Report_dcs015 110652041 黃鵬宇

設計方法

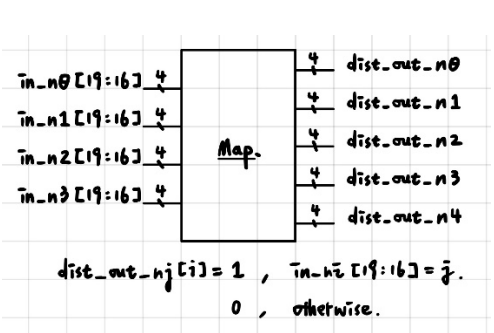
因為這次作業的組合邏輯是四個輸入傳到五個輸出，所以我電路設計上做一個 map 紀錄傳送情形，再判斷各個輸出的 condition，接著透過 priority mux 去決定資料傳送到哪個對應的輸出，並同時將對應的 ack 訊號拉起來。

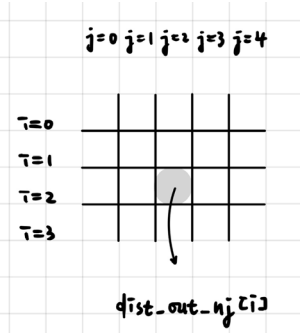
我追求更好 performance 使用的技巧為簡化 condition 那邊的組合邏輯，先將 map 當中去到相同輸出的訊號加起來，代表有 n (介於 0 到 4)個輸入想去那個輸出，若 sum 起來的值小於等於 2 就讓 condition 等於 sum，反之則直接給值 10_2。

```
98   always_comb begin
99       //dest0
100      //out_n0[17:16] = 2'b00;
101      if (sum_out0 < 2) begin
102          out_n0[17:16] = sum_out0;
103      end
104      else begin
105          out_n0[17:16] = 2'b10;
106      end
```

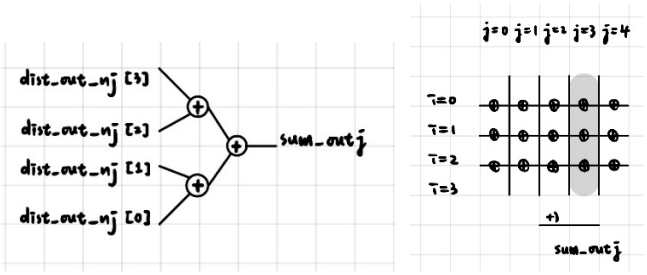
Block diagram (mapping, sum, condition, data & ack)

mapping

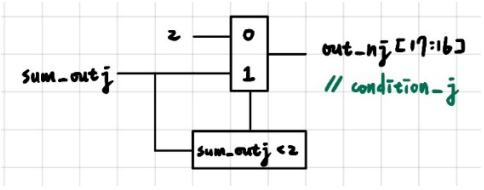




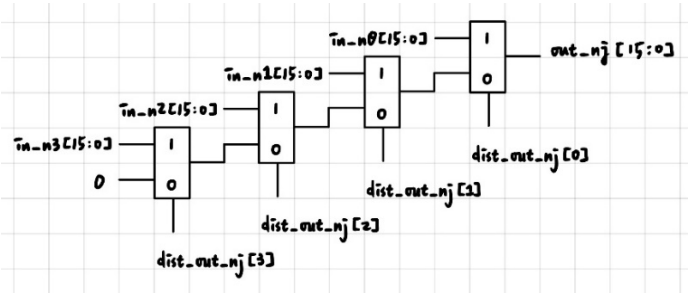
sum



condition



Data & ack



心得

這次 HW 讓我練習實現簡單的組合邏。我覺得以 cell-based design 來說，有良好的 coding style 正確寫出電路跑完模擬只是基本，關鍵在於如何將逐條的 spec.透過反覆思考想出電路架構，和該如何優化電路 performance。這次組合邏輯電路多為平行化操作，也不太會用到太複雜的演算法，在壓面積方面的效果有限。