

## Lab 8

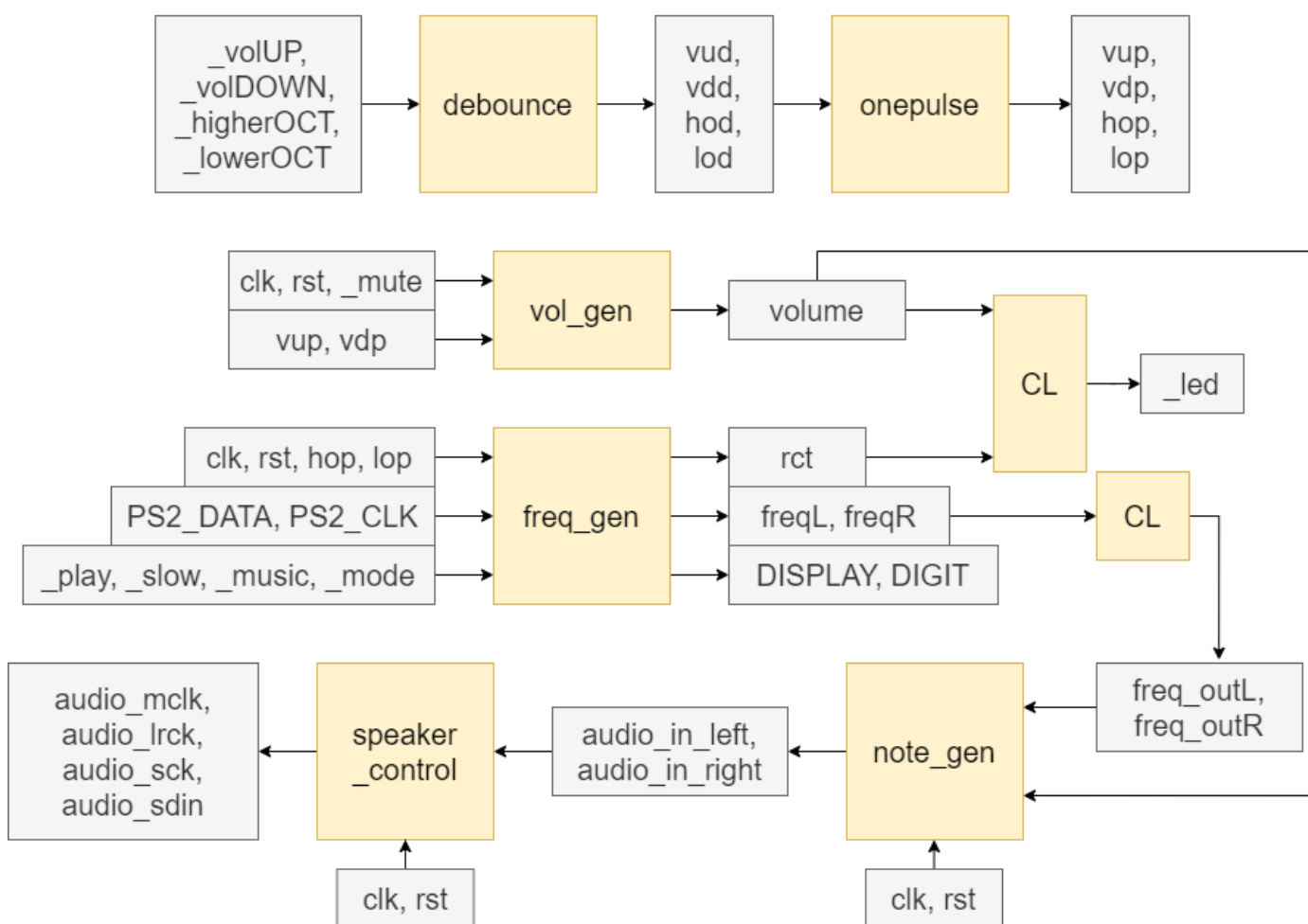
學號: 108032053

姓名: 陳凱揚

### 1. 實作過程

在 module lab8 中，如下圖 1 所示，我首先將 4 個按鍵輸入處理成 onepulse，接著使用自己製作的 module vol\_gen 取得 volume (音量的大小聲)、module freq\_gen 取得 rct (音高) 和 freqL, freqR，並直接輸出已處理好的 DISPLAY 和 DIGIT，而此時的 volume 和 rct 即可決定 led 的輸出。最後再將音訊部分的 freqL、freqR，依照模板的方式通過 module note\_gen 和 module speaker\_control，其中剛剛的 volume 會接至 note\_gen 裡，決定輸出訊號的振幅，振幅越大則音量越大，控制方式如下圖 2 所示 (以 audio\_left 為例)，只要根據 volume 大小輸出不同大小的振幅即可 (注意以二補數表示)。

Block Diagram(lab 8)



▲ 圖 1

```

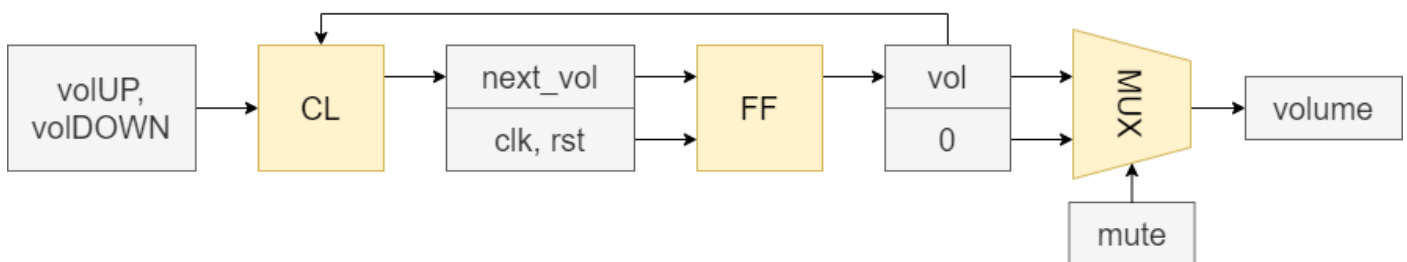
// audio_left
always @* begin
    audio_left = 16'h0000;
    if(note_div_left != 22'd1) begin
        case(volume)
            3'd1: audio_left = (b_clk == 1'b0)?16'hE000:16'h2000;
            3'd2: audio_left = (b_clk == 1'b0)?16'hD000:16'h3000;
            3'd3: audio_left = (b_clk == 1'b0)?16'hC000:16'h4000;
            3'd4: audio_left = (b_clk == 1'b0)?16'hB000:16'h5000;
            3'd5: audio_left = (b_clk == 1'b0)?16'hA000:16'h6000;
        endcase
    end
end
end

```

▲ 圖 2

下圖 3 為 module vol\_gen 的 block diagram，其中有一個 flip-flop 儲存目前的音量大小，可以由按鍵 volUP、volDOWN 來增減，而在最後由 mute 來決定輸出的 volume 是否要靜音。

### Block Diagram(vol\_gen)

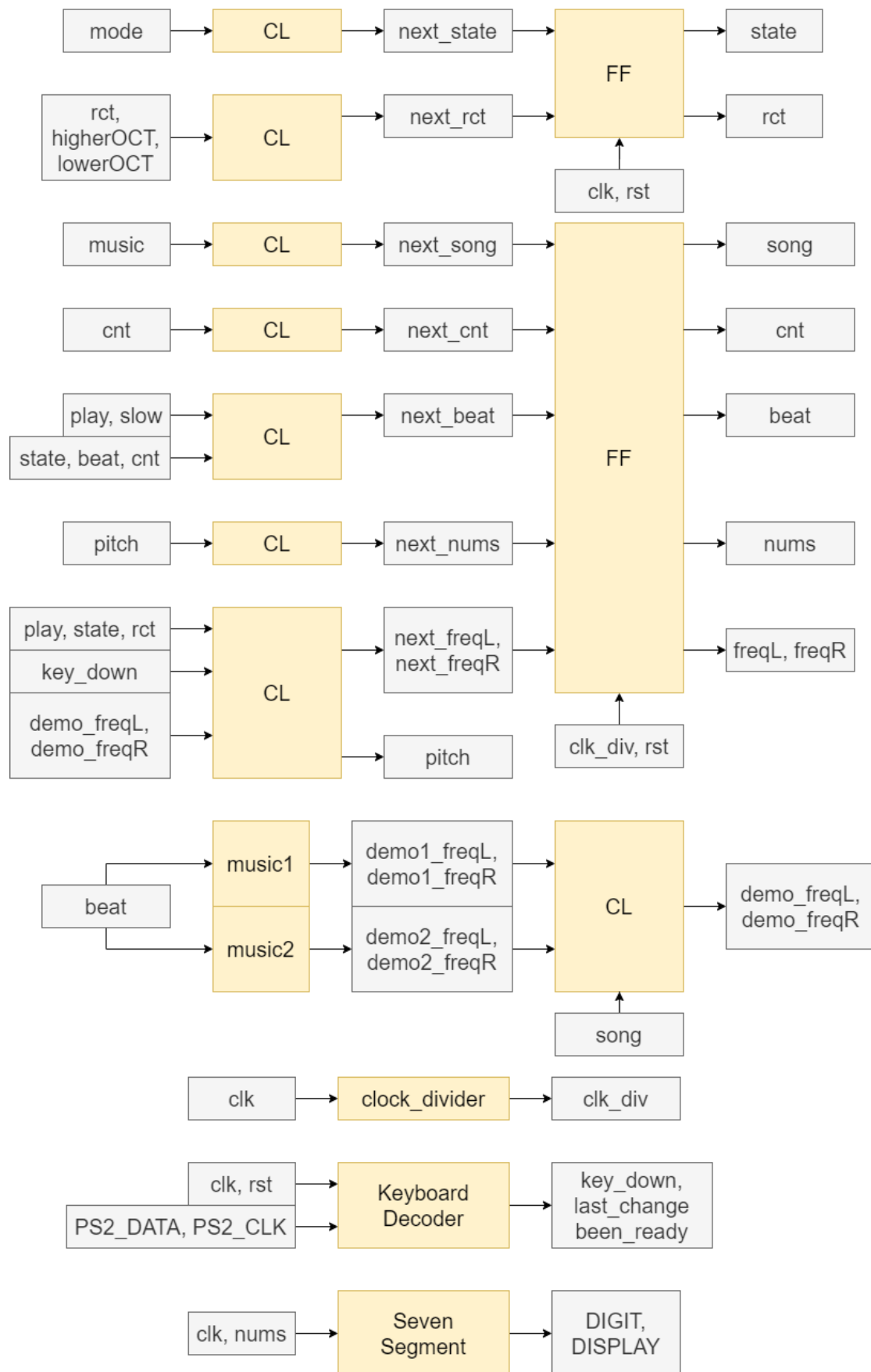


▲ 圖 3

下圖 4 為 module freq\_gen 的 block diagram，首先 clock\_divider、KeyboardDecoder、SevenSegment 等 module 先處理了除頻、鍵盤輸入、數字顯示。接著設計了 8 個 flip-flop，其中 2 個使用原始的 clk synchronize，另外 6 個使用除頻後的 clk\_div，以下分別為其儲存的資訊。此外 pitch 會在取得 freq 時直接以 8bits 記錄要輸出音高的 2 個數字，以便 nums 的輸出；而 beat 會輸入進 module music1, music2 並輸出兩首歌曲對應的頻率，再以 song 選擇是哪一個頻率給 demo\_freq。

- (1) state：只是將以開關輸入的 mode synchronize。
- (2) rct：儲存音高資訊，可由 higherOCT 和 lowerOCT 等按鍵增減音高大小。
- (3) song：只是將以開關輸入的 music synchronize。
- (4) cnt：大小只有 1bits，不停的在 0 與 1 震盪，用於降速歌曲的撥放。
- (5) beat：記錄目前拍子所在。slow = 1 時，每 2 個 cycle 才增加 1 以降速。而當歌曲切換時歸零。
- (6) nums：記錄要輸出的數字，前 2 個數字永遠輸出“-”，後 2 個數字直接輸出 pitch。
- (7) freqL、freqR：在 PLAY 時根據 keydown 輸出對應的頻率，在 DEMO 時則會輸出 demo\_freq，並根據 rct 調整音高。

## Block Diagram(freq\_gen)



▲ 圖 4

## 2. 學到的東西與遇到的困難

我覺得這次的作業概念上不難，但在實作上會遇到不少小問題，尤其在歌曲撥放時，因為需要輸入大量的音訊資料，很容易不小心產生小 bug，像是相同音高的音符之間的小靜音。而且因為這次要處理的 input、output 很多，一時之間會不知道從何下手，所以我覺得 module 的劃分顯得格外重要，而我一開始就將 freq 和 volume 採用不同的 module 產生，很快地就降低了 module lab8 的複雜度，也讓難度降低很多，可以只專注在設計當前較小的 module 中，設計速度也因此快上不少。

## 3. 想對老師或助教說的話

記得當年剛進公司不久，還不夠熟悉整個程式架構，所以做第一個專案的時候東卡西卡做到一半，我發現有一個被 import 的 module 好像都沒有被用到，想說要找當初寫這段 code 的人問，就找了一個前輩問說這是誰寫的  
結果那個前輩似乎不巧剛跟老闆吵完一架心情不太好，就不耐煩地反問我：「它有影響到什麼嗎？沒有的話南無觀世音菩薩很重要嗎？」  
我被嚇到什麼話也不敢說，就默默回座 debug，腦海一直在想他那句是在代替髒話還是什麼  
  
我現在回想起來，才突然發現他當時應該是說  
「那麼管誰 import 啥很重要嗎？」

發文傳送門 <https://init.engineer>

純靠北工程師