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: Gautham Lankapalli	Marks: /10	Date:
USN:1MS18CS061	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

CO - Lab - 5

Name: Gautham. L
USN: 1MS18CS06
SEM: 4 + B

1) List out the steps in designing ALU.

(i) Add the 2 i/p pins, name them A and B.

(ii) Add or, and, ex-or, not gates and a 1-bit address.

(iii) Connect the A's and B's of all the gates to their respective pins.

(iv) Add an output pin and name it result.

(v) Add a 1-bit multiplier with 3-select bits.

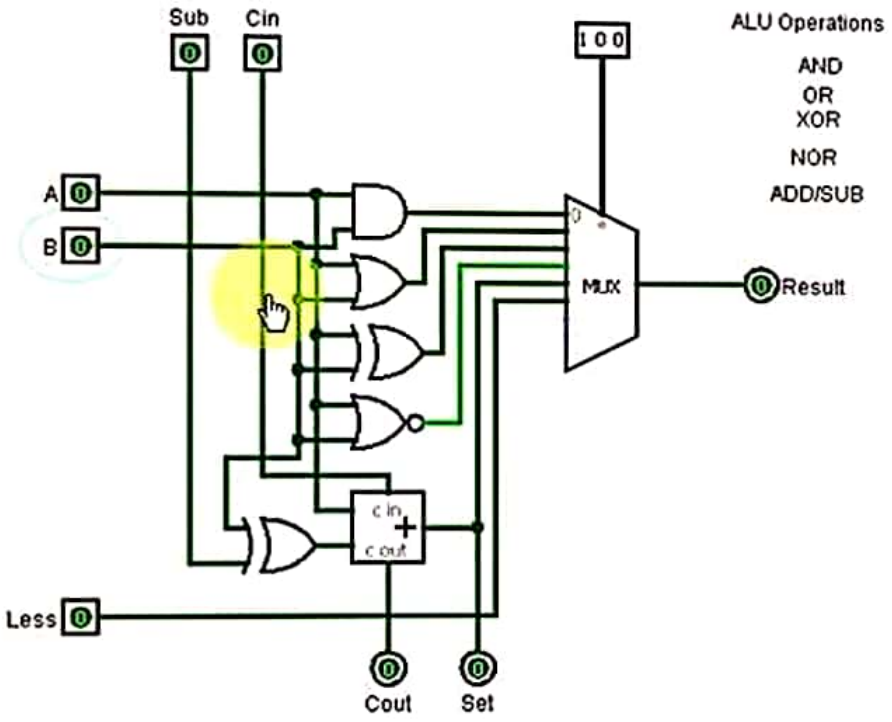
(vi) Connect outputs of all the gates to the mux.

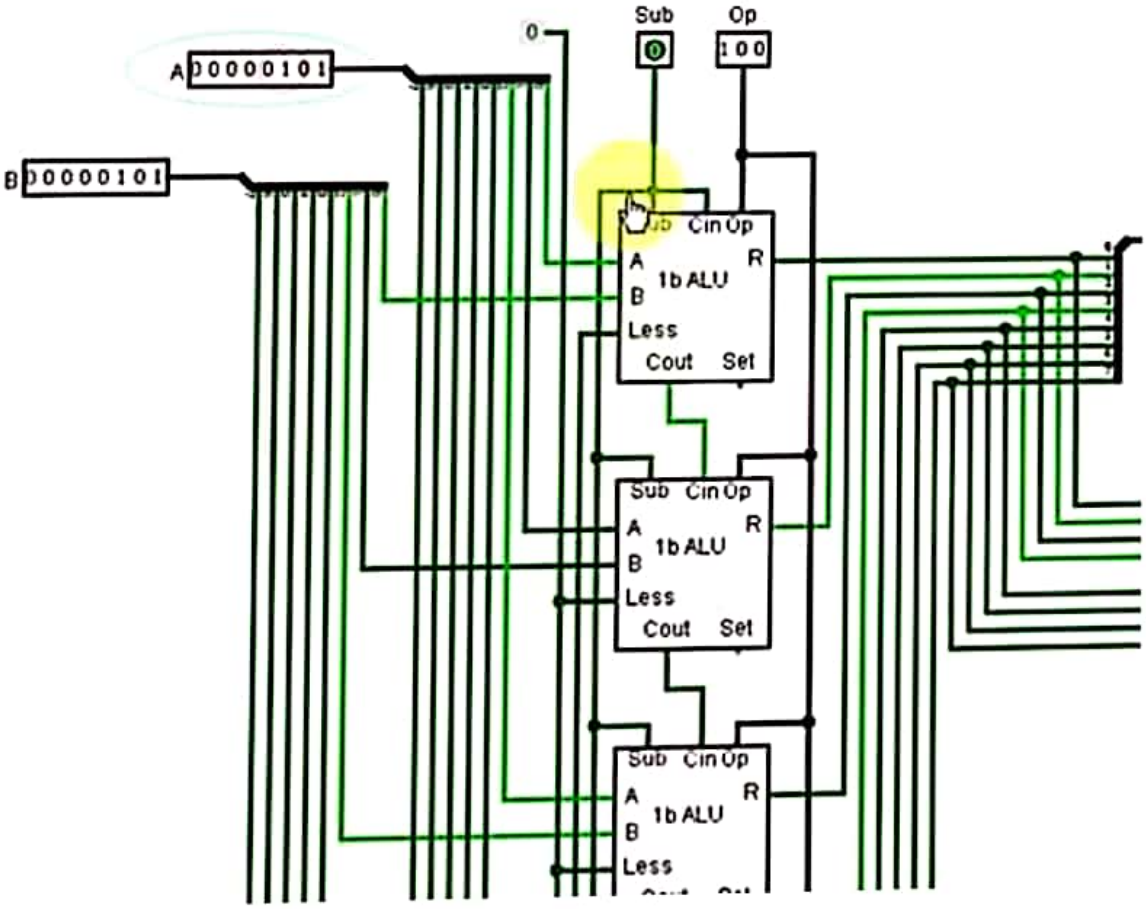
(vii) Connect 3-bit input pin to Mux.

(viii) Add i/p pin to Cin, output pin to Cout.

(ix) Add another i/p and name it less. Connect it to the mux.

(x) Add an output pin and name it set connect it to the ~~op~~ of adder. unit.





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: Gautham Lankapalli	**Marks: /10**	**Date:**
USN:1MS18CS061	**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

Name : Gautham. L
USN : 1MS18CS061
SEM : 4 - B

Co - Lab - 6

1) List out the steps in designing memory system.

(i) Add a RAM with seperate load and store selected.

(ii) Add a Counter and Connect Q to A of the Ram.

(iii) Add a Controller buffer and Connect its o/p to the Ram.

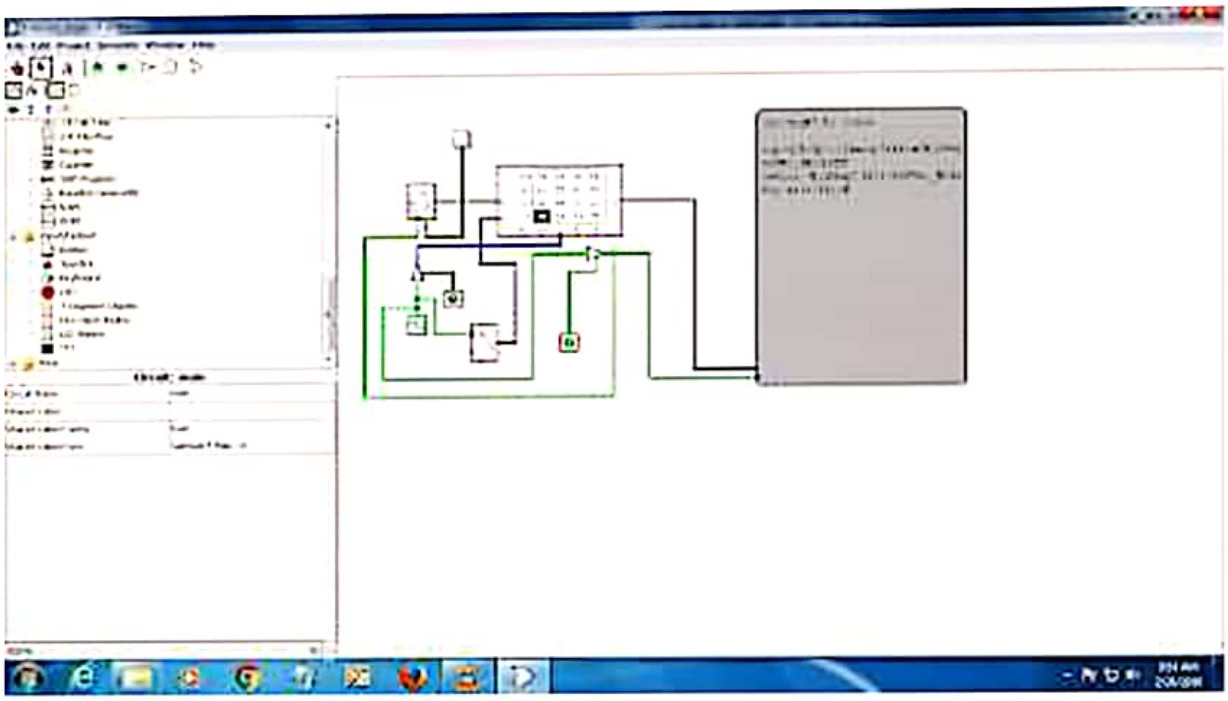
(iv) Add a clock and connect to the i/p of the buffer.

(v) Add a TTY unit with 32 rows and Columns. Make the connections with Ram.

(vi) Add a 7-bit random number generator, Connect Q to D.

(vii) Connect the output of the second buffer to the Counter.

(viii) Connect a button to the counter.



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:	Marks: /10	Date:
USN:	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   S10,20(S1)
sub  S11, 42, S3
add  S12, S3, S4
lw   S13, 24(S1)
add  S14, S5, S6
```

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

Time (in clock cycles) →

	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	Data access Execution	lw Data accen	Write back			
add $12, $3, $4			Instruction fetch	Instruction decode	Execution	Data access	Write back		
lw $13, 24($1)				Instruction fetch	Instruction decode	Execution	Data accen	Write back	
add $14, $5, $6					Instruction fetch	Instruction decode	Execution	Data accen	Write back

Time (in clock cycles) →

CC1 CC2 CC3 CC4 CC5 CC6 CC7 CC8 CC9

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