Lab Report 1

Yuhuang Chen (804449266), Zeyuan Xu (004255573)

1 Part 1: 1 Bit ALU

1.1 Introduction

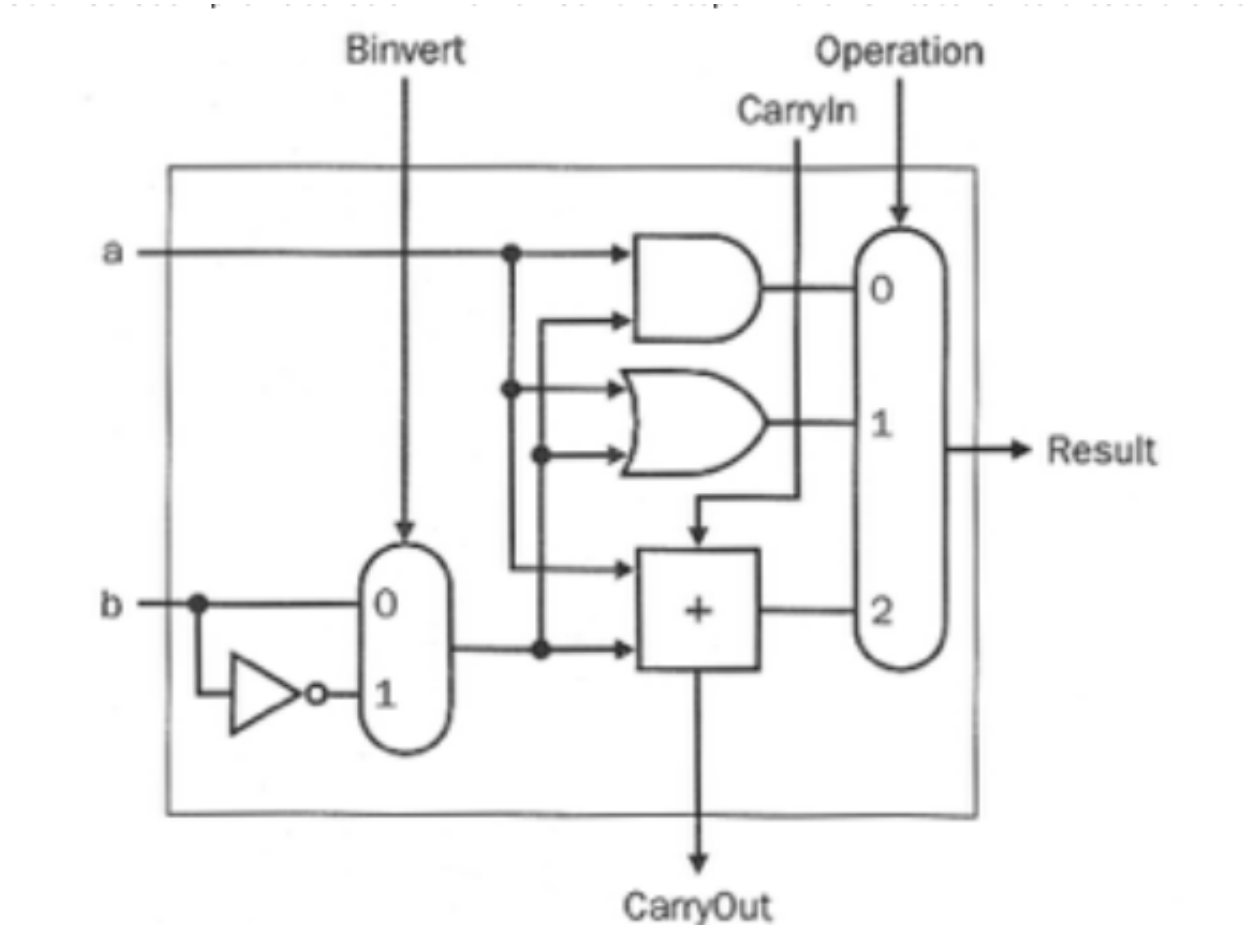


Figure 1: 1-Bit-ALU

In the first part, we simply need to follow the diagram (figure 1) to implement an 1-bit ALU, which has components as: OR gate, AND gate, NOT gate, 1-bit full adder, 2-1 mux, and 3-1 mux. The 2-1 mux is easily implemented via the logical gate specified in figure 2. The 4-1 mux is simply hooked up in the logic specified in figure 3. The schematic diagram is:

1.2 Simulation Result

The simulation result for the 1-bit ALU is shown in figure 4, in which all operations are tested and verified.

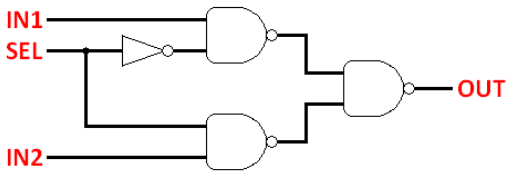


Figure 2: 2-1 mux

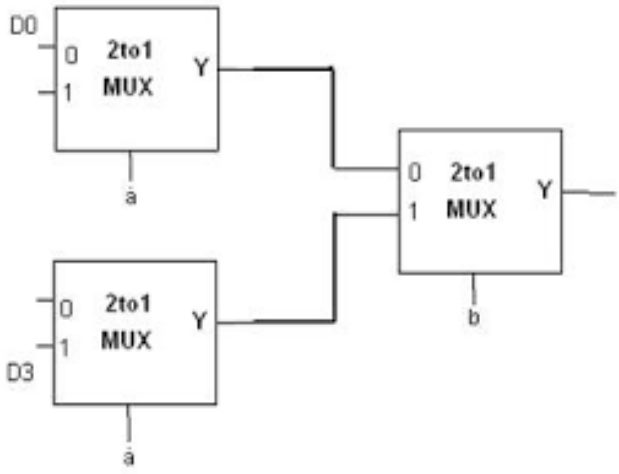


Figure 3: 4-1 mux

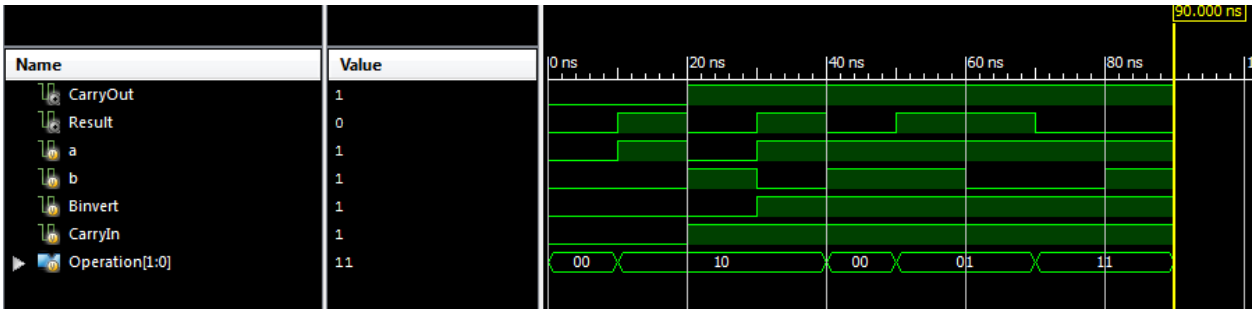


Figure 4: 1-bit ALU demo

1.3 Discussion

Nothing particular is notieable about the 1-bit ALU. It sets the logic for further implementation of 16 bit ALU, whose structural builds upon it.

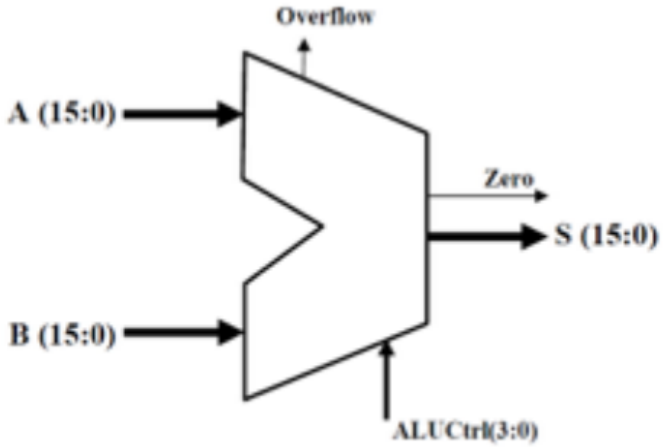
2 Part 2: 16 Bit ALU

2.1 Introduction

The 16-bit ALU consists consists of operations specified in figure 5. The most complicated part of the ALU design is to design the 16 bit full adder, which should also contain overflow detection. The 16-1 mux is built upon by 4-1 mux, and the logic is depicted in figure 6. The 16-1 mux is shown in figure 7. The adder is implemented as a ripple adder. When buiding the 16-bit adder, we first build 4-bit adder from 1-bit adders, then using the same method to construct the 16-bit adder from the 4-bit adders. The logic can be depicted in figure 8.

2.2 Simulation Result

The simulation result is in figure 9.



ALU Ctrl	Description
0000	Subtraction
0001	Addition
0010	Bitwise OR
0011	Bitwise AND
0100	Decrement
0101	Increment
0110	Invert
1100	Arithmetic Shift Left
1110	Arithmetic Shift Right
1000	Logical Shift Left
1010	Logical Shift Right
1001	Set on Less than or Equal

Figure 5: 16-bit ALU

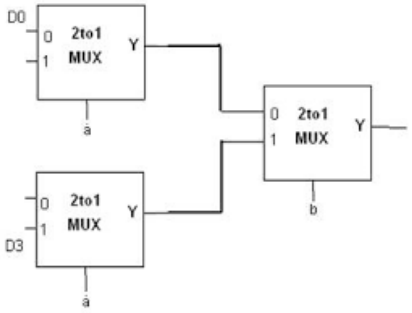


Figure 6: 4-1 mux

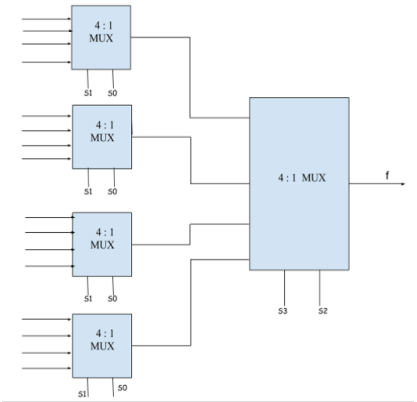


Figure 7: 16-1 mux

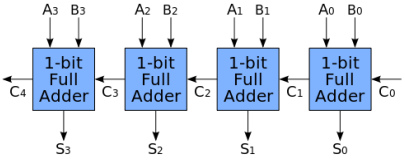


Figure 8: 4 bit adder

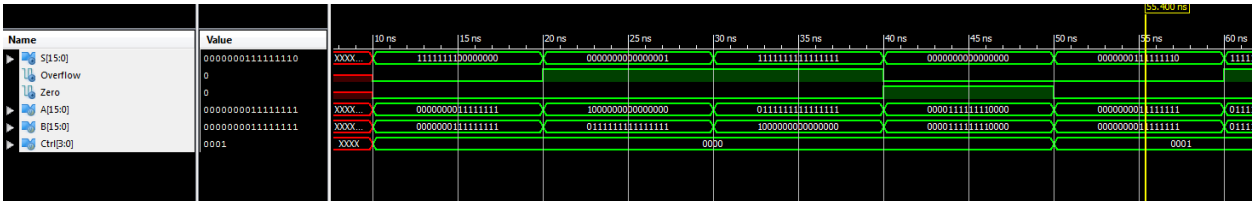


Figure 9: 16-bit ALU demo

2.3 Discussion

The difficulties encountered for the 16-bit ALU design are: 1. The overflow handling: the logic is that when two positive numbers add, and the result's sign bit is 1, have an overflow. This generic case is complicated by: in the subtraction case, when a positive number subtract a negative number, but result is negative, or a negative number minus a positive number, but result is positive. When we implement the set less than operation, we also need to specify the logic:

```
logic here
```

3 Part 3: Register File

3.1 Introduction

3.2 Simulation Result

3.3 Discussion

4 Part 4: Questions and Answers