

Computer Organization 2023 Lab 4

Finished part:

根據測試，我的 code 可以跑過所有公開測資

(A) basic score: 75 / 75

Congratulation. You pass TA's pattern

Problems you met and solutions:

我遇到了兩個問題，分別是第一次跑的時候只有 25 分，lw, sw, bne, beq, jump 全部都錯，這個的解法是我忘記他們不是 R-type，所以不小心在 ALU_Ctrl.v 的地方將他們的 OP code 當成 function field 加進去，所以只要是新加的指令都會錯，解決方法是把他刪掉，不要加 function field，只要用 OP code 就可以了；另一個問題是，在我解決這個問題後，發現只要是 bne 就都會錯，這個是我 debug 最久的地方，花了 7 個多小時，重複開了 7 個專案，但是有時候會 bne 全部都錯，但是有時候卻可以拿 75 分，讓我一度對 vivado 失望透頂，但最後發現問題是在 ALUOP_i 的地方，我忘記在 ALUOP_i 判斷右邊的 3 個 bit 加上 3' b，而唯一一個(除了 lui)ALUOP 是 1 開頭的就只有 bne，所以我猜測有時候 vivado 會將這段 code 解讀成十進位的 110，有時候又解讀成二進位的 110，所以有時候成功有時候會失敗。

```
(ALUOp_i == 3'b011) ? 4'b0010 :  
(ALUOp_i == 3'b000) ? 4'b0010 :  
(ALUOp_i == 3'b001) ? 4'b0110 :  
(ALUOp_i == 3'b110) ? 4'b0110 :  
(ALUOp_i == 3'b101) ? 4'b0010 : 4'b0000;
```

If you create additional module, please give a short explanation here:

我有增加 2 個 module，分別是 ALU_1bit.v 以及 Full_adder.v，增加的理由是因為這個是我在 HW2 就寫好的，我懶得再去重新寫一個只用到 32-bit 的 ALU，所以我是用 HW2 寫出來的 32 個 1-bit ALU 兜出來的 32-bit ALU 來實作這次作業。