Computer Architecture HW#4

B03901078 蔡承佑

November 20, 2017

1 Construct Essential Elements

There are several modules are todos of this homework, listed as following.

1. `Adder.v`, which performs `PC=PC+4` operation.
2. `Control.v`, which translates instruction codes into control signals.
3. `ALU_Control.v`, which translates function bits and `ALU_Op` codes into `ALU_Ctrl` signals.
4. `ALU.v`, which performs some basic arithmetic and logical operations. e.g. add, sub, and, or, mul.
5. `Sign_Extend.v`, which extends a 16-bit immediate value into a 32-bit value, while keeping the sign.
6. `MUX32.v` and `MUX5.v`, which output the appropriate data according the selecting signal.

2 Connecting the Wires

With all modules completed, now the remaining jobs are connecting these wires.

```
wire   [31:0] inst, ALU_in1, currentPC, nextPC;
wire          RegDst, ALUSrc, RegWrite, Zero;
wire   [1:0]  ALUOp;
wire   [2:0]  ALUCtrl;
wire   [31:0] RSData, RTData, RDData, imm32;
wire   [4:0]  writeReg;
```

First, we need a clock for execution. At the clock edge, the CPU fetches new instruction from memory according to the PC, and the PC is duplicated one to the Adder to get next PC. When new instruction is fetched, the instruction is sent to `Control` for control signals, `Registers` for register values, `Sign_Extend` for extended immediate value, `ALU_Control` for ALU_Ctrl signal. Then following the readings of `Control` we can decide whether r_t data or immediate value is used, which operation should ALU performs, whether the value in registers should be replaced, etc. After all of these are done, waiting for the next edge and fetch new instruction based on the PC calculated in the last cycle, and start a new cycle.