Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Experiment No : 01

Course No : CSE3110

Course Title : Digital System Design Lab

Submission Date : 27.12.2021

Submitted to : Ms. Mohsena Ashraf, Ms. Nusrat Jahan Lisa

Submitted by

Group No : 01

Section : A2

ID	Name
190104026	Tahsina Muthaki
190104027	Asif Iftekher Fahim
190104028	Safwan Ibne Masuk
190104037	Ahmadul Karim Chowdhury

Introduction:

Arithmatic Logic Unit (ALU) is a digital circuit which used to perform anithmetic operations like addition, subtraction and logical operations such as NOT, OR, AND, XOR etc. Most of the operations of a CPU are performed by ALU as it is a part of furdamental things of CPU or any kind of computing circuits. Most of those operations of a CPU are performed by one or more ALU's, which load data from input registers. The control unit tells the ALU what operation to perform and ALU stores the result in an output register. In this experiment, we made a 4 bit Arithmetic Logic Unit (ALU) with three (3) selection bits (S_2, S_1, S_0). In our experiment table, S_1 is the mode selection bit as given and S_0 & S_2 are there for selecting functions. According to the given table, we had to perform 4 anithmatic operations where $S_1 = 0$ and 2 logical operations where $S_1 = 1$.

Problem Statement:

S_2	S_1	S_0	Output	Function
0	0	0	Ai+1+1	Transfer A with Carry
0	0	1	Ai-Bi-1	Subtraction with Borrow
1	0	0	Ai-1	Decrement A
1	0	1	Ai+Bi	Add
0	1	X	Ai⊕Bi	XOR
1	1	X	Ai|Bi	OR

Table Generation:

S_2	S_1	S_0	X_i	Y_i	Z_{in}	Output
0	0	0	Ai	All 1	1	Ai+1+1
0	0	1	Ai	$\overline{Bi}$	0	Ai-Bi-1
1	0	0	Ai	All 1	0	Ai-1
1	0	1	Ai	Bi	0	Ai+Bi
0	1	X	Ai⊕Bi	0	0	Ai⊕Bi
1	1	X	Ai|Bi	0	0	Ai|Bi

$$X_i = S_2'S_1'S_0'A_i + S_2'S_1'S_0A_i + S_2S_1'S_0'A_i + S_2S_1'S_0A_i$$
$$+ S_2'S_1S_0'(A_i \oplus B_i) + S_2'S_1S_0(A_i \oplus B_i) + S_2S_1S_0'(A_i + B_i)$$
$$+ S_2S_1S_0(A_i + B_i)$$

$$= S_2'S_1'S_0'A_iB_i + S_2'S_1'S_0'A_iB_i' + S_2'S_1'S_0A_iB_i + S_2'S_1'S_0A_iB_i'$$
$$+ S_2S_1'S_0'A_iB_i + S_2S_1'S_0'A_iB_i' + S_2S_1'S_0A_iB_i + S_2S_1'S_0A_iB_i'$$
$$+ S_2'S_1S_0'A_iB_i' + S_2'S_1S_0'A_i'B_i + S_2'S_1S_0A_iB_i' + S_2'S_1S_0A_i'B_i$$
$$+ S_2S_1S_0'A_iB_i + S_2S_1S_0'A_i'B_i + S_2S_1S_0A_iB_i + S_2S_1S_0A_i'B_i$$

$$= S_2'S_1'S_0'A_iB_i + S_2'S_1'S_0'A_iB_i' + S_2'S_1'S_0A_iB_i + S_2'S_1'S_0A_iB_i'$$
$$+ S_2S_1'S_0'A_iB_i + S_2S_1'S_0'A_iB_i' + S_2S_1'S_0A_iB_i + S_2S_1'S_0A_iB_i'$$
$$+ S_2'S_1S_0'A_iB_i + S_2'S_1S_0'A_i'B_i + S_2'S_1S_0A_iB_i' + S_2'S_1S_0A_i'B_i$$
$$+ S_2S_1S_0'A_iB_i + S_2S_1S_0'A_iB_i' + S_2S_1S_0'A_i'B_i + S_2S_1S_0A_iB_i'$$
$$+ S_2S_1S_0A_i'B_i + S_2S_1S_0A_iB_i$$

$$= S_2'S_1'S_0'A_iB_i + S_2'S_1'S_0'A_iB_i' + S_2'S_1'S_0A_iB_i + S_2'S_1'S_0A_iB_i'$$
$$+ S_2S_1'S_0'A_iB_i + S_2S_1'S_0'A_iB_i' + S_2S_1'S_0A_iB_i + S_2S_1'S_0A_iB_i'$$
$$+ S_2'S_1S_0'A_iB_i + S_2'S_1S_0'A_i'B_i + S_2'S_1S_0A_iB_i' + S_2'S_1S_0A_i'B_i$$
$$+ S_2S_1S_0'A_iB_i + S_2S_1S_0'A_iB_i' + S_2S_1S_0'A_i'B_i + S_2S_1S_0A_iB_i'$$
$$+ S_2S_1S_0A_i'B_i + S_2S_1S_0A_iB_i$$

$$= \Sigma(2, 3, 6, 7, 9, 10, 13, 14, 18, 19, 22, 23,$$
$$25, 26, 27, 29, 30, 31)$$

K-map:

S_2S_1 \ $S_0A_iB_i$	$S_0'A_i'B_i'$	$S_0'A_i'B_i$	$S_0'A_iB_i$	$S_0'A_iB_i'$	$S_0A_iB_i'$	$S_0A_iB_i$	$S_0A_i'B_i$	$S_0A_i'B_i$
$S_2'S_1'$			1	1	1	1		
$S_2'S_1$		1		1	1		1	
S_2S_1		1	1	1	1	1	1	
S_2S_1'			1	1	1	1		

$$\therefore X_i' = S_2A_i + S_1'A_i + A_iB_i' + S_1A_i'B_i$$

$$= A_i(S_1' + B_i' + S_2) + S_1A_i'B_i$$

NOW,

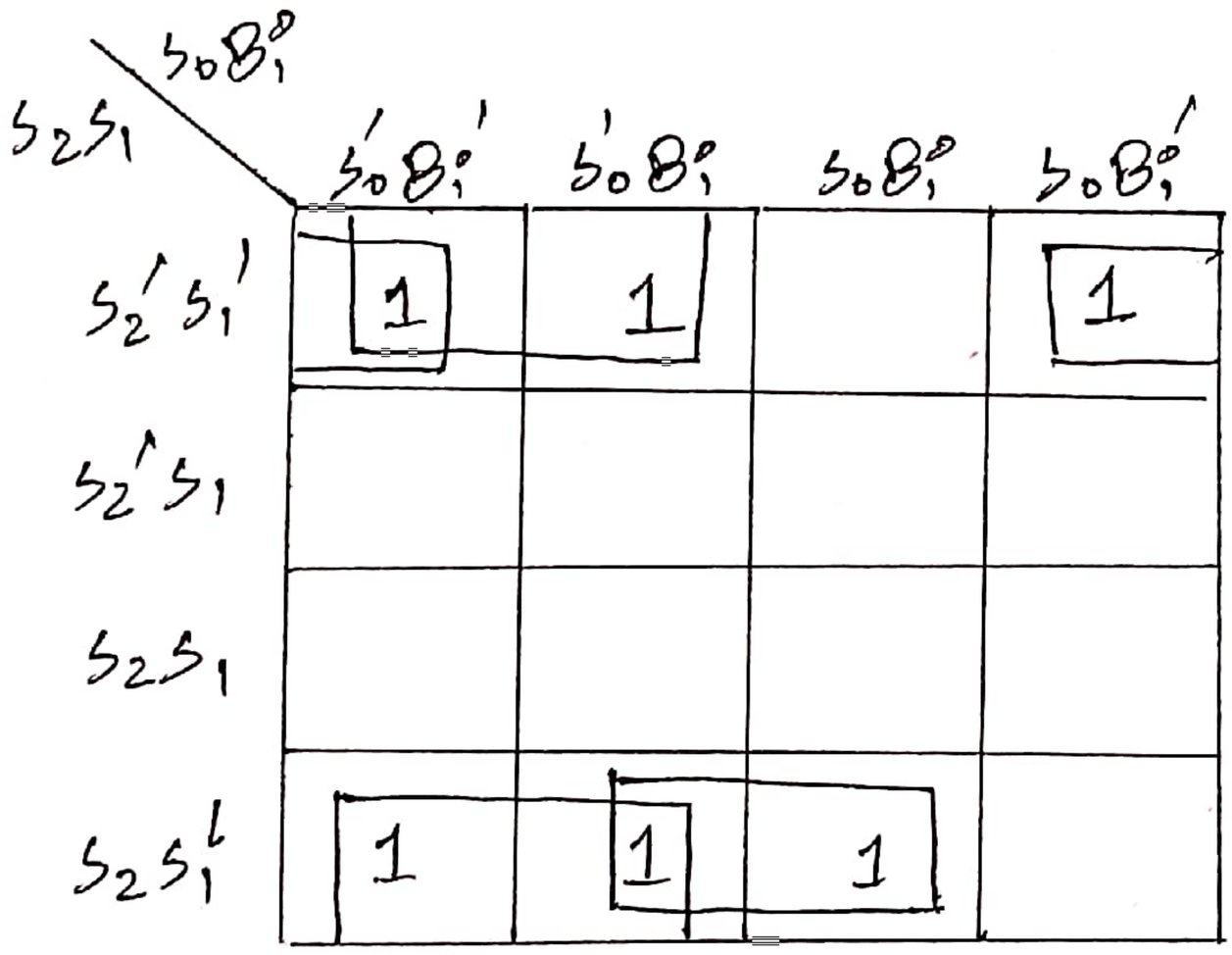
$$Y_i = S_1'S_0' + S_2'S_1'S_0B_i^{0'} + S_2S_1'S_0B$$

$$= S_1'S_0'(S_2 + S_2') + S_2'S_1'S_0B_i^{0'} + S_2S_1'S_0B_i^{0}$$

$$= S_1'S_0'S_2 + S_1'S_0'S_2' + S_2'S_1'S_0B_i^{0'} + S_2S_1'S_0B_i^{0}$$

$$= S_2S_1'S_0'B_i^{0} + S_2S_1'S_0'B_i^{0'} + S_2'S_1'S_0'B_i^{0} + S_2'S_1'S_0'B_i^{0'} + S_2'S_1'S_0B_i^{0} + S_2S_1'S_0B$$

$$= \Sigma(0,1,2,8,9,11)$$



From Kmap,

$$Y_i = S_1'S_0' + S_2'S_1'B_i^{0'} + S_2S_1'B_i^{0}$$

$$= S_1'(S_0' + S_2'B_i^{0'} + S_2B_i^{0})$$

$$\therefore Y_i = S_1'(S_0' + (S_2 \odot B_i^{0}))$$

And,

$Z = S_2' S_1' S_0' . 1$

Equipment and Budget:

Gate Name	Total Gate Use	IC Name	Total IC Use	Per Unit Cost	Price Total
AND	26	7408	7	13	91
OR	17	7432	5	13	65
NOT	15	7404	3	13	39
XOR	12	7486	3	13	39
				Total Cost	234 Taka

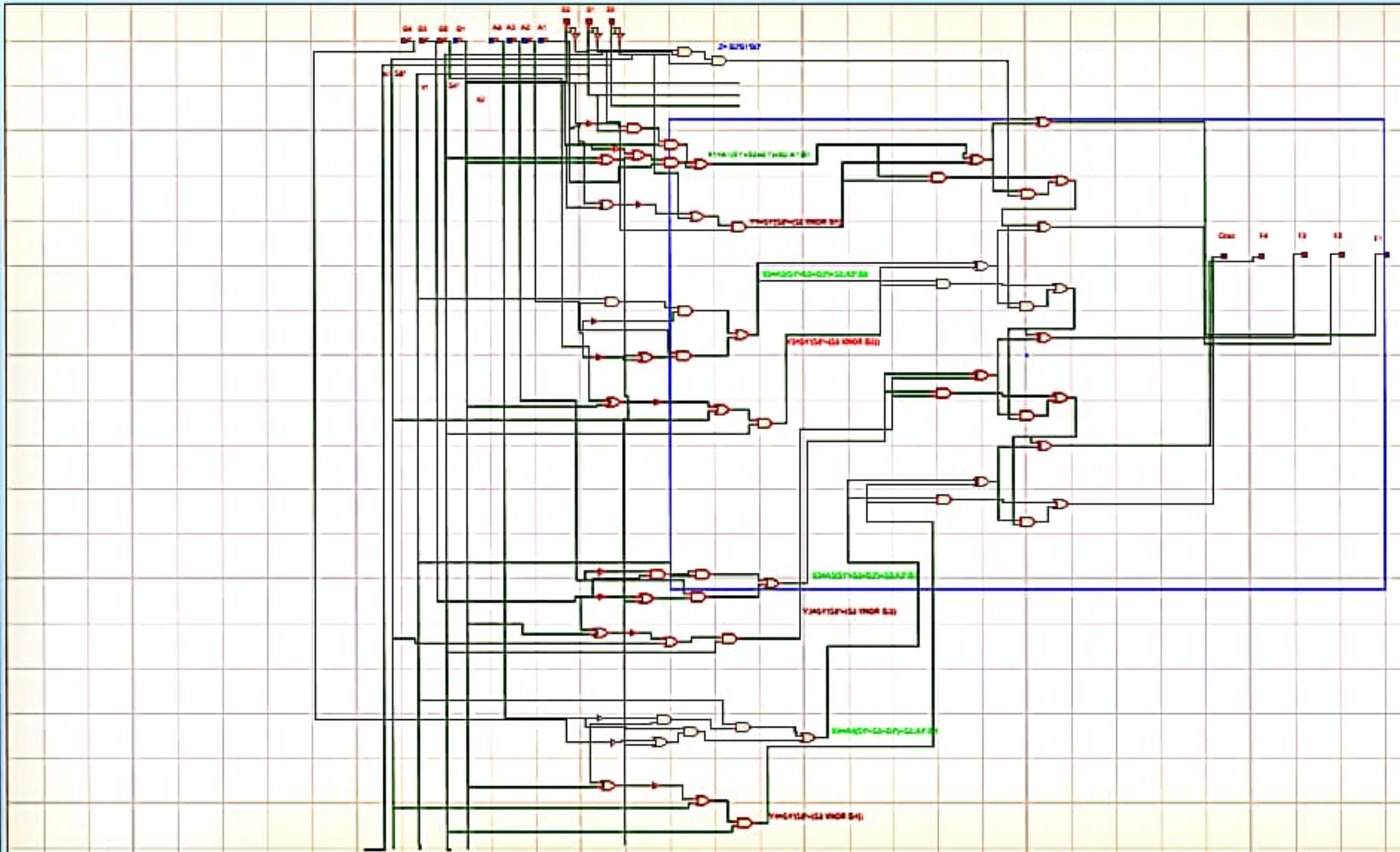
A2_G1 - Proteus 8 Professional - Schematic Capture

File Edit View Tool Design Graph Debug Library Template System Help

Schematic Capture

DEVICES

AND
AND_3
LOGICPROBE (BIG)
LOGICSTATE
NOT
OR
XOR



No Messages

Root sheet 1

-29500.0

-7400.0

Type here to search

76°F Haze

8:05 PM
12/26/2021

Result:

Operation	Input			Output	
	$S_2 S_1 S_0$	$A_3 A_2 A_1 A_0$	$B_3 B_2 B_1 B_0$	Cout	$F_3 F_2 F_1 F_0$
Ai+1+1 Transfer A with carry	0 0 0	1 1 0 1	1 0 1 0	1	1 1 0 1
Ai-Bi-1 Subtraction with Borrow	0 0 1	1 0 0 1	1 0 0 0	1	0 0 0 0
Ai-1 Decrement A	1 0 0	1 0 0 0	1 1 0 0	1	0 1 1 1
Ai+Bi Add	1 0 1	0 1 0 0	0 0 1 0	0	0 1 1 0
Ai Xor Bi XOR	0 1 X	1 1 1 0	0 1 0 0	0	1 0 1 0
Ai | Bi OR	1 1 X	1 0 1 0	0 1 0 1	0	1 1 1 1

Conclusion: In this experiment we have designed a four bit ALU which performs arithmetic and logical operations. Here we have used simplified equations by the help of k map simplification., we have carefully implemented the equations in the proteus software with the help of AND, OR, XOR and NOT gate. For the 4-bit adder we have constructed 4 individual 1-bit adder using XOR, AND and OR gate. Different input combinations have been tested in our constructed 4-bit ALU in software (Proteus) and corresponding output is correct each time.