105061254 林士平 邏輯設計實驗報告 Lab3

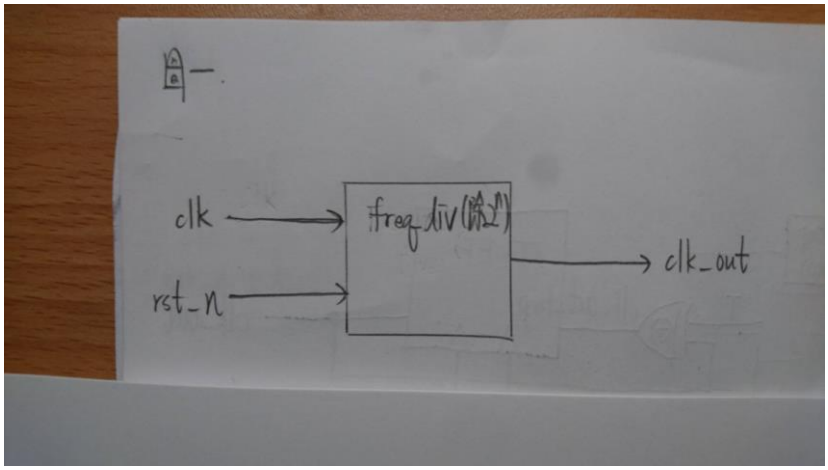
1.

(1) Design specification :

A. Inputs and outputs(表一)：

Inputs	clk, rst_n
Outputs	clk_out
↑表一：Inputs and outputs of 1.	

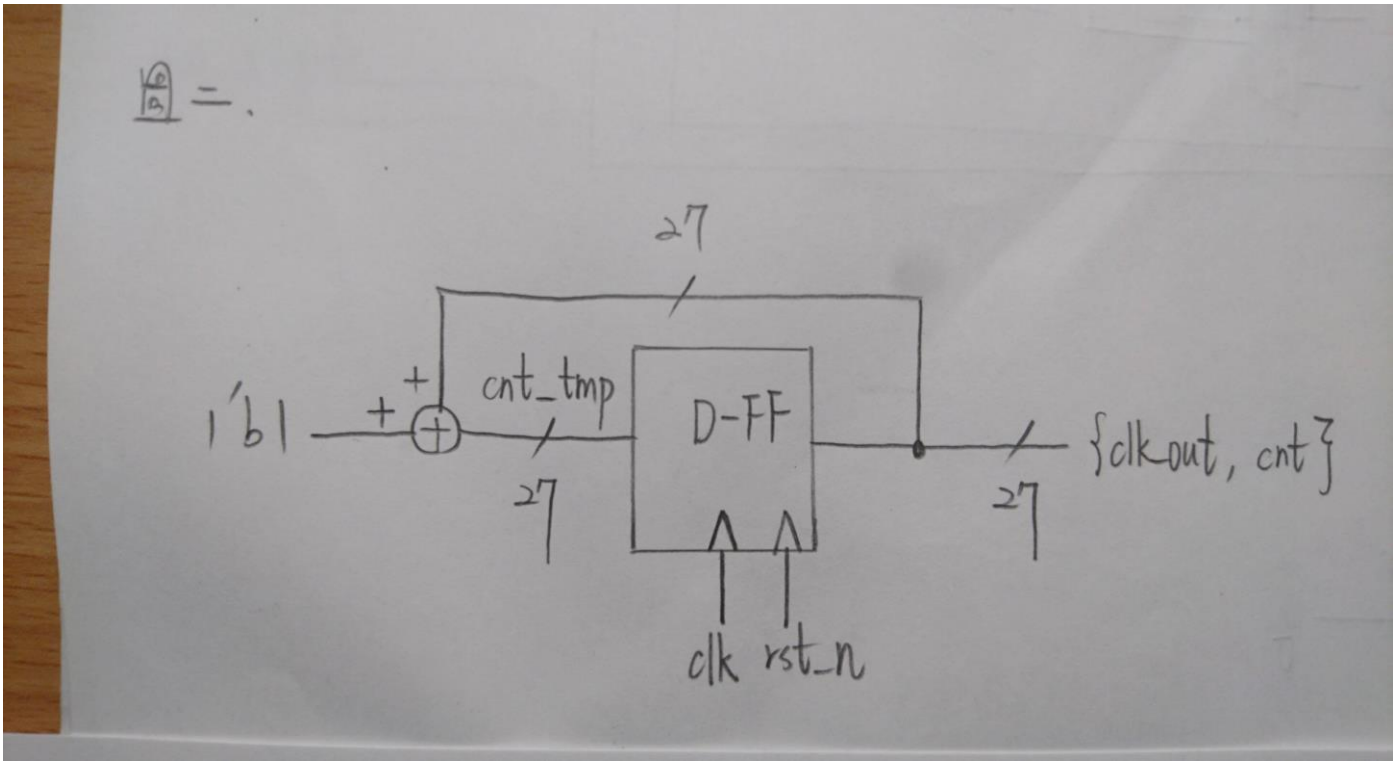
B. Block diagram(圖一)：



↑圖一：The block diagram of 1.

(2) Design implementation :

A. Logic diagram(圖二)：



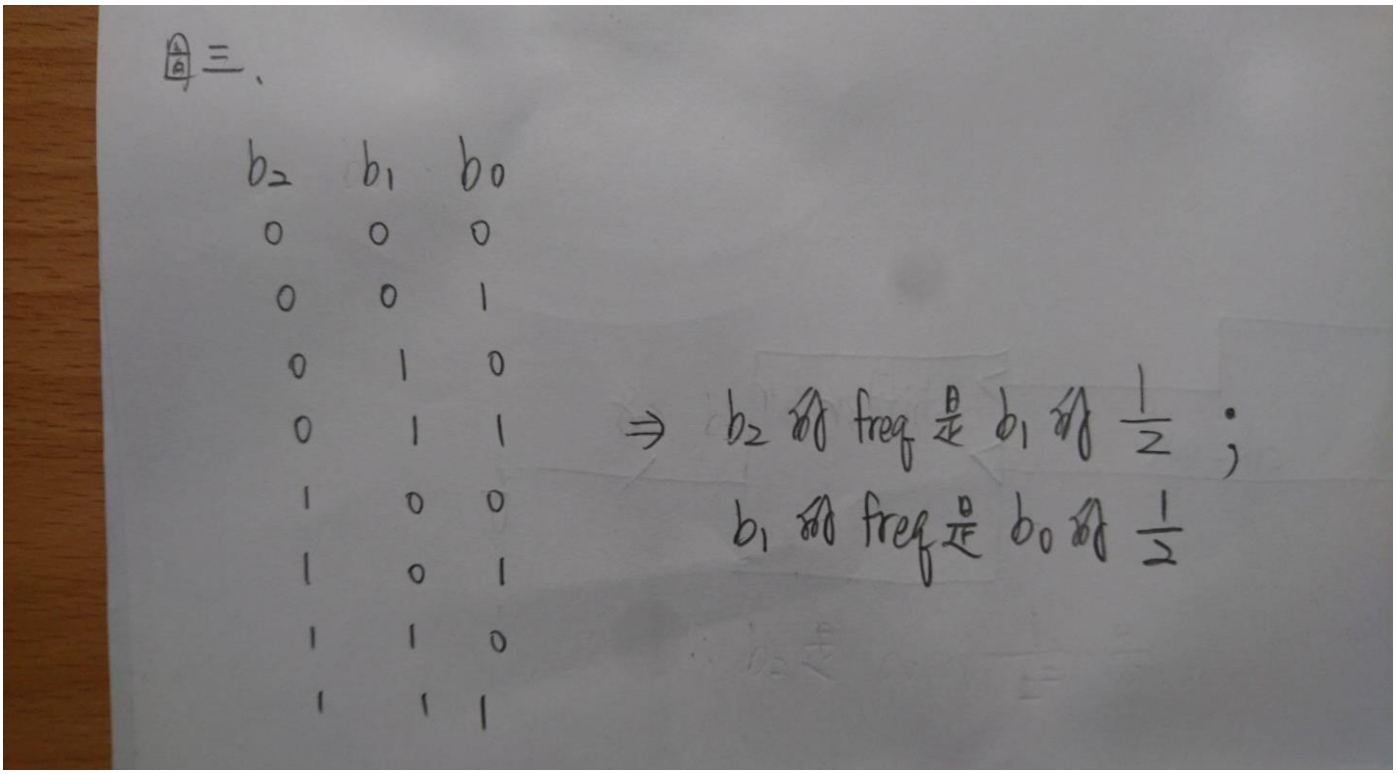
↑圖二：logic diagram of 1.

B. I/O pin assignment(表二)：

I/O	clk	rst_n	clk_out
LOC	W5	V17	U16
↑表二：I/O pin assignment of 1.			

C.功能與做法說明：

本題為一個能將 clk 除以 2^n 的 frequency divider，其中 input 為 clk(FPGA 板上的原始 clock)和 rst_n(negative edge trigger 的 reset，可以將 D-FF 的輸出{clk_out, cnt}歸零)，output 為除頻後的 clock(clk_out)。除以 2^n 其實就是做一個 n 個 bit 的 counter，並取它最高位元的值，原理說明如下圖（圖三）。所以 combination circuit 的部分作加 1 的動作，將加完的數值暫存在 cnt_tmp，作為 D-FF 的輸出{clk_out, cnt}的 next state，當 clock 為 positive edge 時，將 cnt_tmp 傳給{clk_out, cnt}，如此一直重複（除非按下 rst_n 開關）。



↑圖三：除 2^n 器原理說明

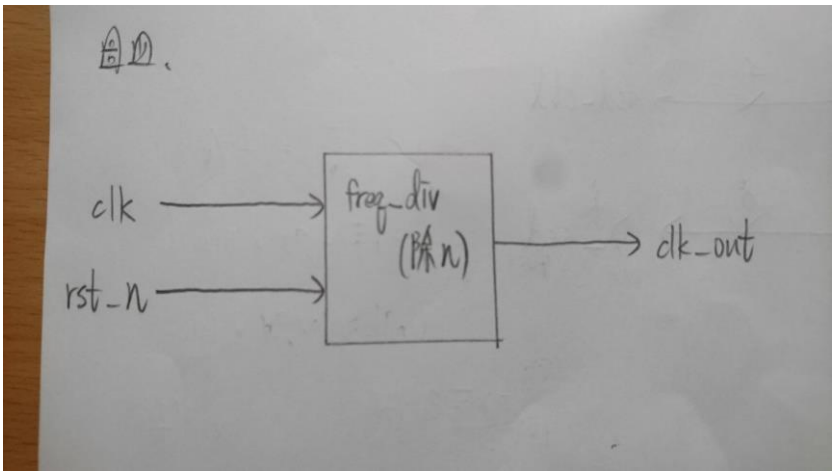
2.

(1) Design specification :

A. Inputs and outputs(表三)：

Inputs	clk, rst_n
Outputs	clk_out
↑表三：Inputs and outputs of 2.	

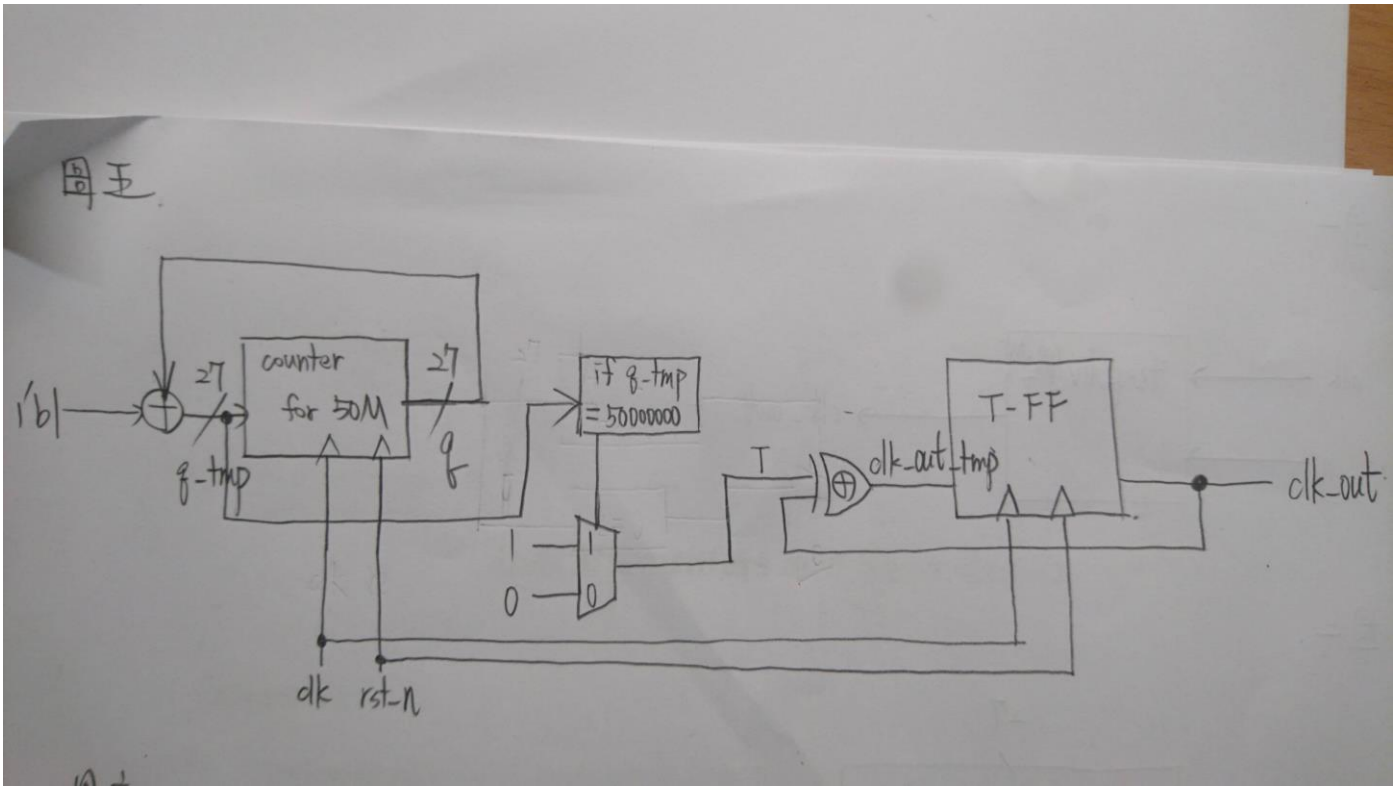
B. Block diagram(圖四)：



↑圖四：The block diagram of 2.

(2) Design implementation :

A. Logic diagram(圖五)：



↑圖五：logic diagram of 2.

B. I/O pin assignment(表四)：

I/O	clk	rst_n	clk_out
LOC	W5	V17	U16
↑表四：I/O pin assignment of 1.			

C.功能與做法說明：

本題為一個能將 clk 除以 n 的 frequency divider，其中 input 為 clk(FPGA 板上的原始 clock)和 rst_n(negative edge trigger 的 reset，可以將 FF 的輸出 clk_out 和 q 歸零)，output 為除頻後的 clock(clk_out)。首先有個 counter 可以數到至少 50M，判斷當 counter 數到 50M 時，便要讓輸出訊號(clk_out)toggle，如此能讓 clk_out 的頻率是 clk 的 1/100M 倍，也就是 1Hz。

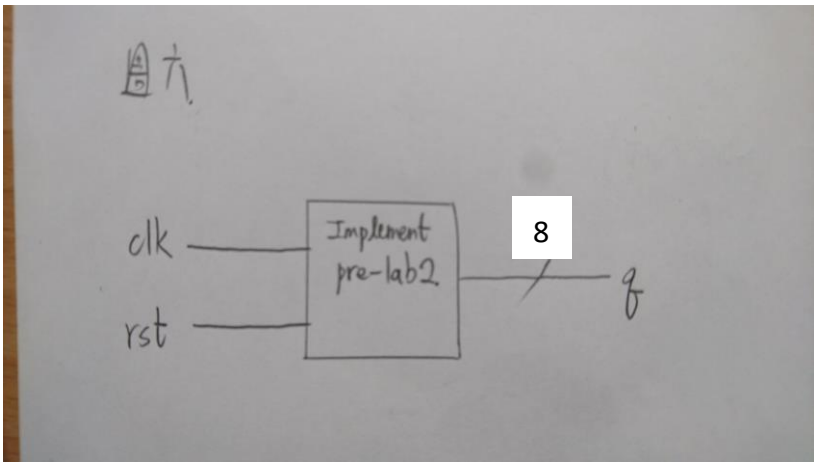
3.

(1) Design specification :

A. Inputs and outputs(表五)：

Inputs	q[7:0]
Outputs	clk, rst
↑表五：Inputs and outputs of 3.	

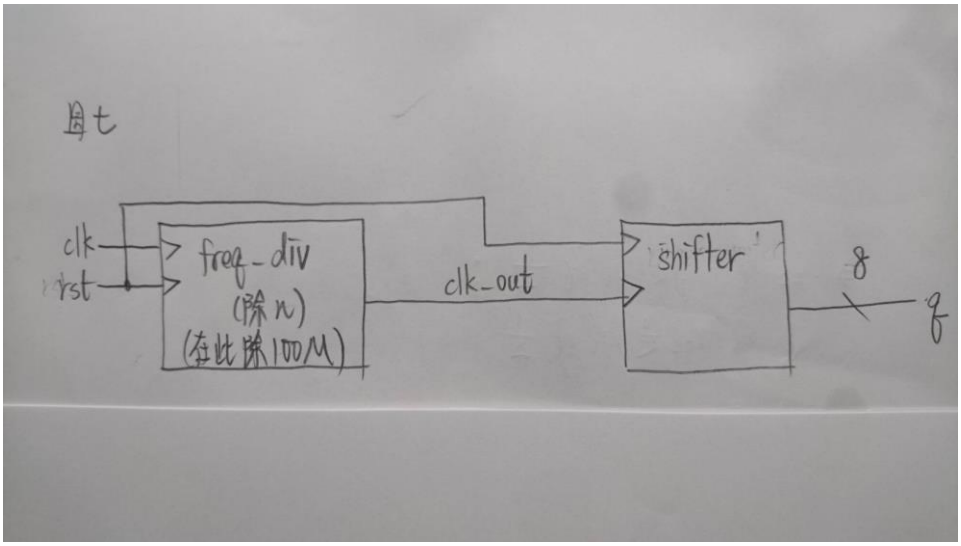
B. Block diagram(圖六)：



↑圖六：The block diagram of 3.

(2) Design implementation :

A. Logic diagram(圖七)：



↑圖七：logic diagram of 3.(其中 frequency divider 內部見圖五，shifter 內部見 pre_lab3 結報圖七)

B. I/O pin assignment(表六)：

I/O	q[7]	q[6]	q[5]	q[4]	q[3]	q[2]	q[1]
LOC	V14	U14	U15	W18	V19	U19	E19
I/O	q[0]	clk	rst				
LOC	U16	W5	V17				
↑表六：I/O pin assignment of 4.							

C.功能與做法說明：

本題要將 pre-lab3 第二題實際 implement 到 FPGA 板上：將 pre-lab3 的 8-bit 輸出 assign 給 8 個 LED 燈，讓 LED 燈可以呈現交錯閃爍的狀態。本題有三個模組，首先是 frequency divider，它將原本 FPGA 板上 clock 的頻率除以 100M，輸出 1Hz 的 clk_out，再把 clk_out 接到 shifter 作為它的 clock。再來是 shifter，讓輸出(q)可以在 01010101 和 10101010 之間跳動。最後是 top_module，將上述的兩個 module 整合起來，使整體來說，輸入為 clk(FPGA 板上的 clock)和 rst，輸出為 q。其中，rst 後 clk_out <= 0、q <= 01010101(此 q 為 shifter 的 q)。

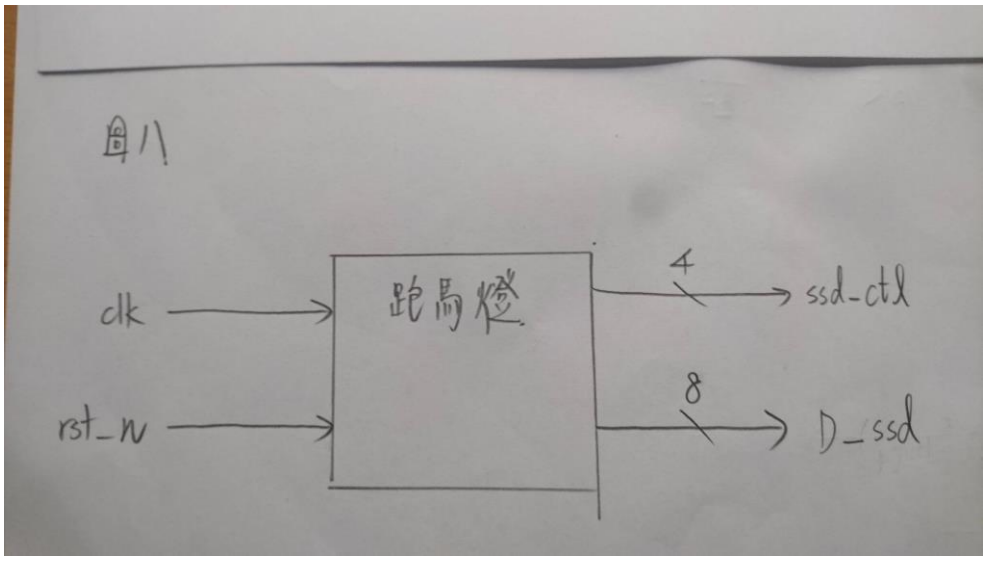
4.

(1) Design specification :

A. Inputs and outputs(表七)：

Inputs	clk, rst_n
Outputs	ssd_ctl[3:0], D_ssd[7:0]
↑表七：Inputs and outputs of 4.	

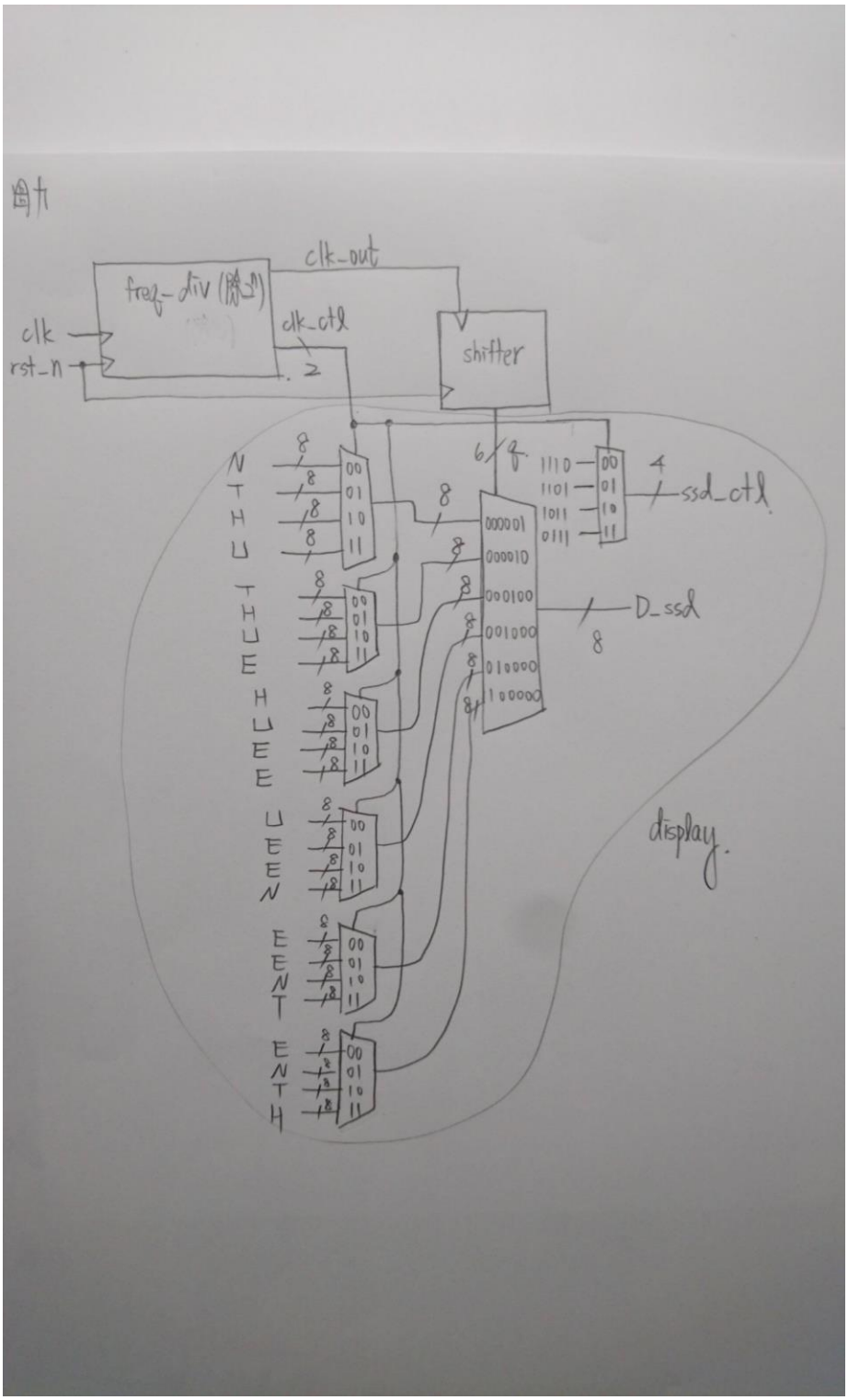
B. Block diagram(圖八)：



↑圖八：The block diagram of 4.

(2) Design implementation :

A. Logic diagram(圖九)



←圖九：logic diagram of 4.

B. I/O pin assignment(表八)：

I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]
LOC	W7	W6	U8	V8	U5	V5	U7
I/O	D_ssd[0]	clk	rst_n	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]
LOC	V7	W5	V17	U2	U4	V4	W4
↑表八：I/O pin assignment of 4.							

C.功能與做法說明：

本題為製作一個跑馬燈，讓四個七段顯示器 NTHUEE。總共有四個 module，首先是 frequency divider，將原本 FPGA 板上的 clock 轉換成頻率較低的 clk_out 和 clk_ctl，其中 clk_ctl 頻率較 clk_out 高，功能為視覺占留，讓人眼可以看到四個七段顯示器顯示不同數字；clk_out 則是輸入 shifter，作為 shifter 的 clock。再來是 shifter，用來決定四個七段顯示器要顯示哪四個不同字母，如果 q 值是 000001 則顯示 NTHU、q 值是 000010 顯示 THUE……以此類推，變換頻率為 $1/2^{27}$。然後是 display，由一堆 mux 組成，用來決定輸出 ssd_ctl(哪個七段顯示器要顯示)和 D_ssd(七段顯示器要顯示什麼字母)的值。最後是 top_module，將上述三個模組接起來，所以總體來說，輸出為：D_ssd 和 ssd_ctl，輸入為：clk(FPGA 板上的 clock)和 rst_n。

5. Discussion

第一題我覺得最困難也是最神奇的地方在設計原理，竟然用 n-bit counter 取最高位元，就能做出除以 2^n 的除頻器，對於兩年沒有接觸邏輯設計的我來說花了一些功夫理解。不過在了解原理之後，方塊圖便能很輕易地畫出來。

第二題花了我不少時間，尤其感謝老師的提示，讓我想到還有 T-FF 這個東西可以用，而且最重要的是，如果電路中有兩個 Flip Flop，最好就要用兩個 always 來分別表示兩個 Flip Flop 做的事情，如果像寫軟體一樣想要將 code 省到最省反而會找自己麻煩，寫 verilog(硬體描述語言)最重要的是要能寫得清楚，清楚表達 Logic diagram 中每個 block 做的事情。

第三題讓我學到 top_module 的寫法，包括：如何連接各個 module、wire 的使用……等，學會了之後讓我可以完成更大型的 module，實現更多功能。

第四題花了我非常多的時間。首先我考慮如何讓四個七段顯示器分別顯示不同字母，好不容易做出來之後，再想辦法讓四個字母可以移動。最一開始我想用兩個 frequency divider 分別輸出兩個 divided clock 來控制 ssd_ctl 和 D_ssd，但用 testbench 跑 simulation 發現訊號會有問題(會有半個週期沒有訊號)，所以後來改用一個 frequency divider，才解決這個問題。之後上課才知道，一個電路中最好不要有兩個 clock，因為可能會有相位不同的問題。

最後，我一直很好奇=和<=的不同，所以在上機時間請教助教，感謝助教花相當多的時間解釋，我的理解是這樣：<=會同時動作，但=會有時間先後順序之分，所以在 Flip Flop 中最好用<=，讓所有資料在 positive clock edge 時同步更新。如果我的理解錯誤還請助教指正。

6. Conclusion

這次的實驗花了我相當多時間，尤其是跑馬燈。雖然很累，不過打完真的很有成就感。從這次實驗中，我學到最重要的一點是一定要先把 logic diagram 畫出來，基本上只要畫出來，code 就一定可以打出來，可是如果沒有畫 logic diagram，就算 code 打出來也不一定可以成功產生 bitstream。

覺得打完這個 lab 之後，我的 verilog 能力又增進了一大步。