ALU design

2024年5月8日 17:05

Input

1. **X and Y**:
    - These are two 32-bit input registers that store the two operands for the ALU (X and Y). Each register has 32 binary bits.
2. **opCode**:
    - This is the operation code that determines the operation the ALU will perform. The opCode controls the multiplexer (MUX) to select the appropriate operation.

Calculate Part

1. **Arithmetic and Logic Units**:
    - **Multiplication**: Used to perform multiplication of X and Y. The part has two input, and a c_out to determine the overflow bits, the result will access mux1 and output in result1. The overflow part will access mux2 and output in result2.
    - **Divider**: Used to perform divider of X and Y. The part has two input, and two output. The first output is upper, upper denotes the quotient in a divider operation, which is the integer part of the divider result. The second output is rem, is the leftover part of the divider operation. The upper result will access mux1 and output in result1, the rem result will access mux2 and output in result2.
    - **Adder**: Used to perform addition of X and Y. It uses c_in and c_out to handle carry-in and carry-out.
    - **Subtractor**: Used to perform subtraction of Y from X.
    - **AND Gate**: Performs bitwise AND operation on X and Y.
    - **OR Gate**: Performs bitwise OR operation on X and Y.
    - **XOR Gate**: Performs bitwise XOR operation on X and Y.
    - **NOT Gate**: Performs bitwise NOT operation on X or Y.
2. **Multiplexer (MUX)**:
    - The MUX selects the output of the ALU based on the value of the opCode. The diagram shows two MUXes, which are used to select result1 and result2.
3. **Comparator**:
    - Compares X and Y to determine if they are equal. If they are equal, the output is 1; otherwise, the output is 0.

Output

1. **result1 and result2**:
    - These are the output registers of the ALU, which store the computation results. Depending on the opCode, the MUX selects the corresponding operation result and outputs it to result1 or result2.
2. **equal**:
    - This is the output of the comparator, indicating whether X and Y are equal. If they are equal, this output bit (equal) is 1; otherwise, it is 0.

Process

1. The inputs X and Y are transferred to the ALU through their respective registers.
2. The opCode determines the operation to be performed by the ALU and controls the operation units and the MUX selection.
3. Different operation units in the calculation part perform the corresponding operations based on the values of X and Y.
4. The MUX selects the appropriate calculation result based on the opCode and transfers the result to result1 and result2.
5. If X and Y need to be compared for equality, the comparator outputs the result to equal.

Through these steps, this 32-bit ALU can perform various operations and output the corresponding results to the designated registers.