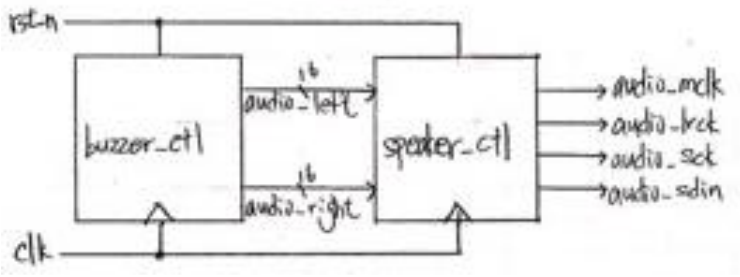
Lab10_1
Design Specification

- ✓ For a speaker:
  Input:
  clk, rst_n
  Output:
  audio_mclk, audio_lrck, audio_sck, audio_sdin
- ✓ Draw the block diagram of the design.



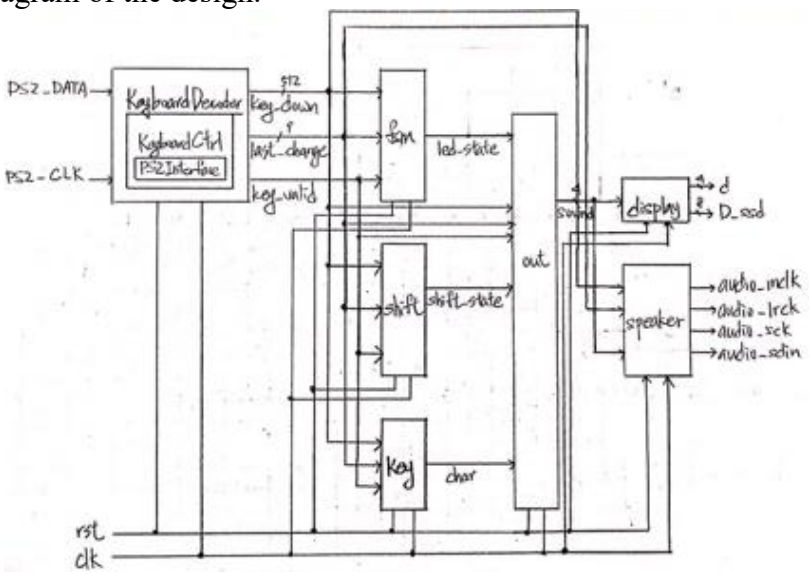
Design Implementation

- ✓ 本題由 buzzer_control 與 speaker_control 組成。
- ✓ 本題與 lab8 概念相同，差別在於是自動發出聲音。因此我在 buzzer_ctl 中加了一個頻率為一秒的 clock，用以控制每秒輸出的聲音頻率。每過一秒後 counter 會加一，並將 note_div 輸入下一個音調的頻率。
- ✓ I/O pin

I/O	audio_mclk	audio_lrck	audio_sck	audio_sdin	clk	rst_n
VOC	A14	A16	B15	B16	W5	V17

Lab10_2
Design Specification

- ✓ For a keyboard speaker:
  Input:
  clk, rst
  Inout: PS2_CLK, PS2_DATA
  Output:
  [3:0]d, [7:0]D_ssd, audio_mclk, audio_lrck, audio_sck, audio_sdin
- ✓ Draw the block diagram of the design.



Design Implementation

- ✓ 本題由 keyboardDecoder、fsm、shift、key、out、display、speaker 組成
- ✓ 其中 keyboardDecoder、fsm、shift、key、out 皆與 lab9_4 設計原理相近，在此不再贅述。
- ✓ Out 為一個 decoder，將 key 輸出的 5 bits 訊號轉成 4 bits，目的是為了方便判定輸出 14 個音中的哪一個音。並透過接收 fsm 與 shift 的 state，判斷是大寫還是小寫(高音或低音)。
- ✓ speaker 接收 keyboard 輸出的 key_down 與 last_change，以確定鍵盤被按下。其頻率的控制為透過 out 輸出的 sound 判斷。
- ✓ display 接收 sound，為一個 decoder，顯示 14 個音的唱名。在我的設計中，我將低音用 7-segment 的點表示，高音則沒有。
- ✓ I/O pin

I/O	audio_mclk	audio_lrck	audio_sck	audio_sdin	clk	rst	PS2_CLK	PS2_DATA
VOC	A14	A16	B15	B16	W5	U18	C17	B17
I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
LOC	W7	W6	U8	V8	U5	V5	U7	V7
I/O	d[3]	d[2]	d[1]	d[0]				
LOC	W4	V4	U4	U2				

Lab10_3

Design Specification

- ✓ For a two-digit decimal adder/subtractor/multiplier:
- ✓ For a key board display:

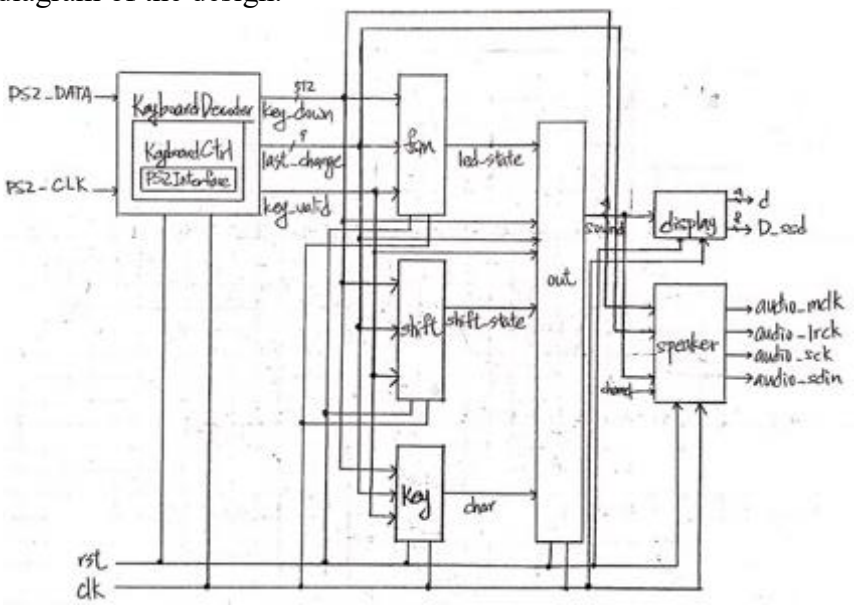
  Input:

  clk, rst, channel

  Inout: PS2_CLK, PS2_DATA

  Output:

  [3:0]d, [7:0]D_ssd, audio_mclk, audio_lrck, audio_sck, audio_sdin
- ✓ Draw the block diagram of the design.



Design Implementation

- ✓ 本題由 keyboardDecoder、fsm、shift、key、out、display、speaker 組成
- ✓ 本題與第二題相似，差別在於第二 channel 為和弦，因此多了一個 channel 的 input，並在 buzzer_ctl 中加了另一個 note_div。
- ✓ 為了製造左右耳輸出不同聲音，因此我在 buzzer_ctl 中加了一組新的控制 note_div 的功能，

將其作為左耳的輸出。在我的設計中，是先判斷 channel 為 0 或 1，0 則是第二題的模式，左耳聲音頻率等於右耳；1 則是和弦模式，左耳聲音頻率高右耳兩個音。

✓ I/O pin

I/O	audio_mclk	audio_lrck	audio_sck	audio_sdin	clk	rst	PS2_CLK	PS2_DATA
VOC	A14	A16	B15	B16	W5	U18	C17	B17
I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
LOC	W7	W6	U8	V8	U5	V5	U7	V7
I/O	d[3]	d[2]	d[1]	d[0]	channel			
LOC	W4	V4	U4	U2	V17			

Discussion

在第一題中，主要的問題是要怎麼控制每秒換一次聲音頻率。我用的方法是用陣列，設變數 i 讓他每隔一秒加一，並在一開始設定陣列內容的值。之後發現其實可以用 counter 去做，其實意思是一樣的。

第二題與第三題與 lab8_2 很類似，差別只在於控制音頻的輸入是用 keyboard。主要是將 lab9_4 與 lab8_2 結合。之所以會用很多個 decoder 是為了在表示上更為清楚。另外須注意輸入的鍵號與聲音的輸出須符合正確要求。

Conclusion

這次利用鍵盤作為輸入，輸出聲音，為前兩個 lab 的結合，沒什麼太大的麻煩，主要是電腦問題，因此一開始無法跑老師給的 keyboardDecoder 程式。

