

# Lab10: Electronic Ogran

107061112 王昊文

## Experiment

1. Play the 14 sounds repeatedly based on the sound table. Every sound is played for one second.

## Design Specification

✓ I/O

**Input** clk; // crystal clock

**Input** rst\_n; // low active reset

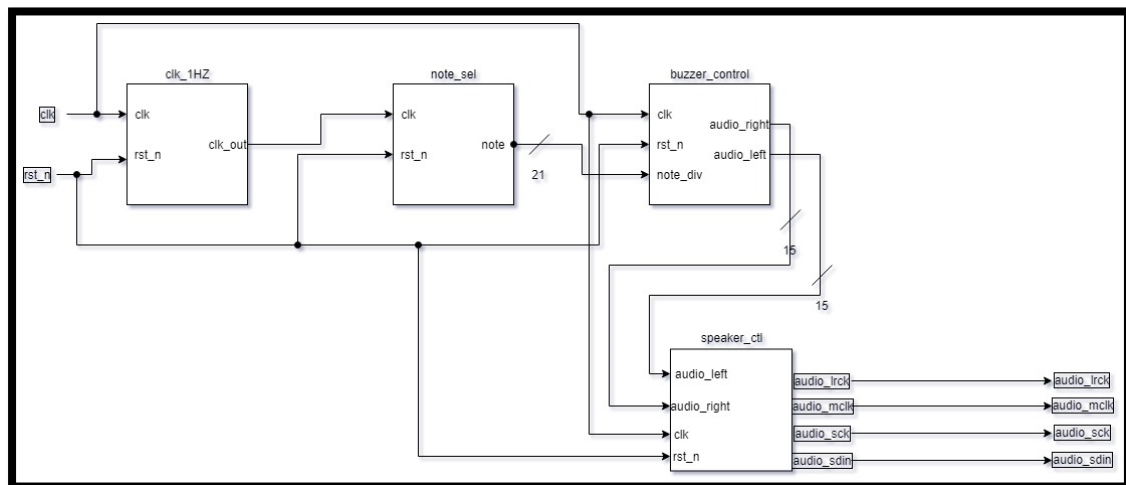
**Output** audio\_mclk; // master clock 25MHZ

**Output** audio\_lrck; // left-right clock 25MHZ / 128, sending signals

**Output** audio\_sck; //serial clock 25MHZ / 4

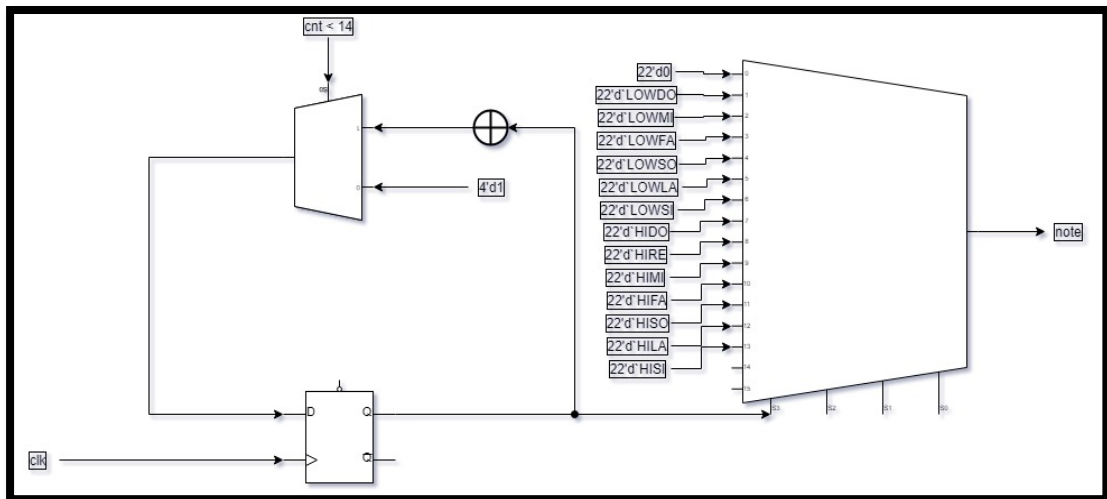
**Output** audio\_sdin; // serial audio data input, lrck to send signals

✓ Logic Block:



第一題想法很簡單，關於後半部的聲音處理，我們只需要先將之前用過的模組即可。我們只需要考慮如何輪流播放不同的音，從滴到高，而且一秒一個。我使用了一個note\_sel的模組，利用一赫茲的counter去數，然後從1數到14，數到不同值的時候輸出不同值即可。後面會有詳細說明。

✓ Note\_sel:



從上圖就可以很清楚看出，利用類似ring counter的概念不斷從1到14輪迴，然後數到不同值的時候輸出不同的音。

## ✓ Design Implementation

Input	
<i>clk</i>	<i>rst_n</i>
W5	V17

Output			
<i>audio_mclk</i>	<i>audio_lrck</i>	<i>audio_sck</i>	<i>audio_sdin</i>
A14	A16	B15	B16

## ✓ Discussion

這一題非常簡單，只需要透過counter的特性就可以達到目標。這一切都讓上學期學的邏輯設計都串連起來，（儘管細部的邏輯都忘記了），利用counter可以幫助我們完成很多事情，這讓我想到之後final project，如果要做遊戲的話可以應用在很多地方。

## Experiment

### 2. Electronic Organ

2.1 Integrate the keypad as the keyboard of the electronic organ. Keys c, d, e, f, g, a, b, C, D, E, F, G, A, B (two octaves from mid-Do) represent the sounds from low to high frequencies.

2.2 Display your playing sound (Do, Re, Mi, Fa, So, La, Si) in the 7-segment LED

## Design Specification

✓ I/O

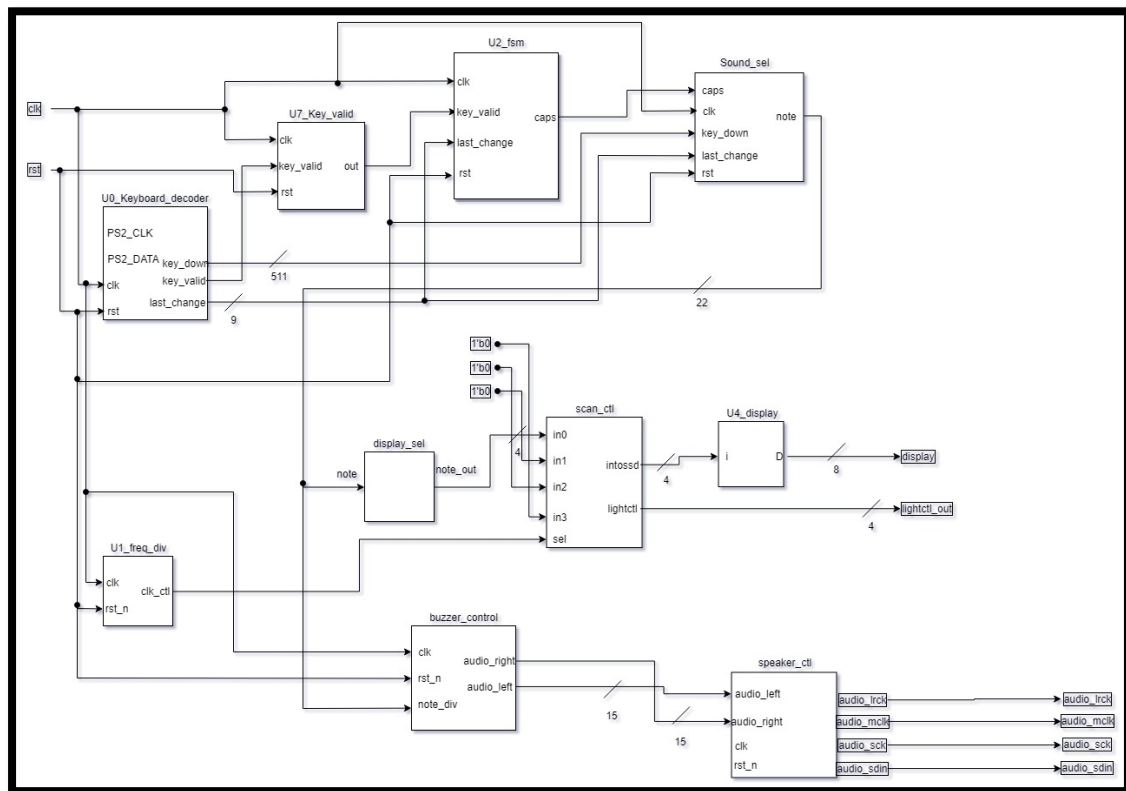
Inout PS2\_DATA, PS2\_CLK; // keyboard data

**Input** clk; // crystal clock  
**Input** rst; // high active reset

**Output** audio\_mclk; // master clock 25MHZ  
**Output** audio\_lrclk; // left-right clock 25MHZ / 128, sending signals  
**Output** audio\_sck; // serial clock 25MHZ / 4  
**Output** audio\_sdin; // serial audio data input, lrclk to send signals

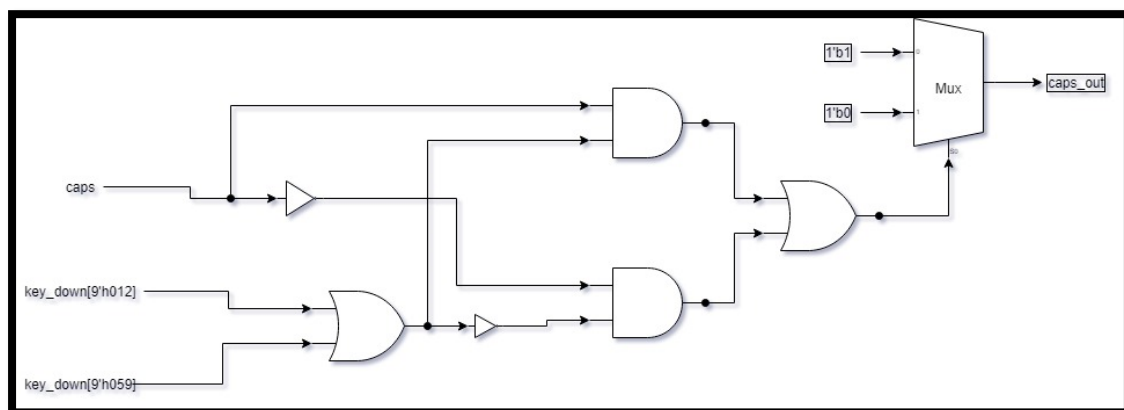
// display outputs  
**Output** [3:0]lightctl\_out;  
**Output** [7:0]display;

✓ Logic Block:



在上圖中我們沿用了上個lab的模組，來達成大小寫不同音的效果，因此我們利用上個lab的模組來判斷現在是否進入大寫狀態，其中的key\_valid，fsm模組就是。再來就是如果目前是大寫狀態，我們就必須透過選擇來輸出想要的音色。其中sound\_sel就是做這樣的功能，後面會有詳細的介紹。然後，透過目前輸出的音色，來選擇輸出的7段顯示器輸出，

✓ Sound\_sel



基本上這個就是沿用前面lab的模組。沿用到這個lab也很簡單，只要判斷目前為caps狀態，就輸出高八度音，若不是caps狀態，按鍵就輸出較低八度的音。至於輸出什麼音的方法，基本上就是利用keyboard decoder輸出的key\_down來判斷目前為cdefgab哪個按鍵，然後做相對應的頻率輸出即可。（利用MUX來選擇，例如：if key\_down == [9'h021]，則輸出do的音）。

#### ✓ display\_sel

至於display\_sel的部分，負責7段顯示器輸出的內容，利用MUX去做，判斷如果現在輸出的音，則輸出1，如果目前按下去的音為re，則輸出2，依此類推。

#### ✓ Design Implementation

Inout	
<i>PS2_CLK</i>	<i>PS2_DATA</i>
C17	B17

Input	
<i>clk</i>	<i>rst_n</i>
W5	V17

Audio Output			
<i>audio_mclk</i>	<i>audio_lrck</i>	<i>audio_sck</i>	<i>audio_sdin</i>
A14	A16	B15	B16

Output ssd							
<i>display[0]</i>	<i>display[1]</i>	<i>display[2]</i>	<i>display[3]</i>	<i>display[4]</i>	<i>display[5]</i>	<i>display[6]</i>	<i>display[7]</i>
V7	U7	V5	U5	V8	U8	W6	W7

Output lightctl			
<i>lightctl[0]</i>	<i>lightctl[1]</i>	<i>lightctl[2]</i>	<i>lightctl[3]</i>
U2	U4	V4	W4

## ✓ Discussion

這一題也不難，只是將鍵盤的功能一起合起來，功能就變的很強大。除了聲音之外，如果應用在遊戲，我們也可以思考怎麼活用鍵盤，例如控制角色的走動等等，或是按下什麼鍵會發出系統特製的聲音等等，這一下讓我們可以發揮的地方大大增加。這一題讓我看到了FPGA功能的多元性。

## Experiment

3. (Bonus) Playback double tones by separate left and right channels. If you turn one DIP switch off, the electronic organ playback single tone when you press keyboard (as in Prob. 2). If you turn DIP switch on, left (right) channels play Do(Mi), Re(Fa), Mi(So), Fa(La), So(Si) when you press the keyboard.

## Design Specification

✓ I/O

**Inout** PS2\_DATA, PS2\_CLK; // keyboard data  
**Input** clk; // crystal clock  
**Input** rst; // high active reset

**Output** audio\_mclk; // master clock 25MHZ

**Output** audio\_lrck; // left-right clock 25MHZ / 128, sending signals

**Output** audio\_sck; //serial clock 25MHZ / 4

**Output** audio\_sdin; // serial audio data input, lrck to send signals

✓ Logic Block:



W5	V17
----	-----

Audio Output			
<i>audio_mclk</i>	<i>audio_lrclk</i>	<i>audio_sck</i>	<i>audio_sdin</i>
A14	A16	B15	B16

Output ssd							
<i>display[0]</i>	<i>display[1]</i>	<i>display[2]</i>	<i>display[3]</i>	<i>display[4]</i>	<i>display[5]</i>	<i>display[6]</i>	<i>display[7]</i>
V7	U7	V5	U5	V8	U8	W6	W7

Output lightctl			
<i>lightctl[0]</i>	<i>lightctl[1]</i>	<i>lightctl[2]</i>	<i>lightctl[3]</i>
U2	U4	V4	W4

## ✓ Discussion

這一題超級簡單，把上一題的功能拿來改一改就可以了。

## ✓ Conclusion

這一次的lab將聲音與鍵盤合起來，難度不高，但是做起來之後功能就很強大。只要學會了螢幕顯示怎麼製作之後，我相信透過這個lab學到的東西就足以讓我們至做出一個像樣的遊戲。儘管如此，我仍然覺得滑鼠部分沒有學到有點可惜，我試著上網找過open source有沒有製作滑鼠p2s介面的code可惜找不太到。如✓碩製作遊戲的話沒有這部分會有點小小吃虧。

最後是做完這個lab我有點小疑問，不知道聲音輸出的時候有沒有辦法輸出更複雜的聲音，譬如人的聲音，或是其他流行音樂的聲音，很好奇究竟這個到底辦不辦的到。