**數位電路實驗Lab3 Report**

第四組 B02901093吳岳 B02901097張哲銘 B02901098宇德原

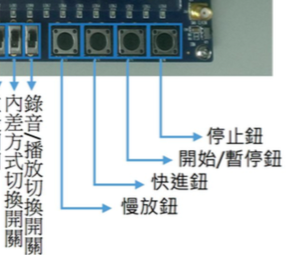
1. 摘要

這次的實驗目的在於用Verilog程式語言去實作出一台數位錄音機，並用DSP的概念去處理聲音訊號。實驗步驟是先從認識錄音機的基本功能開始，規劃不同模組去處理錄音機的各個功能，再設計finite state machine來完成。透過過程中的邏輯思維以及數位電路的開發流程，藉此培養我們的思考與硬體設計能力。

1. 系統架構

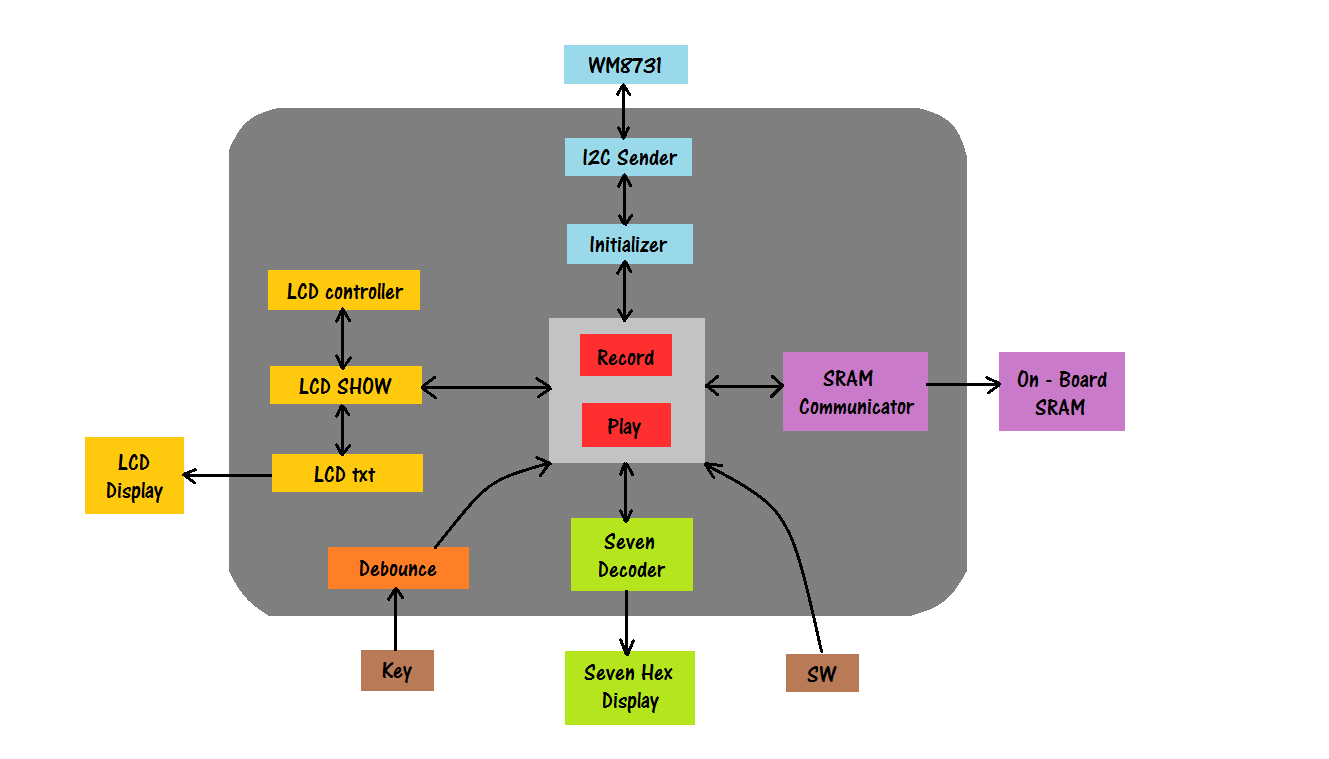
* DE2\_115控制界面

以下是我們設定的控制界面, SW[0]是做錄音/播放的切換, SW[1]是設定內差方式(0次/1次), Key[0]是Stop/ Reset, Key[1]在record/play module 個別是設定 play & pause的切換, Key[2]/[3] 是快進跟慢放。

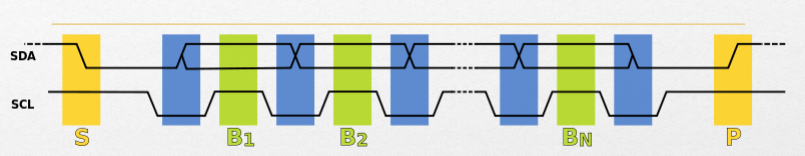


* 內部Module

1. 7 Display & LCD display
2. Initializer : 內含I2C
3. SRAM Communicator
4. SW, KEY 介面
5. Audio Codec
6. Core : 包含最重要的兩個處理模組: Play & Record



1. 各模組的功能
   1. I2C Sender + Initializer



在使用WM8731之前必須先透過I2C protocol傳遞初始化的資料才能正常使用並啟動它，因此我們寫了兩個模組，一個是Initializer，也就是負責傳送初始化的資料給I2C sender，比如: Left Line in, Right Headphone Out, Digital path control…等等，而I2C sender模組就負責產生SCL這個特殊的clock訊號(也就是2個clock的low + 1個clock的high)，並且在SCL是high的時候將Initializer傳來的訊號交給WM8731，然後每傳8個bit的資訊後就要把控制權交給WM8731並且接受ACK一次，直到把24個bit傳完後，就回到原始的state，然後等到把所有的資料傳完後，就結束這兩個模組的運作。

* 1. SRAM Communicator

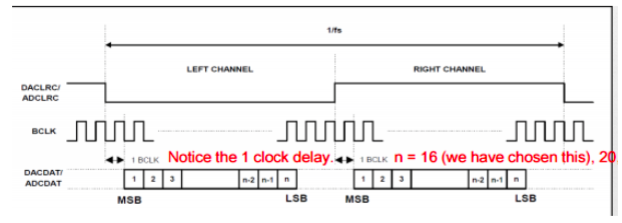
這部分就是負責接受來自Record以及Play的指示來將資料交給DE2-115板子上面SRAM的媒介，如果是Record的模組在運作時，此模組就會把得到的二補數資料傳進SRAM裡面，並且紀錄現在最後一筆資料在SRAM裡面的address，而狀態是在Play的話，就會根據使用者希望調成幾倍速播放來決定讀出來的address是需要一次跳幾碼，然後把SRAM根據這個address我輸出的資料傳給Play這個模組來作播放的動作。



* 1. Play

Play 在I2S的做法跟Record差不多,只是從ADC 轉成DAC, 並從Sram\_Commnicator讀資料。只是這邊比較複雜的是速度的處理, 加速的話比較簡單, 就是跳address,而減速的話分成零次內插跟一次內插。零次內插指的是插入的訊號在兩筆訊號間都直接選擇上一筆的資料來播放，而一次訊號是指在兩筆訊號間插入的訊號跟原始訊號呈線性關係。

這邊我們設定兩個特別的參數: speed\_down\_wait\_count跟sram\_data\_prev, 如果設定完play\_speed, play module會計算出對應的speed\_down\_wait\_count,每經過一個LRC, speed\_down\_wait\_count會減一。藉此設定線性內差的係數。此外, sram\_data\_prev的設定由play left/right state的prev\_start決定, 當左聲道轉換成右聲道(posedge LRC)時, prev\_start會重新設定。



* 1. Record

將錄的資料藉由Codec的模組傳到我們設定好的record module裡面, 透過Key的值設定input做state轉換,因為record在錄資料時只需左右聲道取其一, 而這邊我們選ADC\_LRCK高點時錄資料, 每次錄16 bits的資料, 等counter >16時就不再讀DAT的資料, 將每回合錄到的資料傳給SRAM\_Communicator做處理。

* 1. LCD

在這次的實驗中，我們用LCD顯示器來達到三個功能--顯示錄音器目前的狀態、顯示錄音或播放時的秒數，以及播放時的播放倍速。而我們用了三個模組來控制LCD顯示器，分別是：LCD\_SHOW、LCD\_controller和LCD\_txt。

最主要的模組式LCD\_SHOW，負責接收Play或Record傳來的state狀態(iST)和播放倍速(iSPEED)，再傳給LCD\_txt處理成要顯示出來的文字，最後傳給LCD顯示出來。LCD\_txt就是負責把目前的資訊轉換成要顯示出來的數字和英文。而LCD\_controller則是負責控制LCD\_EN、LCD\_RS和delay等訊號，好讓每個字的輸出都能符合LCD的輸出標準。

* 1. SevenHex\_Decoder

DE2\_115有8個7段Decoder, 從Hex0到Hex7, 我依序設定為：

* Hex0 : Record module state
* Hex1 : Play module state
* Hex2 : play\_complete 訊號
* Hex3 : SW[0] 訊號
* Hex4 : Play\_speed 十位數
* Hex5 : Play\_speed 個位數
* Hex6 : Speedup Parameter
* Hex7 : State 圖示化
  1. Debounce

這部分的模組純粹是要解決按鈕的訊號不構穩定的問題而已，透過這個模組的幫助，我們可以得到一個比較平整且穩定的Keydown，也就是按下去後的訊號會維持比較穩定的1。