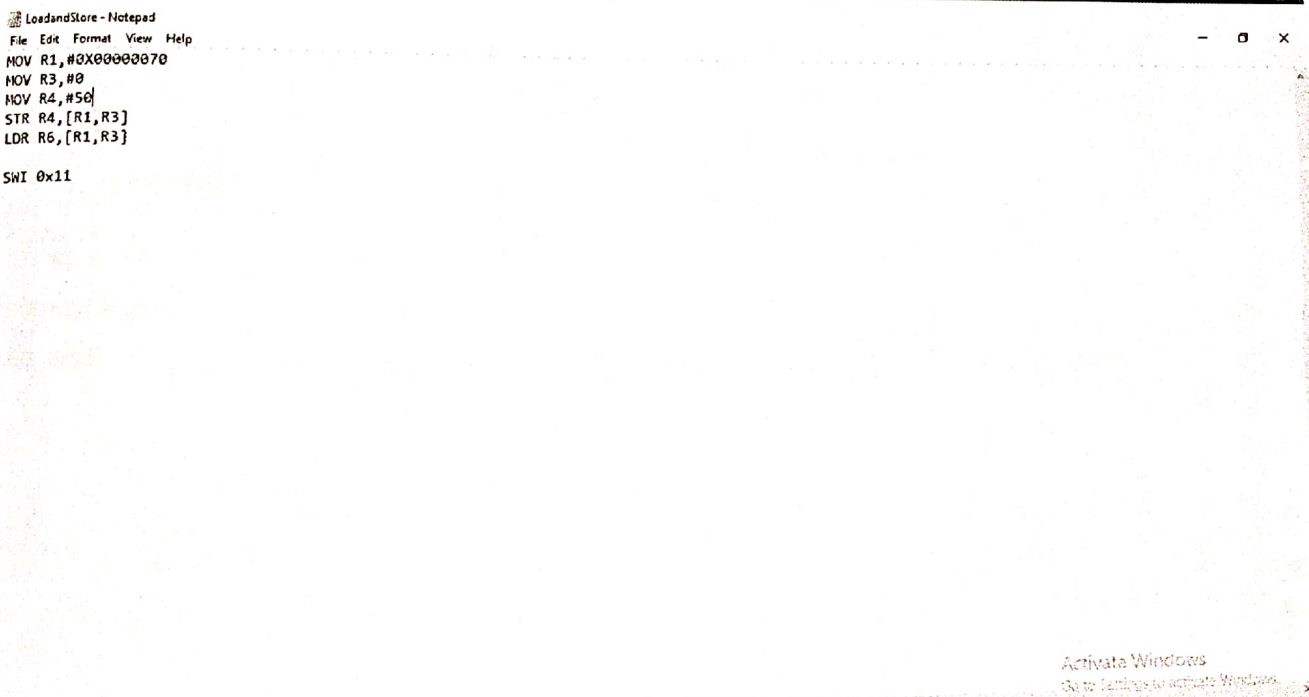
OutputView WatchView

Type here to search

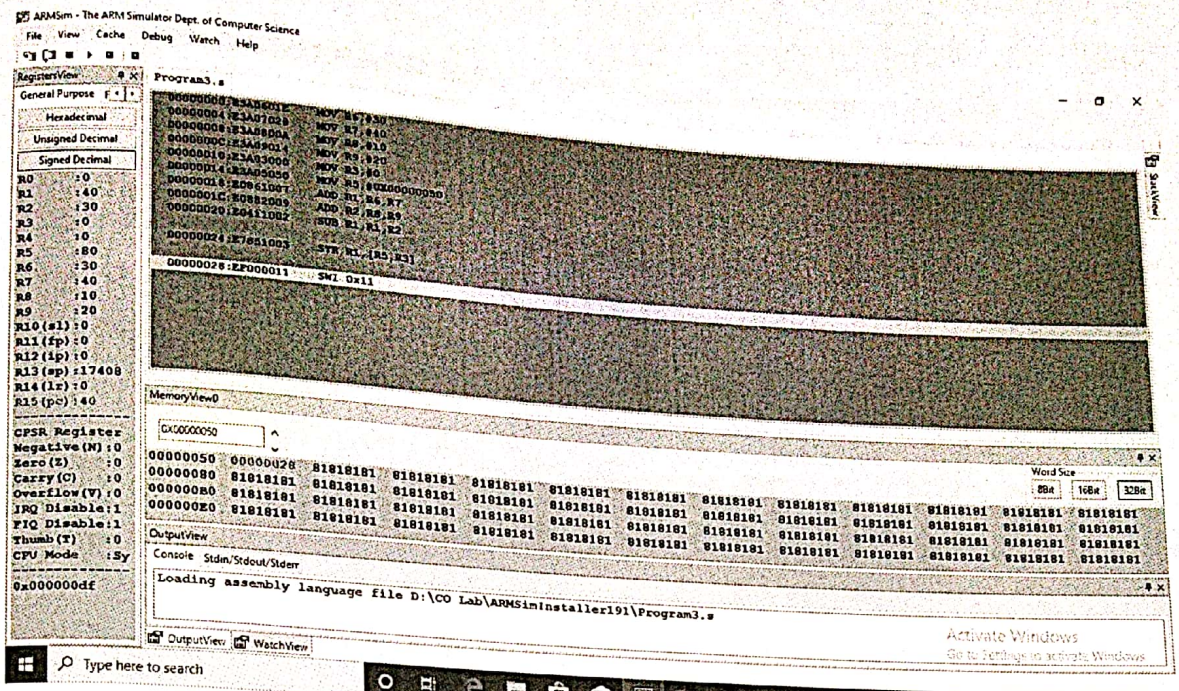
ENG 11:38 AM 04-02-2020

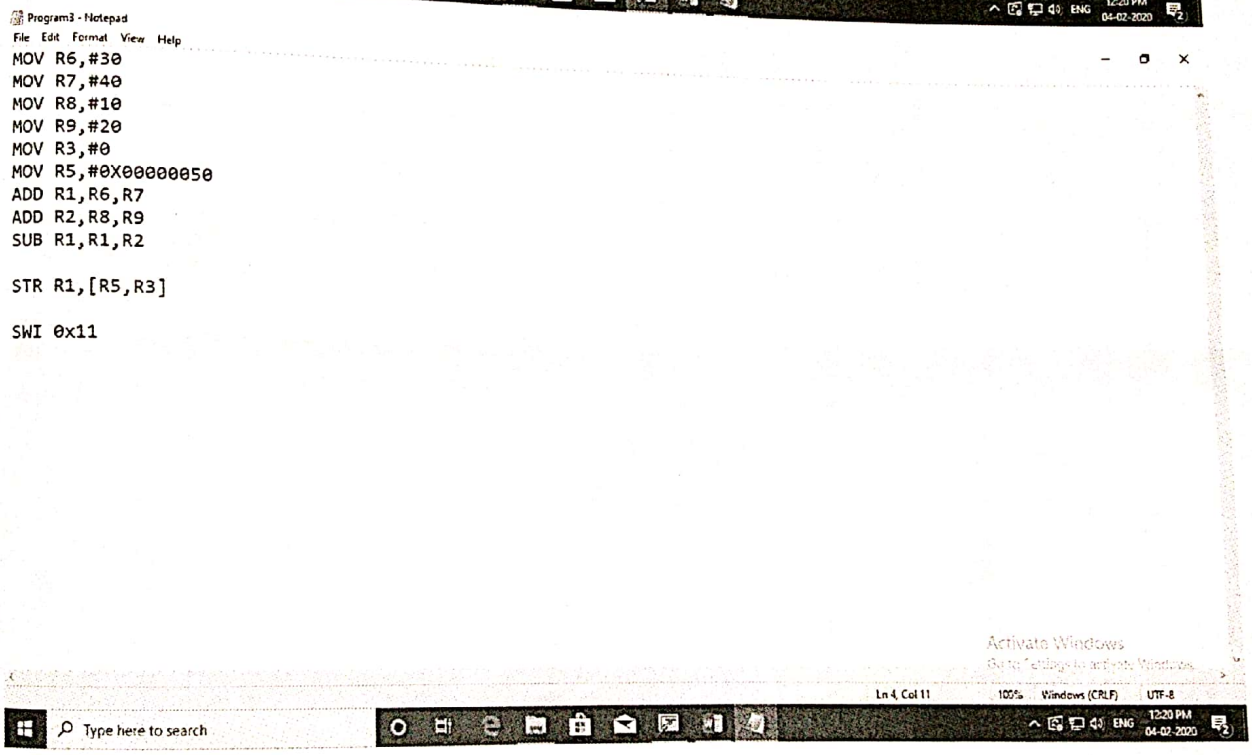
Program 2



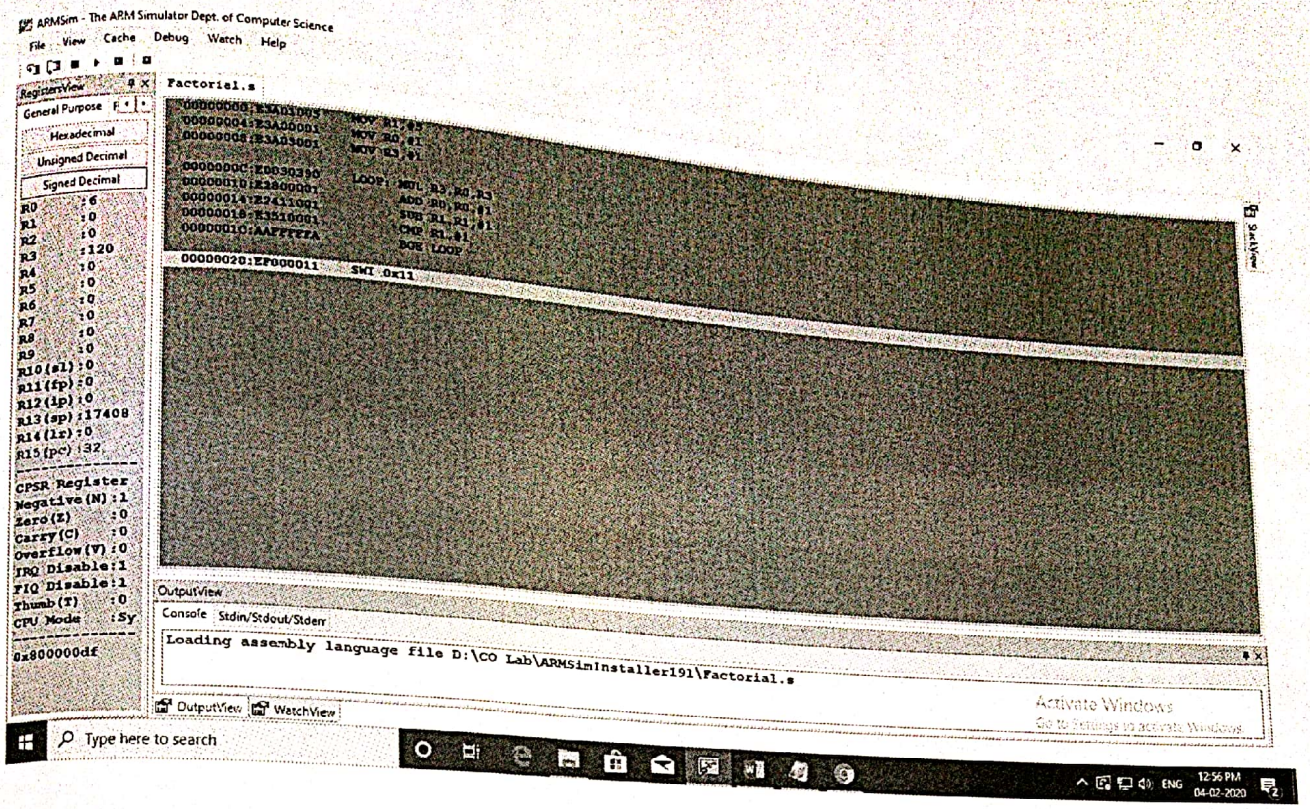


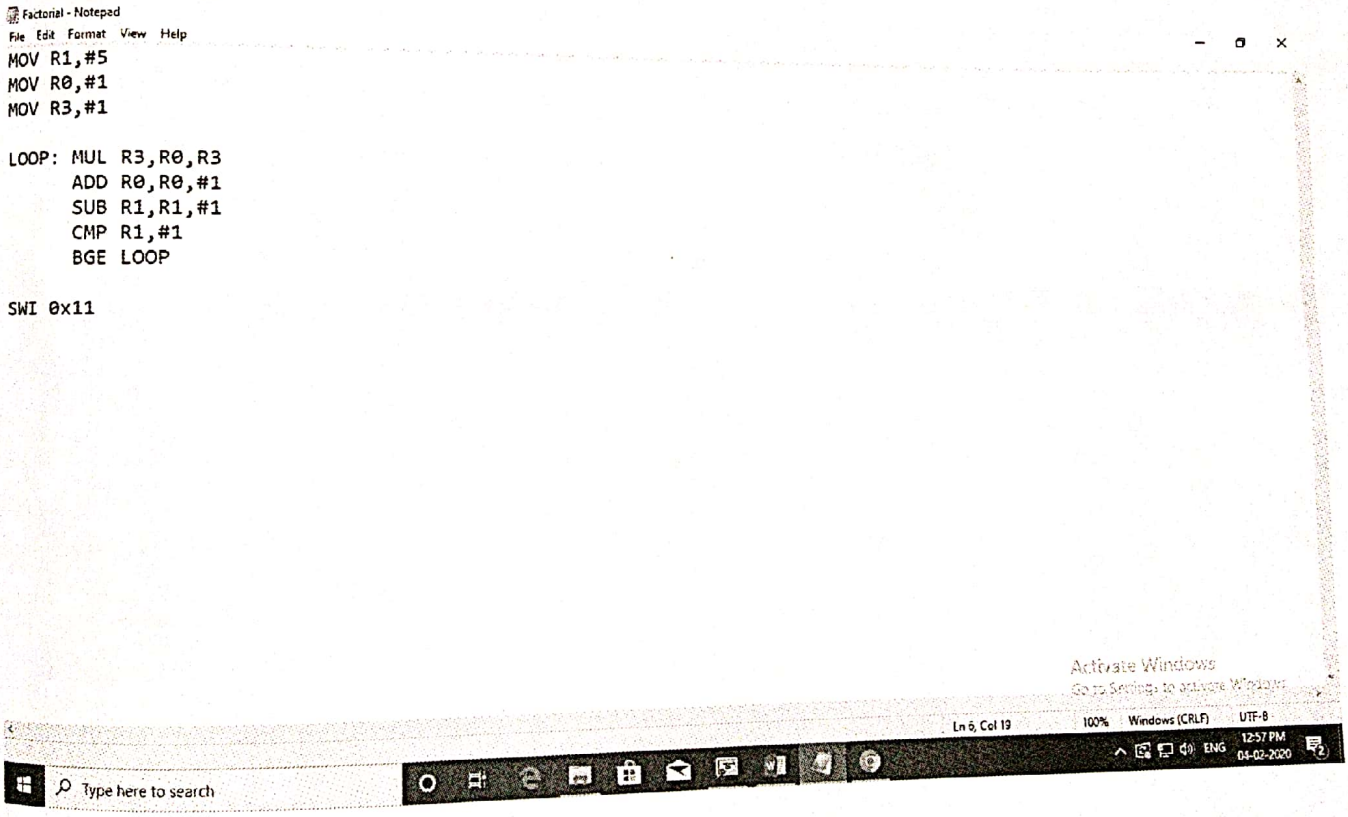
Program 3





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: Manish V		
USN: 1MS18CS067	**Marks:** /10	**Date:**
	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,R5,#4
      SUB R2,R2,#1
      CMP R2,#2
      BGT loop
SWI 0x11
```

SUM ARRAY

2)
```
MOV R0,#0x00001000
MOV R1,#0
MOV R2,#5
MOV R5,#1
MOV R6,#0
MOV R4,#0
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
add:
    LDR R6,[R0,R1]
    ADD R1,R1,#4
    ADD R4,R4,R6
    SUB R2,R2,#1
    CMP R2,#0
    BGT add
STR R4,[R0,R1]
SWI 0x0011
```

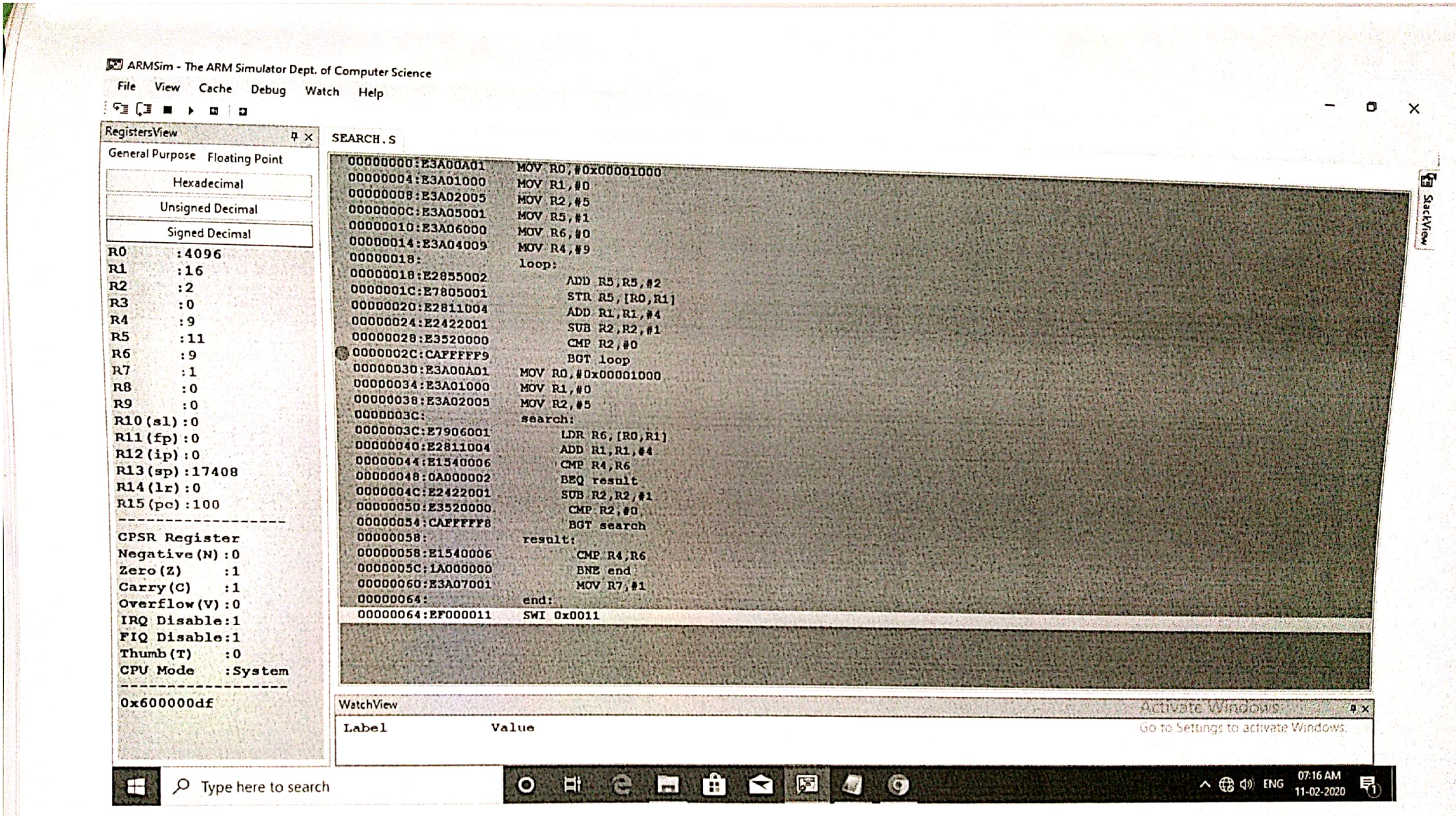
1) Search

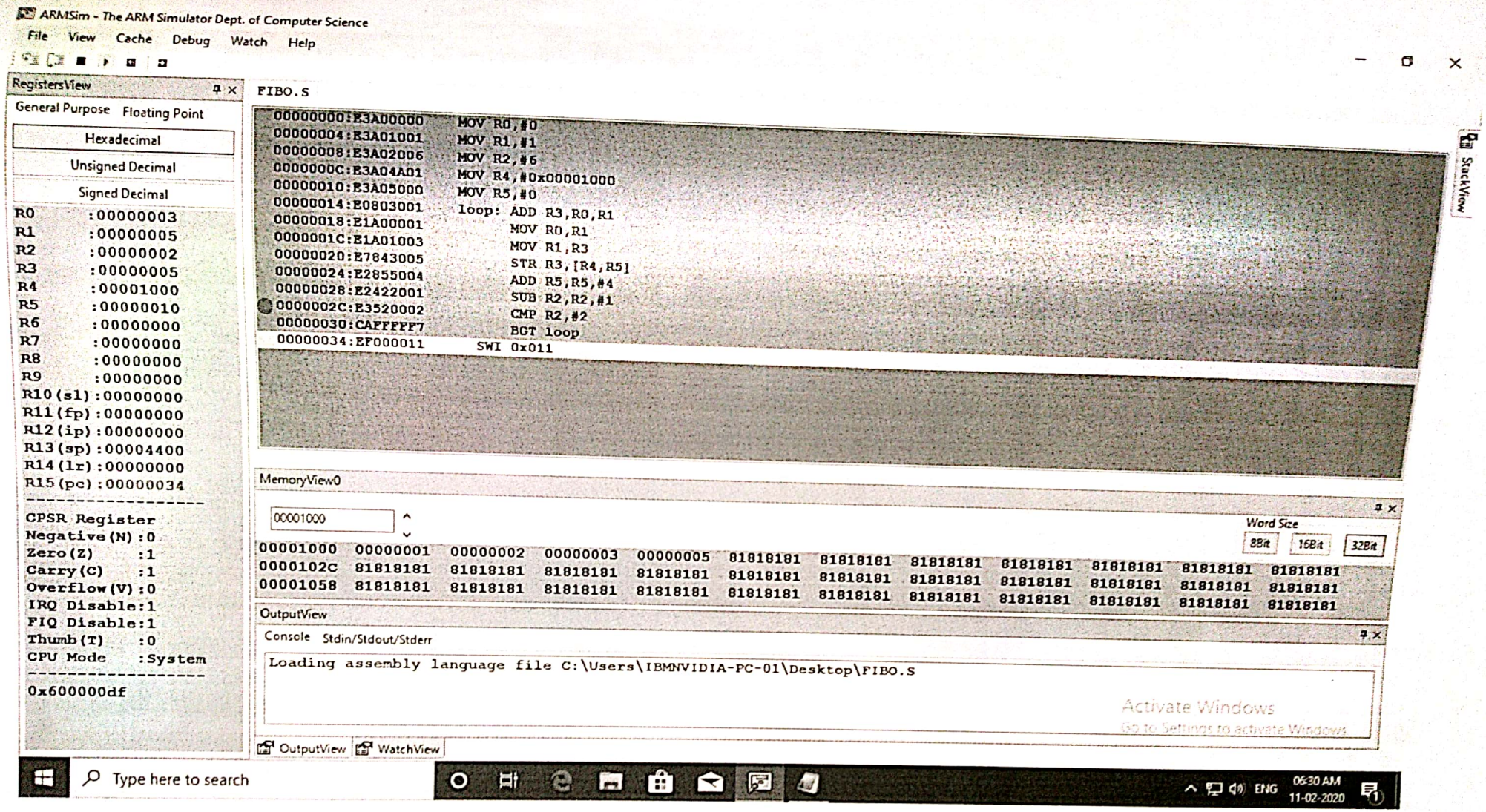
```
MOV R0,#0x0000a000
MOV R1,#0
MOV R2,#5
MOV R5,#1
MOV R6,#0
MOV R4,#9
loop: ADD R5,R5,#2
    STR R5,[R0,R1]
    ADD R1,R1,#4
    SUB R2,R2,#1
    CMP R2,#0
    BGT loop
MOV R0,#0x00001000
MOV R1,#0
MOV R2,#5
search:
    LDR R6,[R0,R1]
    ADD R1,R1,#4
    CMP R4,R6
    BEQ result
    SUB R2,R2,#1
    CMP R2,#0
    BGT search
result:
    CMP R4,R6
    BNE end
    MOV R7,#1
end:
SWI 0x0011
```

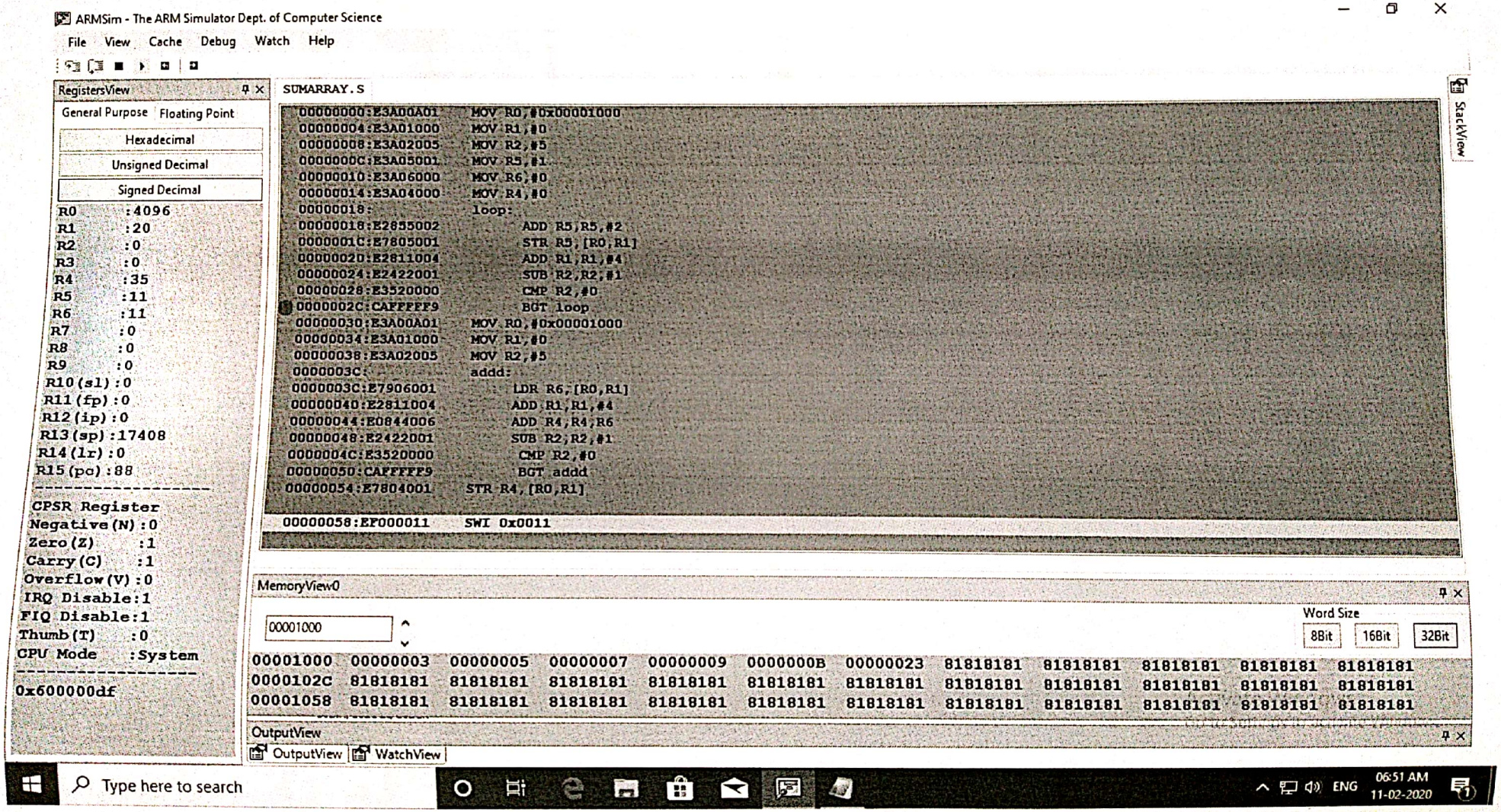
Results/Conclusions and Snapshots: Take the snap shot of registers file and memory view

3)

```
.equ SWI_open, 0x66
.equ SWI_close, 0x68
.equ SWI_print, 0x66
.equ SWI_RdInt, 0x60
.equ stdout, 1
.equ SWI_prstr, 0x69
.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 start:
MOV r5, #student
MOV r0, r9
LDR r8, = Array
bcs afterloop
str r8, [r8, r5]
add r5, r5, #4
MOV r1, r0
MOV r0, #stdout
Sen SWI, print
add r4, r4, #1
LDR r1, = Newline
str SWI_prstr
swi SWI_prstr
bal loopstart
afterloop:
MOV r5, #20
loop: LDR r1, [r8, r5]
SUB r4, r4, #1
SUB r5, r5, #4
MOV r1, r2
MOV r0, #stdout
LDR r4, Newline
stri 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: Manish V	**Marks: /10**	**Date:** 23 | 5 | 20
USN: 1MS18CS067	**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

1) Add the two i/p pins, Name them A & B.
2) Add OR, AND, EX-OR, 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 pin and name it Result.
5) Add a 1-bit multipower with 3 select bits.
6) Connect the outputs of all gates to the muse.
7) Connect 3-bit input pin to muse.
8) Add an Ex-OR gate. Connect its o/p to Cout. The first i/p must be connected to B and the second to other i/p pin sub.
9) Add i/p pin to cin and o/p pin to cout.
10) Add another i/p and name it Less connect it to muse.
11) Add an output pin and name it set. Connect it to the multiplexer o/p of adder unit.

Snapshots :

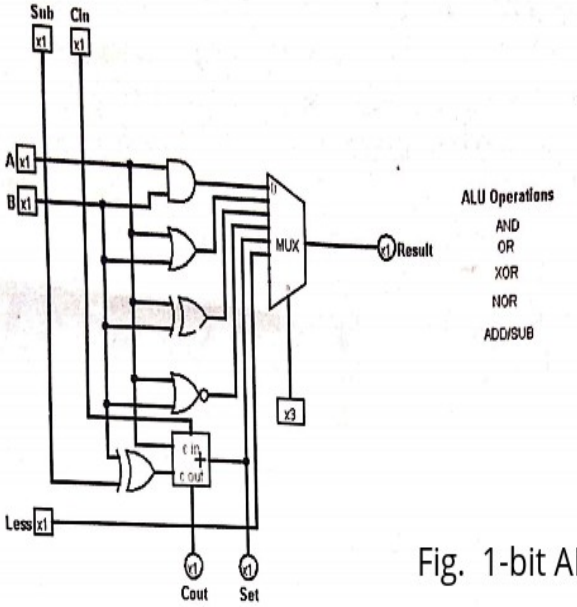
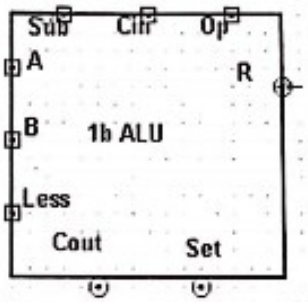
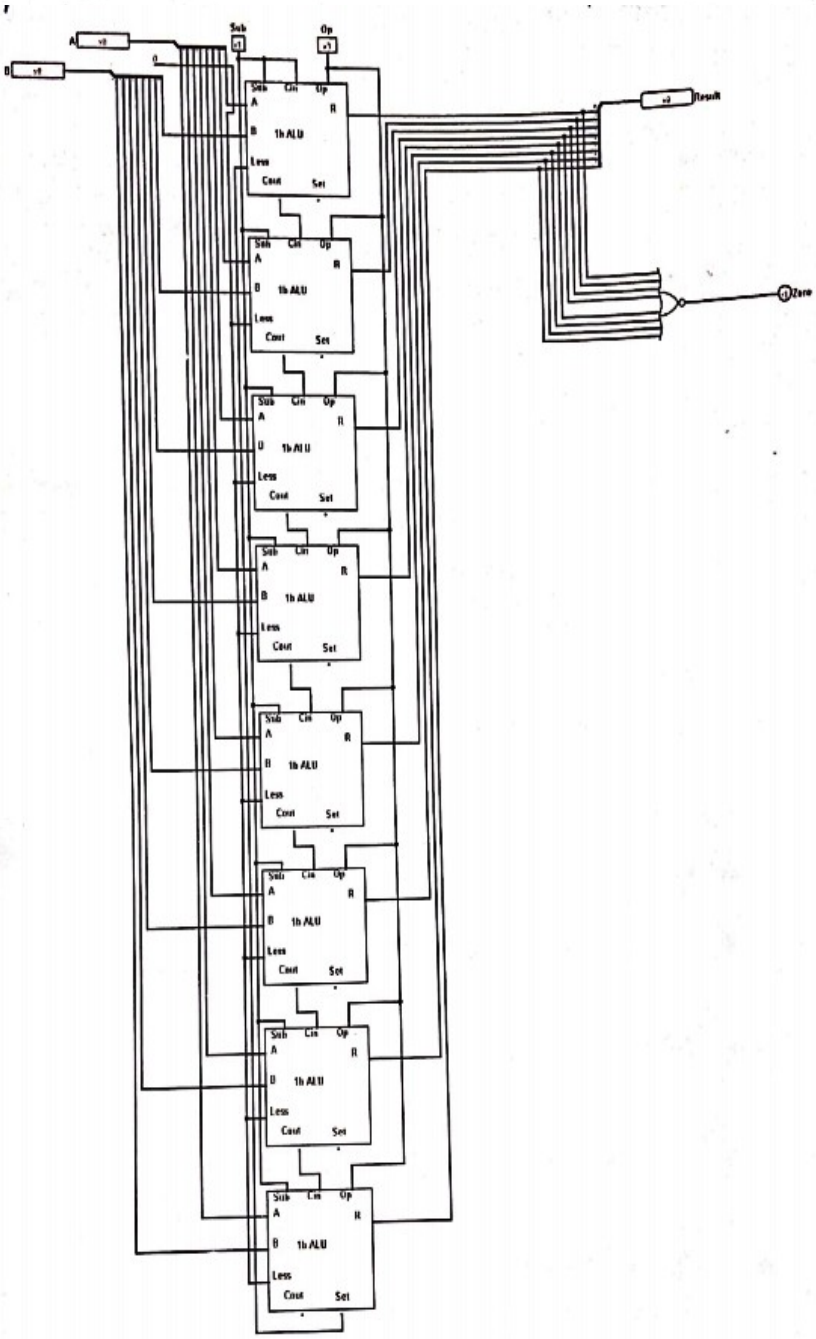


Fig. 1-bit ALU



ALU Object

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: Manish V	**Marks:** /10	**Date:** 23/5/20
USN: 1MS18CS067	**Signature of the Faculty:**	

Objective: To simulate the writing operation on memory.

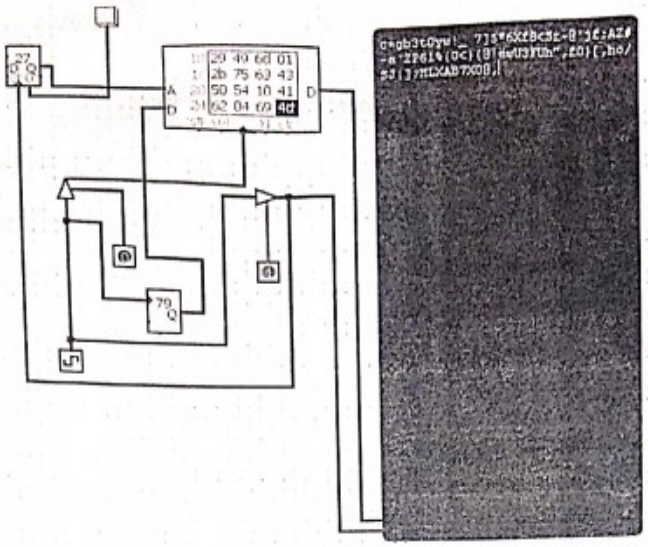
Simulator Description: Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to 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.

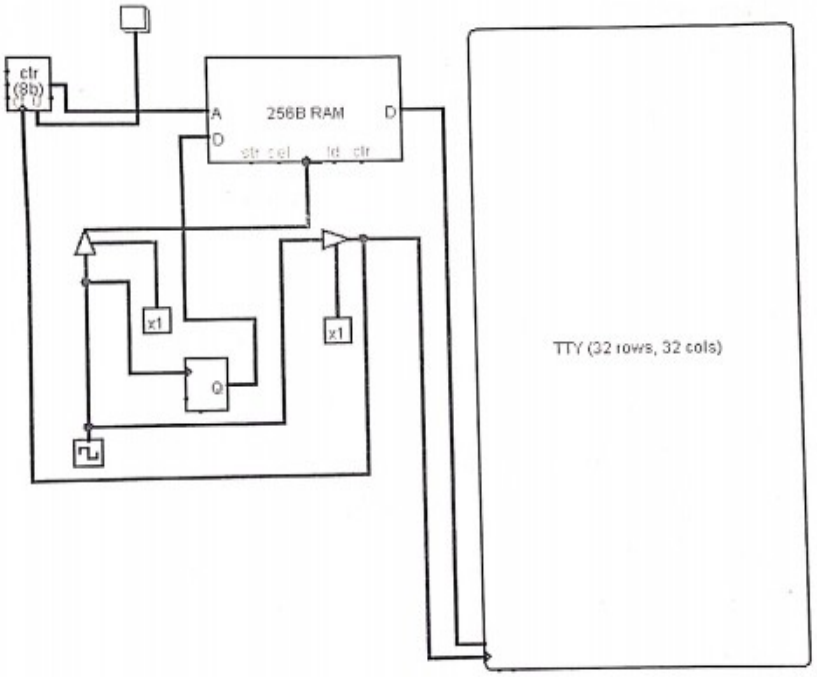
Activity to be performed by students:

List out the steps in designing memory system

1) Add a RAM with seperate load and store selected.
2) Add a counter and convert Q to A of the RAM.
3) Add a controller buffer and convert its o/p to RAM.
4) Add a clock and connect to the i/p of the buffer.
5) Add a TTY 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 controlled buffer connect it to TTY. 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.

Snapshots:


27
A
D
29 49 60 01
2b 75 63 43
50 54 10 41
62 04 69 4d
D
79
Q


ctr
(8b)
A
256B RAM
D
D
x1
x1
Q
TTY (32 rows, 32 cols)

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: Manish V	**Marks:** /10	**Date:** 23/5/20
USN: 1MS18CS067	**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:

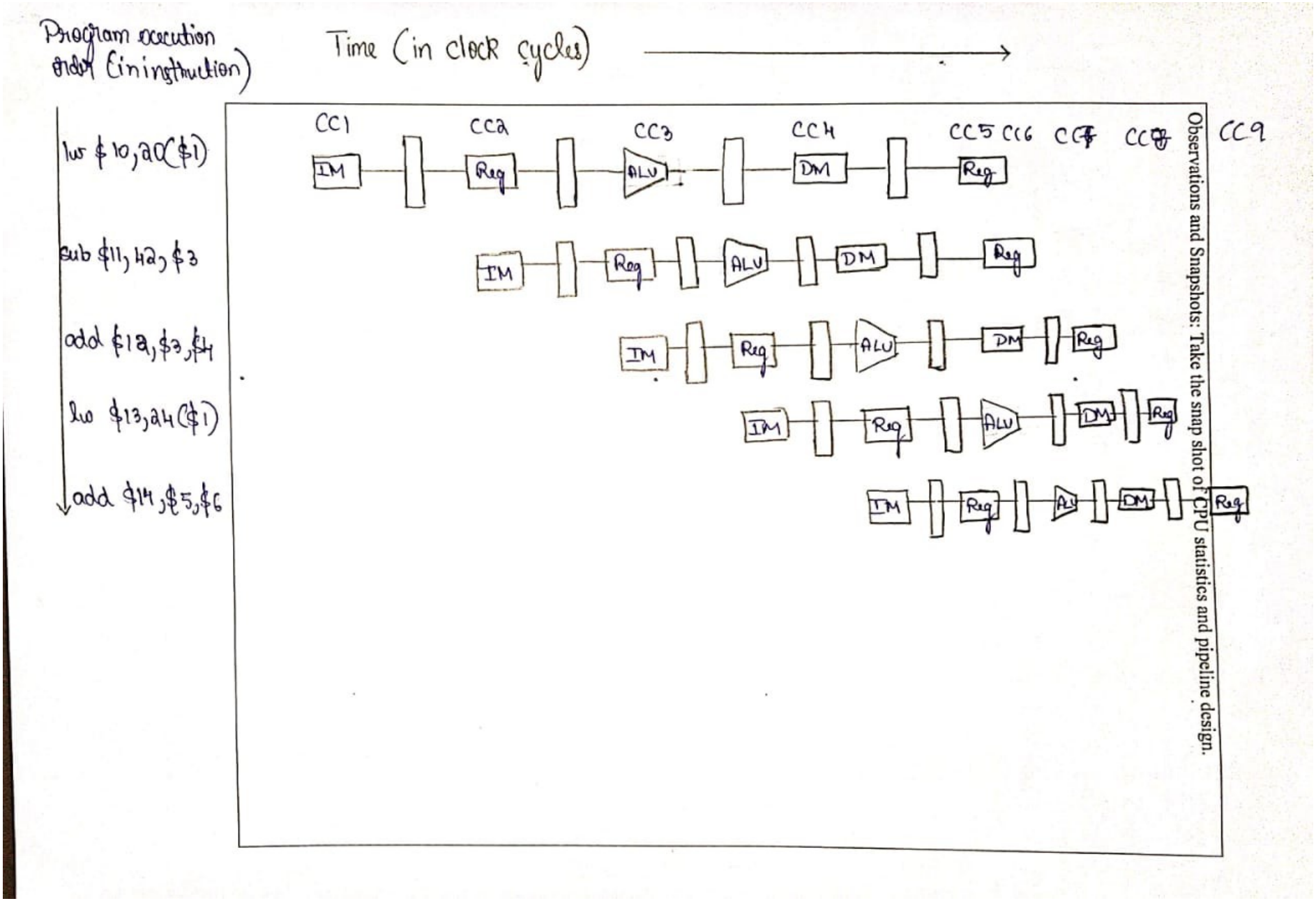
- Fetch the next instruction from memory into an internal CPU register.
- Decode the instruction to determine which function sub-circuits it requires.
- Read any input operands required from high-speed registers or directly from memory.
- Execute the operation using the selected sub-circuits.
- Write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

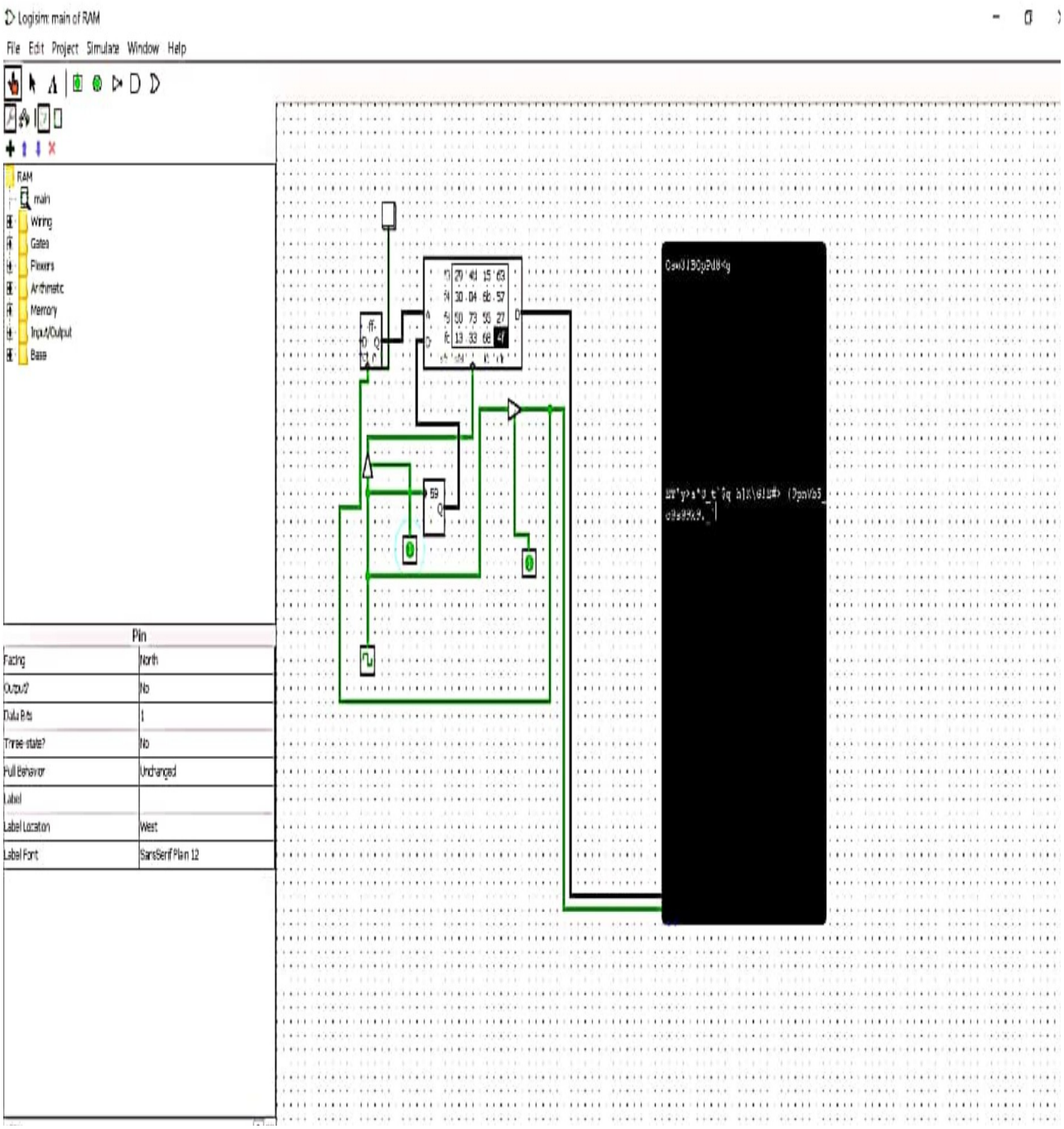
Activity to be performed by students:

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
```


Program execution order (in instruction)
Time (in clock cycles)
lw $10,20($1)
sub $11,42,$3
add $12,$3,$4
lw $13,24($1)
add $14,$5,$6
CC1
CC2
CC3
CC4
CC5
CC6
CC7
CC8
CC9
IM
Reg
ALU
DM
Reg
Observations and Snapshots: Take the snap shot of CPU statistics and pipeline design.

Snapshots:


Logisim: main of RAM
File Edit Project Simulate Window Help
RAM
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Pin
Facing
North
Output?
No
Data Bits
1
Three-state?
No
Full Behavior
Unchanged
Label
Label Location
West
Label Font
SansSerif Plain 12