COMPUTER ARCHITECTURE HOMEWORK #4

B07902143 陳正康

Module Explanation

(1) Control
Decodes instruction and generate correct control signal for ALU, multiplexers, registers and memories. In particular, in my implementation, control module use 6-bit opcode as only input, and passes ALUOp signals to ALU Control module rather than directly to ALU. Also ALUSrc and RegWrite is output for ALU and register file.

(2) ALU Control
Reads ALUOp signal generated by Control module, and funct7 and funct3 field from instructions, and then generates ALU control signal (ALUCtl) to specify the kind of calculation ALU should do.

(3) Sign Extend
Sign-extends the 12-bit immediate value in instruction into 32-bit number. For shift operations, extend as they normally do, though only lower 5 bits of the sign-extended result is used. The output should have the same sign as input.

(4) ALU
Input Op1, Op2 and ALUCtl, output Res and Zero. ALU does the calculation specified by ALU control signal. ALU inputs two operands and outputs the result. There may be arithmetic operations, logic operations, set-if-less-than, etc.

(5) PC
Input clock signal, reset bit, start bit, and next cycle PC, and outputs the PC of current cycle. Program Counter is a register storing address of the instruction being executed currently. PC changes upon every positive edge of clock signal if start bit is on. This module changes its internal register "pc_o" at positive edge of clock signal. When reset signal is set, PC is reset to 0.

(6) Instruction Memory
Input address and output instruction. Instruction memory holds all instructions. Instruction specified by address in PC is read out for decoding, then put into execution, control, etc.

(7) Register File
Inputs clock, Rs1addr, Rs2addr, RDaddr, RegWrite, output Rs1Data Rs2Data. Register file accommodates all the registers needed in RISC-V. Input signals Rs1addr and Rs2addr indicate which register to be read out, and RDAddr indicates which to be written. Specified register is modified upon positive edge of clock if RegWrite control signal is on.

(8) Testbench
Flips clk signal every CYCLE_TIME/2 for executing instructions sequentially. At positive edge of clk, PC changes and new instruction is read. At each cycle the value in registers are printed out.

(9) CPU
Input clock, reset and start bit. Clock, reset and start bit are passed to registers like PC, register file. All modules are connected in CPU. Necessary MUXes are added, like alu_op2 for ALU may be output of sign extend module or rs2 from register file, depending on ALUSrc.

Development Environment

Mainly ModelSim on Windows is used on development stage, but the source code is also tested on Linux with iverilog.