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: 03

Section: B_2

Group Members:

18.01.04.077 Sabrina Mostafij Mumu
18.01.04.084 Safwan Muntasir
18.01.04.096 Purnendu Talukder
18.01.04.097 Arifur Rahman Jawad

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 4-bit 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
0	0	0	A_i	Transfer A
0	0	1	A_i-B_i	Subtraction
0	1	0	A_i-1	Decrement A
0	1	1	A_i+B_i	Add
1	0	X	$A_i.B_i$	AND
1	1	X	$A_i \wedge B_i$	OR

Function Generation:

S_2	S_1	S_0	C_{in}	X_i	Y_i	Output	Function
0	0	0	0	A_i	0	A_i	Transfer A
0	0	1	1	A_i	$\bar{B}_i$	A_i-B_i	Subtraction
0	1	0	0	A_i	All 1	A_i-1	Decrement A
0	1	1	0	A_i	B_i	A_i+B_i	Add
1	0	X	0	$A_i.B_i$	0	$A_i.B_i$	AND
1	1	X	0	A_i+B_i	0	$A_i \wedge B_i$	OR

$X_i=\overline{S}_2A_i+S_2\overline{S}_1(A_iB_i)+S_2S_1(A_i+B_i)$

$= \overline{S}_2A_i+S_2\overline{S}_1A_iB_i+S_2S_1A_i+S_2S_1B_i$

$= \overline{S}_2A_i+S_2A_i(\overline{S}_1B_i+S_1)+S_2S_1B_i$

$= \overline{S}_2A_i+S_2A_i(\overline{S}_1+S_1)(B_i +S_1)+S_2S_1B_i$

$= \overline{S}_2A_i+S_2A_iB_i+S_2S_1A_i+ S_2S_1B_i$

$= A_i(\overline{S}_2+S_2B_i)+S_2S_1A_i+ S_2S_1B_i$

$= A_i(\overline{S}_2+S_2)(\overline{S}_2+B_i)+S_2S_1A_i+ S_2S_1B_i$

$= \overline{S}_2A_i+A_iB_i +S_2S_1A_i+S_2S_1B_i$

$= A_i(\overline{S}_2+S_2S_1)+A_iB_i+S_2S_1B_i$

$= A_i(\overline{S}_2+S_2)(\overline{S}_2+S_1)+A_iB_i+S_2S_1B_i$

$= A_i\overline{S}_2+S_1A_i+A_iB_i+S_2S_1B_i$

$= A_i(\overline{S}_2+S_1+B_i)+S_2S_1B_i$

$Y_i= \overline{S}_2\overline{S}_1S_0\overline{B}i + \overline{S}_2S_1S_0Bi + \overline{S}_2S_1\overline{S}_0.1$

Now using K-map we get:

S_0B_i/ S_2S_1	$\overline{S}_0\overline{B}_i$	$\overline{S}_0B_i$		S_0B_i	$S_0\overline{B}_i$
$\overline{S}_2\overline{S}_1$					1
$\overline{S}_2S_1$	1	1		1	
S_2S_1					
$S_2\overline{S}_1$					

$$\text{So, } Y= \overline{S}_2S_1\overline{S}_0 + \overline{S}_2S_1B_i + \overline{S}_2\overline{S}_1S_0\overline{B}_i$$

$$= \overline{S}_2S_1(\overline{S}_0+B_i) + \overline{S}_2\overline{S}_1S_0\overline{B}_i$$

$$= \overline{S}_2S_1(\overline{S_0\overline{B}_i}) \qquad \Big[\text{using De-Morgan's Law}\Big]$$

$$= \overline{S}_2\Big[S_1(\overline{S_0\overline{B}i}) + \overline{S}_1(S_0\overline{B}i)\Big]$$

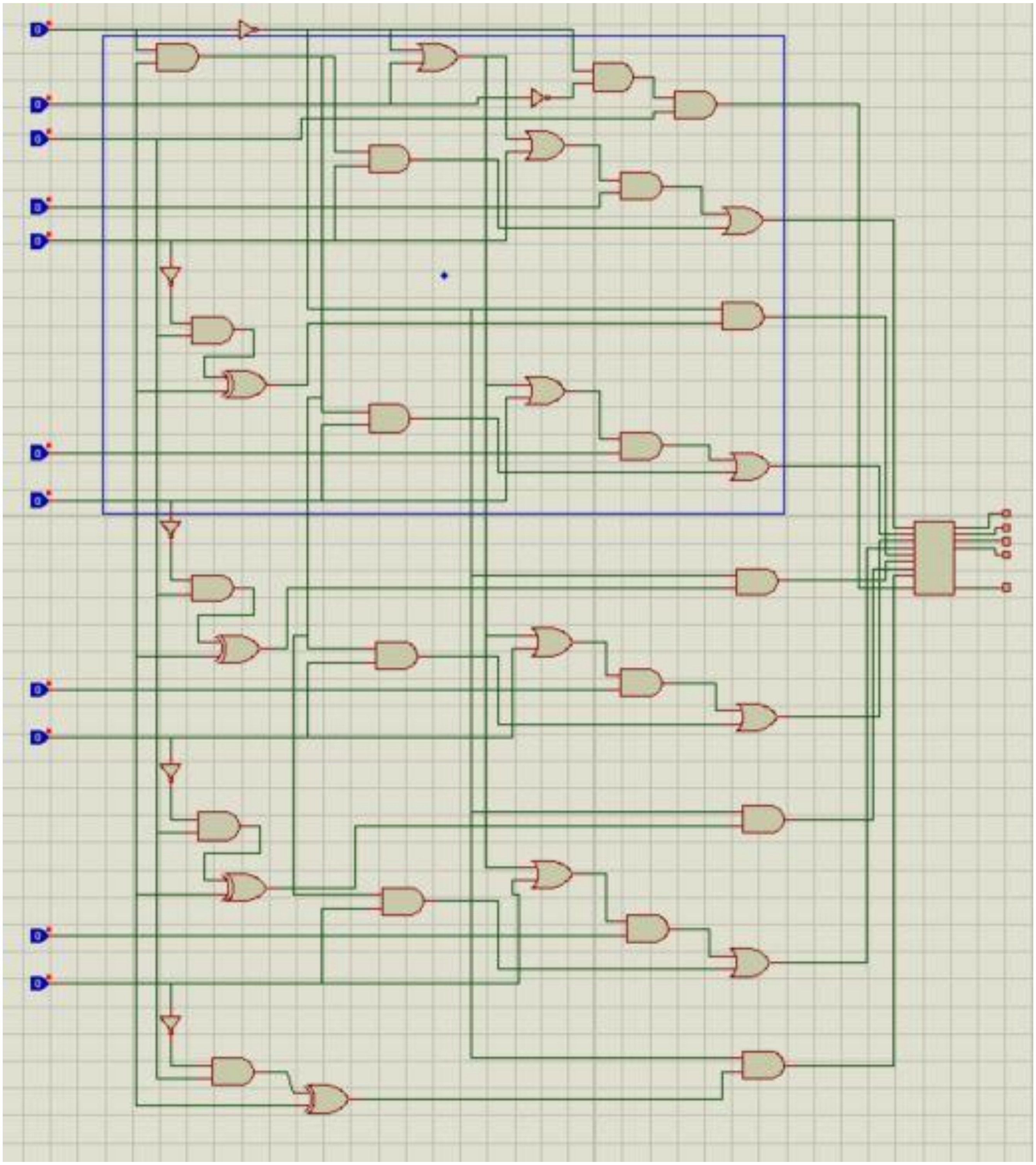
$$= \overline{S}_2\Big[S_1\oplus S_0Bi\Big]$$

$$C_{in} = \overline{S}_2\overline{S}_1S_0$$

Equipment and Budget:

Gate Name	IC	Amount	Price per IC(tk)	Price (tk)
AND	7408	5	30	150
OR	7432	2	29	58
NOT	7404	1	25	25
XOR	4030	1	26	26
4-bit full adder	4008	1	40	40
			Total	**299(tk)**

Simulation:


0
0
0
0
0
0
0
0
0
0
0
0

Result:

i. For Transfer 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
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			1	0	1	0	0	0	0	0	0	1	0	1	0
			1	1	1	1	0	0	0	0	0	1	1	1	1

ii. For Decrement 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	0	0	1	1	1	0	1	0	1	1	0	1	1	0
			1	1	0	1	0	0	1	0	1	1	1	0	0

iii. For Subtraction 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	1	1	1	1	1	1	1	0	1	1	0	0	1	0
			0	1	1	0	0	0	1	0	1	0	1	0	0

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
0	1	1	1	1	0	0	0	0	1	1	0	1	1	1	1
			1	0	1	0	0	1	0	0	0	1	1	1	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
1	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 OR 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	X	1	1	1	1	0	0	0	0	0	1	1	1	1
			0	1	0	1	1	0	0	0	0	1	1	0	1

Conclusion: To perform the ALU simulation we have used 4 bit full adder. We solved these 4 bit ALU functions using k-map. The total cost for performing the simulation was reasonable. After completing circuit implementation, it worked perfectly and we faced no error.