

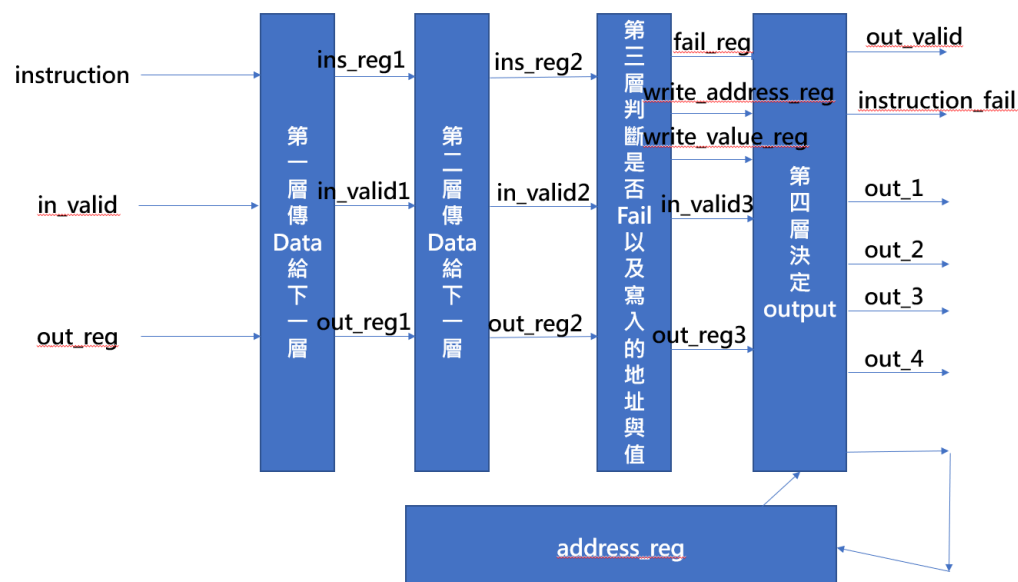
DCS HW04 Report

109511207 電機 13 蔡宗儒

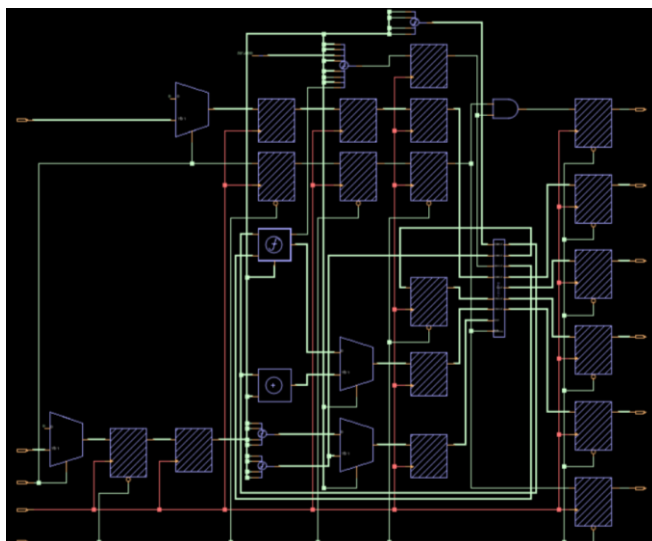
(一) 如何設計作業

切 Pipeline，我切的方式和助教提供的方法不一樣。我一樣分成四層，但第一層跟第二層都是將資料傳到下一層，第三層則判定是否有不合法的指令，並用一條 fail_reg 存起來供下一層使用。第三層還判定了需寫入的地址與計算需寫入的值為多少，然後一樣傳入下一層，基本上我的第三層就是將助教提供的做法的第二與第三層合併起來，這麼做的原因不外乎是省面積，因為能寫較少 DFF。而最後一層就是判定 fail_reg 是否為 high，是的話將 instruction_fail 拉起來，並將 4 個 out 輸出為 0，不是的話則根據第三層傳入的資料來更改暫存器的值，再用 out_reg 的給的 address 抓出每個暫存器的值給 4 個 out。

(二) 架構圖



附上 nSchema 圖



(三) 遇到的困難與解決方法

如何少寫很多 Case

跟同學討論時發現大家遇到的問題都是寫了很多 Case，而事實上助教只有給 6 個 address，但 5 個 bits 的 address 會有 32 種可能，就會導致一些浪費。而我剛好發現可以把每個 address 的第 1 跟第 3 個 bit 當成最右邊的兩個 bit，第 4 跟第 0 個 bit 做 Xor 當成第 2 個 bit，這樣就能完美兜出 0~5，我就能宣告暫存器的地址為 0~5，在找地址時就不用寫很大的 Case 去做判斷了。如下圖。但這只適用於 out_reg 給的地址不會有這六個地址以外的可能，不然就會爆掉、抓錯值，也因為助教有說 out_reg 會是已知的值我才敢這麼做。

```
assign out_1_comb = (fail_reg) ? 0 : address[{out_reg3[4]^out_reg3[0],out_reg3[3],out_reg3[1]}];
assign out_2_comb = (fail_reg) ? 0 : address[{out_reg3[9]^out_reg3[5],out_reg3[8],out_reg3[6]}];
assign out_3_comb = (fail_reg) ? 0 : address[{out_reg3[14]^out_reg3[10],out_reg3[13],out_reg3[11]}];
assign out_4_comb = (fail_reg) ? 0 : address[{out_reg3[19]^out_reg3[15],out_reg3[18],out_reg3[16]}];
```

而我也設了一個 32bit 的變數 judge(只有在 address 為 8、16、17、18、23、31 時會是合法的，所以我讓 judge 的這 6 個 bit 為 1，其他 26 個 bit 都為 0，以此來判定是否合法)用來判斷 Rs、Rt、Rd 是否有給予合法的值。如下圖。

```
assign judge = 32'b10000000100001110000000100000000;
```

(四) 心得

這次的作業其實沒有花很多時間，因為有很多考試剛好都在最近，有點喘不過氣，所以也沒有花什麼時間去壓面積，甚至很擔心測助教的測值就不會過了 QQ。而且用了上面我自認為較好的方法去優化也沒有和其他同學的面積差很多，我認為原因應該是上面的寫法最後 EDA Tool 還是會合成出 MUX，與原本用 Case 一樣，優化的地方其實僅有把 address 的 bit 數變小而已。但是這麼做還是可以少寫一些 Case，我甚至聽到有同學寫了 19 個 Case。

(五) Area

```
Combinational area:          38576.261613
Buf/Inv area:                4454.049760
Noncombinational area:      31234.896294
Macro/Black Box area:       0.000000
Net Interconnect area:      undefined (No wire load specified)

Total cell area:             69811.157907
Total area:                  undefined
```