Course No: CSE 306

Assignment on 4bit ALU Design and Simulation

Section: B2

~~Grop~~

Group No: 5

Student IDs: 1705096
1705095
1705097
1705102
1705104

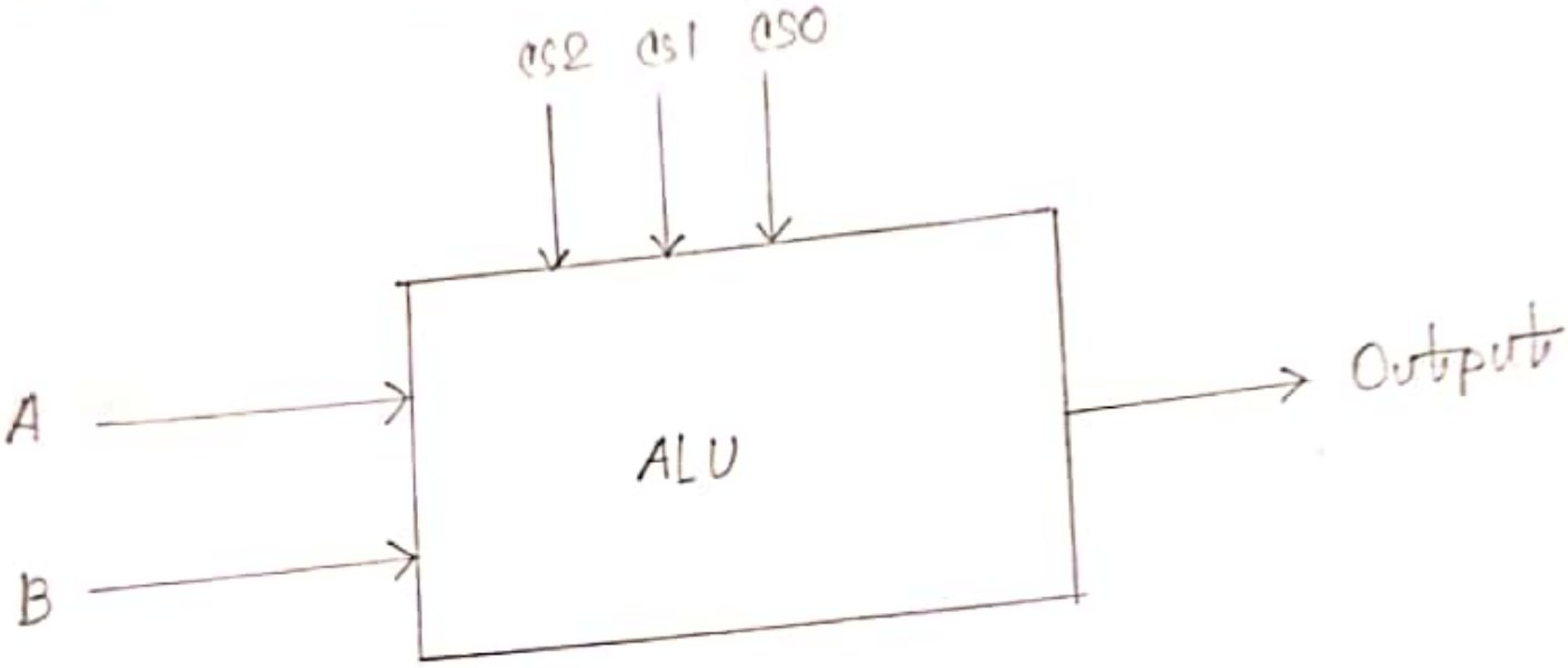
Date of Submission: 26-3-2021

Introduction:

An arithmetic logic unit is a multi-operation, combinatorial logic digital function. It can perform a set of basic arithmetic operations and a set of logic operations. The ALU has a number of selection lines to select a particular operation in the unit. The selection lines are decoded within the ALU so that k selection variables can specify up to 2^k distinct operations; Here we design 4 bit ALU with 3 selection lines.

Problem Specification with assigned ~~int~~ instructions:

Here we are going to design a 4 bit ALU with 3 selection variables that can simulate the following design:



cs2	cs1	cs0	Output
0	0	0	A+B
0	1	0	A+B+1
0	x	1	A ⊕ B
1	0	0	A
1	1	0	A+1
1	x	1	AB

Truth Table & Required k-maps :

Function Table :

CSO	CS2	CS1	X_i	Y_i	Z_1	z_i $(i\neq 1)$	Outputs
0	0	0	A_i	B_i	0	C_i	$A+B$
0	0	1	A_i	B_i	1	C_i	$A+B+1$
0	1	0	A_i	0	0	C_i	A
0	1	1	A_i	0	1	C_i	$A+1$
1	0	X	A_i	B_i	0	0	$A \oplus B$
1	1	X	A_i+B_i'	B_i'	0	0	AB

We have, $x_i = PA_i + Q(A_i+B_i') = (P+Q)A_i + QB_i'$

$Y_i = RB_i + SB_i'$

$z_i = TC_i$

where P, Q, R, S, T is function of selectors for which $P=1$ when $x_i = A_i$, $Q=1$ when $x_i' = A_i + B_i'$, $R=1$ when $Y_i = B_i$, $S=1$ when $Y_i = B_i''$, $T=1$ when $z_i = C_i$

Truth Table:

cs0	cs2	cs1	P	Q	R	S	T	P+Q	Z_1
0	0	0	1	0	1	0	1	1	0
0	0	1	1	0	1	0	1	1	1
0	1	0	1	0	0	0	1	1	0
0	1	1	1	0	0	0	1	1	1
1	0	x	1	0	1	0	0	1	0
1	1	X	0	1	0	1	0	1	0

K-maps:

cs0 \ cs2 cs1	00	01	11	10
0	1	1	1	1
1	1	1	1	1

K-map for P+Q

cs0 \ cs2 cs1	00	01	11	10
0	0	0	0	0
1	0	0	1	1

K-map for Q

cs0 \ cs2 cs1	00	01	11	10
0	1	1	0	0
1	1	1	0	0

k-map for R

cs0 \ cs2 cs1	00	01	11	10
0	0	0	0	0
1	0	0	1	1

k-map for S

cs0 \ cs2 cs1	00	01	11	10
0	1	1	1	1
1	0	0	0	0

k-map for T

cs0 \ cs2 cs1	00	01	11	10
0	0	1	1	0
1	0	0	0	1

k-map for Z_1

Equation:

From k-maps, we have,

$P+Q=1$, $Q = cs2\,cs0$, $R = cs2'$, $S = cs2\,cs0$

$T = cs0'$, $Z_1 = cs1\,cs0'$

So that, $X_i = A_i + cs_2\,cs_0\,B_i'$, $i=1,2,3,4$

$Y_i = cs_2'B_i + cs_2\,cs_0\,B_i'$, $i=1,2,3,4$

$Z_i = cs_0'\,C_i$, $i=2,3,4$

$Z_1 = cs_1\,cs_0'$

For flags we have,

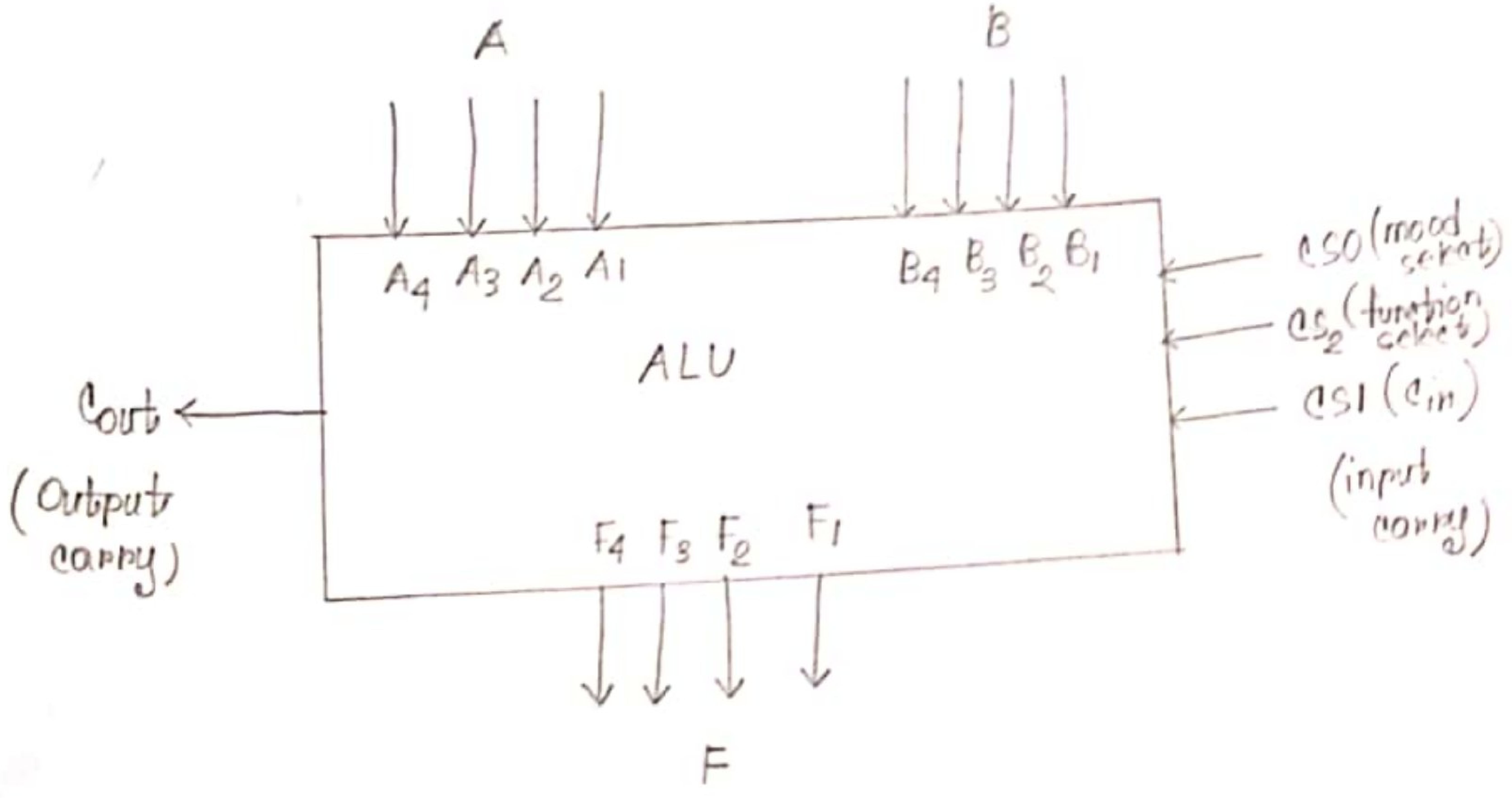
$V = C_{out} \oplus C_4$

$C = C_{out}$

$S = F_4$

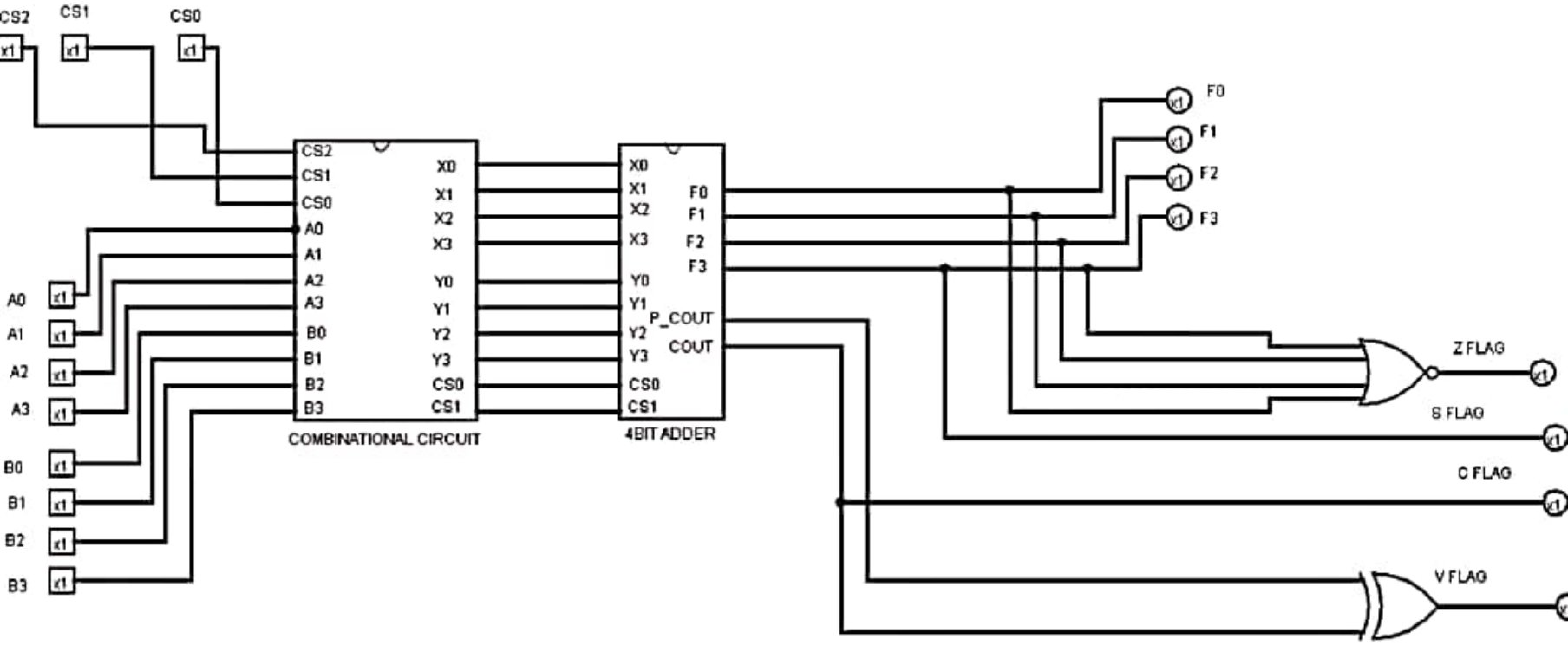
$Z = (F_1 + F_2 + F_3 + F_4)'$

Block Diagram:

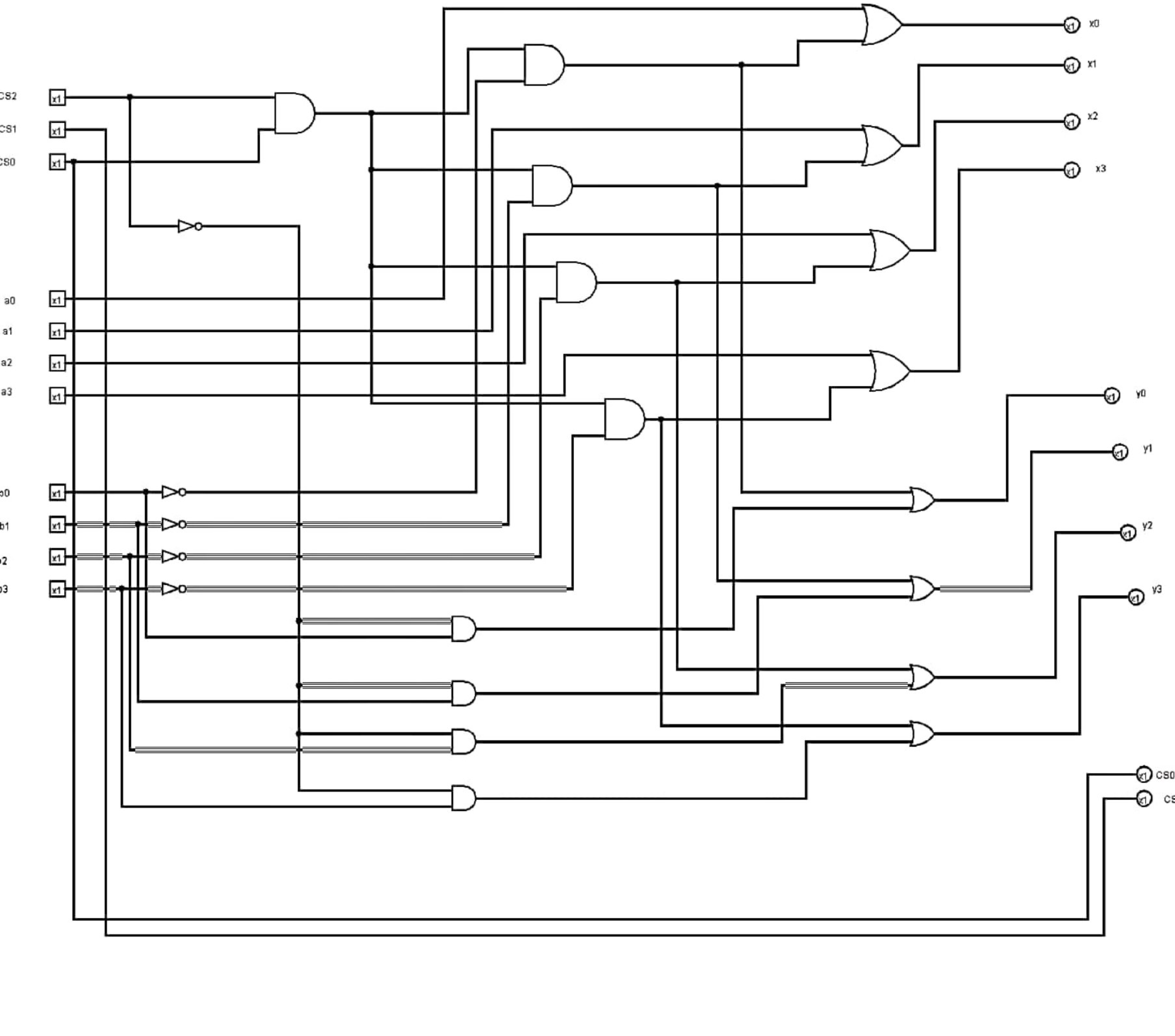


Block Diagram of 4 bit ALU

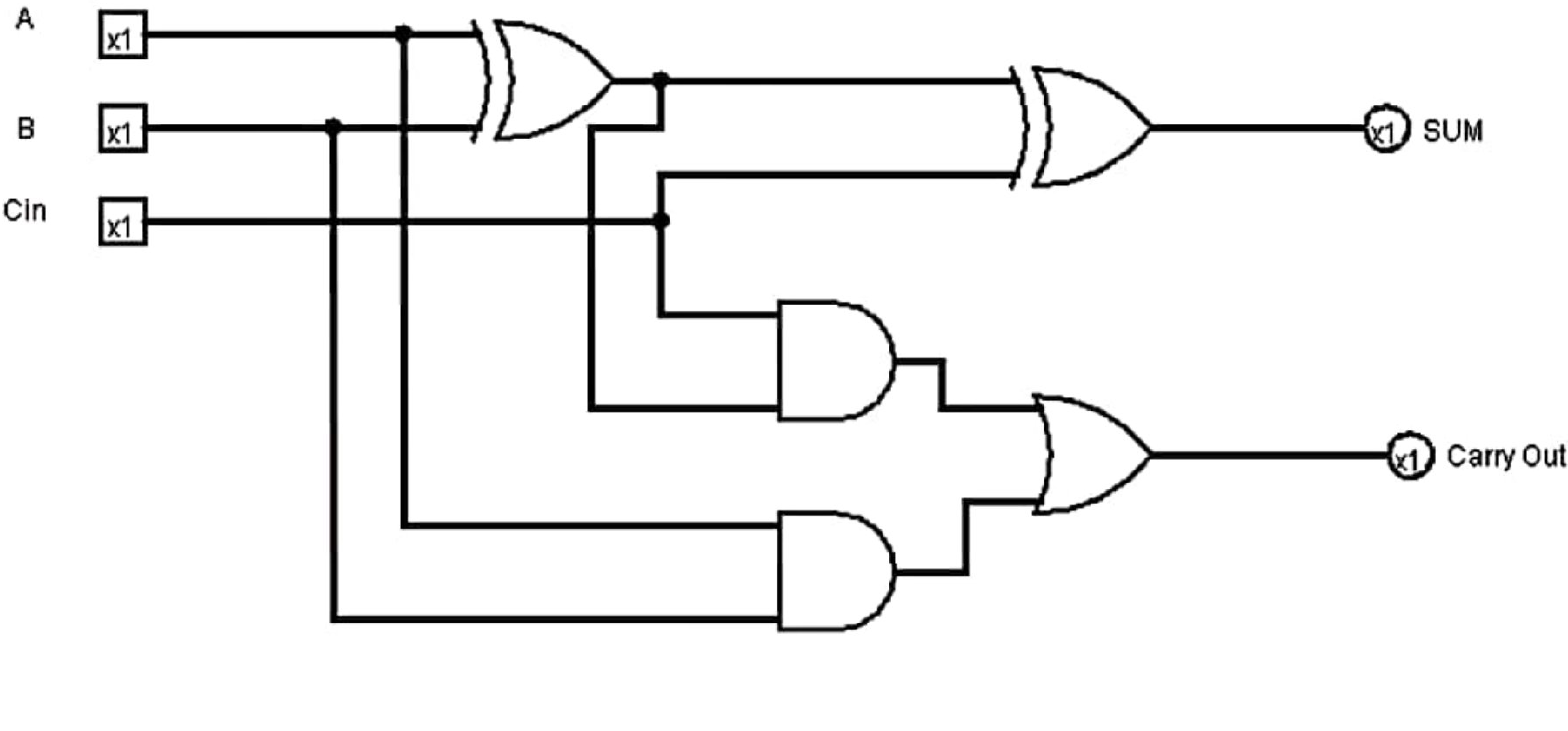
Circuit Diagram:



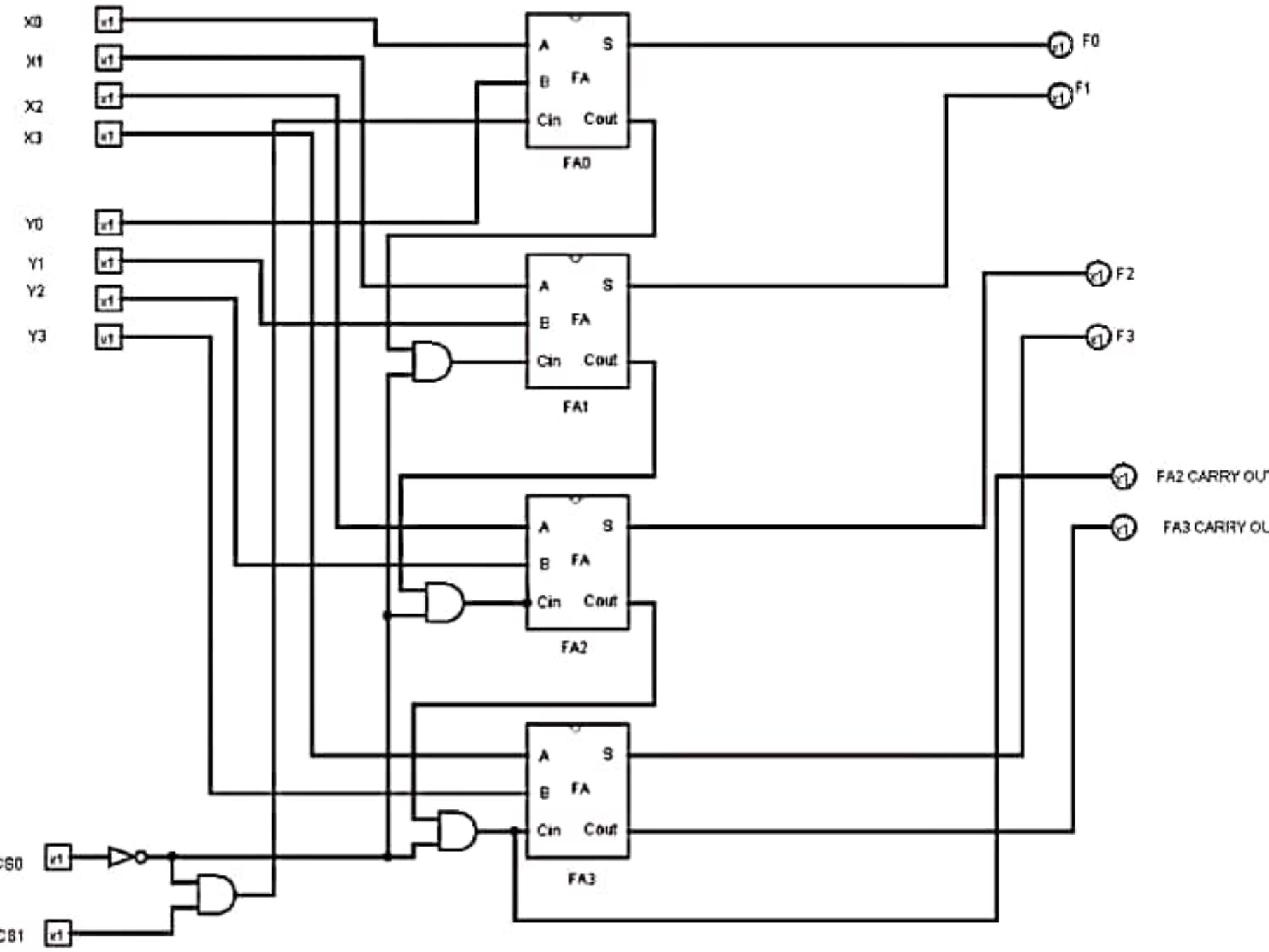
Full ALU



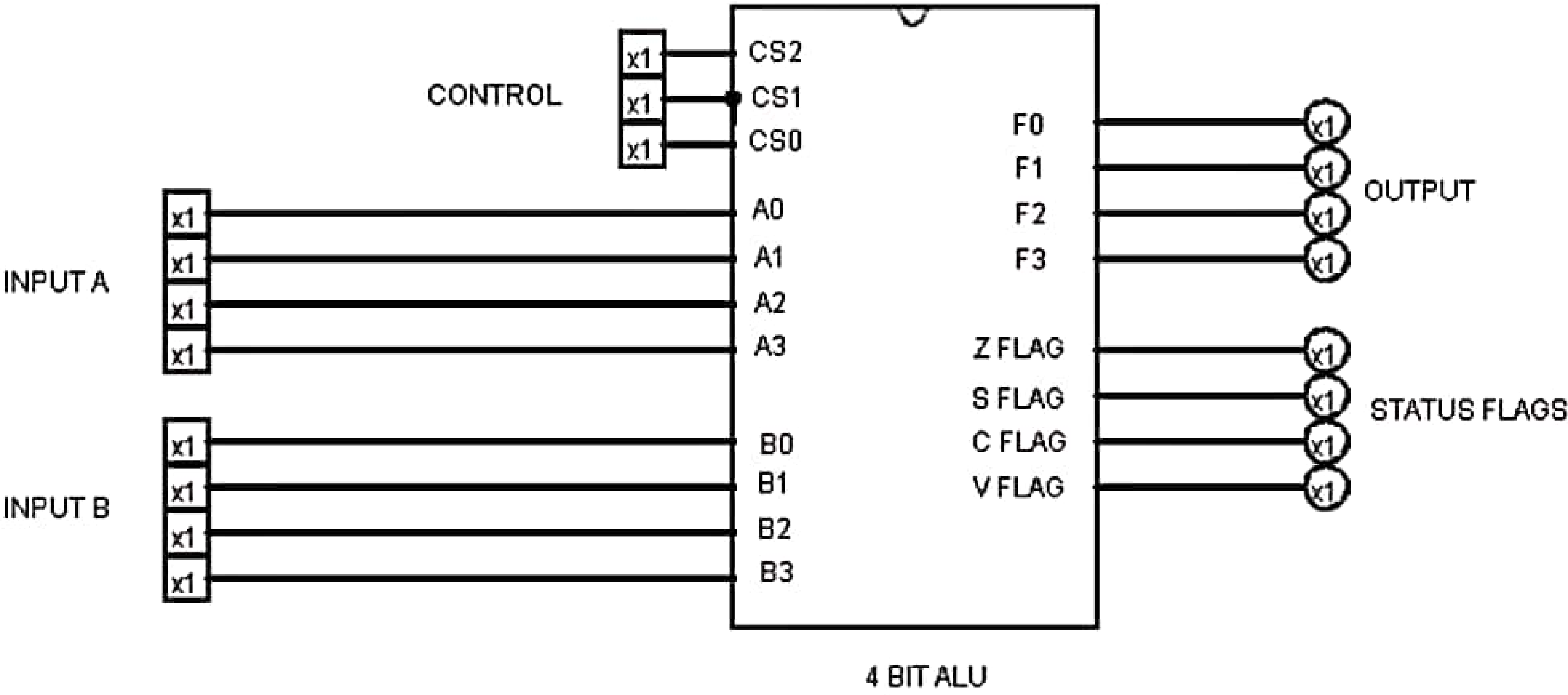
Combinational Part



Full Adder



4bit Parallel Full Adder Part


CONTROL
x1
x1
x1
CS2
CS1
CS0
INPUT A
x1
x1
x1
x1
A0
A1
A2
A3
INPUT B
x1
x1
x1
x1
B0
B1
B2
B3
F0
F1
F2
F3
OUTPUT
x1
x1
x1
x1
Z FLAG
S FLAG
C FLAG
V FLAG
STATUS FLAGS
x1
x1
x1
x1
4 BIT ALU

Full ALU

ICs used with count as chart:

IC name	Ic count
7425	1
7486	2
7404	1
7408	5
7432	3

Simulator Used:

Logisim 2.7.1

Discussion:

We designed the ALU in logisim simulator. First we made a full adder using basic gates, i.e XOR, AND and OR gates. Then we focused on the combinatorial circuit. The combinatorial circuit has three control inputs. cs0, cs1 and cs2 which determine the operations the ALU will perform. The combinatorial circuit also has two 4 bit inputs. After making the combinatorial circuit, we made 4 bit full adder circuit circuit using the subcircuit of the full adder we made before. This 4bit full adder has 2, 4bit inputs, 1 bit ~~in~~ carry in and 1 bit carray out, one 4bit output.

After making all necessary sub-circuits, we made the main ALU circuit. We used the sub-circuit of the combinational circuit and the sub-circuit of 4-bit adder. We also implemented the flag register.