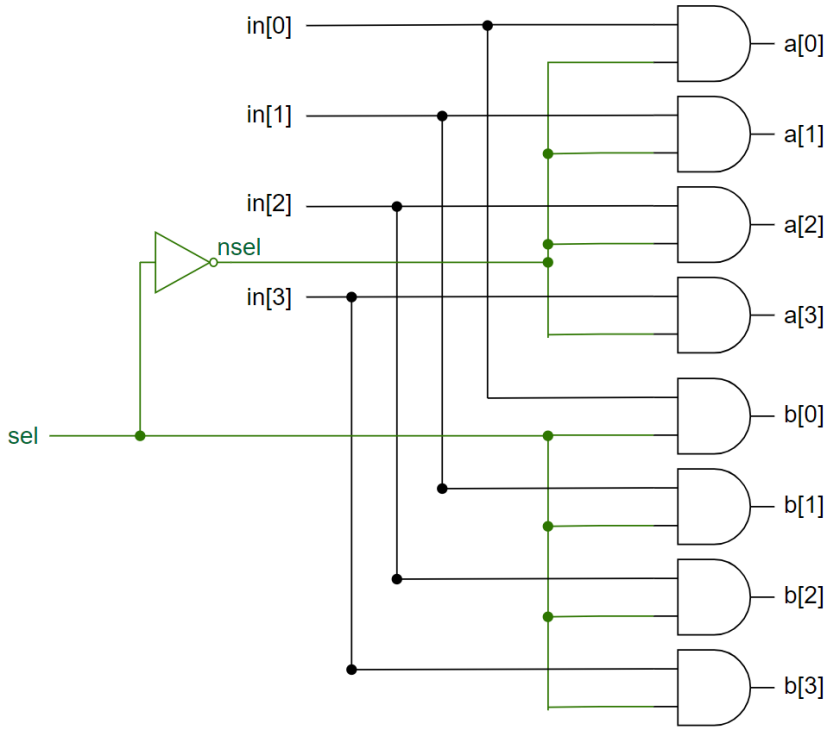
Lab 1 report

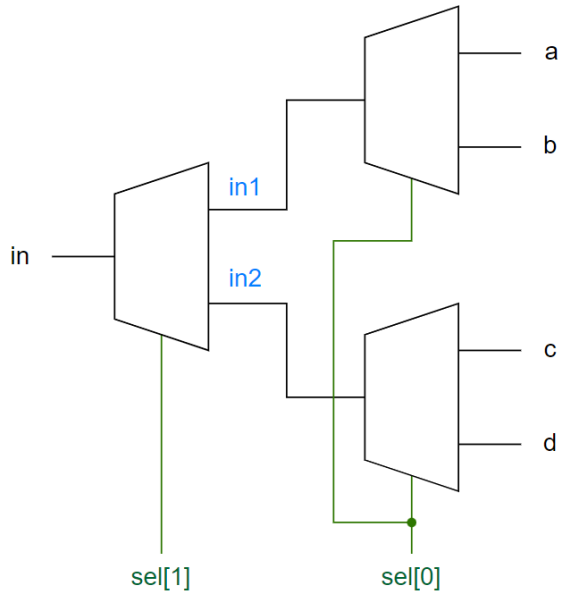
組員：110062221 李品萱
110062213 唐翊雯

I. (Gate-level) 4-bit 1-to-4 de-multiplexer (DMUX)

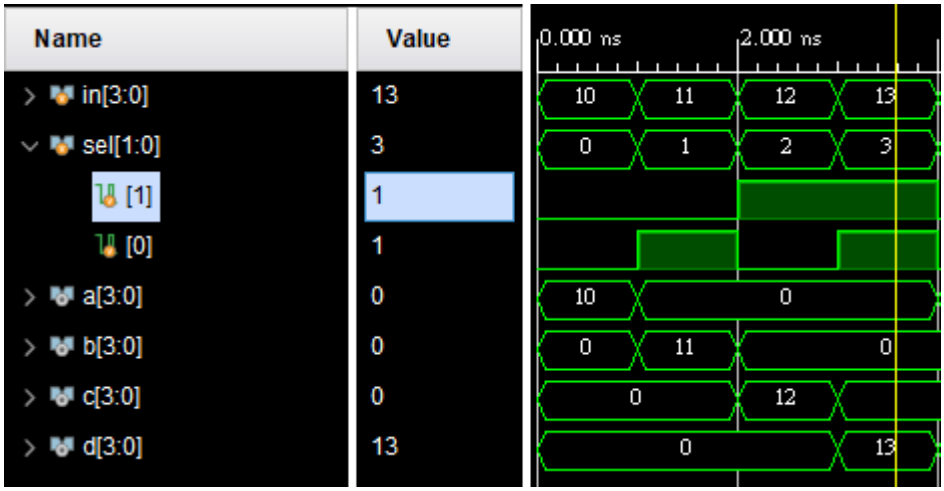
這題要用三個 1-to-2 DMUX 接出 1-to-4 DMUX，所以我們先做出 1-to-2 DMUX 的部分，一個 2-to-1 DMUX 由一個 not gate 與 8 個 and gate 組成，gate-level circuit 如下。



接著使用三個 1-to-2 DMUX 組成一個 1-to-4 DMUX，gate-level circuit 如下，這邊我們會先判斷 sel[1] 是 1 或 0，再判斷 sel[0] 對應到的 output 位置，也就是說，第一個 DMUX 會接出兩條線，然後分別接到兩個 DMUX 決定最後選到哪個訊號，這邊也可以看出 DMUX 的功能與 MUX 相反。

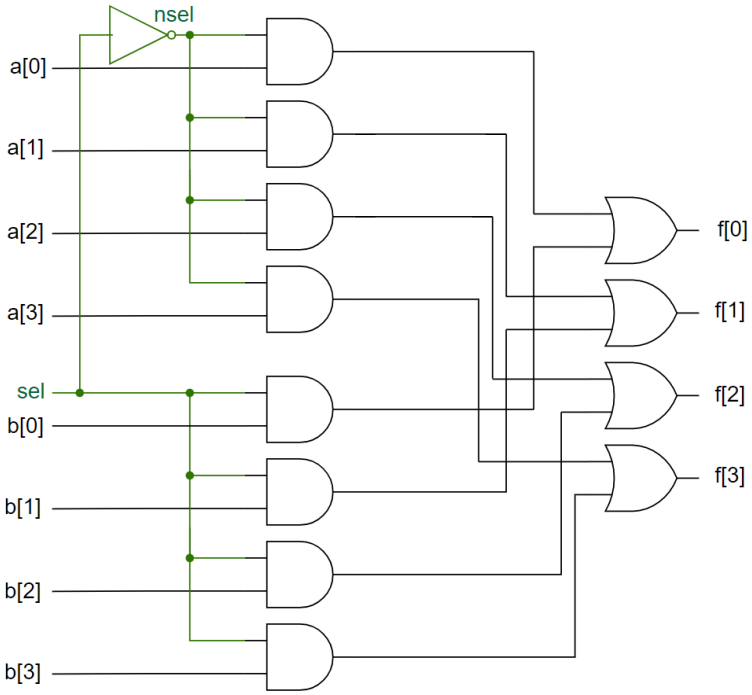


在 simulation 的結果中，我們也測試了每個 sel，sel 為 00、01、10、11 時分別會將 output 送到 a、b、c、d，而未被選到的位置則會輸出 0。

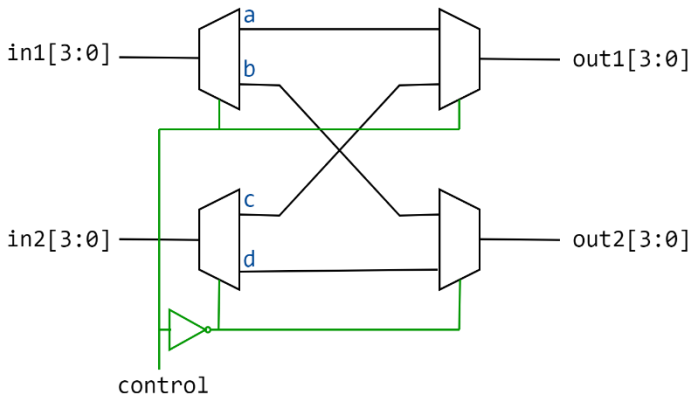


II. (Gate-level) 4-bit simple crossbar switch with MUX/DMUX

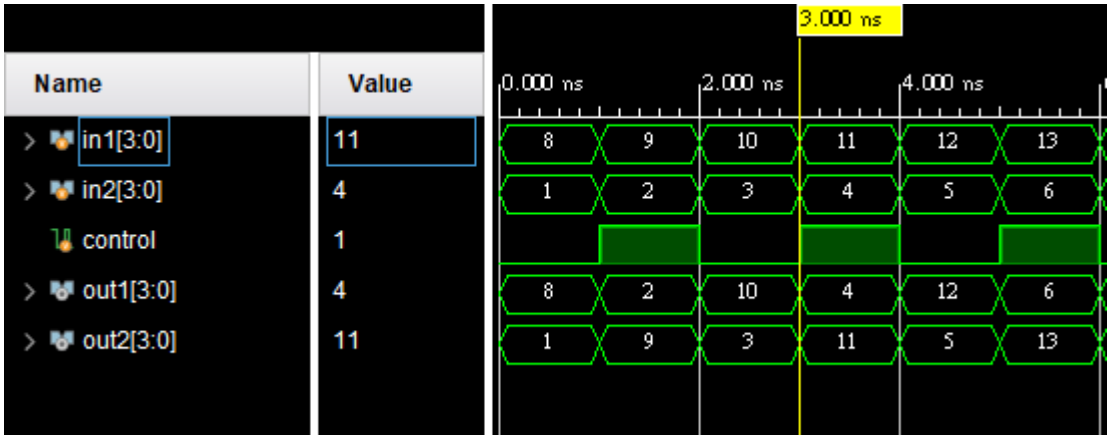
整個 circuit 由兩個 1-to-2 DMUX 與兩個 2-to-1 MUX（皆為 4 bit）組成。DMUX 的 gate-level circuit 如前一題所示，而 MUX 的 gate-level circuit 如下：



在整個 crossbar 中，我們先用 DMUX 決定它們會送到哪一條線，再用 MUX 進行多選一的操作。舉例來說，當 control 是 1 時，in1 會由 b 送到下圖中右下方的 MUX，對於此 MUX 來說 control 是 0，因此它會選擇 wire b 進來的訊號；而對於 in2 接到的 DMUX 來說 control 是 0，因此會由 wire c 接出去，此時對於右上方的 MUX control 是 1，c 會接到 out1。同理可知，當 control 為 0 時，in1 會接到 out1，in2 會接到 out2。

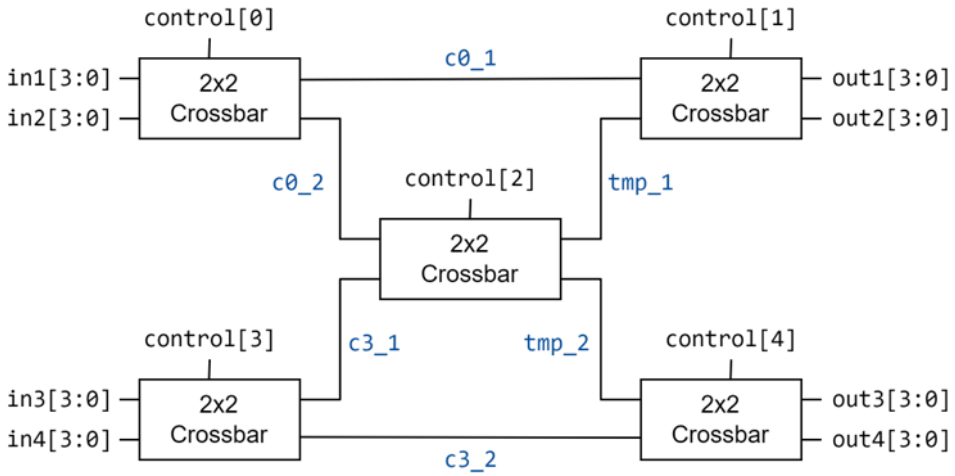


在 testbench 的部分，我們讓 control 在 0、1 交替變換，也可看出前面藉由電路圖得到的結果。



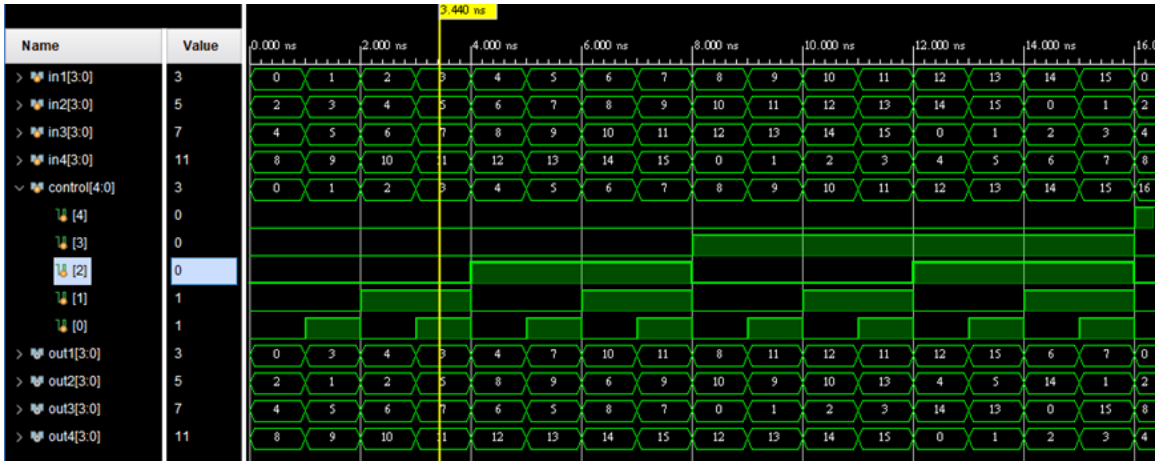
III. (Gate-level) 4-bit 4x4crossbar with simple crossbar switch

這題會用到前面實做過的 2x2Crossbar，首先我們將 input 送進 2x2Crossbar 並用 control 訊號做 select 的動作，而後，c0_2、c3_1 這兩條線接到的訊號會再送進 2x2Crossbar 做一次 select，再藉由 tmp_1、tmp_2 分別送進兩個 2x2Crossbar 並 select 後送到對應的 output。



從下面 simulation 的結果及上面的電路圖中我們可以觀察到，當 control[2] 為 0 時，in1、in2 必不會對應到 out3 或 out4，當 control[2] 為 1 時，in1、in2 必不會同時位於 out1 及 out2 或 out3 及 out4，同理，in3、in4 也不會同時位於 out1 及 out2 或 out3 及 out4，所以它不可能出現的組合如下：

- [(in1, out3), (in2, out4), (in3, out1), (in4, out2)]
- [(in1, out3), (in2, out4), (in3, out2), (in4, out1)]
- [(in1, out4), (in2, out3), (in3, out1), (in4, out2)]
- [(in1, out4), (in2, out3), (in3, out2), (in4, out1)]

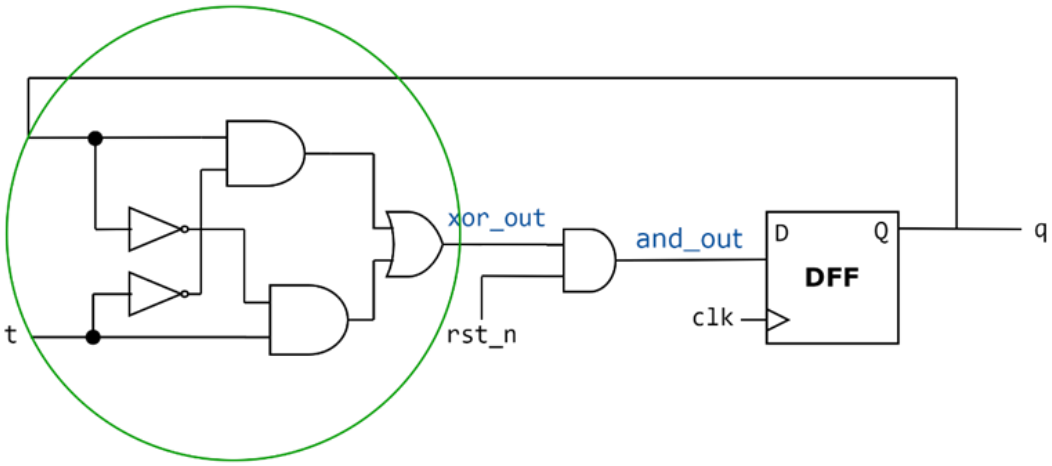


因為這題的 Testbench 主要目的是要確認 control 對各個 input 做怎樣的選擇，所以這部分我們用枚舉的方式跑過一次所有可能的 control，確認不同的 control 訊號組合下出現的結果是正確的。

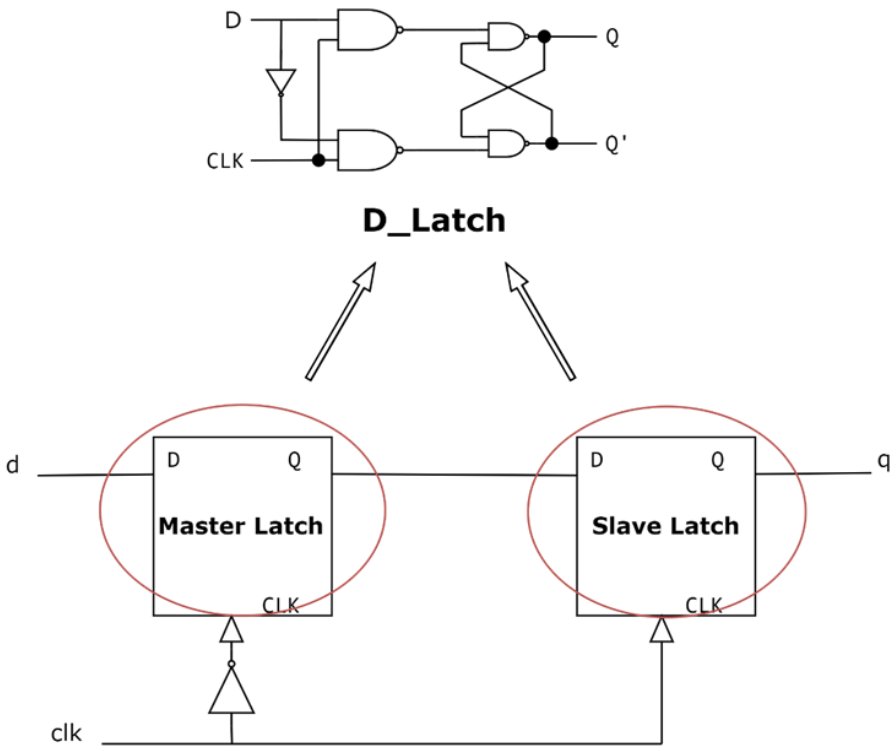
```
19
20      initial begin
21          repeat (2 ** 5) begin
22              #1 control = control + 5'b1;
23              in1 = in1 + 4'b1;
24              in2 = in2 + 4'b1;
25              in3 = in3 + 4'b1;
26              in4 = in4 + 4'b1;
27          end
28          #1 $finish;
29      end
```

IV. (Gate-level) 1-bit toggle flip flop (TFF)

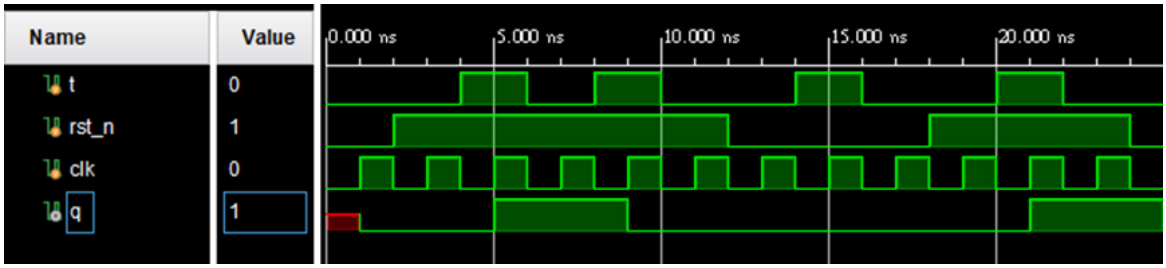
在這題中，我們將 t 跟 q 接上 xor gate 再與 rst_n 做 and，再接上前面實作過的 DFF，而 xor gate 的 gate level circuit 如下圖綠色圈起處所示，而此處 TFF 會將 output Q 再送進一開始的 xor gate，這會使現在的 q 值對下一個 clk 時的 q 值產生影響，這部分我們也可以在後面的 simulation 結果中看到。



其中，由下圖可知 D flip flop 是由 Master Latch、Slave Latch 及 clock 組成，clock 在 low 時會將訊號從 master latch 送至 slave latch，clock high 時再由 slave latch 送出，即為 positive edge 的 DFF。



經過 simulation 後，我們可以從 waveform 中觀察到，當 clk 為 high 且 rst_n 為 1 時，q 與 t 做 xor 的值就會被送出，維持一個 clock cycle；而當 clk 為 high 但 rst_n 為 0 時，q 值會為 0，因為此時相當於把 0 送進 DFF，其結果為 0。此外，在 waveform 中，我們可以看到 q 一開始有紅色的區塊，此為電路尚未給值時呈現的狀態。



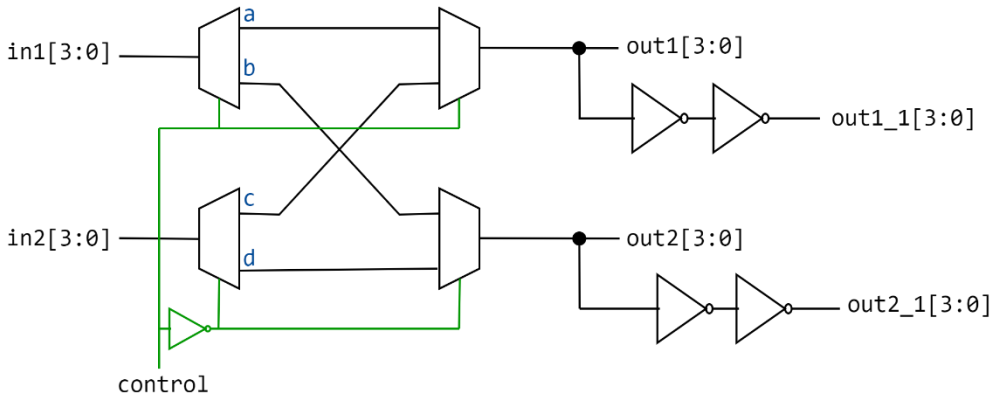
在 testbench 的部分，我們參考了 basic lab DFF 的 testbench，對應不同的 clock 改變 rst_n 及 t 的值，這邊也有刻意使 rst_n 的變化頻率與 t 不同以方便觀察 q 的結果，雖然看起來頗為冗長，但至少有達到我們確認其特徵的目的。

```
initial begin
    // rst_n = 1'b0;
    @(negedge clk) rst_n = 1'b1;

    @(negedge clk) t = 1'b1;
    @(negedge clk) t = 1'b0;
    @(negedge clk) t = 1'b1;
    @(negedge clk) t = 1'b0;

    @(negedge clk) rst_n = 1'b0;
    @(negedge clk) t = 1'b1;
    @(negedge clk) t = 1'b0;
```

v. FPGA demonstration: (Gate-level) 4-bit simple crossbar switch with MUX/DMUX

這一題 Code 的部分只有基於 4-bit simple crossbar 的 code 做一些修改：由於題目規定一個 output 要對應到兩個 LED，這部份我們兩個人對 fan-out 有不同的做法，如下：

我們的第一種作法是在原本的 out1 及 out2 各自再接出兩個 not gate，並再開 out1_1、out2_1 這兩個 output 接原本的 output 做完兩次 not 的結果，這個方法是參考講義上 fan-out 那一段的作法。



在寫 fan-out 的過程中，我們原本的寫法是下圖中註解的部分，但後來在網路上剛好看到語法的資料，發現一樣的事可以精簡成一行，如下圖。

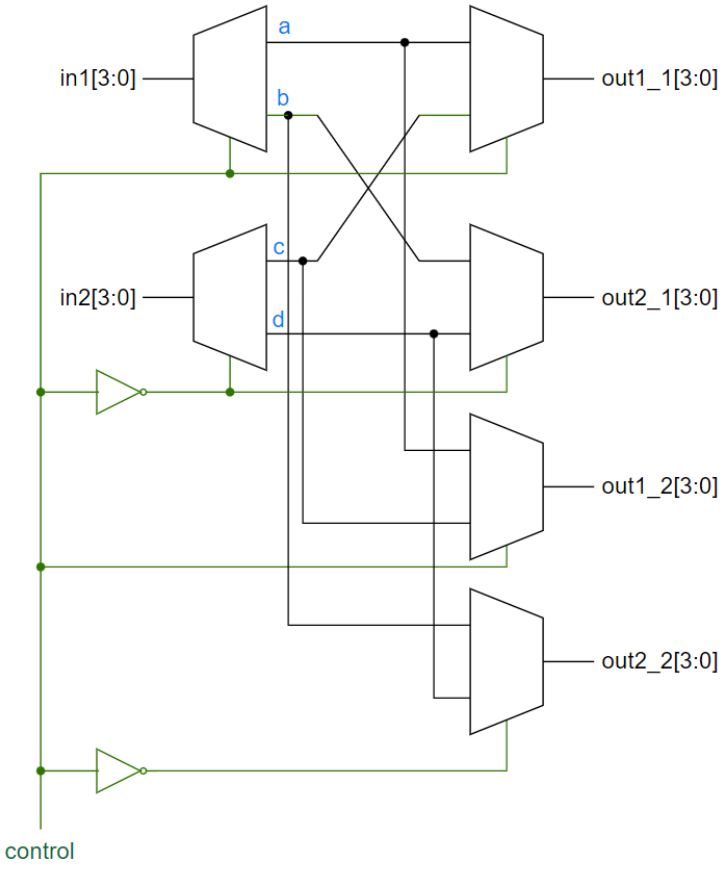
```
not n2[3:0](out1_tmp, out1);

// not n2(out1_tmp[0], out1[0]);
// not n3(out1_tmp[1], out1[1]);
// not n4(out1_tmp[2], out1[2]);
// not n5(out1_tmp[3], out1[3]);
```

第二種方法是我們另外開了兩個 4 bit 的 output 變數：out1_2 與 out2_2，並多使用兩個 MUX，依照原先的 out 1 與 out 2 接線的方式，分別將 DMUX 的 output 接到兩個 MUX 的 input，再分別將兩個 MUX 的 output 接到 out1_2 與 out2_2（於是 out 1 與 out1_2 的值會相同；out 2 與 out2_2 的值會相同），最後再分別對應到 LED 燈上。如下所示：上圖為 4-bit simple crossbar 的 code，而下圖為修改後的 code（這段 code 將 out 1 重新命名為 out1_1、將 out 2 重新命名為 out2_1）。

```
43 module Crossbar_2x2_4bit(in1, in2, control, out1, out2);
44 input [4-1:0] in1, in2;
45 input control;
46 output [4-1:0] out1, out2;
47 wire ncont;
48 wire [3:0] a, b, c, d;
49
50 not n1(ncont, control);
51 Dmux_1x2_4bit d1(in1, a, b, control);
52 Dmux_1x2_4bit d2(in2, c, d, ncont);
53 Mux_2x1_4bit m1(a, c, control, out1);
54 Mux_2x1_4bit m2(b, d, ncont, out2);
55
56 endmodule
```

```
43 module Crossbar_2x2_4bit(in1, in2, control, out1_1, out2_1, out1_2, out2_2);
44 input [4-1:0] in1, in2;
45 input control;
46 output [4-1:0] out1_1, out2_1;
47 output [4-1:0] out1_2, out2_2;
48 wire ncont;
49 wire [3:0] a, b, c, d;
50
51 not n1(ncont, control);
52 Dmux_1x2_4bit d1(in1, a, b, control);
53 Dmux_1x2_4bit d2(in2, c, d, ncont);
54 Mux_2x1_4bit m1(a, c, control, out1_1);
55 Mux_2x1_4bit m2(b, d, ncont, out2_1);
56 Mux_2x1_4bit m3(a, c, control, out1_2);
57 Mux_2x1_4bit m4(b, d, ncont, out2_2);
58
59 endmodule
```

而這個做法的 gate-level circuit 如下：



VI. Summary

這次 Lab 的內容雖然對學過邏設的我們來說不算太陌生，但許久沒碰仍然會感到生疏（例如犯一些像是忘記宣告 wire 是幾 bit 而找不出 bug 的錯），透過這次 Lab，除了複習了一些邏設的知識外，也學到了新的東西，像是如何依照題目要求接出對應的線、設計自己的 testbench、nWave 的使用及將成果在 FPGA 上呈現等等。

這也是我們第一次接觸在沒有 tb 的狀況下寫 code，一開始會有點害怕，不確定自己寫的東西是不是對的，在寫 tb 的時候也會不知道從何下手，但透過參考助教在 basic 的 tb 及自己實作之後，我們漸漸對這部分有點概念，也會開始思考自己寫出來的 tb 測試過後是否能保證我們丟進去 test 的 code 是正確的，對 tb 有更好的了解。

FPGA 的部分也很有趣，透過把 code 的內容燒到板子上之後能夠動手測試我們寫的東西是否正確，看到 LED 正確的亮起來時真的很有成就感。在操作板子的過程中，我們遇到 Auto connect 找不到板板的問題，而後透過到 C:\Xilinx\Vivado\2020.2\data\xicom\cable_drivers\nt64 找到 install_drivers.cmd 執行並重新開機解決。

經過這次 Lab，我們對整個設計測試有更多的了解，也感受到實作過程及結果的有趣，應該算邁入了 Verilog 的已知用火階段。

VII. Contributions

- **Code:**
  (Gate-level) 4-bit 1-to-4 de-multiplexer (DMUX) by 李品萱
  (Gate-level) 4-bit simple crossbar switch with MUX/DMUX by 李品萱
  (Gate-level) 4-bit 4x4crossbar with simple crossbar switch by 唐翊雯
  (Gate-level) 1-bit toggle flip flop (TFF) by 唐翊雯
  FPGA: (Gate-level) 4-bit simple crossbar switch with MUX/DMUX by 李品萱

- **Report:**
  兩人先描述各自在 code 部分負責的題目及畫電路圖，再由唐翊雯統整。