

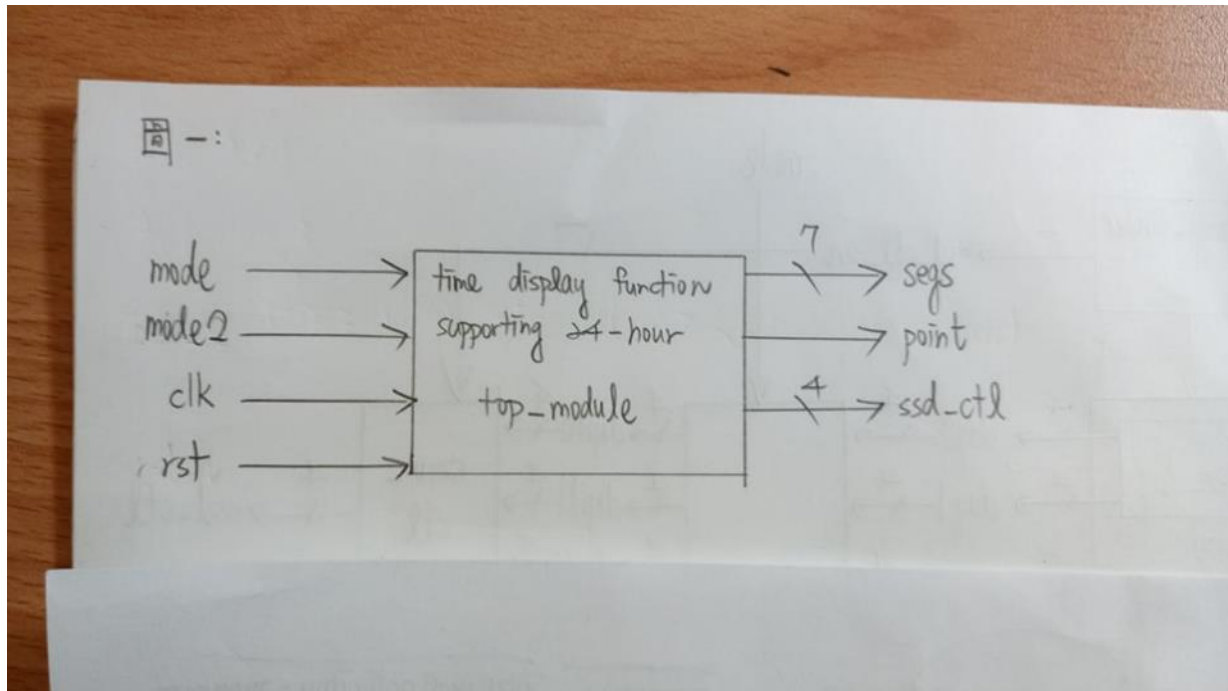
1.

(1) Design specification :

A. Inputs and outputs(表一) :

Inputs	clk, rst, mode, mode2
Outputs	ssd_ctl[3:0], segs[6:0], point
↑ 表一 : Inputs and outputs of 1	

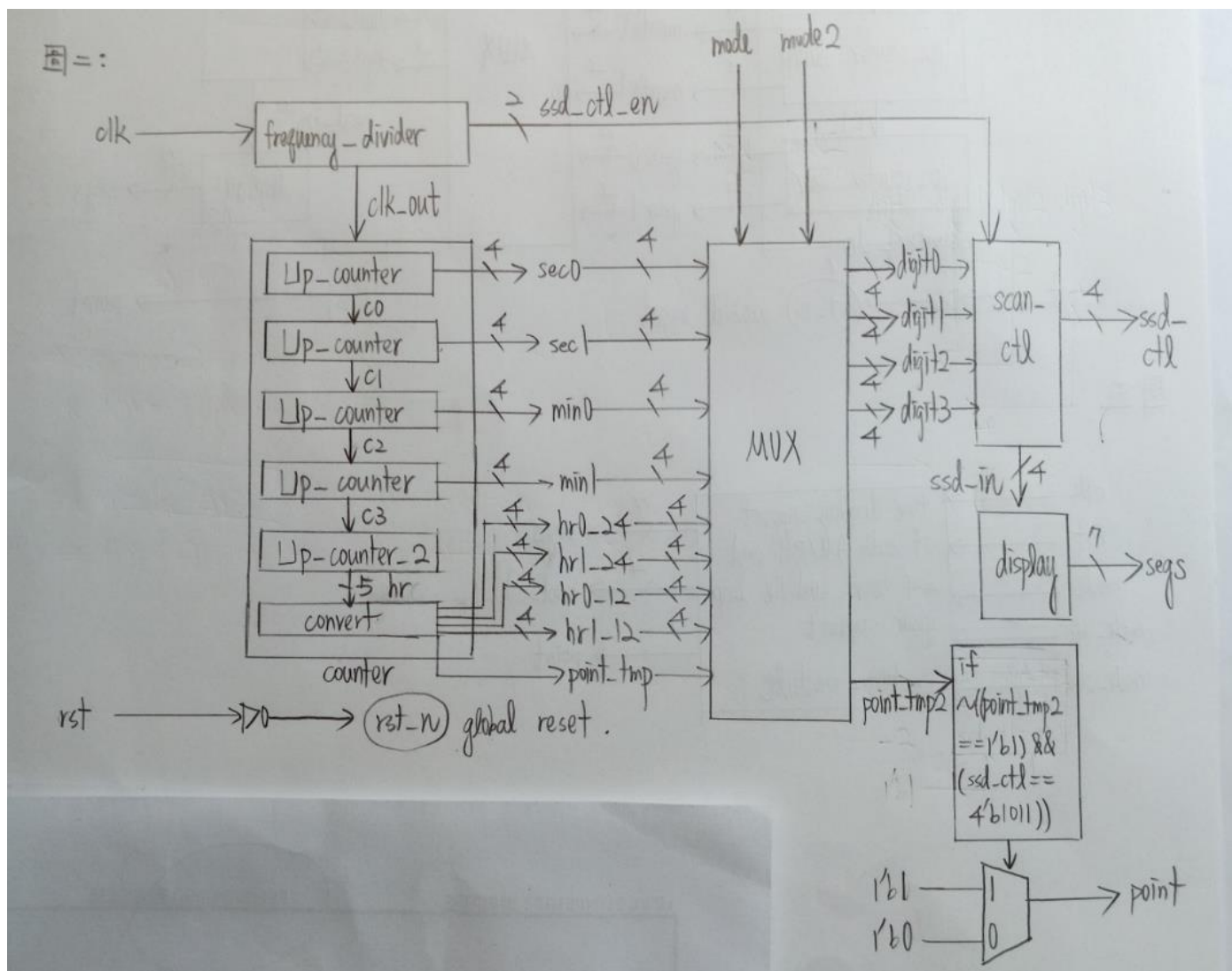
B. Block diagram(function table)(圖一) :



↑ 圖一 : The block diagram of 1

(2) Design implementation :

A. Logic diagram(function table)(圖二) :



B. I/O pin assignment(表二) :

I/O	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	point	segs[6]	segs[5]
LOC	W4	V4	U4	U2	V7	W7	W6
I/O	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]	mode	mode2
LOC	U8	V8	U5	V5	U7	V17	V16
I/O	rst	clk					
LOC	T17	W5					

↑ 表二：I/O pin assignment of 1

C.功能與做法說明：

本題為實現時鐘顯示時間的功能，總共有兩個模式：時分和秒，其中時分又可分為兩個模式：AM/PM 制和 24 小時制。

本題最重要的兩個模組為 counter 和 MUX：counter 模組共有 6 個子模組，首先前四個為 Up_counter 模組，負責數出秒的個位(sec0)、秒的十位(sec1)、分的個位(min0)、分的十位(min1)，模組和模組之間有 carry 相連，決定進位與否；接著是 Up_counter_2，他負責數出小時(hr)，並且將數出的結果輸入 convert 模組進行 AM/PM 制和 24 小時制的轉換；convert 模組其實就是多功器，決定如何將數出的結果(hr)轉換成 24 小時和 AM/PM 制，並決定何時要顯示點(point_tmp)，最後輸出 AM/PM 制(hr0_12、hr1_12)和 24 小時制(hr0_24、hr1_24)的小時。MUX 模組負責從 sec0、sec1、min0、min1、hr0_12、hr1_12、hr0_24、hr1_24 中決定最後七段顯示器要輸出哪四個數字，由兩個 DIP switch 控制(mode 和 mode2)，mode 決定顯示的時分為 AM/PM 制或 24 小時制、mode2 決定顯示時分還是秒。

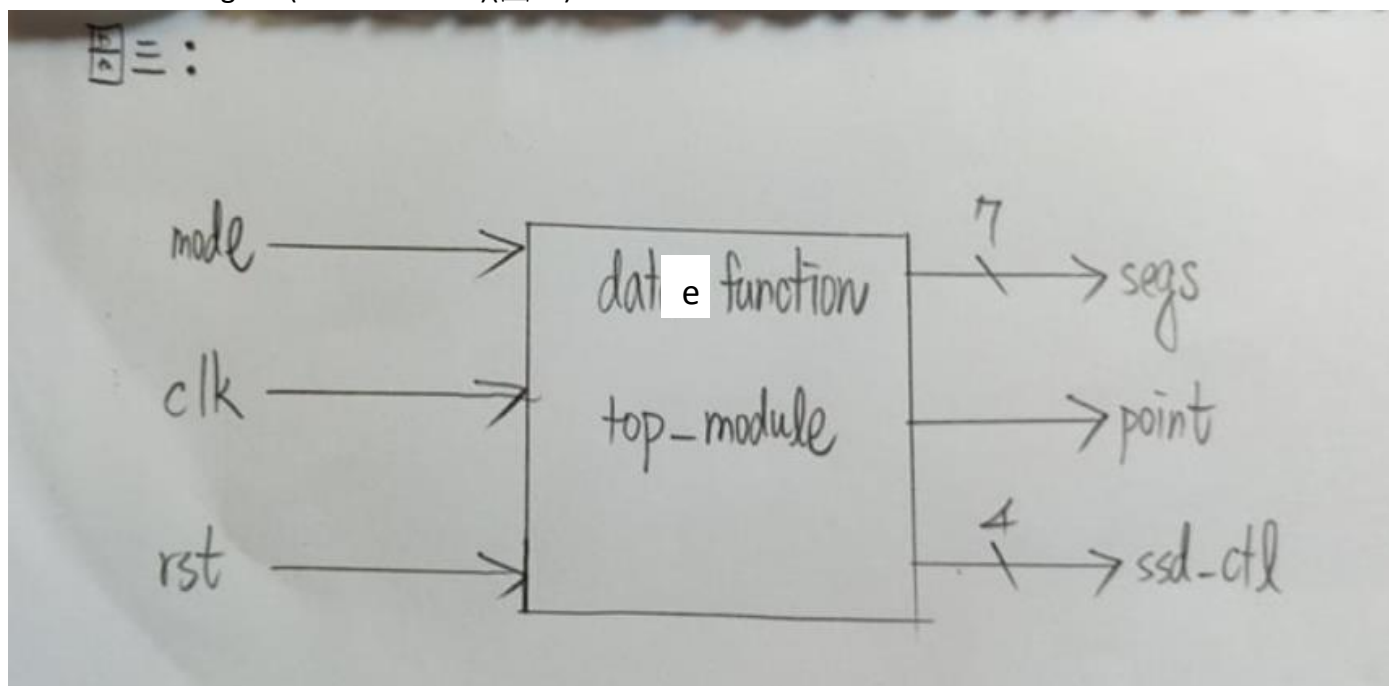
2.

(1) Design specification :

A. Inputs and outputs(表三)：

Inputs	clk, rst, mode
Outputs	ssd_ctl[3:0], segs[6:0], point
↑ 表三：Inputs and outputs of 2	

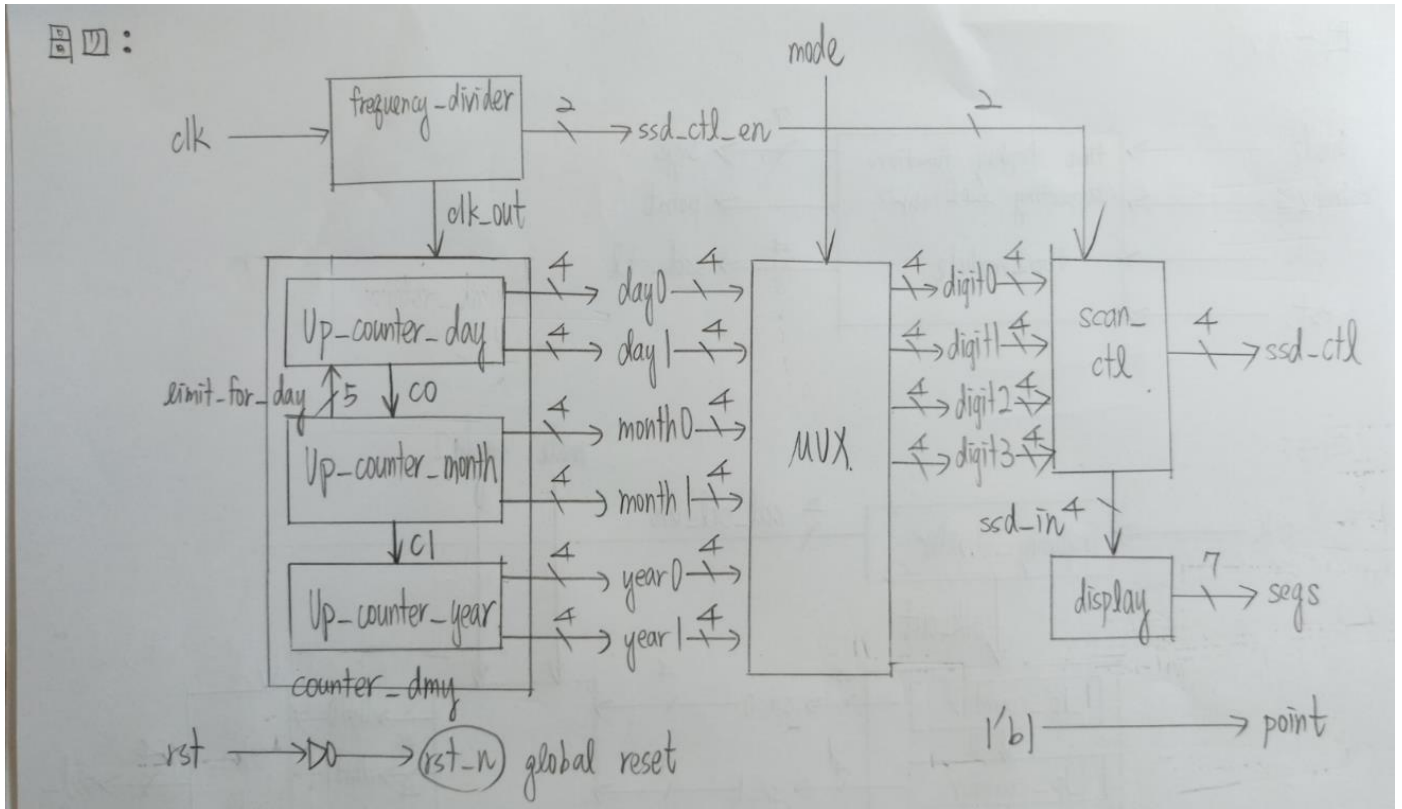
B. Block diagram(function table)(圖三)：



↑ 圖三：The block diagram of 2

(2) Design implementation :

A. Logic diagram(function table)(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]	point	segs[6]	segs[5]
LOC	W4	V4	U4	U2	V7	W7	W6
I/O	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]	mode	clk
LOC	U8	V8	U5	V5	U7	V17	W5
I/O	rst						
LOC	T17						

↑ 表四：I/O pin assignment of 2

C.功能與做法說明：

本題為實現時鐘顯示年月日的功能，總共有兩個模式：月日和年。

本題最重要的兩個模組為 `counter_dmy` 和 `MUX`：`counter_dmy` 模組有三個子模組，分別是 `Up_counter_day`、`Up_counter_month` 和 `Up_counter_year`，三個模組分別輸出日的個位(`day0`)、日的十位(`day1`)、月的個位(`month0`)、月的十位(`month1`)、年的個位(`year0`)、年的十位(`year1`)。三者間以 `carry` 連接，決定是否要進位。比較需要注意的一點是，`Up_counter_month` 會決定 `Up_counter_day` 的 `limit`，因為不同的月份會有不同的日子數，所以在這兩個模組間有條 `wire(limit_for_day)` 依據 `Up_counter_month` 數到哪個月來決定 `Up_counter_day` 的 `limit`。`MUX` 模組則是依據不同的 `mode`(月日和年)來決定七段顯示器要顯示哪些數字。

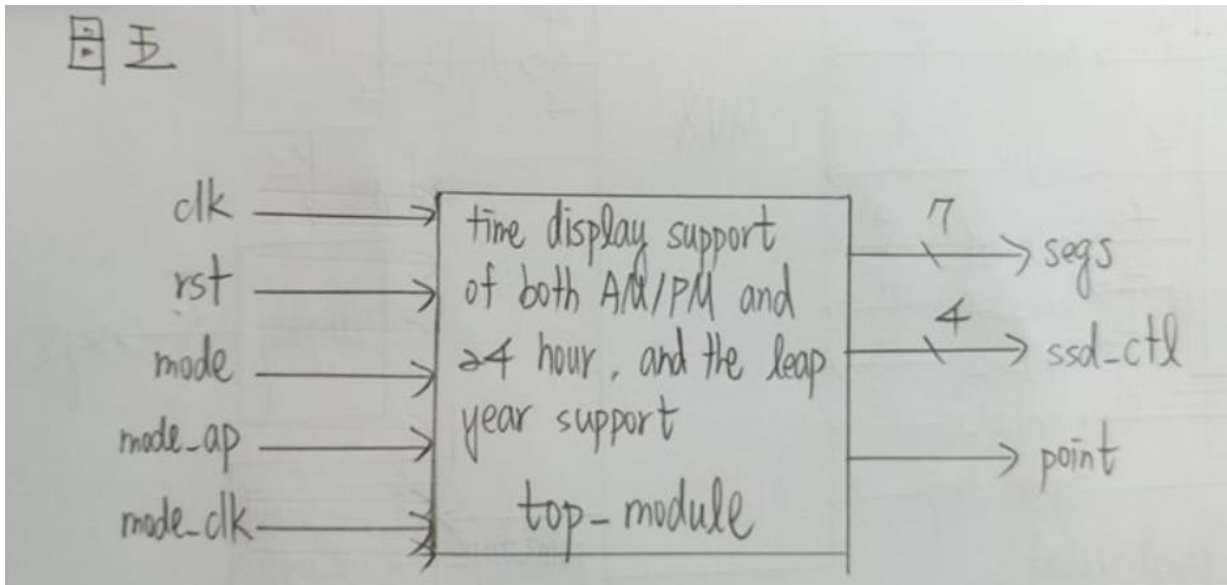
3.

(1) Design specification :

A. Inputs and outputs(表五) :

Inputs	clk, rst, mode, mode_ap, mode_clk
Outputs	ssd_ctl[3:0], segs[6:0], point
↑ 表五 : Inputs and outputs of 3	

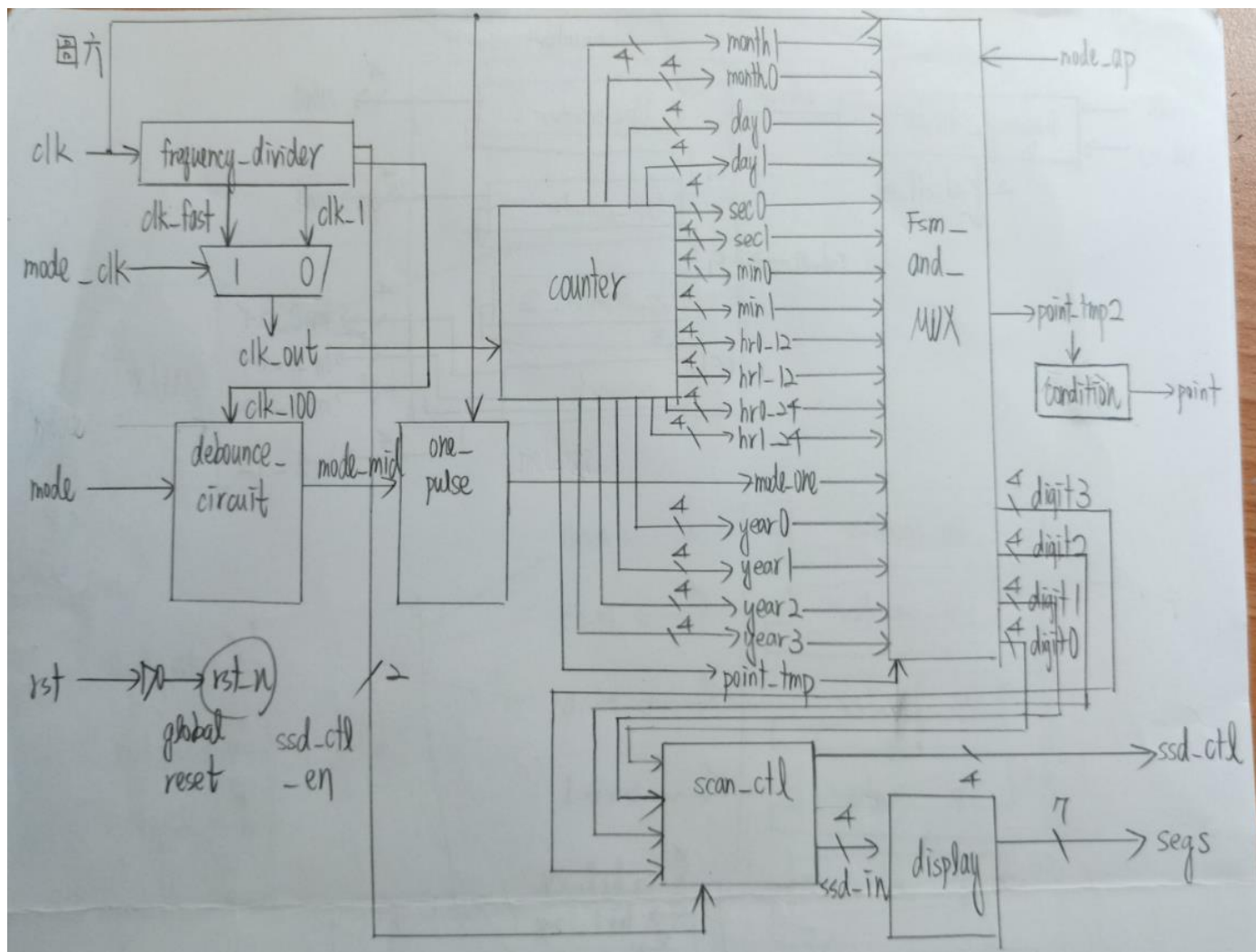
B. Block diagram(function table)(圖五) :



↑ 圖五 : The block diagram of 3

(2) Design implementation :

A. Logic diagram(function table)(圖六) :



↑ 圖六：logic diagram of 3

B. I/O pin assignment(表六) :

I/O	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	point	segs[6]	segs[5]
LOC	W4	V4	U4	U2	V7	W7	W6
I/O	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]	mode	mode_ap
LOC	U8	V8	U5	V5	U7	T17	V17
I/O	mode_ap	rst	clk				
LOC	V17	U17	W5				

↑ 表六：I/O pin assignment of 3

C.功能與做法說明：

本題為整合前面兩題的功能，並且要讓年可以顯示從 2000 到 2200。

本題最重要的模組為 counter 和 FSM_and_MUX：counter 模組整合了前兩題的 counter 和 counter_dmy 模組，從秒到分到時到日到月到年，彼此之間以 carry 相連，最後輸出數出來的結果，送到 FSM_and_MUX 來決定最後七段顯示器要顯示的數字。FSM_and_MUX 包含了一個 FSM，總共有四個 state：state0 顯示年、state1 顯示月日、state2 顯示時分、state3 顯示秒，按鈕輸入來決定 state 的轉換，其中在 state2 時，如果扳動 V17 DIP switch，可以決定要顯示 24 小時制還是 AM/PM 制。

5. Discussion

本次的 Lab 因為不涉及按鈕的長按短按，以及複雜的 FSM 設計，所以相較 Lab5 我覺得比較容易。我覺得比較不同且花時間的地方有：counter 的設計，尤其是 12 小時制和 AM/PM 轉換的地方，花了我一些時間，而且 counter 不一定只能數個位數，數到 1000、2000 都是可以的，這是和前面 Lab 不大相同的地方；再來是要決定讓七段顯示器顯示哪四個數字，必須加個 MUX 來做選擇；最後是整合前兩題的功能，必須用 FSM 來決定現在在哪個 state(功能)，以此來決定七段顯示器的四個輸出。

6. Conclusion

在寫完上次的 Lab 後，這次的 Lab 輕鬆許多，是 counter 的延伸，讓我學會更多 counter 的應用，進一步完成更多時鐘的功能。