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: Joel John Thomas	**Marks:** /10	**Date:**
USN: 1MS18CS053	**Signature of the Faculty:**	

Activity I: Assembling and disassembling of a computer

Objective: To demonstrate the functional units of a system.

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

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

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

Answer the following questions.

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

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	Joel John Thomas	Branch:	CSE
USN/Roll No. :	1MS18CS053	Sem/Sec:	IV / B
Subject :	CO lab	Subject Code:	

Q1. Write down the detailed procedure to assemble a system.

Ans. step-1 : Remove side panels on case. After removing the case from the box. The panels are removed from this case with thumb screws.

step-2 : Insert Motherboard - Before acctting the board in, the I/O panel faceplate in the back of the case. Once the bard is resting in the case, line up the first hole.

step-3 : check clearances - Being that the comp. includes high performance components, some of them are large enough that clearance can become an issue.

step-4 : Front Panel connections - Attach the connections for the buttons, light, USB ports and audio connections.

step-5 : Install Powr Supply - cables that are needed are plugged into the unit.

step-6 - lower Motherboard - The longest motherboard power cable is to be connected.

step-7 : Installing optical Drive - The optical drive for the comp. is a DVD/CD read/write combo

step-8 : Installing the hard drives. Comp. uses 4 drives, 2 in raid and the rest for a main drive and miscellone storage.

step-9 : Connect Cables - Connect the cables the hard drives & optical drives.

step-10 : Installing RAM - The slots are known as are the RAM sticks.

step-11 : Install graphic cards and expandcards - The network card and audio card for the comp. are connected into the slots below the graphic cards.

step-12 : Cable Managment - Organising and hiding cables for high airflow and security/safety.

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

Ans: Troubleshooting is a form of a problem solving often applied to repair failed products or process on a machine or a system. It is a logical, systematic search for the source of a problem in order to solve it and make the product or process operational again. Troubleshooting is needed to identify the symptoms. Determing the most likely cause is a process of elimination. Eliminating potential causes of a problem. Finally, troubleshooting requires confirmation that the soln. restores 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 restore the problem & implement the soln.

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

step-6: Document findings, actions and outcomes.

Q3. list out the procedures to install extra memory card to a system.

Ans3:
* Disconnect the power cable from the system & if needed, unplug other back-panel cables so that you can safely turn your system on to its side.
* 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 cooler. If there's already RAM in your system, eject it by pressing ly firmly on the tabs on the motherboard at either end of the slots.
* To install new RAM, line up notches in the bottom of the sticks with gaps in the slot on the motherboard. Make sure the wings are either ends of slot pushed back, so that they are tilted away from the RAM. Then the wing will clamp in and hold the memory securely.
* once the sticks have clicked into it, it confirms that the wing clips are locked in to hold the stick firmly in their slots and then close the PC back up. Plug all the cables back in and try to boot the system.

Q4 With a neat diagram explain different cases used to connect functional units in a system.

Ans (i) 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.

(ii) DVI cable: Connect one end to comp. monitor other end to DVI port on comp.

(iii) HDMI cable: Connect one end to comp. monitor, television. Connect other end to HDMI port on computer.

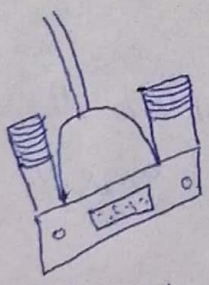
(iv) PS/2 cable: Connect one end to PS/2 keyboard, PS/2 mouse. Connect other end to PS/2 ports on comp. Purple PS/2 port: keyboard. Green PS/2 port: mouse.

(v) ~~Ethernet~~ USB Cable: Connect one end to USB device. Connect other end to USB ports on comp.

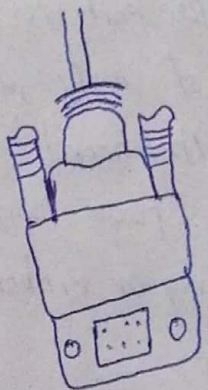
(vi) Ethernet Cable: Connect one end to router, network switch. Connect other end to ethernet port on comp.

(vii) Computer Power Cord (kettle plug): Connect one end to AC power socket. Connect other end to power supply unit, comp. monitor.

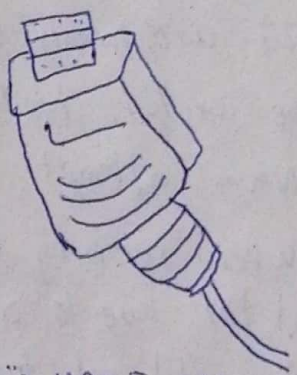
Diagrams



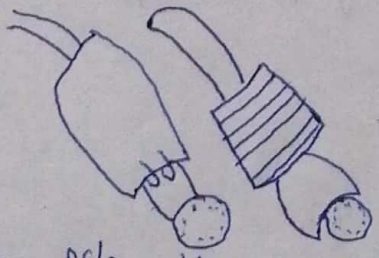
(i) VGA cable



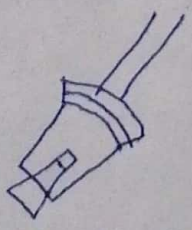
(ii) DVI cable



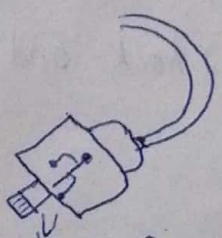
(iii) HDMI cable



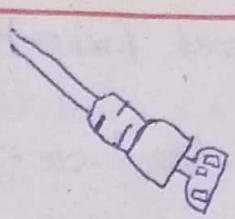
(iv) PS/2 cable



(v) Ethernet cable



(vi) USB cable



(vii) Computer Power Cord (Kettle Plug)

Q5. Discuss the safety precautions one should take while removing components of a system.

Ans. A few warnings and reminders before you start disassembling your comp. tower to keep both your unit and yourself safe.

* Fully shutdown and unplug the comp. before you make any attempts to dissassemble the tower.
* Take off any metal objects on your arms or fingers. Even if your unit is unplugged, they may remain some electric charge.
* Make sure your hands are completely dry.
* Work in a cool areas to avoid prespiration.
* Before touching any part within the tower put your hands against another metal surface to remove static charge.
* Prepare a place to keep any screws you may remove.
* Handle all parts with care. Place each parts on a stable surface.
* If components do not come of easily, dont remove with force.
* Be careful handling the motherboard.
* Never attempt to remove the power source, a box attached to the side.
* When removing any cables, wires or ribbons make sure to grasp the wire at the base to avoid breakage.
* Be careful not to drop any small parts into unreachable areas.
* Take note that the 3 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: Joel John Thomas	Marks: /10	Date:
USN: 1MS18CS053	Signature of the Faculty:	

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

Objective: To simulate inter communication between CPU and memory.

Simulator Description: MarieSim is a computer architecture simulator based on the MARIE architecture. It provides users with interactive tools and simulations to help them deepen their understanding of the operation of a simple computer. One can observe how assembly language statements affect the registers and memory of a computer system.

Activity to be performed by students:

1. Draw the interconnection between memory and a processor.

2. 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

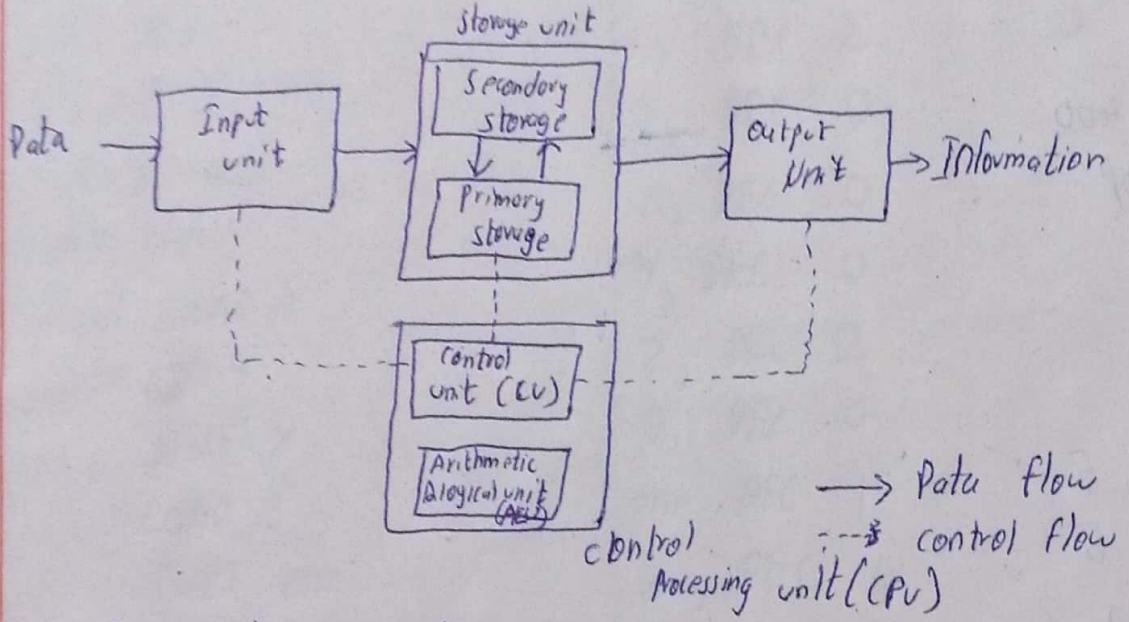
Data Sheet

MARKS :

Name :	Joel John Thomas	Branch:	CSE
USN/Roll No. :	1MS18CS053	Sem/Sec:	IV / B
Subject :	CO lab	Subject Code:	

Q1 Draw the inter connection b/w memory and processor

Ans.



Q2. list out the steps required to execute an instruction.

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

(i) Fetch instruction (ii) Decode information (iii) Perform ALU operation

(iv) Access memory (v) Update register file (vi) Update the Program counter (PC)

Q3: 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
 LOA B
SUB one
STORE B
SKIPCOND 400
JUMP loop
LOAD F
OUTPUT
HAULT
G, DEC 5
H, DEC 5
Y, DEC 1
I, DEC 6
A, DEC 0
B, DEC 0
F, DEC 0
ONE, DEC 1
```

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

Ans.

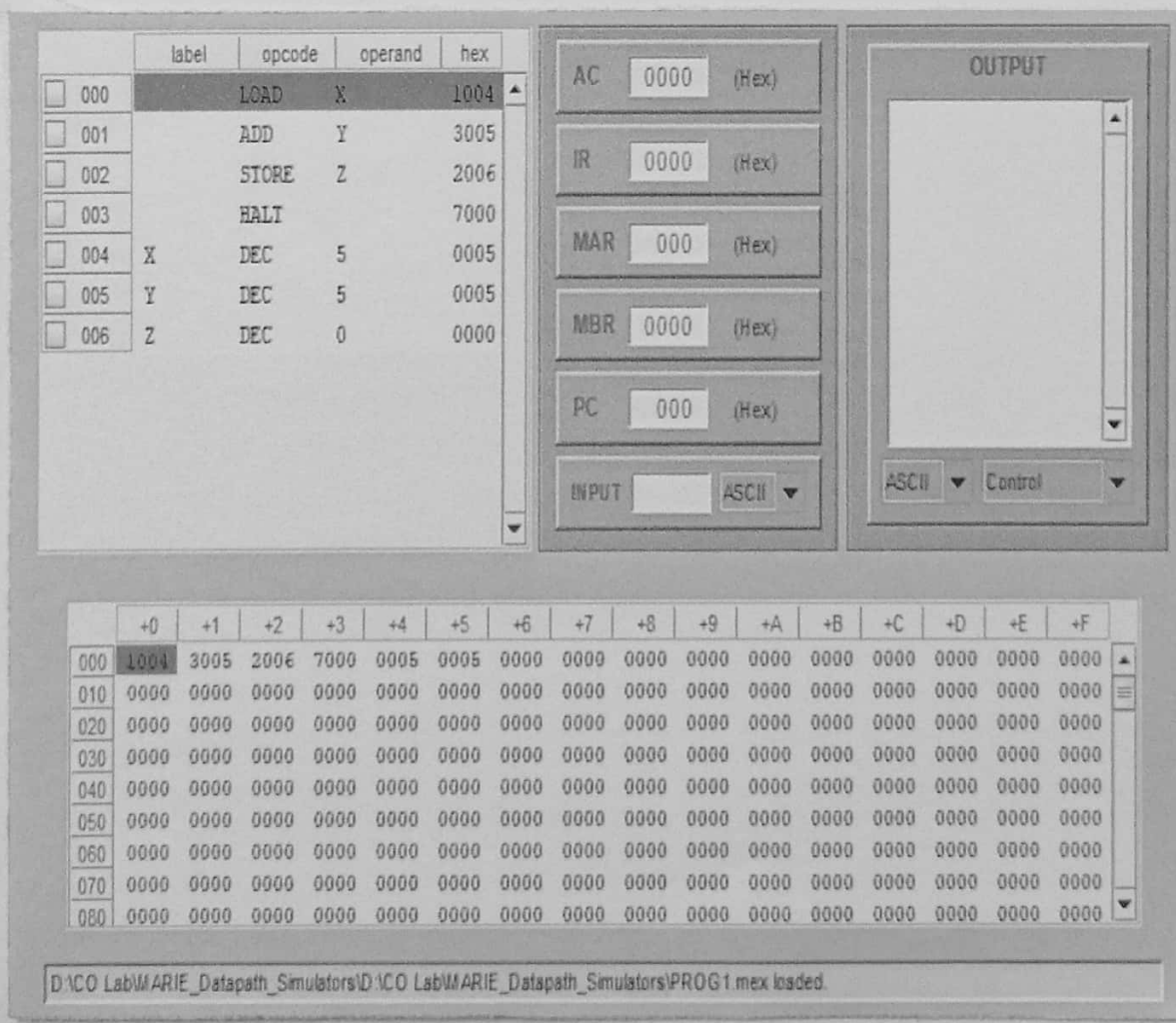
```
LOAD B
STORE O
first LOAD B
ADD X
STORE X
LOAD O
SUBT one
STORE O
SKIPCOND 400
JUMP first
Second LOAD A
       ADD Y
       STORE Y
       LOAD C
       SUBT one
       STORE C
       SKIPCOND 400
       JUMP Second
third  LOAD four
       ADD Z
       STORE Z
       LOAD Y
        SUBT one
        STORE Y
        SKIPCOND 400
        JUMP third
LOAD D
ADD X
SUBT Z
OUTPUT
HALT
A, DEC 6
B, DEC 3
C, DEC 2
O, DEC 0
X, DEC 0
Y, DEC 0
Z, DEC 0
D, DEC 0
one, DEC 1
four, DEC 4
```

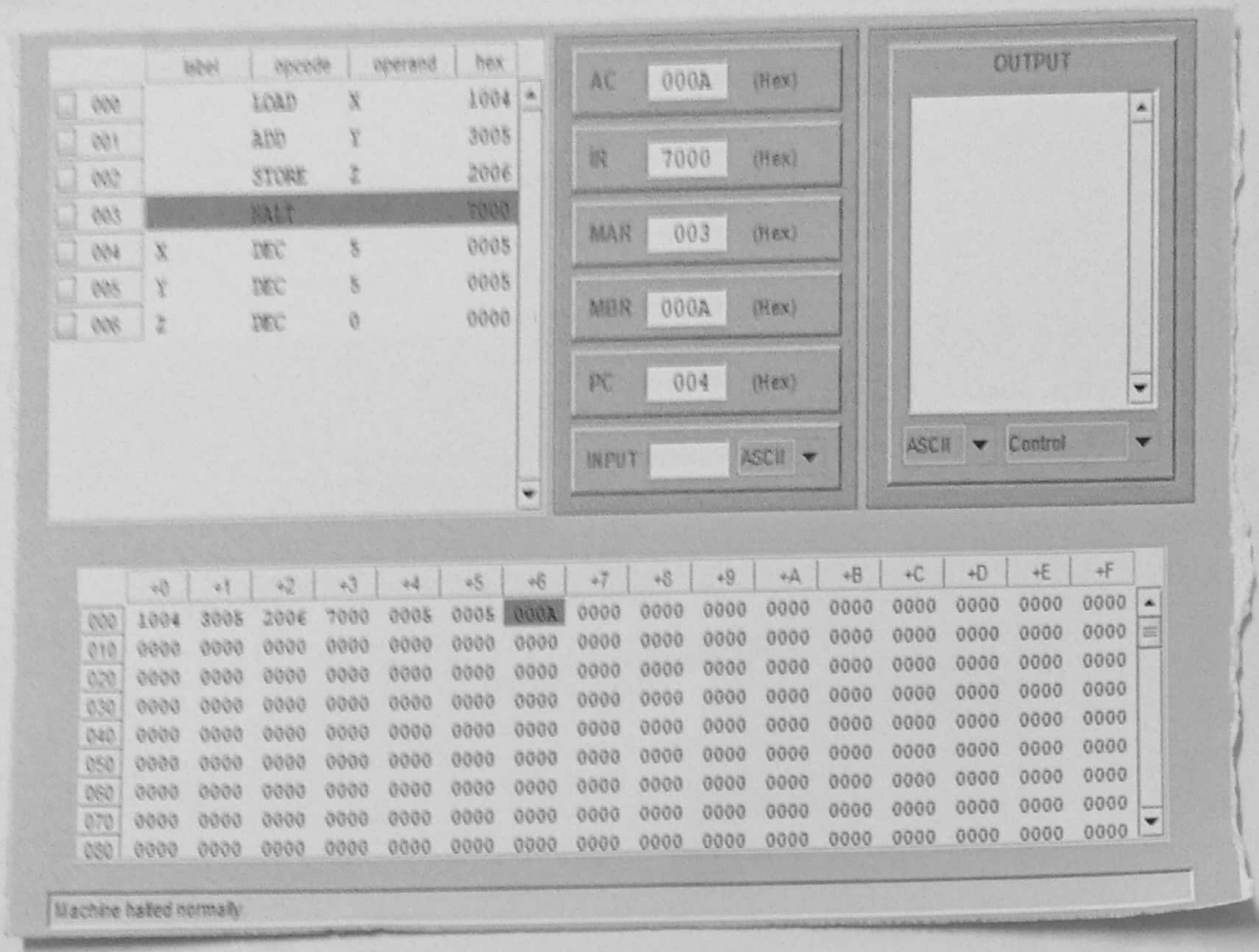
Q4.
Ans4. Describe the factors affecting the performance of a processor.

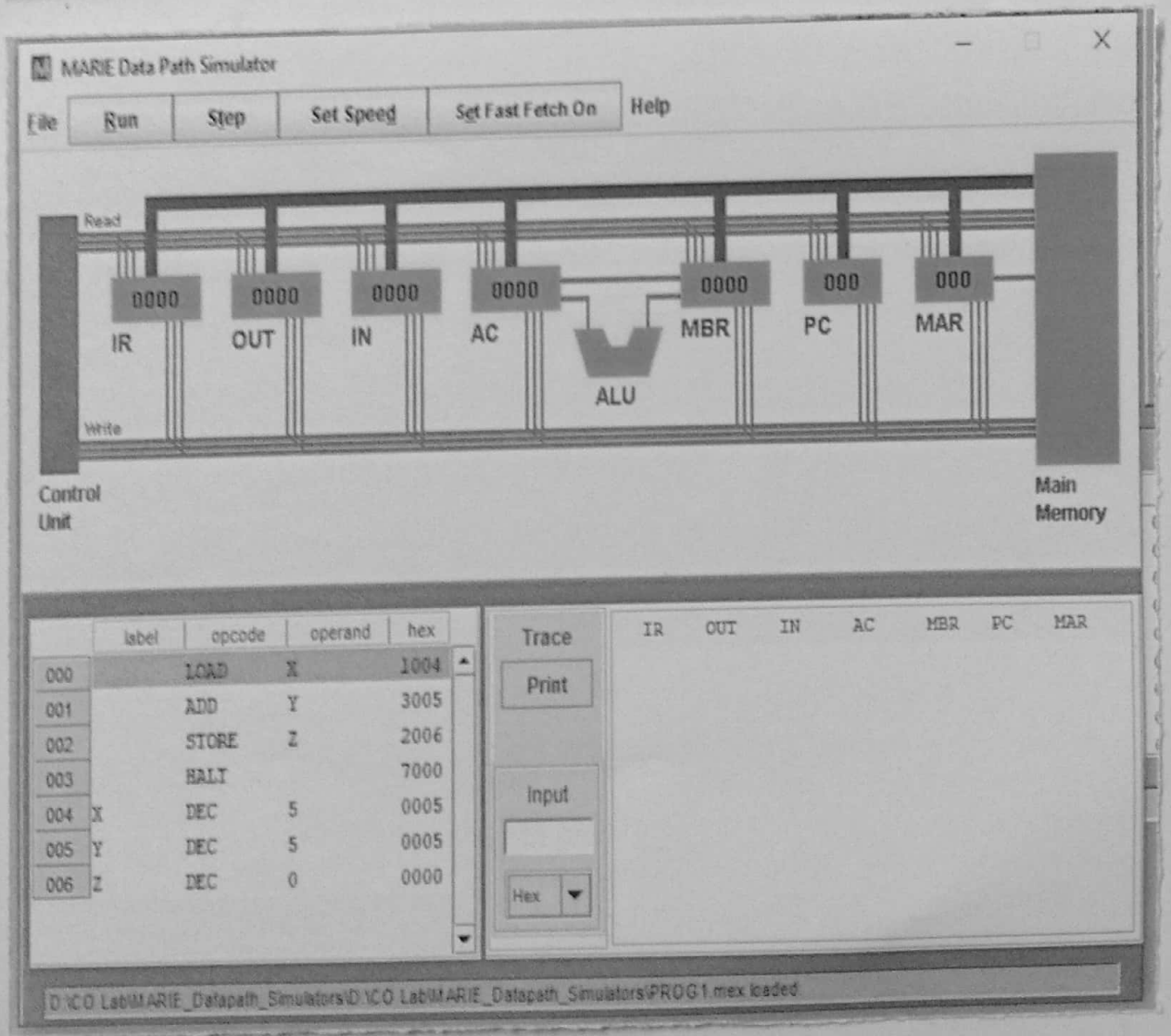
Ans 4. There are ~~fact~~ factors:

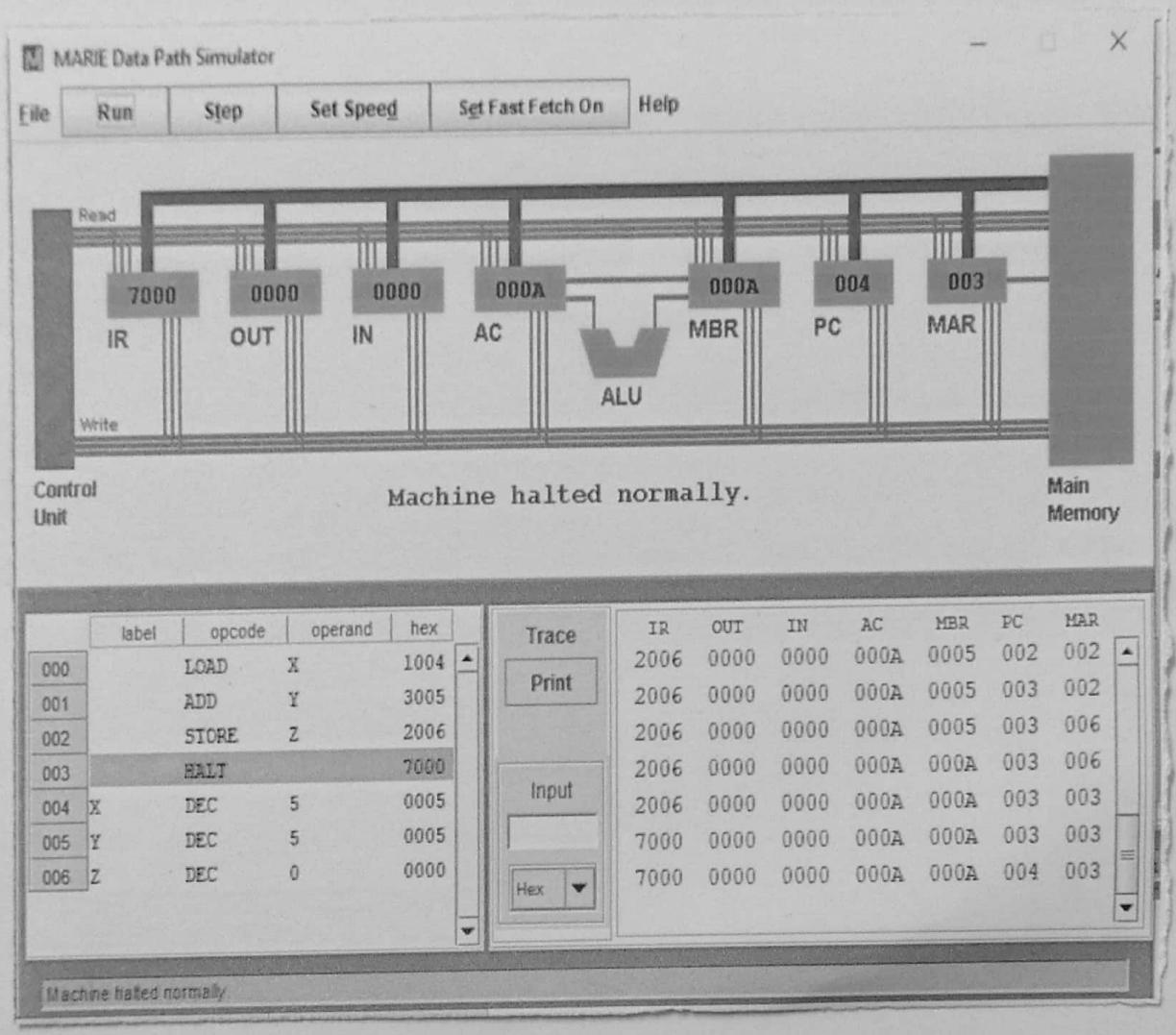
* Multiple cores: We have dual, quad and even oct cores processors with its own fetch and execute cycles.
* Clock speed: The processor requires a clock pulse in order to operate correctly. 1 clock cycle = 1Hz. A PC bus clock speed in the GHz region.
* Cache Memory: It is a small amount of high performance RAM that is built into the processor. This RAM stores the data which has to be repeatedly used by the processor and does not need request from memory.
* word length: The no. of bits the CPU can process simultaneously. For ex: a 32-bit processor is faster than a 16-bit processor because of its wider word length.
* Address Bus Width: It is the width of the address bus and determines the maximum amount of addressable locations. For ex, an address bus of 8-bit means that you can have 256 addresses (0 to 255)
* Data Bus width: It is the no. of bits that can be transfered 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
          ------------------------------------------------
```









LAB-3

Name: Joel John Thomas
USN: 1MS18CS093
Sem: IV Sec: B

Objective: To simulate ARM Instruction set using ARM sim Simulator

Simulator Used: ARM Sim 1.91 is a desktop application asumming in a circumstances enviornment. It allows user to simulate the execution of ARM assembly language programs on a system based on the ARM7TDMI processor.

ARM enable the user both to debug ARM assembly program and to monitor the state of the system while a program execute.

Activity to be performed by students.

* Write an ARM program to perform basis arithmetic operation.
* Write an ARM program to demonstrate the ucoverlay of load and Store instructions.
* write on ARM program to evaluate expression f = (g+h) - (i+j)
* write on ARM program to find the Sem of all elements of an array
* write on ARM program to find the factorial of a number.

Answers

Ans1.
```
mov R5, #10
mov R6, #20
ADD R7, R5, R6
SWI0X11
```

Ans2.
```
mov R1, #0X00000070
mov R3, #0
mov R4, #50
STR R4, [R1, R3]
LDR R6, [R1, R3]
SWI 0X11
```

Ans3

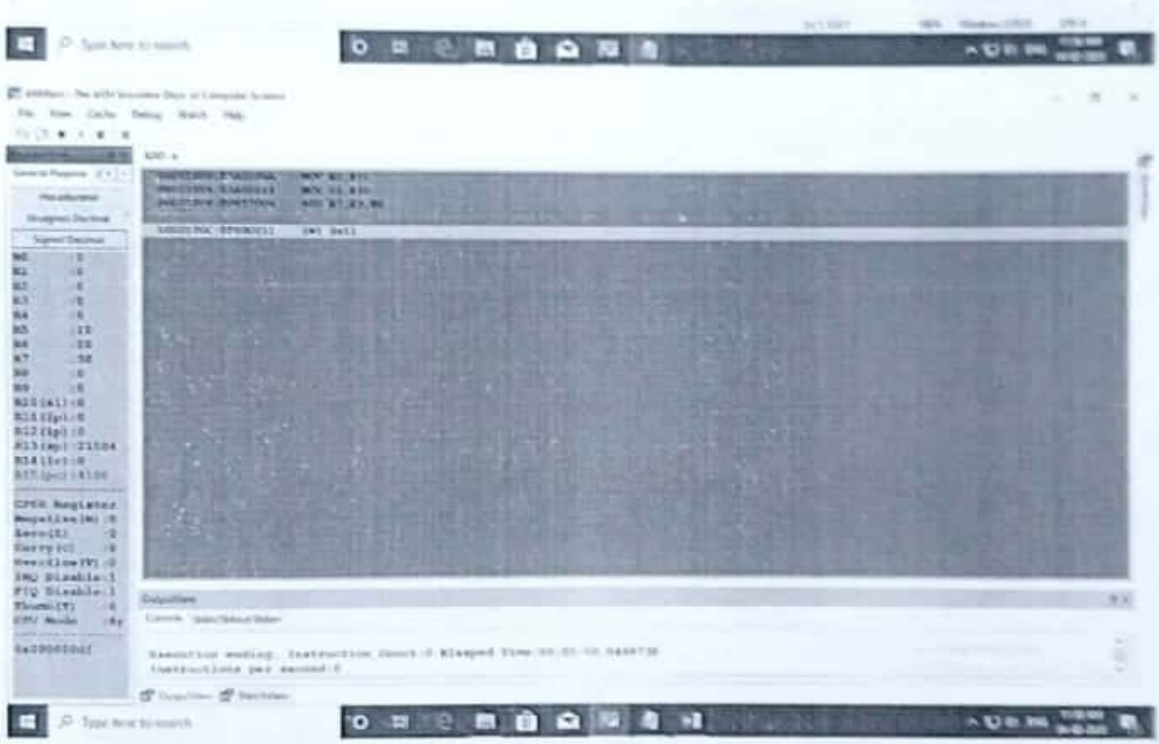
```
mov  R0, #30
mov  R7, #40
mov  R8, #10
mov  R9, #20
mov  R3, #0
mov  R5, #0x00000050
ADD  R1, R0, R7
ADD  R2, R8, R9
SUB  R1, R1, R2
STR  R1, [R5, R3]
SWI 0x11
```

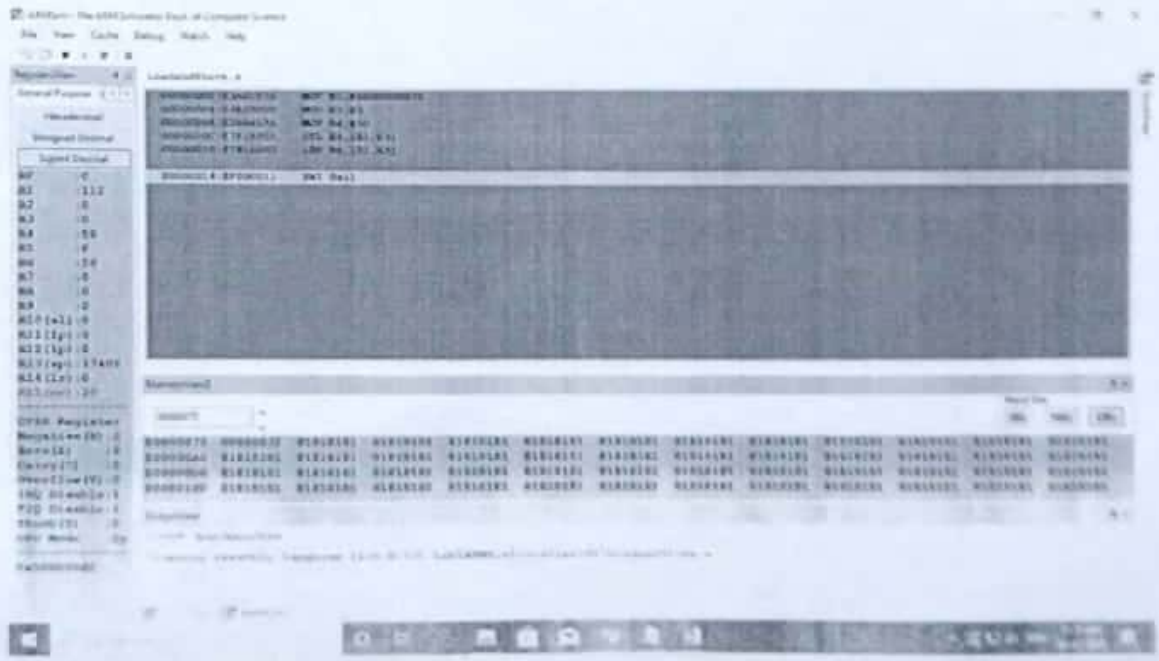
Ans4

```
mov  R0, #5
LDR  R1, =array
Loop LDR R2, [R1], #4
ADD  R3, R3, R2
SUB  R0, R0, #1
Cmp  R0, #0
BNE  loop
array DCD 0x00000001, 0x00000002, 0x00000003, 0x00000004,
          0x00000005
   SWI 0x11
```

Ans5

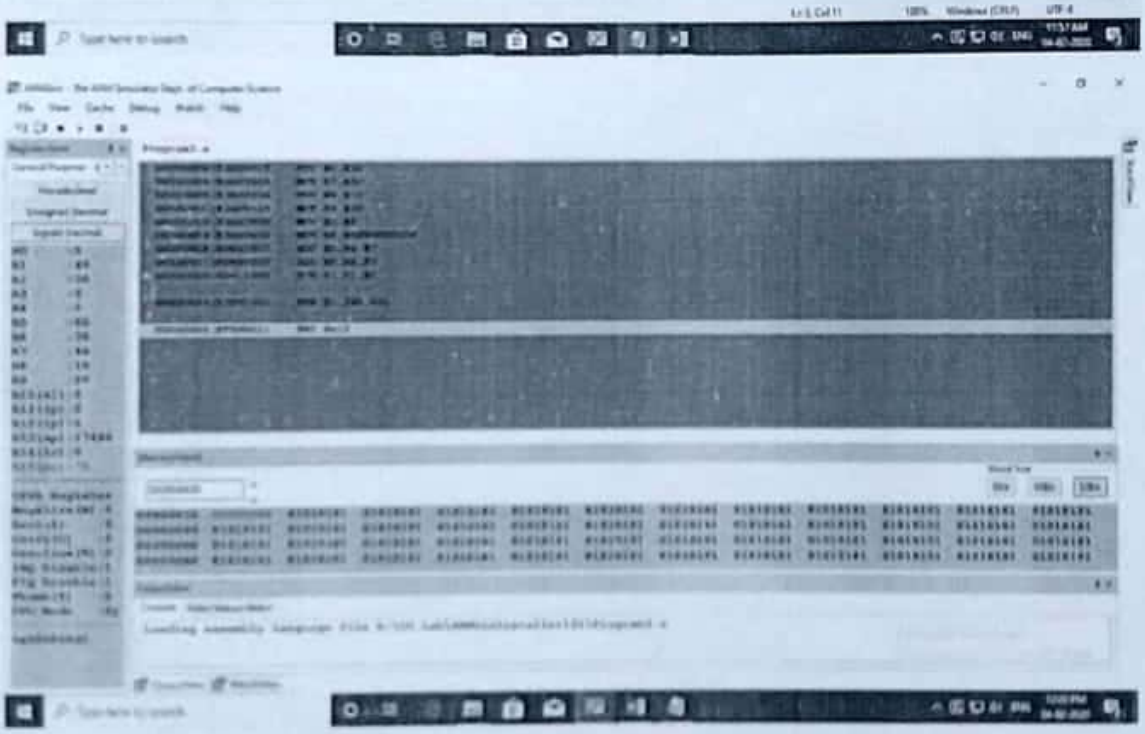
```
mov  R1, #5
mov  R0, #1
mov  R3, #1
loop: MUL  R3, R0, R3
      ADD  R0, R0, #1
      SUB R1, R1, #1
      Cmp R1, #1
      BGE loop
      SWI 0x11
```





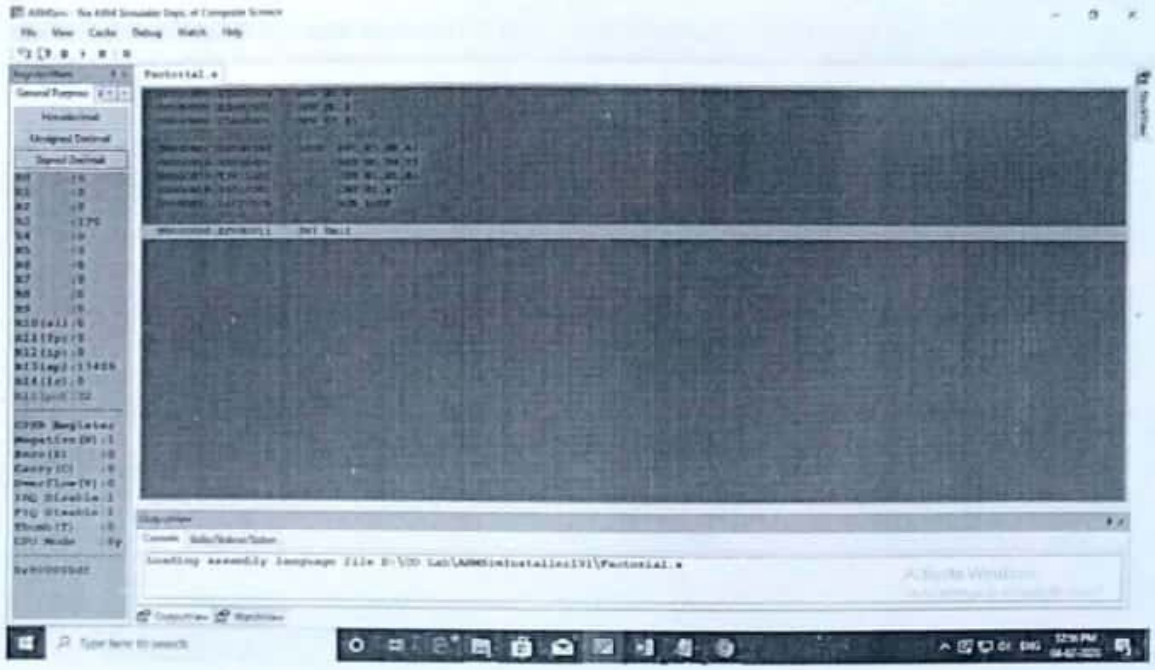
Scanned with CamScanner

```
MOV R1,#0X80000070
MOV R3,#0
MOV R4,#50
STR R4,[R1,R3]
LDR R5,[R1,R3]

SWI 0x11
```



```
MOV R6,#30
MOV R7,#40
MOV R8,#10
MOV R9,#20
MOV R3,#0
MOV R5,#0X80000050
ADD R1,R6,R7
ADD R2,R8,R9
SUB R1,R1,R2

STR R1,[R5,R3]

SWI 0x11
```


Scanned with CamScanner



```
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
```

Scanned with CamScanner

Lab - 4

Name: Joel John Thomas
USN: 1MS18CS053
Sem: IV Sec: B

Ans1. **Fibonacci Series**

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

Ans2. **Search an element in the array**

```
ldr r0, =0x00002000
mov r1, #14
mov r2, #17
mov r3, #18
mov r4, #12
mov r5, #16
mov r6, #20
str r1, [r0]
str r2, [r0, #4]
str r3, [r0, #8]
str r4, [r0, #12]
str r5, [r0, #16]
str r6, [r0, #20]
mov r3, #'y'
```

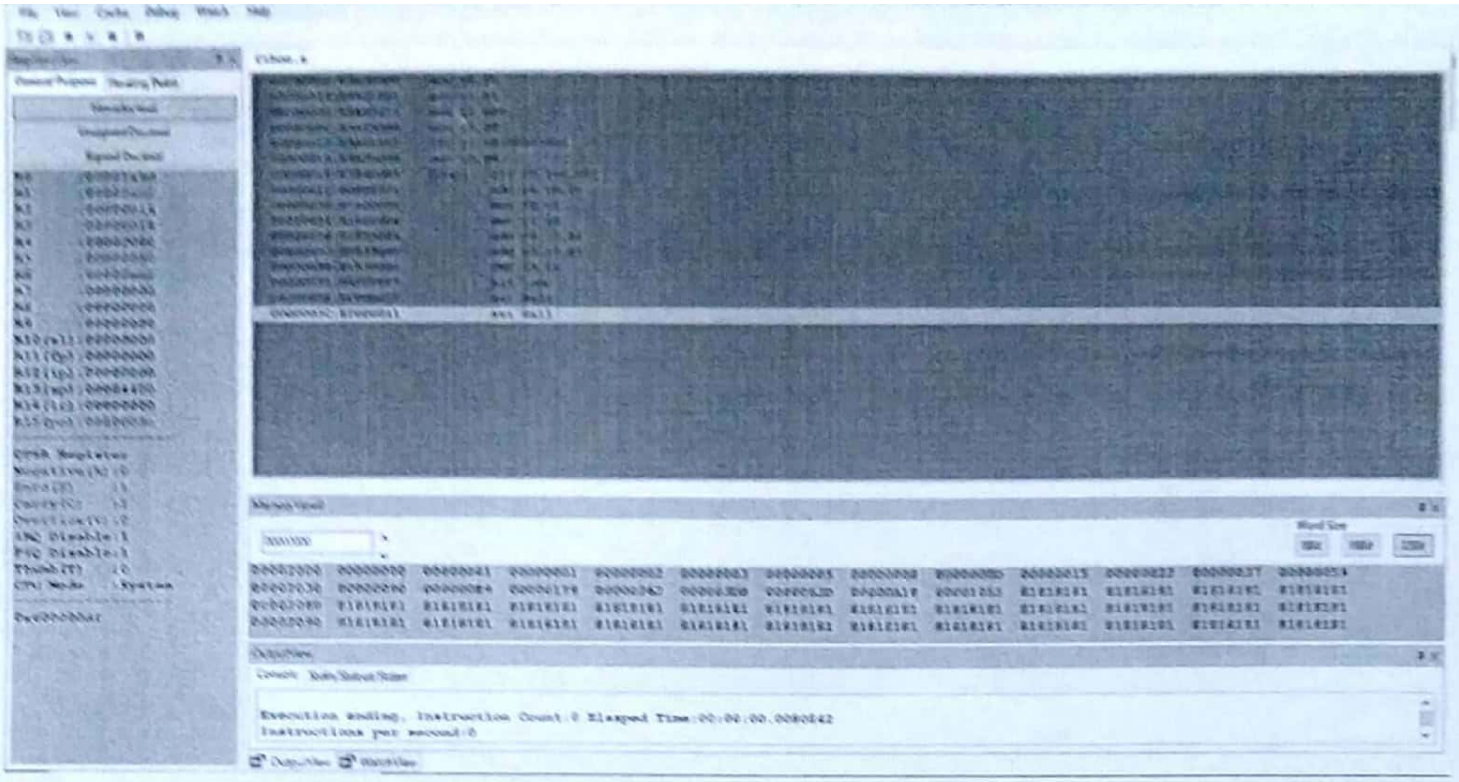
```
mov r8, #4
mov r4, #'n'
mov r1, #16
ldr r0, =0x0000 2000
mov r5, #4
loop: ldr r2, [r0, r5]
      sub r8, r8, #1
      add r5, r5, #4
      beq Printly
      cmp r8, #0
      beq printn
      bne loop
printn: str r4, [r0]
      ldr r0, [r0]
      swi 0x000
      bend
printy: str r3, [r0]
      ldr r0, [r0]
      swi 0x00
end   swi 0x11
```

Ans

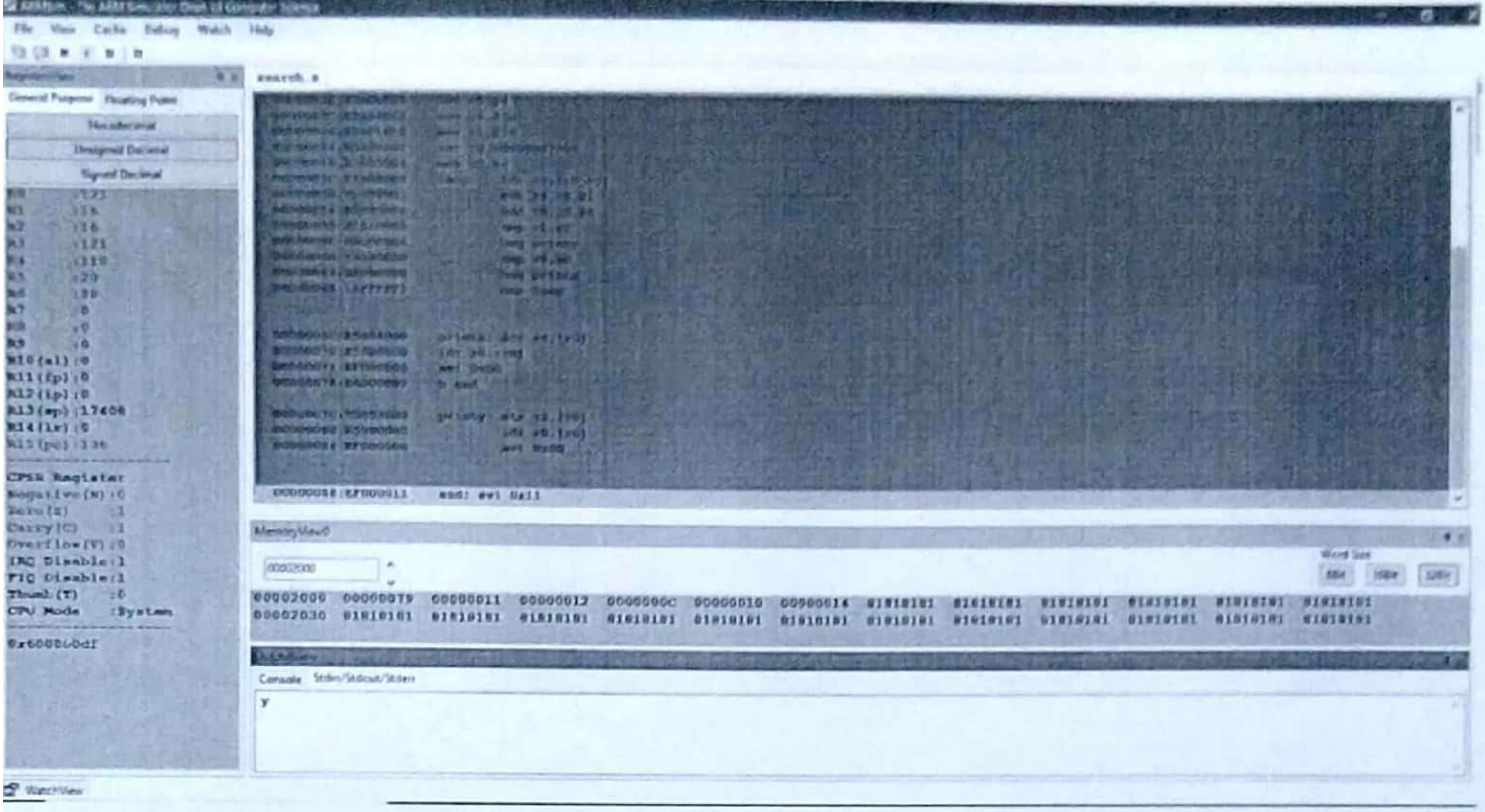
Reverse of an array

```
.eqn swi-open, 0x66
.eq  swi-close, 0x68
.eq  swi-Print, 0x6b
.eq  swi-Rd Int, 0x6c
.eq  student, 1
.eq  swi-Pr Str, 0x69
.eq  swi-Exit, 0x11
global _start
text
LDR r0, =Fibname
mov r1, #0
swi swi open
bcs Exit
mov r9, r0
mov r5, #0
```

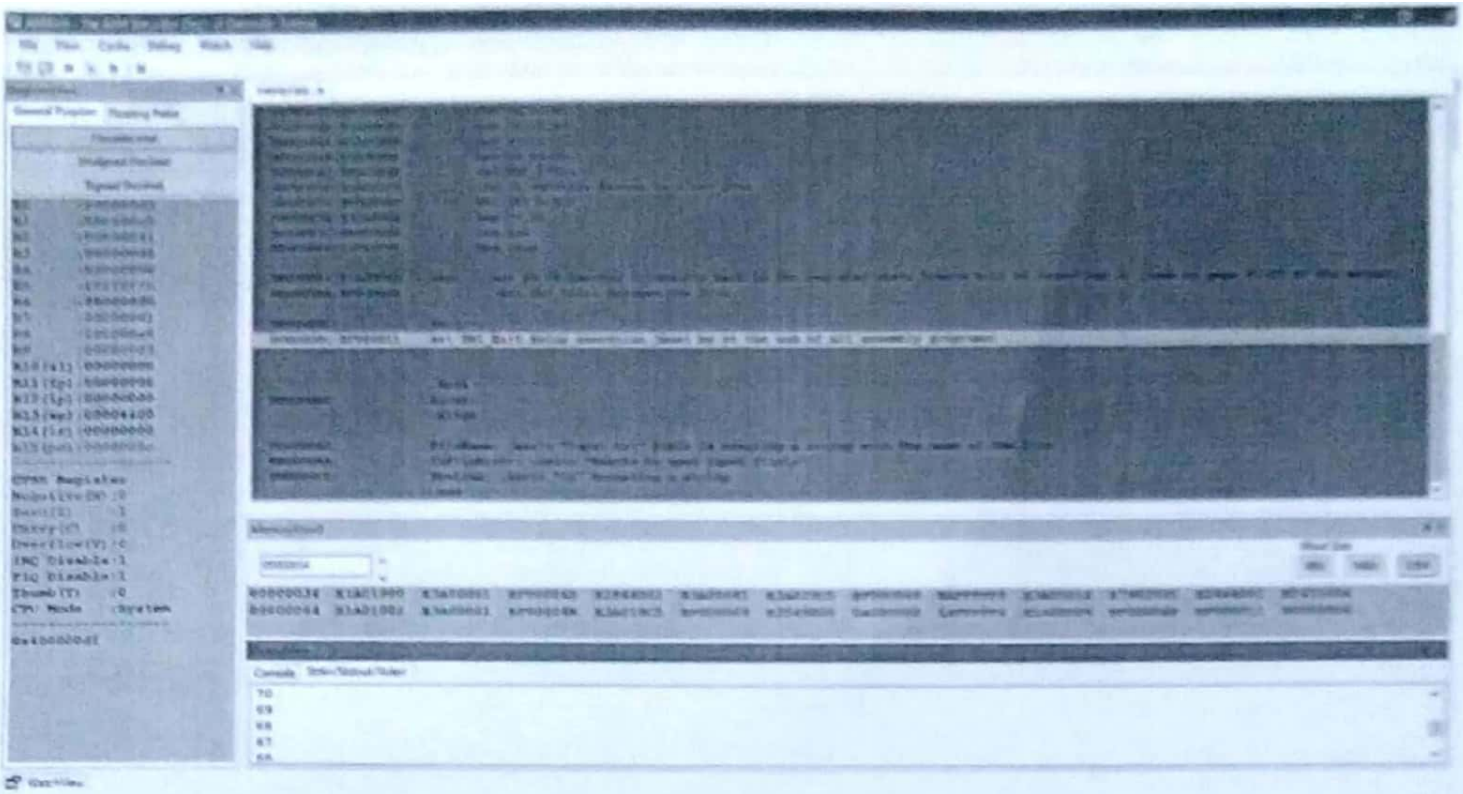
```
loop start:
 mov  r5, #stdout
 mov  r0, r9
 LDR  r8, = Array
 bcs  after loop
 str  r0, [r8, r5]
 add  r5, r5, #4
 mov  r1, r0
 mov  r0, # stdout
 str  SWI_print
 add  r4, r4, #1
after loop:
 mov  r5, #20
 loop: LDR  r2, [r8, r5]
   sub  r4, r4, #1
   sub  r5, r5, #4
    mov  r1, r2
   beq  end
   beq  loop

 end: mov  r0, r9
       SWI_close

Exit: SWI  SWI Exit
```



Scanned with CamScanner



Scanned with CamScanner



Scanned with CamScanner

Computer Organization Lab

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

Name: Joel John Thomas	**Date:**
USN: 1MS18CS053	**Signature of the Faculty:**

Objective: To simulate the

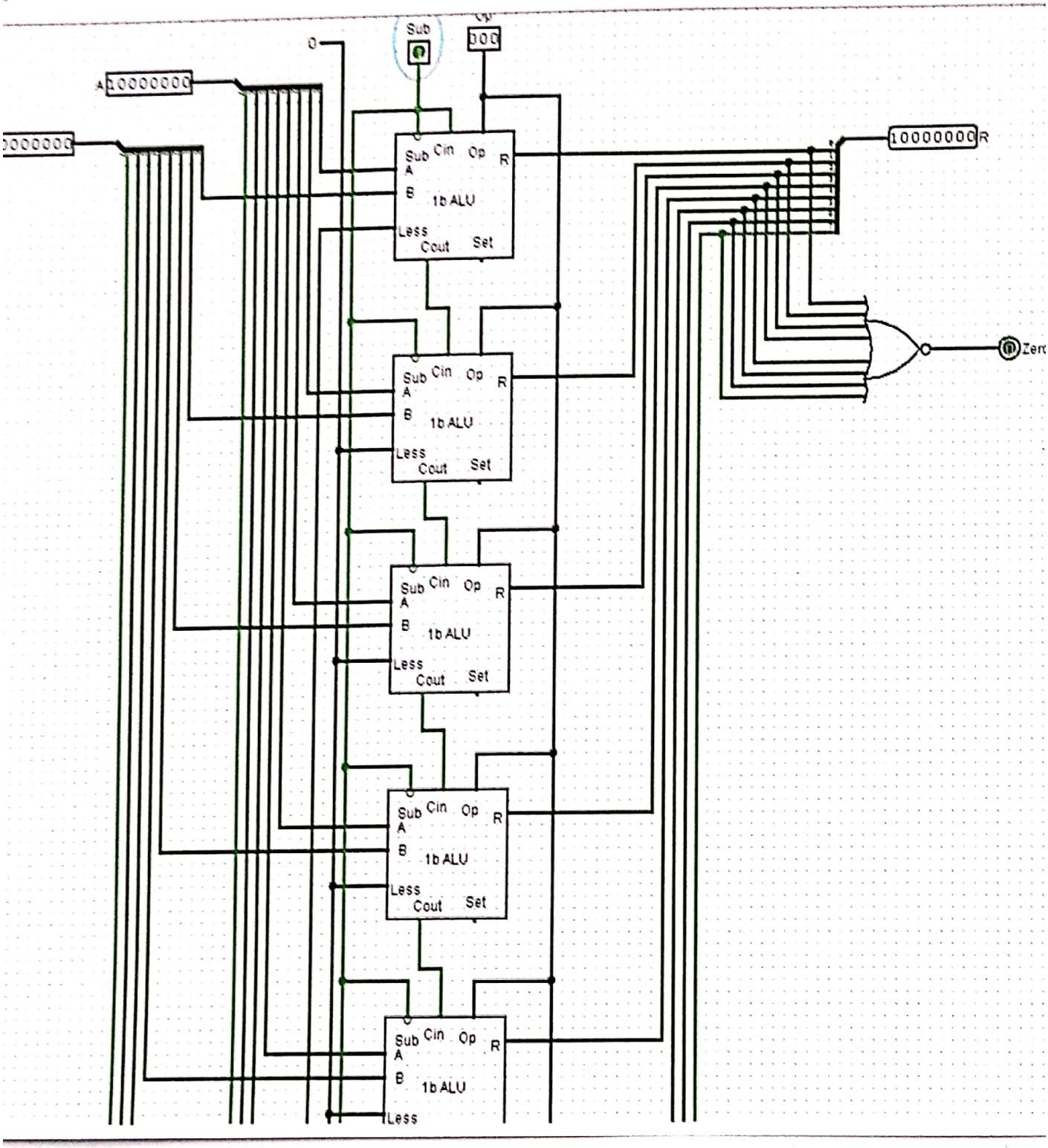
Simulator Description: Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to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

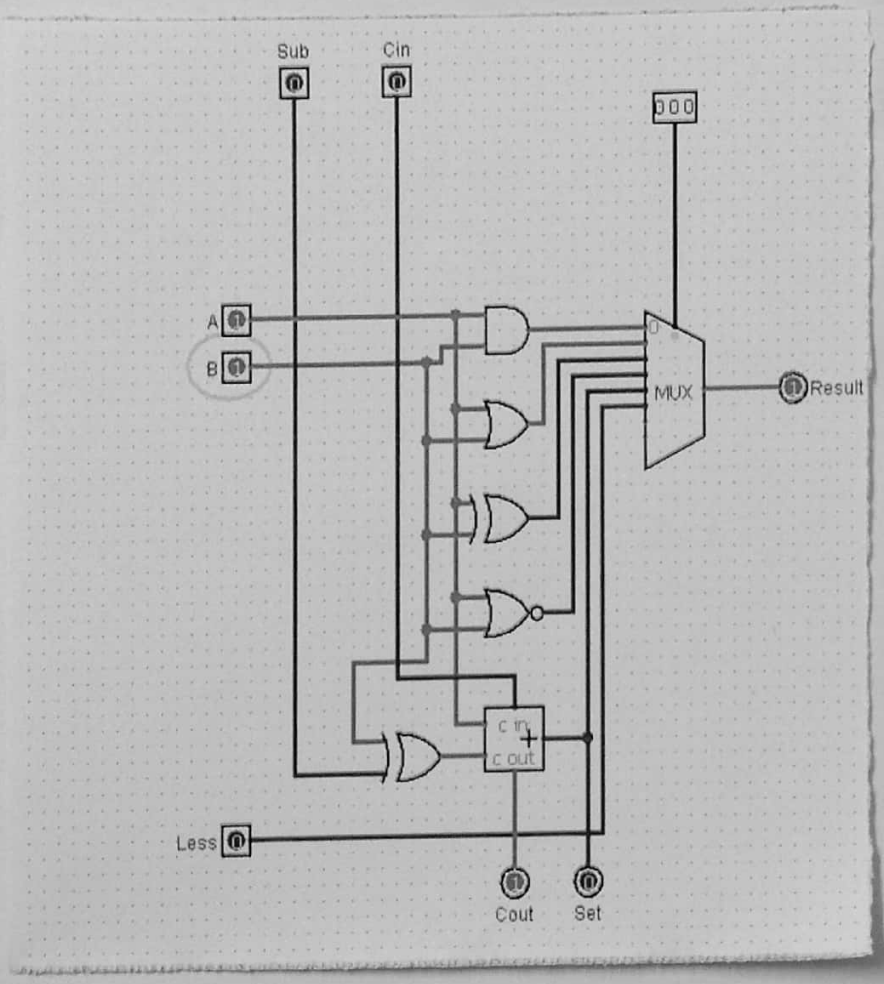
List out the steps in designing ALU

* Add the two i/p pins. Name them A and B
* Add or, and, ex-or, nor gates and a 1-bit adder
* Connect the A's and B's of all the gates to their respective pins.
* Add an o/p pin and name it result.
* Add a 1-bit multiplexer with 3 select bits.
* Connect o/p's of all the gates to the mux.
* Connect 3-bit input pin to mux.
* Add i/p pin to Cin, and output pin to Cout.
* 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
* Add another i/p and name it less. Connect it to the mux.
* Add an output pin and name it set, connect it to the ~~multiplexer~~ output of adder unit.

Snapshot:



Snapshot:



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: Joel John Thomas	**Marks:** /10	**Date:**
USN: 1MS18CS053	**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

* Add a RAM with seperate load and store selected.
* Add a counter and connect Q to A of the RAM.
* Add a controller buffer and connect its o/p to the RAM.
* Add a clock and connect to the i/p of the buffer.
* Add a TTY unit and with 32 rows and columns.
  Make the connections with RAM.
* Add a 7-bit random number generator, connect Q to D.
* Add another controller buffer, connect it to TTY.
  Also add an i/p pin to the buffer.
* Connect the o/p of the second buffer to the counter.
* Connect a button to the counter.

Observations and Snapshots:

Gpu Snapshots: