Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

LAB REPORT

Course No: CSE 3110

Course Title: Digital System Design Lab

Experiment No: 01

Name of the Experiment: Designing a 4-bit ALU (Arithmetic and Logic Unit)

Submitted By-

Group No: 01

Section: A_1

Group Members:

20210204014 – Zunaed Aman

20210204004 – Mushfiqur Rahman

20210204011 – Rajiun Salihan Nabi

Introduction: An Arithmetic Logic Unit (ALU) is used to perform arithmetic such as addition, subtraction, multiplication, division and logic operations such as AND, OR etc. It represents the fundamental building block of the central processing unit (CPU) of a computer. In this experiment, we made a 4bit Arithmetic Logic Unit (ALU). So, we needed two selector bit S_0 and S_1. And for selecting arithmetic and logical operation, we used S_2. So, 3 selector bit were needed in total.

Problem Statement:

S_2	S_0	S_0	Output	Function
1	1	1	A_i +1	Increament A
1	0	1	A_i+B_i+1	Add with carry
1	1	0	A_i+B_i	Add
1	0	0	A_i +1+1	Transfer with carry
0	0	X	$A_i.B_i$	AND
0	1	X	A_i’	Complement A

Function Generation:

S_2	S_1	S_0	X_i	Y_i	Z	Output	Function
1	1	1	A_i	0	1	A_i +1	Increament A
1	0	1	A_i	B_i	1	A_i+B_i+1	Add with carry
1	1	0	A_i	B_i	0	A_i+B_i	Add
1	0	0	A_i	All 1	1	A_i +1+1	Transfer with carry
0	0	X	$A_i.B_i$	0	0	$A_i.B_i$	AND
0	1	X	A’	0	0	A_i’	Complement A

$\mathbf{S_2\ \ S_1\ \ S_0}$

$\mathbf{X} = A_i'\ S_2'S_1 + S_2'S_1'(A_i\,.\,B_i) + A_i\,S_2$

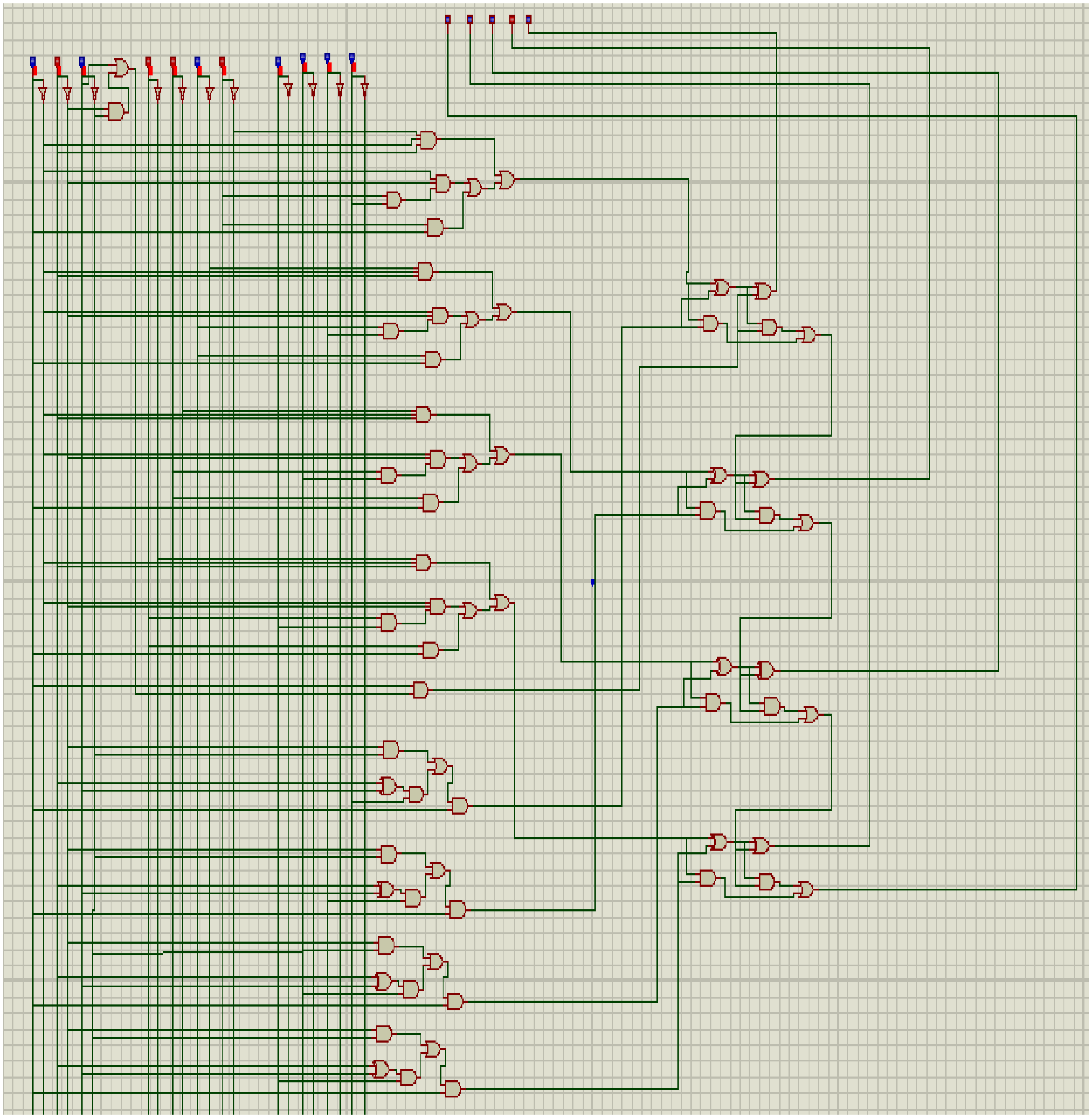
$$\begin{aligned}\mathbf{Y} &= (B_i + B_i\,')\ S_2\ S_1'S0' + B_i.S2(S1 \oplus S0)\\ &= S2.S1'.S0' + B_i.S2(S1 \oplus S0)\\ &= S2(\ B_i\,(S1 \oplus S0) + S1'.S0')\\ &= S2(\ B_i\,(S1 \oplus S0) + S1'.S0')\end{aligned}$$

$\mathbf{Z(C_{in}) =}\ S_2\,(S_0\ +\ S_1'S_0')$

Equipment and Budget:

Gate Name	IC	Amount	Price per IC(tk)	Price (tk)
AND	7408		20	
OR	7432		20	
NOT	7404		20	
XOR	4030		25	
				Total :

Simulation:



Result:

i. For Increament A Operation –

Input											**Output**				
S_2	S_1	S_0	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	C_{out}	F_3	F_2	F_1	F_0
1	1	1	1	0	1	0	0	0	0	0	0	1	0	1	1
			0	1	1	0	0	0	0	0	0	0	1	1	1
			1	1	1	0	0	0	0	0	0	1	1	1	1

ii. For Add with carry Operation-

Input											**Output**				
S_2	S_1	S_0	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	C_{out}	F_3	F_2	F_1	F_0
1	0	1	1	0	0	1	0	1	1	0	1	0	0	0	0
			0	1	1	0	0	1	0	0	0	1	0	1	1

iii. For Add Operation-

Input											**Output**				
S_2	S_1	S_0	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	C_{out}	F_3	F_2	F_1	F_0
1	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0
			1	1	0	0	0	1	0	1	1	0	0	0	1

iv. For Addition Operation-

Input											**Output**				
S_2	S_1	S_0	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	C_{out}	F_3	F_2	F_1	F_0
1	0	0	1	1	0	0	0	1	0	0	1	1	1	0	0
			1	0	0	0	0	1	0	1	1	1	0	0	0

v. For AND Operation-

Input											Output				
S_2	S_1	S_0	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	C_{out}	F_3	F_2	F_1	F_0
0	0	X	0	0	1	1	1	1	1	0	0	0	0	1	0
			1	0	0	1	1	0	1	1	0	1	0	0	1

vi. For Complement Operation-

Input											Output				
S_2	S_1	S_0	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	C_{out}	F_3	F_2	F_1	F_0
0	1	X	1	1	1	0	0	0	0	0	0	0	0	0	1
			0	1	0	1	0	0	0	0	0	1	0	1	0

Conclusion

To perform the ALU simulation we have used four 1 bit full adders using half adder. We solved these 4 bit ALU functions using formulas. The total cost for performing the simulation was reasonable. After completing circuit implementation, it worked perfectly and we faced no error.