

Lab09 (109061256 陳立萍)

Lab9_1: Play the sounds with 14 tones repeatedly based on the sound table. Every sound is played for one second..

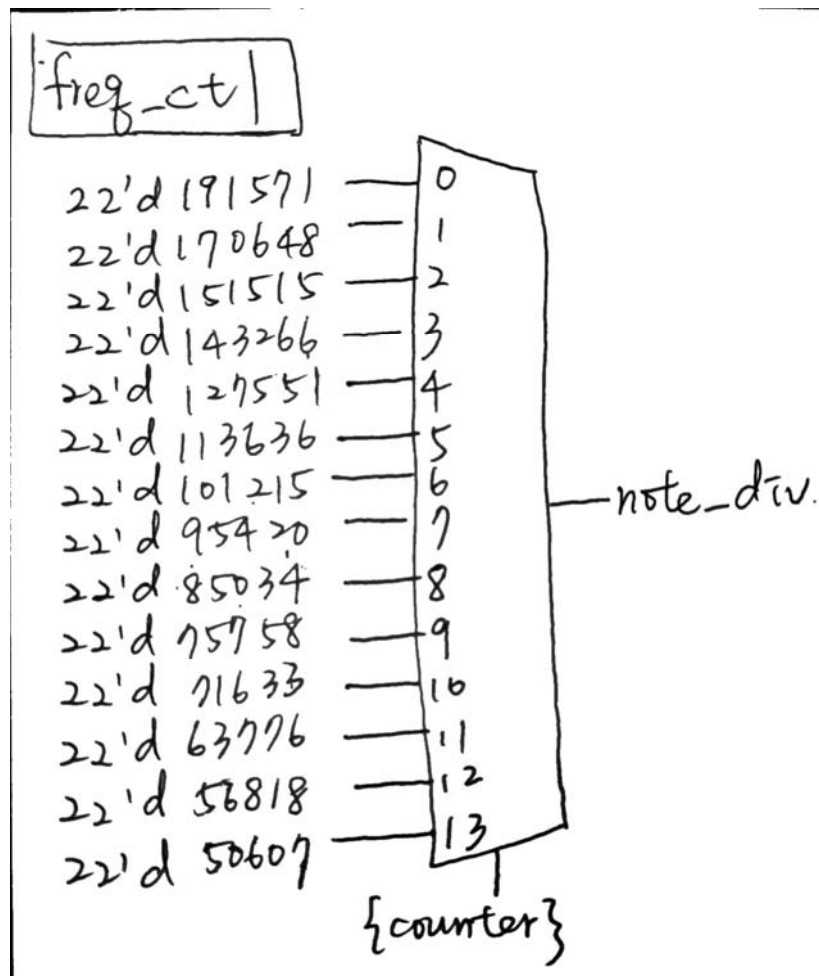
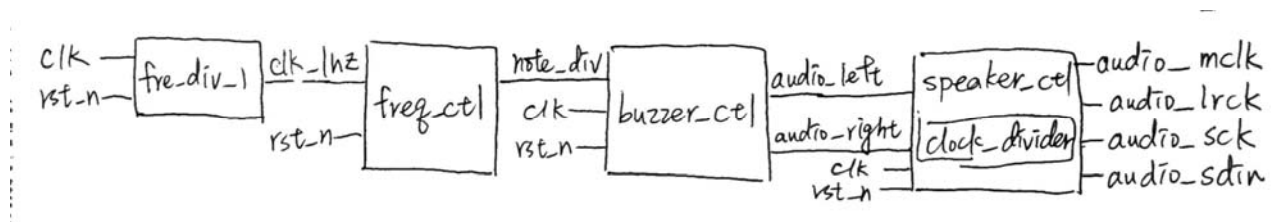
Design Specification

input : clk(100MHz), rst_n(reset)

output : audio_mclk(25MHz), audio_lrck(25/128MHz),

audio_sck(6.25MHz), audio_sdin(serial output)

logic diagram



I/O pin assignment:

audio_mclk	audio_lrck	audio_sck	audio_sdin	clk	rst_n
A14	A16	B15	B16	W5	R2

Design Implementation

設計方法：

這題用到共 5 個 module，其中 `buzzer_ctl.v`：用來製造特定的頻率。`speaker_ctl.v`：用來將 `audio_left` 和 `audio_right` 轉成 1bit 輸出（做 parallel to serial 的處理）。`clock_divider.v`：來除出三種頻率，分別為 25MHz(mclk), 25/128MHz(lrck), 6.25MHz(sck/sampling clock)。上述在 lab7 已經介紹過，可以直接 include 進來使用，另外還會用到 1hz 除頻的 module，因此也將 `fre_div_1.v` include 進來。

這次新增的 module 是

*`freq_ctl.v`：主要是用來循環播放 14 個音階。首先先把 1hz 拉線進來當作 clock，再用一個 4bit 的 counter 不斷循環從 0 數到 13 而這個 counter 會接到多工器作為選擇的判斷，0 的時候 `note_div` 輸出中音 Do，1 輸出中音 Re……13 輸出高音 Si，接著歸零到 0 又開始播放中音 Do，之後 `note_div` 會輸出給 `buzzer_ctl.v` 做特定頻率輸出，達到題目的循環播放要求。

最後用 `lab9_1_top.v` 將所有 module 連接在一起。

Lab9_2: Electronic Organ

2.1 Integrate the PC keyboard 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 sounds (Do, Re, Mi, Fa, So, La, Si) in the 7-segment LED

Design Specification

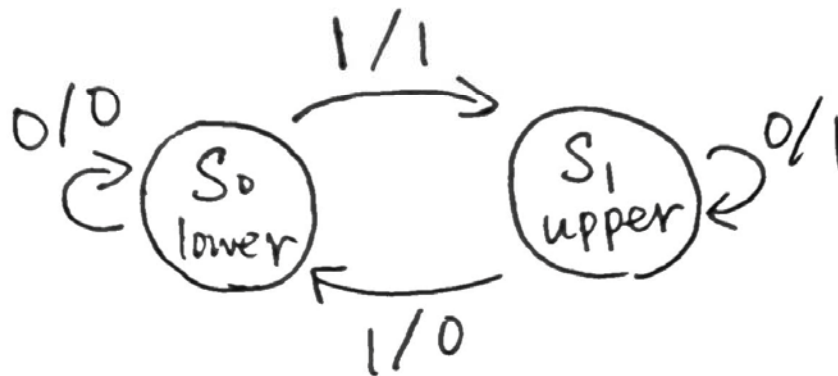
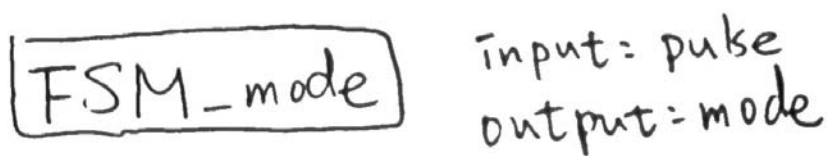
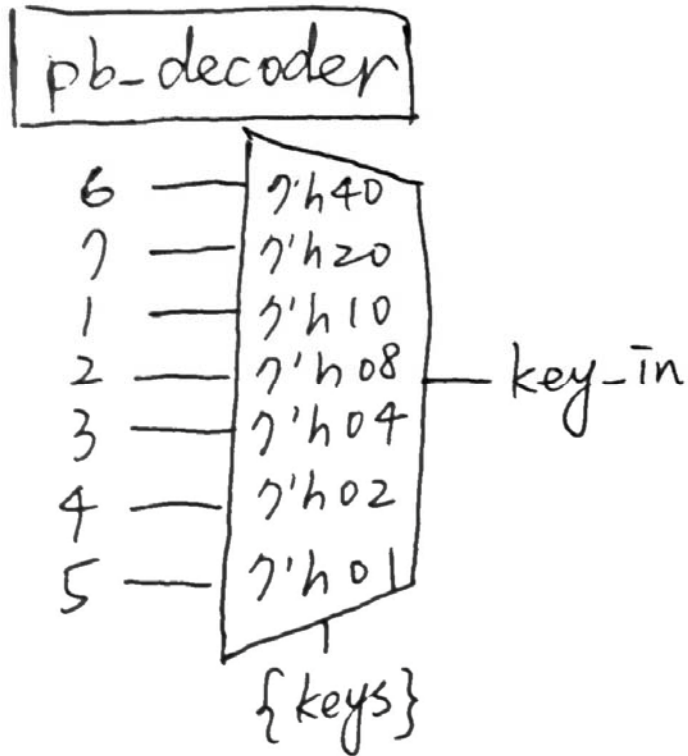
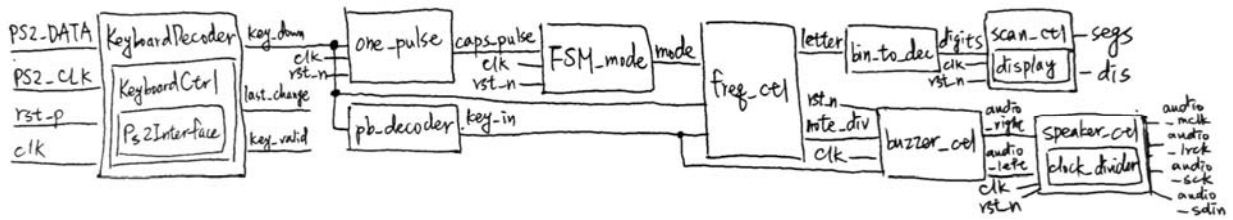
inout : PS2_DATA, PS2_CLK

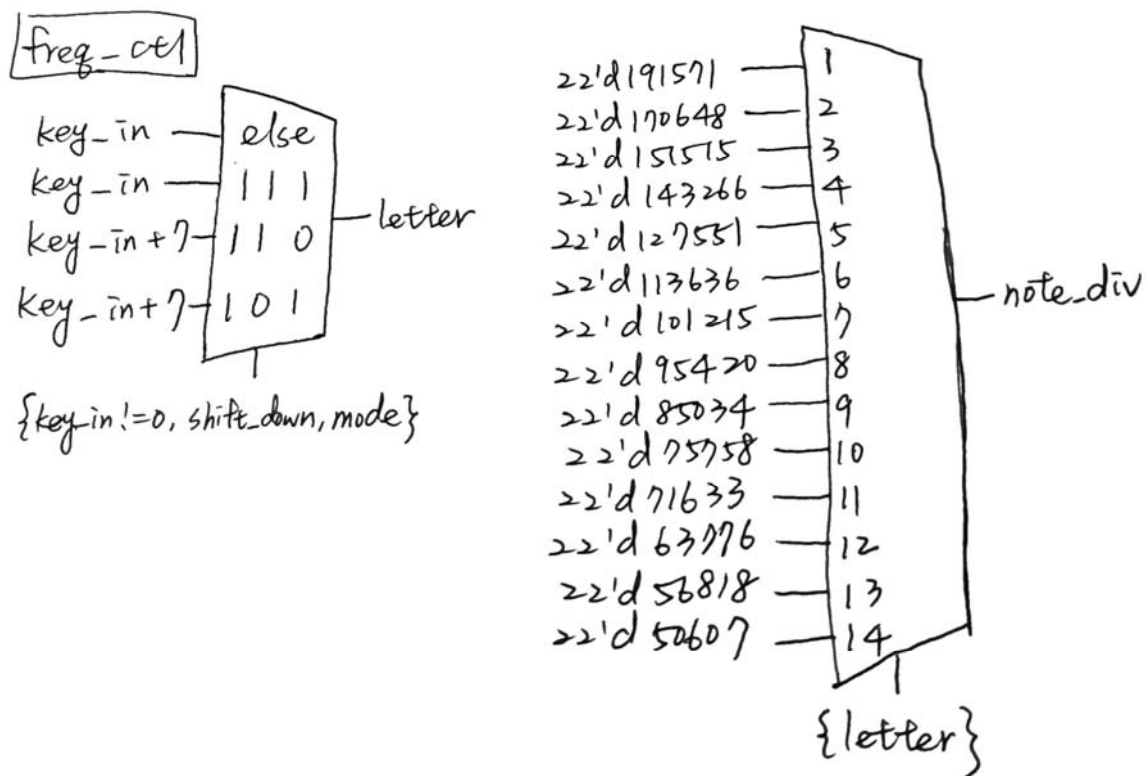
input : clk(100Mhz), rst_n(reset)

output : audio_mclk(25MHz), audio_lrck(25/128MHz),
audio_sck(6.25MHz), audio_sdin(serial output)

[7:0]segs(七段顯示器圖形), [3:0]dis(四個七段顯示器)

logic diagram





I/O pin assignment:

PS2_DATA	PS2_CLK	rst_n	clk
B17	C17	R2	W5

segs[7]	segs[6]	segs[5]	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]
W7	W6	U8	V8	U5	V5	U7	V7

dis[3]	dis[2]	dis[1]	dis[0]
W4	V4	U4	U2

audio_mclk	audio_lrclk	audio_sck	audio_sdin
A14	A16	B15	B16

Design Implementation

設計方法：

這個題目一共用到 14 個 module。其中 KeyboardDecoder.v：鍵盤處理，one_pulse.v：判斷是否重新按下按鍵（caps_lock），產生一個 pulse，buzzer_ctl.v：用來製造特定的頻率，speaker_ctl.v：用來將 audio left 和 audio right 轉成 1bit 輸出（做 parallel to serial 的處

理)，clock_divider.v：來除出三種頻率，分別為 25MHz(mclk), 25/128MHz(lrck), 6.25MHz(sck/sampling clock)，bin_to_dec.v：二進位轉十進位，scan_ctl.v：製造視覺暫留，display.v：七段顯示器輸出，之前已介紹過，不再贅述，此次新增的 module 如下：

- *pb_decoder.v：將鍵盤上 abcdefg 的 key_down 組成 7bit 的組合訊號輸入進來，哪一個 bit 是 1 代表那個字母被按下，依序由多工器輸出 key_in 編碼，c 為 1，d 為 2，e 為 3，f 為 4，g 為 5，a 為 6，b 為 7。
- *FSM_mode：為一個 FSM 狀態機，用來判斷是否切換大小寫。狀態分為大寫（1）與小寫（0）狀態，由 one_pulse 產生的大小寫按鍵 pulse 進來做狀態轉換的觸發。
- *freq_ctl.v：這裡我多做了和 lab8_4 一樣有 shift 的功能，用多工器來做 letter 暫存器的輸出選擇，多工器的選擇由 pb_decoder 的輸出 key_in, shift_down(key_down[18]), FSM_mode 的輸出 mode 來判斷，分為四種狀況，1.key_in 不為 0（有按鍵被按下）且有按下 shift 且 mode 是大寫，則輸出 letter=key_in（低音），2.key_in 不為 0（有按鍵被按下）且有按下 shift 且 mode 是小寫，則輸出 letter=key_in+7（高音），3.有按下按鍵且 mode=1 則輸出 letter=key_in+7（高音），4.剩餘情況皆讓 letter=key_in（低音）。之後這個 letter 暫存器會變成另一組多工器的 note_div 輸出判斷，1 是輸出中音 Do……14 輸出高音 Si。

最後再用 lab9_2_top.v 將上述 module 連接。

3.Lab9_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

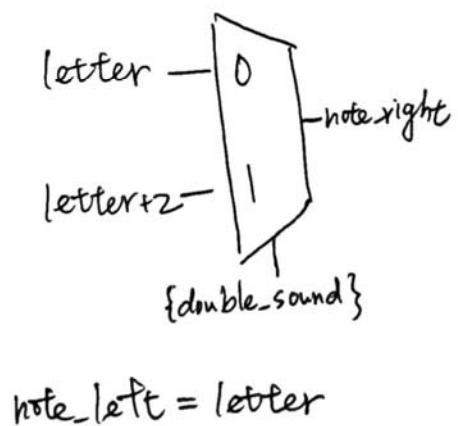
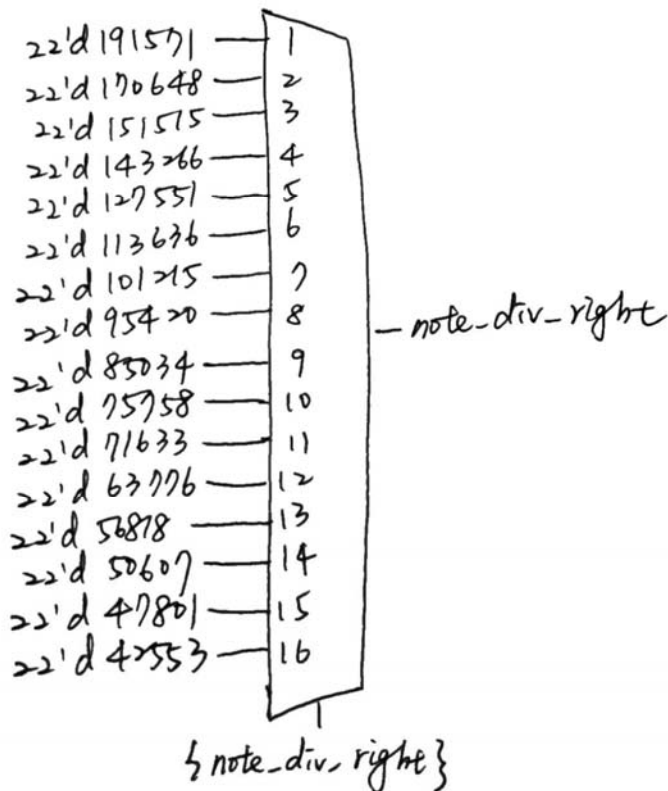
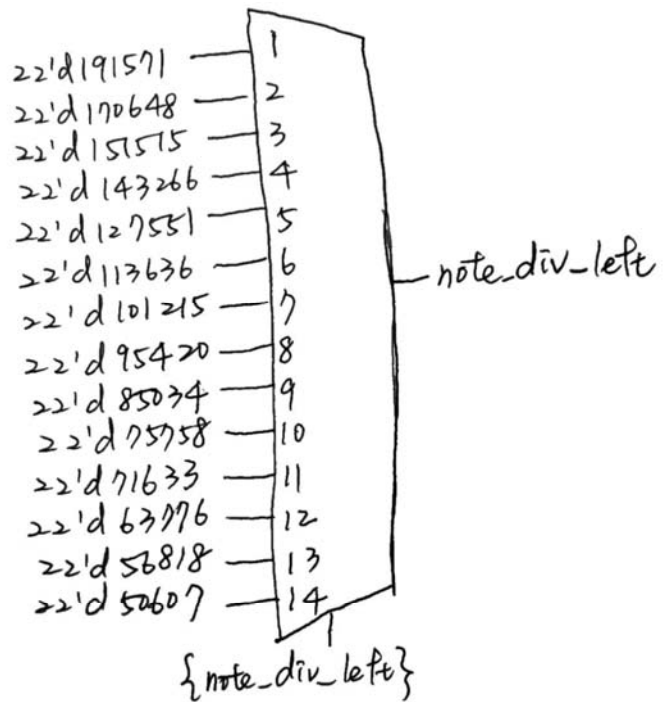
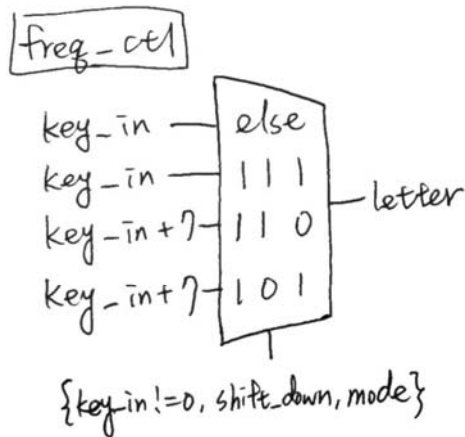
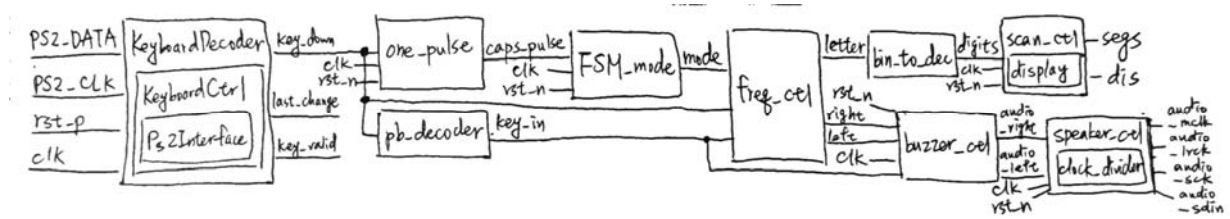
inout : PS2_DATA, PS2_CLK

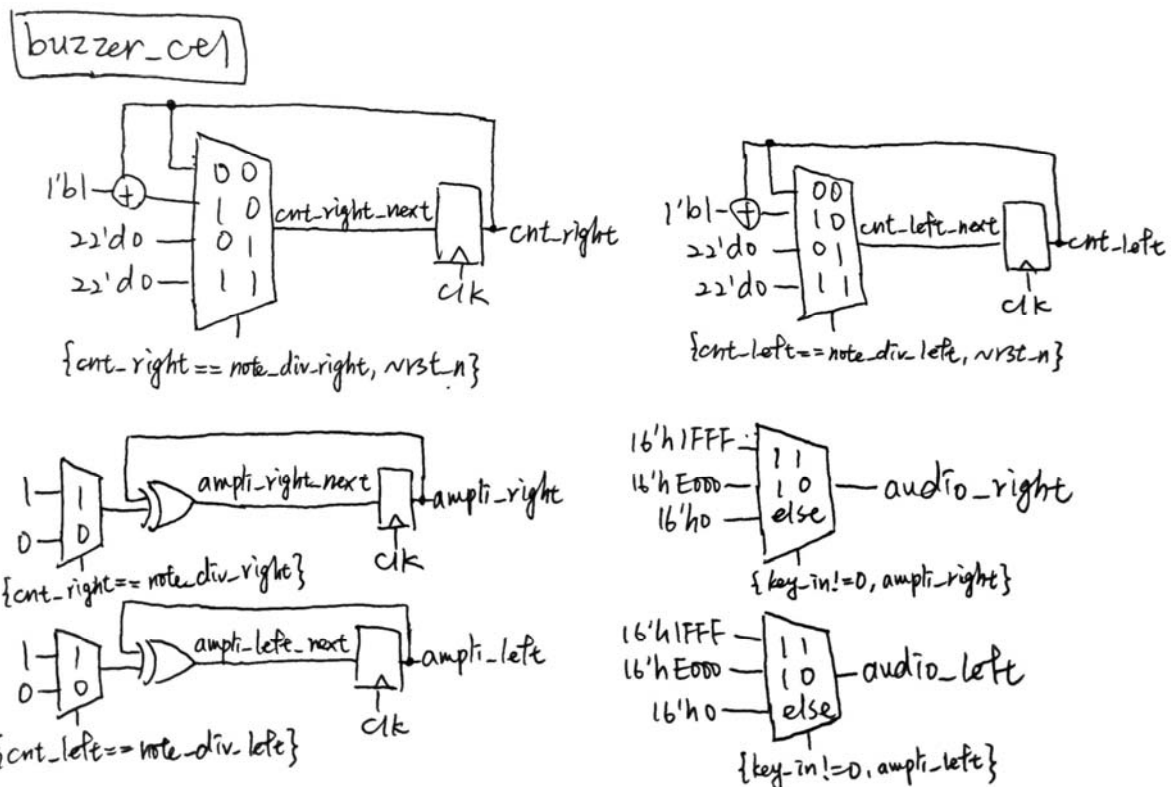
input : clk(100Mhz), rst_n(reset) , double_sound(select one/two sound)

output : audio_mclk(25MHz), audio_lrck(25/128MHz),
audio_sck(6.25MHz), audio_sdin(serial output)

[7:0]segs(七段顯示器圖形), [3:0]dis(四個七段顯示器)

logic diagram





I/O pin assignment:

PS2_DATA	PS2_CLK	rst_n	clk	double_sound
B17	C17	R2	W5	V17

segs[7]	segs[6]	segs[5]	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]
W7	W6	U8	V8	U5	V5	U7	V7

dis[3]	dis[2]	dis[1]	dis[0]
W4	V4	U4	U2

audio_mclk	audio_lrclk	audio_sck	audio_sdin
A14	A16	B15	B16

Design Implementation

設計方法：

這題大致上和第二題一樣，不一樣的地方只有 `buzzer_ctl.v` 與 `freq_ctl.v`。

* `freq_ctl.v`：這裡和第二題不一樣的地方在於多了一個 input: `double_sound` 來控制一個多工器，當 `double_sound` 為 0 的時候代表左右耳同聲，因此右耳暫存器(`note_right`)保持 letter 給的值，當 `double_sound` 為 1 時代表右耳高左耳兩個音階，因此右耳暫存器的值

為此時的 letter 再+2，左耳暫存器(note_left)則保持原本的 letter 值。之後 note_left 和 note_right 會各自控制一組多工器來分別輸出 note_div_right 和 note_div_left，也就是左右耳個別的頻率除數，給 buzzer_ctl.v。

* buzzer_ctl.v：和之前的不一樣在於輸入的除數有兩個，一個是給右耳的 note_div_right 另一個是給左耳的 note_div_left，分別用來除出各自的頻率給 audio_right 和 audio_left 去輸出給 speaker_ctl.v。

最後再用 lab9_3_top.v 整合上述各個 module。

Discussion

Lab9_1

這題其實非常簡單，把 lab7 的東西拿進來用而已。不過一開始我還是遇到一個問題，就是每當在 reset 後開始循環播放，結果 reset 後的第一個 Do, Re 播放非常快，其他時候的 Do, Re 都正常，後來思考後覺得是因為 positive edge 還是 negative edge 觸發的問題，因為我的 1hz 輸出，一開始是從 0 開始，所以如果 freq_ctl 設 posedge，那麼在 reset 後半個 1hz 週期就會遇到正緣，就會觸發讓 Do 指撥放半秒就變 Re。

Lab9_2

這題其實就是把 lab7 和 lab8 組合在一起而已，所以難度不大，不過我還是不小心接錯線，打錯字，更不小心的是宣告的時候忘記給予變數 bit 數目，導致聲音多工器的判斷 key_in 只有 1 個 bit，所以一開始用耳機聽的時候按下不同的字母，都只有 Do(1)和沒聲音(0)。

Lab9_3

這題基本上只有把第二題小小修改，再加上一個 DIP switch input 做判斷，因此沒有太大的問題與難度。

Conclusion

這次的 lab 做的好快，基本上都只是在貼上 lab7 與 lab8 的 module 到這個 lab，再加以修改而已。另外因為做的很快，所以原本第 2、3 題只有要求大小寫的功能，我還自己另外做了 shift 的功能，如此一來在玩電子琴可以更方便的切換音高。學期初的時候根本沒有想到會做出這麼龐大的 lab 與各種功能的結合，這個 lab 真的讓我很有成就感，下一個 VGA 感覺會很難，希望自己可進步更多。

References

教授授課投影片：語法運用，符號運用，設計觀念。