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: Vaishnavi D.Jadhav	**Marks: /10**	**Date:24/05/2020**
USN:1MS19CS410	**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.

Name :- Vaishnavi . Jadhav USN:- 1MS19CS410

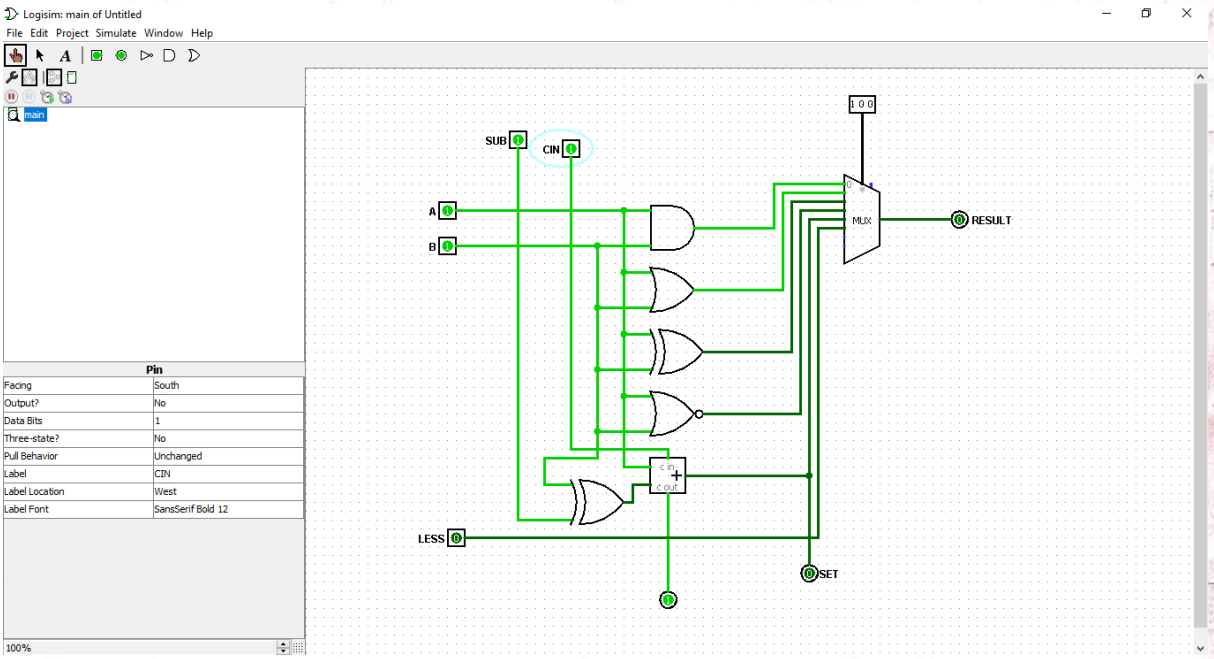
Course code :- CS45
Course :- Computer Organization
Activity V :- Designing ALU to perform arithmetic and logical functions using Logisim simulator.

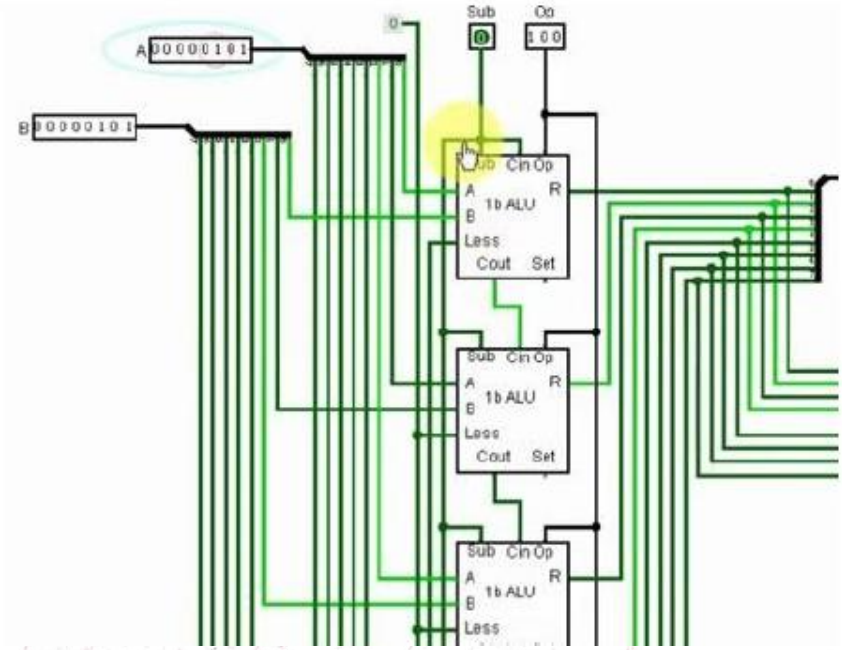
LIST OUT THE STEPS IN DESIGNING ALU.

1] Add Add two Input pins, Label them as A and B.
2] Drag and place OR, AND, EX-OR, NOR gates and 1-Bit ADDER.
3] Connect the input pins to all the gates i.e to their respective pins.
4] Add an output pin and label it as RESULT.
5] Drag and place / Add a 1-bit Multiplexer with 3 select bits.
6] Connect outputs of all the gates to Input of the multiplexer
7] Connect a 3-bit Input pin to multiplexer.
8] Add a Input pin, connect it to CIN of ADDER and label it as CIN.
9] Add a ~~Input~~ Output pin, connect it to COUT of Adder and label it as COUT.
10] Add an EX-OR gate, Connects its output pin to the ADDER's second Input pin.
11] Connect B to the first Input of EXOR gate.
12] Add a Input pin and label it as SUB, Connect SUB to second Input pin of XOR gate.
13] Add input pin and label it as LESS, Connect it to the multiplexer.
14] Add output pin and label it as SET, Connect it to the output of ADDER circuit.

Name :- Vaishnavi. Padhar USN :- 1MS19CS410

SNAPSHOTS:-





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: Vaishnavi D.Jadhav	**Marks: /10**	**Date:24/05/2020**
USN: 1MS19CS410	**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.

Name:- Vaishnavi. Jadhav USN:- 1MS19CS410

Course code:- 5545

Course:- Computer Organization.

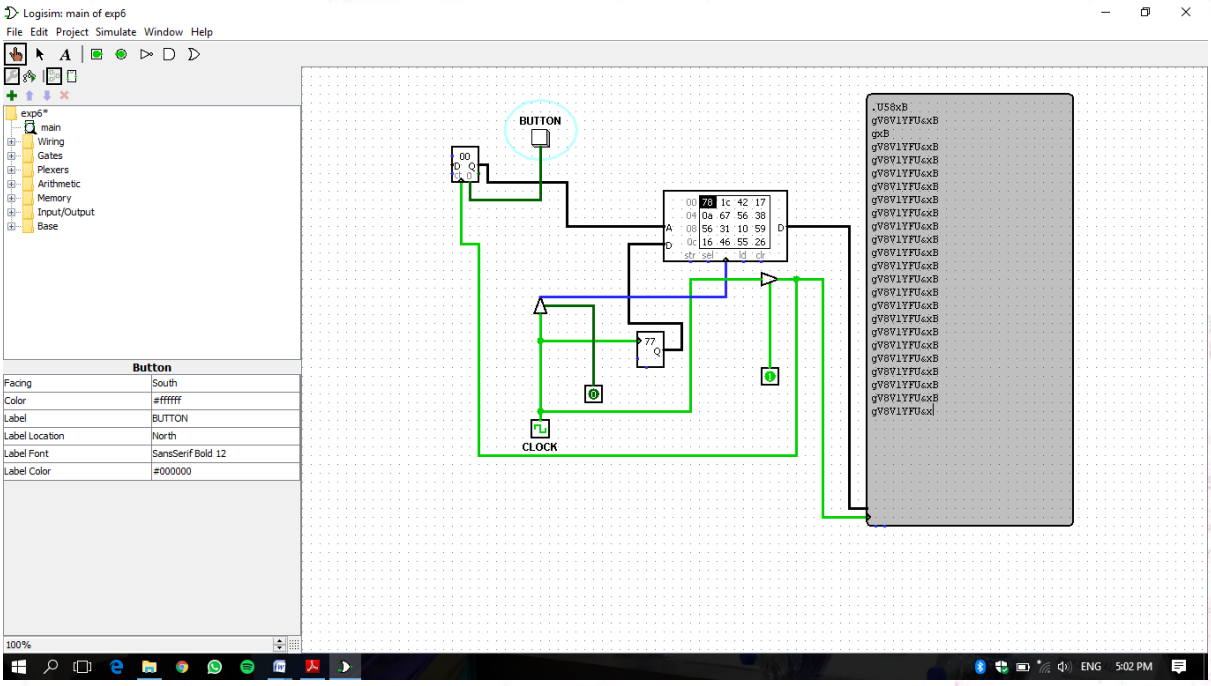
Activity:-(VI) Designing memory system using logisim simulator.

LIST OUT 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 controller Buffer and connect its output 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 and input pin to the buffer.
8] Connect the output of second buffer to the counter.
9] Connect a button to the counter.

Name:- Vaishnavi . Jadhav USN:- 1MS19CS410

SNAPSHOTS:-



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: Vaishnavi D.Jadhav	**Marks: /10**	**Date:24/05/2020**
USN: 1MS19CS410	**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.

Name :- Vaishnavi. Jadhav USN :- 1MS19CS410

Course code :- CS45

Course name :- Computer Organization

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

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

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

Name:- Vaishnavi. Jadhav USN:- 1MS19CS410

Program execution order (in instruction)

Time (in clock cycles) →

CC1 CC2 CC3 CC4 CC5 CC6 CC7 CC8 CC9

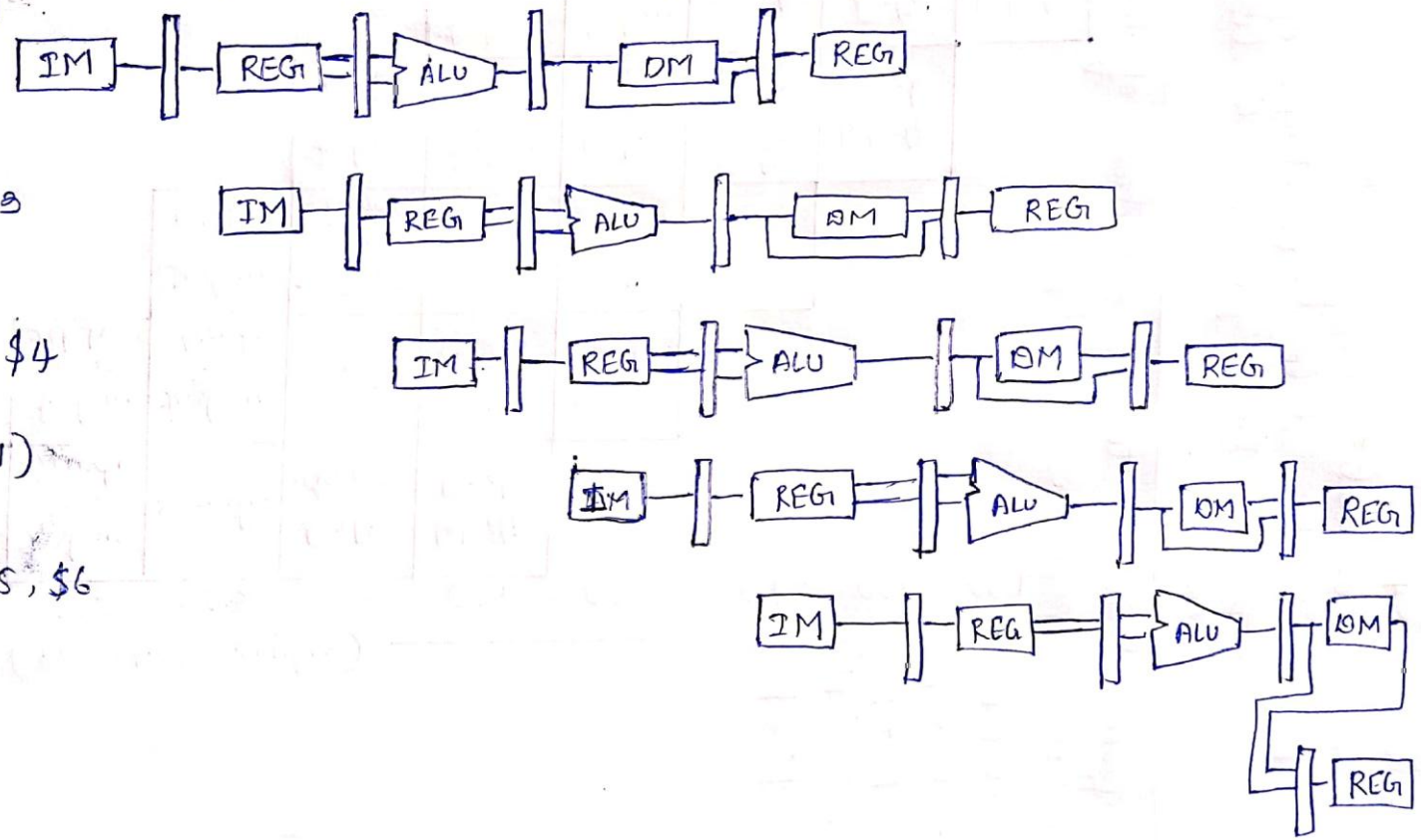
lw $10, 20($1)

sub $11, $2, $3

add $12, $3, $4

lw $13, 24($1)

add $14, $5, $6



SNAPSHOTS :-


CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
CPU INSTRUCTIONS IN MEMORY (RAM)
PAdd LAdd Instruction Base T
1000 0000 LDW @R06, 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
Program Instructions Optimize-Assemble
ADD NEW... SHOW... UNDO
INSERT ABOVE... MOVE DOWN EDIT...
INSERT BELOW... MOVE UP DELETE
COPY PASTE ABOVE PASTE BELOW
Cache - Pipeline Execution Unit
SPECIAL CPU REGISTERS
PC SR
GENERAL PURPOSE CPU REGISTERS
Reg Val (D) C Val (D)
RESET ALL
Instruction Pipeline 0: CPU 0
LAdd Instruction Pipeline Stages
1005 SUB #42, R03
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
Statistics
Clocks 17 Busy Stages 14
Inst. Count 9 Data Hazards 0
CPI 1.89 Control Hazards 0
SF 2.65 Inst. Syncs 1
Colour Code
Pipeline Stages: Fetch Decode Read Operands Execute Write Result
Pipeline Bubbles: Stage Busy Data Hazard Control Hazard Inst. Seq. Sync.
Control
Stay on top
No instruction pipeline
Do not insert bubbles
Pipeline stages 5
No history recording
Enable hazard sounds
Stop at instruction LAdd
FLUSH SAVE IMAGE...
History
Fast
PLAY
SAVE... LOAD...
Optimizations
Enable operand forwarding Suspend
Enable jump prediction Suspend
SHOW JUMP TABLE...
ENG 3:11 PM