

Team01 Lab3 Report

– 使用所需器材與架設方式

這次實驗用到的器材有DE2-115 FPGA 板、電源線與USB傳輸線。打開FPGA版的電源並將左下角的開關切到RUN。利用 USB 傳輸線將電腦與 FPGA 板連接後，用Quartus II 把程式燒到FPGA上。將麥克風連接Mic In，喇叭(或耳機)連接Line Out 及架設完成。



DE2-115 FPGA 板



電源線



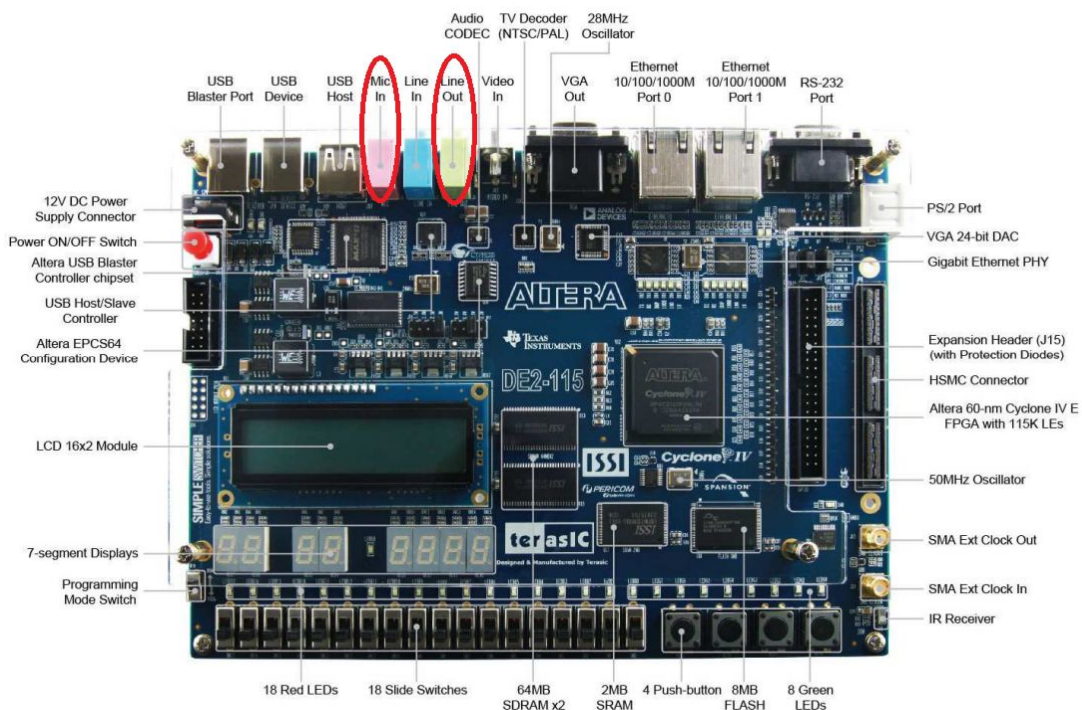
USB 傳輸線



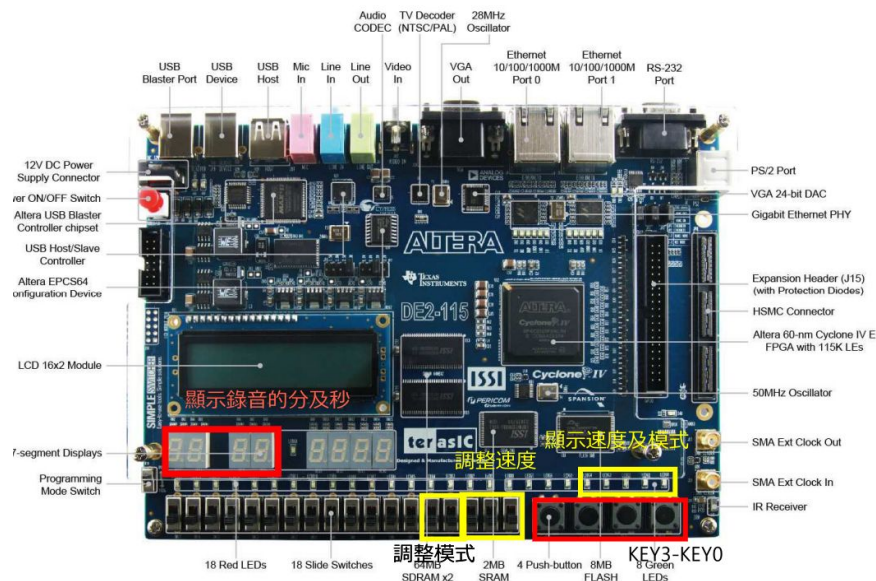
麥克風



喇叭 (或耳機)



– 使用方式與詳細步驟



Key3 : RESET

Key2 : STOP (切換至IDLE模式)

Key1 : 進入 PLAY 模式、PLAY 模式下的 PLAY/PAUSE

Key0 : 進入 RECORD 模式、RECORD 模式下的 PLAY/PAUSE

七段顯示器 : 左邊顯示目前的分鐘數, 右邊顯示秒數

SWITCH : 向上為1, 向下為0

LED : 與SWITCH同步, SWITCH向上為亮, 向下為暗

4號SWITCH (右起第5個) : 切換零次一次內插, 1為一次, 0為零次

3號SWITCH (右起第4個) : 切換快速或慢速, 1為快速, 0為慢速

2至0號SWITCH : 調整速度, 三個SWITCH構成0~7的數字, 分別對應1~8 (以下稱為N), 快速模式下會加快N倍, 慢速模式下會放慢N倍

使用方法 :

1. 任何時候按下 RESET 按鈕, 會清空所錄的內容, 並切換至 IDLE 模式
2. IDLE 模式下按下 KEY1 可進入 PLAY 模式
3. IDLE 模式下按下 KEY0 會清空所錄的內容, 進入 RECORD 模式
3. RECORD 模式下按下 PLAY/PAUSE 即可開始/暫停錄音, 記憶體用完則跳回 IDLE 模式
4. PLAY 模式下按下PLAY/PAUSE 即可開始/暫停撥放, 播完則跳回 IDLE 模式
5. 任何時候按下 STOP 可回到 IDLE 模式
6. 任何時候調整 SWITCH 可調整速度及內插法

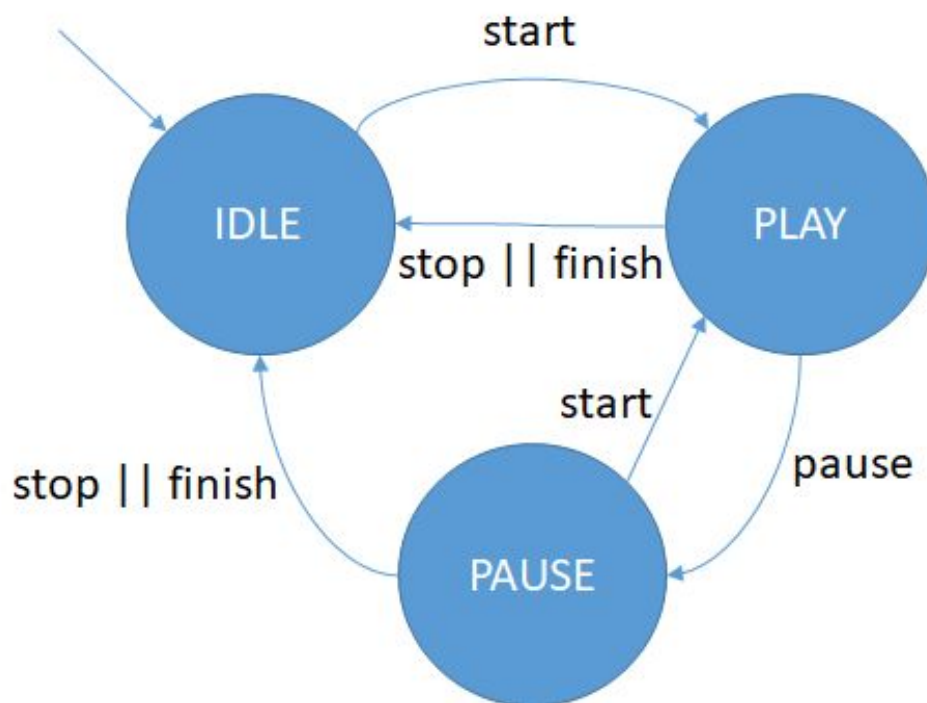
– 實作設計技術細節與巧思

I2C Initializer:

有兩個主要的部份，SEND及FINISH。FINISH的部分會將SCL與SDA拉高，若還沒完成Initialize，則將二者拉低，進入SEND。

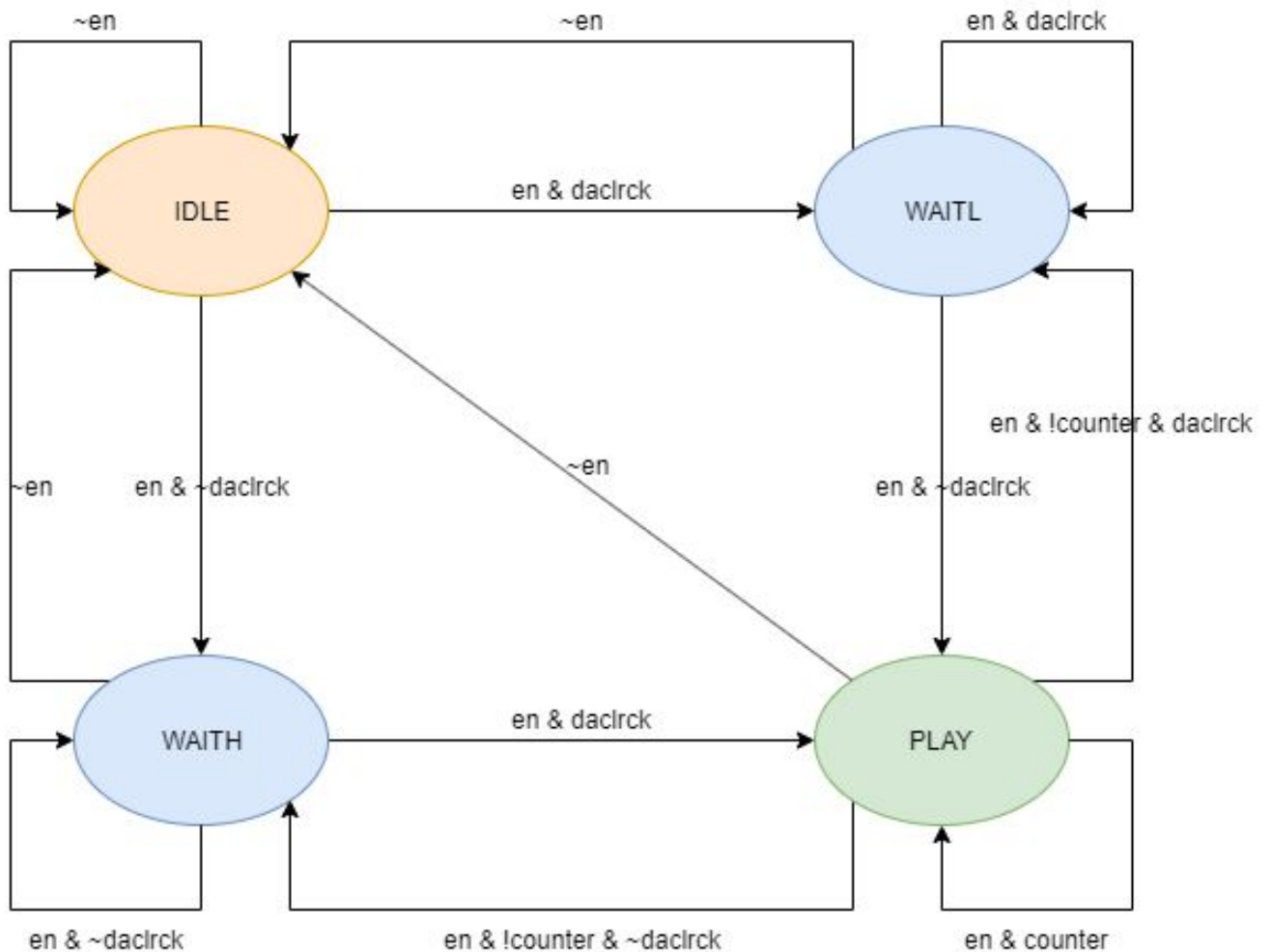
SEND以3個Cycle為單位，SCL分別為低高低，SDA在SCL為低的兩個Cycle之間切換其值，其餘時間不會改變，當傳輸24bit之後，會將SCL及SDA拉低並切換至FINISH。關於ACK的部分，由於利用OEN判斷是否要assign 1'bz的邏輯已寫在Top，因此ACK時不須設定SDA。

DSP:



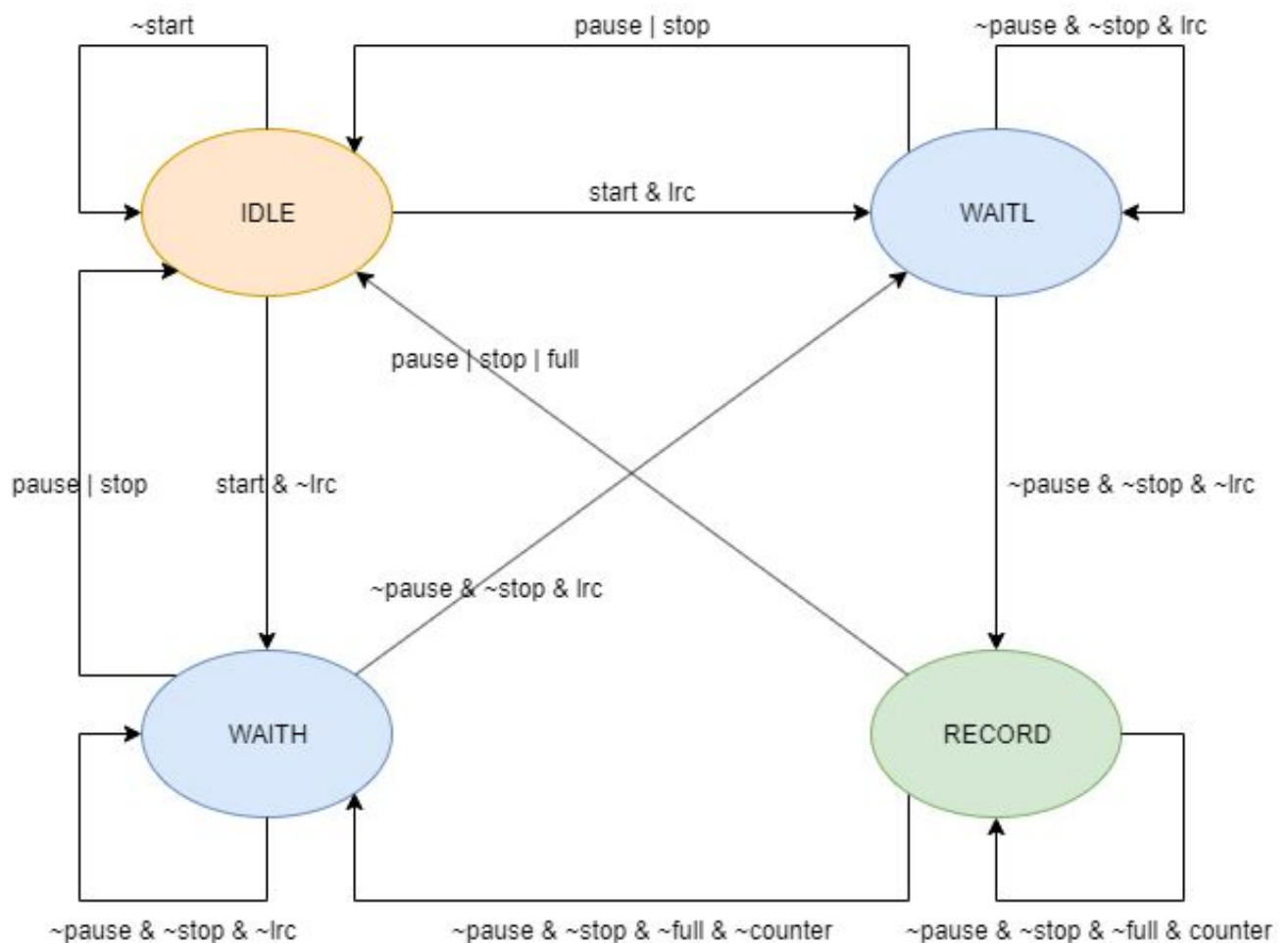
FSM 中分成3個states，當沒有播放時會待在IDLE；PLAY負責去SRAM拿值，接著做DSP之後傳給AudPlayer進行播放；PAUSE則是負責記錄目前播放的位置，當回到PLAY時可以繼續播放。DSP的部分，分成正常播放、快速播放(2-8倍速)和慢速播放(1/2-1/8倍速)。在正常和快速播放這兩種模式中，不需要額外的資料處理，只是需要改變SRAM位置的間隔。慢速播放則分成零次內差和一次內差兩種模式，零次內差需要重複當前的值；一次內差則是用當前的值和下個值做線性的內差。插入幾個點是看慢幾倍。

AudPlayer:



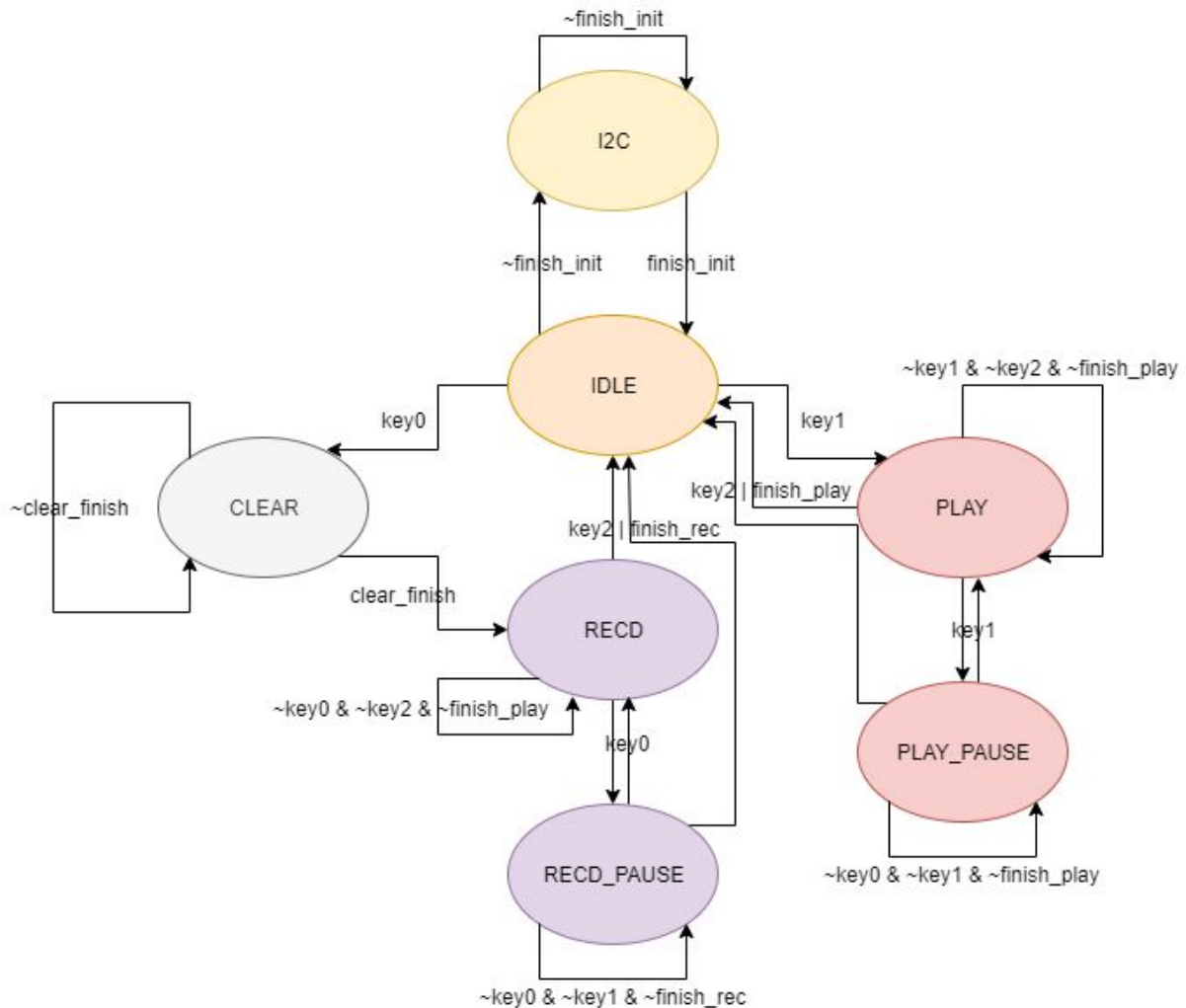
state graph如上圖所示。reset完畢後會進入IDLE的state，並且將counter設為15。在IDLE state當中，當player的module接收到en的訊號，會依據現在的dac1rck是處於高電位或是低電位決定要進入哪個state，若dac1rck為0，表示現正處於低電位，要等待下半個高電位的週期，所以進入WAITH的state，反之則進入WAITL的state。因本次的設定是兩個聲道都會進行聲音的播放，所以兩個state是對稱的，也就是說，當處在WAITL時接收到dac1rck，將進入PLAY的state；當處在WAITH時接收到 $\sim dac1rck$ ，也將進入PLAY的state。進入PLAY state之後，便從dac_data的MSB開始輸出，並且將counter逐步減1，當counter歸零之後，會依據當時的dac1rck決定要回到WAITL或是WAITH以等待下次的data，如此持續循環。直到en變為0，不管當時在哪一個state，都會回到IDLE。

AudRecorder:



state graph如上圖所示。reset完畢後會進入IDLE的state，並且將counter設為15，將address設為-1。在IDLE state當中，當recorder的module接收到start的訊號，會依據現在的lrc是處於高電位或是低電位決定要進入哪個state，若lrc為0，表示現正處於低電位，要等待下半個高電位的週期，所以進入WAITH的state，反之則進入WAITL的state。因本次的設定是只有一個聲道(左聲道)會進行聲音的紀錄，當處在WAITH時接收到lrc，會先進入WAITL等待下一次lrc低電位(表示左聲道的資料傳輸)，才進入RECORD的state，並且在此時將address值加1。進入RECORD state之後，便依序讀取16次data的bit，從MSB開始，assign為一個16個bits的data，當counter歸零之後，會回到WAITH以等待下次循環。另外不管在甚麼時候接收到stop的訊號，都會回到IDLE state，並將address設為-1，表示歸零。若接收到pause的訊號，也會回到IDLE state，但address值不會變，使得下次要繼續record時可以接續此address繼續存值。

Top control:



state graph如上圖所示，reset完畢後會進入IDLE state。在IDLE state當中，若finish_init尚未被設定為1，則會進入I2C state進行initialization，直到I2C initialize module將此訊號設定為1，才會回到IDLE state。接著若key0被按下，會先進入CLEAR state以將memory清空才進入RECD state，在此state中，若按下key0，會進入RECD_PAUSE state，此state中按下key0則會回到RECD state。並且在這兩個state當中，若key2被按下，或者是finish_rec被設為1(記憶體滿了)，都會回到IDLE state。若在IDLS state中按下key1，則會進入PLAY state，在此state中，若按下key1，會進入PLAY_PAUSE state，此時按下key1則會回到PLAY state。並且在這兩個state當中，若key2被按下，或者是finish_play被設為1(記憶體播完了)，都會回到IDLE state。

– 碰過的問題或挑戰與解決方式

我們這次的實驗主要遇到了以下幾個問題：

1. Top 無法正常切換 State

一開始燒到板子上沒有任何反應，我們為了Debug把Top的State顯示在七段顯示器上，發現State無法正常切換，最後找到BUG是出在Top的always_ff，使用的clk應要是i_clk而不是原先寫的AUD_BCLK，還有觸發RESET的應是i_rst_n的negedge而不是posedge。

2. I2CInitializer無法正常切換State

前一個問題解決後，I2C也無法切換State，這次也有兩個原因，第一個出在start，我們原本寫的start是trigger signal，只會拉起來一個Cycle，不過由於Top的clock是12MHz，I2C只有100KHz，start為1的時間太短導致I2C沒有接收到start的訊號，把start改為constant 1即可解決。

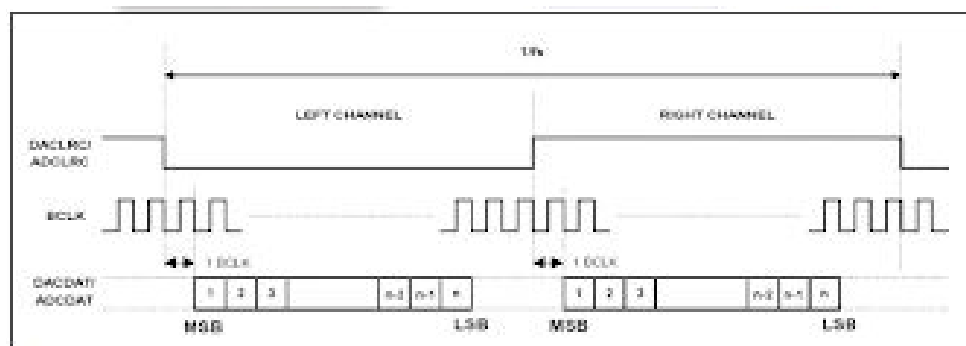
第二個原因在Top.sv，input port的i_clk_100k的k是小寫，然而傳入I2CInitializer的port的i_clk_100K的K是大寫，clk不會動，自然也無法切換state。

3. DE2-115的接線問題

助教給的DE2-115.sv的接線有兩個小問題，首先接到AltPll的reset_n的應是KEY[3]而不是key3down，另一個是傳入SevenHexDecoder的recd_time及play_time沒有宣告，所以預設是1個bit，讓七段顯示器只會顯示1和0。

System Verilog 看到未宣告的logic名稱會自動宣告，不會跳出error，這對於top module的接線是很方便的設計，不需要每條接線都宣告，但當如果小心打錯名字，如前一個問題的大小寫K，Debug時就會非常困難。

4. I2S的cycle問題



如上圖所示，DAT channel需要在LRC channel切換後的一個BLCK cycle過後開始存取資料。然而因為我沒有注意到是在BCLK訊號negedge時才會開始DAT channel的存取，所以原本以posedge trigger的寫法刻意延遲一個cycle開始存取資料，在上圖設計中便會造成MSB的存取錯誤，使得一開始播放出來的聲音有非常大的雜音。後來發現這個問題後，將刻意延遲的寫法改掉，聲音便變得非常清晰。

5. 嘗試使用SDRAM

為了使用SDRAM，我們查詢ED2-115的使用手冊以確認每個腳位的功能，卻發現依照容量理應有25bit的Address，SDRAM卻只有提供13bit的Address input，於是我們查詢SDRAM的製造商ISSI所寫的使用說明，發現SDRAM的存取有些複雜。

一個SDRAM分為四個Bank，由BA腳位select要使用的Bank，每個word在Bank的Address分為Row(13bits)及Column(10bits)，且由於SDRAM的運作原理，在select Row之前要先Precharge。

依我的理解，存取SDRAM的大致流程為：

1. Precharge
2. Activate Row (此時選擇Row Address)
3. Read/Write (此時選擇Column Address)
4. 若要選擇同一Row的其他Column，重複第三步；若要選擇其他Row，跳至第一步

而存取時也要注意每個指令都有不同的Latency。

實際寫好之後，雖然確認過電路的行為與所期望的相同，卻無法如預期般讀寫，應是誤解了某部分的操作流程，不過由於當時時間所剩不多，只好放棄使用SDRAM。

附上ISSI的SDRAM使用手冊：

http://www.issi.com/WW/pdf/42-45R-S_86400F-16320F.pdf?fbclid=IwAR3cahrUtp8xeCsFnRecXfhcVHRQ3yY-9Su0l8k4U-0_9L3oXWB1MG-G2Us