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

Department of CSE

Programme: B.E
Course: Computer Organization
CS45

Term: Jan to May 2019
Course Code:

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

Name: NISHIT KHAITAN	Marks: /10	Date: 22/05/20
USN: 1MS18CS086	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

Name - Nishit Khaitan
USN - 1MS18CS086
Course - Computer Organisation (CS45)

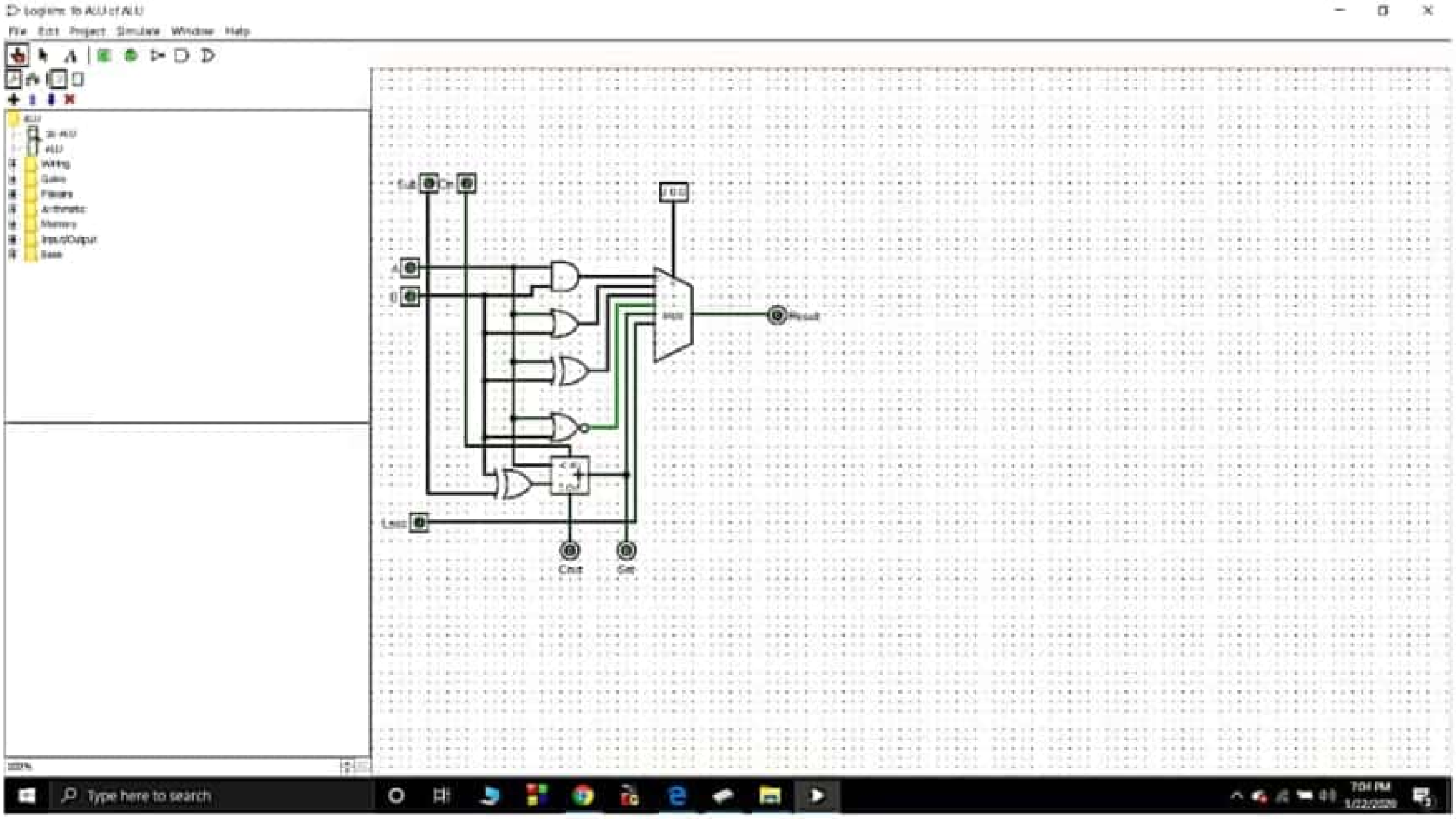
Date : 22/05/2020
IV - B
Activity : V

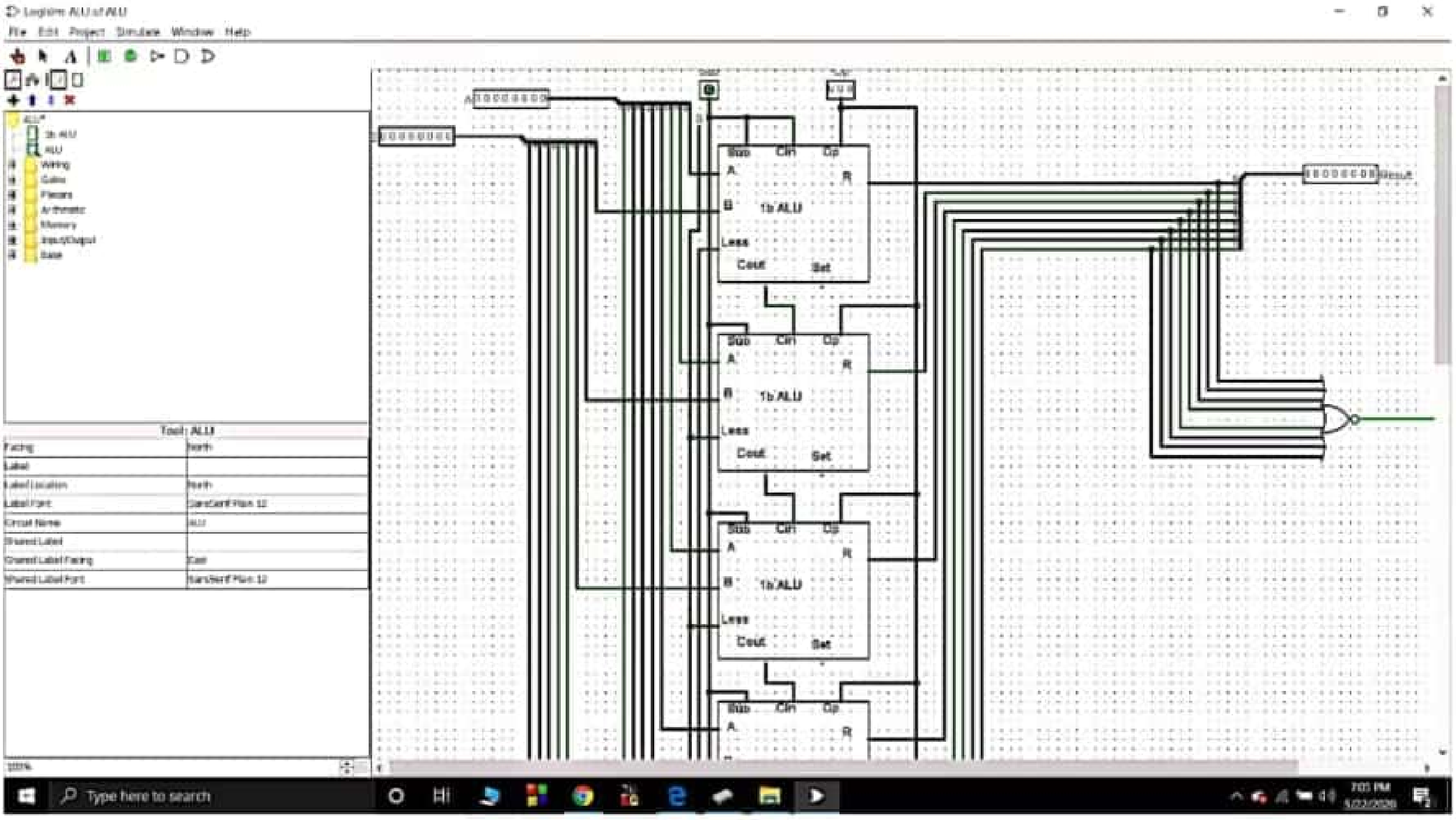
Designing an ALU to perform arithematic and logical functions using logism simulator

List out the steps in designing ALU

1. Add the two i/p pins, Name them A and 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 the outputs of all gates to the mux
7. Connect 3-bit input pin to mux
8. Add input pin to Cin and output pin to Cout
9. Add an EX-OR gate. Connect its O/P to Cout. The first input must be connected to B and the second to another input pin SUB
10 Add another input and name it less. Connect it to the MUX
11. Add an output pin and name it set, connect it to the output of adder unit.

Snapshots :


Logisim: 1b ALU of ALU
File Edit Project Simulate Window Help
ALU
32-ALU
ALU
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Sub
Cin
A
B
MUX
Result
Less
Cout
Set
Type here to search
7:04 PM
1/22/2020



Logisim: ALU of ALU

Tool: ALU

Facing	North
Label	
Label Location	North
Label Font	SansSerif Plain 12
Circuit Name	ALU
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Plain 12

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: NISHIT KHAITAN	Marks: /10	Date: 22/05/20
USN: 1MS18CS086	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

Observations and Snapshots:

Name - Nishit Khaitan Date- 22/05/20

USN - 1MS18CS086 IV-B

Course - Computer Organisation (CS45) Activity : VI

Designing memory system using Logism simulator

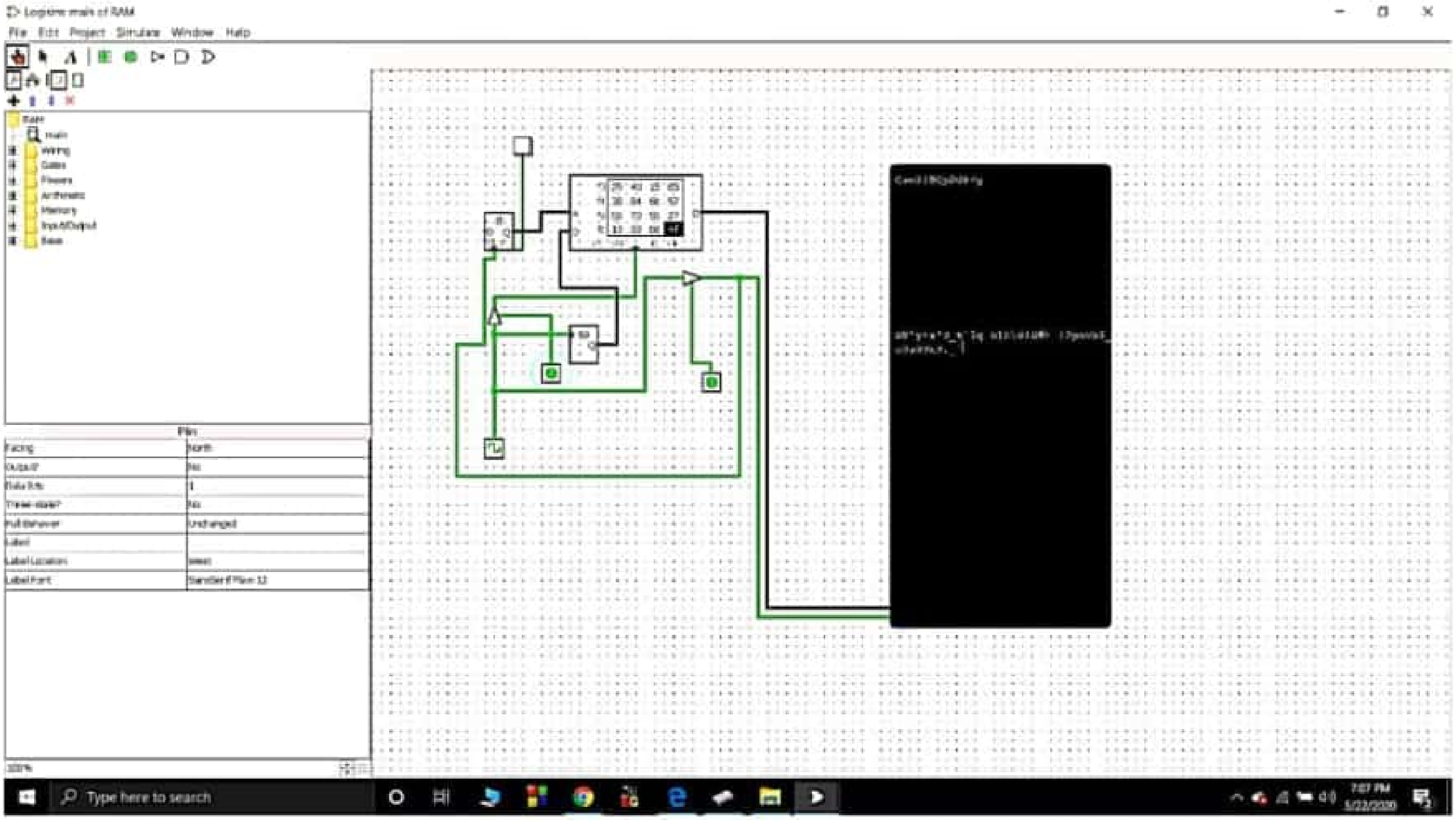
List out thee steps in designing memory system :

1. Add a RAM with separate load and store selected

~~2. Add a counter (clock) and connect to the~~

2. Add a counter and convert Q to A of the RAM
3. Add a controller buffer and connect its O/P to the RAM
4. Add a clock and connect to the input 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 input pin to the buffer
8. Connect the O/P of the second buffer to the counter
9. Connect a button to the counter

Snapshot :



Pin

Facing	North
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12

```
[illegible]

[illegible]
```

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

Department of CSE

Programme: B.E
Course: Computer Organization
CS45

Term: Jan to May 2019
Course Code:

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

Name: NISHIT KHAITAN	Marks: /10	Date: 22/05/20
USN: 1MS18CS086	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.

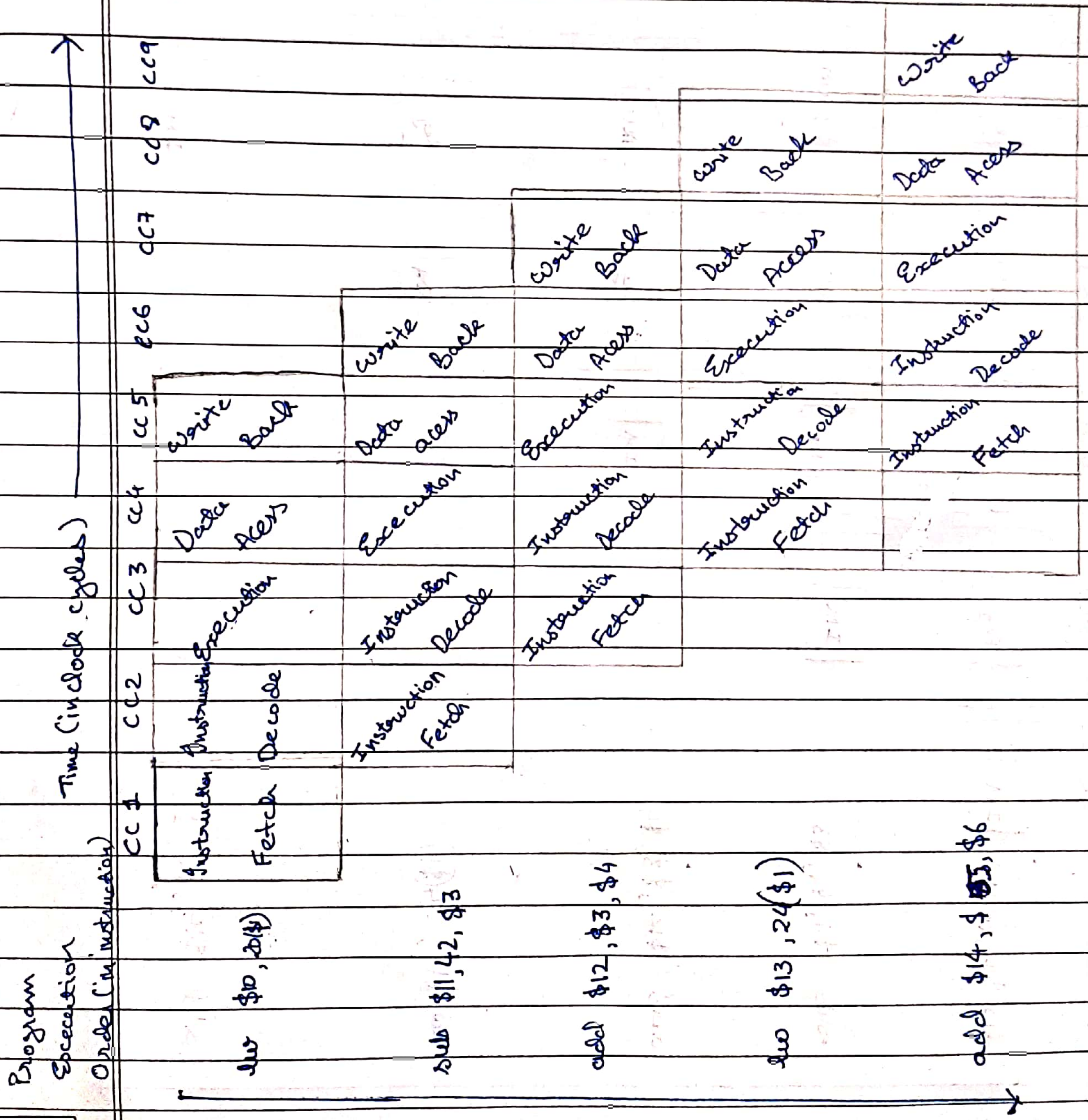
```
lw   $10,20($1)
sub   $11, 42, $3
add  $12, $3, $4
lw   $13, 24($1)
add  $14, $5, $6
```

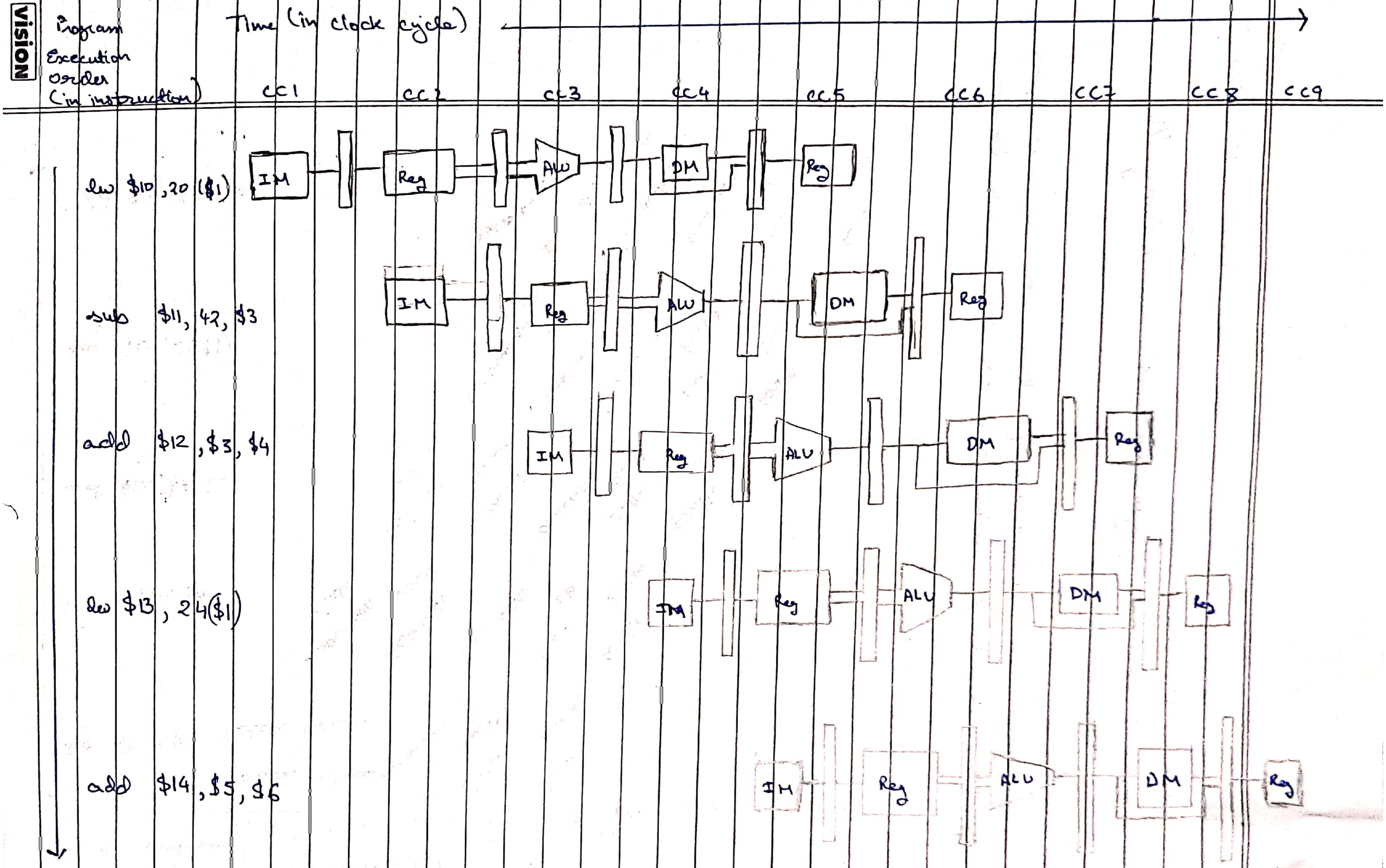
Observations and Snapshots: Take the snap shot of CPU statistics and pipeline design.

Name - Nishit Khaitan
USN - 1MS18CS086
Course - Computer Organisation (CS45)

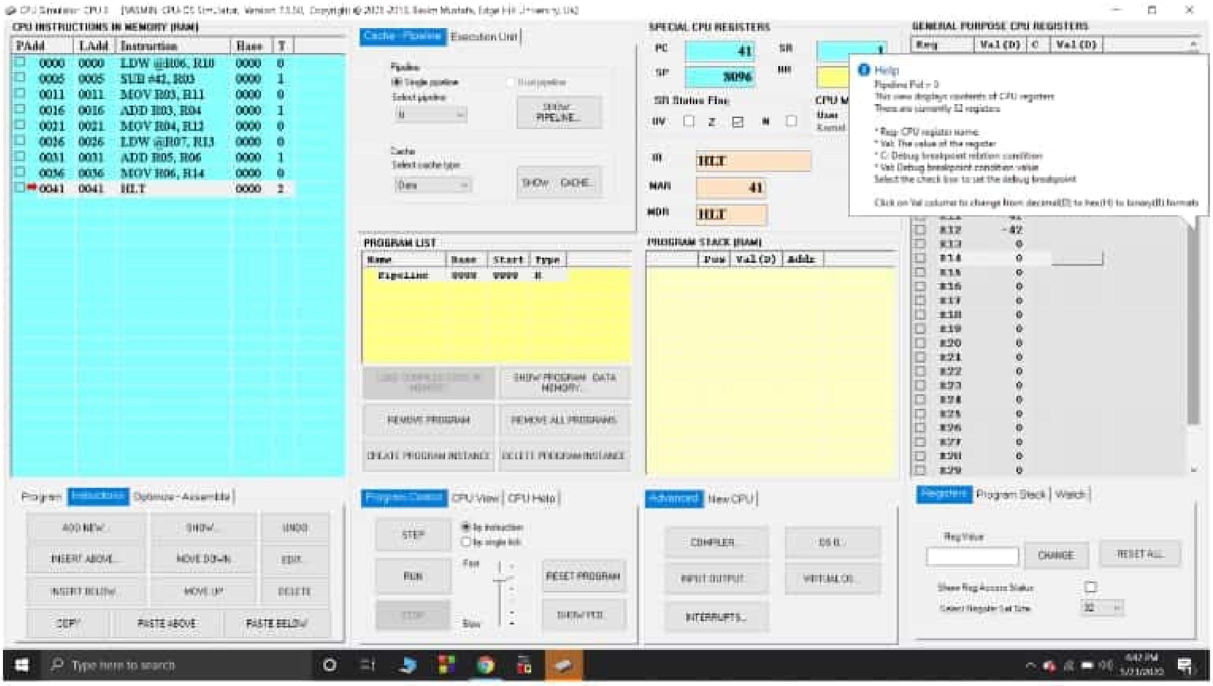
Date - 22/05/20
IV - B
Activity : VII

Analyzing the performance of the CPU by overlapping of instructions using ~~CPU-OS~~ CPU-OS-SIM simulator.




Program Execution order (in instruction)
Time (in clock cycle)
CC1
CC2
CC3
CC4
CC5
CC6
CC7
CC8
CC9
lw $10, 20($1)
sub $11, $2, $3
add $12, $3, $4
lw $13, 24($1)
add $14, $5, $6
IM
Reg
ALU
DM
Reg

Observations and Snapshots: Take the snap shot of CPU statistics and pipeline design.



CPU INSTRUCTIONS IN MEMORY (RAM)

PAdd	LAdd	Instruction	Base	T
0000	0000	LDW @R06, R10	0000	0
0005	0005	SUB #42, R03	0000	1
0011	0011	MOV R03, R11	0000	0
0016	0016	ADD R03, R04	0000	1
0021	0021	MOV R04, R12	0000	0
0026	0026	LDW @R07, R13	0000	0
0031	0031	ADD R05, R06	0000	1
0036	0036	MOV R06, R14	0000	0
0041	0041	HLT	0000	2

SPECIAL CPU REGISTERS

PC 41

SP 8096

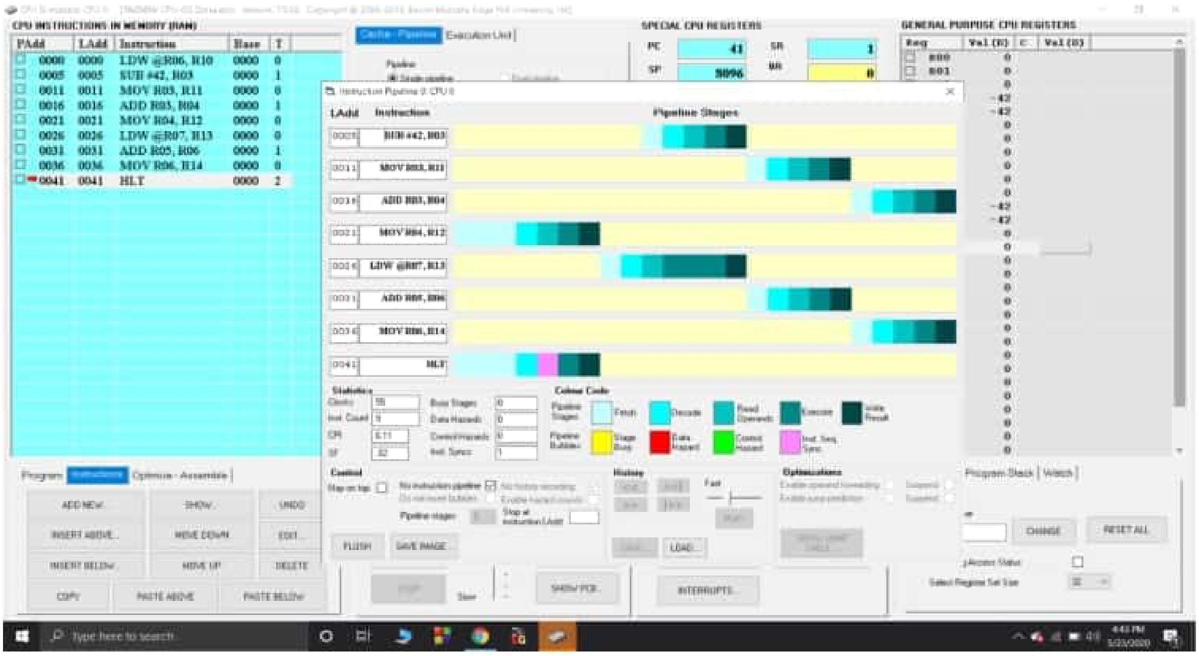
IR HLT

MAR 41

MDR HLT

PROGRAM LIST

Name	Base	Start	Type
Pipeline	0000	0000	0



Instruction Pipeline 0, CPU 0

LAdd	Instruction
0005	SUB #42, R03
0011	MOV R03, R11
0016	ADD R03, R04
0021	MOV R04, R12
0026	LDW @R07, R13
0031	ADD R05, R06
0036	MOV R06, R14
0041	HLT

Statistics

Clocks	55	Busy Stages	0
Inst. Count	9	Data Hazards	0
CPI	6.11	Control Hazards	0
SF	32	Inst. Sync	1

Scanned with CamScanner