|  |
| --- |
| **EECS 2070 02 Digital Design Labs 2019**  **Lab 8** |
| **學號：104021215 姓名：熊磊** |

1. 實作過程

* 先利用clock\_divider做出想要的頻率，然後將input中的volumeUP, volumeDOWN, rst做debounce及one pulse的處理。
* 音量是用volume這個register來儲存。調整volume的方式是用always block來處理，再送到note\_gen這個module來處理。音量總共分5個level，選擇上沒有特別設計，單純用聽的來決定音量大小。ampH和ampL互為2’s complement。

A screenshot of text

Description automatically generated A close up of text on a screen

Description automatically generated

* 靜音則是在送進note\_gent前，就先判斷\_mute是否開啟，走得根據\_music把對應的音樂signal送到note\_gen產生。

A picture containing object, clock

Description automatically generated

* 使用player\_control來控制音樂進度、循環、播放、暫停，\_repeat在player\_control用來判斷當播放進度結束時(ibeat>= LEN)，是否要重新播放。若是，則將ibeat歸零，否則設為12’d511。\_play用來控制next\_ibeat是否要+1。

A picture containing object

Description automatically generated

* 而為了方便產生對應的樂譜頻率，我額外寫了一個python程式來把簡譜轉成verilog signal。

A screenshot of a cell phone

Description automatically generated

* 兩首音樂實現的方式是，直接寫兩個module，然後在送進note\_gen實再決定要拿哪一個signal。

A picture containing object

Description automatically generated

* 另外用FSM來控制音樂狀態。如果再從music0切換到music1，player\_control要做reset，因此分了3個states。

A screenshot of text

Description automatically generated

* 七段顯示器就依照目前\_music的狀態來顯示右聲道toneR播放的音符。

A screenshot of a computer

Description automatically generated A close up of text on a black background

Description automatically generated

1. 學到的東西與遇到的困難

這次lab一開始沒想到要用finite state machine，所以花了很多力氣在想要怎麼換歌的時候重新播放，後來決定直接用FSM寫。在產生音樂訊號的地方因為太長了，果斷寫一個python來產生XD。

1. 想對老師或助教說的話

謝謝老師與助教!