lab1 report

B10902105 何珮瑄

Modules

I add new modules below:

Adder

Adder has two inputs `data1_in` `data2_in` and one output `data_o`.

Adder takes two 32-bit binary numbers as inputs (`data1_in` and `data2_in`), performs binary addition on them, and outputs the result as a 32-bit binary number on the `data_o` output port.

ALU_Control

ALU_Control has two inputs funct_i, ALUOp_i and one output ALUCtrl_o.

ALU takes control signals (`funct_i` and `ALUOp_i`) and generates the appropriate control signal for the ALU operation as `ALUCtrl_o`. This control signal specifies the type of operation (e.g., addition, subtraction, bitwise AND, XOR, etc.) to be performed by the ALU based on the input conditions.

ALU

ALU has inputs `data1_i` `data2_i` `ALUCtrl_i` , and output `data_o`

It takes two 32-bit binary numbers, performs a specific operation on them based on the control signal (`ALUCtrl_i`), to determine the operation as a 32-bit binary number on the `data_o` output port.

Control

Control has one input `Op_i` and outputs `ALUOp_o`, `ALUSrc_o`, `RegWrite_o`

It generates control signals (`ALUOp_o`, `ALUSrc_o`, `RegWrite_o`) based on the input opcode `Op_i`, determining the type of ALU operation, the source of data for the ALU operation, and whether data should be written back to registers

MUX32

MUX32 has inputs `data1_i` `data2_i` `select_i` and output `data_o`

MUX32" module is a 32-bit multiplexer that takes two 32-bit data sources (`data1_i` and `data2_i`) and selects one of them based on the control signal (`select_i`). The selected data is then provided as the output on the `data_o` .

Sign_Extend

one input `data_i` and one output `data_o`

`{{20{data_i[11]}}` creates a 20-bit concatenation of the most significant bit (`data_i[11]`), and concatenation with the original `data_i[11:0]`. Sign_Extend module takes a 12-bit input, sign-extends it to 32 bits, and provides the sign-extended output on the `data_o`

CPU

Finally, the CPU connects these components together as the final data path given in the spec.

Development Environment:

macOS & iverilog