

EECS 2070 02 Digital Design Labs 2019
Lab 8
學號：107062314 姓名：陳柏均

0. 前言

FPGA 板能做的事情真的很多，這次 lab8 已經到可以自己做出音樂的感覺，我個人很喜歡聽音樂，因此在做的時候也很享受在其中。雖說自己真的很愛聽音樂，但其實僅僅限制在這聽的局限內。

這次 lab8 我認為對我的挑戰性真的很高，這是由於個人是個音痴，完全看不懂任何跟五線譜有關的東西，含音符、音節及各種符號，總之只要是在五線譜上的東西我剛開始是一律看不懂的。但基於對聽音樂的興趣，我最後仍是試著克服它並完成這次 lab。

1. 實作過程

(i) 稍微理解 Sample code 及做基本功能

看完題目以及上課的一些教材後，我先開始做的東西是音量，助教給的 code 是固定的一個音量，同時也有附上靜音；而題目所要求的是除了靜音外要分五個 level，而我的做法是將音量的數字大小先試著平均成五個部份，使其能聽出差別。

而這個地方需要用到振幅的概念，有分正的地方跟負的地方，我是先決定正的大小為多少，然後再用工程計算機將其轉成十進位的數字加負號最後再轉回十六進位的數以得到負數的部份。

0620(HEX) → 1568(DEC)
negative
-1568(DEC) → F9E0(HEX)

(註:處理音量振福的部分將十六進位跟十進位互相轉換)

至於音量顯示於 led 的方式只要判斷目前是哪個 level 並且將需要給的值給 led 便可。Play 與 pause 的功能我是將_play 當做 en 傳入 module 中，使其達到_play 會播放上面的旋律，而 pause 則停留在原音符並且保持靜音。這些基本功能都做完後，我就開始進入這次的主題，也是最繁雜的部分，那就是音樂創作。

(ii)理解五線譜將其轉換成程式

上面提過我對五線譜真的是一竅不通，因此我會比大部分人多了這一個理解的部分，這個部分我主要是看上課的講義以及詢問同學們很多關於譜的知識，其實一直到這次的 lab，我才算是真的了解五線譜是如何呈現的。

講義的幾分音符那個地方幫助我甚大，我剛開始幾乎每次都得看那個對照表以彌補我這方面知識嚴重不足。題目要求是兩首歌，我選的是助教在 lab8 下面附上的告白氣球，以及自己找的李夢尹的「雨下的瞬間」。為了自己方便我皆是選擇用九個音節。

音 符	簡 譜 (以 Do 為例)	休 止 符	簡 譜	附 點 音 符	簡 譜 (以 Do 為例)	附 點 休 止 符	簡 譜
全音符	1 - - -	全休止符	0 0 0 0	附點全音符	1 - - - 1 -	附點全休止符	0 0 0 0 0
二分音符	1 -	二分休止符	0 0	附點二分音符	1 - -	附點二分休止符	0 0 0
四分音符	1	四分休止符	0	附點四分音符	1 .	附點四分休止符	0 .
八分音符	<u>1</u>	八分休止符	<u>0</u>	附點八分音符	<u>1</u> .	附點八分休止符	<u>0</u> .
十六分音符	<u>1</u>	十六分休止符	<u>0</u>	附點十六分音符	<u>1</u> .	附點十六分休止符	<u>0</u> .

(註:在講義上幫助我非常大大的一張對照表)

選完音樂之後呢，我還要先將五線譜轉換為我看得懂的簡譜，並依照五線譜上的音節(幾分音符)去算一個音符到底需要持續多久，而在這個地方我再次體會到程式帶給人類的方便性，我打了個簡單的 c++ 程式碼，讓使用者輸入音符和其持續時間便可以印出一長串轉換後的程式碼。

不得不說，這個經驗使我知道有時候程式真的可以省下我們許多時間，尤其是當要做重複的事情時。順帶一提，我所謂的輸入音符以我個人的程式來講，我是以 1~7 代表中央 Do 到 Si，而 -7~-1 代表著高一階的那些旋律；而另外還有一些半音我則是用其他英文字母代替以獲得輸入時的方便性。

```

15
Please enter the note:-2
Please enter the time:
16
Please enter the note:-1
Please enter the time:
8
Please enter the note:-1
Please enter the time:
32
Please enter the note:0
Please enter the time:
32
Please enter the note:100
print
12'd0: toneL = `sil; 12'd1: toneL = `sil; 12'd2: toneL = `sil; 12'd3: toneL = `sil;
12'd4: toneL = `sil; 12'd5: toneL = `sil; 12'd6: toneL = `sil; 12'd7: toneL = `sil;
12'd8: toneL = `sil; 12'd9: toneL = `sil; 12'd10: toneL = `sil; 12'd11: toneL = `sil;
12'd12: toneL = `sil; 12'd13: toneL = `sil; 12'd14: toneL = `sil; 12'd15: toneL = `sil;

12'd16: toneL = `sil; 12'd17: toneL = `sil; 12'd18: toneL = `sil; 12'd19: toneL = `sil;
12'd20: toneL = `sil; 12'd21: toneL = `sil; 12'd22: toneL = `sil; 12'd23: toneL = `sil;
12'd24: toneL = `sil; 12'd25: toneL = `sil; 12'd26: toneL = `sil; 12'd27: toneL = `sil;
12'd28: toneL = `sil; 12'd29: toneL = `sil; 12'd30: toneL = `sil; 12'd31: toneL = `sil;

12'd32: toneL = `sil; 12'd33: toneL = `sil; 12'd34: toneL = `sil; 12'd35: toneL = `sil;
12'd36: toneL = `sil; 12'd37: toneL = `sil; 12'd38: toneL = `sil; 12'd39: toneL = `sil;
12'd40: toneL = `g; 12'd41: toneL = `g; 12'd42: toneL = `g; 12'd43: toneL = `g;
12'd44: toneL = `g; 12'd45: toneL = `g; 12'd46: toneL = `g; 12'd47: toneL = `g;

12'd48: toneL = `hf; 12'd49: toneL = `hf; 12'd50: toneL = `hf; 12'd51: toneL = `hf;
12'd52: toneL = `hf; 12'd53: toneL = `hf; 12'd54: toneL = `hf; 12'd55: toneL = `hf;
12'd56: toneL = `he; 12'd57: toneL = `he; 12'd58: toneL = `he; 12'd59: toneL = `he;

```

(註:輸入音符及需要維持的時間長度以得到一長串需要用到的 code)

(iii)將兩個音樂進行轉換及 seven segment 的顯示

最後需要做的事則是能將兩個音樂進行轉換以及將 cdefgab 這些音符顯示在七段顯示器上面, 我認為到最後這就沒有到太難, 因為這些觀念以及打法在之前都大致學過。

判斷兩個音樂的部分我是在 music example 的 module 之中以 _music 為判斷依據, 當使用者要播放其中一首時, 則將在上步驟所轉換後的音符貼上即可, 若要播放另一首歌則同理; 而其他地方都保持 sil(靜音)的狀態。

而至於 seven segments 得顯示部分, 由於在前面有 define 過, 我直接將此值照之前學七段顯示器的方式呈現出來, 主要差別大概是這次的英文字母顯示應該會比較多一點, 相對這次而言, 之前的 lab 大致上都以數字為主, 這或許是少數不一樣的地方。

```

`hc:begin display=7'b0100111; end
`hd:begin display=7'b0100001; end
`he:begin display=7'b0000110; end
`hf:begin display=7'b0001110; end
`hg:begin display=7'b1000010; end
`ha:begin display=7'b0100000; end
`hb:begin display=7'b0000011; end

`c:begin display=7'b0100111; end
`d:begin display=7'b0100001; end
`e:begin display=7'b0000110; end
`f:begin display=7'b0001110; end
`g:begin display=7'b1000010; end
`a:begin display=7'b0100000; end
`b:begin display=7'b0000011; end

`l:begin display=7'b0001110; end
`k:begin display=7'b1000010; end
`m:begin display=7'b0000011; end
`n:begin display=7'b0000011; end
`dash:begin display=7'b0111111; end
default: begin display=7'b1111111; end

```

(註:seven segments 顯示音高)

2. 學到的東西與遇到的困難

(i) 音節的分割

由於同音的時候容易出現連音的狀況，畢竟 FPGA 板無法像樂器一樣那麼能分辨出節拍的差別，因此當他遇到同音時它也判斷不出來究竟是幾個音節。而這樣播放出來的音樂在需要分拍的地方它卻連起來聽起來好像有點怪。

而依照助教 lab8 的指示，只要在一個音跟一個音之間加上一個 sil，便可以將音節的效果呈現出來。如下面是我選的的二首歌「雨下的瞬間」的一部分譜，如下圖所圈地方，若是同音但是卻非連的音節，我就會將前面那一個音節的最尾端變成 sil，以造成有點停頓的感覺。



(註:雨下的瞬間音高重複的部分)

```
12'd40: toneR = `m; 12'd41: toneR = `m; 12'd42: toneR = `m; 12'd43: toneR = `m;
12'd44: toneR = `m; 12'd45: toneR = `m; 12'd46: toneR = `m; 12'd47: toneR = `sil;
12'd48: toneR = `m; 12'd49: toneR = `m; 12'd50: toneR = `m; 12'd51: toneR = `m;
12'd52: toneR = `m; 12'd53: toneR = `m; 12'd54: toneR = `m; 12'd55: toneR = `m;
```

(註:將第一個音的最後一個變為 sil)

(ii)左聲道及右聲道

之前聽音樂的時候有發現到有些歌它好像用耳機播放的時候會故意分成左聲道及右聲道, 而這次 lab 明顯就有用到這一功能, 我認為這個功能真的大大增加了音樂的豐富性。

右聲道主要是播放主旋律, 而左聲道是播放伴奏, 再聽老師講解這次的 lab 之前, 一直沒想過還可以分成左聲道跟右聲道去處理, 我相信是由於 FPGA 板要在一個聲道同時播放多個音應該是有一定的困難性。 雖說有了伴奏大概等於要多寫許多音符, 但我認為這是非常值得的。

若左聲道及右聲道皆只播放主旋律, 便會顯得單調而乏味, 而當加上伴奏的時候, 整個感覺就完全不一樣了。 音樂明顯豐富非常多, 也讓聽者更能體會此首音樂的美妙。

而能有主旋律的和伴奏同時存在大概便是右聲道跟左聲道的功勞，這次我學到了其大致的道理，更理解到了平常在聽耳機時為何有時候會出現左右耳所播放出來的聲音不一樣的狀況了。

(iii)音頻及音高輸入的問題

這次我還有學到並非給的 FPGA 板什麼頻率，它便會呈現出那個音，由上課的講義上可以知道，是需要經過一些轉換才能得到我們真正想要的那一個音。而由助教所給的 code 便可以稍稍理解其中的道理，並不是值皆使用我給的音頻，而是有改變過的。

```
assign freq_outL = 50000000 / (_mute ? `silence : freqL);  
assign freq_outR = 50000000 / (_mute ? `silence : freqR);
```

(註:由這段 code 可以明顯看出 FPGA 板所接收的並非我們一般認知的音頻)

另外，當 c++輸入音高的時候，由於我的 note 輸入型態是數字的狀態，而結果卻是要轉成 char 的形式以呈現 cdefgab 等英文字母以利之後用到，剛開始想了一小段時間要如何達到這樣的效果，由於 c~g 是連續的，我就運用 ASCII 的差值去將它從一個數字變回需要輸出的英文字母，而 a 及 b 我則無另外想方法，直皆使用 if else 去判斷並印出。

```
char music[1000];  
else (註:存音符的型態是 char)  
{  
    cout<<"`h"<<(char) (music[i]+50);  
}
```

(註:強制型別轉換轉回 char 的型態並印出)



(註:運用 ASCII 進行轉換)

3.想對老師或助教說的話

這次的 lab 雖說表面上來看應該是對我最難的部分，的確在理解音樂的部分真的花的比我打程式的時間還要多，但是愈做到後來，愈是發現其中的樂趣。綜合了許多之前所學，運用在這次 lab，以前所犯的錯誤也不易再犯，我想老師及助教出那麼多次 lab 的目的就是要我們熟悉再熟悉，除了每一次有新知識新觀念外，更是用到許多之前所用到的東西。

這是最後一個 lab，我認為在處理音樂的部分應該是它比之前 lab 繁雜的原因，但我同時也享受在其中，畢竟其實包括音樂(五線譜)的部分對我來說幾乎也都是新知識，在過程中也遇到許多困難點(還是以音樂居多當然也有 code 的部分)，但也一一克服了他們，最終終於創造出兩首在 FPGA 板上能播放出來的歌了。

多次的 lab 使我犧牲了很多睡眠，有好多次也因為除不了的錯而覺得煩躁，但是我相信這些過程都有它一定的意義的，而學習不就是如此？沒有愈到挫折，一帆風順的學習如何稱作學習，我在這學期從 verilog 中所學到的東西比我想像中的還多很多，每次 lab 打出來都是一種特別的成就，這次也不例外。

相信未來還有許多困難及挑戰需要面對，也許未來的難度就不能以現在 lab 來做相比了，但也是因為邏輯設計實驗安排給我們的 lab，才能讓我們更加進步，讓這些以在腦海中的基礎還有很多經驗運用在未來的領域之中。

最後，我這次運用的一個五線譜網站
<https://musescore.com/user/29866117/scores/5366537> 我覺得真的很棒，可以分音節去播放該首歌，由於我是看不懂五線譜之人，因此它真的給我很大的幫助。

笑話時間:

小王在 10 樓人事部門工作，一個月前，被調到 9 樓行政部門去了。今

天，小王同學打電話到人事部門找他：「小王在麼？」接電話同事說：

「小王已不在人事了。」

小王同學：「啊啊?!什麼時候的事啊，我怎麼不知道啊，還沒來得及送

他呢?」「沒關係，你可以去下面找他啊.....」