Lab8: Speaker

107061112 王昊文

Experiment

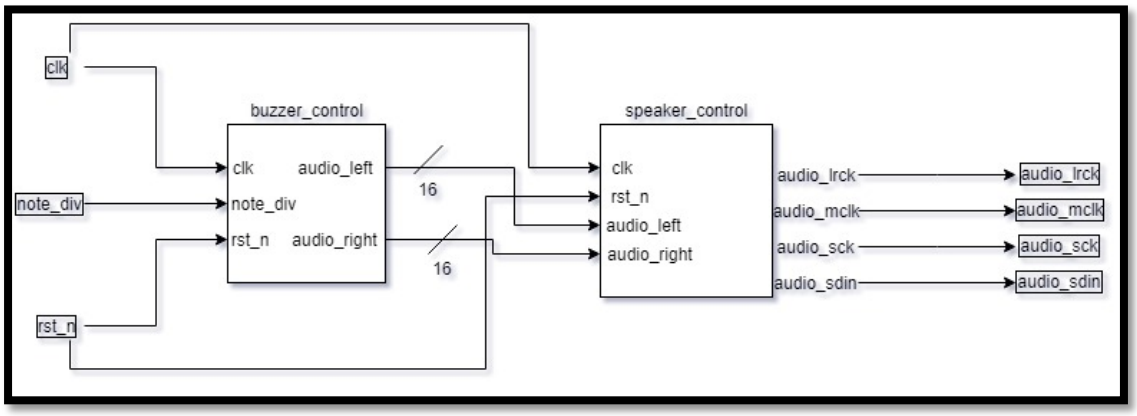
1 Please design an audio-data parallel-to-serial module to generate the speaker control signal with 100MHz system clock, 25 MHz master clock, (25/128) MHz Left-Right clock (Fs), and 6.25 MHz (32Fs) sampling clock.

Design Specification

✓ I/O

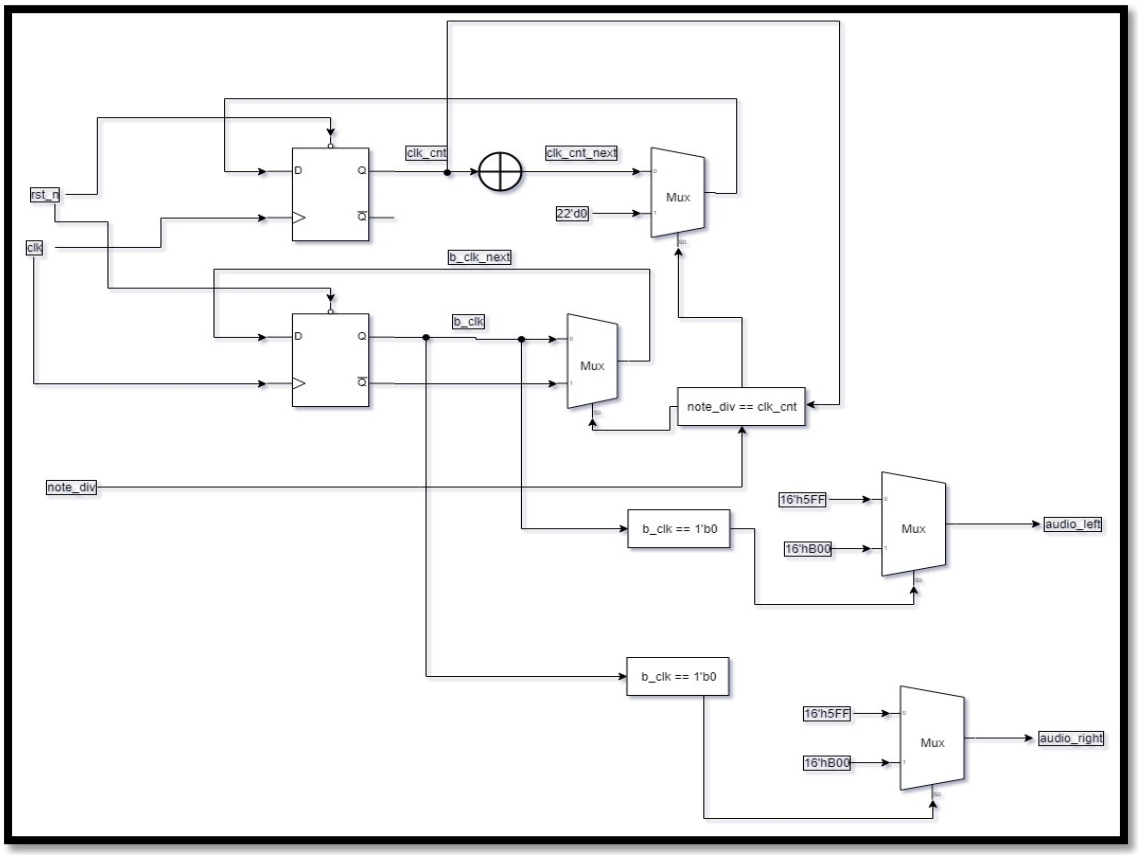
```
Input clk;          // crystal clock
Input rst_n;        // low active reset

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:



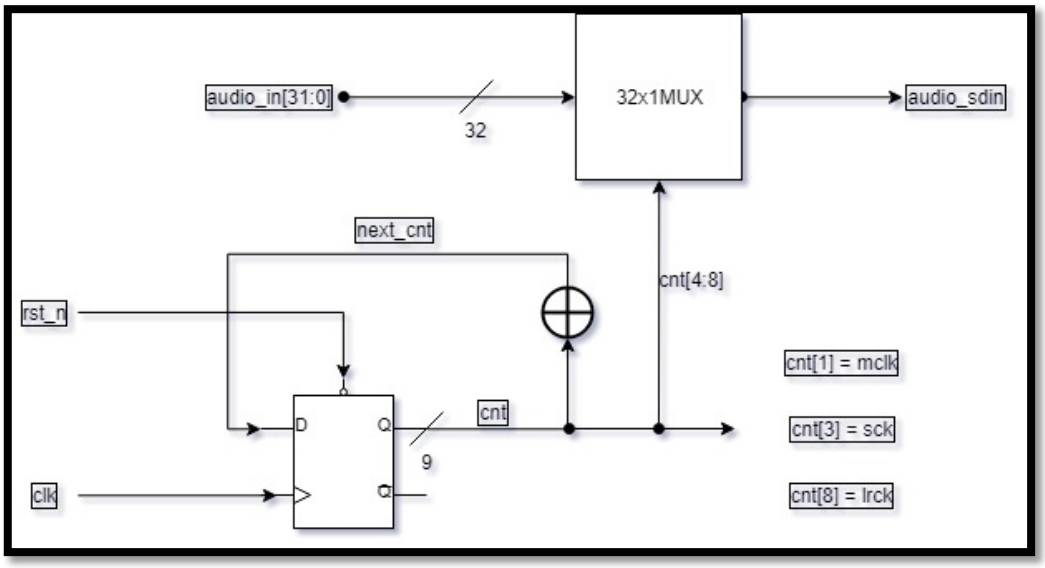
根據使用者輸入的note_div進行buzzer輸出。Buzzer control即為轉換成頻率，然後會依頻率送出震幅訊號給speaker control，以parallel的形式送進去。Speaker_control則為提供speaker需要的頻率，然後將parallel的資料改成serial的形式output資料，為sdin。然後其他三個output皆為speaker需要的頻率。

Buzzer control:



在這個模組之中，clk_cnt扮演上數的角色，跟著crystal clock不斷上數。這時，會判斷clk_cnt是否達到使用者輸入之note_div（音的頻率），每當達到條件，b_clk便會反向一次，b_clk為發出聲音訊號的指標。然後再來判斷b_clk是否為零，若為0則當時為波谷，發出波谷震幅的訊號，反之則為波峰，發出波峰震幅的訊號。輸出的audio_left，audio_right為parallel訊號，送進speaker control會做piso的處理。

Speaker control (clock signals):



這個模組，我利用一個9bit的cnt(接crystal clock)不斷上數，然後抽取其中的bit來當作想要的頻率即可。Master clock為25MHZ，除以4相當於2的2次方，故取第二個bit的變化來當作輸出，其餘的輸出依此類推。

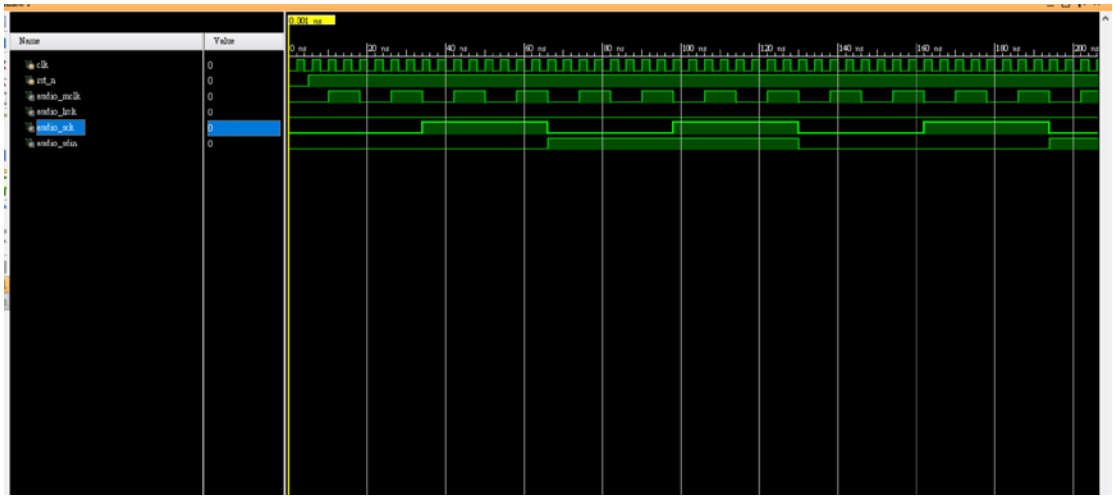
在音訊輸出的部分，由於輸出必須在一個serial clock之內將32個資料全部輸出，然後我們知道left right clock為serial clock快，因此我們可以將cnt[8:4]抽出來當作資料輸出的指標，數到特定數字時輸出必須的資料，利用這樣的方法，我們就可以成功將32個音訊資料在一個serial clock cycle之內將資料全部輸出。

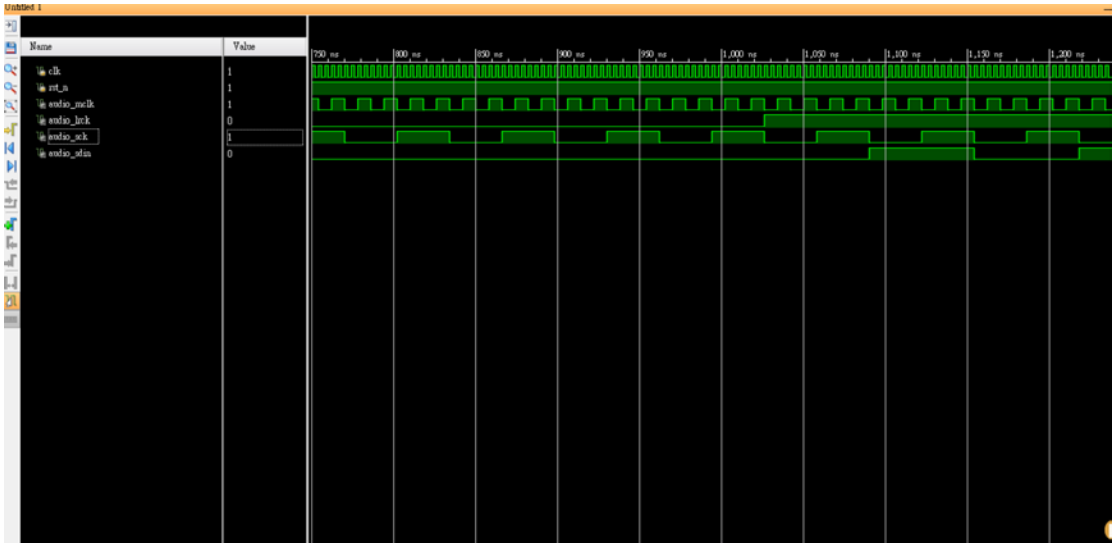
✓ Design Implementation

Input	
clk	*rst_n*
W5	V17

Output			
audio_mclk	*audio_lrck*	*audio_sck*	*audio_sdin*
A14	A16	B15	B16

✓ Verilog Simulation





Sdin比較難從圖形上檢查，但是其他的clock可以看出符合頻率的求。

✓ Discussion

做到這邊總算是松了一口氣，總算不用在做clock方面的東西。其實以整體來說，難度並不高，最重要的是釐清speaker所需要的input以及output是什麼，還有clock的頻率要確定除對。最後要確認sdin在送出的時候要空一個lrck cycle，訊號才會是正確的。

其實這一題在搞懂speaker花了些許時間，瞭解之後很快就做完了。很快的speaker就做出了聲音。這一題大致上沒有遇到什麼特別的問題。

Experiment

2 Speaker control

2.1 Please produce the buzzer sounds of Do, Re, and Mi by pressing buttons (Left, Center, Right) respectively. When you press down the button, the speaker produces corresponding frequency sound. When you release the switch, the speaker stops the sound.

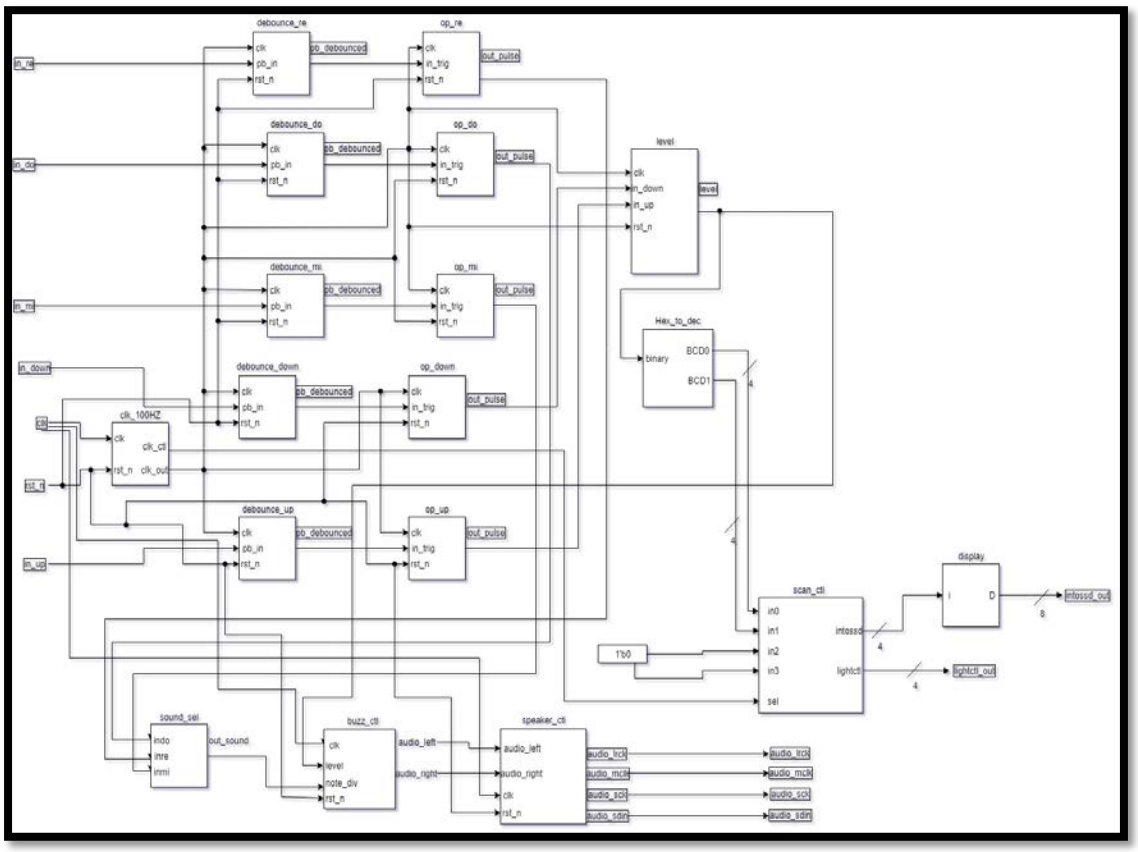
2.2 Please control the volume of the sound by pressing button (Up) as increase and (Down) and decrease the volume. Please also quantize the audio dynamic range as 16 levels and show the current sound level in the 7-segment display.

Design Specification

✓ I/O

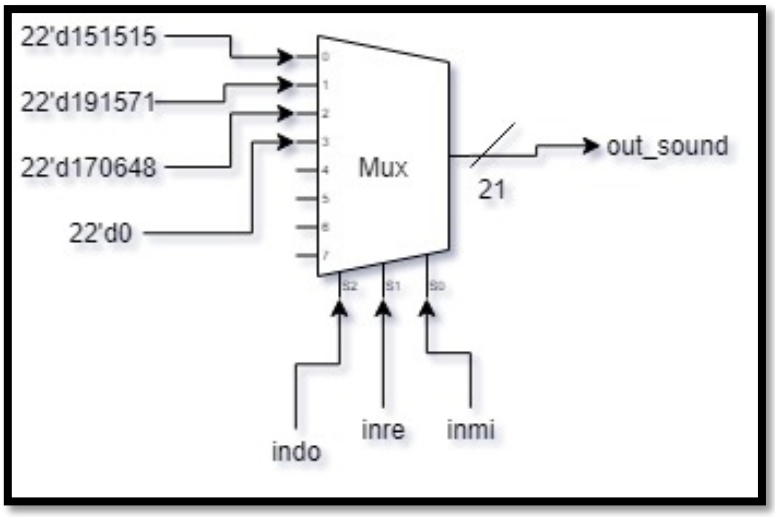
```
Input clk;          // crystal clock
Input rst_n;        // low active reset
Input indo;         // input sound do
Input inre;         // input sound re
Input inmi;         // input sound mi
Input inup;         // input volume up
Input indown;       // input volume down

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
// display outputs
Output [3:0]lightctl_out;
Output [7:0]display;
```

✓ **Logic Block:**



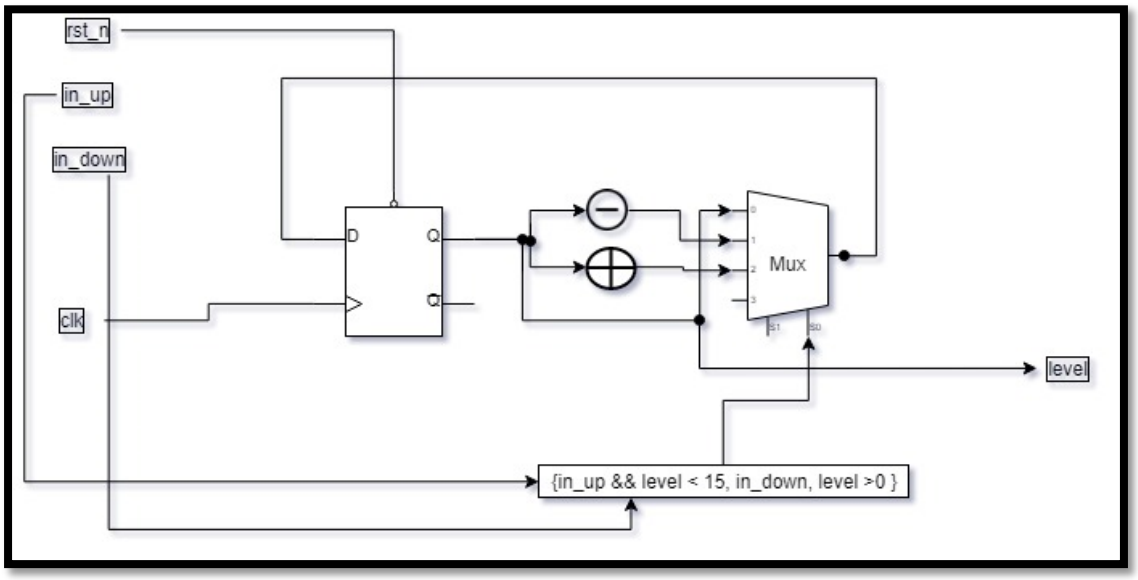
這一題其實跟上一題一樣，只是必須客制化使用者的聲音與音量。因此就必須針對使用者設定的按鍵做一些訊號處理。因此前面有很大一部份是在做one pulse，debounce的處理。其中會有兩個模組可以來選擇想要的音高跟音量。Sound_sel可以根據目前的輸入來選擇音高，level模組則可以選擇音量。Hex to dec則為轉換為人看的懂得十進位制顯示在seven segment display上面。

Sound_sel:



使用MUX來選擇音高

Level：



利用此邏輯來判斷level的值，然後進入speaker control就可以選擇想要的聲音大小。若in up，in_down都沒有進來，level就會維持原本的值。然後根據是in_up，in_down進來來判斷level目前加一或減一。

✓ Design Implementation

Input	
clk	*rst_n*
W5	V17

Input				
indo	*inre*	*inmi*	*inup*	*indown*
W19	U18	T17	T18	U17

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

Output lightctl			
lightctl[0]	*lightctl[1]*	*lightctl[2]*	*lightctl[3]*
U2	U4	V4	W4

Output			
audio_mclk	*audio_lrck*	*audio_sck*	*audio_sdin*
A14	A16	B15	B16

✓ Discussion

由於上一題很快就做完，我還以為這一題難度也不高。後來測試的時候聲音會忽大忽小，甚至有大的雜音。由於第一題做出來聲音沒問題，讓我想了很久原因所在，後來發現我的testbench根本就是錯的，除頻完全除錯。搞了半天修正後才正確。但是我很好奇仍然有些微的雜音存在，通通都除對了，也空了一個cycle，不知道為什麼背景還是有很明顯的雜音存在，助教看了code也都對。不確定是不是板子的問題，還是正常現象.....。

✓ Conclusion

這次的lab總算是接觸到了不同東西，讓我們看見了最簡單的喇叭的原理。但是貌似是指能發出非常電子式的聲音，非常好奇真正市面上的喇叭，以及耳機，究竟是如何發出各種不同的立體聲，如人聲、樂器聲等等不同的音效，因為這次的speaker貌似是指能發出單一種聲音。這讓我想到了期末要做出來的final project，如果想要做出遊戲的話，感覺必須要做出更加多變的音效才可以。究竟有什麼樣的方法可以讓喇叭發出更多不同的聲音呢？？