

Exp1: 放出 16 個頻率的聲音，一個聲音持續一秒鐘。

這題需要用到 Lab08 的聲音 module。我寫了一個新的 module(note.v)用來讀取 frequency divider 的 clk_out，就是一秒鐘的頻率，輸出一個值 note_div，分別為 16 種不同的頻率，用 50000000 除以講義給的頻率表上的頻率，還有兩個[3:0]note1、[3:0]note2，用來輸出 SSD 的音名。將 note_div 輸出至 lab08 之 note_gen.v，接著輸出至 speaker_control.v，再修改 ssd.v 之輸出七段顯示器，當 note1、note2 都為 0 時，分別在第三位及第四位顯示"D"及"o"，即可完成題目之要求。

Exp2:當按下 0~9、A~F 時，輸出 16 個頻率的音符，再顯示至 SSD。

這個題目需要用到 lab09 的鍵盤讀取。因此我將 lab09 之讀取鍵盤的 module(keyboard.v)用至這題，keyboard.v 當讀取的 8 位數值剛好為鍵盤的 0~9、A~F 時，輸出 0 至 15 的值 note，分別為 16 種頻率的音，再傳入上題之 note.v，輸出 note_div 後傳入 note_gen.v、speaker_control.v，最後傳入 ssd.v 輸出在七段顯示器上，但我發現一個問題，我在存取 USB 輸入之值，也就是鍵盤傳入的值時，檢查 even parity 成功後存取 8 位數的值，再判斷這個值代表鍵盤上的哪個按鍵，而並沒有一個停止的機制。我再次研究，當鍵盤按下時會有一個訊號，當鍵盤放開時，有一個快速的"F0"會被讀取，接著又會有一個相同的訊號。因此我在讀取鍵盤數值的地方設計一個 cnt，如圖(二)，當讀取一個新的值時會+1，因此當讀到 F0 時將 cnt 存入，令它要在讀完 F0 後的一個值停止，如圖(三)，直到下一個值的讀取才會輸出 note。State Diagram 如圖(四)。接著又有一個問題，顯示我的 audio_lrck 等等的電壓為 1.8V，與其他的 3.3V 不同，我查了許久，原來是 XDC file 裡所有東西都要設置成 LVCMOS33，才排除了這個問題。其餘就如同 exp1，輸出聲音及 SSD 即可。

```
if(CODEWORD != RELEASE) begin
    if(CODEWORD == ZERO) notes <= 5'd0;
    if(CODEWORD == ONE) notes <= 5'd1;
    if(CODEWORD == TWO) notes <= 5'd2;
    if(CODEWORD == THREE) notes <= 5'd3;
    if(CODEWORD == FOUR) notes <= 5'd4;
    if(CODEWORD == FIVE) notes <= 5'd5;
    if(CODEWORD == SIX) notes <= 5'd6;
    if(CODEWORD == SEVEN) notes <= 5'd7;
    if(CODEWORD == EIGHT) notes <= 5'd8;
    if(CODEWORD == NINE) notes <= 5'd9;
    if(CODEWORD == a) notes <= 5'd10;
    if(CODEWORD == b) notes <= 5'd11;
    if(CODEWORD == c) notes <= 5'd12;
    if(CODEWORD == d) notes <= 5'd13;
    if(CODEWORD == e) notes <= 5'd14;
    if(CODEWORD == f) notes <= 5'd15;
end
```

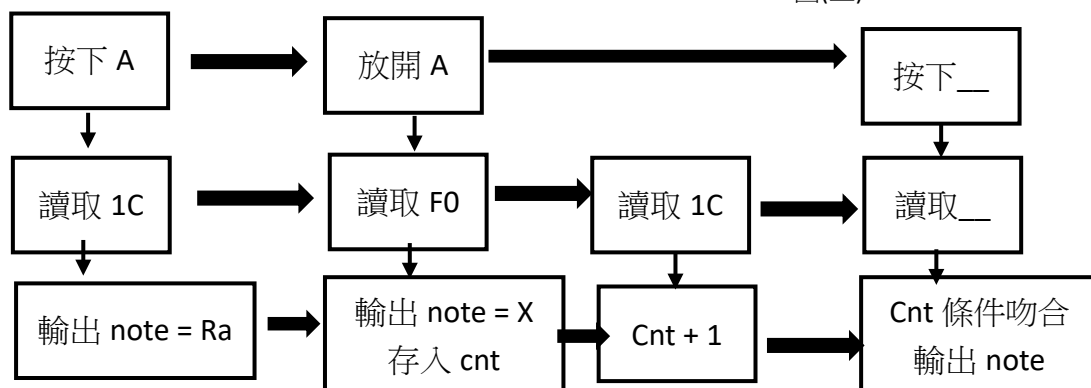
圖(一)

```
cnt <= cnt + 1;
CODEWORD <= scan_code[8:1];
```

圖(二)

```
if(cnt2 == cnt) begin notes <= 5'd16; end
else if (cnt2 > cnt) begin notes <= 5'd16; end
else if (cnt2 != cnt) begin state <= 3'd0; end
```

圖(三)



Bonus: 分成左、右聲道，以一個 I/O switch 控制單音或雙音模式，輸出兩種不同聲音。

這題需要修改 Lab8 的 `note_gen.v`，原本輸出 `audio_right` 及 `audio_left`，現在只需要輸出一個 `audio_out`，呼叫 `note_gen.v` 兩次，將兩個不同頻率的 `note_div` 輸入後，即可輸出兩個不同的 `audio_right` 及 `audio_left`。因此這題需要先在 `keyboard.v` 新增一個 `note2`，我設計的是當按第一排的按鍵會輸出 `note`，為低音部分的 Do~Si，第二排會輸出 `note2`，為高音部分的 Do~Si，輸出後傳入 `note_gen.v` 兩次，分別輸出 `audio_left` 及 `audio_right`，接著在輸入 `speaker_control.v`，即可輸出雙聲道的電子琴。再將單聲道、雙聲道模式加入，只要在 `keyboard.v` 中，加入簡單的 if、else 即可，當 switch 是 on 時，`note` 及 `note2` 會不同，當 switch 是 off 時，`note` 及 `note2` 會相同。



圖(五)

心得：這次作業在製作按下馬上有聲音、放開馬上停止的鍵盤電子琴花非常久時間，因為很難用 `testbench` 看讀取的訊號為多少，我不清楚它是怎麼傳入訊號的，也很難用顯示的方式顯示出按的東西是甚麼，用了許久及測試了許多方法才慢慢摸索出按鍵輸入的運作模式，我認為我跑了 `bitstream` 至少跑了 50 次。雖然真的用了很久，但是自己找出來真的非常開心，也從只能一直發出聲音的電子琴變成可以用按下及放開控制聲音的電子琴。