

Homework #4

資工三 B04902009 蕭千惠
16th November, 2017

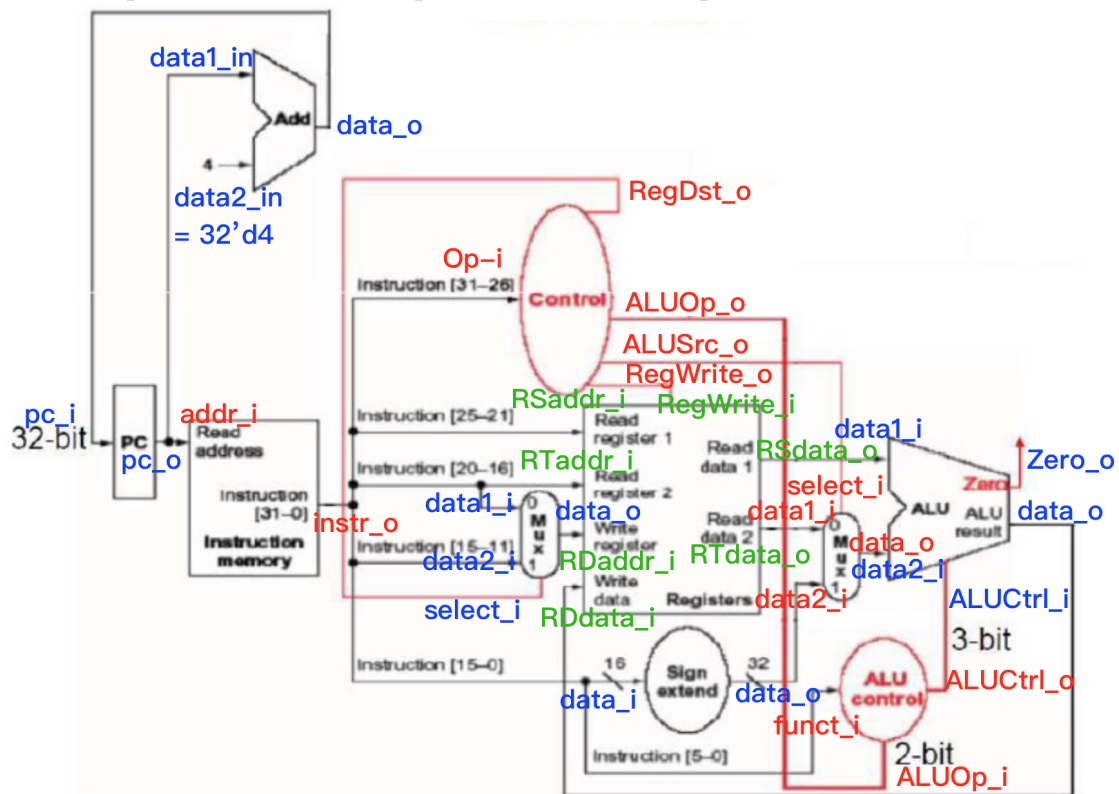
1 Coding Environment

Apple: OS X EL Capitan

2 Module implementation explanation

2.1 CPU

畫出 data path 來釐清誰的 output 會連接到誰的 input



2.2 Adder

$$\text{data}_o = \text{data1_in} + \text{data2_in}$$

2.3 Control

依照下表判斷 Opcode 對應到哪個指令，並依照該指令輸出對應的 Control Signal

Instruction	Opcode (Input)	ALUOp (Output)	RegDst (Output)	ALUSrc (Output)	RegWrite (Output)
R-type	000000(0x00)	11(Rtype)	1	0	1
addi	001000(0x08)	00(add)	0	1	1

2.4 ALU

依照下表的 ALU control 值決定要執行哪一種運算

	and	or	add	sub	mul
ALU control	000	001	010	110	101

```
data_o = (ALUctrl_i == 3'b000)? data1_i & data2_i :
          (ALUctrl_i == 3'b001)? data1_i | data2_i :
          (ALUctrl_i == 3'b010)? data1_i + data2_i :
          (ALUctrl_i == 3'b110)? data1_i - data2_i :
          (ALUctrl_i == 3'b101)? data1_i * data2_i :
          32'd0;
```

2.5 ALU_Control

依照下表的 Opcode 和 function code 決定輸出的 ALU ctrl 為何

R-type (Opcode=0x00, ALUOp=11)

Instruction	Opcode	ALUOp	Func_code	ALU_ctrl
add	000000(0x00)	11(Rtype)	100000(0x20)	010
sub	000000(0x00)	11(Rtype)	100010(0x22)	110
and	000000(0x00)	11(Rtype)	100100(0x24)	000
or	000000(0x00)	11(Rtype)	100101(0x25)	001
mul	000000(0x00)	11(Rtype)	011000(0x18)	101(Self_defined)

I-type (no Func_code)

Instruction	Opcode	ALUOp	Func_code	ALU_ctrl
addi	001000(0x08)	00(add)	X	010

```
assign ALUctrl_o = (ALUOp_i == 2'b00)? 3'b010 : // addi
                   (ALUOp_i == 2'b11 && funct_i == 6'h20)? 3'b010 : // add
                   (ALUOp_i == 2'b11 && funct_i == 6'h22)? 3'b110 : // sub
                   (ALUOp_i == 2'b11 && funct_i == 6'h24)? 3'b000 : // and
                   (ALUOp_i == 2'b11 && funct_i == 6'h25)? 3'b001 : // or
                   (ALUOp_i == 2'b11 && funct_i == 6'h18)? 3'b101 : // mul
                   3'b000;
```

2.6 MUX5 & MUX32

如果 select 是 1 (True)，輸出 data1，反之則輸出 data2

```
data_o = (select_i == 1'b0)? data1_i : data2_i;
```

2.7 Sign_Extend

輸出為 32 bits，而輸入為 16 bits。

Sign extension 要做的就是將輸入的 Most significant bit 複製到不足的 16 bits 上。

```
data_o = {{16{data_i[15]}}, data_i};
```