CSE 306

Computer Architecture Sessional

Assignment-1: 4-bit ALU Simulation

Section: A2

Group: 02

Group Members:

(A) 1705036
(B) 1705037
(C) 1705038
(D) 1705039
(E) 1705044
(F) 1505048

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

The selection variable cs1 distinguishes between arithmetic and logical operations, while the selection variable cs2 is used as input carry in our ALU design.

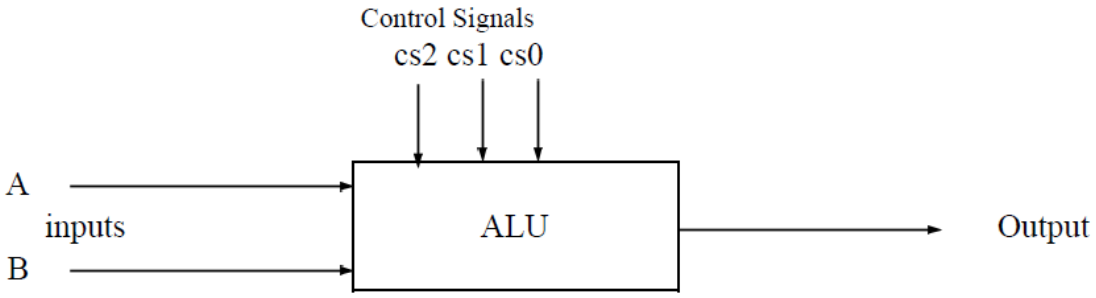
In our ALU design, we use a 4-bit status register. This status register contains 4 status bits that are denoted by C(Carry), S(Sign), V(Overflow) and Z(Zero). These status bits change during arithmetic operations. They indicate the following changes:

- **CF:** Bit C is set 1 when the output carry of the ALU is 1, otherwise it is set 0.
- **SF:** 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.
- **OF:** Bit V is set 1 if the X-OR of carries C_4 and C_5 is 1, otherwise it is set 0.
- **ZF:** 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 and cs2 for performing the following operations:

cin cs2	cs1	cs0	Functions
0	0	0	Decrement A
0	0	1	Subtract with Borrow
1	0	0	Transfer A
1	0	1	Subtract
X	1	0	OR
X	1	1	Complement A



Truth table and required K-maps:

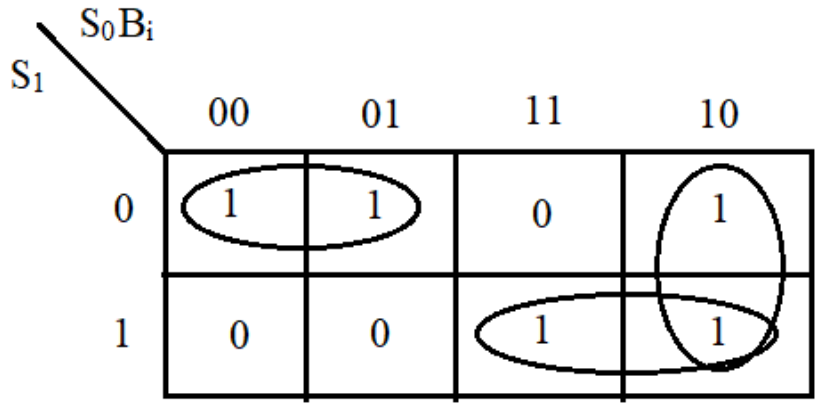
Truth table for the functions:

cs2	cs1	cs0	Function	X_i	Y_i
0	0	0	A - 1	A_i	1
1	0	0	A	A_i	1
0	0	1	A - B - 1	A_i	B_i'
1	0	1	A - B	A_i	B_i'
X	1	0	A ∨ B	$A_i \vee B_i$	0
X	1	1	A'	A_i	1

Truth table for determining Y_i :

S_1	S_0	B_i	Y_i
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

K-map:

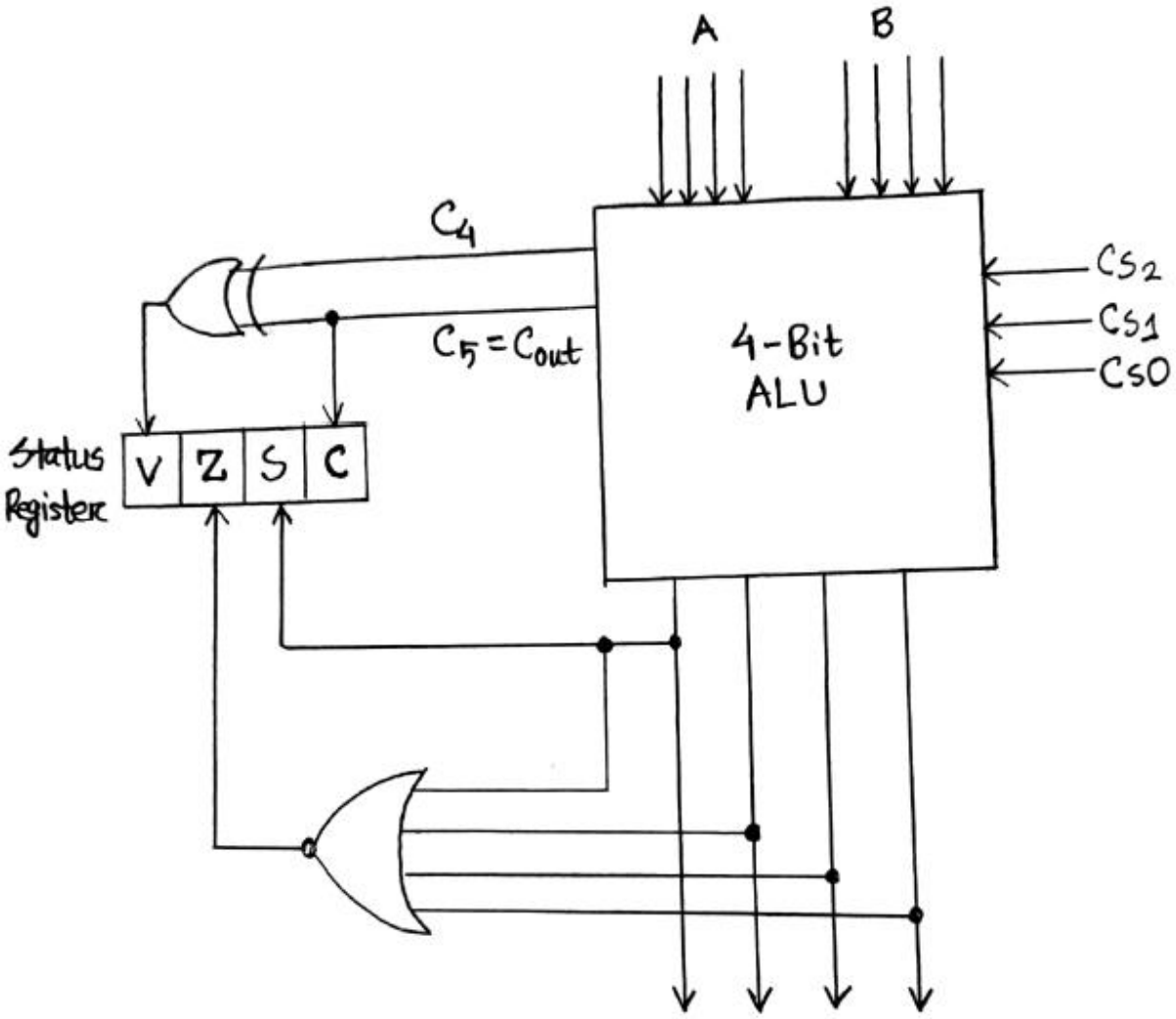


$Y_i = S_1' S_0' + S_0 B_i' + S_1 S_0$

$X_i = A_i + S_1 S_0' B_i$

$Z_i = S_1' C_{in}$

Block Diagram:



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

Complete Circuit Diagram:

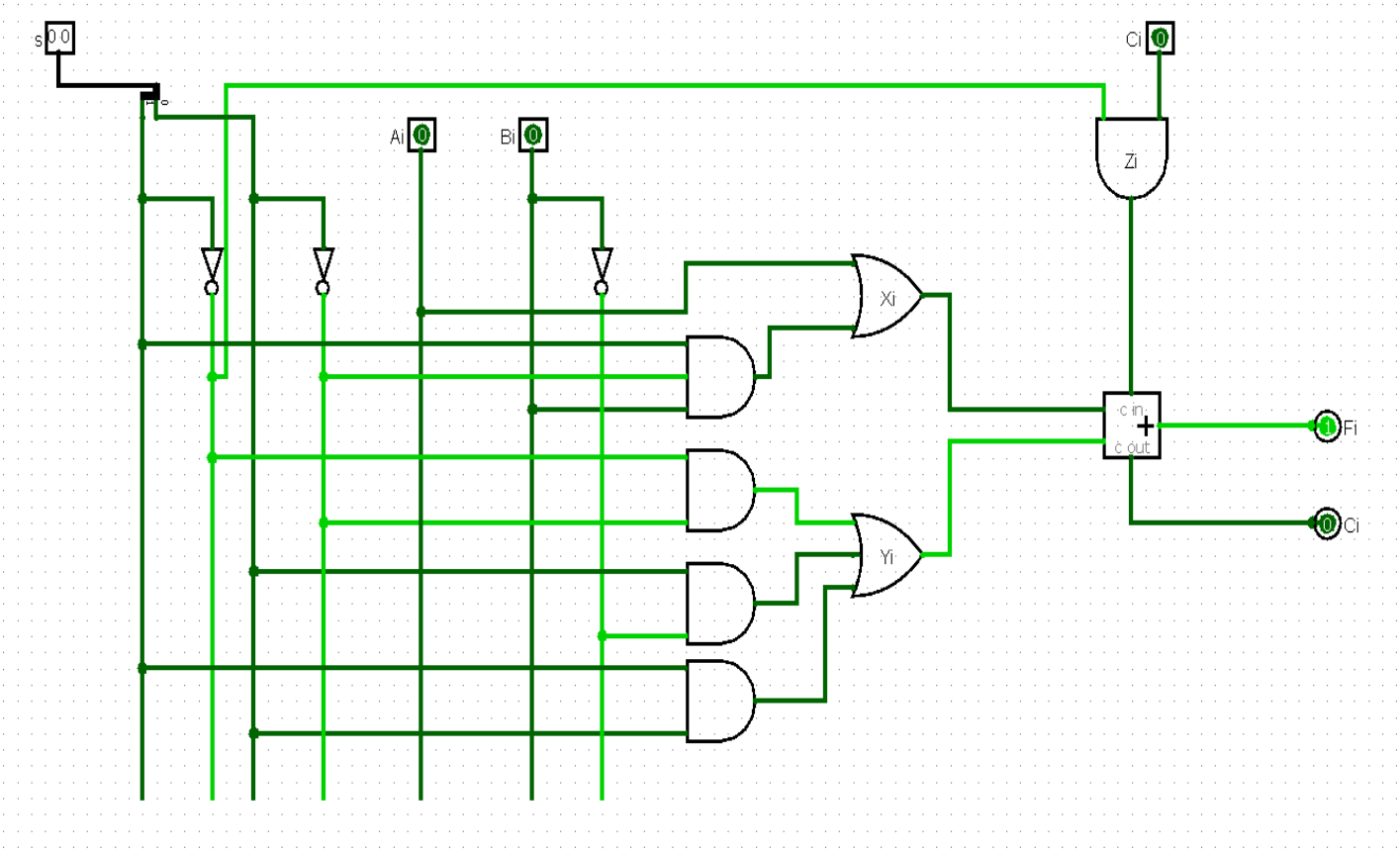


Figure-1: 1-bit ALU

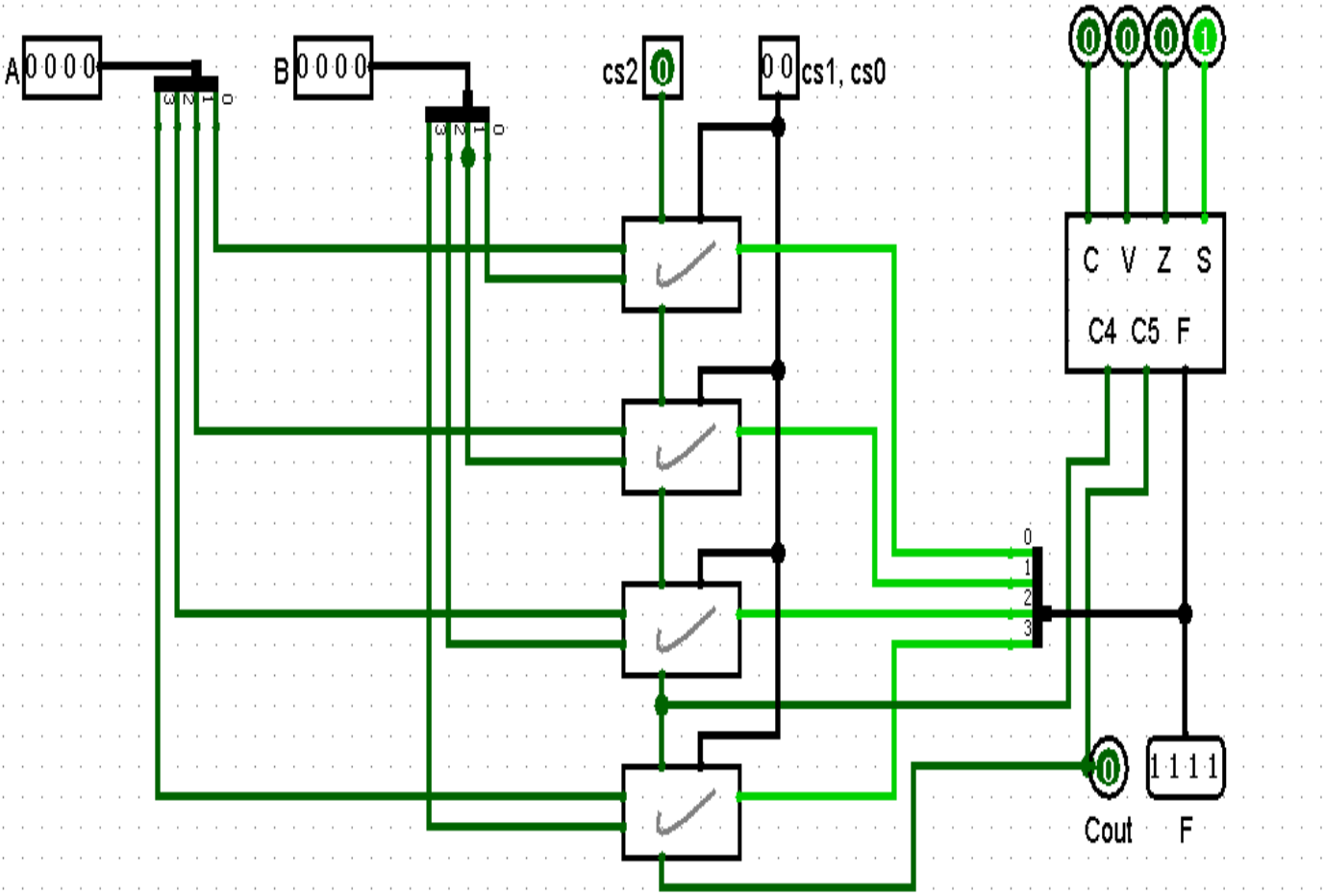


Figure-2: 4-bit ALU

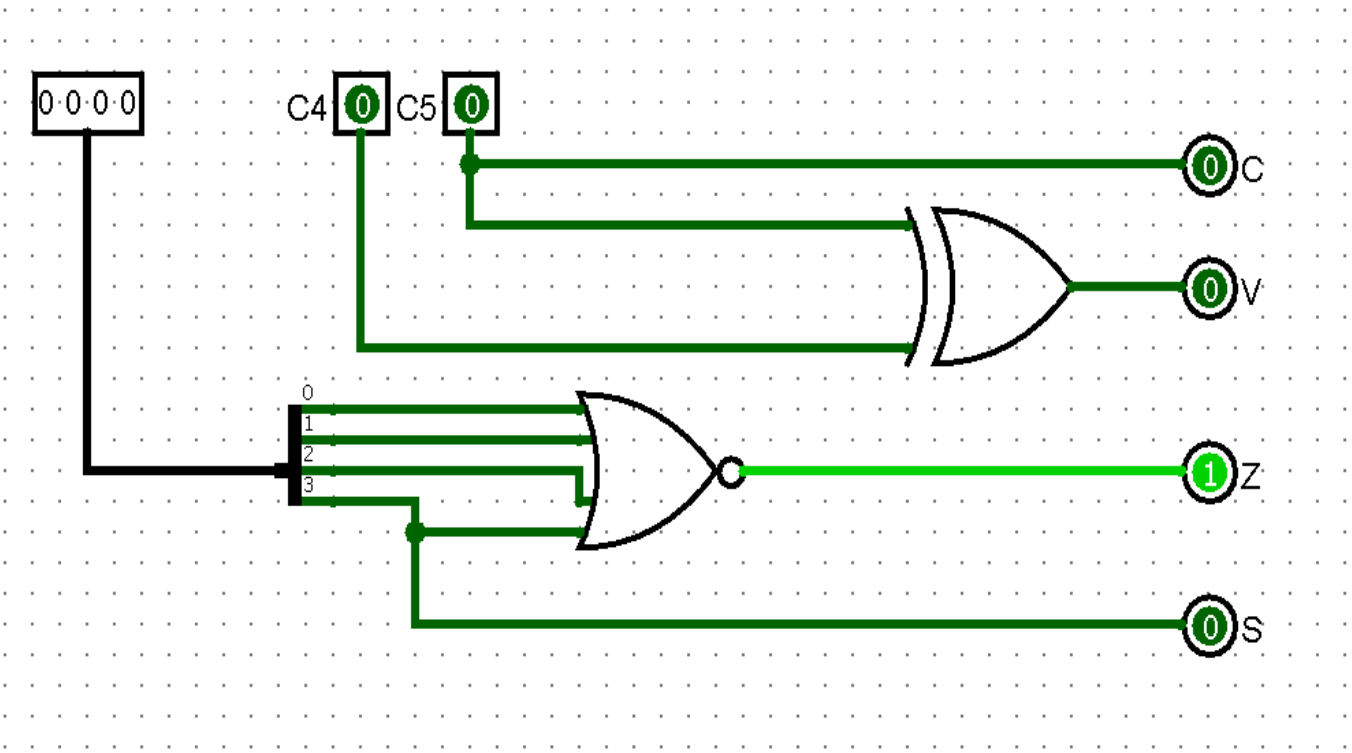


Figure-3: Status Registers

Required ICs:

IC Name	Count
IC 7404	**2**
IC 7408	**6**
IC 7432	**3**
IC 7483	**1**
IC 7402	**1**
IC 7486	**1**

Simulator Used: Logisim-Win-2.7.1

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 is such a way that it can perform both arithmetic 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. Logisim-Win-2.7.1 simulation software was used to simulate the circuit.