

ELEN 21 Lab 6: Mini-calculator with Small
Arithmetic Logic Unit (ALU)

Pre-Lab

Tamir Enkhjargal
October 2018

II. Pre-Lab

1. Fill in the table below for your adder B inputs and carry in input for each of the four operations listed.

| Op-Select Input | A inputs | B inputs | Carry-in | Operation | Notes |
|-------------------|-------------------|-----------------------|----------|-----------------|------------------------|
| $r_1 r_0 p_1 p_0$ | $a_3 a_2 a_1 a_0$ | $b_3 b_2 b_1 b_0$ | | | |
| 0000 | $a_3 a_2 a_1 a_0$ | $b_3 b_2 b_1 b_0$ | 0 | $X = A + B$ | Addition |
| 0001 | $a_3 a_2 a_1 a_0$ | $b_3 b_2 b_1 b_0$ | 1 | $X = A + B + 1$ | Addition and Increment |
| 0010 | $a_3 a_2 a_1 a_0$ | $!b_3 !b_2 !b_1 !b_0$ | 0 | $X = A - B - 1$ | Subtract and Decrement |
| 0011 | $a_3 a_2 a_1 a_0$ | $!b_3 !b_2 !b_1 !b_0$ | 1 | $X = A - B$ | Subtract |

Table 1: Our table for the arithmetic unit and inputs

2. Use a 4-bit adder and other logic components to design this first part of the arithmetic unit.

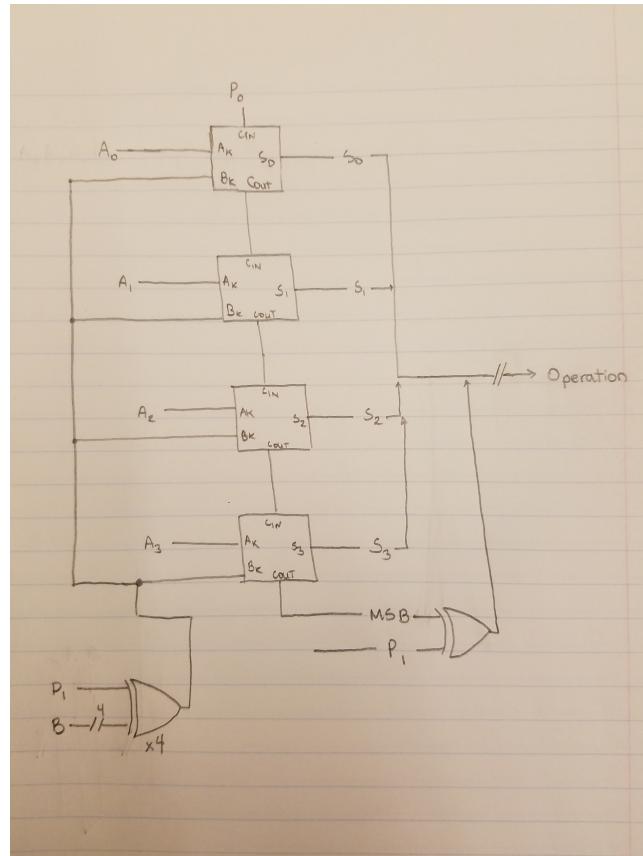


Figure 1: The logical circuit with inputs a, b, and $p_1 p_0$

3. Modify your design to add the other four operations shown in the problem statement table. Try to make a small efficient circuit that will implement 8 operations. Draw a schematic for your design.

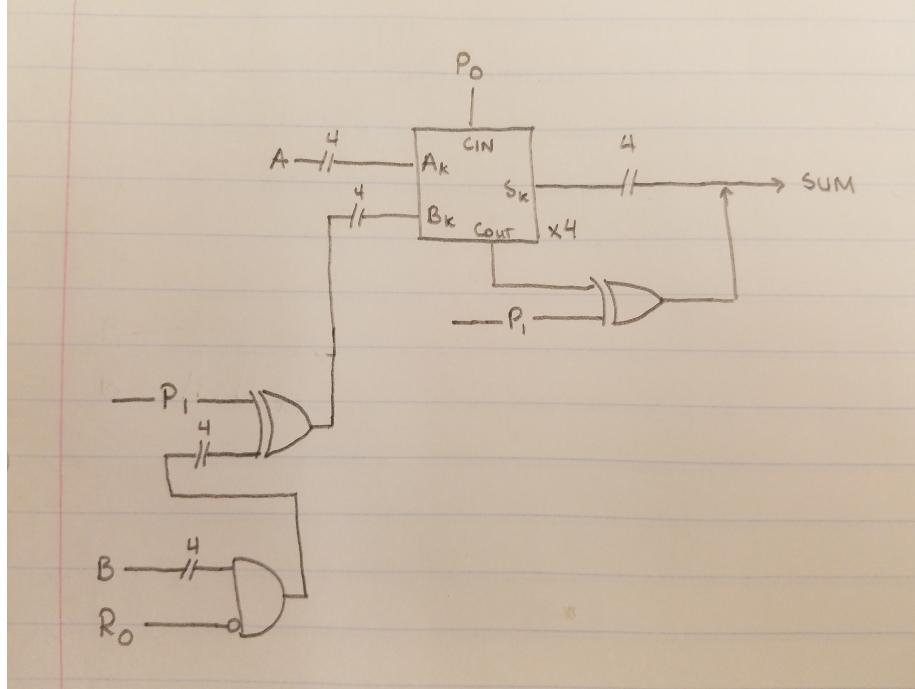


Figure 2: The logical circuit with inputs a , b , p_1p_0 , and r_0

4. Consider how you would efficiently and strategically start your test of your ALU circuit after you implement it. Assume that the 4-bit adder has been tested and is known to function correctly. You will want to test the data path to be sure that the four-bit values of the input switches, the MUXes, the adder inputs and outputs, and the seven segment displays are all consistently ordered and connected. You also want to check that the control logic is working correctly. What does p_1 control? What does p_0 control?

p_1 controls the sign of B , whether we will be adding or subtracting B . p_0 controls the carry-in input.

- 5.** Write the first four steps of your test plan. Indicate what values you would set for the inputs and what you would expect to see at the outputs of each step. Indicate what you would have verified if you see the expected output. There are many possible testing strategies.

Since we know/assume that our 4-bit adder works as intended, we only need to test, just like *Lab 5*, if the seven segment display works as intended with the Verilog code. Here we're testing the first two operations (just like *Lab 5*): adding, and adding and incrementing.

The next thing we need to do is to test our first 2 of 4 new inputs, $r_1 r_0 p_1 p_0$. Test if our next two operations, $X = A - B - 1$ and $X = A - B$ works in conjunction with testing the four different permutations of p_1 and p_0 .

Step three we test if r_0 works as intended, which is to just set $B = 0000$ for the next four operations, where B is not included in the function.

Step four would be testing whatever new operations will be included with what r_1 is going to be set for.

6. Draw the schematic for your design of the small 8-operation ALU from the problem statement using the Altera libraries. You will have two 4-bit inputs a and b , four control inputs $r1r0p1p0$, a 4-bit output x and a carry out C .

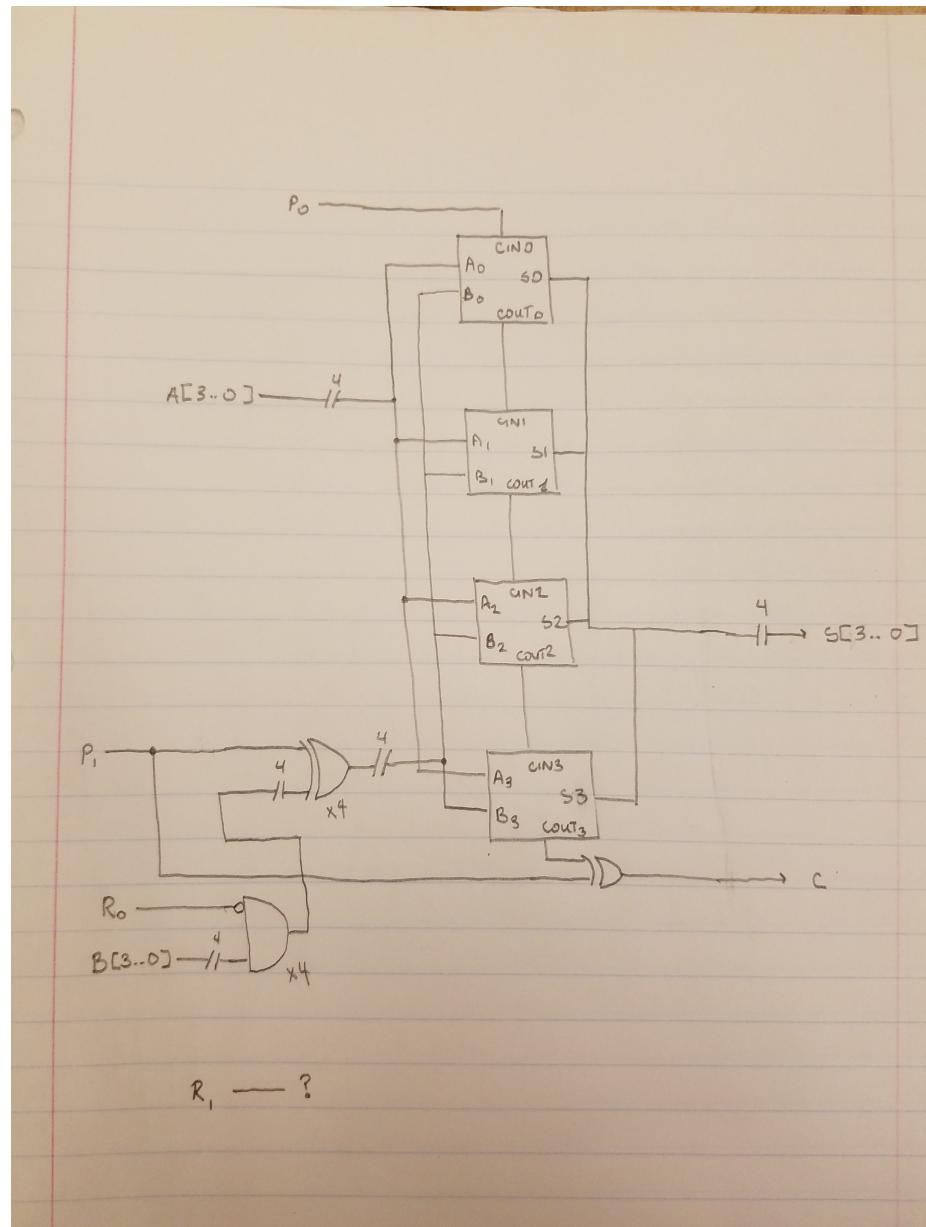


Figure 3: The logical circuit design with inputs as it would appear in Altera