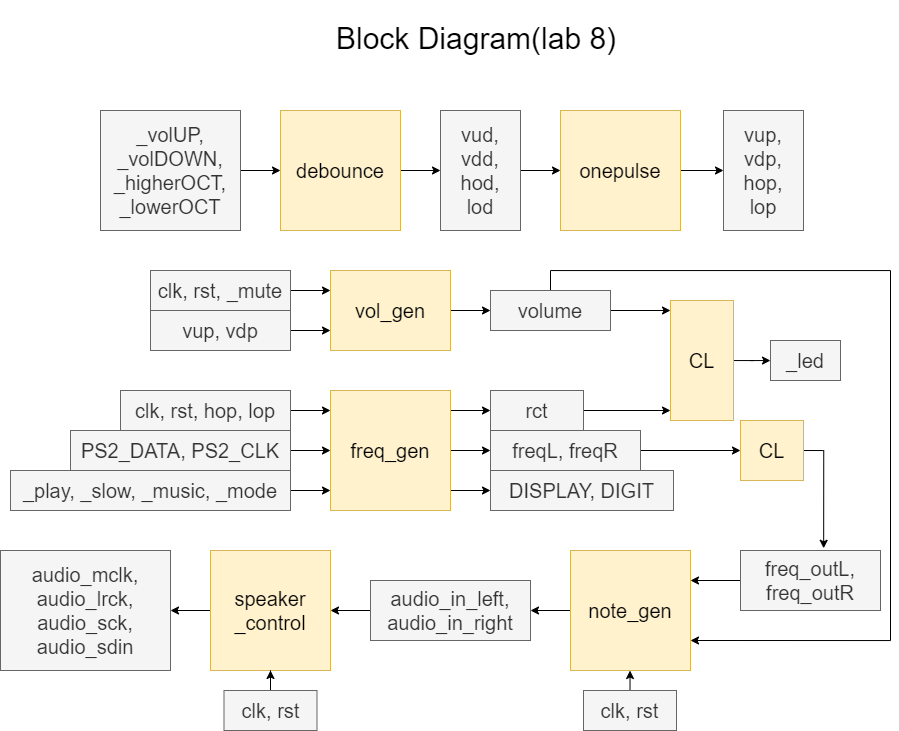
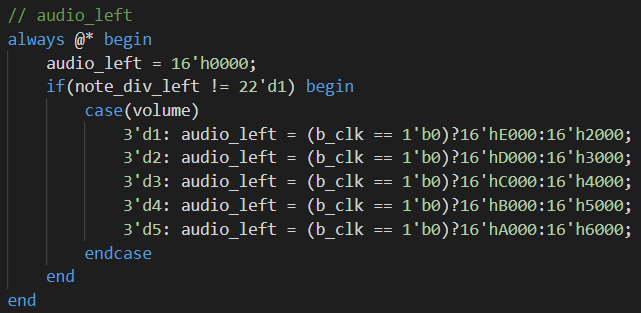
|  |  |
| --- | --- |
| **Lab 8** | |
| 學號: 108032053 | 姓名: 陳凱揚 |

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）。

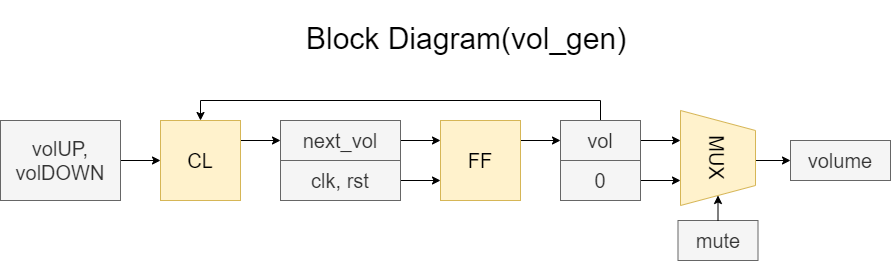


▲ 圖1



▲ 圖2

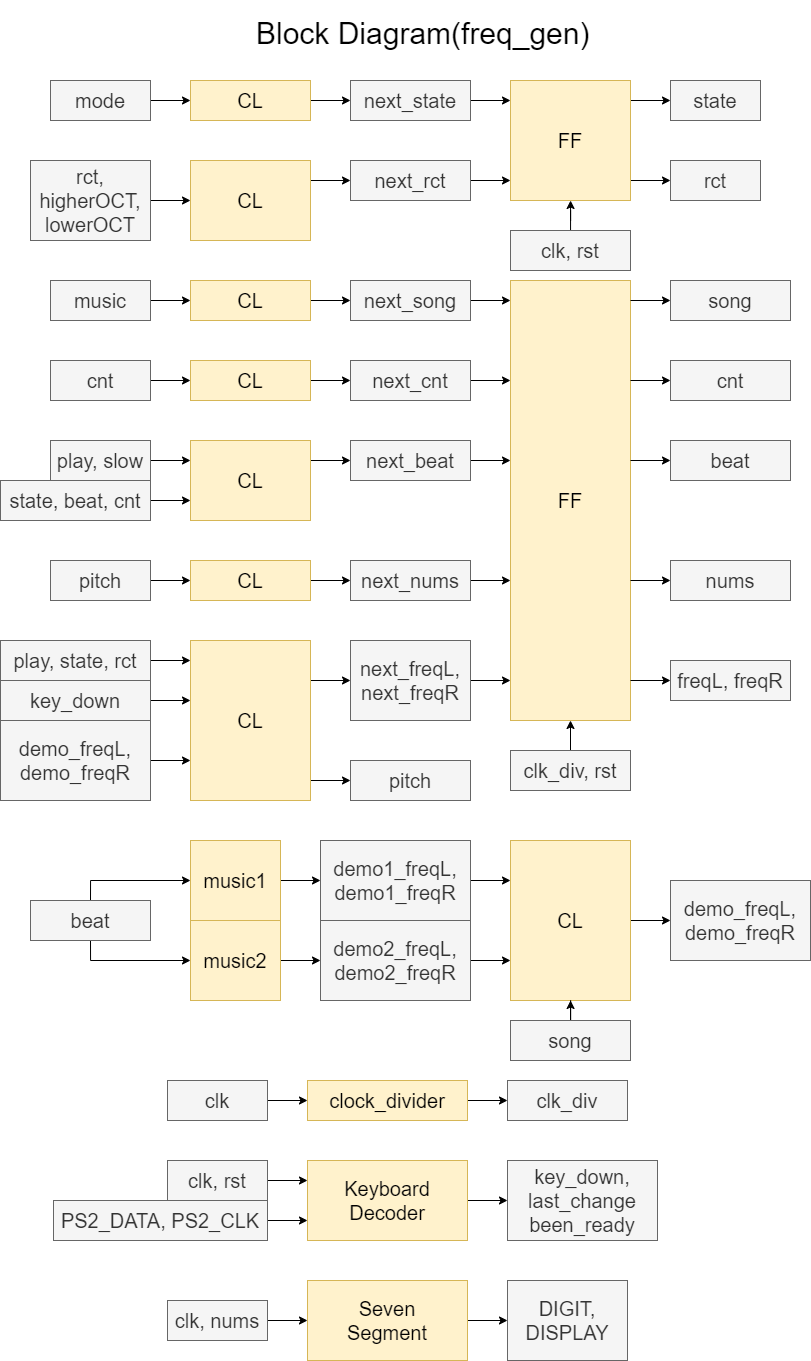
下圖3為module vol\_gen的block diagram，其中有一個flip-flop儲存目前的音量大小，可以由按鍵volUP、volDOWN來增減，而在最後由mute來決定輸出的volume是否要靜音。



▲ 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、frqeR：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。



▲ 圖4

1. 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。

1. 想對老師或助教說的話

