Computer Architecture Spring Lab 1 Report

R11922211 葉小漓

1. Modules Explanation

- **Control**
  Control module reads OP code as input, and outputs the ALUOP, ALUSrc, RegWrite control signals of the current OP code. ALUOP and ALUSrc depends on the second most significant bit of OP code. If it is set to 1, then ALUSrc = 0 and ALUOP = 10. Else, ALUSrc = 1 and ALUOP = 00. RegWrite is set to 1 in this lab.
- **ALU Control**
  ALU Control module reads ALUOP, func3 and func7 as input, and outputs the instruction number of which instruction to be operated, according to the inputs. (The instruction number ranges from 000 to 111, mapping to the eight instructions defined in spec.)

ALUOP	func3	func7	Instruction No.	Instruction
10	111	0000000	000	and
10	100	0000000	001	xor
10	001	0000000	010	sll
10	000	0000000	011	add
10	000	0100000	100	sub
10	000	0000001	101	mul
00	000	Don’t care	110	addi
00	101	0100000	111	srai

- **Adder**
  Adder module takes pc as input and outputs pc + 4.
- **MUX32**
  MUX32 module takes rs2, immediate and ALUSrc as input and outputs immediate if ALUSrc bit is set or rs2 otherwise.
- **Sign extend**
  Sign extend module reads the 12 bit input and outputs a 32 bit sign extension of the immediate. The is immediate is derived as follows: if the most significant 7 bits is 0100000, then the immediate is the least significant 5 bits. Else, the immediate is the whole 12 bits.
- **ALU**
  ALU module reads op1, op2 and ALU instruction number as input, and outputs result of performing the instruction on op1 and op2. It also outputs a zero signal indicating whether the ALU result is zero or not.
- **CPU**

- CPU module takes a clock and reset signal as input and updates the PC and register file during each clock cycle or resets the values when reset signal os zero. To do this, the CPU connects clock and reset signal to the input of PC and Registers.
- The output of Adder is connected to the PC input, the output of PC is connected to the instruction address port of Instruction_Memory and the input port of Adder.
- The output of Instruction_Memory is selectively connected to the RS1, RS2 and RD address port of Registers, and OP code port of Control, the input of Sign_Extend, and the func3 and func7 port of ALU_Control.
- The ALUOP wire connects the ALUOP port of Control and the ALUOP port of the ALU_Control.
- The ALUSrc wire connects the Control's and MUX32's ALUSrc port.
- The RegWrite wire connects the Regwrite_i port of Registers and the RegWrite port of Control.
- The immd wire connects the output of Sign_Extend and the immd port of MUX32.
- The ALU_ctrl wire connect the output of ALU_Control and the ALU_ctrl port of ALU.
- The ALU_res wire connects the output of ALU (ALU result port) to the RD data port of Registers.
- The rs1 wire connects the RS1 data port of Registers to the op1 of ALU.
- The rs2 wire connects the RS2 data port of Registers to the rs2 port of MUX32.
- The mux_out wire connects the output of MUX32 to the op2 of ALU.
- Finally, a zero wire is defined to connect to the zero port of ALU.

2. Development Environment

- OS: Ubuntu 22.04
- Compiler: iverilog