RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Activity - I

Name :	Moulidhar .B	Branch:	CSE
USN/Roll No. :	1MS18CS075	Sem/Sec:	IV B
Subject :	Computer organization & Architechture	Subject Code:	

(1) Write the detailed procedure to assemble a system

Ans- (1) Remove side panels on case :- After removing the case from the box, the panels are removed from this. Case with thumb screws.

(2) Insert motherboard :- In assembly process, as I was just transferring the parts from one case to another, leaving the CPU cooler installed was the easiest option. Depending on the motherboard, case, CPU and CPU fan, this might need to be done before installing or one in place.

(3) Check Clearance :- Being that this computer includes high performance components, some of them are large enough that clearance can become an issue.

(4) Frontpanel connections :- once the graphics card was removed again, it is time to attach the connections for the buttons, light, USB ports & audio connections.

(5) Install Power Supply :- The powersupply from the previous case was modular so only the cables that are needed are plugged into the unit.

6. Power motherboard :- with the motherboard power being the largest cable and sometimes just long enough. Running this cable first and plugging it into the board, if there is a second cable for the CPU remember to connect it as well

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

8. Installing the hard drives :- The size and number of hard drives your computer contains is completely dependent on your style of use and storage needs.

9. Connect cables :- It is the cables are keyed so they will only fit in one direction into the board, the cable that is attached to the optical drive.

10. Install RAM :- If your computer uses more than one stick refer to the manual for which slot to install the stick.

11. Install graphics card & Expansion cards :- If your computer does not come with a graphics card integrated into the motherboard, then insert one.

12. Find product :- The assembly of a brand new computer can take several hours just to remove and mount in a case.

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

Ans. Troubleshooting is a systematic approach to problem solving that is often used to find and correct issues with complex machines, electronics, computers and software systems.

The first steps in troubleshooting is gathering information on the issue, such as undesired behaviour or a lack of expected functionality. Other important information includes related symptoms and special circumstances that may be required to reproduce the issue.

once the issue and how to reproduce it are understood, the next step might be to eliminate unnecessary components in the system and verify that the issue persists, to rule out incompatibility and third-party causes.

Continuing, assuming the issue remains, one might next check common causes. Depending on the particular issue and the troubleshooter's experience, they may have some ideas. They may also check product documentation and/or conduct research on a support database or through a search engine.

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

Ans (1) disconnect the power cable from your system and if needed, unplug other back-panel cables so that you can safely turn your system on to its side.

(3) List out the procedure to install extra memory card to System.

Ans 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're most commonly found next to the processor and its corner. If there already RAM in your System, eject it by pressing firmly on the tabs on the motherboard at either end of the slots. The memory stick will pop out and you can remove them gently.

Step 3:- To install the new RAM, line up notches in the bottom of the sticks in gaps in the slot on the motherboard. As its 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 into hold the sticks firmly in their slots. and close the pc back up. plug all the cables back in and try to boot the System.

(4) Explain with neat diagrams, various 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 port on Computer

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	Moulidhar.B	Branch:	CSE
USN/Roll No. :	1MS18CS075	Sem/Sec:	IV B
Subject :	COA Lab	Subject Code:	

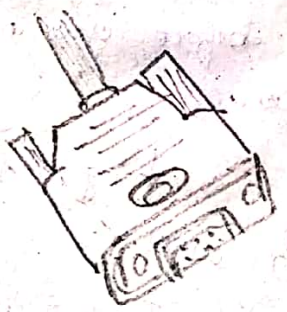
4) PS/2 Cable :- Connect one end to PS/2 keyboard, PS/2 mouse. Connect other end to PS/2 ports on computer. Purple PS/2 port keyboard.

5) Ethernet Cable :- Connect one end to router, network switch. Connect other end to ethernet port of computer.

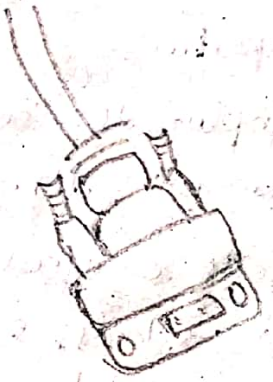
6) USB Cable :- Connect one end to USB device, connect other end to USB ports on computer.

7) Computer power cord (kettle plug) – connect one end to AC power socket. Connect other end to power supply unit of computer monitor.

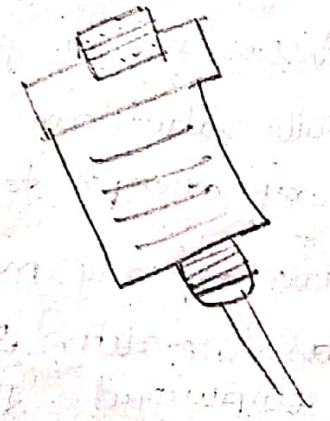
diagram



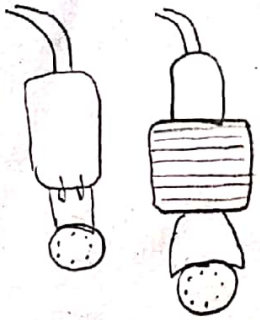
VGA Cable



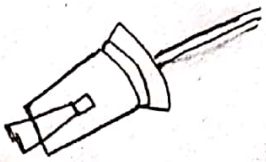
DVI Cable



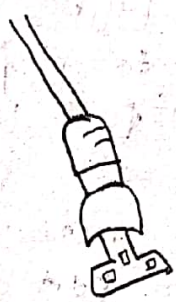
HDMI Cable



PS/2 cable



Ethernet cable



Computer Power cord

USB cable

(5) ~~Write out various cab~~

(5) List out safety precautions while ~~dis~~ removing the components from the system.

Ans. A few warnings and remainders before you start dissassembling your computer.

1) Fully shutdown and unplug the computer before you make any attempts to dissassemble the tower.

2) Take off any metal objects on your arms or fingers such as bracelets, rings or watches. Even if 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 realm as seen in the previous one.

5) Before touching any part within the case, 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.

7) Handle all parts with care.

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

9) Be careful when handling the mother board.

10) Never attempt to remove the power source, a box attached to the side of the unit to which all cables are connected.

11) When removing any cables, wires or ribbons make use of grasp the wire at its base or head to keep it from breaking

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

13) Take note that three most damaging things to 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: Moulidhar.B	**Marks:** 10/10	**Date:** 29/01/20
USN: 1MS18CS075	**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.

RAMAIAH
Institute of Technology

Data Sheet

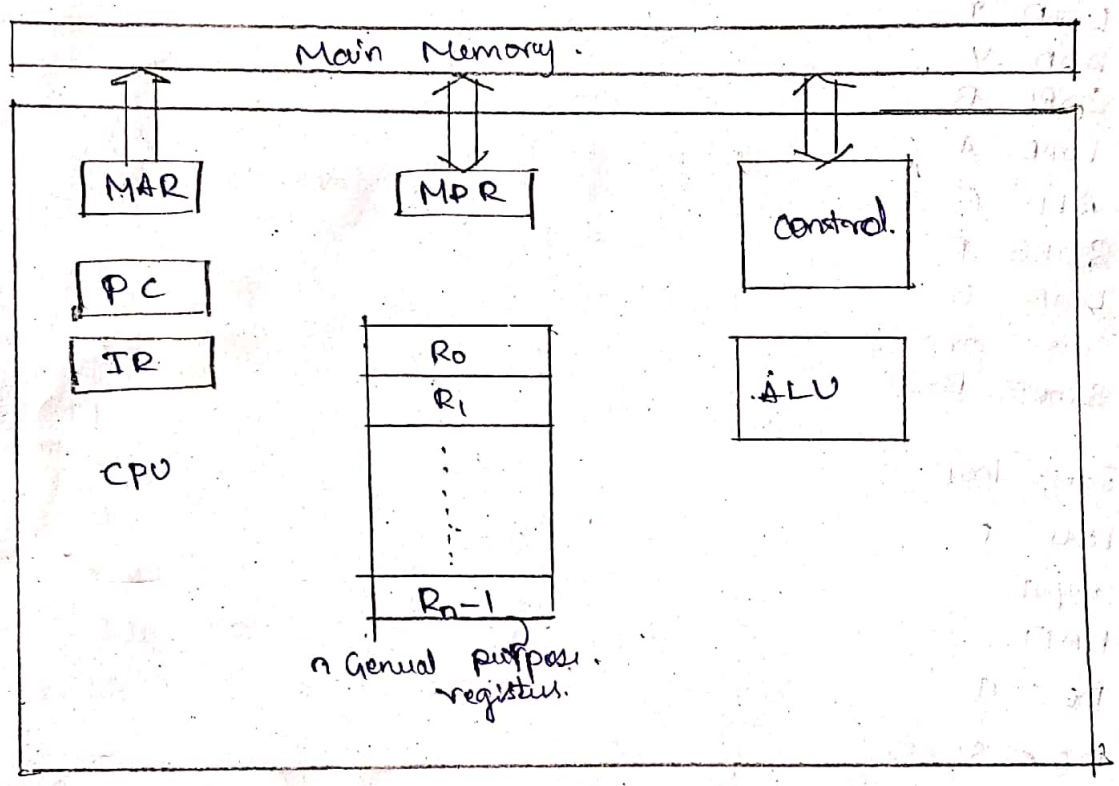
MARKS :

Activity - II

Name :	MOULIDHAR B	Branch:	CSE
USN/Roll No. :	1MS18CS075	Sem/Sec:	IV / B
Subject :	COA Lab	Subject Code:	

(i) Draw the interconnection between memory and processor.

Ans:-



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

Ans:- There are 6 steps involved in the execution of an instruction.

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

Ans:-

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

Ans :

```
        LOAD  B
        STORE 0
        LOAD  B
        ADD   x
        STORE x
        SUBT  one
        STORE 0

        JUMP  first
second  LOAD
        ADD   4
        STORE 4
        LOAD  C
        SUBT  one
        STORE C

        JUMP  second
third   LOAD  four
        ADD   2
        STORE 2
        LOAD  4
        SUBT  on
        STORE 4

        JUMP  third

        LOAD  D
        ADD   x
        SUBT  2
        OUTPUT
        HALT
A       DEC 6
B       DEC 3
C       DEC 2
D       DEC 0
x       DEC 0
4       DEC 0
2       DEC 0
D       DEC 0
one     DEC 1
four    DEC 4
```

(14) Describe the factors affecting the performance of a processor

Ans :- There are few factor affecting the performance of a processor:

* Multiple cores :- Nowaday we have dual, quad and even octa core processors with its own fetch and execute cycles. However, the software should make use of the multiple cores.

* Clock Speed :- The processor requires a clock pulse in order to operate correctly. One clock cycle = 1 Hz. A PC clock speed is normally in the GHz region.

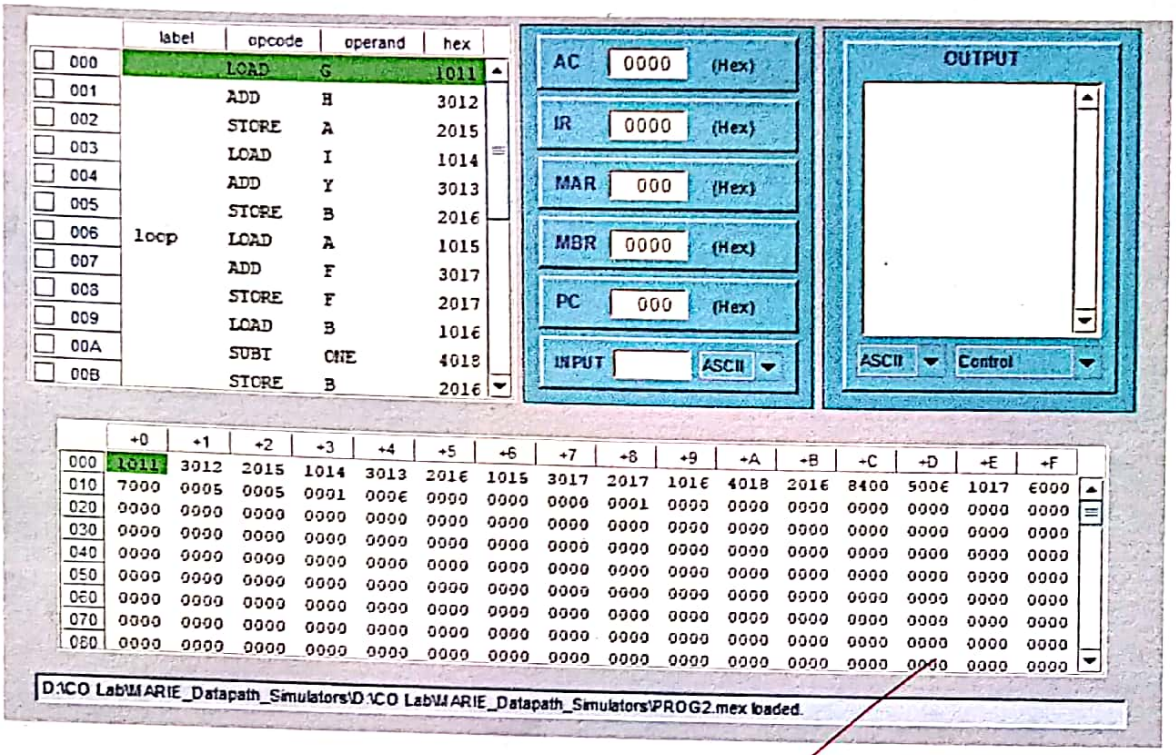
* Cache memory : It is a small amount of high performance RAM that is built into the processor. This RAM stores the data which has to be repeatedly used by the processor and it does not require a request from memory.

* Word length : the no. of bits the CPU can process simultaneously. For example, a 32-bit processor is faster than a 16 bit processor because of the wider 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 bits means that you can have 256 addresses (0 to 255)

* Data Bus width : It is the no. of bits 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.

```
     Assembly listing for: PROG1.mas
                Assembled: Tue Jan 28 06:35:41 IST 2020

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

Assembly successful.

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

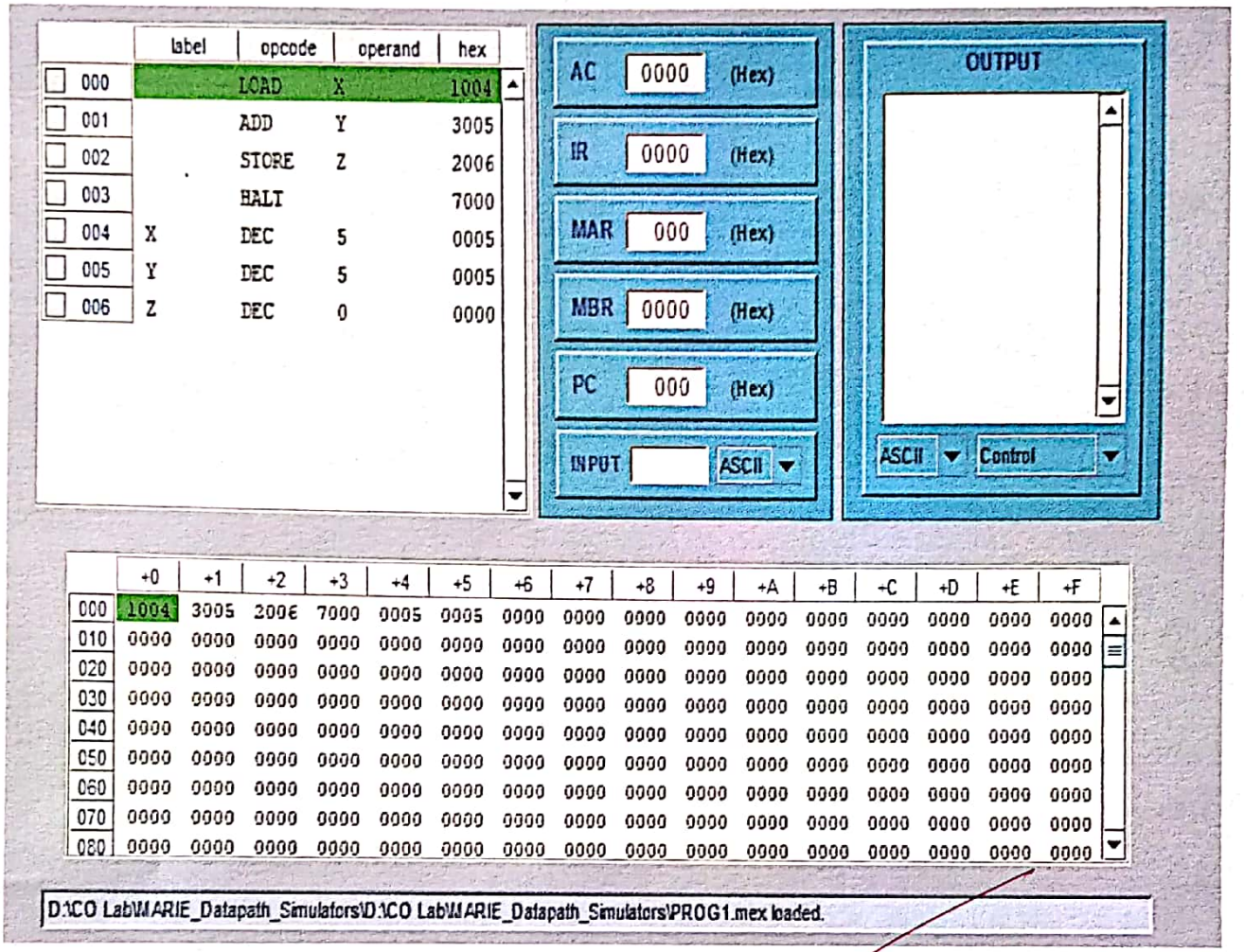


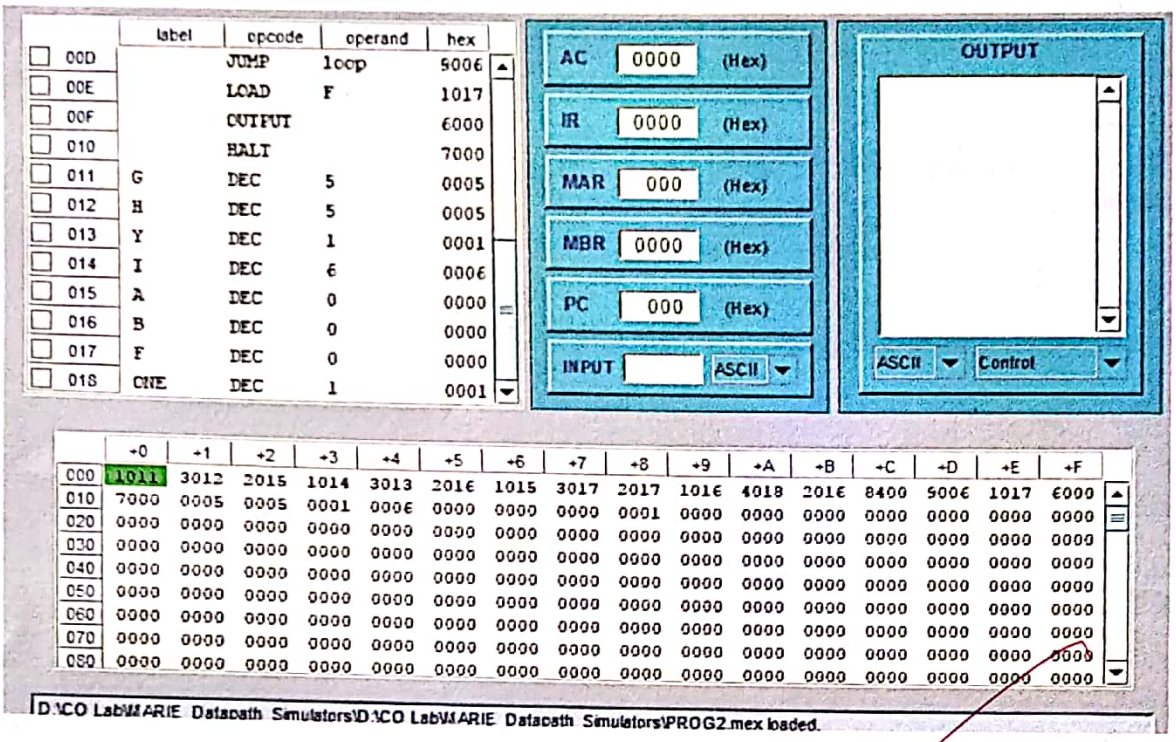
	label	opcode	operand	hex
000		LOAD	G	1011
001		ADD	H	3012
002		STORE	A	2015
003		LOAD	I	1014
004		ADD	Y	3013
005		STORE	B	2016
006	loop	LOAD	A	1015
007		ADD	F	3017
008		STORE	F	2017
009		LOAD	B	1016
00A		SUBT	ONE	4018
00B		STORE	B	2016

AC 0000 (Hex)
IR 0000 (Hex)
MAR 000 (Hex)
MBR 0000 (Hex)
PC 000 (Hex)
INPUT ASCII
OUTPUT
ASCII Control

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	1011	3012	2015	1014	3013	2016	1015	3017	2017	1016	4018	2016	8400	500E	1017	E000
010	7000	0005	0005	0001	000E	0000	0000	0000	0001	0000	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

D:\CO Lab\MARIE_Datapath_Simulators\D:\CO Lab\MARIE_Datapath_Simulators\PROG2.mex loaded.





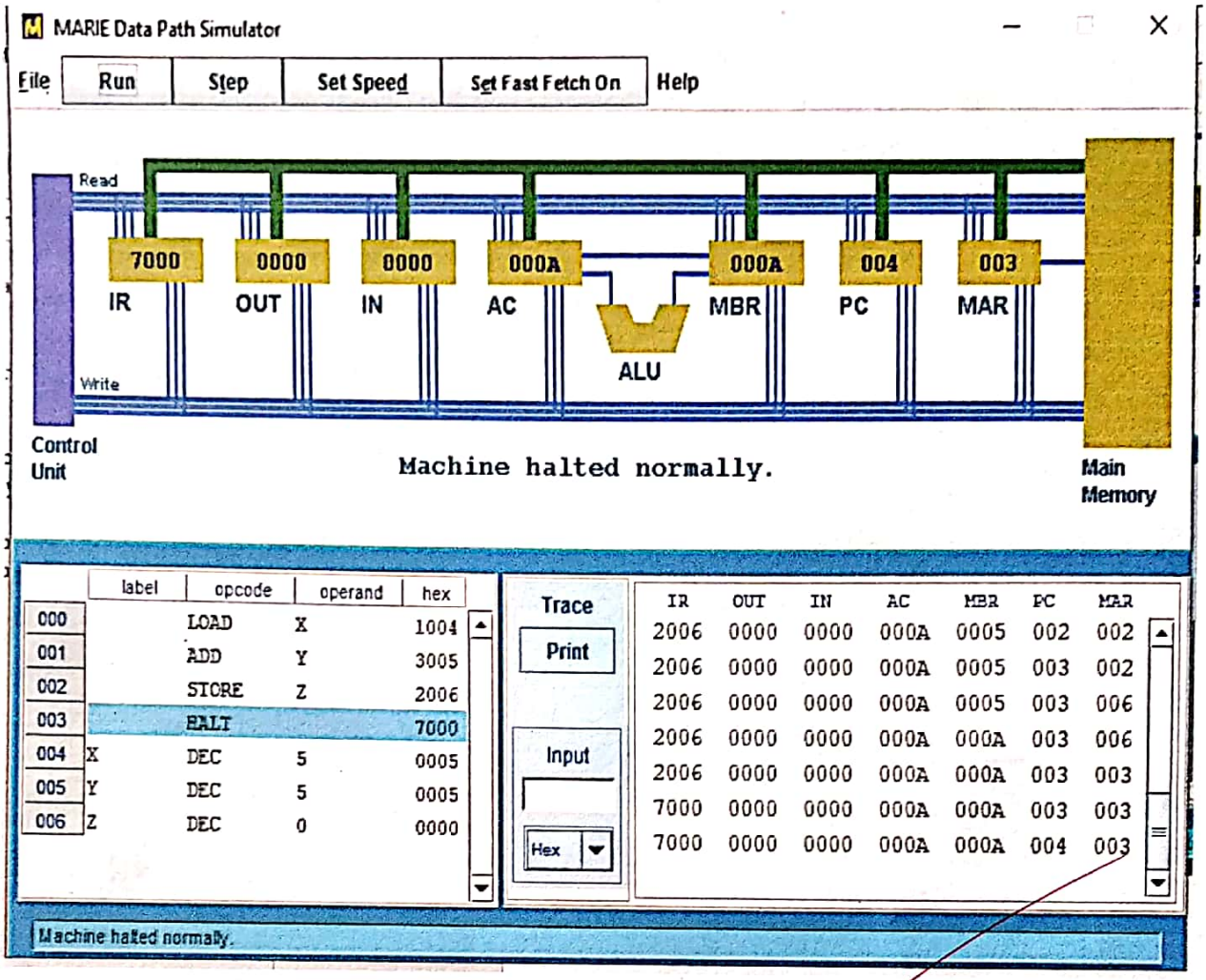
	label	opcode	operand	hex
00D		JUMP	loop	500E
00E		LOAD	F	1017
00F		OUTPUT		6000
010		HALT		7000
011	G	DEC	5	0005
012	H	DEC	5	0005
013	Y	DEC	1	0001
014	I	DEC	6	0006
015	A	DEC	0	0000
016	B	DEC	0	0000
017	F	DEC	0	0000
018	ONE	DEC	1	0001

AC 0000 (Hex)
IR 0000 (Hex)
MAR 000 (Hex)
MBR 0000 (Hex)
PC 000 (Hex)
INPUT ASCII

OUTPUT
ASCII Control

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	1011	3012	2015	1014	3013	201E	1015	3017	2017	101E	4018	201E	8400	500E	1017	E000
010	7000	0005	0005	0001	000E	0000	0000	0000	0001	0000	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

D:\CO Lab\MARIE Datapath Simulators\D:\CO Lab\MARIE Datapath Simulators\PROG2.mex loaded.



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: B.MONLIDHAR	**Marks:** /10	**Date:** 4/02/20
USN: 1MS18CS075	**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 :

Activity - III

Name :	1MS18CS075 / B MOULIDHAR	Branch:	CSE
USN/Roll No. :	1MS18CS075	Sem/Sec:	IV IB
Subject :	COA LAB	Subject Code:	

① Write an ARM program for arithmetic operations

Ans.
```
MOV  R5, #10
MOV  R6, #R0
ADD  R7, R5, R6
SWI  0X11
```

② Write an ARM program to demonstrate the working of load and 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. $f = (g+h) - (i+j)$

Ans
```
MOV  R6, #30
MOV  R7, #40
MOV  R8, #10
MOV  R9, #20
MOV  R3, #0
```

```
MOV   R8, #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 the sum of all the elements of an array.

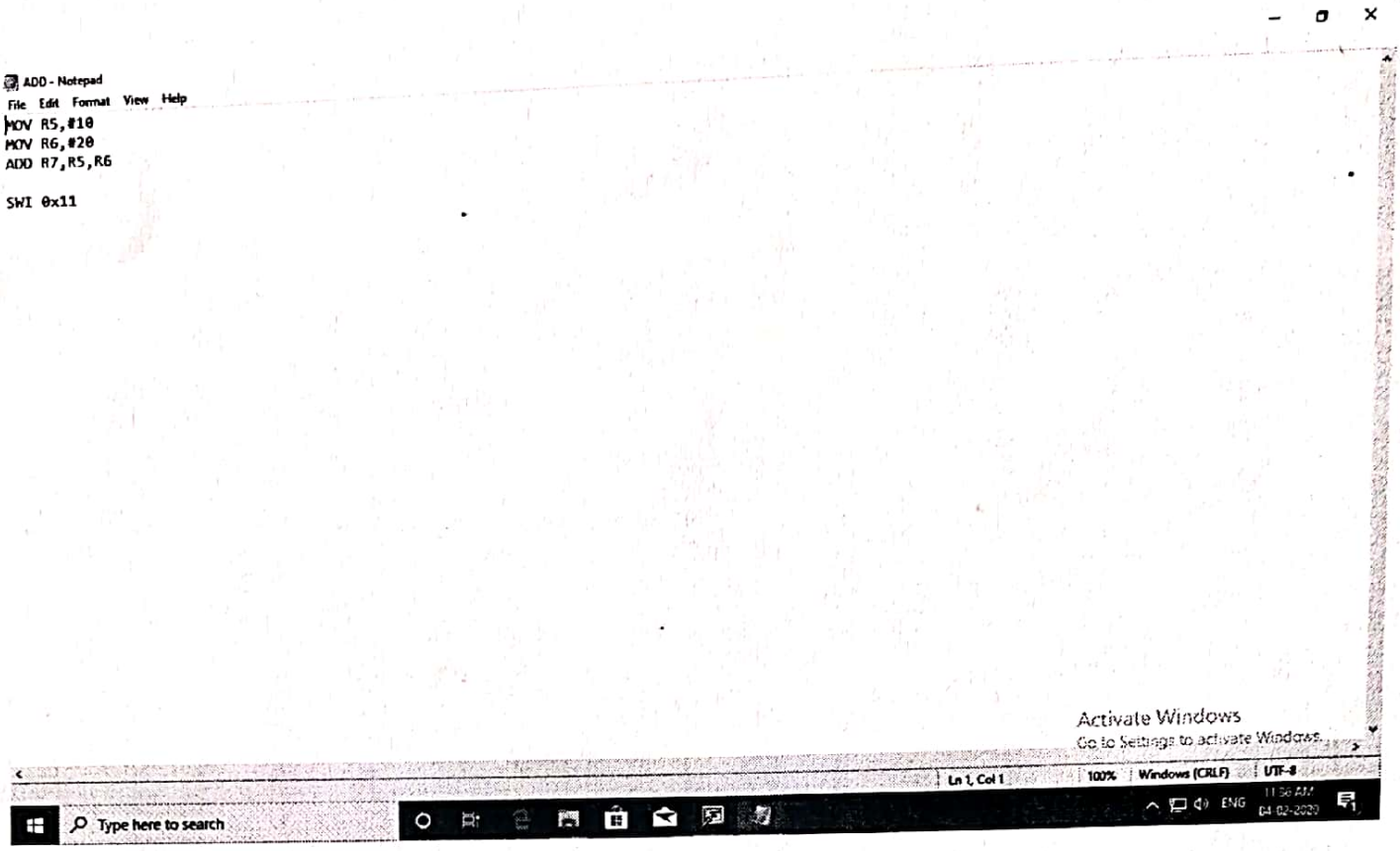
→
```
      MOV   R0, #5
      LDR   R1, =array.
loop  LDR   R2, [R1], #4
      ADD   R3, R3, R2
      SUB   R0, R0, #1
      CMP   R0, #0
      BNE   loop

# array DCD

      SWI   0X11
```

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

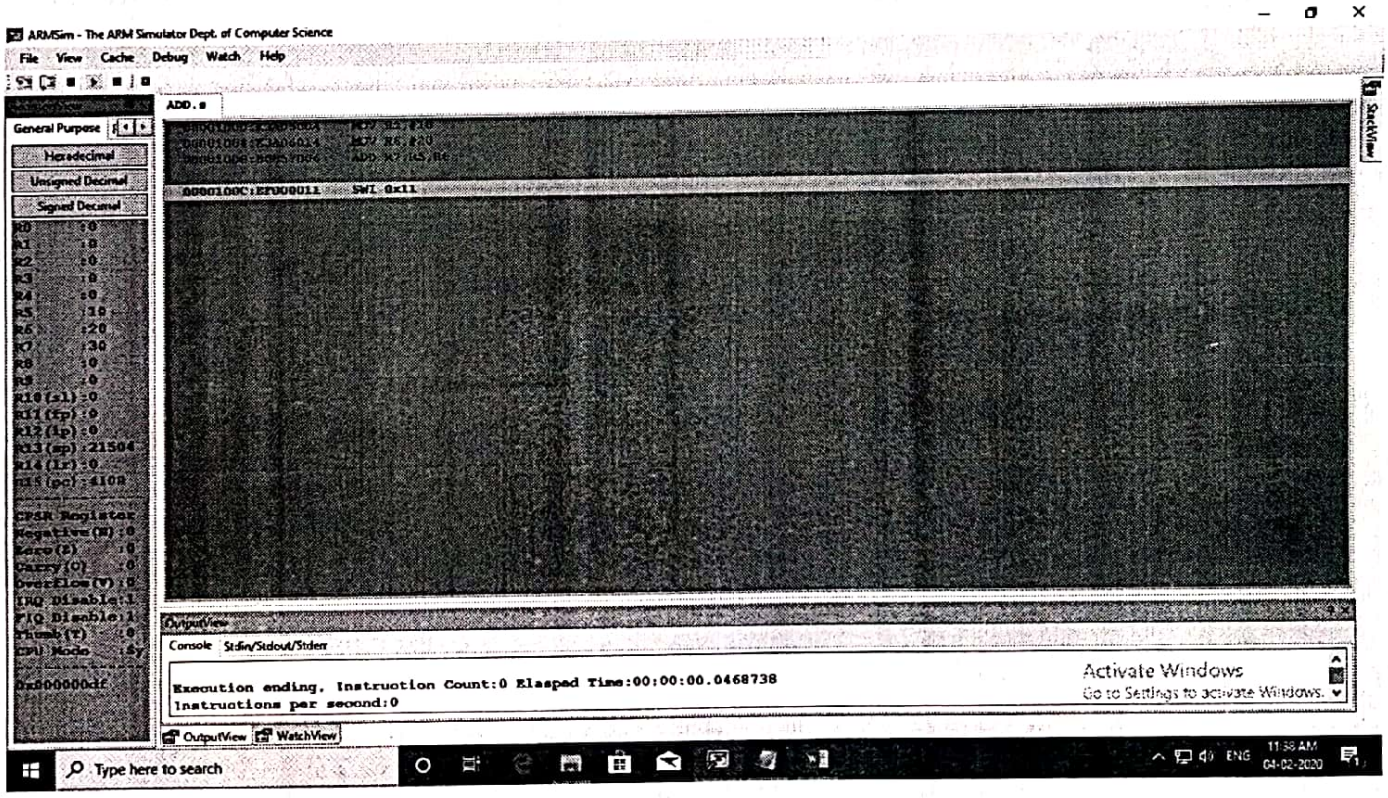
→
```
      MOV   R1, #5
      MOV   R0, #1
      MOV   R3, #1
loop  MUL   R3, R0, R3
      ADD   R0, R0, #1
      SUB   R1, R1, #1
      CMP   R1, #1
```

BGE Loop

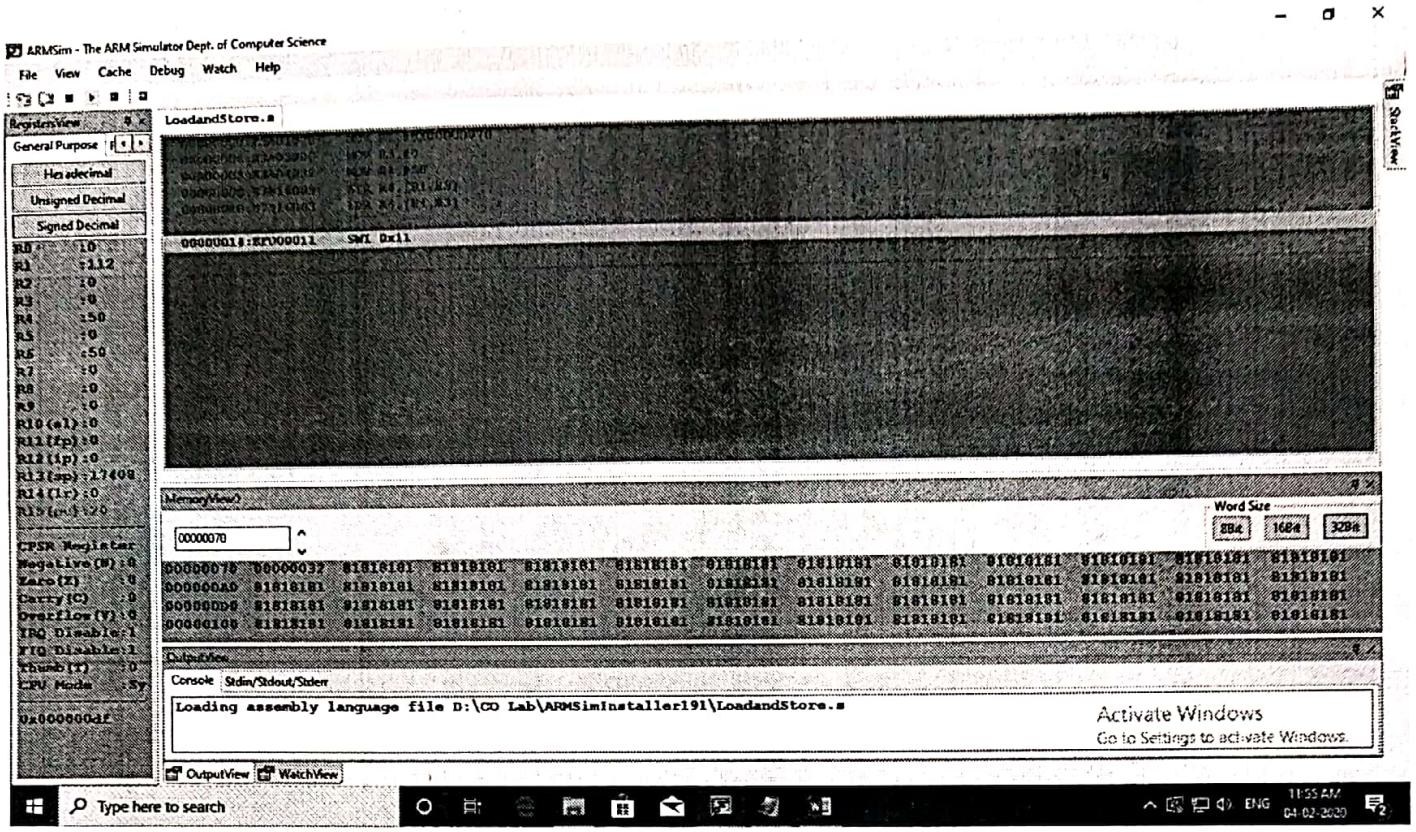
SWE 0X11

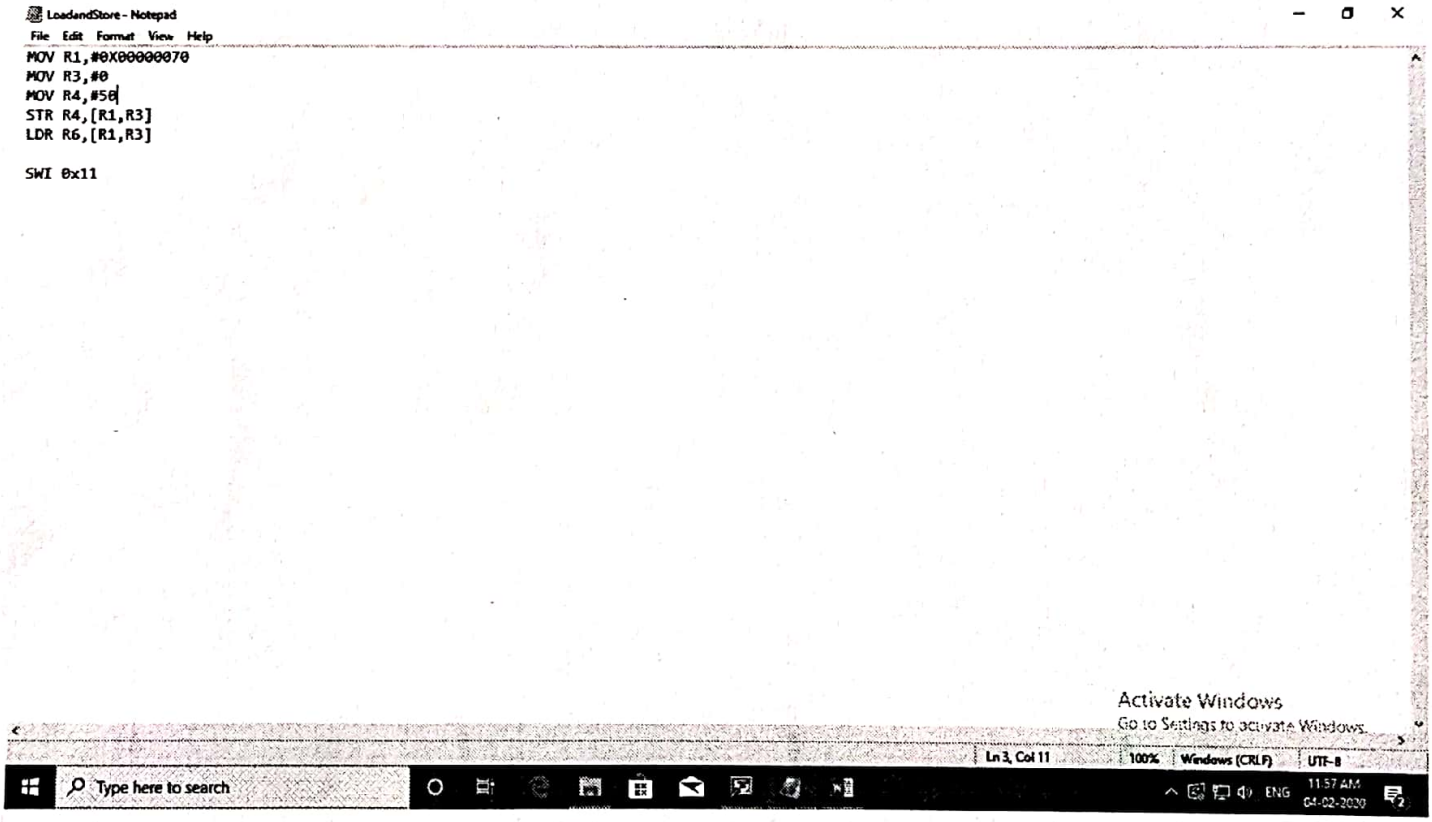


ADD.ASM

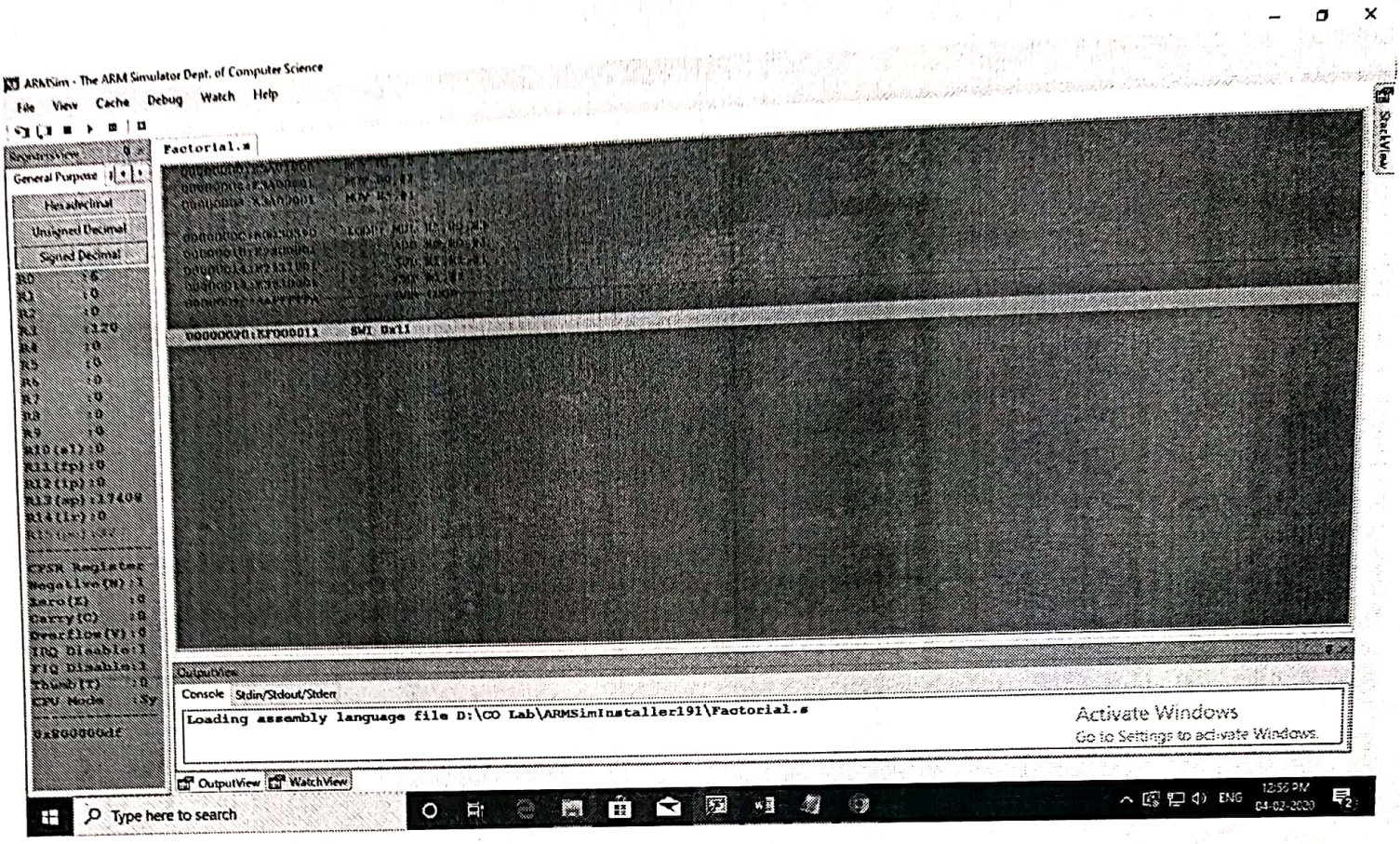


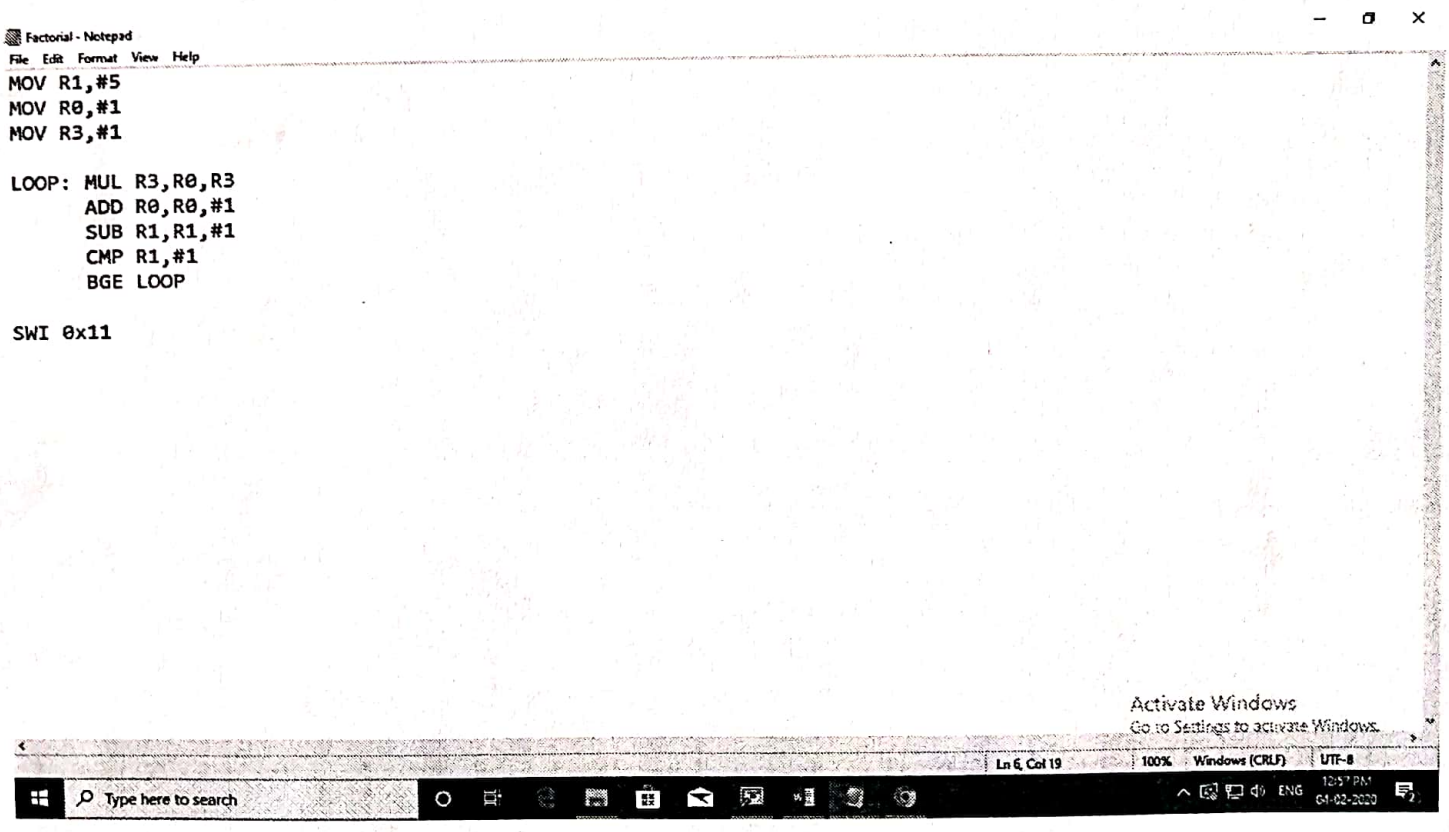
Program 2

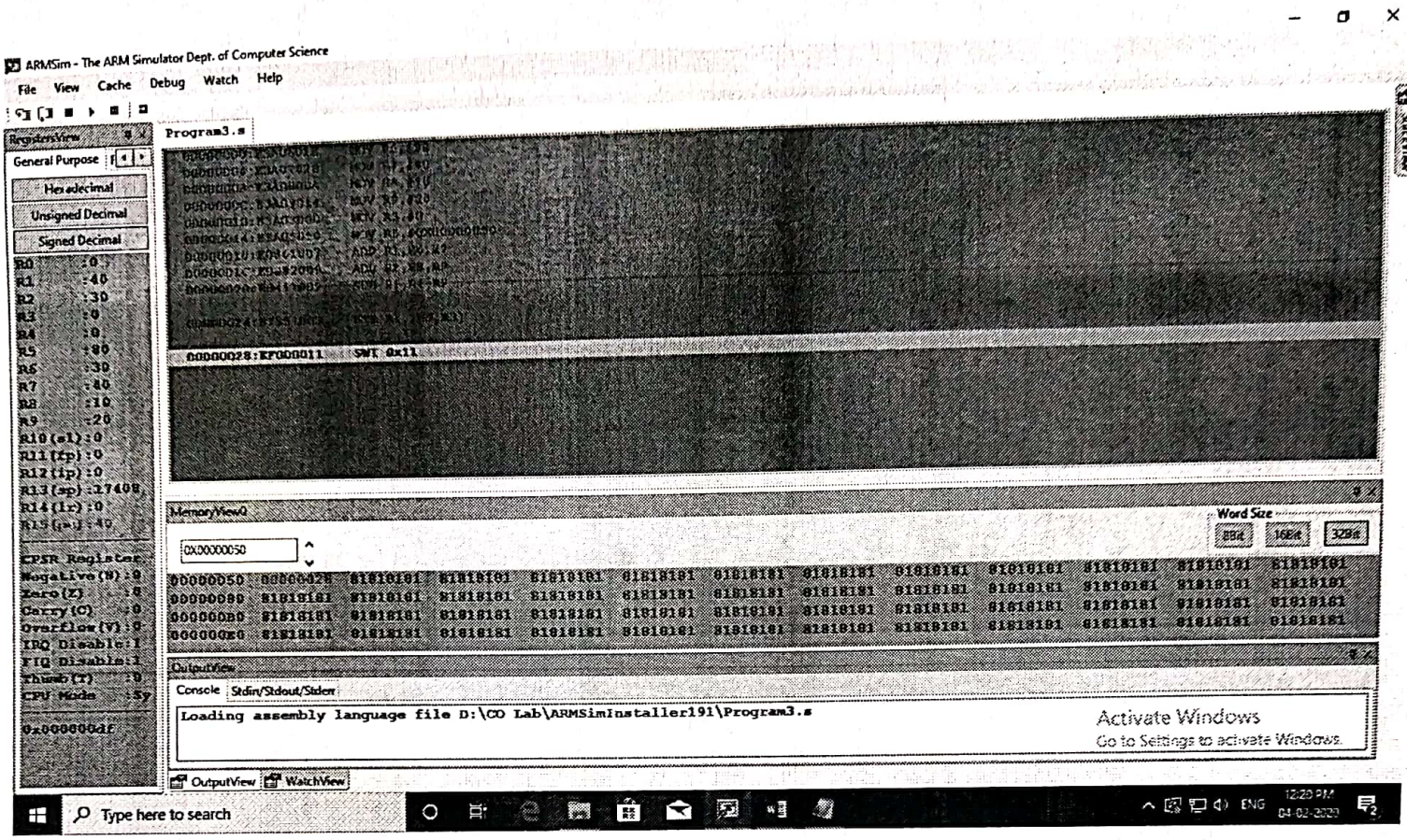


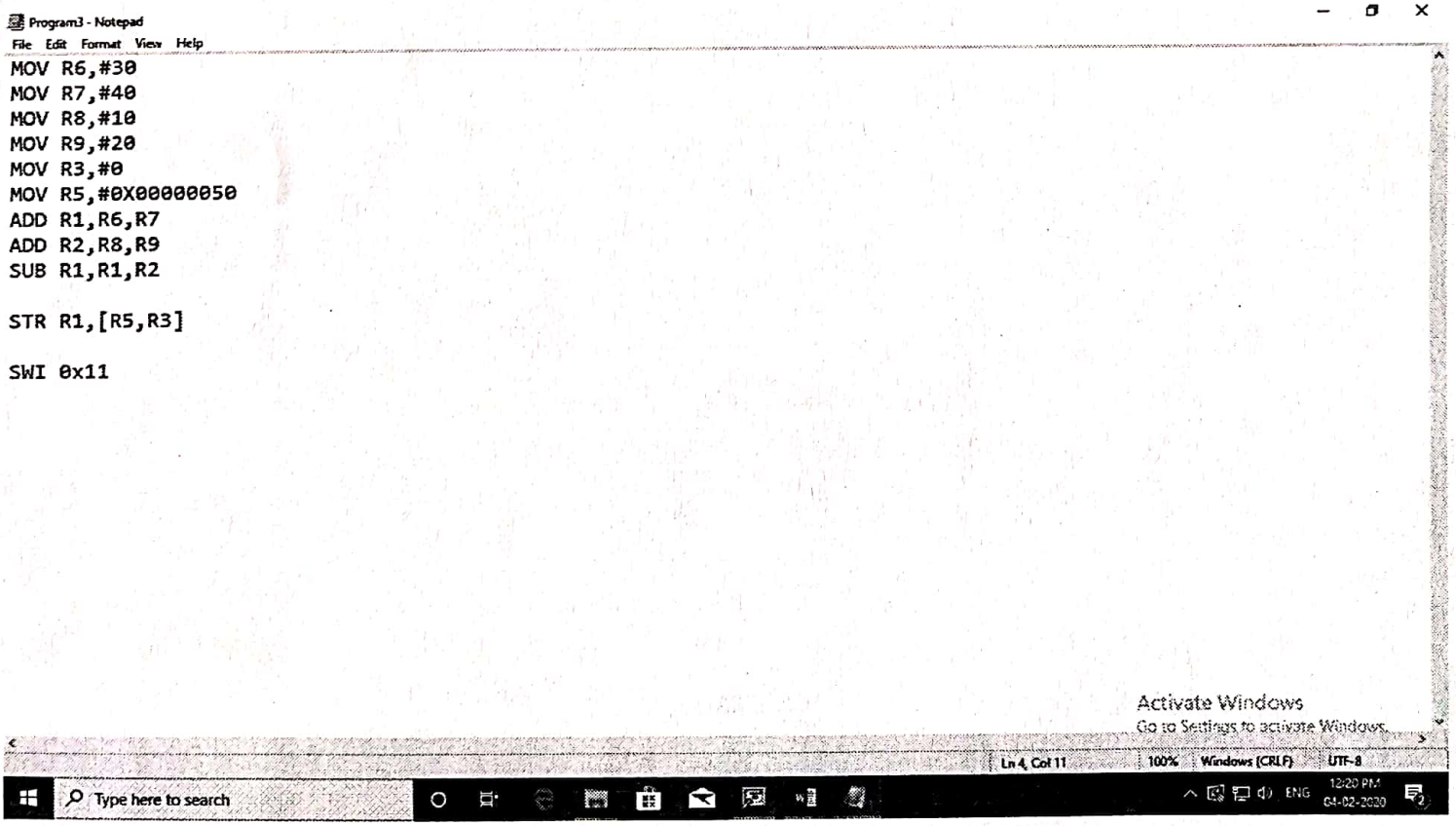


Program 3









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: MOULIDHAR .B	**Marks:** /10	**Date:**
USN: 1MS18CS075	**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.

Data Sheet

RAMAIAH
Institute of Technology

MARKS :

Activity - IV

Name :	MOULIDHAR. B	Branch:	CSE
USN/Roll No. :	1MS18CS075	Sem/Sec:	IV / B
Subject :	COA Lab	Subject Code:	

(1)

```
MOV  R0, #0
MOV  R1, #1
MOV  R3, #10
MOV  R4, #0
MOV  R5, #0X00001000
MOV  R6, #0
ADD  R4, R0, R1

LOOP:
     STR  R4, [RC, #4]
     ADD  R5, R6, #4
     MOV  R0, R1
     MOV  R1, R4
     ADD  R4, R0, R1
     ADD  R6, R6, #1
     CMP  R3, R6
     BGT  LOOP

SWI  0X11
```

(2)

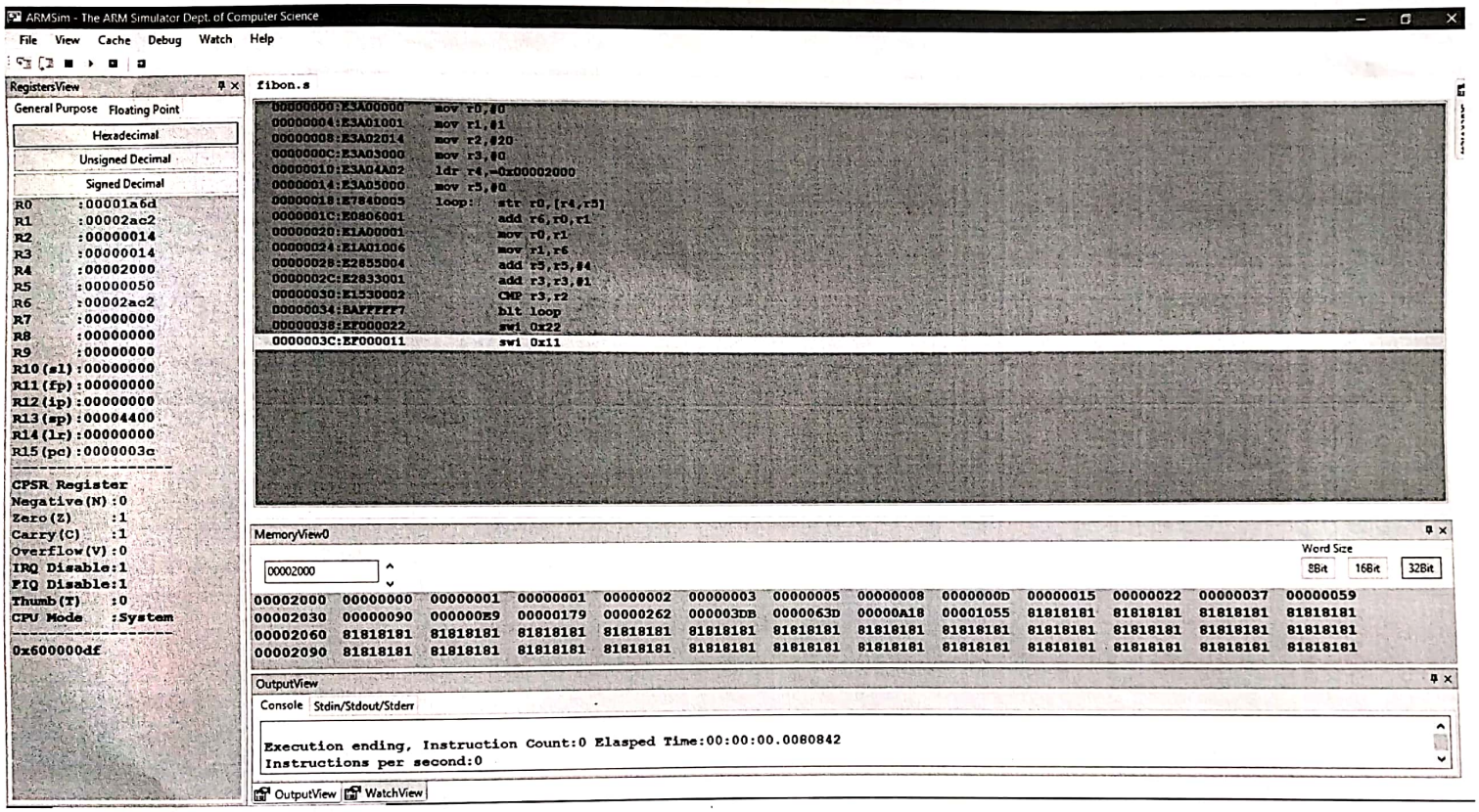
```
        MOV   R8, #4
        MOV   R4, #'n'
        MOV   R1, #16
        LDR   R0, #=0x00002000
        MOV   R5, #4
Loop:   LDR   R2, [R0, R5]
        SUB   R8, R8, #1
        ADD   R5, R5, #4
        CMP   R1, R2
        BEQ   pointy
        CMP   R8, #0
        BEQ   pointn
        BNE   Loop

pointn: STR   R4, [R0]
        LDR R0,[R0] LDR
        SWI   0x00
        B   END

pointy: STR.  R3, [R0]
        LDR   R0, (R0)
        SWI   0x00
```

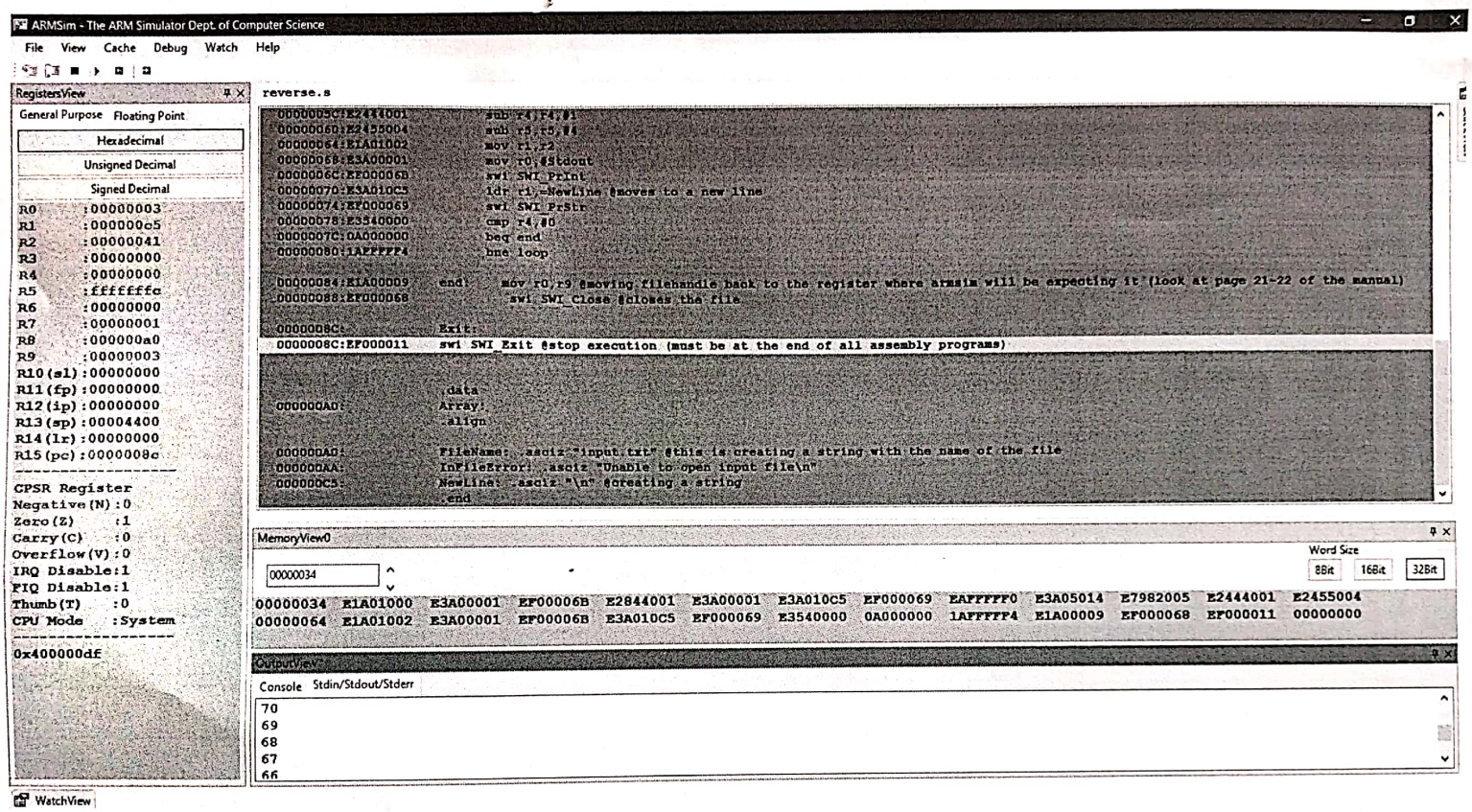
3)

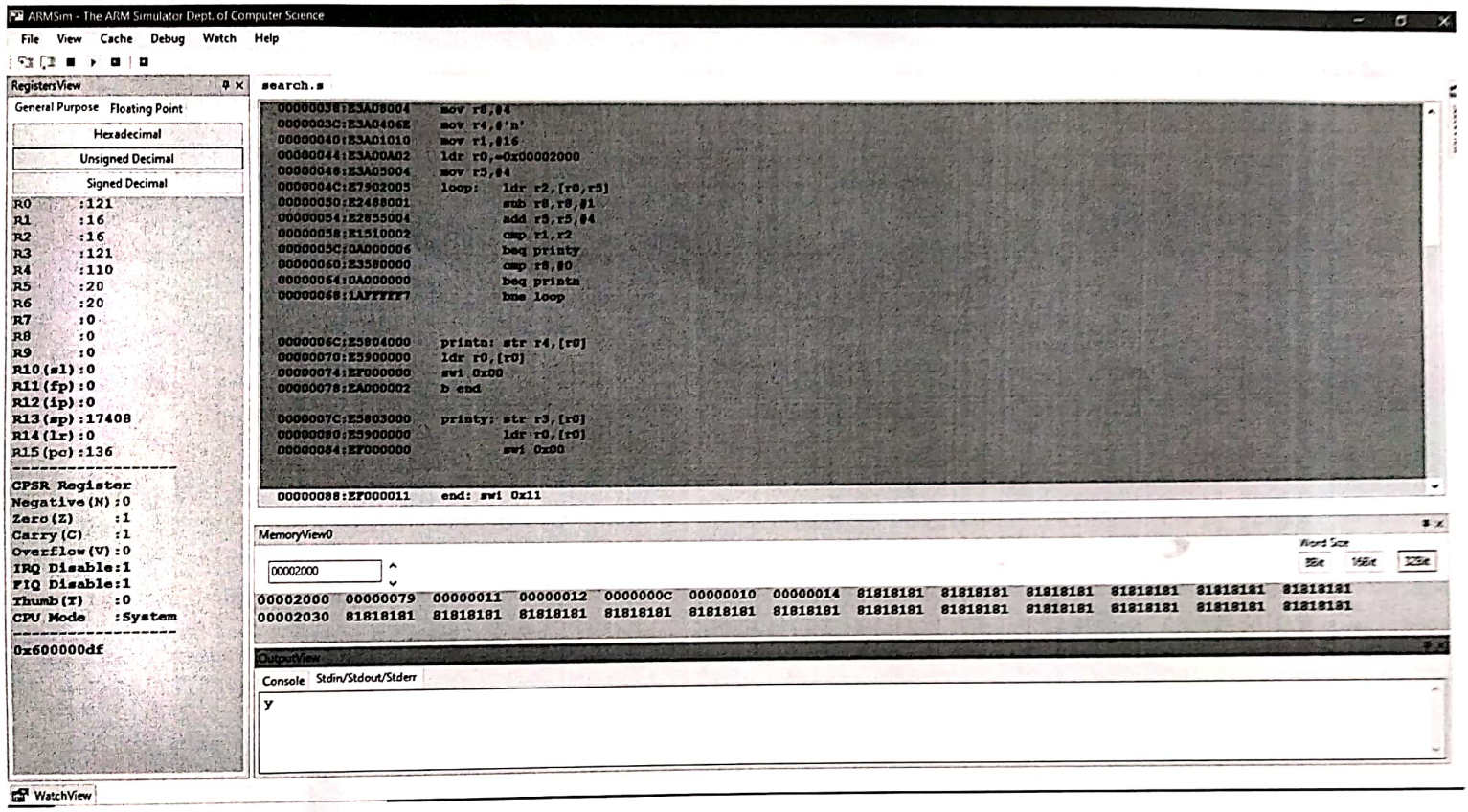
```
.equ SWI_open, 0x66
.equ SWI_close, 0x68
.equ SWI_print, 0x6b
.equ SWI_RdInt, 0x6c
.equ Stdout, 1
.equ SWI_PrStr, 0x69
.equ SWI_Exit, 0x11
global _start
```

```
.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   r1, =Array
    bcs   afterloop
    str   r0, [r1, r5]
    add   r5, r5, #4
    mov   r1, r0
    mov   r0, #stdout
    str   SWI-print
    add   r4, r4, #1
    afterloop:
```

```
        mov  r5, #80
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  -dose
Exit:
        SWI  Exit.
```







Computer Organization and Architecture

MOULIDHAR B
1MS18CS075

Activity V

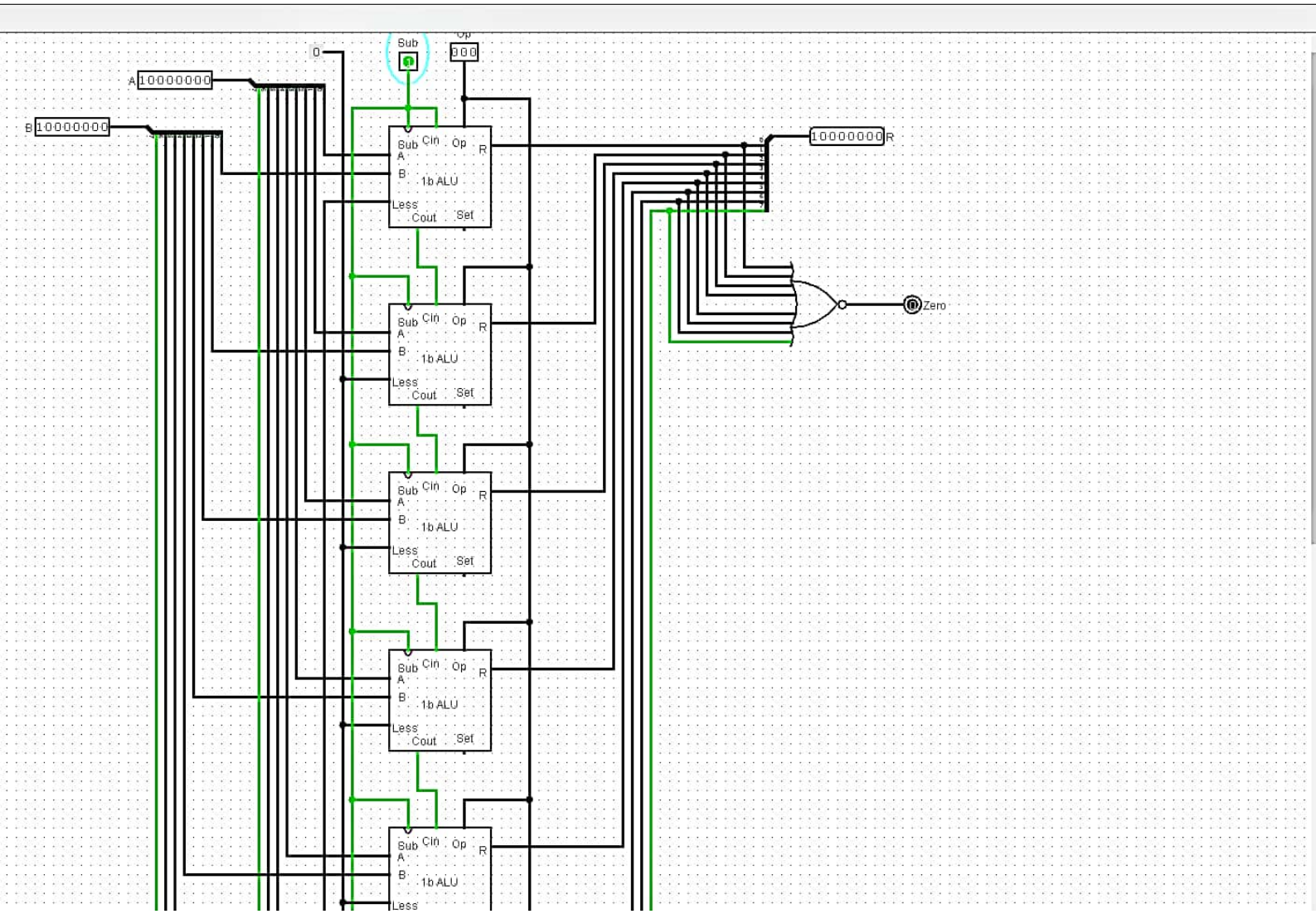
Objective: To Simulate the working of Arithmetic and Logical unit using simulator.

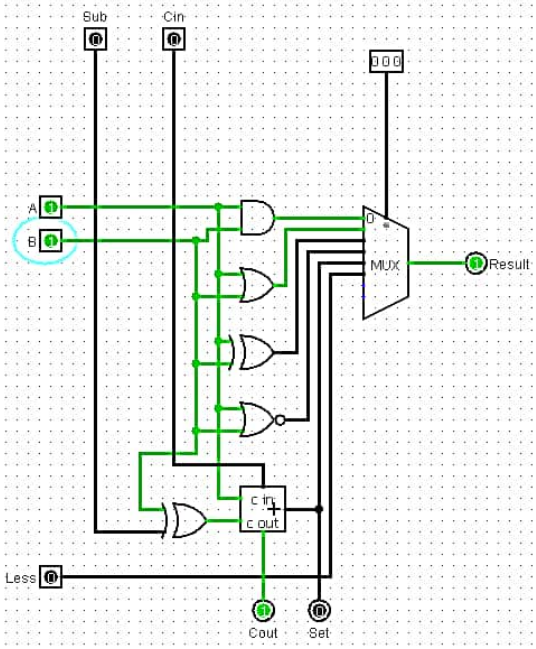
~~The steps in design~~

Simulator Description: Logism is an educational tool for designing and simulating digital logic circuits. with its simple toolbar interface and simulation of circuits as you build them, its 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, Logism can be used to design and simulate entire cpu's for educational purposes.

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 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 ex-or gate connect its o/p to Cout. The first i/p must to connected B and the second to another i/p pin Sub.
10. Add another i/p and name it less. Connect it to the mux.
11. Add an output pin and name it Set, Connect it to the o/p of adder unit.





Computer organization and Architecture

Activity VI

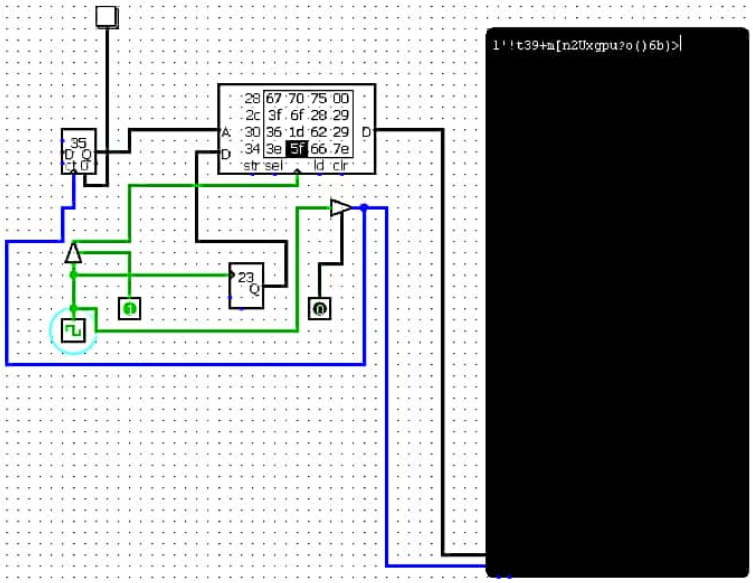
MOULIDHARAN. B
1MS18CS075

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 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, Logism can be used to design and simulate entire CPUs 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 controlled 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 TTY unit with 30 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.



Computer organisation & Architecture

Activity VII

MOULIDHAR-B
1MS18CS075

Objective :- To learn and analyse the performance of the cpu by overlapping of instructions using CPU0-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 & 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.

with diagram demonstrate the execution of the following instruction using pipelining technique.

Time (in clock cycles) →

Program execution order. (in instruction)

- lw $10, 20($1)
- sub $11, $2, $3
- add $12, $3, $4
- lw $13, 24($1)
- add $14, $5, $6