

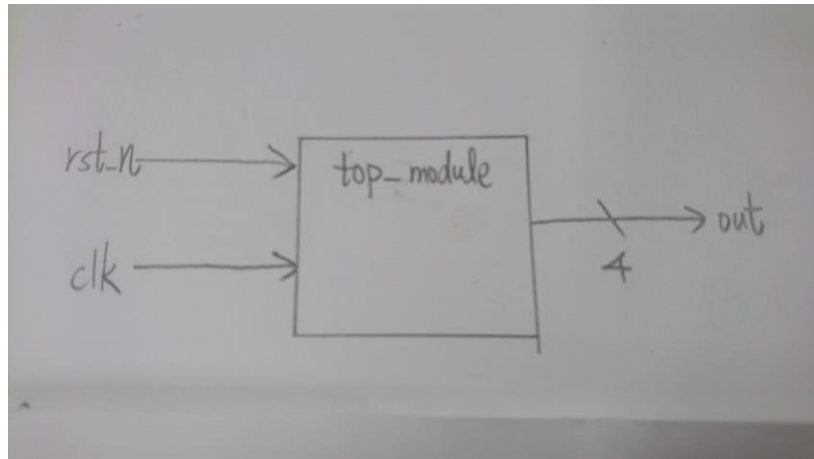
1.

(1) Design specification :

A. Inputs and outputs(表一) :

Inputs	clk, rst_n
Outputs	out[3:0]
↑ 表一 : Inputs and outputs of 1.	

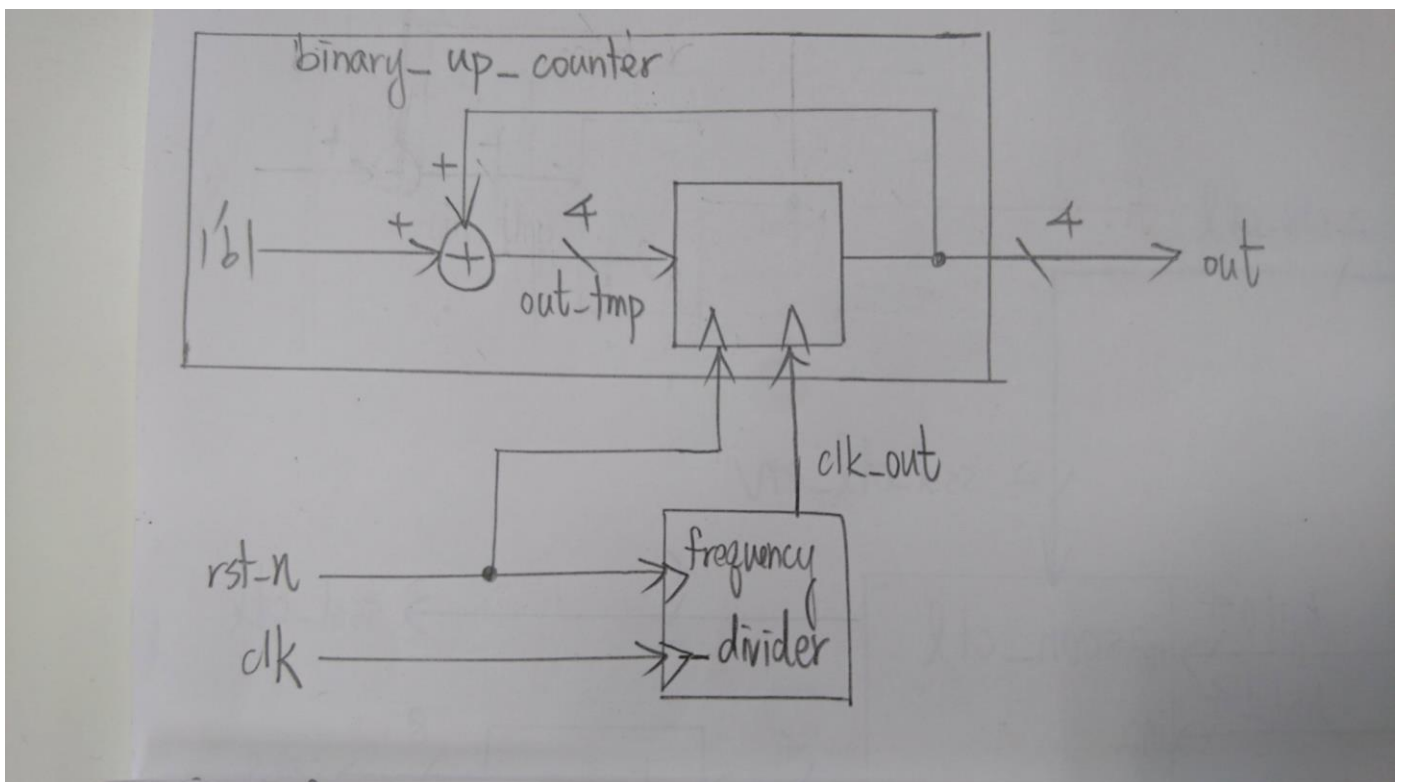
B. Block diagram(圖一) :



↑ 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	out[3]	out[2]	out[1]	out[0]	rst_n	clk	
LOC	V19	U19	E19	U16	V17	W5	

↑ 表二：I/O pin assignment of 1.

C.功能與做法說明：

本題為 4-bit synchronous binary up counter with 1-Hz clock frequency，並且用 4 個 LED 燈來顯示數出來的結果(out[3:0])。這題總共有三個 module：首先 frequency\_divider 把 FPGA 板上的 clock 做除頻，使輸出的 clk\_out 頻率為 1Hz；再來是 binary\_up\_counter 做數數的動作，並輸出數數的結果(out[3:0])，其中 binary\_up\_counter 輸入的 clk 為除頻後的 clk；最後 top\_module 用來把前述兩個 module 連接起來。

2.

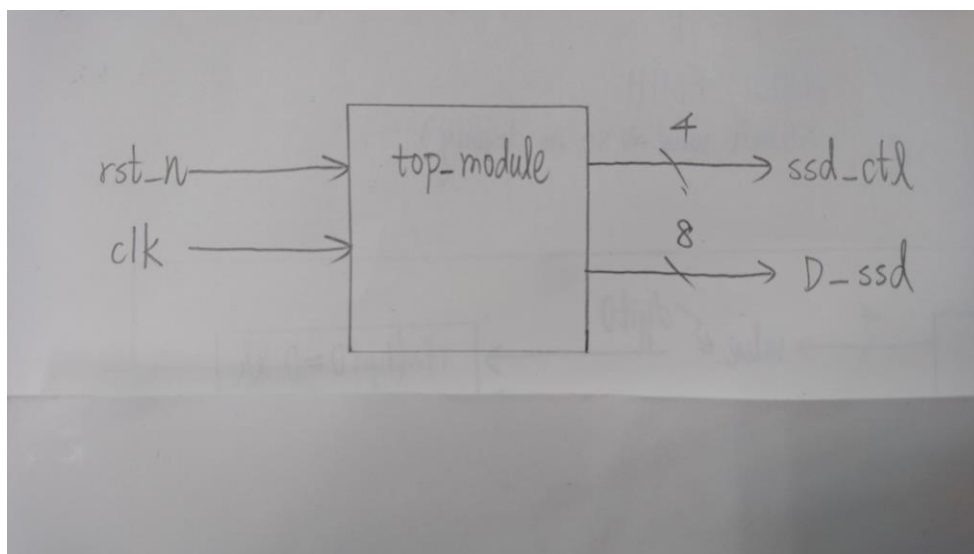
(1) Design specification：

A. Inputs and outputs(表三)：

Inputs	clk, rst_n
Outputs	ssd_ctl[3:0], D_ssd[7:0]

↑ 表三：Inputs and outputs of 2.

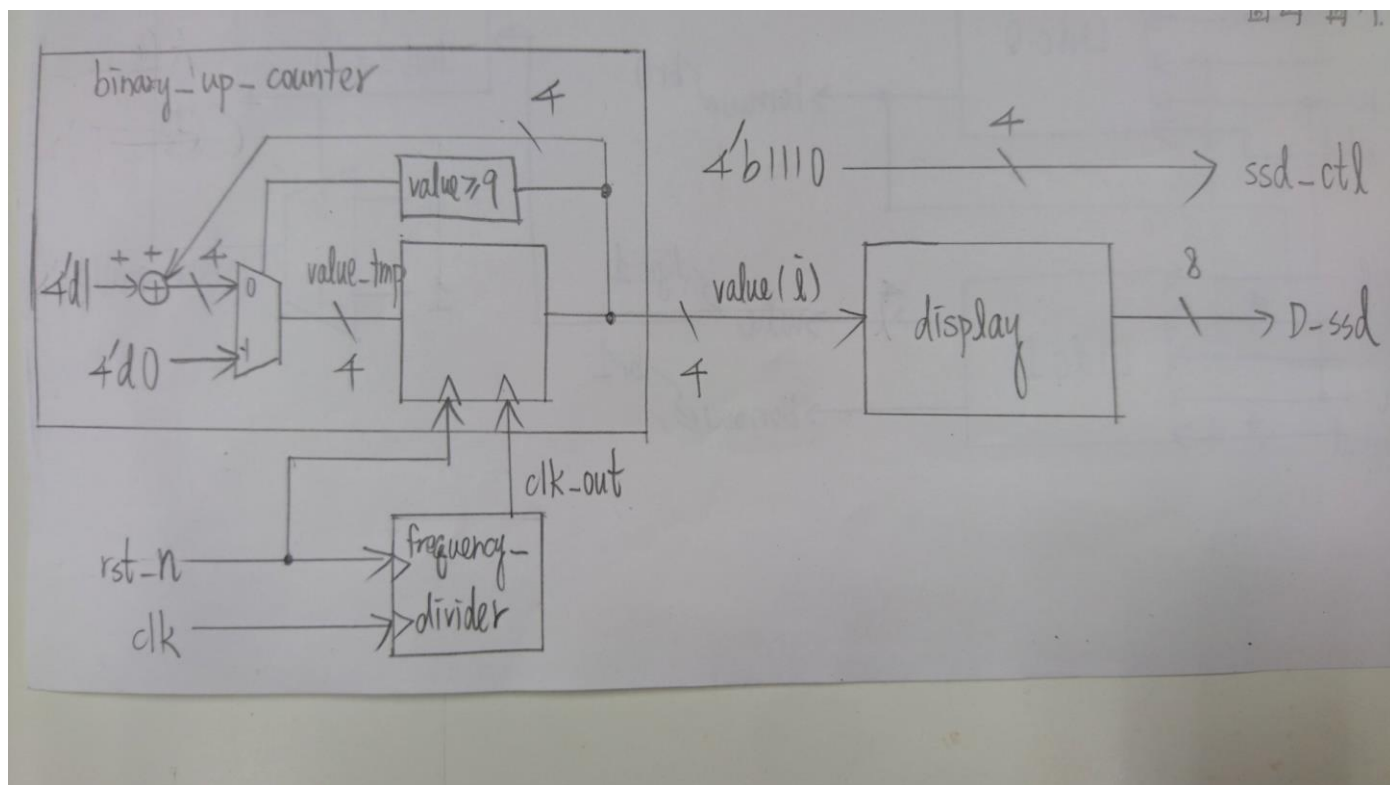
B. Block diagram(圖三)：



↑ 圖三：The block diagram of 2.

(2) Design implementation :

A. Logic diagram(圖四) :



↑ 圖四：logic diagram of 2.

B. I/O pin assignment(表四) :

I/O	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	D_ssd[7]	D_ssd[6]	D_ssd[5]
LOC	W4	V4	U4	U2	W7	W6	U8
I/O	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]	rst_n	clk
LOC	V8	U5	V5	U7	V7	W17	W5

↑ 表四：I/O pin assignment of 2.

C.功能與做法說明：

本題為一個 single digit BCD up counter with the divided clock as the clock frequency，並且將數數的結果(value[3:0])用 seven-segment display 來顯示。這題總共有四個 module：首先 frequency\_divider 把 FPGA 板上的 clock 做除頻，使輸出的 clk\_out 頻率為 1Hz；再來是 binary\_up\_counter 做數數的動作，並輸出數數的結果(value[3:0])，其中 binary\_up\_counter 輸入的 clk 為除頻後的 clk；接著 display 模組接收數數的結果(value[3:0])(i[3:0])並利用 mux 判斷來輸出七段顯示器要顯示的數字(D\_ssd[7:0])；最後 top\_module 把前述三個 module 連接起來，並且多輸出一個 ssd\_ctl[3:0](為常數 4'b1110)，使得只有最右邊的七段顯示器會顯示。

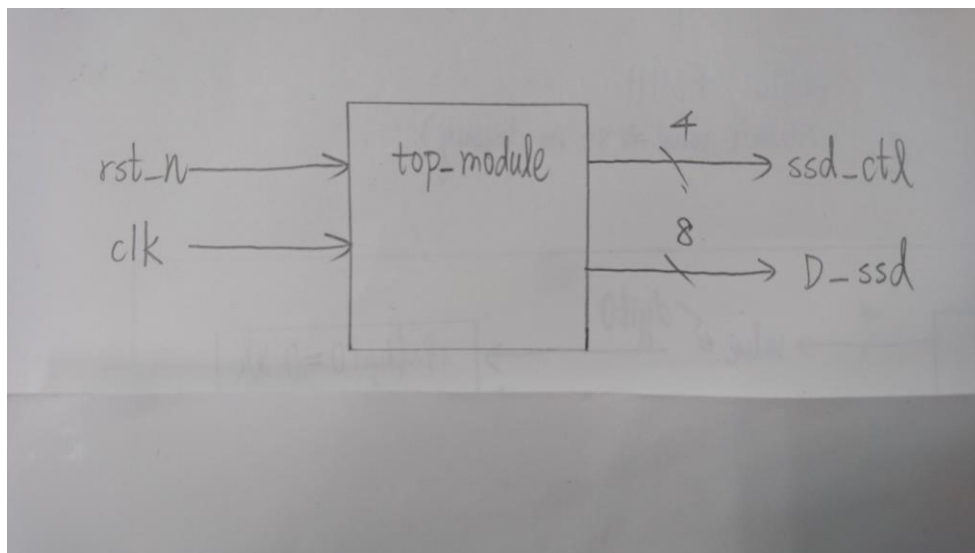
### 3.

#### (1) Design specification :

##### A. Inputs and outputs(表五) :

Inputs	clk, rst_n
Outputs	ssd_ctl[3:0], D_ssd[7:0]
↑ 表五 : Inputs and outputs of 3.	

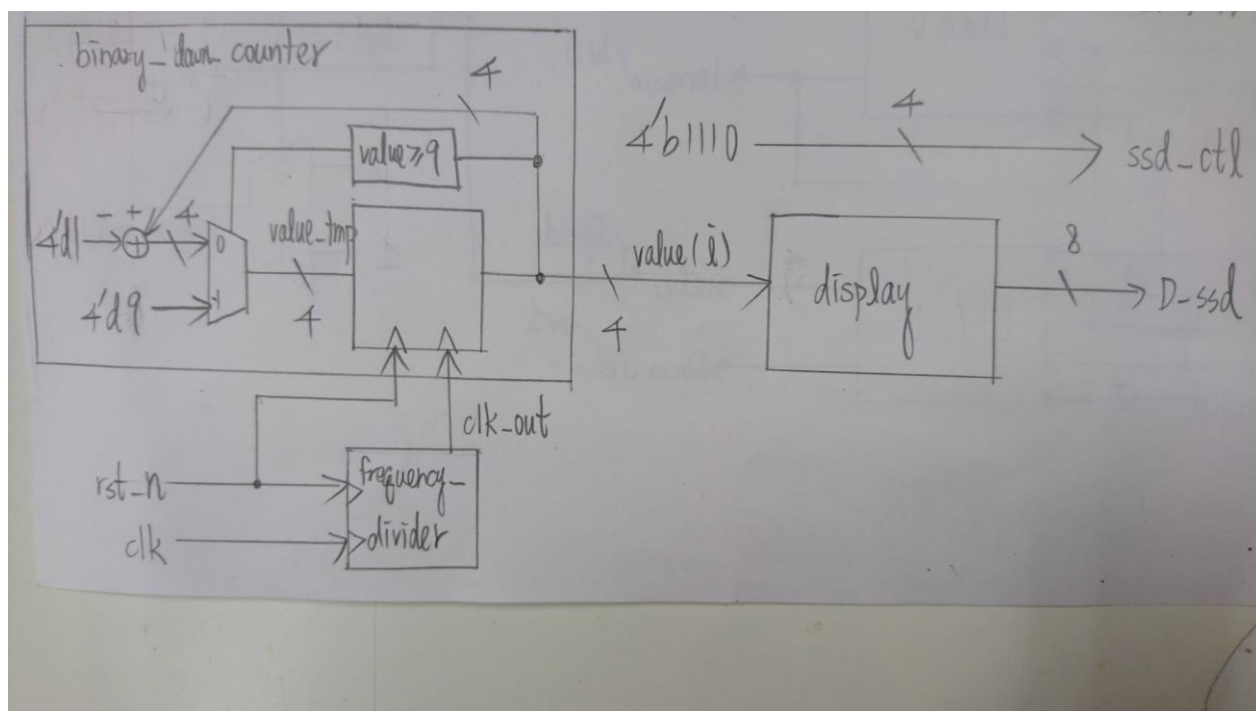
##### B. Block diagram(圖五) :



↑ 圖五 : The block diagram of 3.

#### (2) Design implementation :

##### A. Logic diagram(圖六) :



↑ 圖六 : logic diagram of 3.

B. I/O pin assignment(表六)：

<b>I/O</b>	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	D_ssd[7]	D_ssd[6]	D_ssd[5]
<b>LOC</b>	W4	V4	U4	U2	W7	W6	U8
<b>I/O</b>	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]	rst_n	clk
<b>LOC</b>	V8	U5	V5	U7	V7	W17	W5

↑ 表四：I/O pin assignment of 2.

C.功能與做法說明：

本題為一個 single digit BCD down counter with the divided clock as the clock frequency，並且將數數的結果(value[3:0])用 seven-segment display 來顯示。這題和第二題非常相近，差別只在 counter 模組，這裡是個 binary\_down\_counter，所以要把“加一”換成“減一”，而且當 value 是 0 時要讓下一個 value(value\_tmp)為 9，其餘部分和第三題相同。

4.

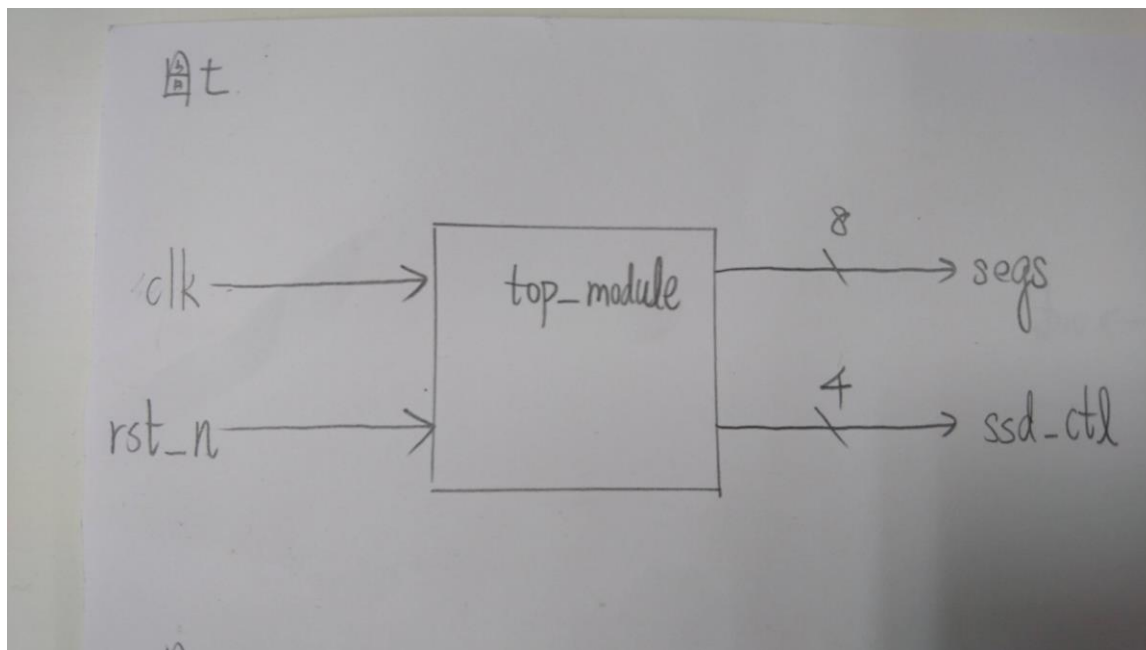
(1) Design specification：

A. Inputs and outputs(表七)：

<b>Inputs</b>	clk, rst_n
<b>Outputs</b>	ssd_ctl[3:0], segs[7:0]

↑ 表七：Inputs and outputs of 4.

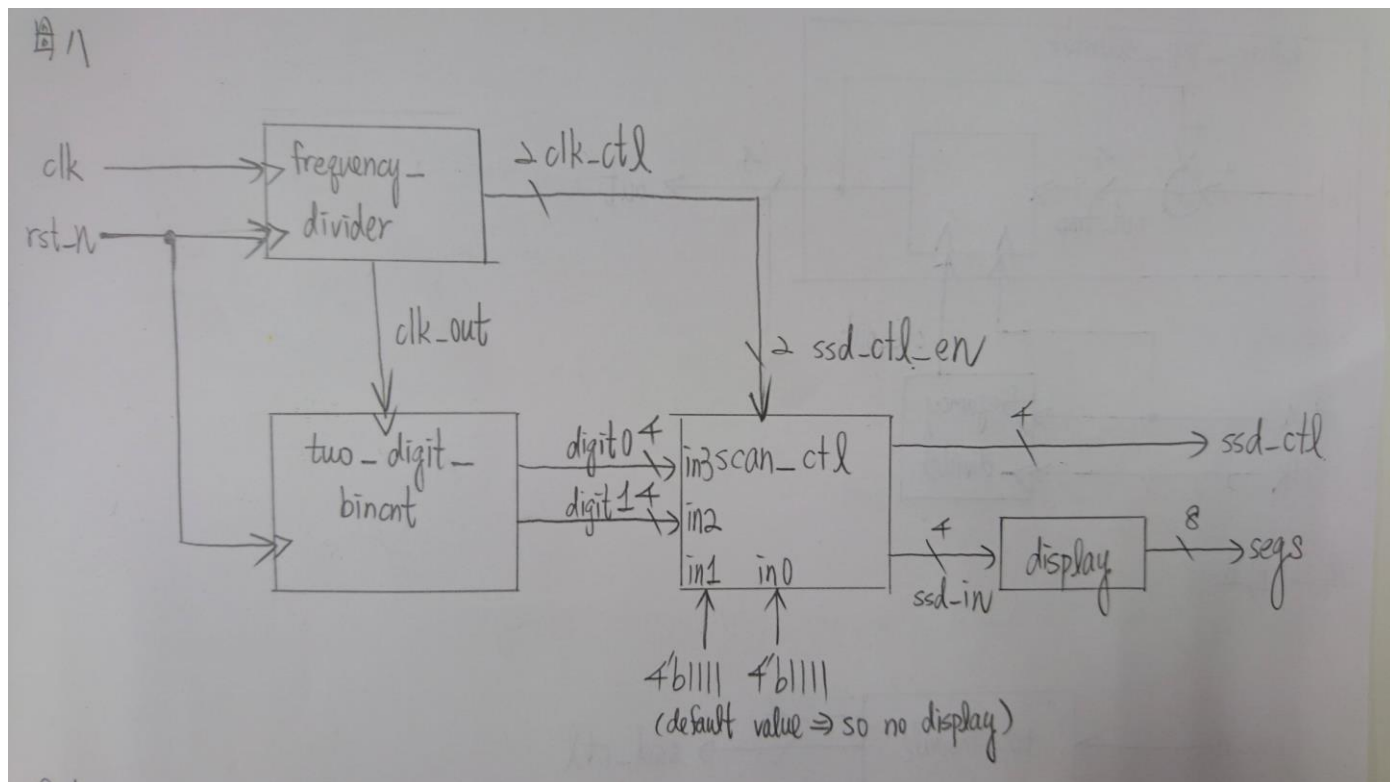
B. Block diagram(圖七)：



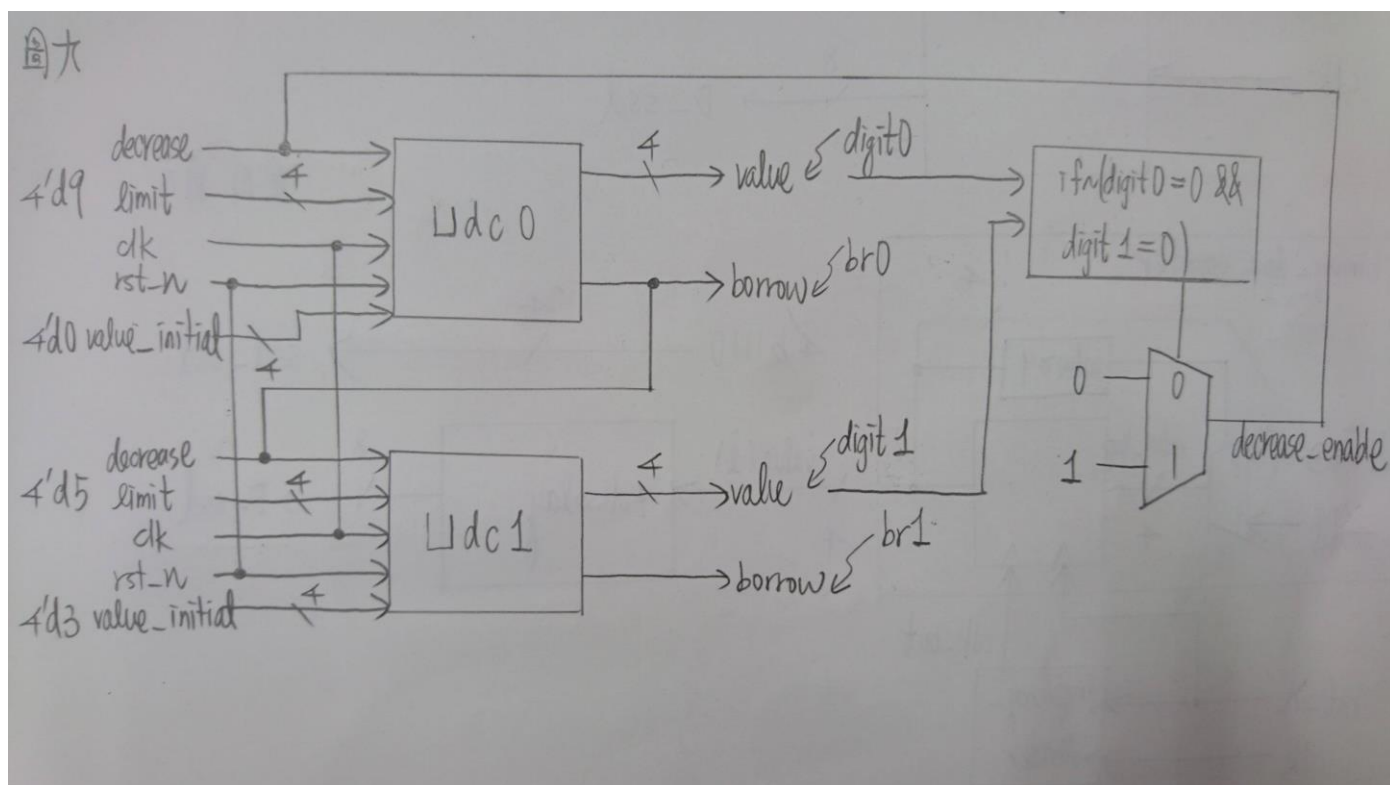
↑ 圖七：The block diagram of 4.

(2) Design implementation :

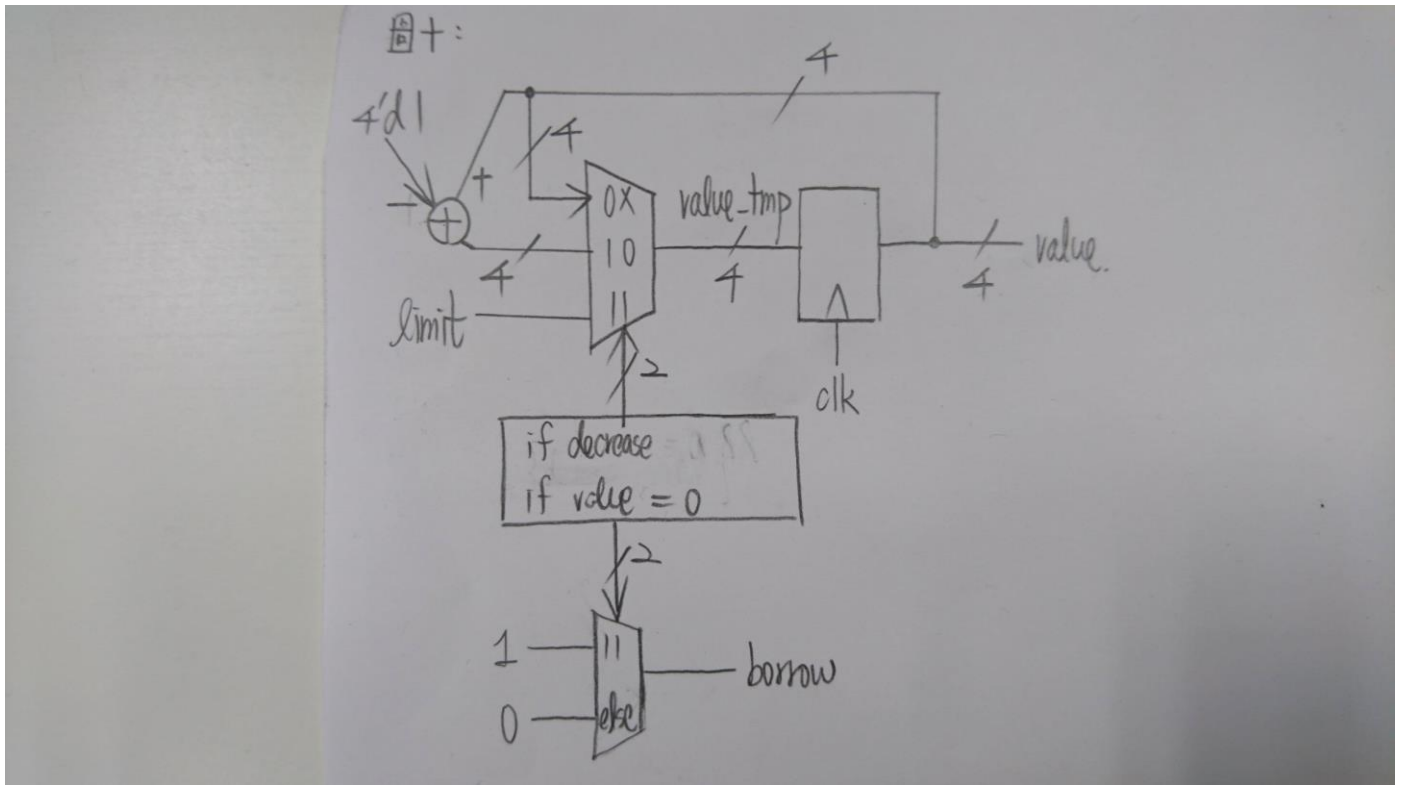
A. Logic diagram(圖八)



↑ 圖八：logic diagram of 4.



↑ 圖九：logic diagram of two\_digit\_bincnt module



↑ 圖十：logic diagram of Udc0 and Udc1 module

#### 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	W4	V4	U4	U2

↑ 表八：I/O pin assignment of 4.

#### C.功能與做法說明：

本題為製作一個 30 秒的 count down timer，並且要停在 00。這題大致上可以分為四個部分：frequency\_divider 模組除頻並拉出兩個 output，一個是頻率 1Hz 的 clock(clk\_out)作為 two\_digit\_bincnt 的 input clock，另一個是 clk\_ctl 作為七段顯示器的視覺佔留；two\_digit\_bincnt 模組為兩位數的 down\_counter，輸出 digit0 和 digit1 表示個位數和十位數要顯示的數字；scan\_ctl 模組和 display 模組前面的 lab 有實作過，輸出 ssd\_ctl 和 segs 負責七段顯示器的顯示。

其中，two\_digit\_bincnt 模組由兩個 down\_counter 組成，一個負責個位數，一個負責十位數，輸入和輸出分別有：decrease(決定是否要下數)、limit(決定 0 之後的下一個數)、clk、rst\_n、value\_initial(決定 reset 後的數字)；value(數數的結果)、borrow(是否要和下一位借位，如果 borrow = 1 則下一位的 decrease = 1)。

最後利用一個 top\_module 把所有的模組包在一起。

## 5. Discussion

第一題只要注意接線的時候不要接錯，基本上不會有太大的困難。

第二題和第三題相當類似，只要做出第二題就一定能做出第三題，基本上也只要注意接線不要接錯，就不會有太大的問題。

第四題難度最高，首先要先設計出一個 **two-digit** 的 **down counter**，還好老師上課的時候有上到，所以這部份沒什麼問題。再來是要使七段顯示器輸出兩個不同的數字，所以要多接個 **scan\_ctl** 模組，這部分因為上次 **lab** 有做過，所以也沒太大的問題。最困難的部份是要控制 **counter** 從 30 開始數，而且數到 0 的時候要停住：要控制 **counter** 從 30 開始只要加一個 **value\_intial** 表示 **counter** 在 **rst\_n** 之後的初始狀態，這部分問題也不大；我花比較多時間在讓 **counter** 數到 0 的時候要停住。最後我找出了兩種方法：一種是改變 **limit**，讓十位數的 **limit = 0**，個位數的 **limit** 本來是 9，當數到 00 時把它的 **limit** 變成 0，如此便能讓 **counter** 最後停在 00；不過上述的方式就浪費了個位數的 **decrease** 了，其實只要控制個位數的 **decrease**，讓個位數的 **decrease** 在 **counter** 數到 00 時變為 0，**counter** 就可以停在 00。

## 6. Conclusion

這次的 **Lab** 讓我有能力可以用 **verilog** 實現更多的功能，並能處理更複雜的問題。