

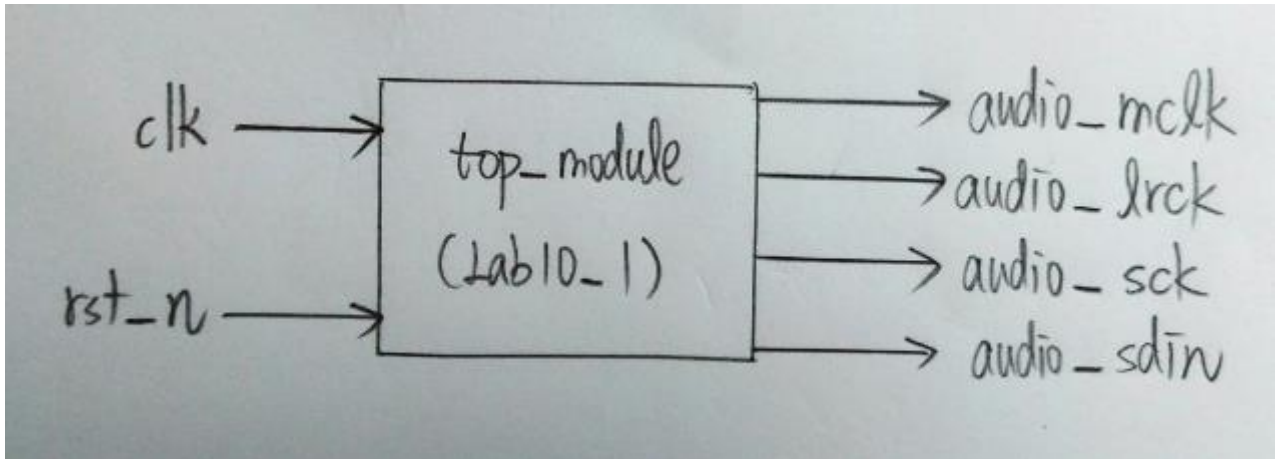
1.

(1) Design specification :

A. Inputs and outputs(表一) :

Inputs	rst_n, clk
Outputs	audio_mclk, audio_lrck, audio_sck, audio_sdin
↑ 表一 : Inputs and outputs of 1	

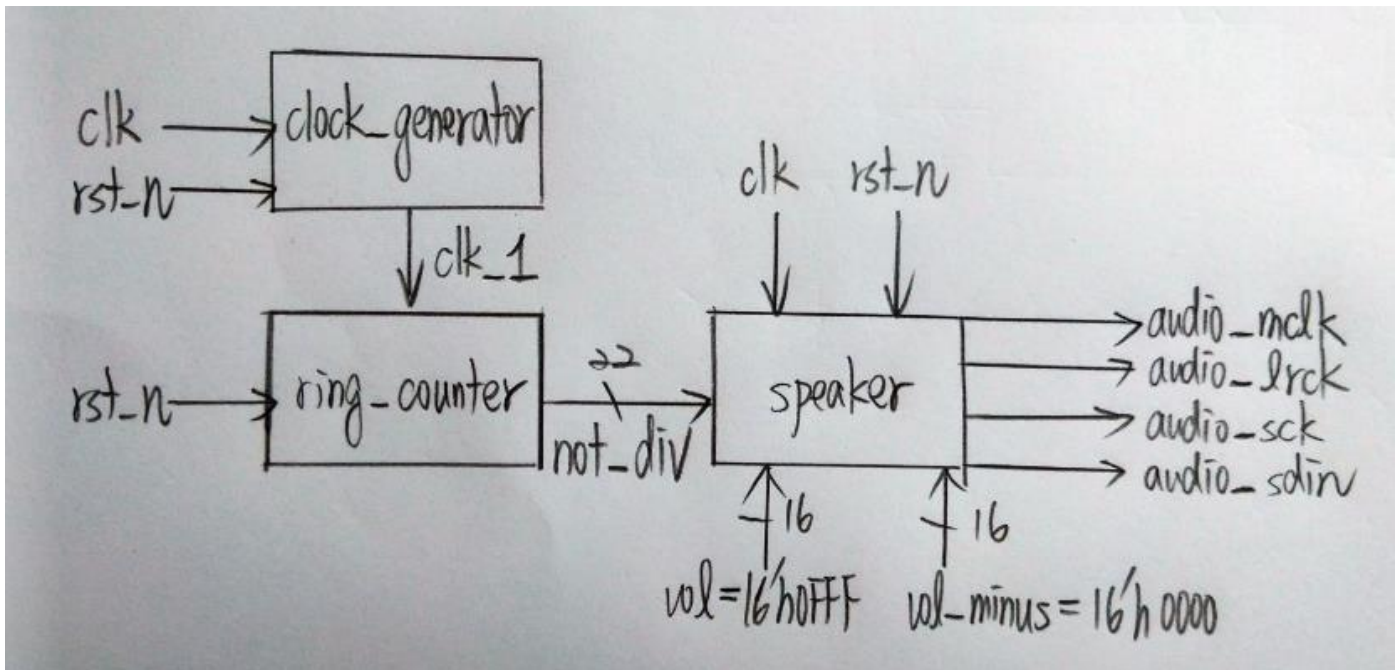
B. Block diagram(function table)(圖一) :



↑ 圖一 : The block diagram of 1

(2) Design implementation :

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



↑ 圖二 : logic diagram of 1

B. I/O pin assignment(表二)：

I/O	clk	rst_n	audio_mclk	audio_lrck	audio_sck	audio_stdin	
LOC	W5	V17	A14	A16	B15	B16	

↑ 表二：I/O pin assignment of 1

C.功能與做法說明：

本題內容為製作一個可以重複撥放 Do, Re, Mi, Fa.....的 speaker。

最重要的模組是 ring counter 模組，決定在第幾秒的時候 speaker 應該要播放什麼音。

counter 數到不同值時會有對應的 note_div 輸出，note_div 會決定聲音的輸出頻率，也就是音高。speaker 模組和 Lab8 相同。

2

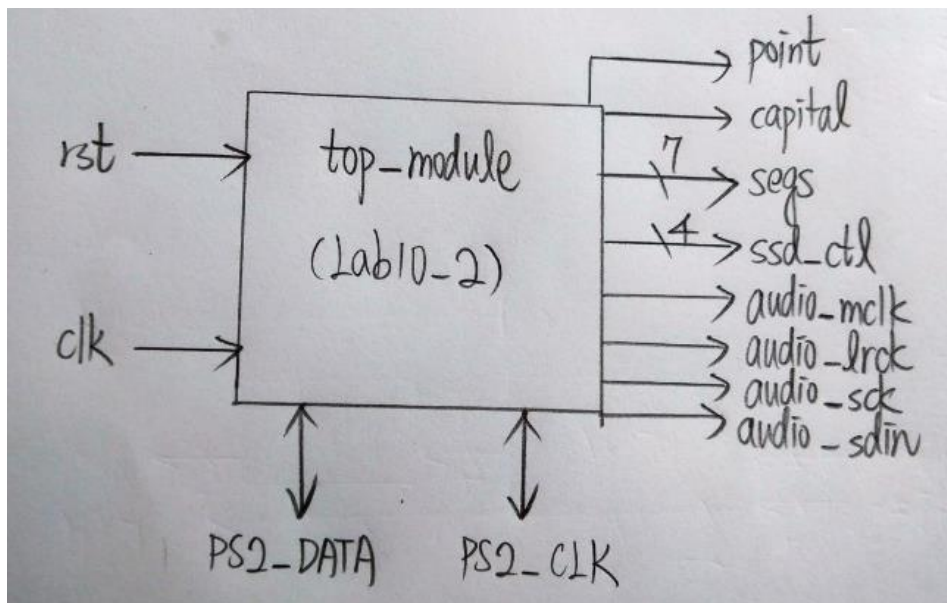
(1) Design specification：

A. Inputs and outputs(表三)：

Inputs	rst, clk
Outputs	[6:0]segs, [3:0]ssd_ctl, point, capital, audio_mclk, audio_lrck, audio_sck, audio_sdin
Inouts	PS2_DATA, PS2_CLK

↑ 表三：Inputs and outputs of 2

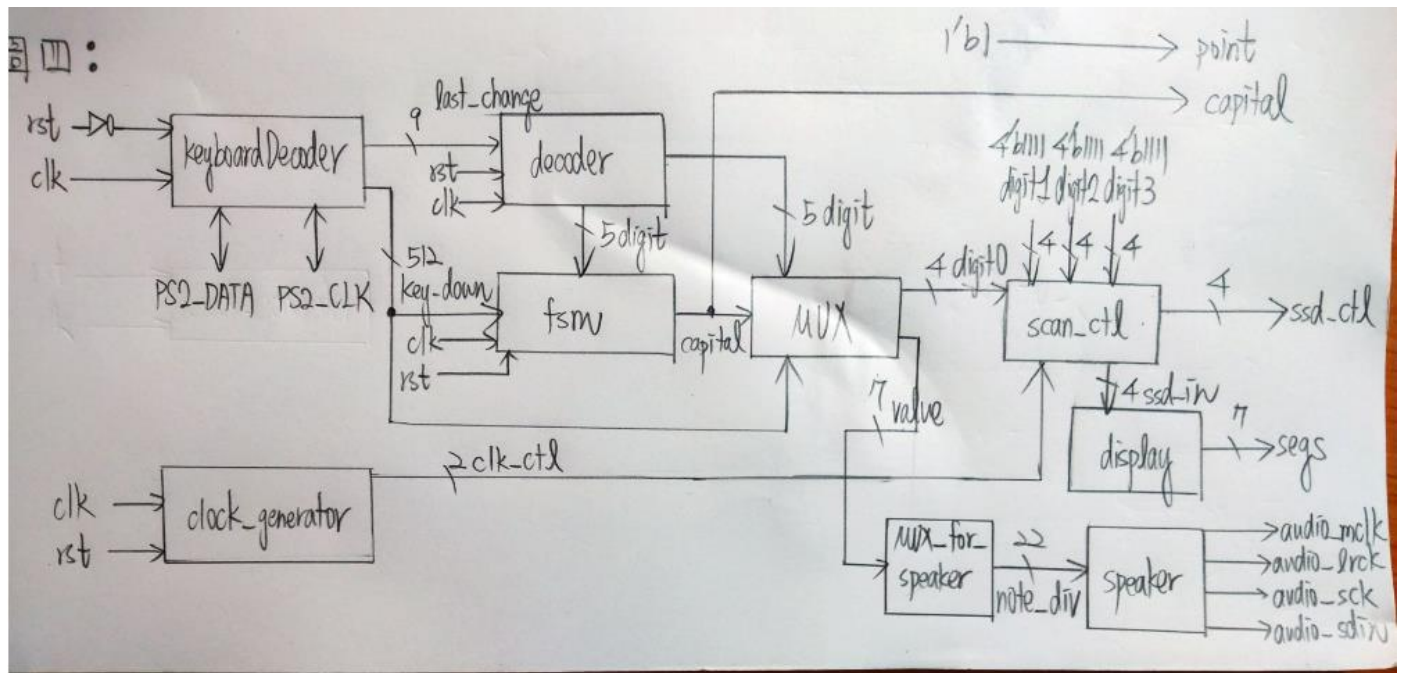
B. Block diagram(function table)(圖三)：



↑ 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	clk	rst	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	segs[6]
LOC	W5	V17	W4	V4	U4	U2	W7
I/O	segs[5]	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]	point
LOC	W6	U8	V8	U5	V5	U7	V7
I/O	PS2_CLK	PS2_DATA	audio_mclk	audio_lrck	audio_sck	audio_stdin	capital
LOC	C17	B17	A14	A16	B15	B16	L1

↑ 表四：I/O pin assignment of 2

C.功能與做法說明：

本題為結合 keyboard 和 speaker，當按下 c,d,e,f,g,a,b 會產生中音 Do 到 Si 的聲音，當按下 C,D,E,F,G,A,B 會產生高八度的 Do 到 Si 的聲音。並且七段顯示器會顯示目前產生的音高(Do -> 1, Re -> 2, Mi -> 3, Fa -> 4, So -> 5, La -> 6, Si -> 7)。

本題其實就是 Lab9_4 和上一題的結合。首先處理 keyboard 訊號：decoder 將由 KeyboardDecoder 模組得到的 last_change 訊號做 decode，得到相對應的 output，將其輸入 fsm 作為控制訊號。fsm 總共有四個 state：小寫 state、大寫 state 和兩個 buffer state，其中：小寫 state 的 capital = 1'b0、大寫 state 的 capital = 1'b1，buffer state 則是用來區分是按了 caps lock 兩次或一次(判斷是否有放開鍵盤)。MUX 則是輸入 capital、digit 和 key_down 來判斷是在 capital = 1'b0 或 capital = 1'b1 的情況下，以及是否有按 shift，由此來判斷按的是 c,d,e,f,g,a,b 還是 C,D,E,F,G,A,B，藉此決定七段顯示器應該顯示的值。

然後將 MUX 得到的值送到顯示模組顯示，以及將它送到 MUX_for_speaker 決定 speaker 的 note_div 值，最後送到 speaker 產生相對應音高的聲音。

3

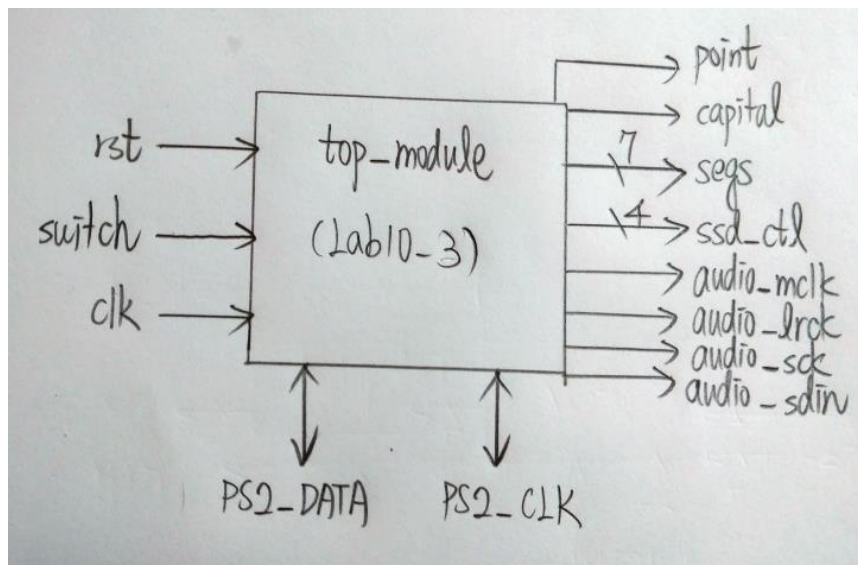
(1) Design specification :

A. Inputs and outputs(表五) :

Inputs	rst, clk, switch
Outputs	[6:0]segs, [3:0]ssd_ctl, point, capital, audio_mclk, audio_lrck, audio_sck, audio_sdin
Inouts	PS2_DATA, PS2_CLK

↑ 表五 : Inputs and outputs of 3

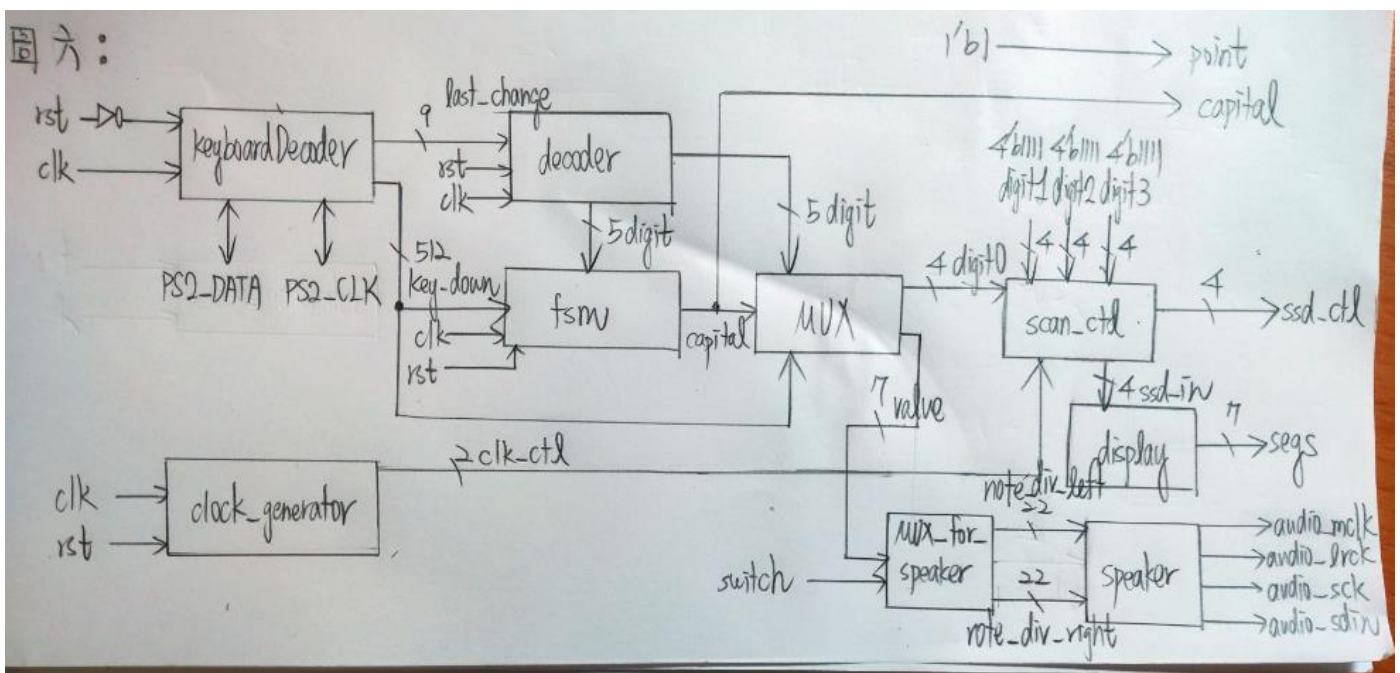
B. Block diagram(function table)(圖五) :



↑ 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	clk	rst	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	segs[6]
LOC	W5	V17	W4	V4	U4	U2	W7
I/O	segs[5]	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]	point
LOC	W6	U8	V8	U5	V5	U7	V7
I/O	PS2_CLK	PS2_DATA	audio_mclk	audio_lrck	audio_sck	audio_stdin	capital
LOC	C17	B17	A14	A16	B15	B16	L1
I/O	switch						
LOC	V16						

↑ 表六：I/O pin assignment of 3

C.功能與做法說明：

本題和前一題非常類似，差別只在於要多接一個開關 input，當開關 on 時 speaker 必須要能夠輸出雙聲道的聲音。

所以需要更改的地方只有 speaker 模組的 buzzer_control，需要讓 audio_left 和 audio_right 的 b_clk 互相獨立，如此才能使 audio_left 和 audio_right 產生不同音高的聲音。所以我在 buzzer_control 模組中加入另一個 frequency divider，產生 b_clk_2 決定 audio_right 的聲音頻率，原本的 b_clk 則是決定 audio_left 的聲音頻率，並且外接 note_div_left 和 note_div_right 來決定除頻的頻率，由此改變 b_clk 和 b_clk_2 的頻率，藉此改變左聲道和右聲道聲音的音高，便可產生和弦。

5. Discussion

本次的 Lab 主要是結合 Lab8 和 Lab9，讓我對 keyboard 和 speaker 有更深入的認識，並且學會如何產生雙聲道聲音，讓聲音的呈現可以更加豐富。

由這次實驗也讓我再次體會到模組化的重要性，如果模組化做的好，這次的 Lab 便可以相當輕鬆，因為其實只要把前面做的功能結合在一起即可。