105061254 林士平 邏輯設計實驗報告 pre_Lab3

1. pre_lab3_1 小題有兩個方法：

方法一

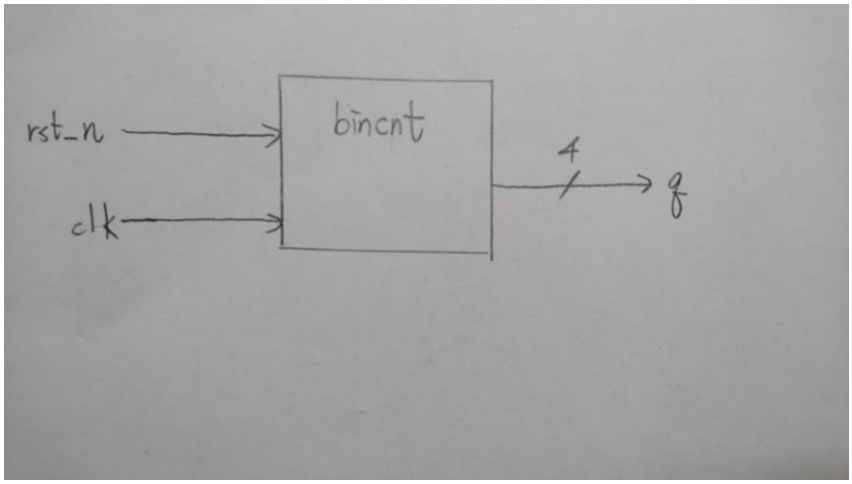
(1) Design specification :

A. Inputs and outputs(表一)：

Inputs	clk, rst_n
Outputs	q[3:0]

↑表一：Inputs and outputs of 1.(方法一)

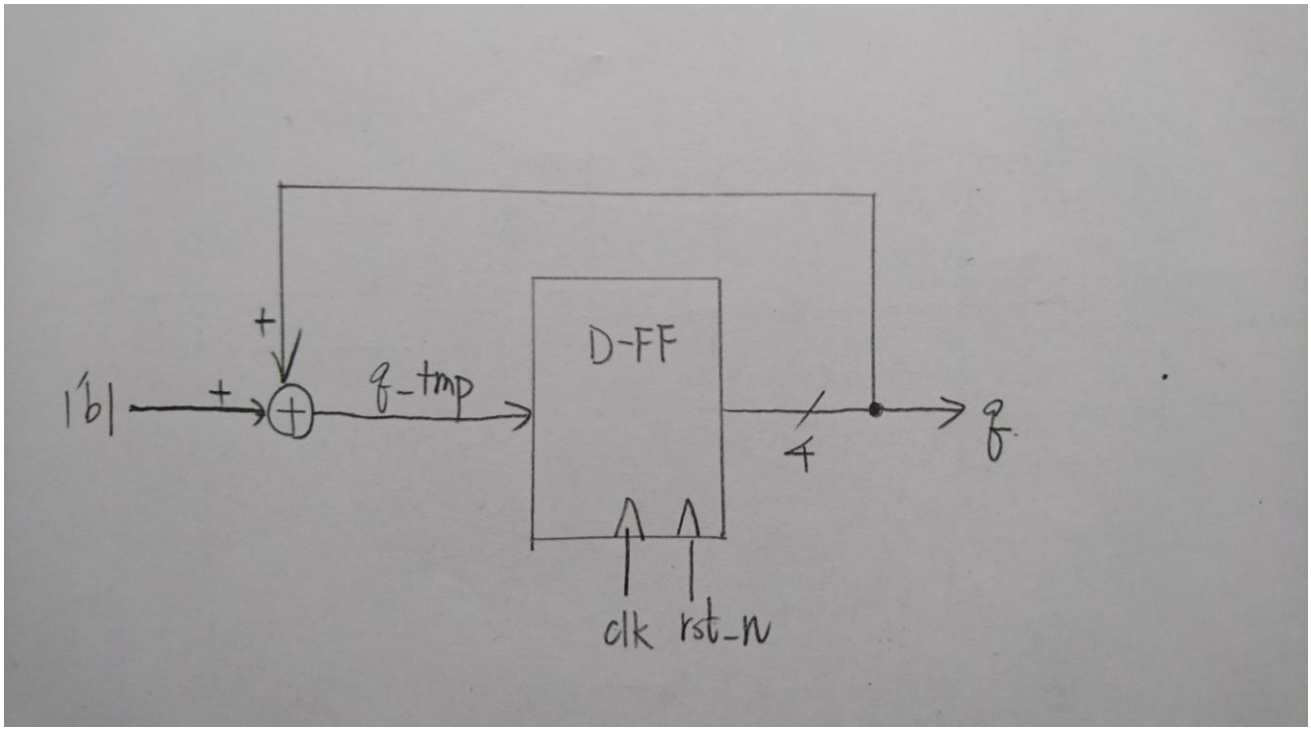
B. Block diagram(圖一)：



↑圖一：The block diagram of 1.(方法一)

(2) Design implementation :

A. Logic diagram(圖二)：



↑圖二：logic diagram of 1.(方法一)

方法二

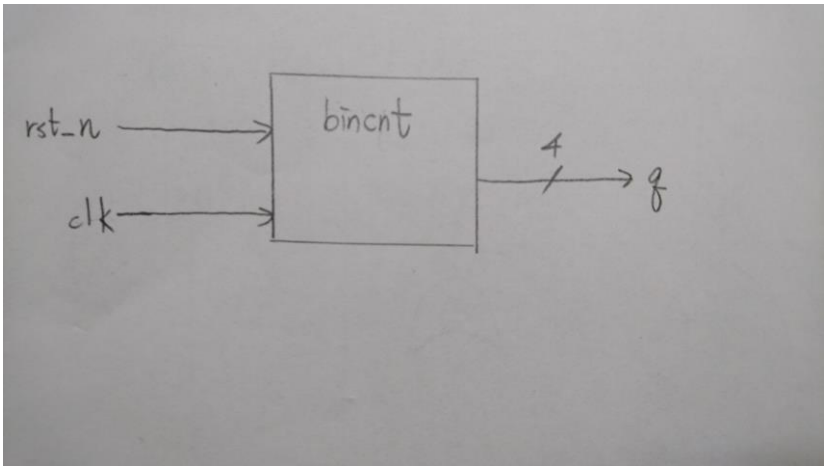
(1) Design specification :

A. Inputs and outputs(表二)：

Inputs	clk, rst_n
Outputs	q[3:0]

↑表二：Inputs and outputs of 1.(方法二)

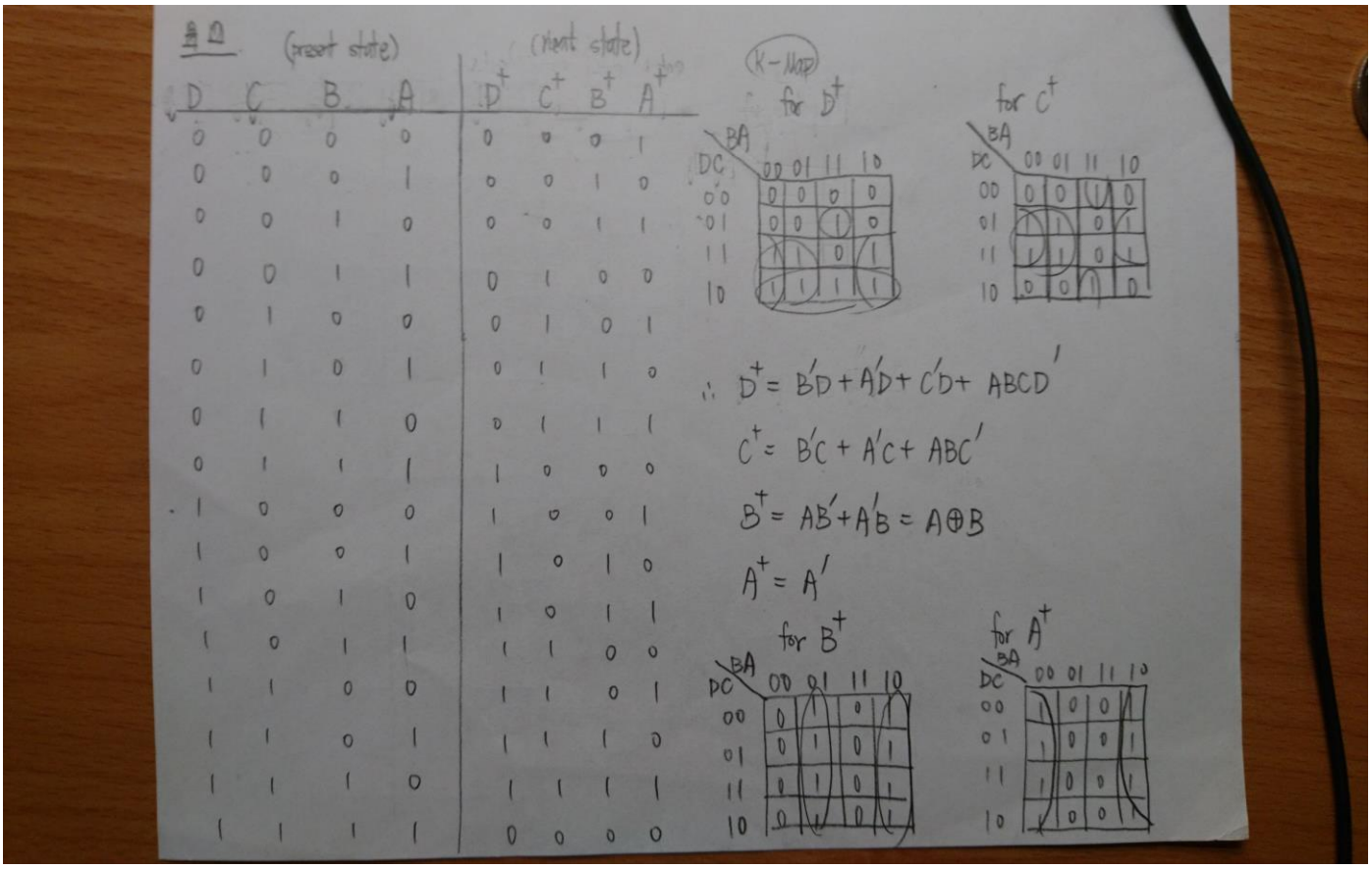
B. Block diagram(圖三)：



↑圖三：The block diagram of 1.(方法二)

(2) Design implementation :

A. Logic function(圖四)：



↑圖四：logic function of 1.計算過程(方法二)

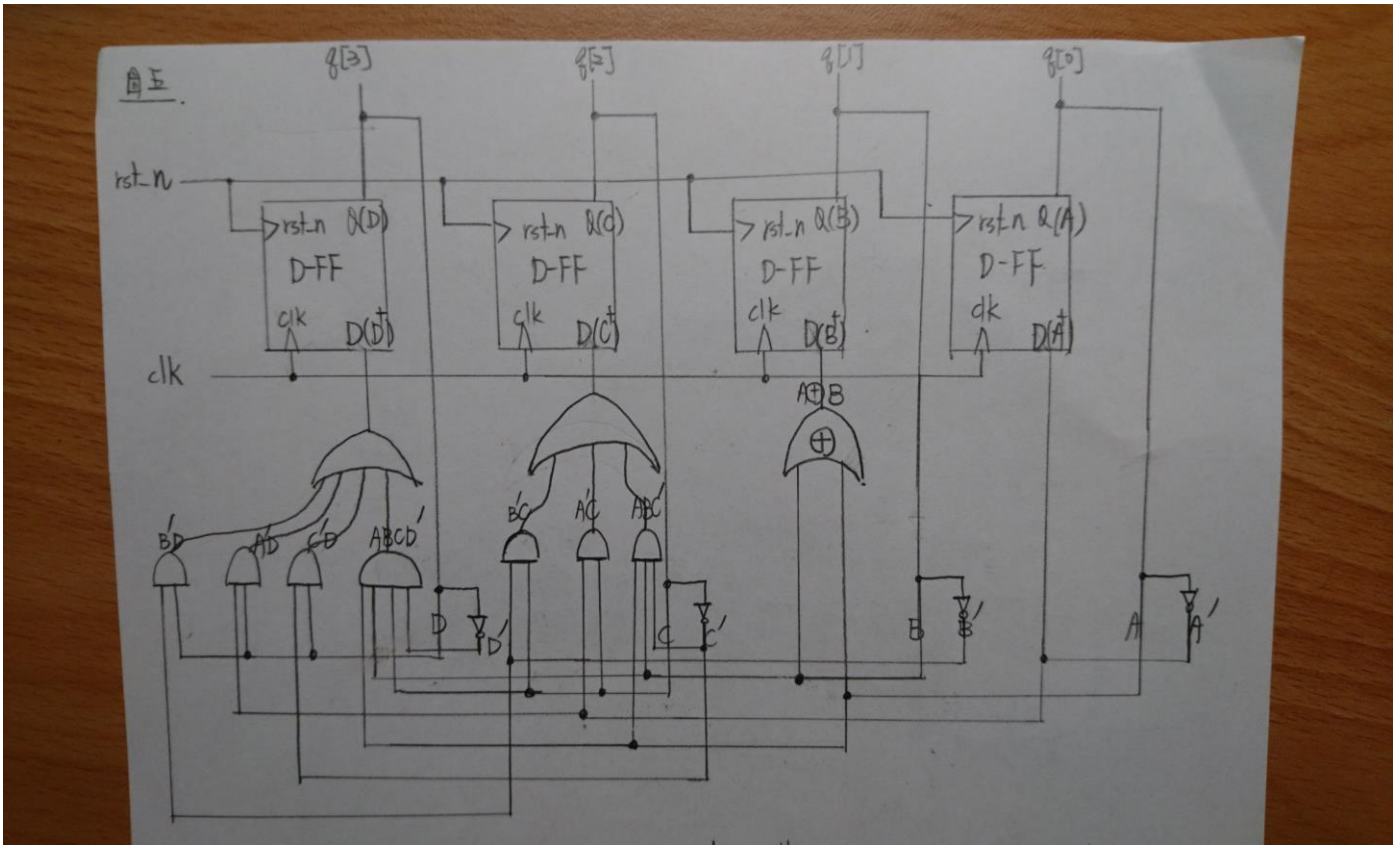
由(圖四)可知 logic function 為：

$$D^+ = B'D + A'D + C'D + ABCD'$$
$$C^+ = B'C + A'C + ABC'$$
$$B^+ = A'B + A'B = A \oplus B$$
$$A^+ = A'$$

其中 q[3:0] = {D,C,B,A}；q_tmp[3:0] = {D^+, C^+, B^+, A^+}，並將上面的四個 logic function 寫成 verilog 的形式如下：

$$q\_tmp[3] =$$
$$((\sim q[1])\ \&\ q[3])\ |\ ((\sim q[0])\ \&\ q[3])\ |\ ((\sim q[2])\ \&\ q[3])\ |\ (q[0]\ \&\ q[1]\ \&\ q[2]\ \&\ (\sim q[3]))$$
$$q\_tmp[2] = ((\sim q[1])\ \&\ q[2])\ |\ ((\sim q[0])\ \&\ q[2])\ |\ (q[0]\ \&\ q[1]\ \&\ (\sim q[2]))$$
$$q\_tmp[1] = q[0]\ \hat{}\ q[1]$$
$$q\_tmp[0] = (\sim q[0])$$

B. Logic diagram(圖五)：



↑圖五：logic diagram of 1.(方法二)

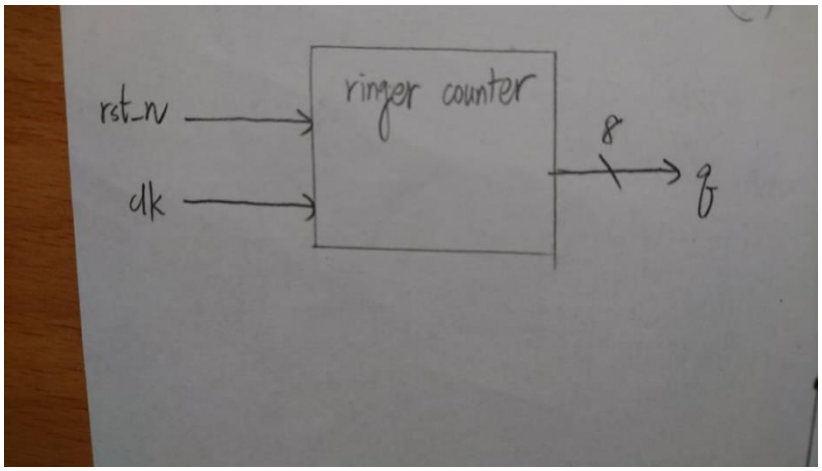
2.

(1) Design specification :

A. Inputs and outputs(表三)：

Inputs	clk, rst_n
Outputs	q[7:0]
↑表三：Inputs and outputs of 2.	

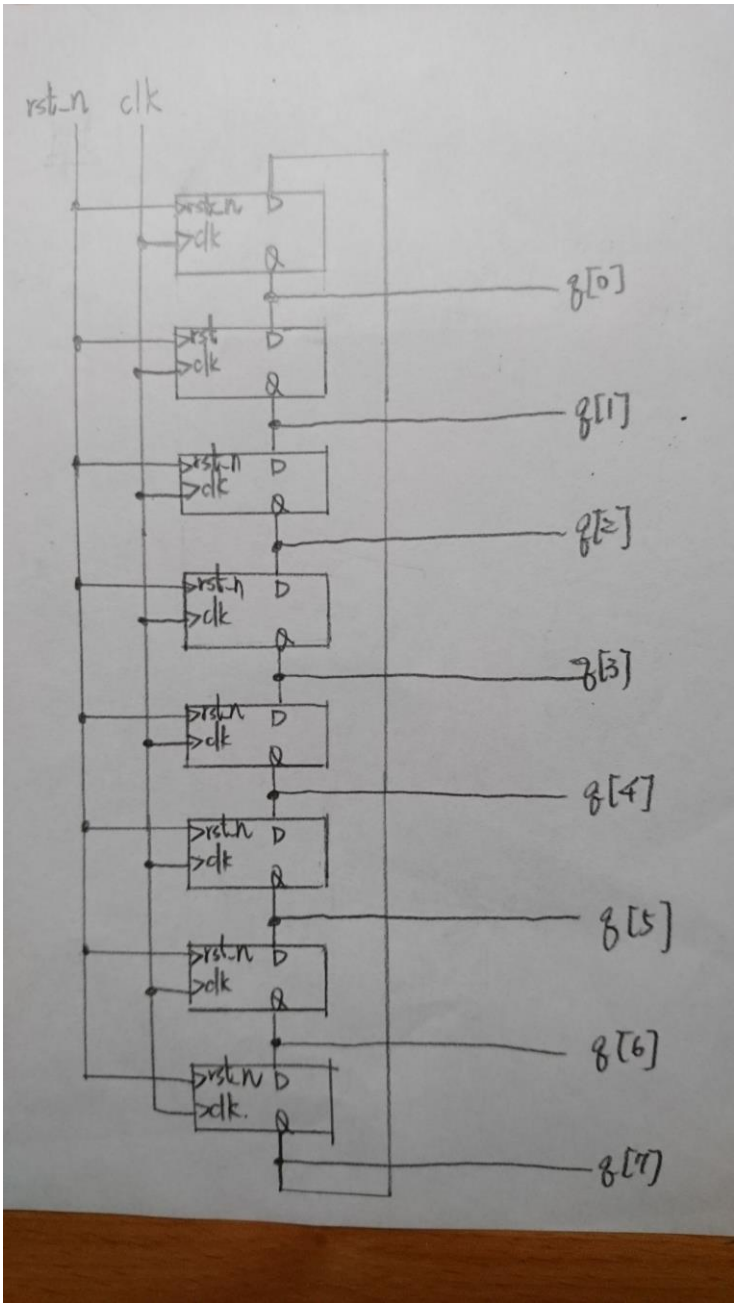
B. Block diagram(圖六)：



↑圖六：The block diagram of 2.

(2) Design implementation :

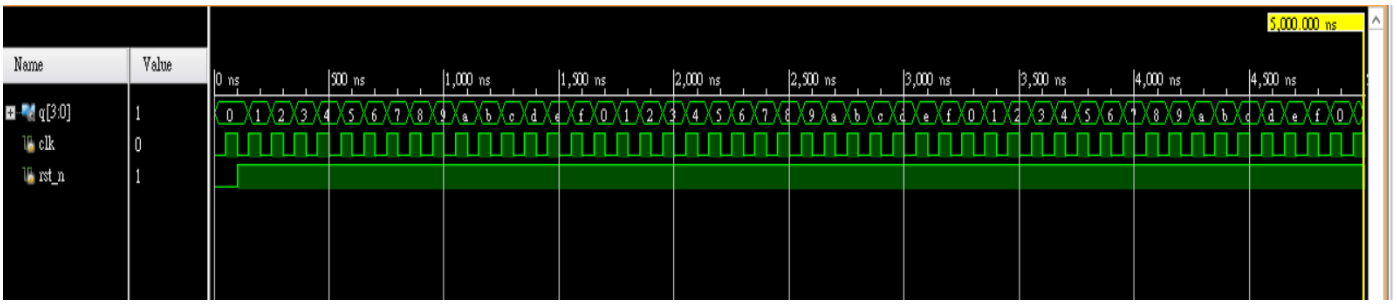
A. Logic diagram(圖七)：



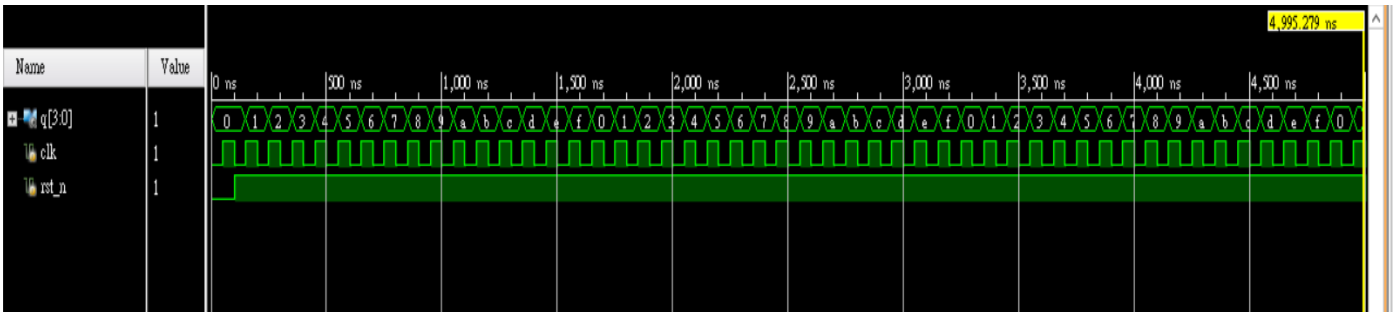
←圖七：logic diagram of 2.
(8 個 D-Flip-Flop)

3. Discussion

第一題是設計一個 4-bit synchronous binary up counter。我使用兩種方法來實現：方法一是直接使用 verilog 提供的 operator(+)，讓 next state = present state + 1，再將 next state 利用 D-FF 輸給下個時刻的 present state，得到的波型圖如圖八所示。方法二是將 4-bit synchronous binary up counter 的 truth table 畫出來，利用 K-map 化簡，找出 next state 和 present state 之間的 logic funtion，便可以利用 present state 以及 and, or, nor 邏輯閘來得到 next state，再將 next state 利用 D-FF 輸入給下個時刻的 present state，得到的波型圖如圖九所示。



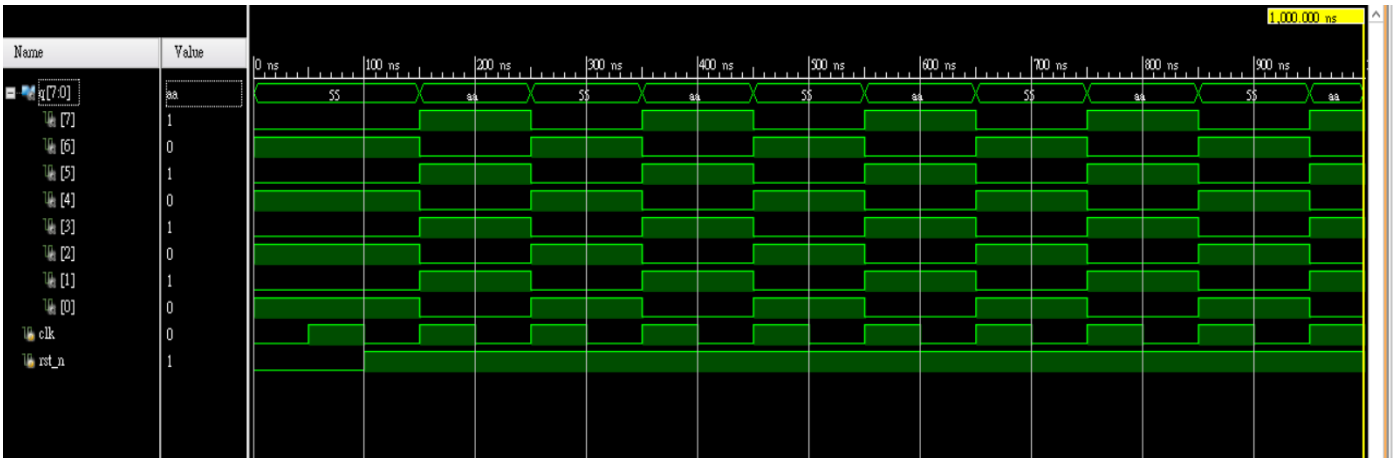
↑圖八：1.(方法一)模擬結果



↑圖九：1.(方法二)的模擬結果

由圖八和圖九可發現，用兩種方法所得到的模擬結果相同。不過方法一的 code 比較精簡，而且可以推廣到 n-bit synchronous binary up counter，實用性和一般性較高。

第二題是要設計一個 shift register 讓輸出可以在 01010101 和 10101010 之間跳動(initial output of DFF output after reset is 01010101)。我的設計方式是將 8 個 DFF 串聯在一起，8-bit output 即為八個 DFF 的 output，最後要將最後一個 DFF 的 output 和第一個 DFF 的 input 連接在一起，形成”ringer” counter。圖十為本題的波型圖。



↑圖十：2.的模擬結果

pre_Lab 的這兩題我遇到最大的問題是不會寫 testbench，不過在參考老師 verilogHDL(2)講義 P.24，並且花了一些時間研究之後，終於想通如何在 testbench 中製造一個 clk，還有了解 initial 和 always 的不同。

4. Conclusion

我覺得這次的 pre_lab 讓我學會更多 testbench 的寫法，並且將邏輯設計課的理論實際應用在 verilog 上(例如：pre_lab3_1 的方法二)，讓我知道寫 truth table、畫 K-map 有實際的功用，而且 pre_lab 的這兩題對於我做 lab3 的題目有非常大的幫助。