Jawahar Education Societys Annasaheb Chudaman Patil College of Engineering, Kharghar, Navi Mumbai

NAME: PRIYUSH BHIMRAO KHOBRAGADE

PRN NO: 211112018

SUBJECT: DIGITAL LOGIC & COMPUTER ORGANIZATION AND ARCHITECTURE LAB

05

AIM: To implement ALU design.

Practical No.5

- **Aim**: To implement ALU design.

- **Objectives**: To implement the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division.

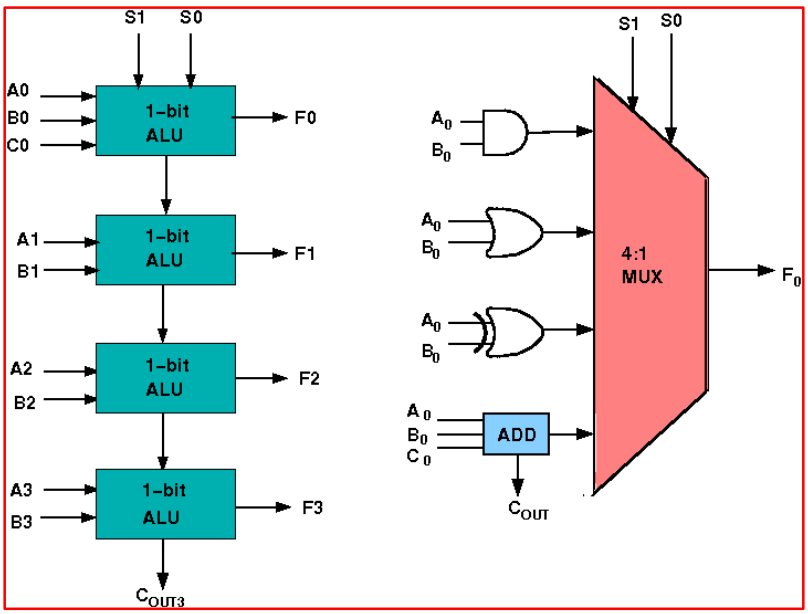
- **Outcomes**: Design the basic building blocks of a computer: arithmetic-logic unit, registers, central processing unit, and memory.

- **Hardware / Software Required:** Virtual Lab simulator for Computer Organization and Architecture developed by the Department of CSE, IIT Kharagpur.

- **Theory**:

Design of ALU:

ALU or Arithmetic Logical Unit is a digital circuit to do arithmetic Operations like addition, subtraction, division, multiplication and logical operations like and, OR, X-OR, NAND, NOR etc.



The 4-bit ALU block is combined using 4 1-bit ALU block

AIM: To implement ALU design.

● Design Issues:

The circuit functionality of a 1 bit ALU is shown here, depending upon the control signal S1 and S0 the circuit operates as follows:
for Control signal S1 = 0 , S0 = 0, the output is A AND B,
for Control signal S1 = 0, S0 = 1, the output is A OR B,
for Control signal S1 = 1 , S0 = 0, the output is A XOR B,
for Control signal S1 = 1 , S0 = 1, the output is A Add B.

The truth table for 16-bit ALU with capabilities similar to 74181 is shown here:

Required functionality of ALU (inputs and outputs are active high)

MODE SELECT				**F_N FOR ACTIVE HIGH OPERANDS**	
INPUTS				LOGIC	ARITHMETIC (NOTE 2)
S3	S2	S1	S0	(M = H)	(M = L) **(C_n=L)**
L	L	L	L	A'	A
L	L	L	H	A'+B'	A+B
L	L	H	L	A'B	A+B'
L	L	H	H	Logic 0	minus 1
L	H	L	L	(AB)'	A plus AB'
L	H	L	H	B'	(A + B) plus AB'
L	H	H	L	A ⊕ B	A minus B minus 1
L	H	H	H	AB'	AB minus 1
H	L	L	L	A'+B	A plus AB
H	L	L	H	(A ⊕ B)'	A plus B
H	L	H	L	B	(A + B') plus AB
H	L	H	H	AB	AB minus 1
H	H	L	L	Logic 1	A plus A (Note 1)
H	H	L	H	A+B'	(A + B) plus A
H	H	H	L	A+B	(A + B') plus A
H	H	H	H	A	A minus 1

The L denotes the logic low and H denotes logic high.

AIM: To implement ALU design.

● Objective:

Objective of 4 bit arithmetic logic unit (with AND, OR, XOR, ADD operation):

1. Understanding behaviour of arithmetic logic unit from working module and the module designed by the student as part of the experiment

2. Designing an arithmetic logic unit for given parameter

Examining behaviour of arithmetic logic unit for the working module and module designed by the student as part of the experiment (refer to the circuit diagram):

Loading data in the arithmetic logic unit (refer to procedure tab for further detail and experiment manual for pin numbers):

- load the two input numbers as:
  - A(A3 A2 A1 A0): A3=1, A2=1, A1=0, A0=0
  - B(B3 B2 B1 B0): B3=1, B2=0, B1=0, B0=1
  - carry in(C0)=0

Examining the AND behaviour:

- load data in select input as:
  - S1=0, S0=0 `
- check output:
  - F3=1, F2=0, F1=0, F0=0
  - cout=0 `

Examining the OR behaviour:

- load data in select input as:
  - S1=0, S0=1 `
- check output:
  - F3=1, F2=1, F1=0, F0=1
  - cout=0 `

Examining the XOR behaviour:

- load data in select input as:
  - S1=1, S0=0 `
- check output:
  - F3=0, F2=1, F1=0, F0=1
  - cout=0 `

AIM: To implement ALU design.

Examining the ADD behaviour:

- load data in select input as:
  - S1=1, S0=1 `
- check output:
  - F3=0, F2=1, F1=0, F0=1
  - cout=1
  -

● **Procedure:**

Procedure to perform the experiment: Design of 4 bit ALU

1. Start the simulator as directed. This simulator supports 5-valued logic.

2. To design the circuit we need 4 1-bit ALU, 11 Bit switch (to give input, which will toggle its value with a double click), 5 Bit displays (for seeing output), and wires.

3. The pin configuration of a component is shown whenever the mouse is hovered on any canned component of the palette. Pin numbering starts from 1 and from the bottom left corner (indicating with the circle) and increases anticlockwise.

4. For 1-bit ALU input A0 is in pin-9,B0 is in pin-10, C0 is in pin-11 (this is input carry), for selection of operation, S0 is in pin-12, S1 is in pin-13, output F is in pin-8 and output carry is pin-7

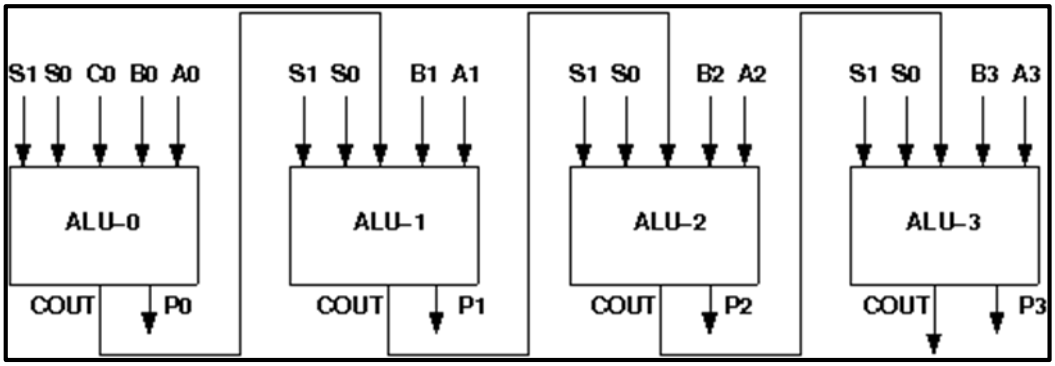
5. Click on the 1-bit ALU component (in the Other Component drawer in the pallet) and then click on the position of the editor window where you want to add the component (no drag and drop, simple click will serve the purpose), likewise add 3 more 1-bit ALU (from the Other Component drawer in the pallet), 11 Bit switches and 5 Bit Displays (from Display and Input drawer of the pallet ,if it is not seen scroll down in the drawer), 3 digital display and 1 bit Displays (from Display and Input drawer of the pallet, if it is not seen scroll down in the drawer)

6. To connect any two components select the Connection menu of Palette, and then click on the Source terminal and click on the target terminal. According to the circuit diagram connect all the components. Connect the Bit switches with the inputs and Bit displays component with the outputs. After the connection is over click the selection tool in the pallets.

7. See the output, in the screenshot diagram we have given the value of S1 S0=11 which will perform add operation and two number input as A0 A1 A2 A3=0010 and B0 B1 B2 B3=0100 so get output F0 F1 F2 F3=0110 as sum and 0 as carry which is indeed an add operation. You can also use many other combination of different values and check the result. The operations are implemented using the truth table for 4 bit ALU given in the theory.

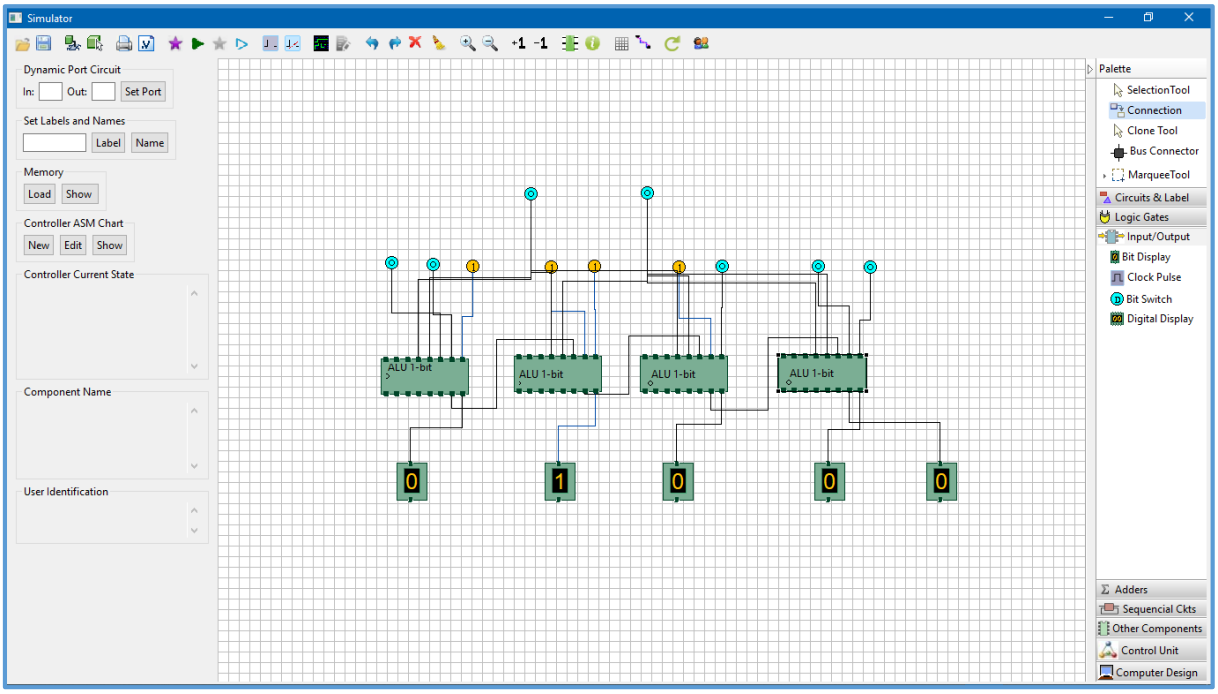
AIM: To implement ALU design.

- **Circuit diagram of 4 bit ALU:**



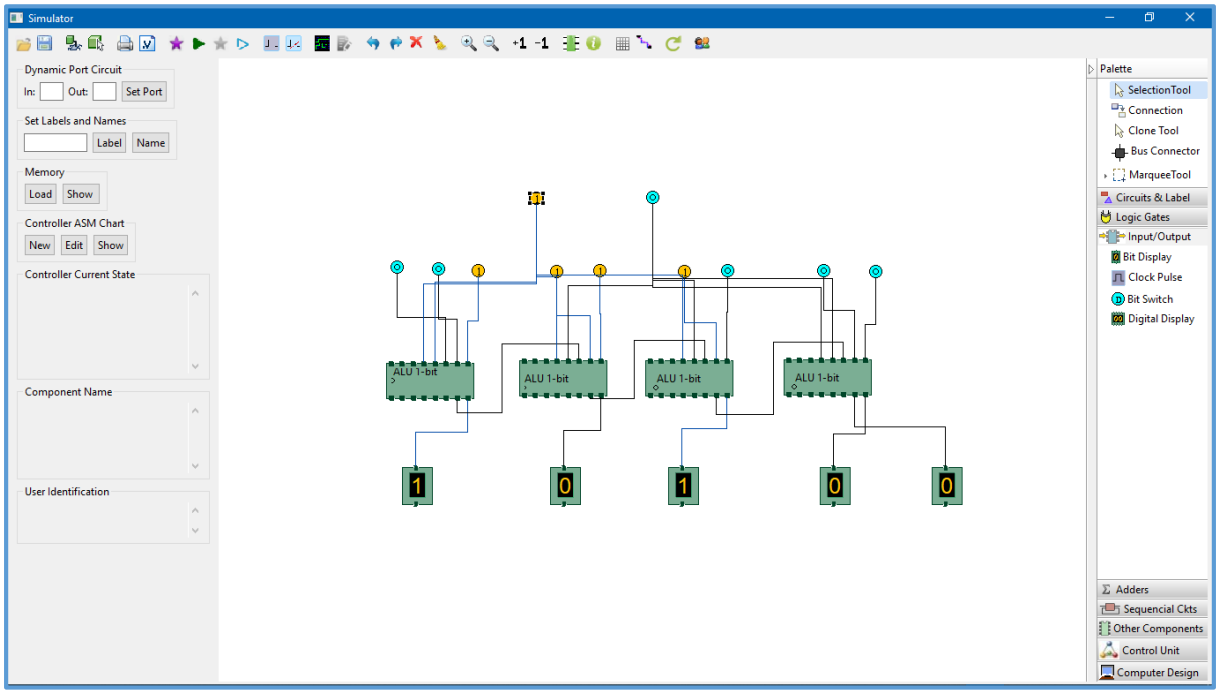
- **Results:**

AND OPRATION:

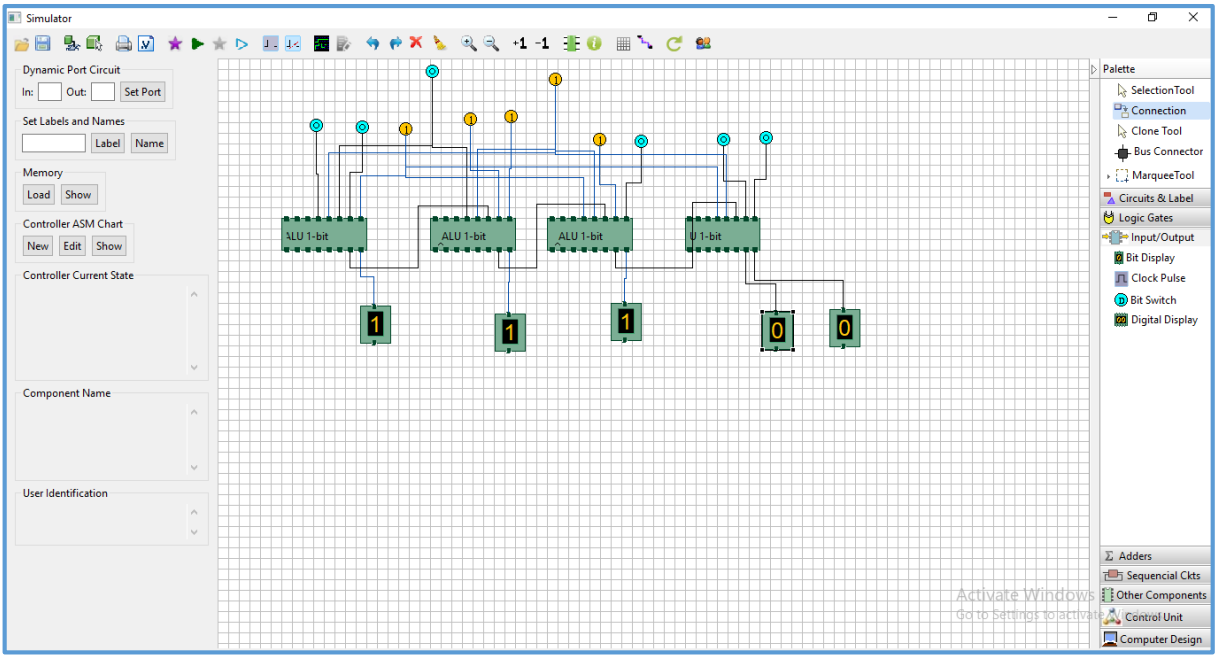


AIM: To implement ALU design.

XOR OPERATION:

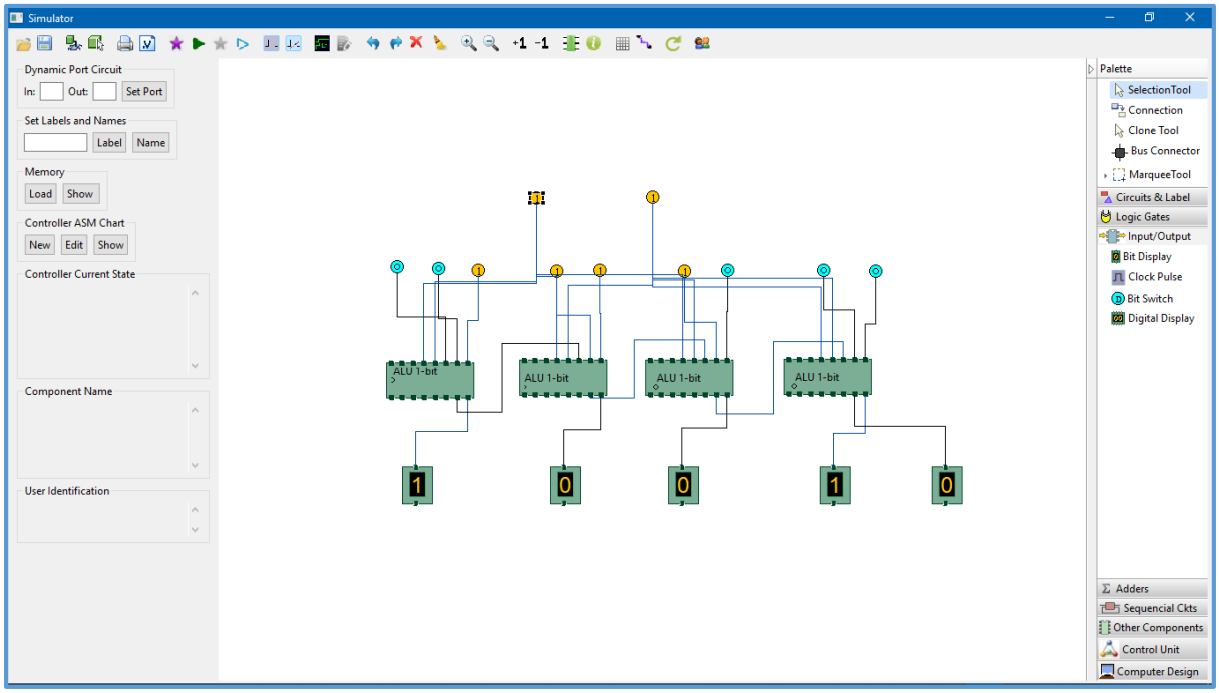


OR OPERATION:



AIM: To implement ALU design.

ADD OPERATION:



● **Conclusion:** In this lab we studied about ALU design and learned how to implement ALU design in VS lab with integrated circuit. And performed arithmetic Operations like addition, subtraction, division, multiplication and logical operations like AND, OR, X-OR, NAND, NOR etc.