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

Department of CSE

Programme: B.E **Term: Jan to May 2020**
Course: Computer Organization **Course Code: CS45**

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

Name: Neville Joseph Roy	**Marks: /10**	**Date:18/05/2020**
USN:1MS18CS082	**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

Building A 1-Bit ALU·

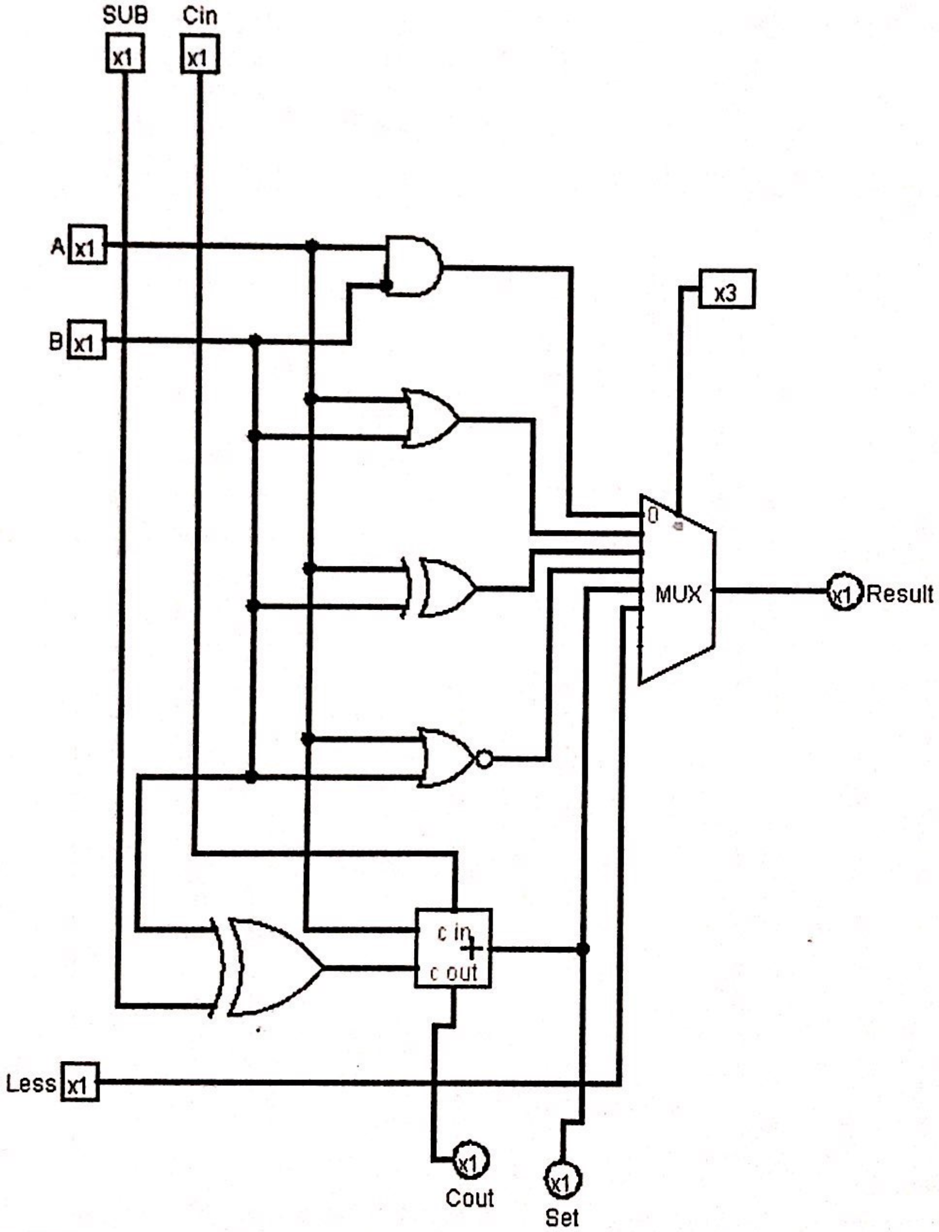
- Select and paste the basic gates ie. the AND, OR, NOR, XOR gates on the screen.
- Select an adder for ADDSUB operations
- Select and paste 2 inputs A and B and 1 output RESUL for the gates and make the respective conditions
- Select a MUX with 3 select bits and give the output of each gate as input for the MUX and give its output to the Output posted earlier.
- Select a 3 data bit input for the select line of the MUX
- Select an input for the C-in and output for C-out of the adder
- To make the adder subtractor, select a XOR gate whose input is one of the initial inputs (B) and SUB input and output is given as the B input of adder. This allows for A-B.
- Create another input Less and add it as Input to the MUX
- Select another Output SET which getsts signal from op of adder
- Switch to object view & resize and name the terminals as needed

BUILDING AN 8-BIT ALU

- Create an empty object.
- Drag 8 1-Bit ALU onto the screen.
- Select a SPLITTER with 8 Fan out, and 8bit width.
- Select an 8-bit input and connect it as the input for the SPLITTER
- This is for input A. Repeat the same process for input B.
- Connect each of the splitter outputs, of both splitters, to the A and B inputs of the 1 bit ALU respectively.
- Connect the C-out of a 1-bit ALU to the C-in of the next ALU and continue for all 8 ALU's, except for the first & last ALU's.
- Select a 3-bit input and connect it to the OP of all the ALU's
- Select a input SUB and connect it to the SUB input of all ALU's
- Since for the Subtract function Both SUB and C-in for adder must be 1, connect the same SUB input to the C-in input of all ALUs.
- Connect the SET of the last ALU to the LESS of the first ALU and the LESS input of all the other ALUs are set to 0 by connecting it to a CONSTANT object.
- Select anothe 8 Fan out splitter which takes the Result of all ALUs as input and the output is connected to an 8-Data bit output.
- For the EQUAL TO operation, select a NOR gate with 8 inputs, which it recieves from the RESULT of each ALU and a 1bit output to show whether result is 0.

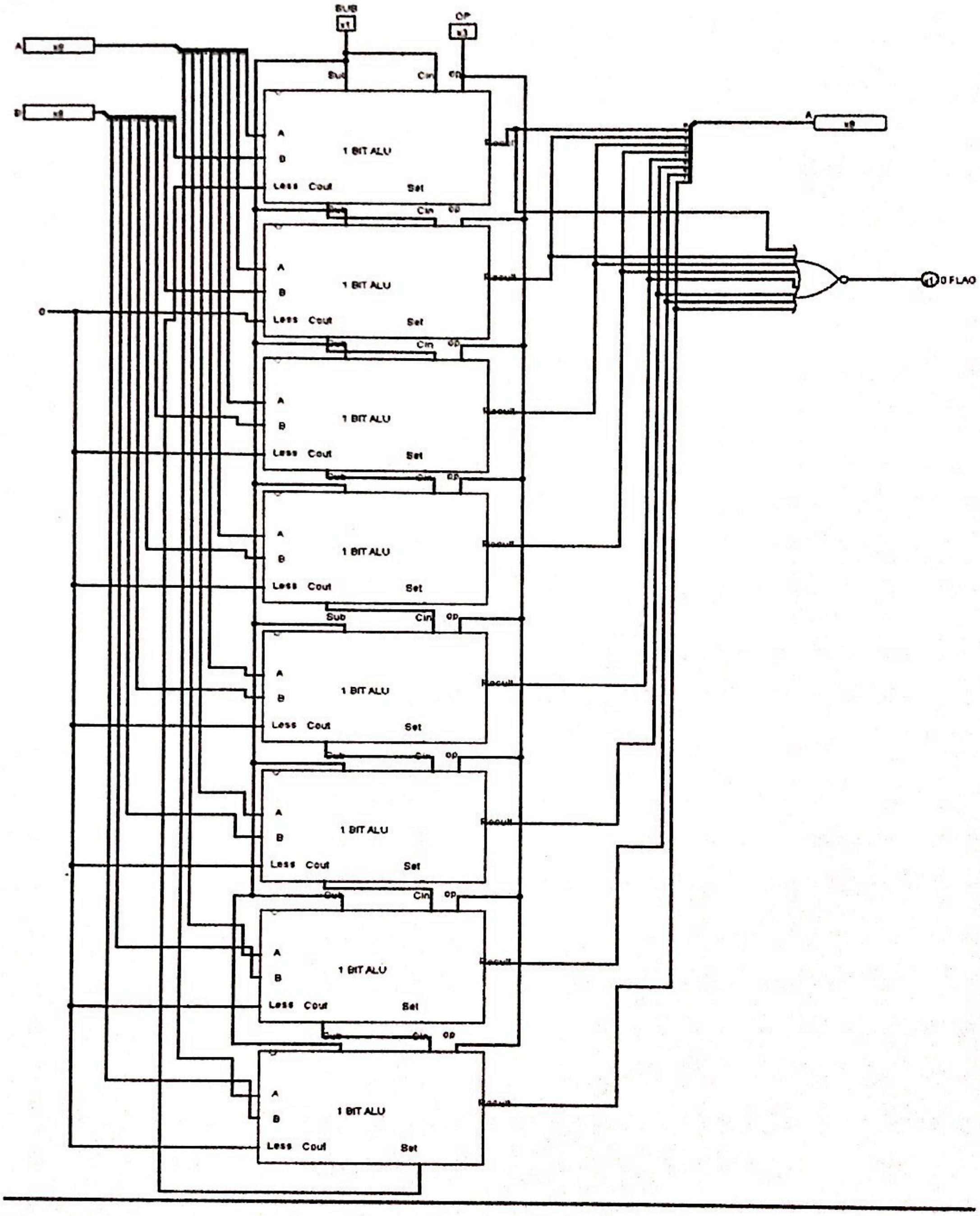
Snapshots:

1-BIT ALU



Public

8-BIT ALU



Public

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

Department of CSE

Programme: B.E | **Term: Jan to May 2020**
Course: Computer Organization | **Course Code: CS45**

Activity VI: Designing memory system using Logisim simulator.

Name: Neville Joseph Roy	**Marks: /10**	**Date:18/05/2020**
USN: 1MS18CS082	**Signature of the Faculty:**	

Objective: To simulate the writing operation on memory.

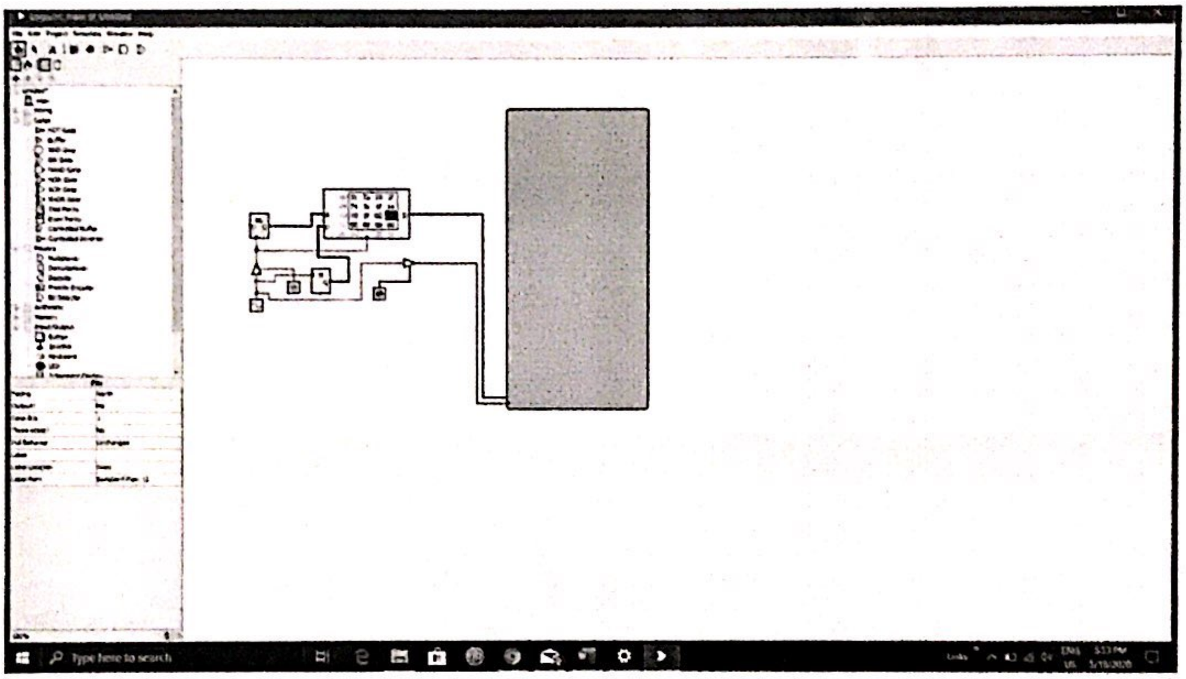
Simulator Description: Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to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

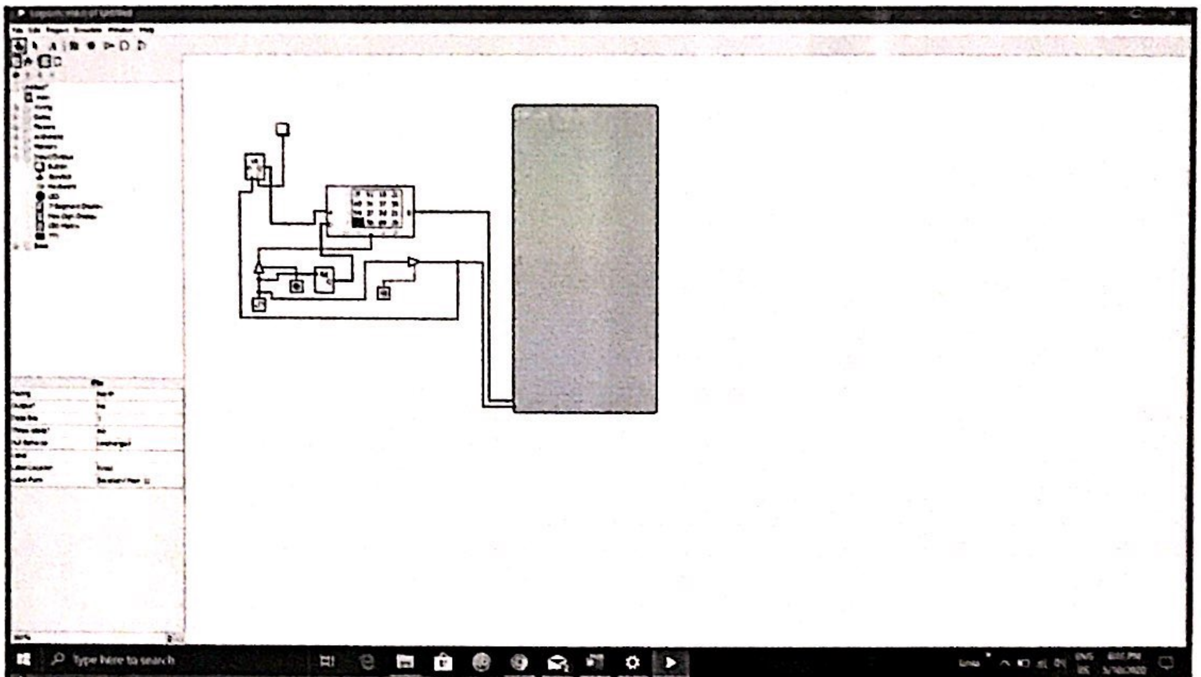
Activity to be performed by students:

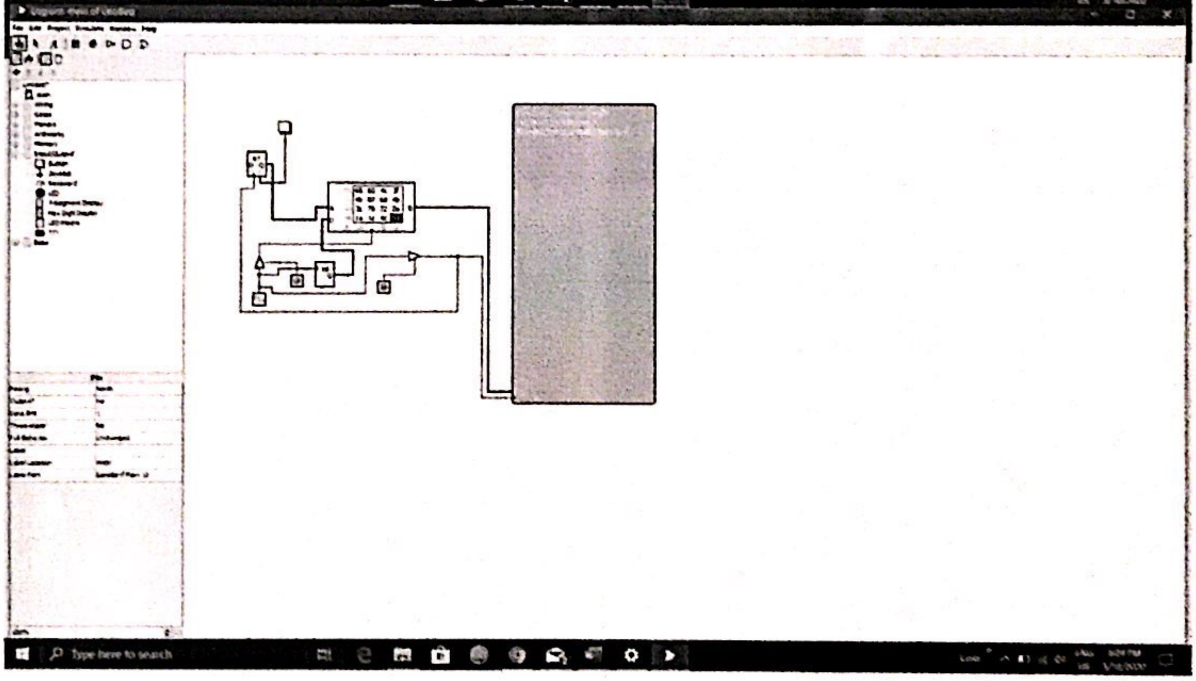
List out the steps in designing memory system

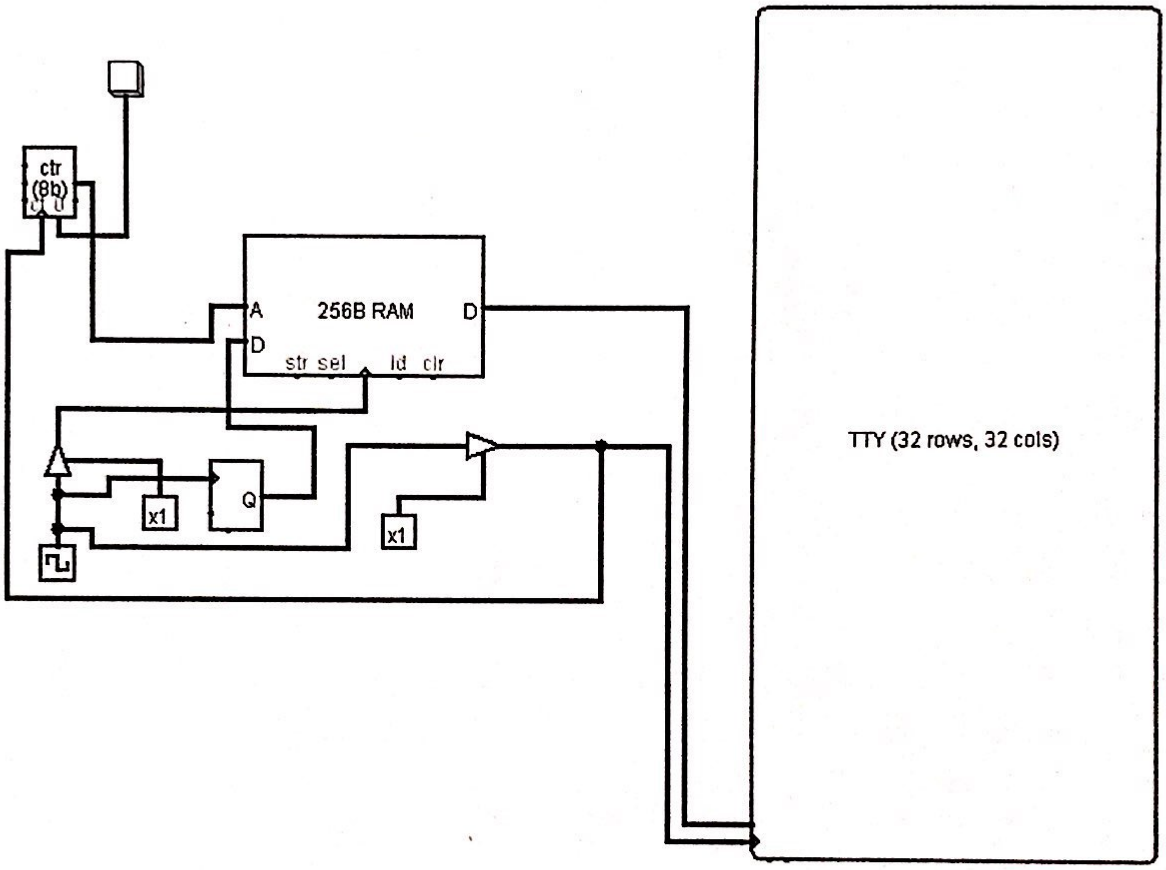
- Add a RAM with separate load and store selected.
- Add a counter and convert Q to A of the RAM
- Add a controller buffer. and convert its o/p to RAM
- Add a clock and connect to the i/p of the buffer.
- Add a TTY unit 0 with 32 rows and columns Make the connection with RAM.
- Add a 7 bit random number generator, connect Q to D
- Add another controlled 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:




Type here to search


Type here to search


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

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

Department of CSE

Programme: B.E
Course: Computer Organization

Term: Jan to May 2020
Course Code: CS45

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

Name: Neville Joseph Roy	**Marks: /10**	**Date:18/04/20**
USN:1MS18CS082	**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:

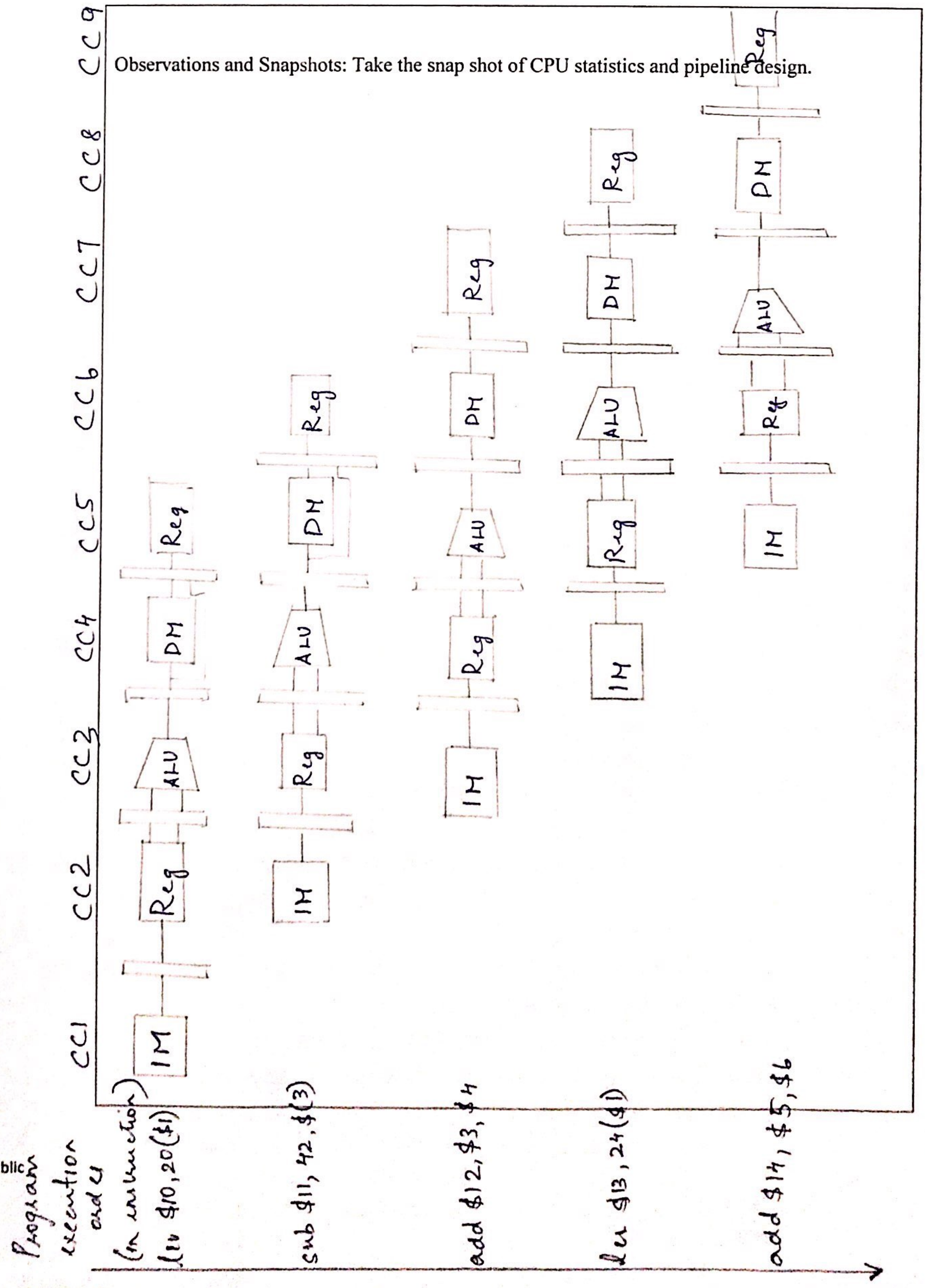
With diagram demonstrate the execution of the following instructions using pipelining technique.

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

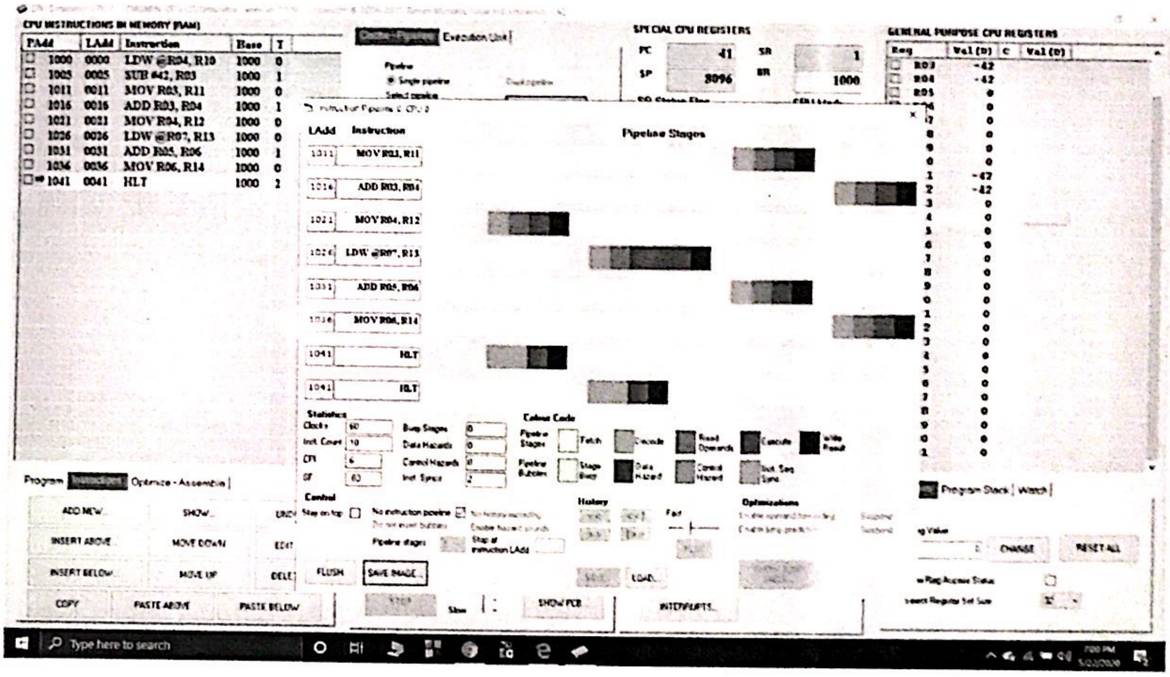
Time (in clock cycles) →

Program execution order (in instruction)	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9
lw $10, 20 ($1)	Instruction fetch.	Instruction decode.	Execution	Data access	Write back				
sub $11, $2, $3		Instruction fetch	Instruction decode	Execution	Data access	Write back			
add $12, $3, $4			Instruction fetch	Instruction decode	Execution	Data access	Write back		
lw $13, 24 ($1)				Instruction Set	Instruction decode	Execution	Data. Access	Write Back.	
add $14, $5, $6					Instruction Set	Instruction decode	Execution	Data Access	Write Back

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



Public



CPU INSTRUCTIONS IN MEMORY (RAM)

PAdd	LAdd	Instruction	Base	T
□ 1000	0000	LDW @R04, R10	1000	0
□ 1005	0005	SUB #42, R03	1000	1
□ 1011	0011	MOV R03, R11	1000	0
□ 1016	0016	ADD R03, R04	1000	1
□ 1021	0021	MOV R04, R12	1000	0
□ 1026	0026	LDW @R07, R13	1000	0
□ 1031	0031	ADD R05, R06	1000	1
□ 1036	0036	MOV R06, R14	1000	0
□ 1041	0041	HLT	1000	2

SPECIAL CPU REGISTERS

PC	41	SR	1
SP	8096	BR	1000

GENERAL PURPOSE CPU REGISTERS

Reg	Val (D)	C	Val (D)
□ R03	-42		
□ R04	-42		
□ R05	0		

Pipeline
● Single pipeline

Instruction Pipeline C CPU 0

LAdd	Instruction
1011	MOV R03, R11
1016	ADD R03, R04
1021	MOV R04, R12
1026	LDW @R07, R13
1031	ADD R05, R06
1036	MOV R06, R14
1041	HLT
1041	HLT

Statistics

Clocks	60	Busy Stages	0
Inst. Count	10	Data Hazards	0
CPI	6	Control Hazards	0
SF	60	Inst. Syncs	2

Control
Stay on top □ No instruction pipeline ☑