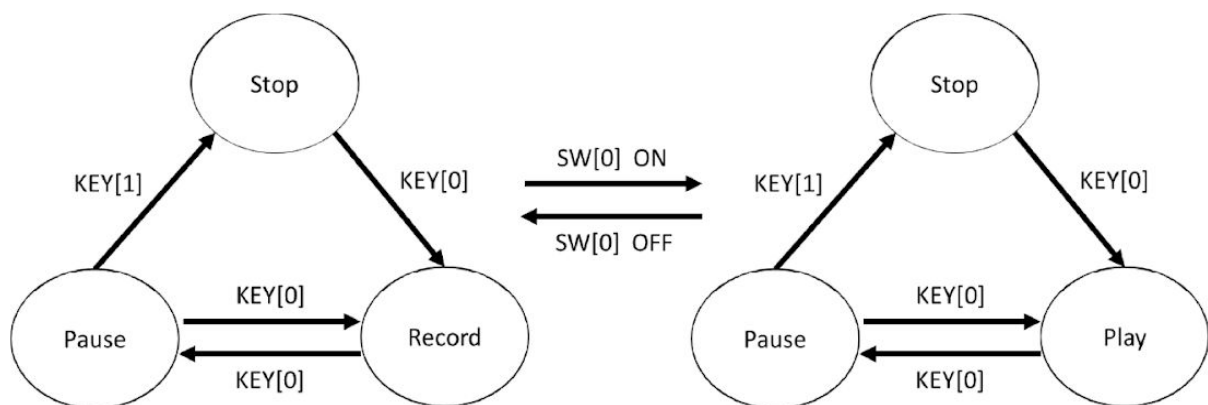


# Lab3

Team04 韓秉勳 蔡昕宇 解正平

## User Manual

- 連結喇叭與麥克風至FPGA板
- Record mode
  - 將SW[0]開關向下撥，七段顯示器顯示RECORD
  - 按KEY[0] 開始錄音(LED全亮)
  - 可在錄音中途按KEY[0]暫停(LED閃爍)，再按一次KEY[0]可繼續錄音
  - 按下KEY[1]可停止錄音
- Play mode
  - 將SW[0]開關往上撥，七段顯示器顯示PLAY
  - 按KEY[0]開始播放
  - 再按KEY[0]可暫停，按KEY[1]可停止播放
  - 按KEY[2]可加快速度，KEY[3]則變慢速，最多支援至8倍速或1/8倍速
  - 在慢速模式時，可透過打開SW[1]啟動interpolation



# Teaching Manual

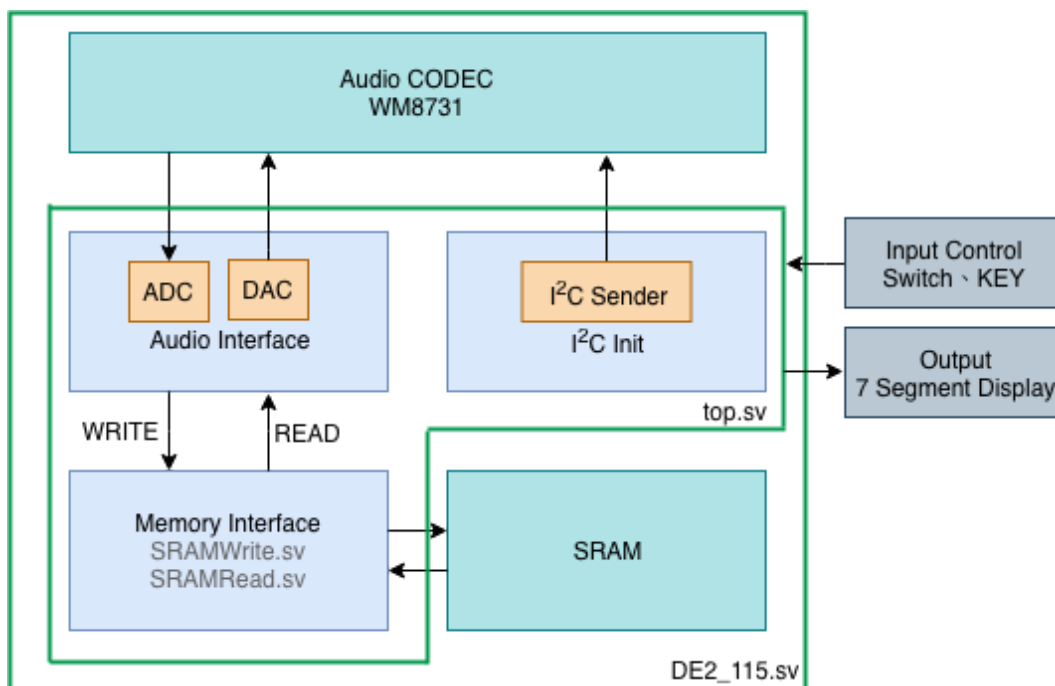
## I. 實驗目的

以FPGA實作錄音機，錄音機可以錄音、播放、暫停、停止以及快慢快進。

## II. 實驗介紹

- A. Main function : 錄音，播放，暫停，停止。
- B. Sampling rate = 32 KHz，每個sample含有2 bytes (16 bits)資料。
- C. 最長可錄音32秒。
- D. 快進倍率為2-8倍，放慢為0.5-0.125倍。
- E. 實作constant interpolation以及linear interpolation。
- F. 使用七段顯示器呈現使用者介面。

## III. 實驗module

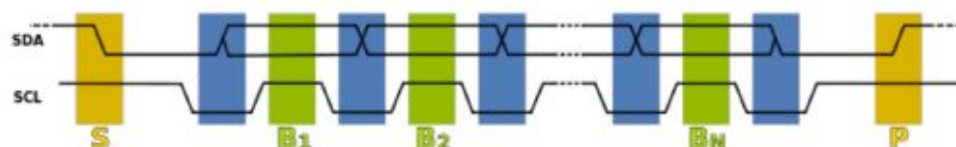


本次實作的 top module 為 top.sv，進一步實作 I2C protocol 的初始化，以及實作 Recorder 的介面和 SRAM 資料記憶的處理。各個 Module 的 finite state 實作在實作方法中詳細述明。

## IV. 實驗方法

### A. I2C protocol

為了初始化 WM8731我們必須想辦法丟資料給此晶片，而溝通方式便是用 I2C protocol，使用SDA和SCL兩個port來管理，如下圖。



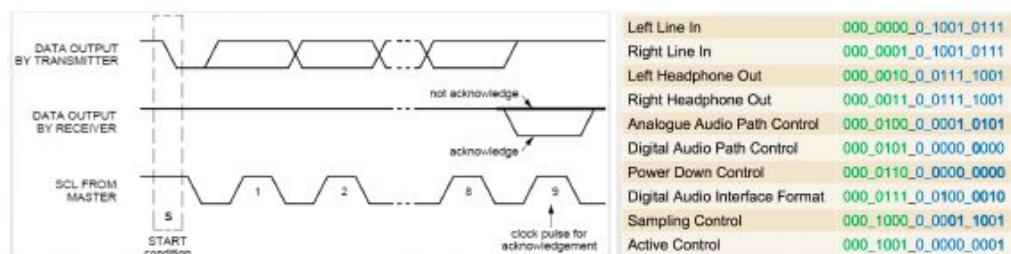
SCL=1, SDA必須是valid data (green)。

SCL=0, SDA設定transferred bit (blue)。

每次傳輸會以8 bits為單位傳輸，傳輸完給acknowledge bit(0)。

當不需要使用I2C protocol, 就將SDA以及SCL設為1。

為了初始化，總共有10個register需要初始化。



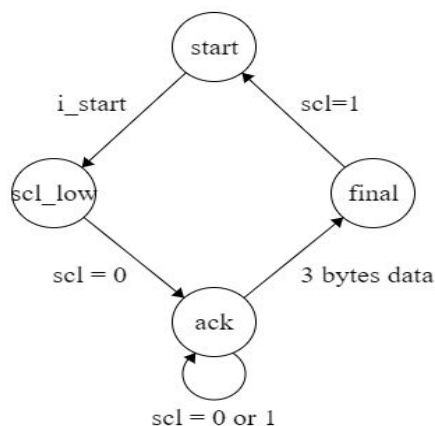
### 1. I2C init :

因為有10個registers需要初始化，表示要用I2C protocol十次，我們設計此module來達到每次傳送24 bits。

### 2. I2C send :

每次需要傳輸24bits資料，但一次只能8bits所以要拆成3組。

- 找到WM8731地址0011010, (0-7bits)(固定)。
- 找到要初始化的registers, (8-14bits)(綠色bits+1藍色bit)。
- 傳送要初始化的資料, (15-23bits)。



Start : 初始化counter以及輸出資料，並讀取input 資料。

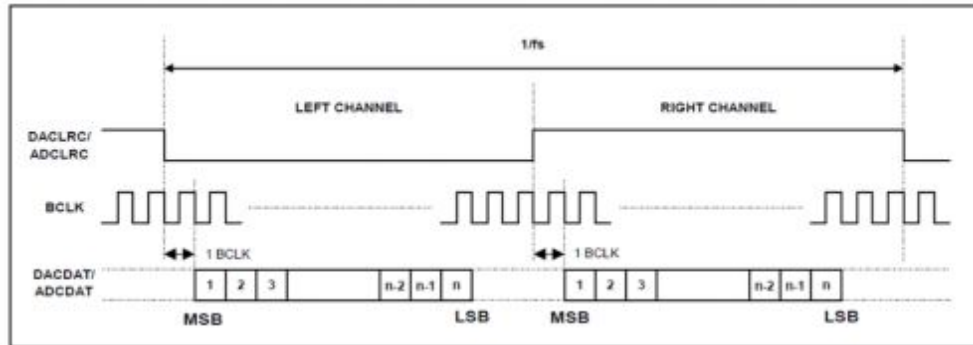
SCL\_low : 將SCL設為0，表示準備傳輸資料，並將輸出資料設為input[23]最左邊的資料，然後shift input 資料 << 1。

Ack : 傳送資料之前需要像圖中綠色藍色一樣，每組8 bit都要先將SCL從1到0等待資料準備好，SCL從0到1才開始傳輸。  
當輸出完一個byte的資料，需要傳送一個ack(1'bz)資料。

Final : 當SCL是1，將輸出資料以及finish資料設為1，輸出資料會使SDA就一直是1，下次使用還需要初始化。

## B. WM8731

當初始完WM8731，DACLRC表示使用者將處理後的音訊送進數位類比轉換器，而ADCLRC則代表使用者由類比數位轉換器接收音訊。我們將BCLK設定為12MHZ，作為module的main clock，然後sample rate設定為32kHz (ADCLK or DACLK)。在一個sample clock中，0表示左聲道1表示右聲道，因為左聲道和右聲道錄音聲音相同，我們只需要取一邊就好，資料紀錄長度我們設定為16bits，把音效晶片出來的資料傳入SRAM當中。



## C. Parallel2Serial and Serial2Parallel

因為音效晶片端是serial傳輸資料，SRAM則是以parallel的格式傳輸。換句話說，需要將兩組資料型態做變換，以16bits當作一個package給sram。

## D. SRAM使用

在本實驗中，CE\_N、LB\_N、UB\_N 的值皆設為 0。由於DE2-115 上記憶體字元長度(word length)為 16bit，共可存 1024k 個字元，總容量約為 2MB。若以單聲道 16bit、32kHz 設定音訊資料規格，所以記錄長度約為 32 秒的聲音訊號。

Mode						I/O PIN		V <sub>DD</sub> Current
	WE	OE	OE	LB	UB	I/O0-I/O7	I/O8-I/O15	
Not Selected	X	H	X	X	X	High-Z	High-Z	I <sub>SB1</sub> , I <sub>SB2</sub>
Output Disabled	H	L	H	X	X	High-Z	High-Z	I <sub>CC</sub>
	X	L	X	H	H	High-Z	High-Z	
Read	H	L	L	L	H	Dout	High-Z	I <sub>CC</sub>
	H	L	L	H	L	High-Z	Dout	
	H	L	L	L	L	Dout	Dout	
Write	L	L	X	L	H	Din	High-Z	I <sub>CC</sub>
	L	L	X	H	L	High-Z	Din	
	L	L	X	L	L	Din	Din	

#### E. ALTPLL

Qsys可以跑出ALTPLL的module，用來產生需要的clock。

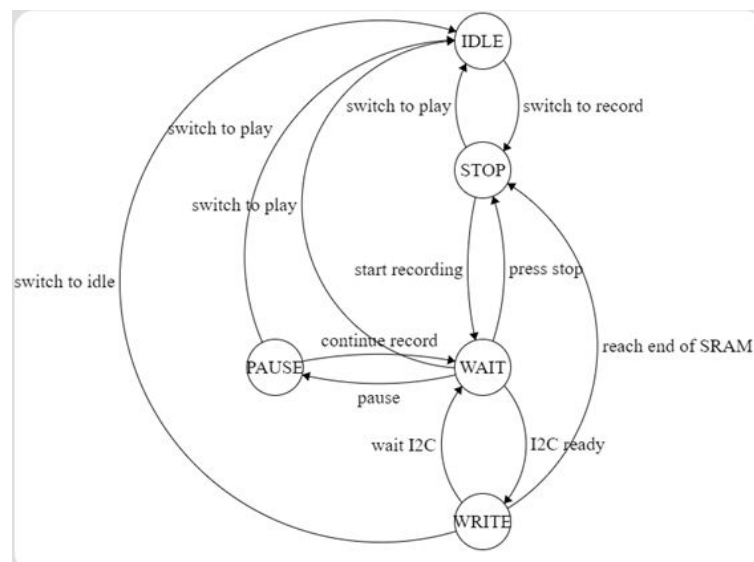
我們將WM873會產生的clock包括12MHz的BCLK，32kHz的LRC\_CLK，作為我們主要的clock訊號，另外ALTPLL的50MHz可以產生12MHz和100kHz，其中12MHz給AUD\_XCK，100kHz給I2C可以初始化WM8731。

#### F. Recorder 實作

由於左右聲道音訊資料相同，為提高儲存時間長度我們只取左聲道的訊號，其FSM如下圖所示。在每個ADCLRCK cycle中，我們將16bit的input serial訊號(以BCLK速度傳送)轉為parallel訊號。要注意的是在ADCLRCK跳之後會有1個BCLK的cycle，要記得跳一個cycle再寫入。當寫入時，資料會在WAIT狀態等16bit都送到後再由WRITE進行寫入作業，並把SRAM address依序增加。

另外，當機器在除了record之外的模式，要把SRAM\_DQ設為high impedance以避免與read data衝突，同時write enable 訊號也要變為disable。

最後，當SRAM address讀到 1048575時，代表SRAM到達儲存極限，故會強制結束錄音，進入STOP狀態。



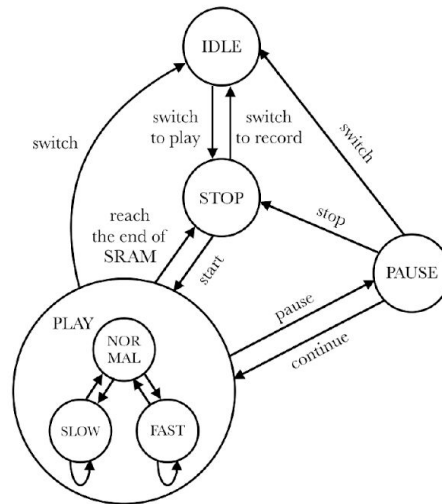
#### G. Player 實作

讀取SRAM資料，再透過I2S傳到WM8731，並轉換為類比訊號由喇叭播放。在播放時也是每次讀取16bit再一個個address進行播放，而播放時還是只取左聲道進行播放。

若要快進時，只需要跳過幾個address進行讀取即可，如若想要4倍速播放，每次address只要從加1變加4即可。

至於慢速播放，如要1/2倍速播放，只需將data hold住2個cycle再輸出即可。若要更精確結果，可以再做linear interpolation，但其須經過大量乘除

法，耗費硬體資源，故替代方案之一是算出cycle中的平均step =  $(b-a) / n$ ，再加回原本的資料點s來避免過多運算。或者也可以將兩個資料以固定比例相加(如  $a*0.25+b*0.75$ )，這比例可以binary floating point表示，避免運用除法。



## H. 七段顯示器

我們將七段顯示器分成三個部分，分別是紀錄時間，標示狀態，標示速度。

### 1. 紀錄時間

因為SRAM總共只有2MB，我們將2MB分成32個時間間隔(秒)，然後查看現在是read或是write哪一個SRAM address，比如說address如果是在0001\_1000\_0000\_0000\_0001，就會顯示時間04，因為只要check MSB的5個bits就可以得知在哪個時間間隔。

### 2. 標示狀態

總共分為兩種mode，顯示record或是play，每個mode又可以顯示除了正常功能還有pause以及stop。

### 3. 標示速度

在play mode，使用者可以調整聲音播放速度，從+8到-8的倍率都可以顯示讓使用者知道現在是多少倍的語速。



Playing at Speed 1/2 to 30 seconds



Recording to 21 seconds

## I. LCD顯示器

實作與七段顯示器顯示一樣的訊息，透過LCD control控制LCD的delay資訊，因為顯示每個文字都需要經過一連串的delay限制，必須符合才能完成設定。每個窗格都是一個地址，需要對每一個地址指定要是甚麼文字，實驗結果如附圖。

