Report for Project 3

Name: Wang Sen

Student ID: 120090234

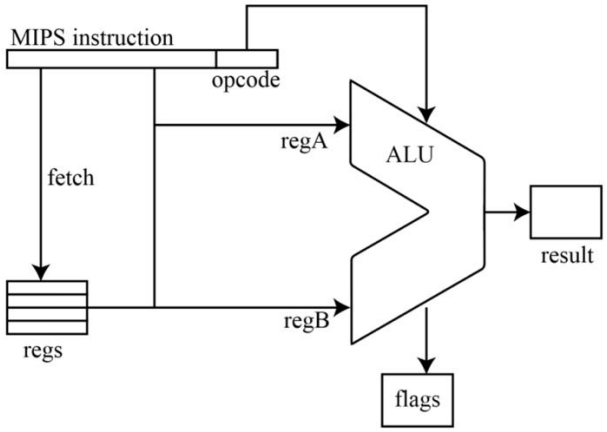
Date: April 10, 2022

1. **Big Picture:**

   ALU, part of CPU in computer, is to do math co-processing. In this project, we are required to use Verilog to implement ALU module and test ALU module by our specific test bench.

   In this project, we need to implement ALU function. In this ALU module, it realizes mathematical operation on two 32-bit singed/unsigned value, such as addition, subtraction, etc.

2. **Data Flow**



   Read instruction → Execute the ALU function→Test bench→Output the result and Output the flag.

3. **Implement:**

   First, we identify opcode and funct to determine what kind of instruction we want to execute. Then we read the numbers in regA, regB or imme to execute the relevant instructions.

4. Tips:

1. The address of regA is 00000, the address of regB is 00001. For convience, I let all rs equal to 00000 and all rt equal to 00001 if they exist. Therefore, rs points to regA and rt points to regB.
2. flag[2] → overflow flag; flag[1] → negative flag; flag[0] → zero flag.
3. For beq and bne: we care about the zero flag more than the result. The result is just the difference of the two input number.
4. For slt, sltu, slti, sltiu: we care about the negative flag more than the result. The result of the sltu and sltiu is just 0 (no less than) or 1 (less than). The result of the slt and slti : if the two input numbers have the same sign, the result will be the difference of the two numbers. If they have different sign, the result is 0 (no less than) or 1 (less than).

   (This may not be rigorous, but it's very convenient. It is acceptable to consider that the result of result is not so important.)
5. For test bench: For each instruction, I have tested the general situation and special situation and made notes in the test_ALU.v.