|  |  |
| --- | --- |
| **Lab 7** | |
| 學號: 109021115 | 姓名: 吳嘉濬 |

1. **Lab Implementation**

**1. Block Diagram of the design with explanation (10%)**

一張含有 文字, 圖表, 方案, 圖解 的圖片

自動產生的描述

player\_control決定下一個要發出的聲音(note)的編號(ibeatNum)，這depend on現在在哪個mode，如果是demonstrate mode，音樂會自動播放下去並循環播放；如果是play mode – piano和play mode - helper，則不自動發出聲音，只有在玩家按下對應鍵盤按鍵時才會發出聲音；而在play mode - helper，歌曲不循環播放且每次進入這個mode時，歌曲都會重頭開始顯示(於7-seg)。

music module在input ibeatNum以及對應的mode當中output出toneL, toneR還有score, seg(used in helper mode)，其中toneL, toneR代表欲發出聲音之左右聲道頻率。

利用toneR(in demonstrate mode and piano mode)以及seg(in helper mode)，對應顯示七段顯示器右邊2個digit；score(in helper mode)對應顯示左邊2個digit。

經過C/L運算：

一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述

處理好的資料成為module note\_gen的input，再搭配input volume訊號，可以output出雙聲道的pos/neg edge frequency(決定音頻, do re mi …)和振幅(決定音量)。

再經過相對底層的module speaker\_control做進一步的處理，最終成功產生想要發出的聲音。

1. **Partial code screenshot with the explanation (35%)**
2. **Volume control**

在note\_gen module底下，利用input \_mute和volume便足以處理volume control：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

`sil被定義為500000，經過C/L處理(可參考最上面code segment截圖)之後note\_div會等於1，所以audio\_left/right會被assign成0；非`sil的note音量大小取決於case(vo) statement下的判斷，volume從1到5，彼此量值成等比數列(2倍)。如果\_mute==1’b1，assign vo = 3’d0。在case(vo) statement當中讓max和min皆等於0以達到靜音效果。

1. **DEMONSTATE mode**

以下反白區域是module player control在demonstrate mode下會使用到的code segment：

**一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

自動產生的描述**

如果\_play==1’b1，音樂會持續播放，如果到底了i.e. ibeat1 + 1 < LEN不成立時，讓next\_ibeat1 = 0，達到循環播放的效果。

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

ibeat1的值在module music底下被存於ibeatNum，對應出欲發出聲音之頻率。

之後再經過note\_gen和speaker\_control的處理，以完成demonstrate mode的實作。

1. **PLAY mode – Piano**

在play mode - piano當中，不會主動發出聲音，對應到下面code segment倒數第4行：

**一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

自動產生的描述**

我讓按鍵盤發出聲音的機制設計在module music，如下圖：

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

按下按鍵output出對應的toneR, toneL值以發出正確的聲音。

1. **PLAY mode – Helper**

先前兩個mode的需求允許我可以讓toneR的值和7-seg要顯示的畫面綁在一起，但是在helper mode當中七段顯示器的值和發出的聲音不再同步了，所以我必須另外設計reg去各自實作出相對應的功能。詳細作法我把它擺在C part: problem encounter那邊。

至於我如何在demonstrate mode和helper mode來回切換時還能儲存demonstrate mode的"進度”，我是利用兩個不同的ibeat去各自存取/更新進度，如下圖：

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

ibeat1存demonstrate mode的ibeat；ibeat0存play mode - helper的ibeat。

helper mode在led燈上提示node位置的機制，設計於以下code：

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

利用seg的值去判斷led需要顯示的位置。(seg類似於toneR，詳情請看part C)

七段顯示器右邊兩個digit的顯示方式也是依據seg的值，如下圖：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

1. **Score mechanism of the Helper**

判斷得分的機制如下圖：

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

這一段code在module music底下。如果toneR==seg，也就是speaker發出的聲音(note)等於當下ibeatNum對應出的preset note的值時，並且不是`sil note，則判斷為得分。

七段顯示器左邊兩個digit顯示helper mode時的得分狀況，其機制如下：

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

我在module music中output score值，並在top module中做運算，讓最左邊的7-seg digit顯示十位數的分數(in base 10)，第二個7-seg digit顯示個位數的分數(in base 10)。

1. **Questions and Discussions (40%)**

A. If we want to provide two songs in the DEMONSTRATE mode by using an additional switch to select

the song, how would you modify your design of Lab 7?在module music當中新增input select (module music是output toneL, toneR的module)，利用if(select)判斷式去branch出兩大類(兩首不同曲子)，之後再根據ibeatNum去得到對應的toneL, toneR。如果考慮歌曲播放當中會來回切換歌曲，並希望曲子可以在斷掉的地方繼續播放下去，那麼就必須再新增input [11:0] ibeatNum2，個別記錄兩個曲子播放的進度以決定下一個要output的toneL, toneR。

B. What happens when we change

next\_ibeat = (ibeat + 1 < LEN) ? (ibeat + 1) : LEN-1;

to

next\_ibeat = (ibeat + 2 < LEN) ? (ibeat + 2) : LEN-1;

?

在clk頻率不變的情況下，相當於ibeat每經過一個clock cycle便增加2，整個ibeat array被traverse的速度變成原本2倍，也就是整首歌的速度會變成原本的2倍。需要注意的是，在這樣的情況下ibeat會一直保持偶數(或奇數)，所以如果遇到連續但分開的同音頻聲音，原本1/16拍的`sil必須改成連續兩個`sil以避免被跳過，造成聲音連在一起的情況。

1. **Problem Encountered (10%)**

實作demonstrate mode和play mode - piano時，我利用module music output的toneL和toneR去判斷7-seg要顯示的對應畫面，如下圖：

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

在top module當中對應toneL, toneR的wire signal是freqL, freqR，我利用freqR的值，也就是melody segment(主旋律)的note去顯示對應的7-seg。

但是這個方法不適用於設計play mode - helper，因為freqR的值一旦被設為非`sil，其對應的note就會發出相對應的聲音，所以我在helper模式當中必須使用別的變數(signal)去決定7-seg要顯示的值，而非前兩個模式所使用的freqR。

所以我另外在module music當中設新的reg [13:0] seg，他被assign的值和toneR是**一模一樣**的，但是seg卻不會直接讓speaker發出聲音(seg只影響7-seg的顯示，但toneR i.e. freqR會直接讓聲音發出)，所以我在top module利用一樣的方式去決定7-seg要顯示的畫面，如下圖：

一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述

如此一來，便解決發出的聲音和對應7-seg顯示的值被互相綁住的困境。

1. **Suggestions (5%)**

感謝教授以及助教提供我實作lab時的協助。