Course ID: CSE 306

Course Title: Computer Architecture Sessional

Assignment-1: 4 bit ALU Simulation

Section: A2

Group: 03

Group Members:

1) 1905034
2) 1905036
3) 1905038
4) 1905054
5) 1905059

Introduction:

An arithmetic logic unit (ALU) is a multi-operation, combinational-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.

In our experiment, there are three selection variables (cs) which can enable us to perform $2^3 = 8$ distinct operations. The four data inputs from A are combined withe the four inputs from B to generate an operation at the F outputs. A combination circuit is used to modify the data imputs, A and B to produce the inputs for the parallel adders to get the desired F outputs.

In our ALU design, we use a 4-bit status registers. The status register contaim 4 status bits that are denoted by C (Carry), S (Sign), V (Overflow), Z (Zero). These status bits change during arithmetic operations. They indicate the following changes:

<u>CF:</u> Bit C is set 1 when the output carry of the ALU is 1, otherwise it is set 0.

<u>SF:</u> Bit S is set 1 when the highest order bit of the output of the ALU is 1, it is set 0 when the highest order bit is 0.

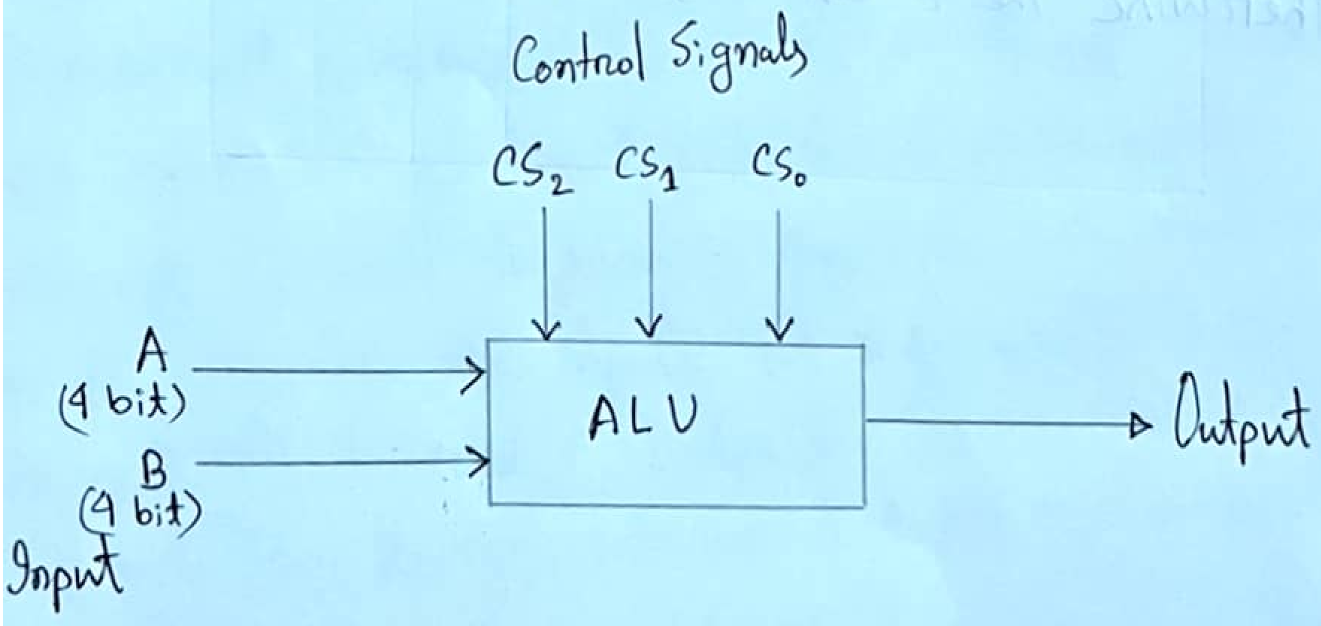
<u>OF:</u> Bit V is set 1 if the X-OR of carries C_3 and C_4 is 1, otherwise it is set 0.

<u>ZF:</u> Bit Z is set 1 if the result is zero, otherwise the Z bit is set 0.

Problem Specification:

Design a 4-bit ALU with three selection bits CS0, CS1, CS2 for performing the following operations:

CS_2	CS_1	CS_0	Functions
0	0	0	Sub
0	0	1	Transfer A
0	1	X	NEG A
1	0	0	AND
1	0	1	Add with carry
1	1	X	OR



Truth Table and required K-maps:

CS_2	CS_1	CS_0	Function	X_i	Y_i	C_{in}
0	0	0	$A_i - B_i$	A_i	$\overline{B_i}$	1
0	0	1	A_i	A_i	0	0
0	1	0	$-A_i$	$\overline{A_i}$	0	1
0	1	1	$-A_i$	$\overline{A_i}$	0	1
1	0	0	$A_i \cap B_i$	$A_i \cdot B_i$	0	0
1	0	1	$A_i + B_i + 1$	A_i	B_i	1
1	1	0	$A_i \cup B_i$	A_i OR B_i	0	0
1	1	1	$A_i \cup B_i$	A_i OR B_i	0	0

Truth table for determining C_{in}:

CS_2	CS_1	CS_0	C_{in}
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

K-Map:

$CS_2 \backslash CS_1CS_0$	00	01	11	10
0	1	0	1	1
1	0	1	0	0

$$C_{in} = \sum m(0,2,3,5)$$

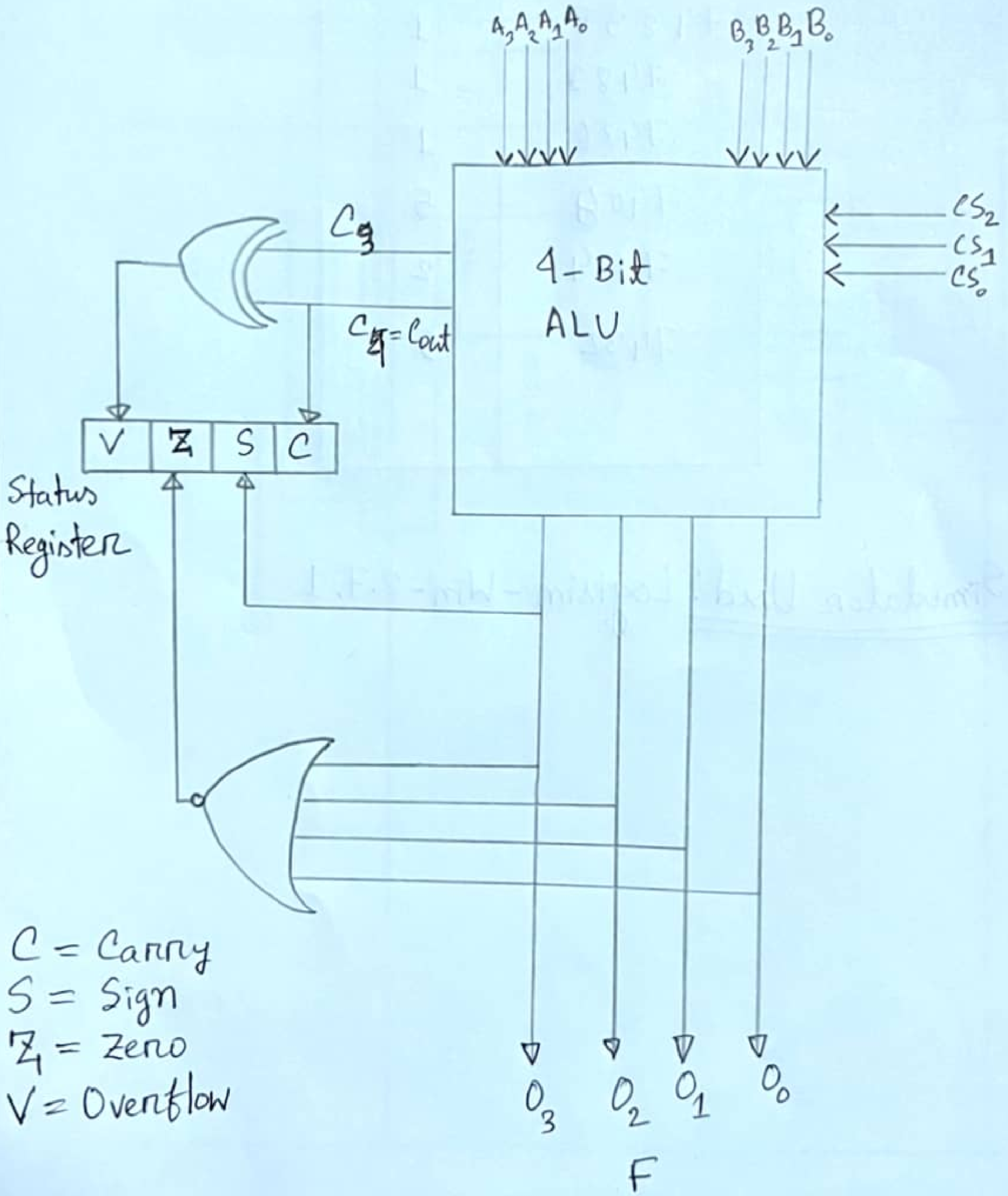
$$= \overline{CS_2}\cdot CS_1 + \overline{CS_2}\cdot\overline{CS_0} + CS_2\cdot\overline{CS_1}\cdot CS_0$$

$$Y_i = \overline{B}\left(\overline{CS_2}\cdot\overline{CS_1}\cdot\overline{CS_0}\right) + B\left(CS_2\cdot\overline{CS_1}\cdot CS_0\right)$$

$$X_i = \left(\overline{CS_2}\cdot\overline{CS_1}\right)\cdot A_i + \left(\overline{CS_2}\cdot CS_1\right)\cdot\overline{A_i} + \left(CS_2\cdot\overline{CS_1}\right)\left(\overline{CS_0}\cdot A_i\cdot B_i + CS_0\cdot A_i\right)$$

$$+ \left(CS_2\cdot CS_1\right)\left(A_i + B_i\right)$$

Block Diagram:

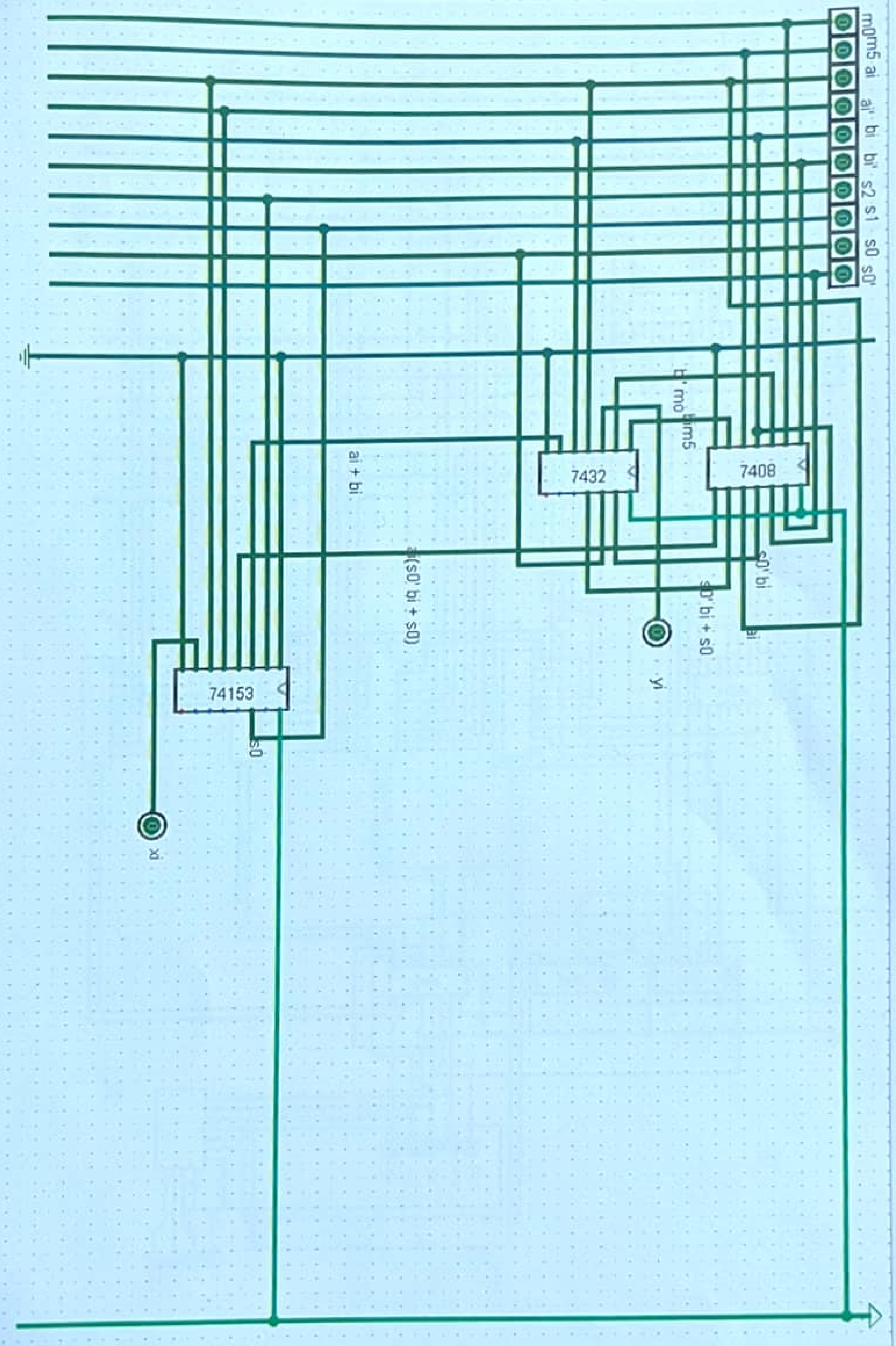


C = Carry
S = Sign
Z = Zero
V = Overflow

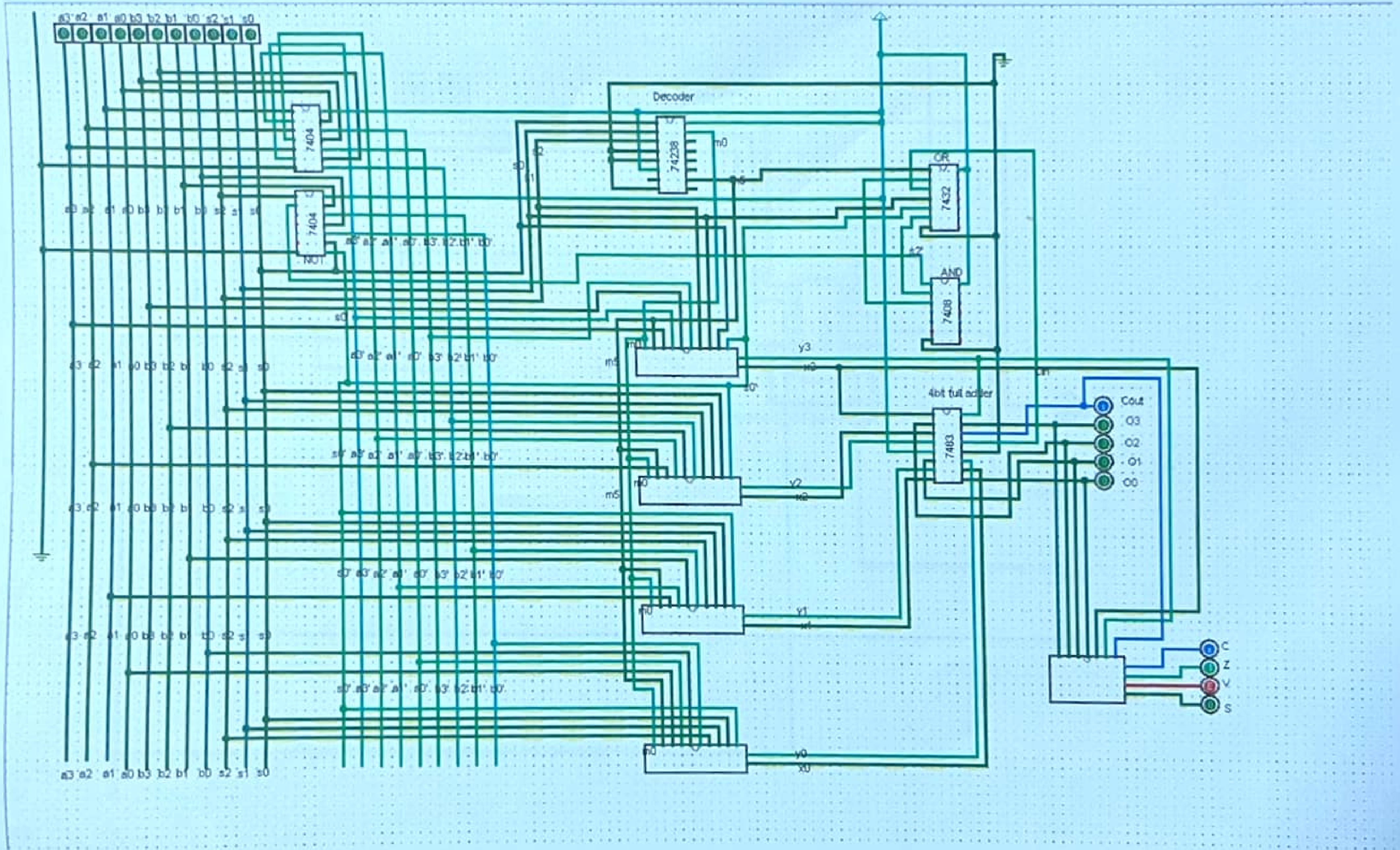
Required ICs:

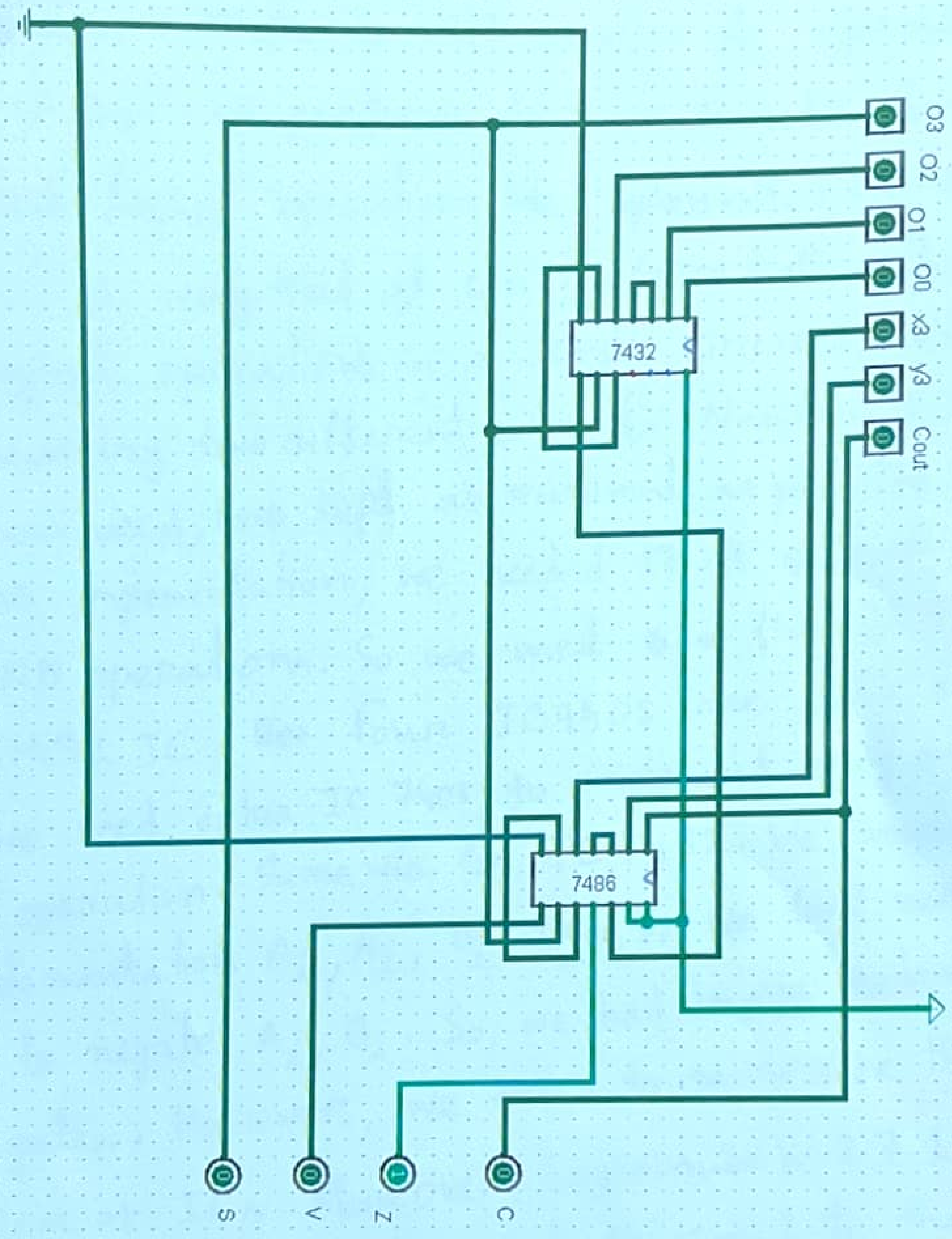
IC	Quantity
74153	2
74238	1
7483	1
7486	1
7408	5
7404	2
7432	5

Simulator Used: Logisim-Win-2.7.1



m0
m5
ai
ai'
bi
bi'
s2
s1
s0
s0'
7408
7432
74153
b' m0 bm5
ai + bi
a(s0' bi + s0)
s0' bi + s0
s0' bi
ai
yi
s0
xi


a3 a2 a1 a0 b3 b2 b1 b0 s2 s1 s0
7404
7404
NOT
a3' a2' a1' a0' b3' b2' b1' b0'
s0'
Decoder
74238
m0
s0
s1
s2
s6
m5
y3
x3
s0'
OR
7432
s7
AND
7408
4bit full adder
7483
Cin
Cout
O3
O2
O1
O0
y2
x2
y1
x1
y0
X0
C
Z
V
S


O3
O2
O1
O0
x3
y3
Cout
7432
7486
S
V
Z
C

Discussion:

In this experiment, we were tasked to implement an ALU that can perform four arithmetic operations and two logical operations. We implemented the ALU in such a way that it can perform both arthimetic and logical operations in a single circuit, instead of requiring two different circuits. Number of IC that was used, was kept as minimal as possible. In our implementation, we needed 17 OR operations, 17 AND operations. So we used five IC 7408, five 7432 IC. Four IC 7408 are fully used, and we used extra IC 7408 to implement only one AND operation. Same was for IC 7432. We needed to implementation A_i, $\bar{A}_i$, B_i, $\bar{B}_i$. So, we took more gates to negate A_i, B_i. So, we had to use extra Not gates. However, we tried to minimize the number of ICs. Moreover, Logisim-Win-2.7.1 simulation software was used to simulate the circuit. Finally, we got appropiate outputs for given inputs and selection bits.