Introduction: We know that An arithmatic Logic Unit (ALU) is used to perform arithmatic such as addition, substraction, 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 have made a 4 bit Arithmatic Logic Unit (ALU). Thus, for selection arithmatic and Logical operation, we need s_2 so three selector bit were needed in total.

Problem statement:

s_2	s_1	s_0	output	Function
1	1	1	Ai+1	Increment A
0	1	1	Ai−Bi	subtract
1	1	0	Ai+Bi	Add
0	1	0	Ai+1+1	Transfer A with carry
1	0	x	Ai + Bi	OR
0	0	x	Ai'	Complement A

Function Generation:

s_2	s_1	s_0	Z	X	Y	Output	Function
1	1	1	1	A_i	0	A_i+1	Increment A
0	1	1	1	A_i	$\overline{B_i}$	A_i-B_i	subtract
1	1	0	0	A_i	B_i	A_i+B_i	Add
0	1	0	1	A_i	All 1	A_i+1+1	Transfer A with carry
1	0	x	x	$A_i \mid B_i$	0	$A_i \mid B_i$	OR
0	0	x	x	A_i'	0	A_i'	complement A

Function simplification Using K-map:

For X:

$X = s_2 s_1 s_0 A_i + s_2' s_1 s_0 A_i + s_2 s_1 s_0' A_i + s_2' s_1 s_0' A_i +$
$s_2 s_1'(A_i+B_i) + s_2' s_1' A_i'$

$= s_2 s_1 s_0 A_i(B_i+B_i') + s_2' s_1 s_0 A_i(B_i+B_i') +$
$s_2' s_1 s_0'(B_i+B_i') + s_2' s_1 A_i (s_0+s_0')(B_i+B_i')$
$+ s_2 s_1' B_i (s_0+s_0')(A_i+A_i') + s_2' s_1' A_i'(s_0+s_0')(B_i+B_i')$

$= s_2 s_1 s_0 A_i B_i + s_2 s_1 s_0 A_i B_i' + s_2' s_1 s_0 A_i B_i +$
$s_2' s_1 s_0 A_i B_i' + s_2 s_1 s_0' A_i B_i + s_2 s_1 s_0' A_i B_i'$
$+ s_2' s_1 s_0' A_i B_i + s_2' s_1 s_0' A_i B_i' + s_2 s_1' A_i (s_0 B_i +$
$s_0' B_i + s_0 B_i' + s_0' B_i') + s_0 s_1' B_i (s_0 A_i + s_0 A_i' + s_0' A_i$
$+ s_0' A_i) + s_2' s_1' A_i' (s_0 B_i +$
$s_0 B_i' + s_0' B_i + s_0' B_i')$

$= S_2 S_1 S_0 A_i B_i + S_2 S_1 S_0 A_i B_i' + S_2' S_1 S_0 A_i B_i +$

$S_2' S_1 S_0 A_i B_i' + S_2 S_1 S_0' A_i B_i + S_2 S_1 S_0' A_i B_i' +$

$S_2' S_1 S_0' A_i B_i + S_2' S_1 S_0 A_i B_i' + S_2 S_1' S_0 A_i B_i +$

$S_2 S_1' S_0' A_i B_i + S_2 S_1' S_0 A_i B_i' + S_2 S_1' S_0' A_i B_i' +$

$S_2 S_1' S_0 A_i' B_i + S_2' S_1' S_0' A_i B_i + S_2' S_1' A_i' B_i' +$

$S_2 S_1' S_0' A_i' B_i + S_2' S_1' S_0 A_i' B_i' + S_2 S_1 S_0' A_i' B_i$

$= \Sigma(0,1,4,5,10,11,14,15,17,18,19,21,22,23,26,27,30,31)$

$S_0A_iB_i$ \ S_2S_1	$S_0'A_i'B_i'$	$S_0'A_i'B_i$	$S_0'A_iB_i$	$S_0'A_iB_i'$	$S_0A_iB_i'$	$S_0A_iB_i$	$S_0A_i'B_i$	$S_0A_i'B_i'$
$S_2'S_1'$	1	1	0	0	0	0	1	1
$S_2'S_1$	0	0	1	1	1	1	0	0
S_2S_1	0	0	1	1	1	1	0	0
S_2S_1'	0	1	1	1	1	1	1	0

$x = S_1 A_i + S_2 A_i + S_2' S_1' A_i' + S_2' S_0' A_i' B_i + S_1' S_0 A_i' B_i$

$= S_1 A_i + S_2 A_i + S_2' S_1' A_i' + S_1' A_i' B_i (S_0 + S_0')$

$= S_1 A_i + S_2 A_i + S_2' S_1' A_i' + S_1' A_i' B_i$

For Y:

$Y = S_2' S_1 S_0 Bi' + S_2 S_1 S_0' Bi + S_2' S_1 S_0'$

$= S_2' S_1 S_0 Bi' + S_2 S_1 S_0' Bi + S_2' S_1 S_0' (Bi + Bi')$

$= S_2' S_1 S_0 Bi' + S_2 S_1 S_0' Bi + S_2' S_1 S_0' Bi +$

$S_2' S_1 S_0' Bi'$

$= \Sigma (4, 5, 6, 13)$

$S_0 B$ / $S_2 S_1$	$S_0' Bi'$	$S_0 Bi$	$S_0 Bi$	$S_0 Bi'$
$S_2' S_1'$	0	0	0	0
$S_2' S_1$	1	1	0	1
$S_2 S_1$	0	1	0	0
$S_2 S_1'$	0	0	0	0

$Y = S_1 S_0' B + S_2' S_1 B'$

For Z:

$Z = S_2 S_1 S_0 + S_2' S_1 S_0 + S_2' S_1 S_0'$

$S_1 S_0$ / S_2	$S_1' S_0'$	$S_1' S_0$	$S_1 S_0$	$S_1 S_0'$
S_2'	0	0	1	1
S_2	0	0	1	0

$Z = S_1 S_0 + S_1 S_2' = S_1 (S_0 + S_2')$

Equipment and Budget:

Gate name	IC	Amount	Price ·Per IC(TK)	Price(TK)
NOT Gate	7404	1	25 Tk	25 Tk
OR Gate	7432	3	28 Tk	84 Tk
AND Gate	7408	5	32 Tk	160 Tk
XOR Gate	7486	1	25 Tk	25 Tk
4bit Full Adder	74LS89	1	40 Tk	40 Tk

Total - 334/-

Simulation:

Result :

Input												Output				
S_2	S_1	S_0	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	Operation Name	Cout	F_3	F_2	F_1	F_0
1	1	1	0	1	1	0	0	0	0	1	Increment A	0	0	1	1	1
			0	0	1	0	0	0	0	1		0	0	0	1	1
0	1	1	0	1	1	0	0	0	1	1	subtract	1	0	0	1	1
			0	1	1	0	0	0	0	1		1	0	1	0	0
1	1	0	0	1	1	0	0	0	0	1	Add	0	0	1	1	1
			0	1	1	1	0	0	0	1		0	1	0	0	0
0	1	0	0	1	1	0	0	0	0	1	Transfer A with carry	1	0	1	1	0
			0	0	0	1	0	0	0	1		1	0	0	0	1
1	0	X	0	1	1	0	0	0	0	1	OR	0	0	1	1	1
			0	0	1	0	0	0	1	1		0	0	0	1	1
0	0	X	0	1	1	0	0	0	0	0	complement A	0	1	0	0	1
			1	1	1	0	0	0	0	0		0	0	0	0	1

Conclusion: In this above experiment we have implement an ALU which can operate an Arithmatic and Logical operations. At first, the equations of Z, X and Y were made by using the given table. while we simplified it using K-map. Though we faced some problem while implementing on proteus as we made 1 bit adder for our own ALU design. After that the ALU were tested by different combination of bits and it provided the correct result.