

# Lab3

許峻源 A103527

1.

新增了“Mux2to1\_1bit”, “Mux2to1\_3bit” 兩個 module，分別用於 Branch\_mux 和 RegDst\_mux。因為不同其他 mux 的 input 是 16 bit 的 data，故把他們分為不同的 module。

2.

instruction	instrt_op	branch	branch_ty	jump	ALUsrc	ALUop	MemWrit	MemRead	MemToRe	RegDst	RegWrite
R	0_0_0	0	0	0	0	1_0	0	0	0	1	1
addi	0_0_1	0	0	0	1	0_0	0	0	0	0	1
lui	0_1_0	0	0	0	0	1_1	0	0	0	0	1
lw	0_1_1	0	0	0	1	0_0	0	1	1	0	1
sw	1_0_0	0	0	0	1	0_0	1	0	0	0	0
beq	1_0_1	1	0	0	0	0_1	0	0	0	0	0
bne	1_1_0	1	1	0	0	0_1	0	0	0	0	0
jump	1_1_1	0	0	1	0	0_0	0	0	0	0	0

3.

	input		output	
instruction	ALUOp	funct	ALU_oper	FUResult
addi, lw, sw	0_0	x	0_0_1_0	0_0
beq, bne	0_1	x	0_1_1_0	0_0
lui	1_1	x	0_0_0_0	1_1
add	1_0	0_0_0_0	0_0_1_0	0_0
sub	1_0	0_0_0_1	0_1_1_0	0_0
and	1_0	0_0_1_0	0_0_0_0	0_0
or	1_0	0_0_1_1	0_0_0_1	0_0
nor	1_0	0_1_0_0	1_1_0_0	0_0
slt	1_0	0_1_0_1	0_1_1_1	0_0
sll	1_0	0_1_1_0	0_0_0_1	0_1
srl	1_0	0_1_1_1	0_0_0_0	0_1

4.

- 一般 PC：由原本的 PC 值加 2，等於跳到下一個指令。
- Branch PC：將 immediate 值 左移 1 bit (\*2)，算出 offset 後，再加上一個 PC 的值，變成 Branch PC 的 address。
- Jump PC：將 jump 指令後的值，左移 1bit，由 13bits 變成 14bits，開頭再加上一個 PC 的前兩個 bits，變成完整的 16-bit address

5.

這次在做 Lab3 時，遇到一些狀況，雖然結果是正確，但在幾次 simulate 的過程中，會跳出一些我不清楚問題在哪的 error，網路上也幾乎沒有資料，幸好這些 error 不影響結果的正確性。