CSE306
4-bit ALU Simulation

Jawad Ul Kabir
1705062

Md. Mahfuzur Rahman Rifat
1705063

Taseen Mubassira
1705065

Ataf Fazledin Ahamed
1705066

Nishat Farhana Purbasha
1705067

31 March, 2021

1 Introduction

An arithmetic logic unit (abbr. ALU) is a multi-operation, combinational-logic digital function- capable of performing a set of basic arithmetic operations and a set of logical operations. The ALU has a set of selection lines which enables a user 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.

2 Problem Specification with Assigned Instructions

Design a 4-bit ALU with three selection bits cs0, cs1 and cs2 for performing the following operations:

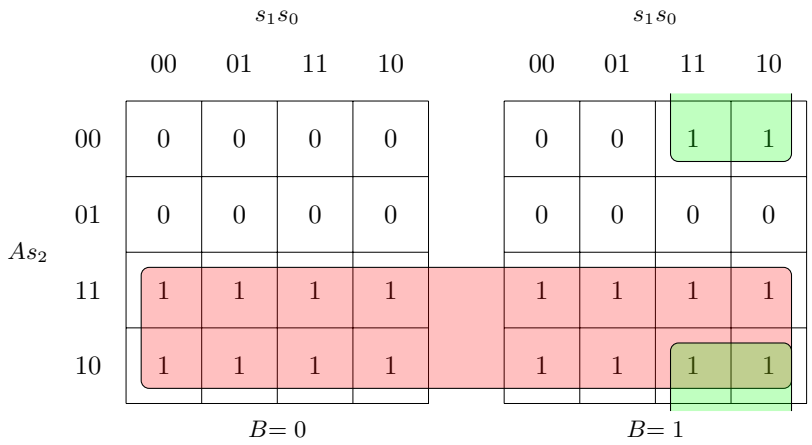
cs2	**cs1**	**cin cs0**	**Functions**
0	0	0	Decrement A
0	0	1	Transfer A
0	1	×	OR
1	0	0	Subtract with Borrow
1	0	1	Subtract
1	1	×	Complement A

3 Truth Table and K-maps

Truth table for the functions:

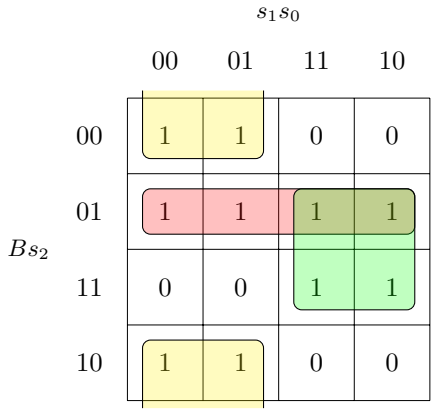
cs2	**cs1**	**cs0**	**Function**	$\boldsymbol{X_i}$	$\boldsymbol{Y_i}$	$\boldsymbol{Z_i}$
0	0	0	$A-1$	A_i	1	C_i
0	0	1	A	A_i	1	C_i
0	1	0	A OR B	$A_i \vee B_i$	0	0
0	1	1	A OR B	$A_i \vee B_i$	0	0
1	0	0	$A-B-1$	A_i	B_i'	C_i
1	0	1	$A-B$	A_i	B_i'	C_i
1	1	0	A'	A_i	1	0
1	1	1	A'	A_i	1	0

K-Map for X_i:



$X_i = A_i + B_i s_2' s_1$

K-Map for Y_i:



$Y_i = B_i' s_2 + s_2 s_1 + s_2' s_1'$

K-Map for Z_i:

$C_i s_0$ \ $s_2 s_1$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	0	0	1
10	1	0	0	1

$Z_i = C_i s_1'$

4 Block Diagram

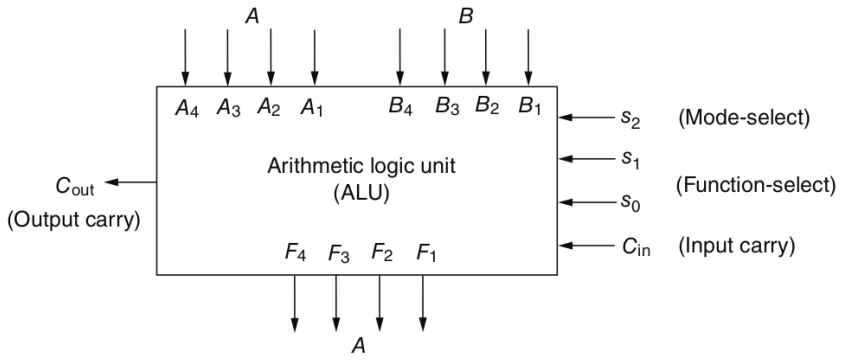


Figure 1: Block diagram of a 4 bit ALU

5 Complete Circuit Diagram

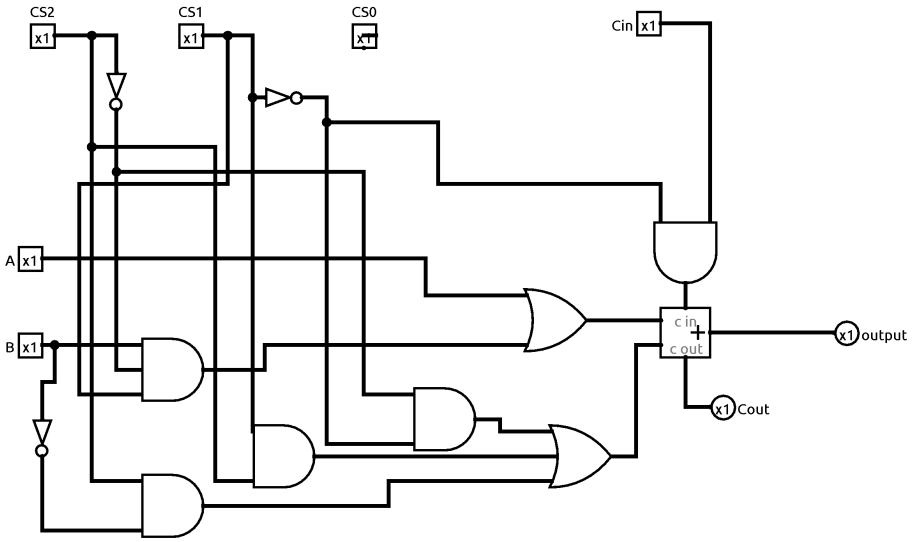


Figure 2: 1 Bit ALU

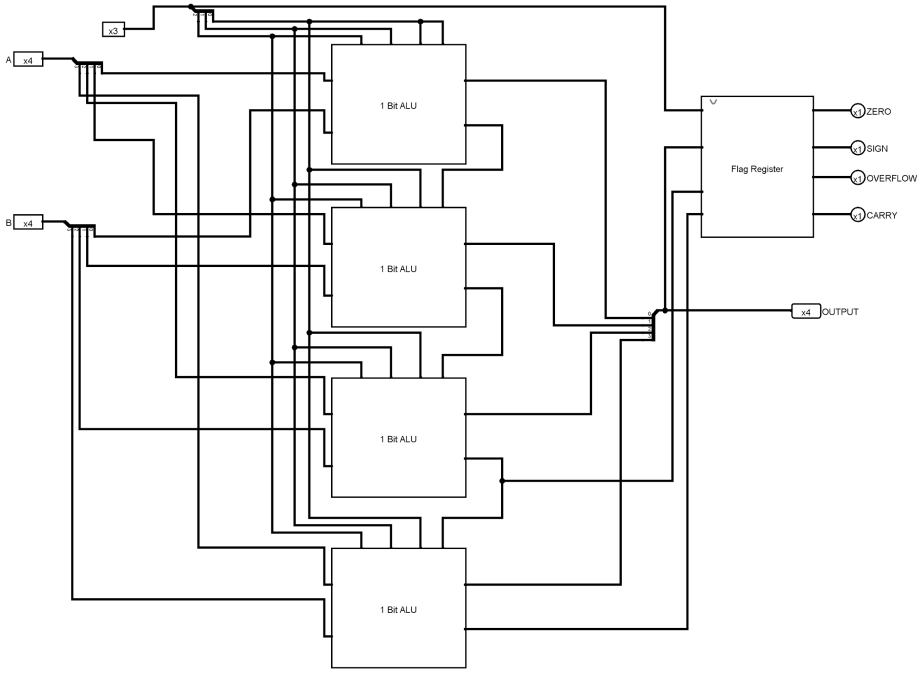


Figure 3: 4 Bit ALU

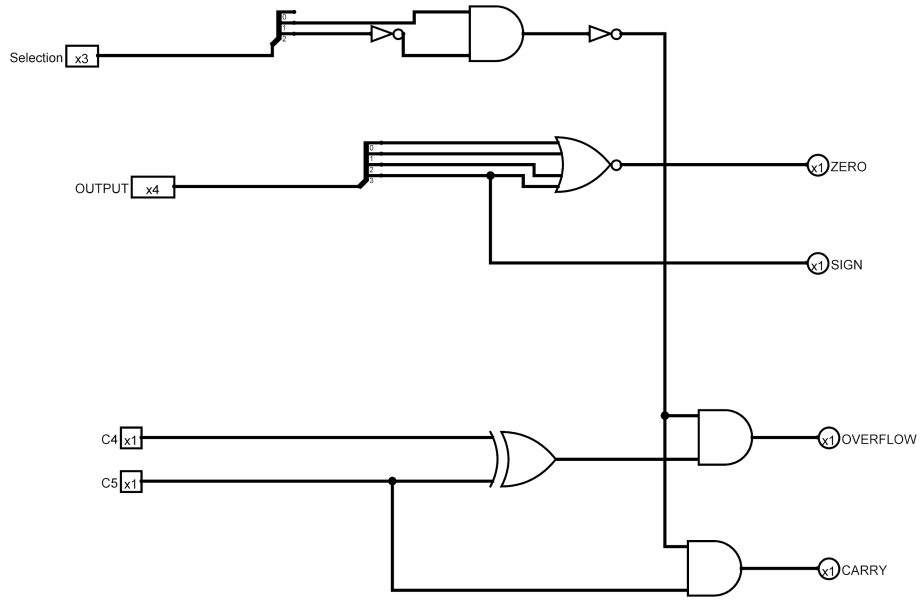


Figure 4: Flag Register

Additional combinational circuit for the Flag Register:
The Flag register requires the Carry flag (C) and the Overflow flag (V) to be cleared (0) after a logical OR operation. The ALU performs an OR operation when s2=0 and s1=1; Therefore,
The modified carry = $C(s_2's_1)'$
The modified overflow = $V(s_2's_1)'$

6 Total Number of ICs Used in the Implementation

Name	IC no.	Number of Gates	Number of ICs
OR	IC 7432	15	4
AND	IC 7408	27	7
NOT	IC 7404	15	4
Adder	IC 7483	4	4

7 Simulator Info

Logisim - Java Platform (Version 2.7.1)

8 Discussion

- We designed the ALU in Logisim simulator.
- Initially we designed an 1 bit ALU and from it by cascading we designed the 4 bit ALU according to the functions that were given as our task. And lastly, we designed a Flag register to show the values of C, S, Z and V flags. We combined the two circuits together and got the 4 bit ALU.
- For the designing of ALU according to the given functions, we formed a truth table, used it to construct the required k-maps for X_i, Y_i and Z_i, derived equations for each of them and used the equations to design the 1 bit ALU.
- We used cs0 as our initial C_{in} bit and the other cs1 and cs2 as selector bits for the required arithmetic and logical operations.
- One important thing to mention here is that the flag register in the ALU does not fully comply with the rules of assembly language in the sense that the values of C, V, S or Z bits may change after a logical NOT operation even though assembly language rules clearly dictate that NOT operations should not affect the Flag register at all.