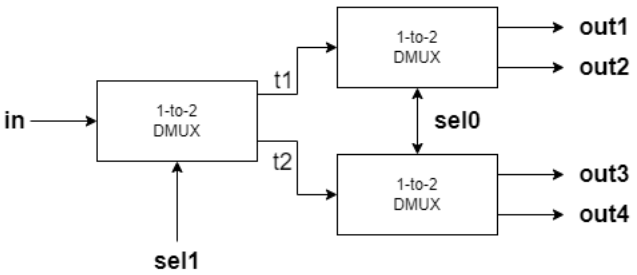
Lab1_Team1_Report

Advanced Question1

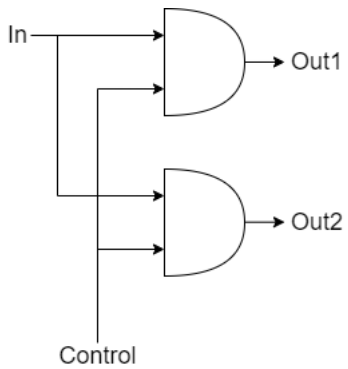
Drawing of the design of Verilog Advanced Question 1

由上述需求，畫成 Block Diagram，如下圖所示:

- Top Module (*圖1-1*)



- Module
  - gate-level circuit of 1-to-2 DMUX (*圖1-2*)



Requirements

1. 以gate-level的形式實作一個 1-to-4 DMUX
2. 依題目指示，會 REUSE 3次 1-to-2 DMUX 來建構

Design Explanation

1. 依照投影片上的題目需求來進行實作，寫出有別於純gate-level((下方*圖5-1*))的設計，而是使用 3 個 1-to-2 DMUX Module 來完成 1-to-4 DMUX的任務，符合題目要求，依此畫出 circuits (*圖1-1*)。

2. 如同 Block Diagram 所示，in、sel1 接入左方的 1-to-2 DMUX，並將兩個 output 接到命名為 t1、t2 的兩條 wire。

- 左方 1-to-2 DMUX: in-> t1、t2

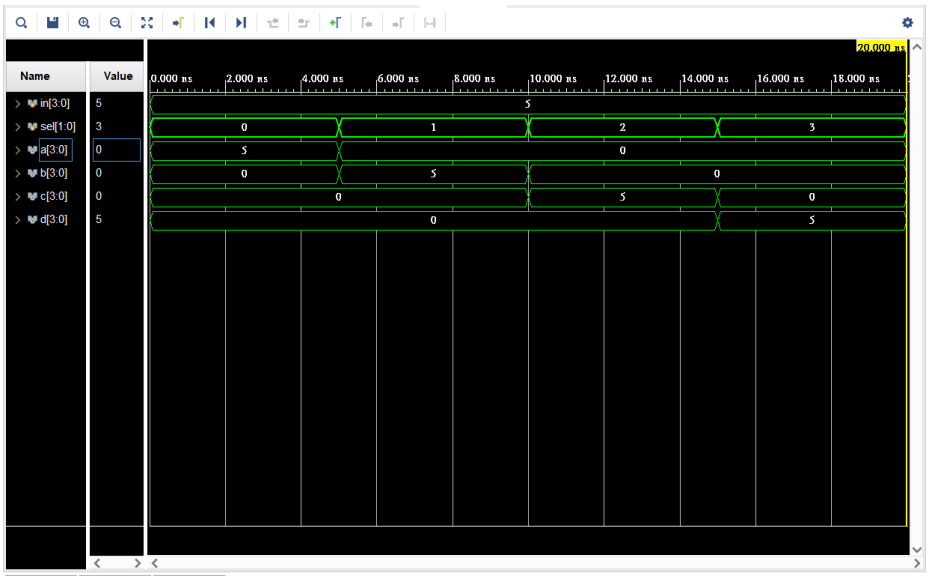
3. 接著再將 t1、t2 分別接入右上方及右下方兩個 1-to-2 DMUX，右上方的 1-to-2 DMUX 依序接入 out1、out2 兩個 output port；右下方的 1-to-2 DMUX 依序接入 out3、out4 兩個 output port。
   - 右上方 1-to-2 DMUX: t1-> out1、out2
   - 右下方 1-to-2 DMUX: t2-> out3、out4

4. 我們的設計將符合以下真值表 (*表1-1*)

sel0	sel1	out1	out2	out3	out4
0	0	0	0	0	in
1	0	0	0	in	0
0	1	0	in	0	0
1	1	in	0	0	0

Testbench Design & Result Explanation

1. 波形圖截圖



2. testbench設計說明

A. 初始化
   - in 為 4'b0101(5)
   - sel 為 2'b00。

B. 因為是2-bit，所以要檢查全部共 4 種 sel 的狀況，來看相對應跑出來的output是否正確，每次循環(5ms)時...

- 固定 5 作為 in 值，便於觀察
- 將 sel 的值固定加1(sel 範圍: 0~3)

3. 波形圖結果說明
   我們將波形圖以及真值表對照，歸納出以下 4 種 sel 狀況之 output:
   - 在 sel = 2'b00 = 0的時候 ➜ out4 = in & 其他 out 為 0
   - 在 sel = 2'b01 = 1的時候 ➜ out3 = in & 其他 out 為 0
   - 在 sel = 2'b10 = 2的時候 ➜ out2 = in & 其他 out 為 0
   - 在 sel = 2'b11 = 3的時候 ➜ out1 = in & 其他 out 為 0

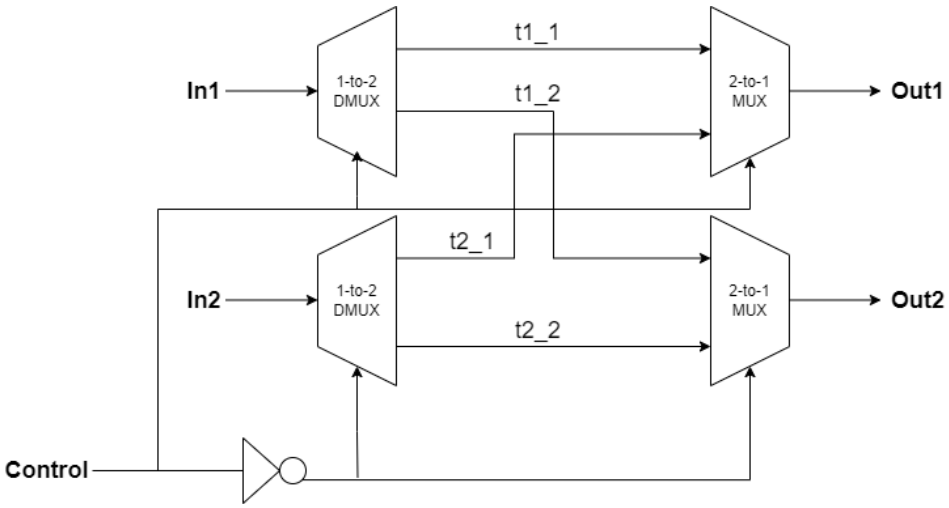
   可以發現結果與我們所預期的結果相吻合。

Advanced Question2

Drawing of the design of Verilog Advanced Question 2

由上述需求，畫成 Block Diagram，如下圖所示:

- Top Module (*圖2-1*)



- Module
  - gate-level circuit of 1-to-2 DMUX (同第一題 / *圖1-2*)

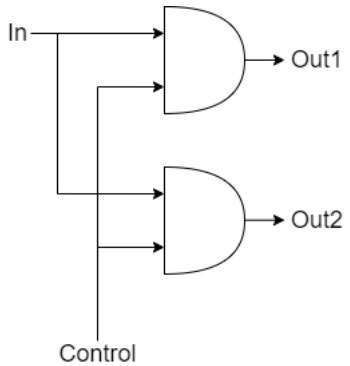
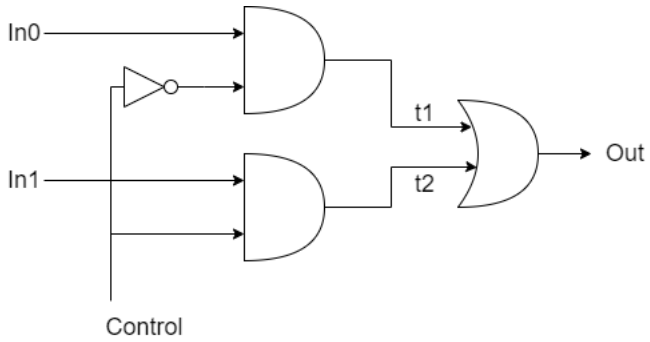


圖2-2

- gate-level circuit of 2-to-1 MUX(*圖2-2*)



Requirements

1. 以gate-level的實作一個 4bit 的 2x2crossbar
2. 依題目指示，會 REUSE 1-to-2 DMUX / 2-to-1 MUX 兩項 Module 來建構

Design Explanation

1. 依照投影片上面的題目需求來進行實作，先設計出要求的 1-to-2 DMUX / 2-to-1 MUX，再使用兩者於crossbar swith中完成 2x2crossbar 的任務，符合題目要求，依此畫出 circuits (*圖2-1*)。

2. 將前一題所做出來的 1-to-2 DMUX 直接拿過來使用；2-to-1 MUX 的部分，依照所學刻畫出吃 input In0、In1 之後，可成功透過 Control signal 選出目標 output 的 Module。

3. 再將以上兩者使用兩者於crossbar swith中 ! 如同投影片的圖上所示，in1[3:0]、control 接入左上 1-to-2 DMUX ；in2[3:0]、 control_bar 接入左下 1-to-2 DMUX，而我們將左上的兩個 output 接到命名為 t1_1、t1_2 的兩條wire；左下的兩個 output 接到命名為 t2_1、t2_2 的兩條wire。

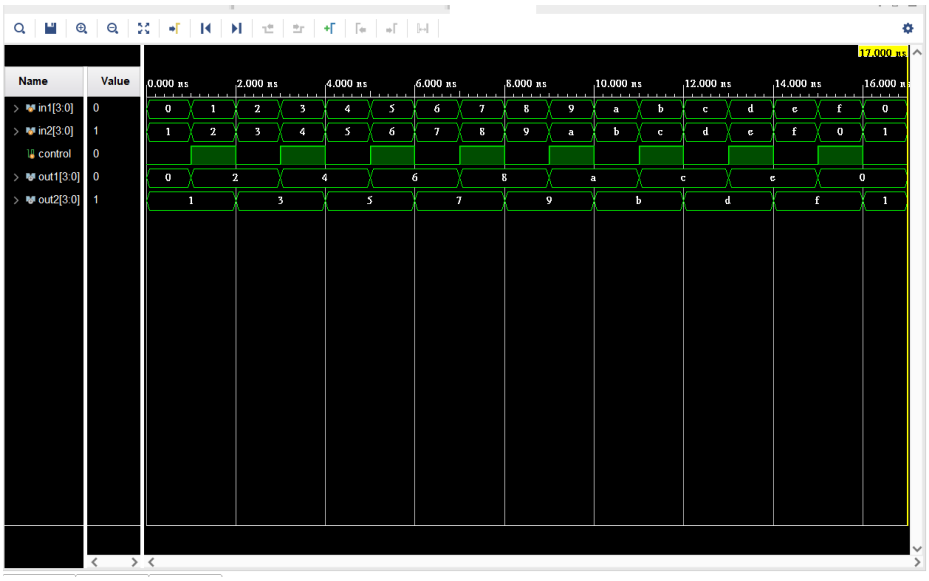
4. 接著，將 t1_1、t2_1 接入右上角的 2-to-1 MUX，並依序接入 out1、out2 到兩個 output port；t2_1、t2_2 接入右下角的 2-to-1 MUX，並依序接入 out3、out4 到兩個 output port。

5. 我們的設計將符合以下真值表(*表2-1*)

control	out1	out2
0	in1	in2
1	in2	in1

Testbench Design & Result Explanation

1. 波形圖截圖



2. testbench設計說明

   A. 初始化
   - 將 in1、in2 分別初始化為4'b0000、4'b0001
   - control 初始化為1'b0。

   B. repeat循環16次檢查相對應跑出來的output是否正確
   - 每次循環(2ms)的時候，我們將 control 的值加1，in1、in2 的值也加1

3. 波形圖結果說明
   我們將波形圖以及真值表對照，確定兩者:
   a. 在control = 0的時候，out1 = in1 & out2 = in2
   b. 在control = 1的時候，out1 = in2 & out2 = in1
   結果與我們所預期的結果吻合。

Advanced Question3

Drawing of the design of Verilog Advanced Question 3

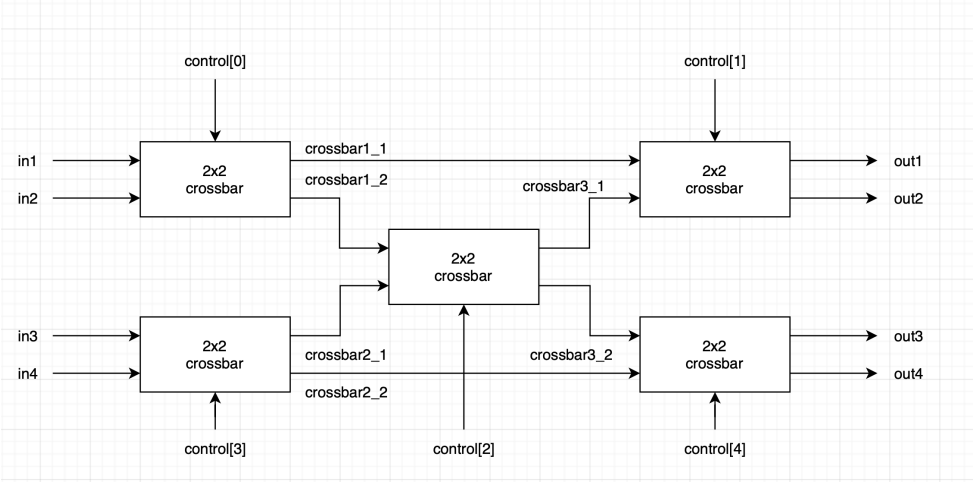


圖3-1

Requirements

1. 以gate-level的形式實做一個4bit的4x4crossbar
2. 重複使用在先前Advanced question1,2以及Basic question1所使用的modules

Design Explanation

1. 我們將前一題所做出來的4bit 2x2 crossbar拿過來並接上相對應的control訊號來符合題目的要求
2. 如同投影片的圖上所示，in1[3:0],in2[3:0],control[0]接入左上角的2x2 crossbar，in3[3:0],in4[3:0],control[3]接入左下角的2x2 crossbar，而我們將左上角2x2 crossbar的兩個output接到命名為crossbar1_1以及crossbar1_2的兩條wire，左下角的2x2 crossbar的兩個output則接到命名為crossbar2_1和crossbar2_2的兩條wire。
3. 接下來再將crossbar1_2,crossbar2_1以及control[2]接到中間的2x2 crossbar，而中間的2x2 crossbar的兩個output再接到命名為crossbar3_1和crossbar3_2的兩條wire。
4. 最後再將crossbar1_1,crossbar3_1,control[1]接入右上角的2x2 crossbar並依序接入out1,out2到兩個output port還有將crossbar3_2,crossbar2_2,control[4]接入右下角的2x2 crossbar並依序接入out3,out4到兩個output port。
5. 我們的設計將符合以下真值表(*表3-1*)
   以下將control簡寫為ctrl

ctrl[4]	ctrl[3]	ctrl[2]	ctrl[1]	ctrl[0]	out1	out2	out3	out4
0	0	0	0	0	in1	in2	in3	in4
0	0	0	0	1	in2	in1	in3	in4
0	0	0	1	0	in2	in1	in3	in4
0	0	0	1	1	in1	in2	in3	in4
0	0	1	0	0	in1	in3	in2	in4
0	0	1	0	1	in2	in3	in1	in4
0	0	1	1	0	in3	in1	in2	in4
0	0	1	1	1	in3	in2	in1	in4
0	1	0	0	0	in1	in2	in4	in3
0	1	0	0	1	in2	in1	in4	in3
0	1	0	1	0	in2	in1	in4	in3
0	1	0	1	1	in1	in2	in4	in3
0	1	1	0	0	in1	in4	in2	in3
0	1	1	0	1	in2	in4	in1	in3
0	1	1	1	0	in4	in1	in2	in3
0	1	1	1	1	in4	in2	in1	in3
1	0	0	0	0	in1	in2	in4	in3
1	0	0	0	1	in2	in1	in4	in3
1	0	0	1	0	in2	in1	in4	in3
1	0	0	1	1	in1	in2	in4	in3
1	0	1	0	0	in1	in3	in4	in2
1	0	1	0	1	in2	in3	in4	in1
1	0	1	1	0	in3	in1	in4	in2
1	0	1	1	1	in3	in2	in4	in1
1	1	0	0	0	in1	in2	in3	in4
1	1	0	0	1	in2	in1	in3	in4
1	1	0	1	0	in2	in1	in3	in4
1	1	0	1	1	in1	in2	in3	in4

ctrl[4]	ctrl[3]	ctrl[2]	ctrl[1]	ctrl[0]	out1	out2	out3	out4
1	1	1	0	0	in1	in4	in3	in2
1	1	1	0	1	in2	in4	in3	in1
1	1	1	1	0	in4	in1	in3	in2
1	1	1	1	1	in4	in2	in3	in1

Testbench Design & Result Explanation

1. 波形圖截圖

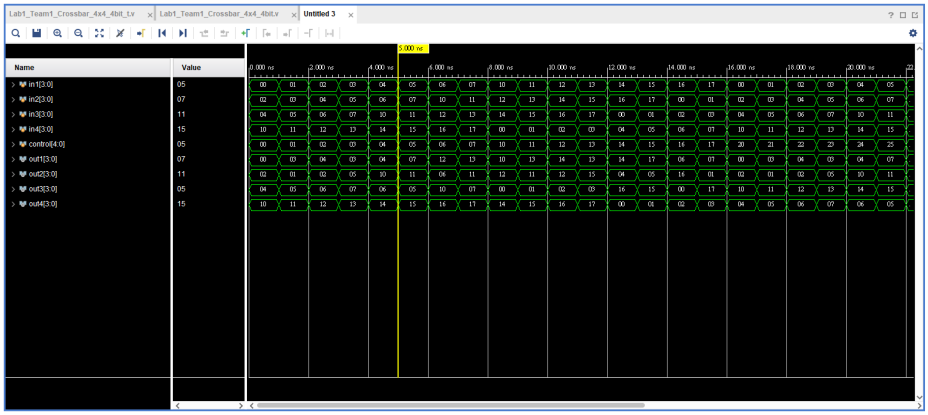


圖3-2

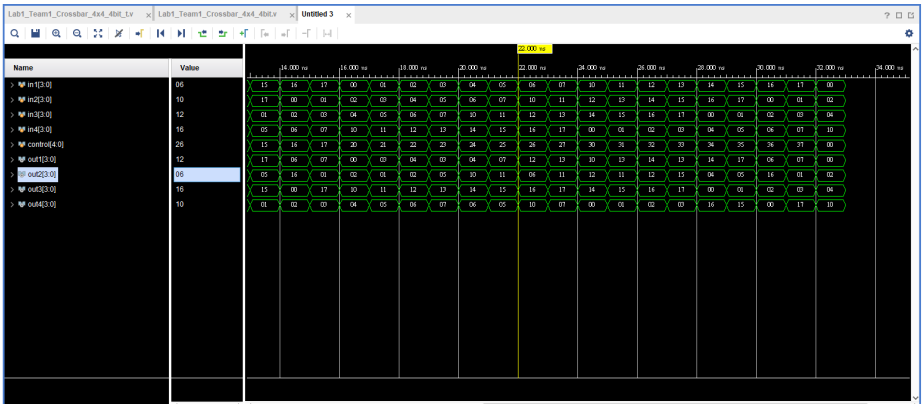


圖3-3

2. testbench設計說明:
   我們首先將in1,in2,in3,in4分別初始化為4'b0000,4'b0010,4'b0100,4'b1000，並把control初始化為5'b00000。
   接著我們使用repeat循環32次來讓control從5'b00000跑到5'b11111並檢查相對應跑出來的output是否正確。
   而為了方便判讀，在每次循環的時候，我們將in1,in2,in3,in4的值也加1，否則波形圖將會如下圖一般，造成判讀上的不便。

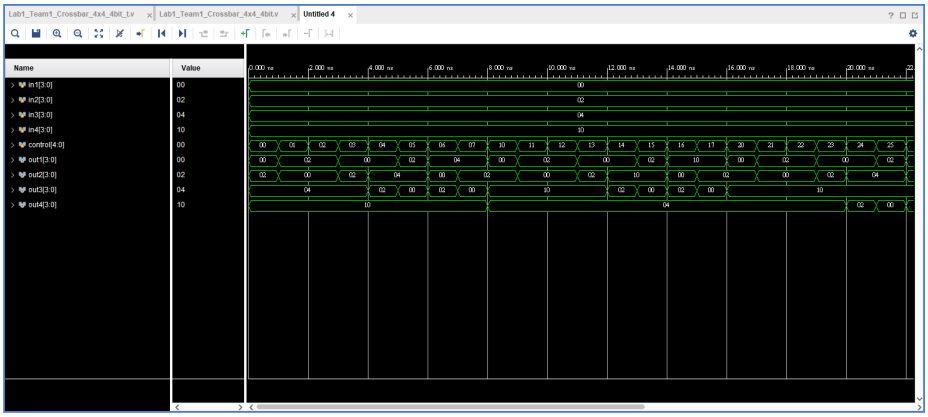


圖3-4

3. 波形圖結果說明:
   我們將波形圖(*圖3-2,3-3*)以及真值表（*表3-1*）來進行對照，可以發現結果與我們所預期的相吻合。

Combinations of input and output pair cannot be achieved by such a crossbar

透過真值表(*表3-1*)以及波形圖(*圖3-2,3-3*)所對應出來的output，我們可以發現有四個組合是無法被此題所實作出來的crossbar達成的。
而此四個組合如同下表(*表3-2*)所示。

out1	out2	out3	out4
in3	in4	in1	in2
in3	in4	in2	in1
in4	in3	in1	in2
in4	in3	in2	in1

Advanced Question4

Drawing of the design of Verilog Advanced Question 4

- gate-level circuit of tff

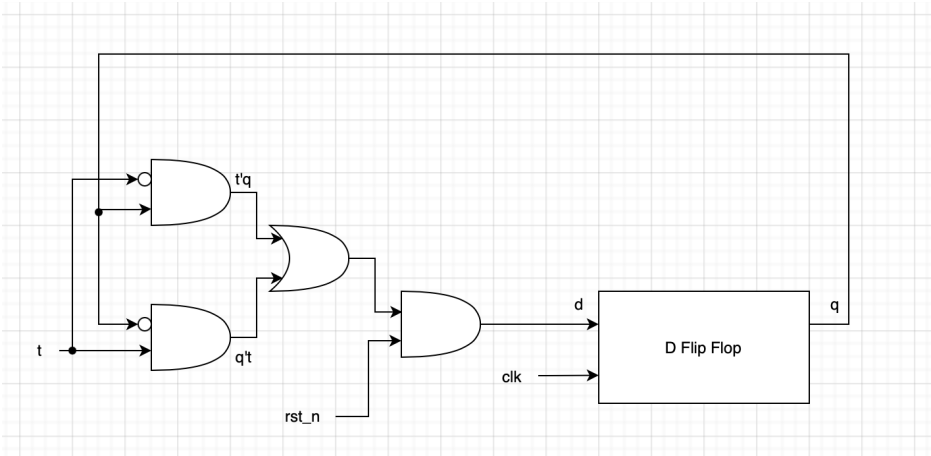


圖4-1

- gate-level circuit of ***dff*** used by tff

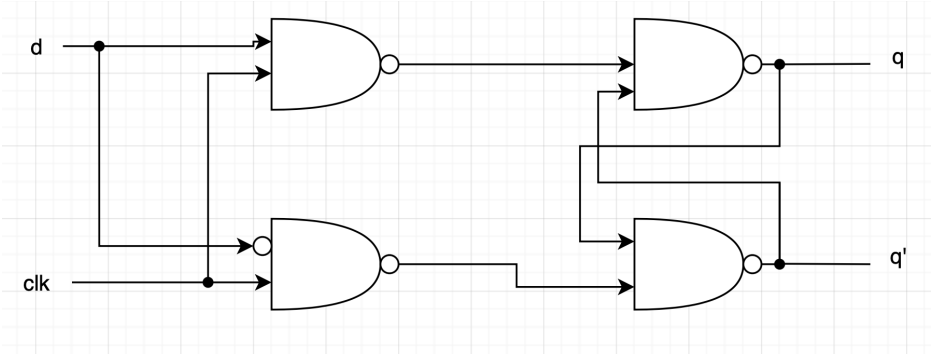


圖4-2

Requirements

1. 以gate-level的形式實做一個tff
2. 重複利用先前在basic question2所使用的dff module來實作
3. 當n_reset為1(reset = 0)時才改變值

Design Explanation

1. 依照投影片上面給出的圖來進行實作，由於禁止直接使用xor，所以我們把xor的部分拆解成tq'+t'q來符合題目要求
2. 將(t xor q)與rst_n做and的運算並將其輸入dff中d的port
3. 將clk輸入dff中clk的port
4. 將dff的output(q)接回來作為下一個state的input
5. 我們的設計將符合以下的真值表(*表4-1*)

rst_n	t	q	q+1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Testbench Design & Result Explanation

1. 波形圖截圖

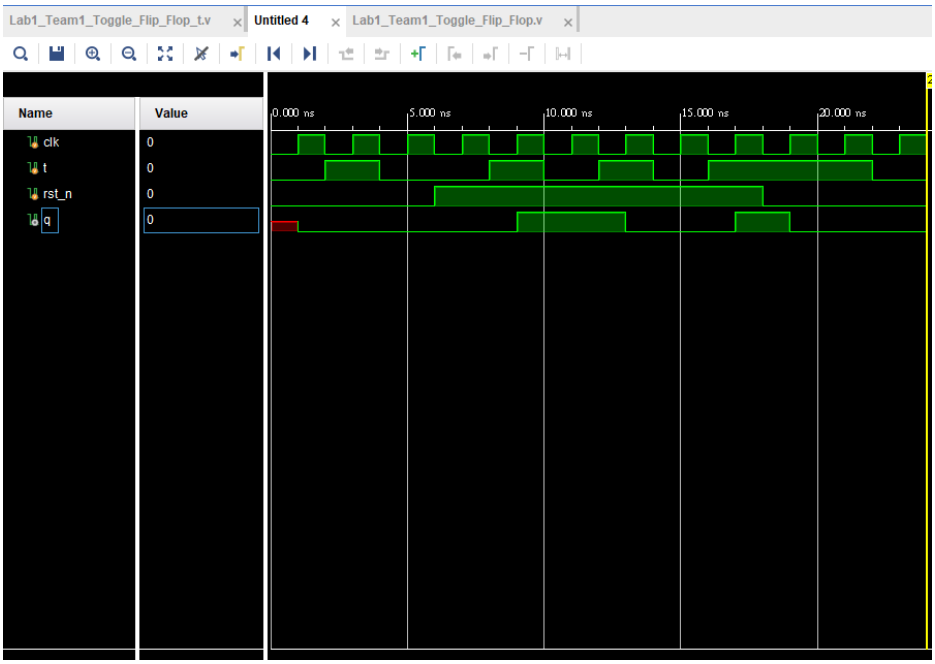


圖4-3

2. testbench設計說明:
   我們首先將clk,t,rst_n都初始化為1'b0，接下來我們讓clk每經過1ns就做反向來產生一個500MHz的clock，然後再改變t的值來確定在rst_n = 0的狀況之下，q一定為0。接著把rst_n升起並測試當t=0時，q是否有維持在前一個cycle的狀態;t=1時，q是否有改變狀態(q+1 = ~q)。最後再將rst_n降下，並再次確認在rst_n = 0時，q一定為0。
3. 波形圖結果說明:
   我們將波形圖以及真值表對照，確定在rst_n = 0的時候，q必為0，而當rst_n = 1時，在t=0的狀態下，q有維持前一個cycle的狀態，而在t=1的狀態下，q也有將前一個cycle的值做反向，而這與我們所預期的結果相吻合。

FPGA Demonstration 1

Drawing of the design of FPGA Demonstration 1

- gate-level circuit of 2x2 crossbar used in fpga demonstration 1

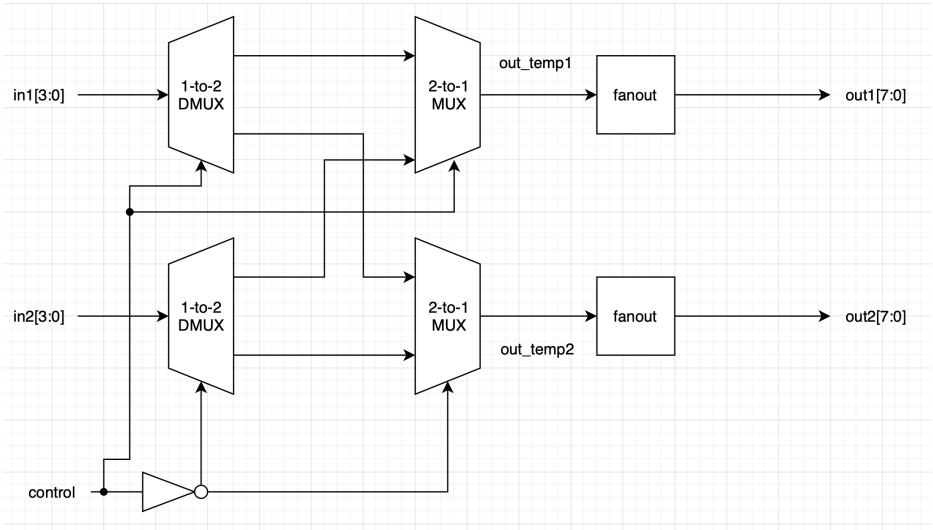


圖5-1

- gate-level circuit of fanout module used in the above design

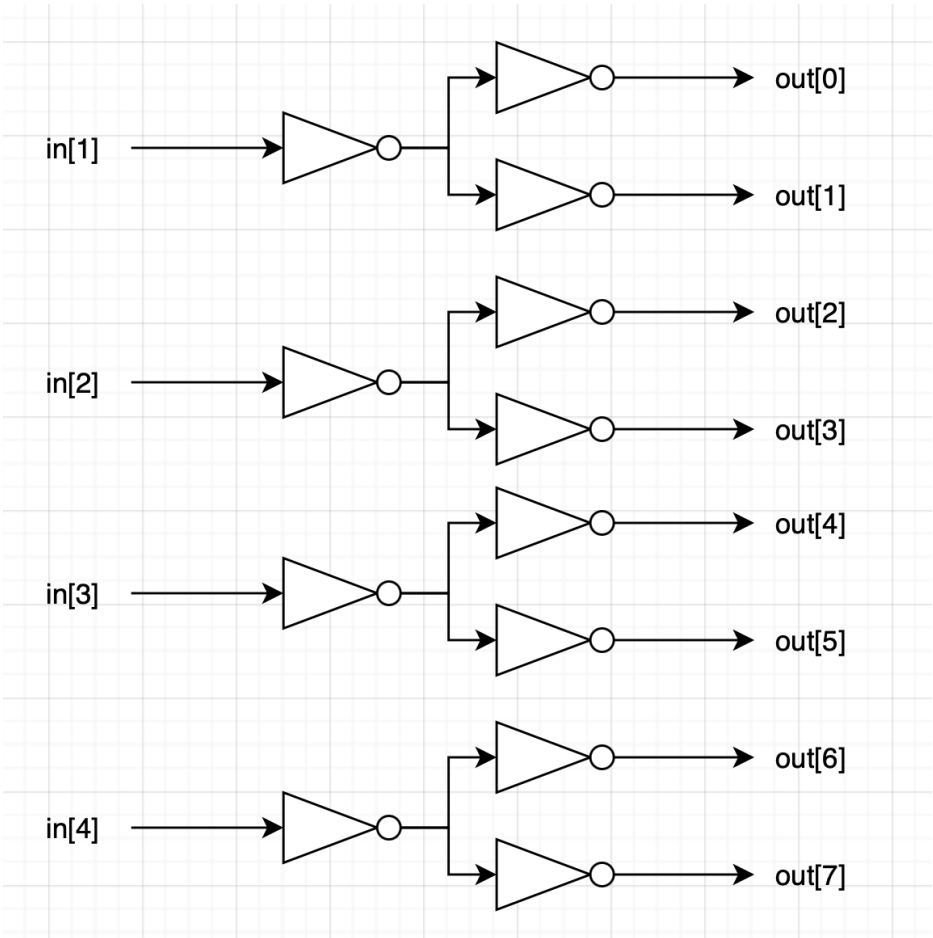


圖5-2

Design Explanation

在這一題的設計上面，我們重複使用了在advanced question2所使用到的2x2 crossbar，並把out1以及out2改成8bit，並將原本的4bit output經由fanout之後接上新的output。而我們將原本的兩個4 bit output port接上命名為out_temp1以及out_temp2的4bit wire，再將這兩條wire分別

接上fanout module，使得out_temp1[0]的訊號得以傳送到out1[0]以及out1[1]，out_temp1[1]則傳到out1[2],out1[3]，以此類推。
如此一來，我們就能夠達成題目的要求，將每個output都對應上兩個LED。

I/O pin assignment

Input	package pin
control	v17
in1[0]	v16
in1[1]	w16
in1[2]	w17
in1[3]	w15
in2[0]	v15
in2[1]	w14
in2[2]	w13
in2[3]	v2

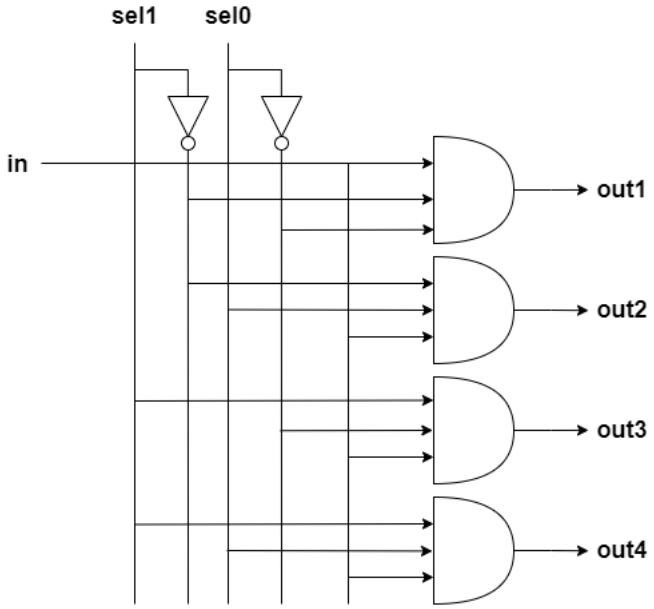
output	package pin
out1[0]	u16
out1[1]	e19
out1[2]	u19
out1[3]	v19
out1[4]	w18
out1[5]	u15
out1[6]	u14
out1[7]	v14
out2[0]	v13
out2[1]	v3
out2[2]	w3
out2[3]	u3
out2[4]	p3
out2[5]	n3
out2[6]	p1
out2[7]	l1

Contribution

施泳瑜：advanced question 1 & 2
朱季葳：advanced question 3 & 4 & FPGA Demonstration

What we have learned from Lab1

1. 沒認真看好規範就開始寫題，第一題也用純gate-level做了一次，更加警惕自己之後要好好看題。



2. 在這次的作業中發生了一些小插曲
   原本在vivado上面測試第三題的verilog design以及testbench的時候突然在message看到2個error，但是log卻沒有任何錯誤訊息，把檔案傳上工作站測試也沒有顯示任何error，於是我就把錯誤訊息丟到google上面尋找，並嘗試在
   https://support.xilinx.com/s/question/0D52E00006hpS1ZSAU/error-usfxsim62-elaborate-step-failed-with-errors-please-check-the-tcl-console-output-or-cproyectosxilinxdds27del7dds27del7dds27del7simsim1behavxsimelaboratelog-file-for-mor?language=en_US (https://support.xilinx.com/s/question/0D52E00006hpS1ZSAU/error-usfxsim62-elaborate-step-failed-with-errors-please-check-the-tcl-console-output-or-cproyectosxilinxdds27del7dds27del7dds27del7simsim1behavxsimelaboratelog-file-for-mor?language=en_US)
   網頁中所提供的做法，而我在關閉所有檔案重開之後，錯誤訊息就神奇地消失了。

3. 在這次作業中，我們體認到了重複利用模組的重要性，以及該如何用gate-level的形式來設計出mux,dmux,2x2 crossbar,4x4crossbar以及tff，最後還有如何將寫好的code燒到fpga版上。