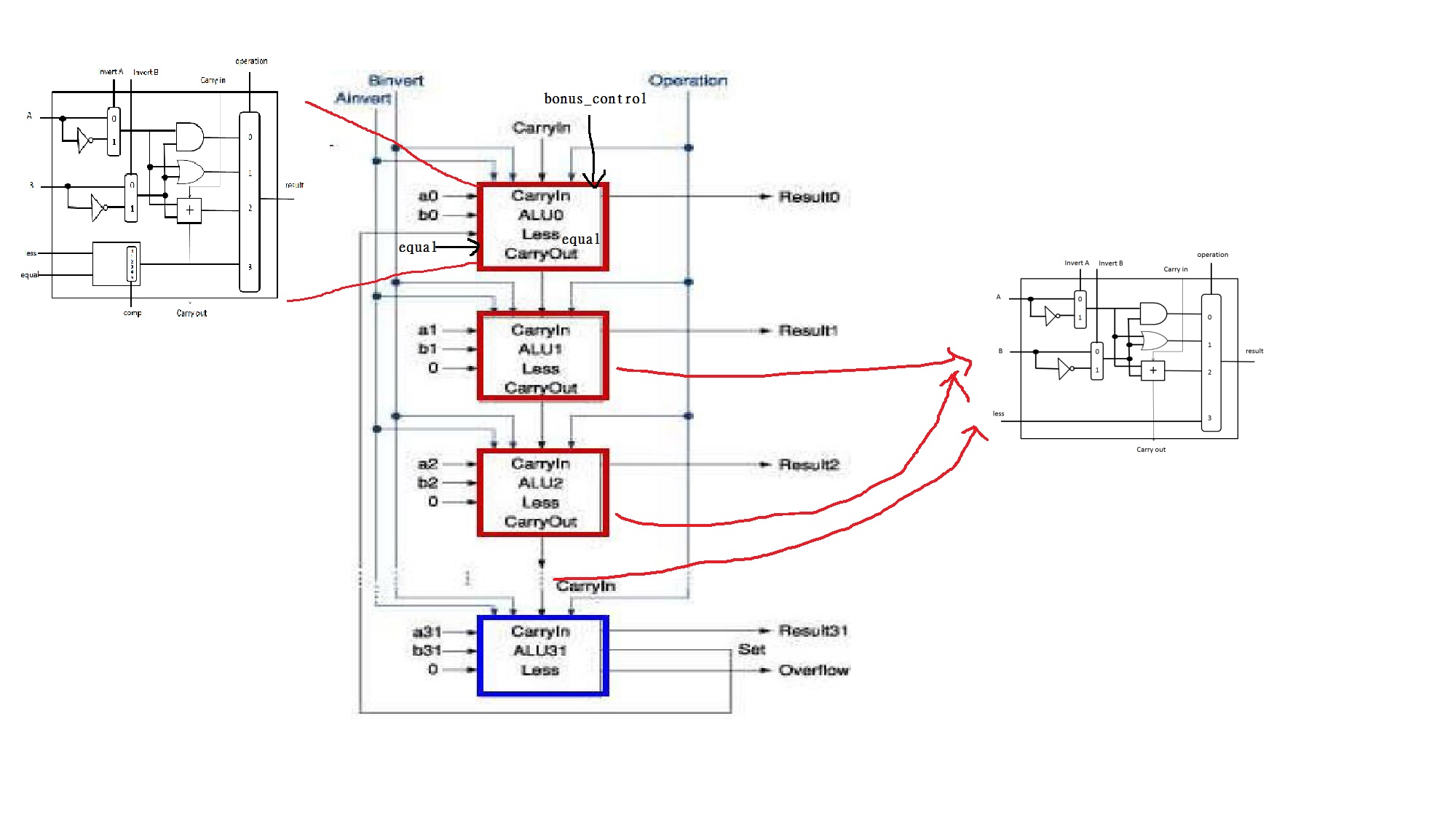
**Computer Organization**

**Architecture diagram:**

**Detailed description of the implementation:**

**Module alu\_top:**

**This is a one bit alu that performs AND,OR,ADDITION,LESS. At the beginning, we use a 2-1 MUX to control invert , then do (src1 XOR A\_invert) and (src2 XOR B\_invert) to get real operand a and b. After we get the real operand a and b, we use a 4-1 MUX to control the operation, so the operation needs 2 bits. When operation=00, performs AND operation. When operation=01, performs OR operation. When operation=10, performs ADD operation. When operation=11, performs LESS operation. After doing the operations above, we will get a one bit result, and pass the FULL ADDER’s carryout to the next edge.**

**Problems encountered and solutions:**

1. **I’m confused with why we need 4 bits for the operation at the beginning, but now I realized that the MSB is the bit that represented A\_invert, and the next bit represented B\_invert.**
2. **The first cin in alu\_top should be B\_invert.**
3. **I forgot to plus one while converting into 2’s complement.**

**Lesson learnt (if any):**

**I have learnt how to design a digital electronic circuit that performs arithmetic and bitwise logical operations on integer binary numbers. Learnt how to convert a logic diagram to Verilog.**