

數位電路實驗三

實驗日期：2015/11/12~12/02

數位錄音機

實驗三－教學手冊

第七組：電機四 B01901147 黃柏瑋

電機四 B01901110 李沂芳

電機四 B01901078 陳彥劭

目錄

(一) 產品簡介-----	3
(二) 程式設計架構-----	3
1. 主架構-----	3
2. State Graph of Finite State Machine -----	3
3. 使用模組與器材-----	4
(三) 實作程式-----	4
1. 總論-----	4
2. DE2_115.sv-----	4
3. top.sv-----	4
4. record.sv-----	4
5. play.sv -----	5
6. i2cSender.sv-----	5
7. hex.sv-----	6
8. debounce.sv-----	6
(四) Debug 補帖-----	7

(一) 產品簡介

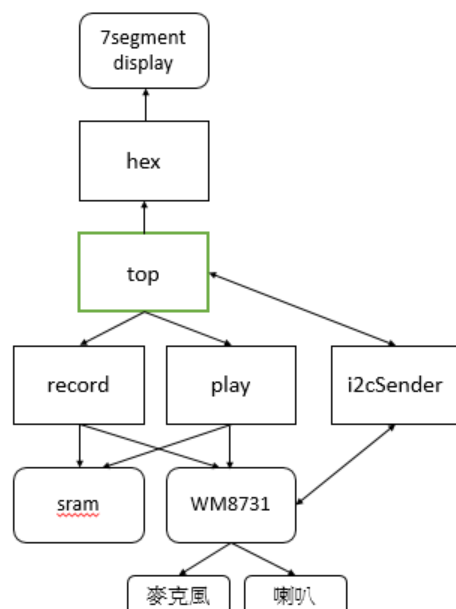
這是一台**數位錄音機**，可以錄音並以 1/2、1/3、1/4、1/5、1/6、1/7、1/8、1、2、3、4、5、6、7 或 8 倍速播放，也可以暫停與停止播放。

本產品的技術可以應用於變聲器、DJ 音響與賣場大聲公重複播放等跟聲音處理相關的地方。

(二) 程式架構設計

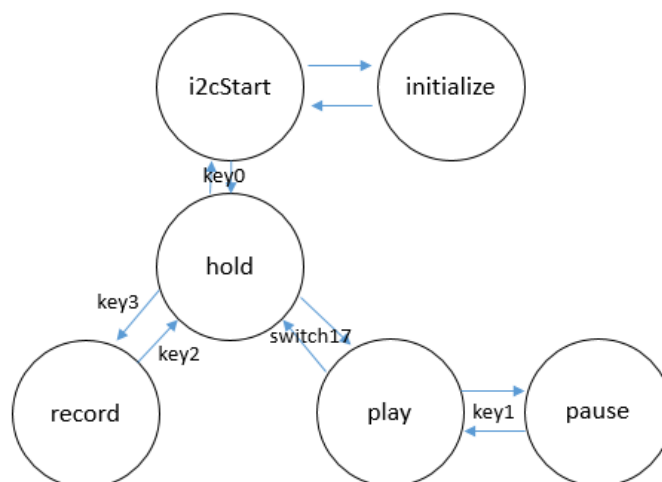
1. 主架構

請見下圖，各模塊在(三) 實作程式中有詳述。



2. State Graph of Finite State Machine

請見下圖。



a. hold

系統等待且不做任何動作。

b. i2cStart

系統重置，往返 initialize 透過 i2cSender 初始化 AudioCODEC。

c. initialize

初始化 AudioCODEC，完成後進入 hold。

d. record

錄音並儲存到 sram 中。

e. play

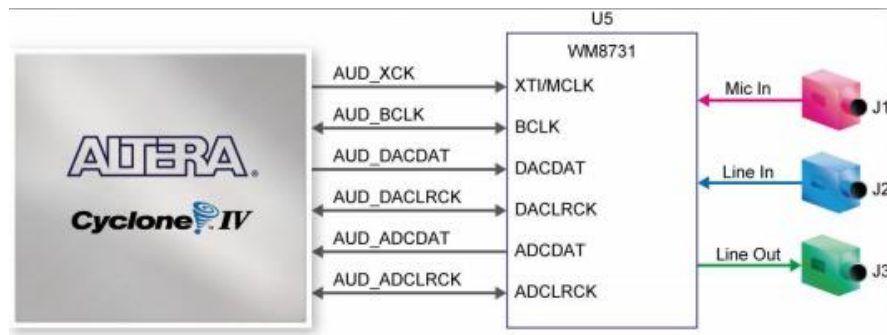
從 sram 取出音訊並播放。

f. pause

中途暫停播放音訊，不做任何動作。

3. 使用模組與器材

DE2_115、PLL、AudioCODEC (WM8731)、SRAM、七段顯示器、麥克風與喇叭。AudioCODEC 與其他模組的連結請見下圖。



(三) 實作程式

1. 總論

實作(二) 程式架構設計的程式碼。

2. DE2_115.sv

此檔為最上層的部分，設定基本 input、output 和 clock，引用各模組並操作整體主架構。若想增加額外功能，可修改此檔。

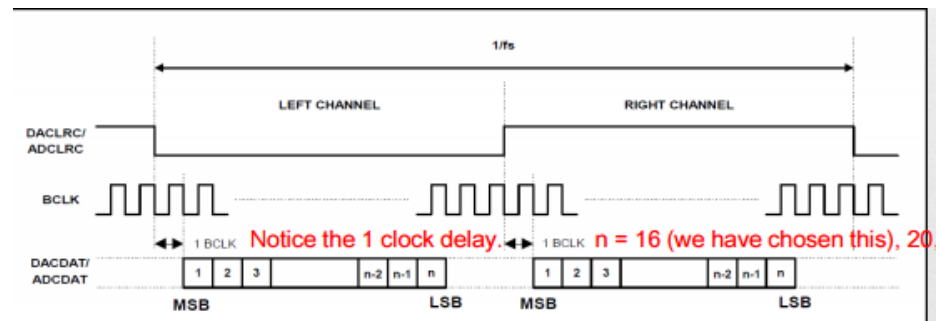
3. top.sv

接收外界按鍵、開關的刺激並輸出相關反應，也控制所有其他 module。實作(二) 程式架構設計 2. State Graph of Finite State Machine。

4. record.sv

將 AUD_adc_data 記錄在 sram 上。top module 每次傳 1 bit 給 record module，record module 每收集 16 bits 後回傳給 top

module(根據如下圖 I2S 協定，在 AUD_adc_lrc 每半個 clock 內，每一個 AUD_bclk 內傳 1 bit 的 AUD_adc_data，記在 sram_data 上)，top module 再將這 16 bits 的 sram_data 根據 sram_addr 寫入 sram。



sram 的控制碼如下圖真值表所示。

Mode	WE	OE	CE	LB	UB	I/O PIN		VDD Current
						I/O0-I/O7	I/O8-I/O15	
Not Selected	X	H	X	X	X	High-Z	High-Z	I _{SB1} , I _{SB2}
Output Disabled	H	L	H	X	X	High-Z	High-Z	I _{OC}
	X	L	X	H	H	High-Z	High-Z	
Read	H	L	L	L	H	Dout	High-Z	I _{OC}
	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 _{OC}
	L	L	X	H	L	High-Z	Din	
	L	L	X	L	L	Din	Din	

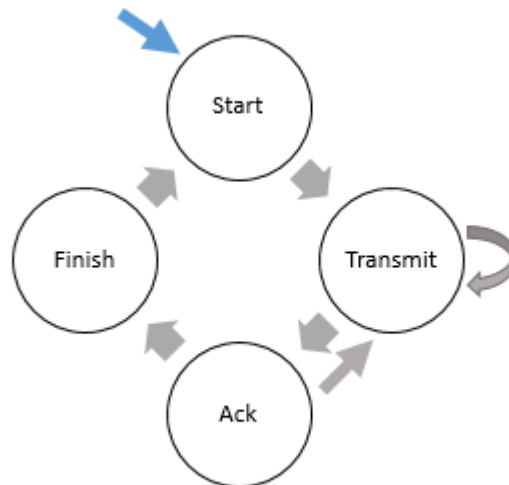
5. play.sv

play 要按照 i2s，把從 SRAM 拿來的資料交給 audio codec，這部分做法跟 record 相似，只是相反。比較麻煩的是加減速的部分。處理加速比較簡單，因為就是跳 address。處理減速是比較麻煩的地方。這裡我們使用一個 slow counter，當 slow counter 走到倍數時，就輸出。譬如慢兩倍，就每兩秒輸出一一次。但中間的要內插。零次內插比較簡單，就是多設一些 register 來記，再用 mux 來決定要用哪個輸出。一次內插就比較麻煩，因為還要計算，要考慮夠不夠快的問題。這裡有兩種做法，一種是把乘除法的數，當作輸入，即時的計算。另外一種是開出很多套硬體，1-8 倍都用，最後只要一個 mux，反正寫 code 時，就是一個 for。最後我們選擇很多套硬體的方法。因為空間夠。以上再加上一些輸出入設定，就可以完成 play 這個模組了。

6. i2cSender.sv

實作 i2c 通訊協定，根據講義模仿 nWave 的波形。

可以分為 4 個 state：Start、Transmit、Ack 與 Finish，如下圖。



a. Start

接收 `i_start` 的指示，進入 Transmit。

b. Transmit

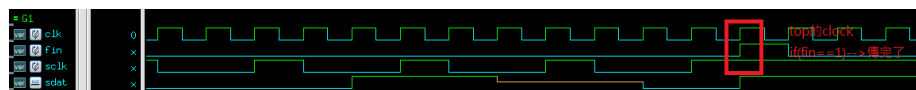
連續傳 8 個 bit，從 LSB 往 MSB 傳。實作方法為每次都傳資料的第 0 位 bit，再將資料 right shift 1 位，下次即可傳下一個 bit。當傳了 8 個 bit 後進入 Ack。

c. Ack

傳送 Ack 給 b. 的資料輸出方。實作方法為將 `o_sdat` 設為 `z` (high impedance)，即表示 Ack，在 nWave 中是 0 與 1 中正間的值。當累計傳送 Ack 的次數等於總資料的 byte 數後(因為每傳 8 bit 後，就要送一個內容為 Ack 的 bit)，進入 Finish。

d. Finish

將 `o_finish` 設為 1，維持 `i_clk` 一個 clock 的長度，波形如下圖。



7. hex.sv

顯示錄音機狀態、快放或慢放的播放倍速與秒數。狀態與播放倍速皆由 `input` 決定，秒數計算方式為將”錄音存在 sram 的終點位址”減去”目前播放到的位址”，再取其 15~19 bit，即為秒數。詳細輸出的位碼對照，請參考實驗一的 `SevenHexDecoder.sv`。

8. debounce.sv

處理按鍵造成的震盪的輸入波形，判斷正確按下按鍵的時間。

(四) Debug 補帖

1. tb_i2c.sv 中 sdat 本來用 logic 宣告，但用 ncverilog 模擬時會出現 structural nest expression 的 error。改成 wire sdat 才過模擬。
2. i2cSender.sv 的變數達 20 餘個，注意在 if...else...中需將在 if (else)出現的變數，在 else (if)也指定一次。
3. 出現的常數都要加上 bit 數，以免預設當成 32 bit，產生 truncation error。
4. 注意資料傳輸順序是從 MSB 或 LSB 開始。