Data Science

Experiment No. 10
Implement ALU design
Name: Vijendra Mane
Roll Number: 26
Date of Performance:
Date of Submission:

Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence and Data Science

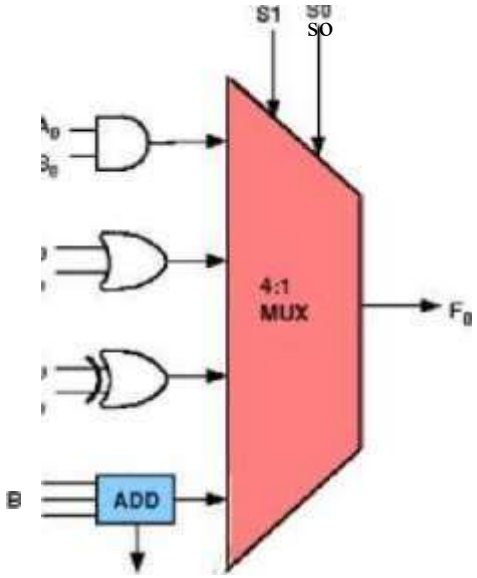
Aim: To implement ALU design

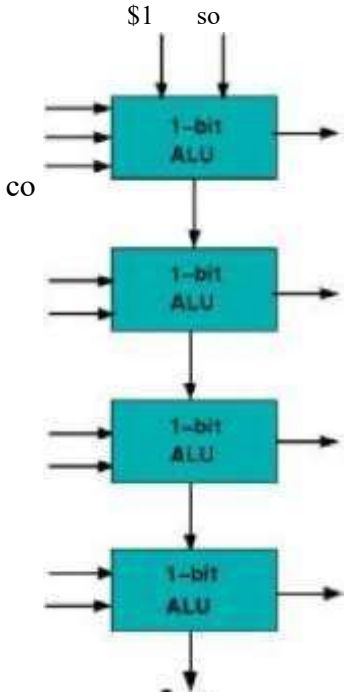
Objective : Objective of 4 bit arithmetic logic unit (with AND, OR, XOR, ADD operation):

1. To understand behaviour of arithmetic logic unit from working module.
2. To Design an arithmetic logic unit for given parameter.

Theory:

ALU or Arithmetic Logical Unit is a digital circuit to do arithmetic operations like addition, subtraction,division, multiplication and logical oparations like and, or, xor, nand, nor etc. A simple block diagram of a 4 bit ALU for operations and,or,xor and Add is shown here :





The 4-bit ALU block is combined using 4 1-bit ALU block Design Issues :

The circuit functionality of a I bit ALU is shown here, depending upon the control signal Sl and SO the circuit operates as follows:

for Control signal Sl 0 , SO — 0, the output is A And B, for
Control signal SI = O , SO = I, the output is A Or B, for
Control signal SI = I , SO = O, the output is A Xor B, for Control
signal SI = I , SO = I, 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				FOR ACTIVE HIGH OPERANDS	
INPUTS				LOGIC	ARITHMETIC (NOTE 2)
S1					
S3	S2	S1	S0	(M = H)	(M = L) (Cn=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

Procedure

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

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

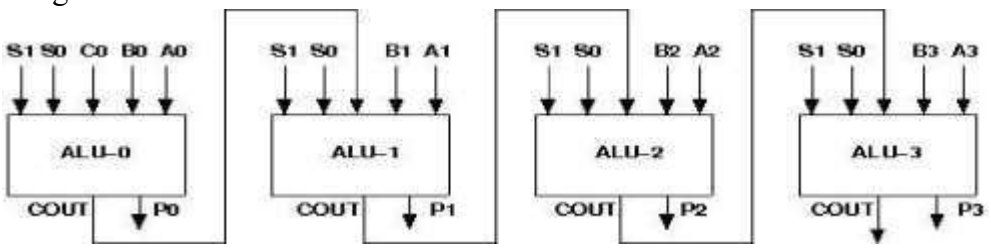
4) For I-bit ALU input AO is in pin-9,BO is in pin-10, CO is in pin-Il (this is input can-y), for selection of operation, SO is in pin-12, Sl is in pin-13, output F is in pin-8 and output carry is pin-7

5) Click on the I -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 I-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,ifit is not seen scroll down in the drawer), 3 digital display and I 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 pallete.

7) See the Output, in the screenshot diagram we have given the value Of Sl SO—I I which will perform add operation and two number input as AO Al A2 O and BO B I B2 133=0100 so get output FO Fl F2 F3=0110 as sum and O 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.

Circuit diagram of 4 bit ALU:

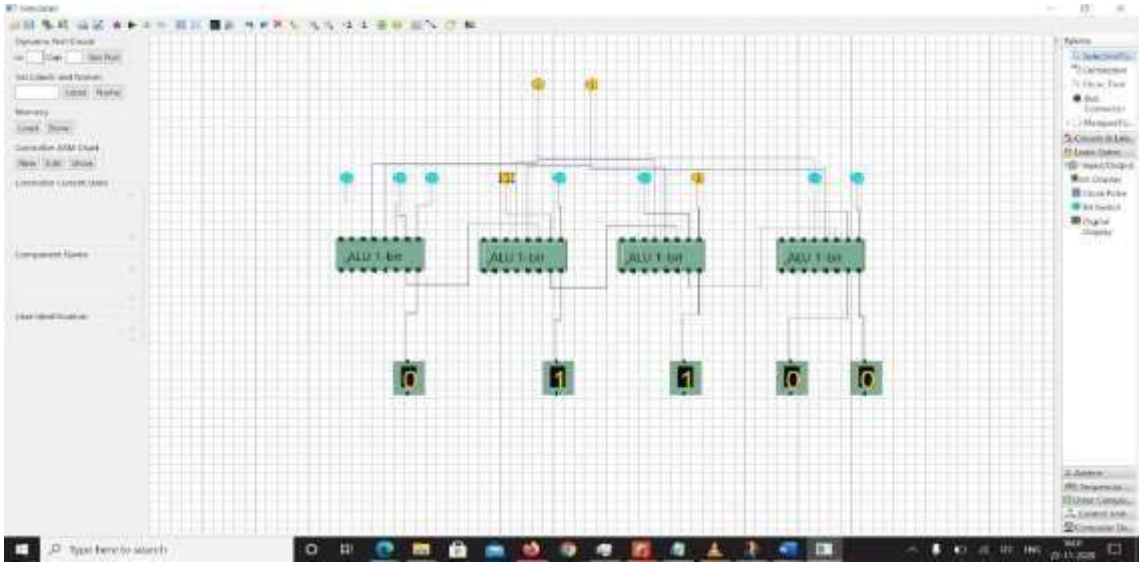


Components required :

To build any 4 bit ALU, we need :

- AND gate, OR gate, XOR gate
- Full Adder,
- 4-to-l MUX
- Wires to connect.

Screenshots of ALU design:



Conclusion:

The aim of the experiment is to design and implement an Arithmetic Logic Unit (ALU) in order to perform arithmetic and logic operations on binary data, with a focus on optimizing speed, efficiency, and versatility in digital computing systems.