North South University

Department of Electrical & Computer Engineering

Project Report (Part 1)

Submitted To:	Syed Mahmud Husain (SMH2)
Course Code:	CSE332
Course Name:	Computer Organization & Architecture
Section:	**9**
Name & ID:	Md. Rifat Ahmed ~ 1931725042
Date of Submission:	15/11/2021

Objectives:

Our objective for project part 1 is to build a MIPS ALU meaning a 32-bit ALU that has 6 to 8 functions.

Truth Table:

ALUOp		Function field, Fx						ALUOp O2 O1 O0	Operation
Binvert	**Ainvert**	**F5**	**F4**	**F3**	**F2**	**F1**	**F0**		
1	1	X	X	X	X	X=1	0	0 0 0	NOR
1	1	X	X	X	X	X=1	1	0 0 1	NAND
0	0	X	X	X	X	0	0	1 0 0	XOR
0	0	X	X	X	X	0	1	0 1 0	Add
0	0	X	X	X	X	1	0	0 1 1	Add with Carry
1	0	X	X	X	X	X=1	0	1 1 0	Sub
1	0	X	X	X	X	X=1	1	1 0 1	Sub with Borrow
0	0	X	X	X	X	1	1	1 1 1	Multiply

Table: Truth Table for the Control Circuit

K-Maps:

Since this is a SOP circuit we'll consider all the don't cares (X) as 1.

For 00,

Binvert Ainvert \ F1F0	00	01	11	10
00	0 (XOR)	0 (Add)	1 (Multi)	1 (Add C)
01	1 (X)	1 (X)	1 (X)	1 (X)
11	1 (X)	1 (X)	1 (NAND)	0 (NOR)
10	1 (X)	1 (X)	1 (Sub B.)	0 (Sub)

$$\therefore 00 = Binvert\, F0 + Binvert\, \overline{F1} + \overline{Binvert}\, F1 + \overline{Binvert}\, Ainvert$$

For 01,

Binvert Ainvert \ F1 F0	00	01	11	10
00	0 (XOR)	1 (Add)	1 (Multi)	1 (Add.C)
01	1 (X)	1 (X)	1 (X)	1 (X)
11	1 (X)	1 (X)	0 (NAND)	0 (NOR)
10	1 (X)	1 (X)	0 (Sub B.)	1 (Sub)

$$\therefore 01 = Binvert\, \overline{F1} + \overline{Binvert}\, F1 + \overline{Binvert}\, F0 + \overline{Binvert}\, Ainvert$$
$$+ Binvert\, \overline{Ainvert}\, \overline{F0}$$

For O2,

Binvert Ainvert \ F1 F0	00	01	11	10
00	1 (XOR)	0 (Add)	1 (Multi)	0 (Add C.)
01	1 (X)	1 (X)	1 (X)	1 (X)
11	1 (X)	1 (X)	0 (NAND)	0 (NOR)
10	1 (X)	1 (X)	1 (Sub B.)	1 (Sub)

$$\therefore O2 = \overline{F1}\,\overline{F0} + \text{Binvert}\,\overline{F1} + \overline{\text{Binvert}}\,\text{Ainvert} + \text{Binvert}\,\overline{\text{Ainvert}}$$

$$+ \overline{\text{Binvert}}\,F1\,F0$$

Circuit Diagram:

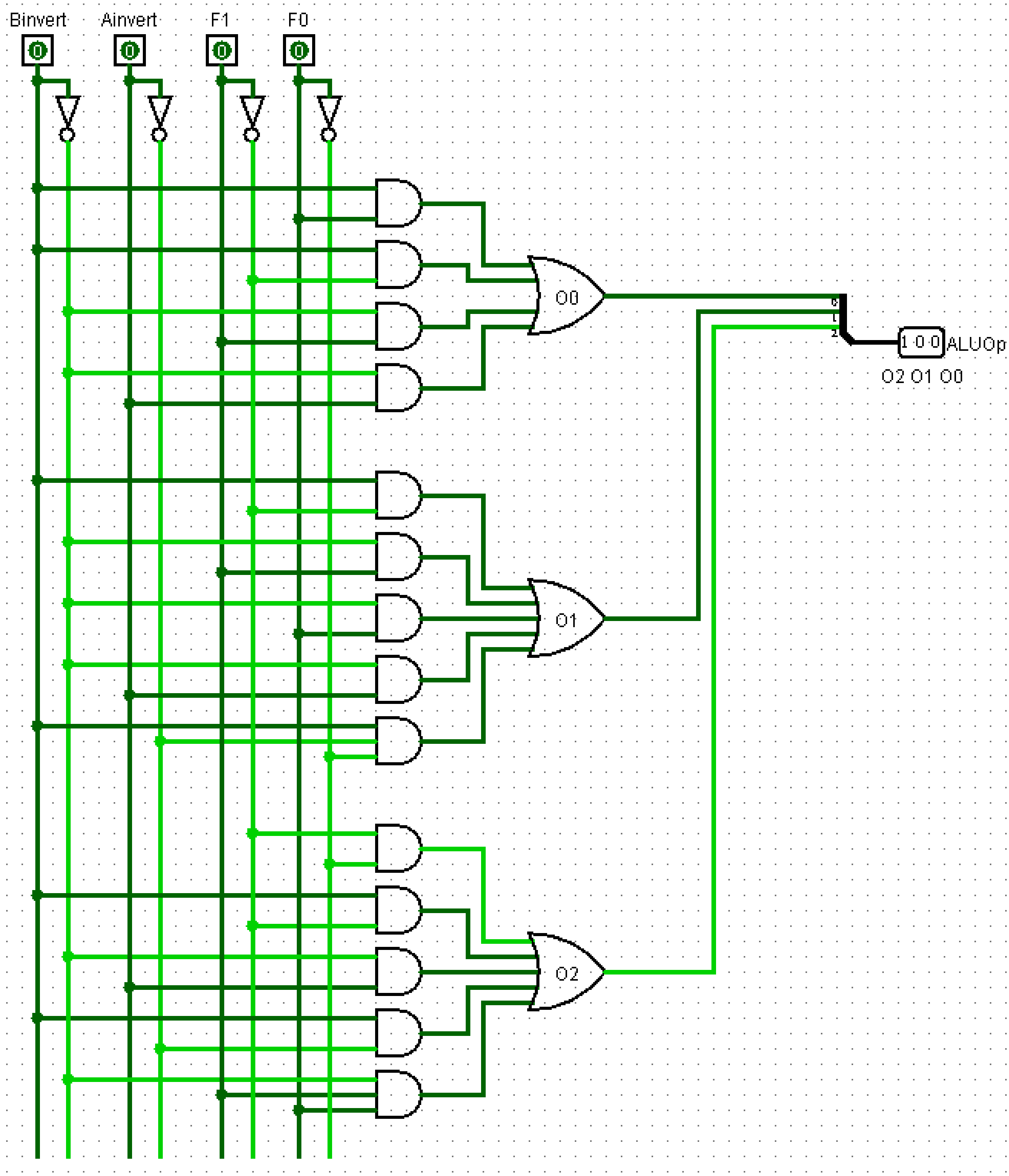

Figure 1: ALU Control Circuit

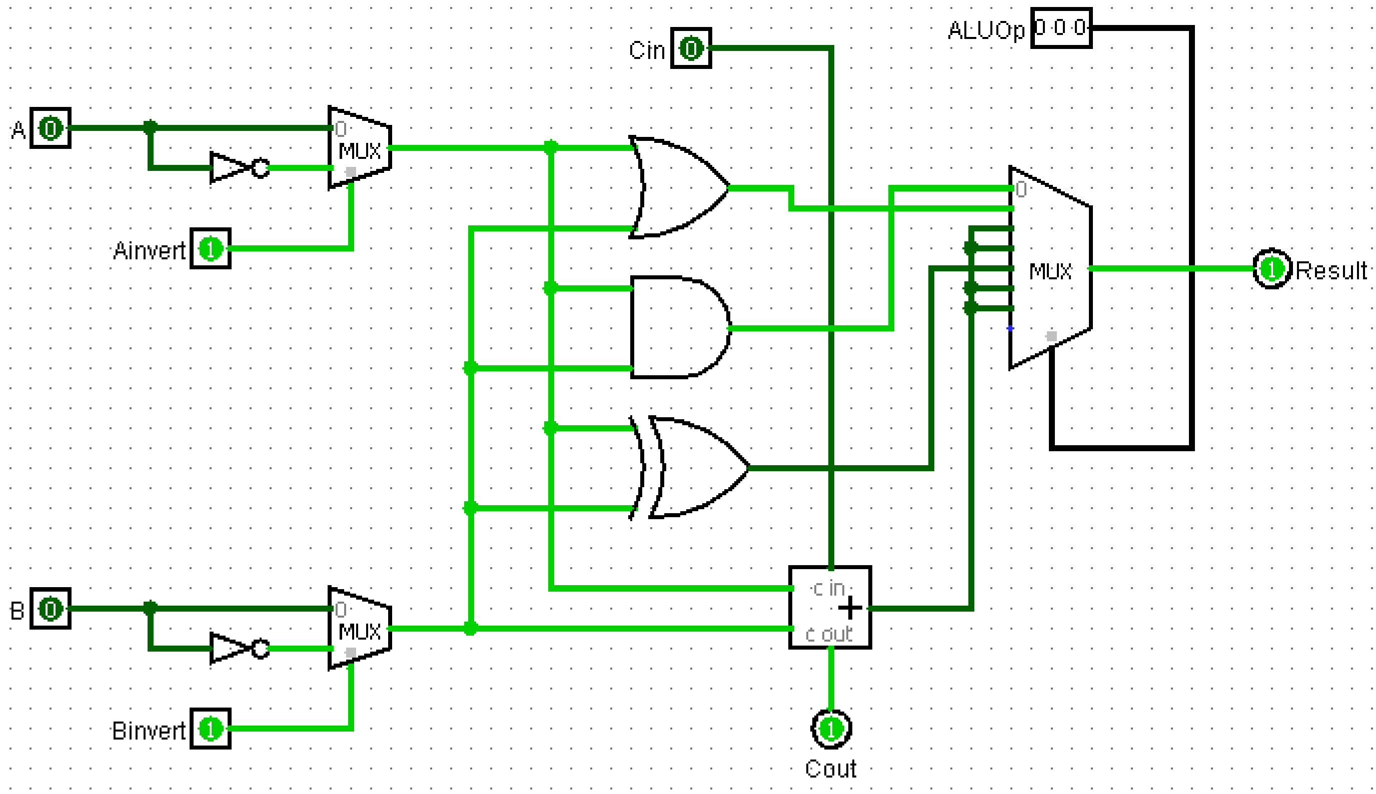

Figure 2: 1-bit ALU with 7 Functions

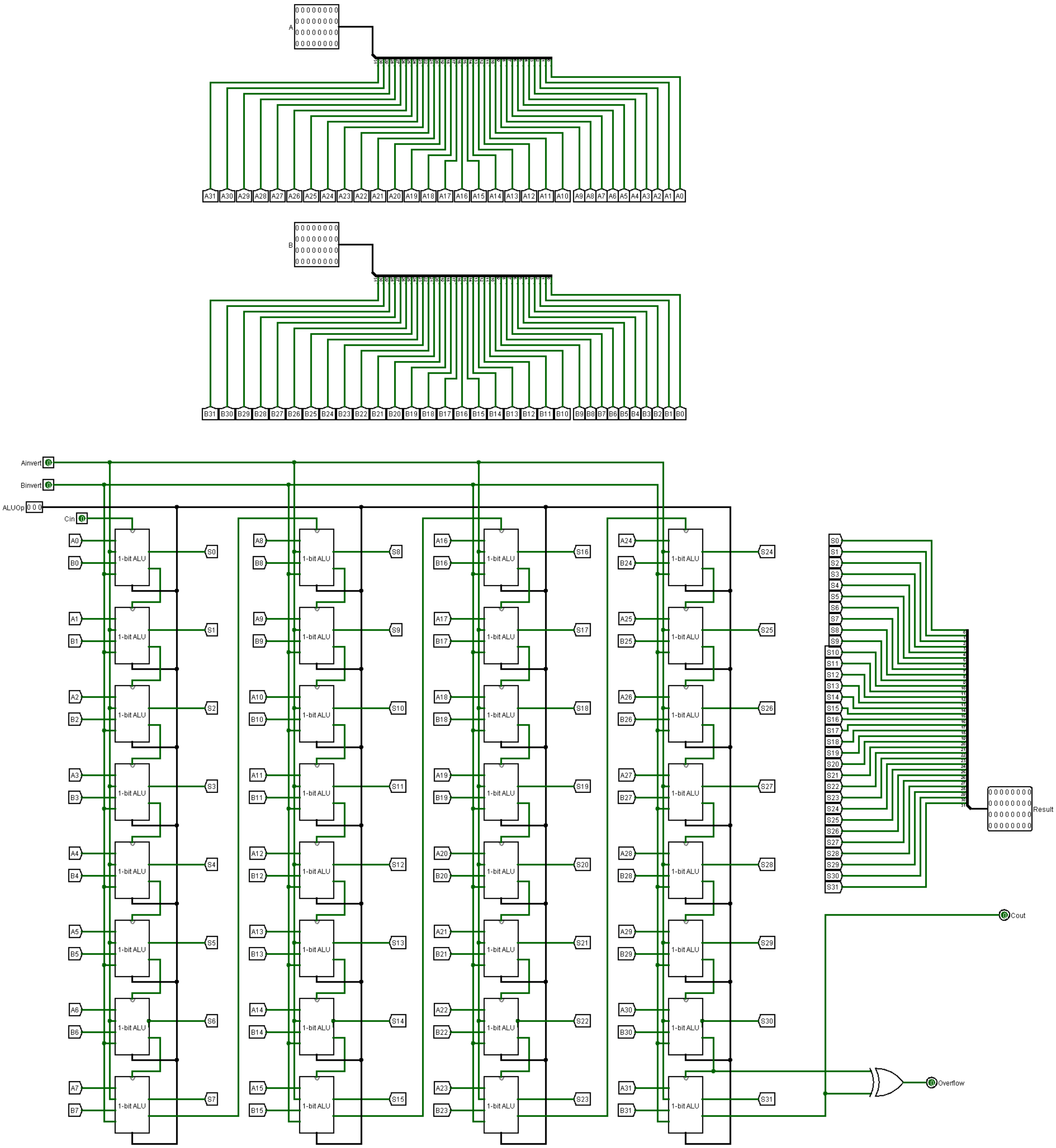

Figure 3: 32-bit ALU with 7 Functions

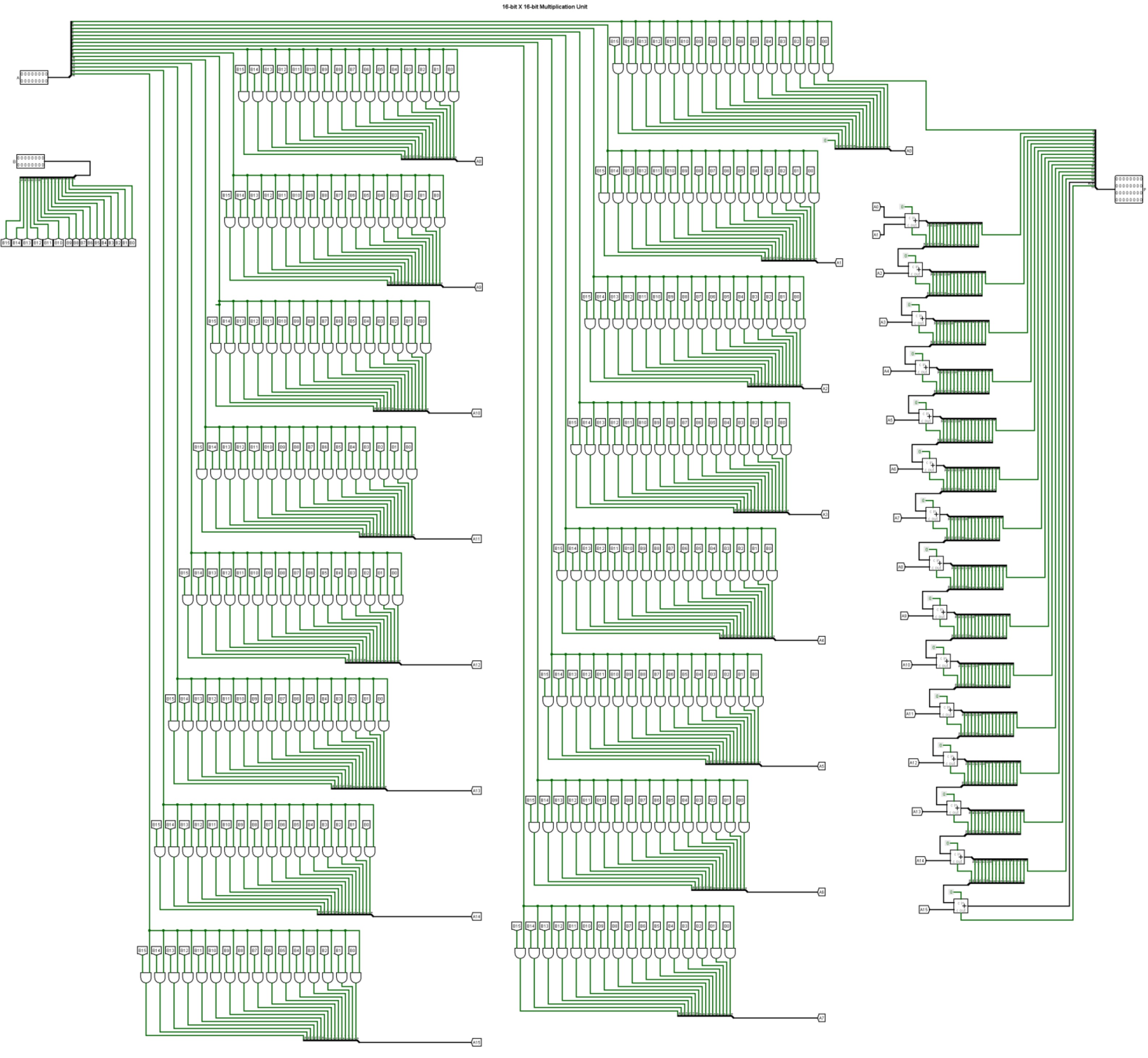

Figure 4: 16-bit by 16-bit Multiplication Unit

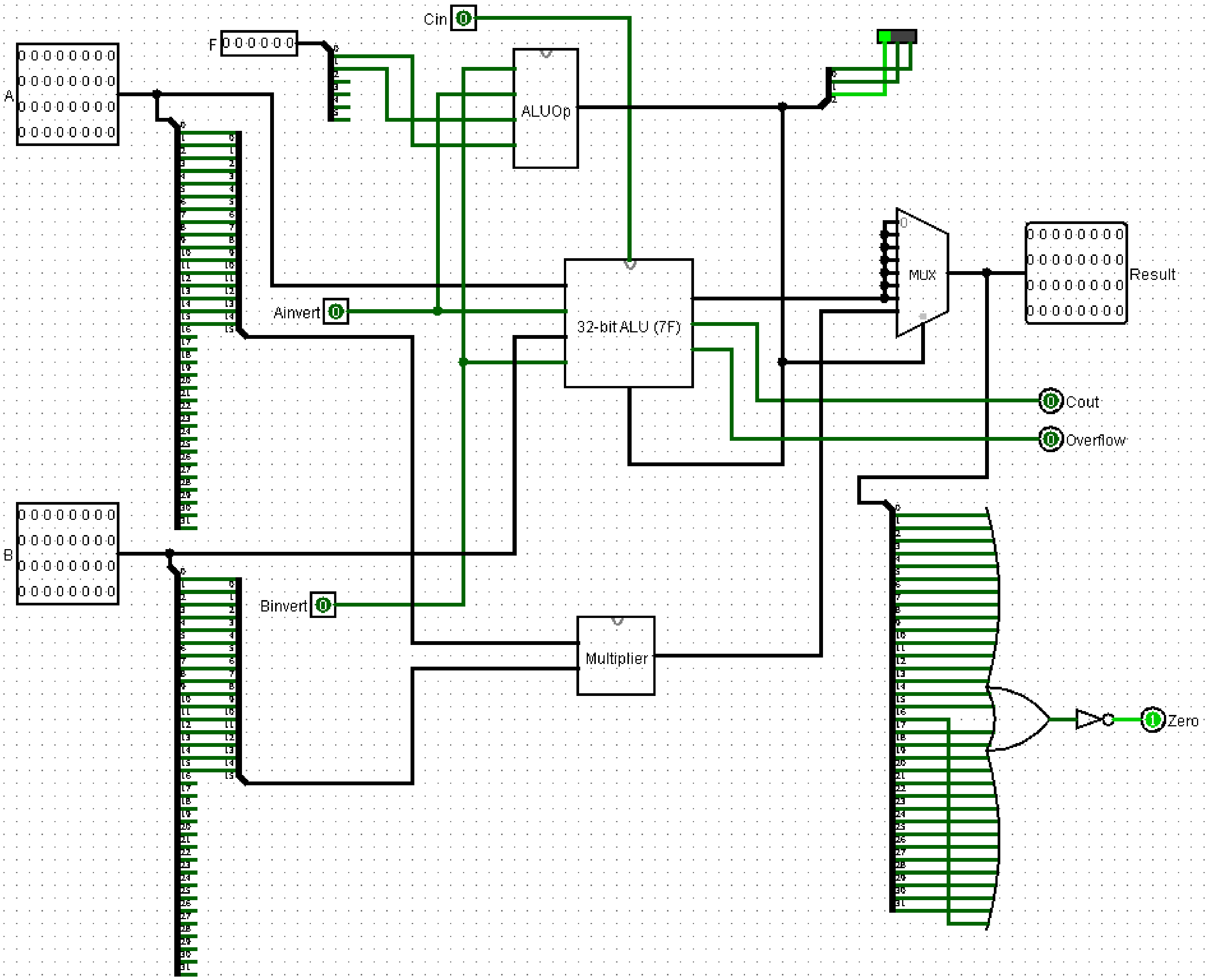

Figure 5: 32-bit ALU with 8 Functions

Discussion:

In this part of the project we had to build a MIPS ALU with 6 to 8 functions. I chose to do NAND, NOR, XOR from logic unit and Add, Add with Carry, Sub, Sub with Borrow from Arithmetic unit and the Multiplication total 8 functions. At first we built the truth table then from there using K-Map we calculated the functions for our ALU control circuit. Then using a splitter we connected the 3 O/P O0, O1, O2. After that we built the 1-bit ALU with 7 functions we excluded Multiplication because cascading it was too much complicated and we had already done a 16-bit by 16-bit Multiplication unit which we'll be using later on. So, using the 1-bit ALU we cascaded it into a 32-bit ALU with 7 functions then we also kept an output for overflow by connecting the last carry out with the last carry in using a XOR gate. Then finally we took the 32-bit ALU with 7 functions and the Multiplier but here we had to split our input A and B as the input for the previous 7 functions were 32-bit but the input for Multiplication unit was 16-bit by 16-bit meaning we only needed the 16 least significant bits from our input A and B. Then we connected the ALU Control to the select bits of the MUX and in the 7. functions ALU and we added the result of the 7 functions ALU to the 7 input pins so that the result gets carried from the 7 function ALU to our MUX, the Final ALU. Then we split the result added them to an OR gate followed by a NOT gate for our zero flag. And thus we got our MIPS ALU.