BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Course No : CSE306
Course Name : Computer Architecture Sessional
Name of the Experiment : 4-bit ALU Simulation
Level/Term : 3-1
Section : B2

Group No : 02
Student ID : 1705092
1705111
1705116
1705118
1705121

Date of Submission : 28-03-2021

Introduction

An Arithmetic Logic unit or commonly known as ALU is a combinational digital circuit that performs arithmetic and logic operations on binary numbers. It is a simple binary calculator. It is a fundamental building block of many types of computing circuits, including the central processing unit (CPU) of computers. In this experiment, we have to design a 4-bit ALU according to design specification.

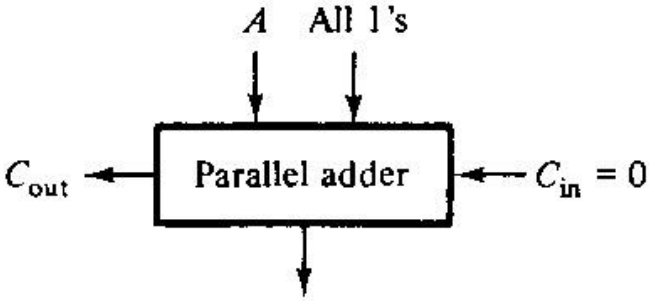
Problem Specification:

CS_2	CS_1	C_{in} CS_0	Function
0	*0*	*0*	Decrement A (A-1)
0	*0*	*1*	Transfer A (A)
0	*1*	*0*	Subtract with borrow (A-B-1 = A+B')
0	*1*	*1*	Subtract (A-B = A+B'+1)
1	*0*	*x*	AND (A^B)
1	*1*	*x*	Complement A (A')

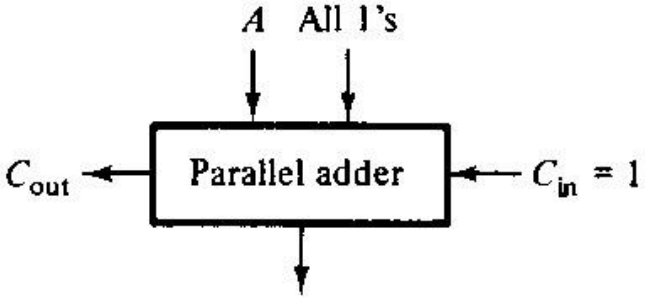
Arithmetic Part

When $CS_2 = 0$ Arithmetic part will be activated.

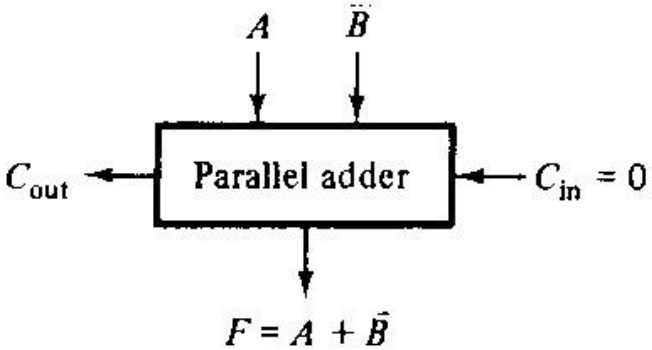
i. Decrement A (A-1) :



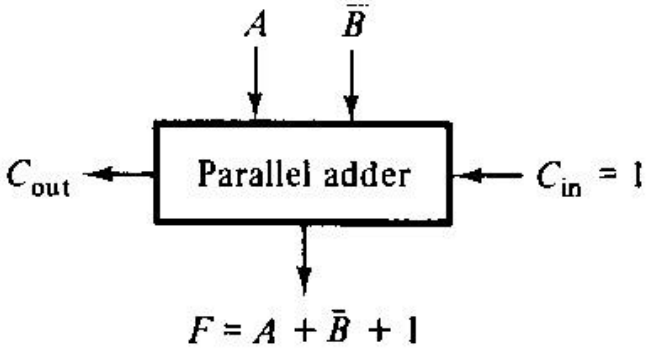
ii. Transfer A (A) :



iii. Subtract with borrow (A-B-1 = A+B') :



iv. Subtract (A-B = A+B'+1) :



Subtraction

Logic Part

When $CS_2 = 1$ Logic part will be activated. And if we look at our previous available arithmetic circuits, forcing $C_{in} = 0$ from the full adder output we get, $sum = x \oplus y$

For logic operation $A \wedge B$,
Let us assume $(A_i \vee K_i)$ in X_i and B_i' in Y_i.

$$Function = (A + K) \oplus B'$$
$$= (A + K).B + (A + K)'.B'$$
$$= AB + KB + A'K'B'$$

$$Now, AB = AB + KB + A'K'B'$$
$$\Rightarrow KB + A'K'B' = 0$$
$$\Rightarrow KB = 0$$
$$\therefore K = \mathrm{B}'$$

$$and, A'K'B' = 0$$
$$\Rightarrow K'B' = 0$$
$$\therefore K = B'$$

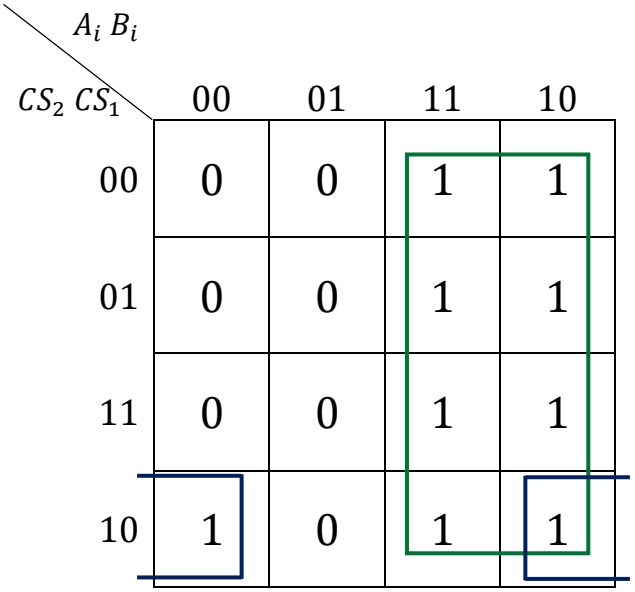
Now if we construct our Function Table,

Function Table :

CS_2	CS_1	C_{in} CS_0	X_i	Y_i	Function
0	0	0	A_i	1	A-1
0	0	1	A_i	1	A
0	1	0	A_i	B_i'	A+B'
0	1	1	A_i	B_i'	A+B'+1
1	0	x	$A_i \vee B_i'$	B_i'	A^B
1	1	x	A_i	1	A'

X_i:

K-Map:



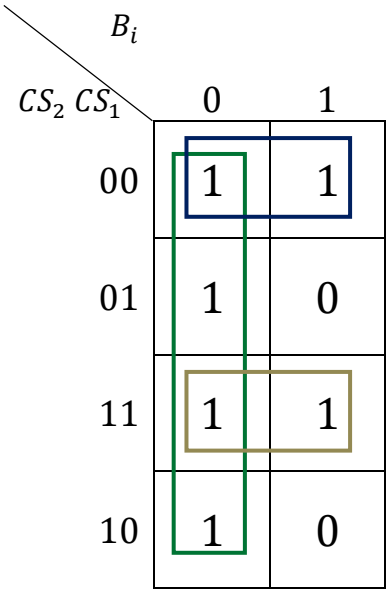
$$X_i = A + CS_2.CS_1'.B'$$

Y_i:

Truth table:

CS_2	CS_1	Y_i
0	0	1
0	1	B_i'
1	0	B_i'
1	1	1

K-Map:



$Y_i = B_i' + CS_2'.CS_1' + CS_2.CS_1$

$\therefore Y_i = B_i' + (CS_2 \oplus CS_1)'$

C_{in}:

When $CS_2 = 1$, CS_0 is don't care (x). We force 0 in C_{in} for logic operation. So,

K-Map:

$CS_2\ CS_1$ \ CS_0/C_{out}	0	1
00	0	1
01	0	1
11	0	0
10	0	0

$C_{in_1} = CS_2'.CS_0$; for the first full adder.

$C_{in_i} = CS_2'.C_{out_{i-1}}$; for the rest three full adders.

Flags

As we use forced C_{in} for our logic / bitwise operation, the C_{out} for flag output calculation is also considered as, $C_{out}\ .CS_2'$. The Flags are designed accordingly to that.

$CF = C_{out_4}$

$SF = F_4$

$ZF = (F_1 + F_2 + F_3 + F_4)'$

$VF = C_{out_4} \oplus C_{out_3}$

Block Diagram

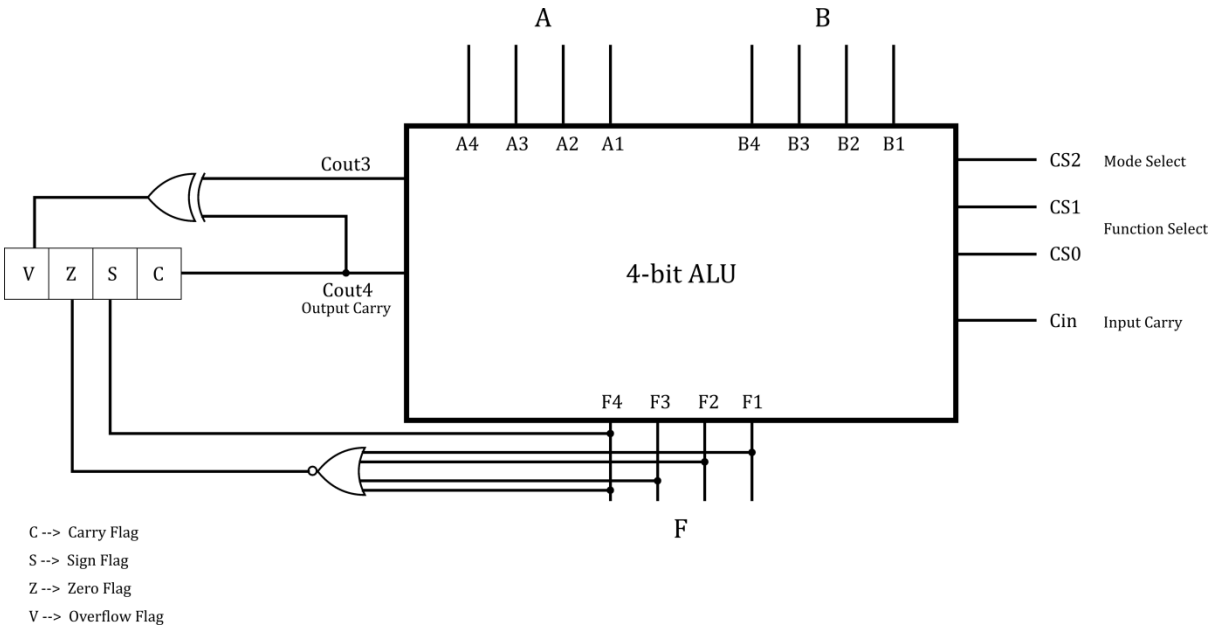


Figure : Block diagram of a 4-bit ALU

Complete Circuit Diagram

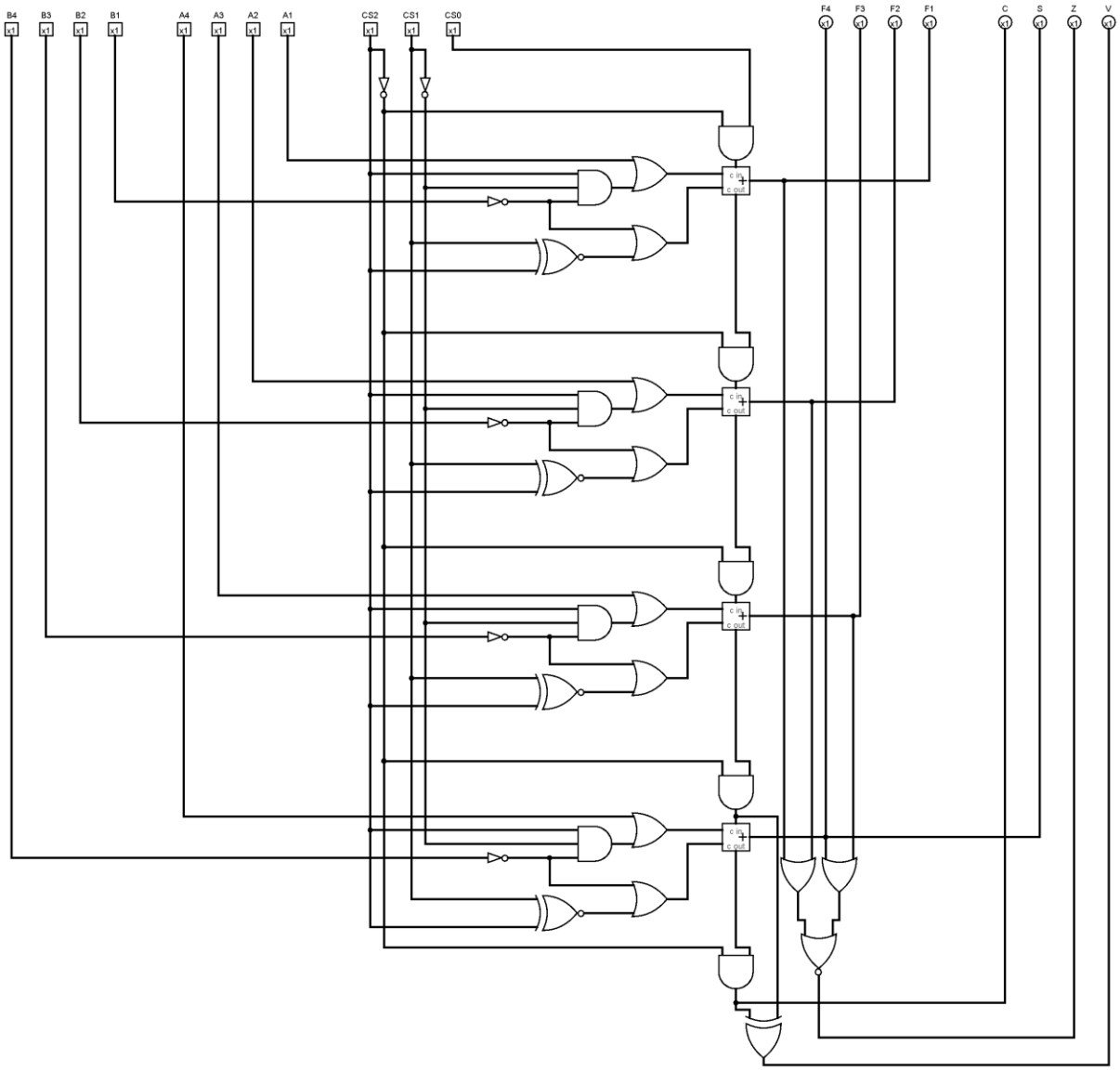


Figure : Circuit diagram of the 4-bit ALU

ICs used with count

Components	Count of Gates	IC Number	Count of ICs
NOT	6	IC74LS04	1
AND	13	IC74LS08	4
OR	10	IC74LS32	3
NOR	1	IC74LS02	1
XOR	1	IC74LS86	1
XNOR	4	IC4077	1
Adder	-	IC7483	4

Simulator used

The simulation software used in this 4-bit ALU simulation experiment is ***logisim-win-2.7.1.exe***

Discussion

In this experiment, we used the simulation software *logisim* to simulate a 4-bit Arithmetic Logic Unit.

As per specification when the mode selection bit CS_2 is 1 the ALU operates as logic unit and the selection bit CS_0 (C_{in}) is don't care. In our circuit design we forced C_{in} to be 0 to work in the logic operations.

And for the Flags to show accordingly, the carry out of each adder is considered to be the AND of CS_2' and C_{out}.

We calculated the input signals of the adders to simplify the equations and minimize the IC count.