

計算機結構 hw3
資工三 t08902109 賈成鎬

Code environment:

作業系統為 windows10, 使用 iverilog 來編譯寫的程式。

在 ./code 下執行

iverilog [-o output-file] *.v

vvp output-file [> output.txt]

Moudle implement explanation:

(1) .Adder.v

兩個 input 為 32bits 的 data1_i 和 data2_i, output 為 32bits 的 data_o.

(2) ALU.v

input 為兩個 32bits 的 data1_i 和 data2_i, 和一個 3bits 的 ALU_Control_i, output 為 32bits 的 data_o, 和 1bit 的 zero。

ALU_Control_i 控制 ALU 的運算類型, 對應如下:

```
begin
  case (ALUCtrl_i)
    3'b001: data_reg = data1_i + data2_i;
    3'b010: data_reg = data1_i - data2_i;
    3'b011: data_reg = data1_i * data2_i;
    3'b100: data_reg = data1_i & data2_i;
    3'b101: data_reg = data1_i | data2_i;
  endcase
end
```

如果 data_reg 的結果為 0 zero = 1, 否則 zero = 0.

(3) MUX32.v

input 為兩個 32bits 的 data1_i 和 data2_i, 和一個 1bit 的 Select_i, output 為 32bits 的 data_o.

Select_i 決定挑選哪個值作為輸出, 如果 select_i 為 0 選 data1_i, 如果 select_i 為 1 選 data2_i.

(4) Sign_Extend.v

input 為兩個 32bits 的 inst_i, output 為 32bits 的 imm_o.

取 inst_i[31:20] 給 imm_o[11:0], 判斷 inst_i 最高位的值, 若最高位為 1, 則 imm_o[31:12] 全為 1, 如果最高位為 0, 則 imm_o[31:12] 全為 0.

(5) Control.v

input 有一個 7bits 的 op_i, output 為 2bits 的 ALU_Op, 1bits 的 ALUSrc_o 和 RegWrite_o.

根據 op_i 的值給 ALU_Op, ALUSrc_o 和 RegWrite_o 賦值, 結果如下:

```
3
4 assign ALUOp_o = (Op_i == 7'b0110011)? 2'b01 : 2'b10;
5 assign ALUSrc_o = (Op_i == 7'b0110011)? 1'b0 : 1'b1;
6 assign RegWrite_o = 1'b1;
7
```

(6) ALU_Control.v

input 為一個 32bits 的 inst_i,一個 2bits 的 ALUOp_i, output 為 3bits 的 ALUCtrl_o.
判斷結果如下:

```
assign ALUCtrl_o = ALUCtrl_reg;

always @(inst_i or ALUOp_i)
begin
    if (ALUOp_i == 2'b10)
    begin
        ALUCtrl_reg = 3'b001;
    end
    else if (inst_i[14 : 12] == 3'b000)
    begin
        if (inst_i[31 : 25] == 7'b0000000)
        begin
            ALUCtrl_reg = 3'b001;
        end
        else if (inst_i[31 : 25] == 7'b0100000)
        begin
            ALUCtrl_reg = 3'b010;
        end
        else if (inst_i[31 : 25] == 7'b0000001)
        begin
            ALUCtrl_reg = 3'b011;
        end
    end
    end
    else if (inst_i[14 : 12] == 3'b111)
    begin
        ALUCtrl_reg = 3'b100;
    end
    else if (inst_i[14 : 12] == 3'b110)
    begin
        ALUCtrl_reg = 3'b101;
    end
end
```

(7) CPU.v

input 為一個 1bit 的 clk_i,和 rst_i 和 start_i.
用 inst 和 inst_addr 來記錄當前指令和指令地址。
最後按照 datapath 把個 module 連接起來。