

# LAB3

GROUP 12 [B01901061 王勝輝、B01901100 沈怡廷、B01901140 陳奕安]

## INTRODUCTION

實驗三的目標為實作出一個數位錄音機。讀者可以透過閱讀本手冊，了解如何運用 Altera DE2-115 FPGA Board，透過撰寫 SystemVerilog，完成此一實驗。

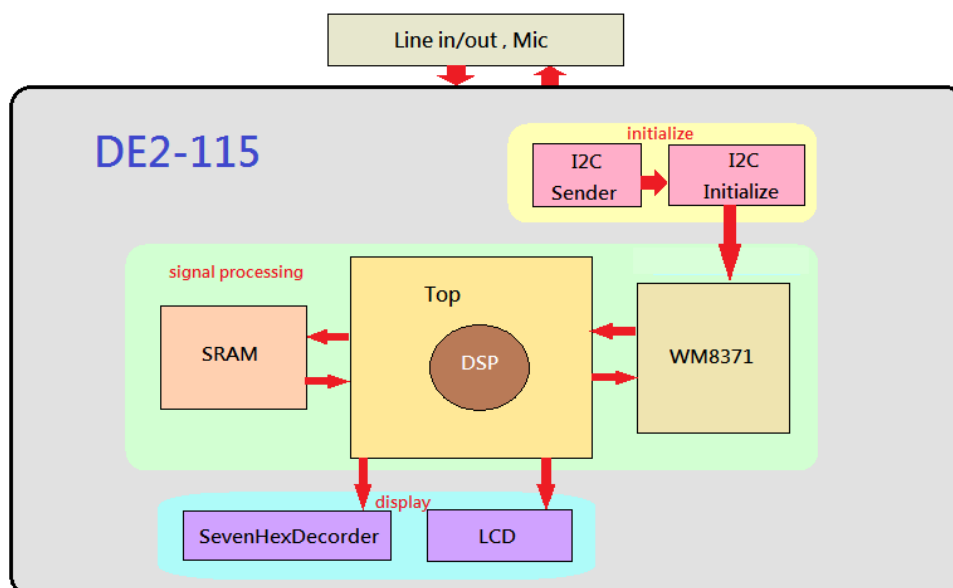
錄音機的功能與特色如下：

- 具備錄音、播放、暫停、停止播放等基本功能。
- 取樣頻率為 32kHz，每個取樣為 16 bits。
- 可以錄製的時間為 32 秒。
- 具備倒放、回音、快速播放(2, 3, 4, 5, 6, 7, 8 倍速)以及慢速播放(1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 倍速，慢速播放含內插信號功能(0 次內插與 1 次內插皆支援))。
- 以容易了解的方式顯示目前秒數與系統狀態。

## EQUIPMENT

- Altera DE2-115 FPGA 板 x 1
- PC 麥克風 x 1
- Line in 線 x 1
- Line out 喇叭 x 1

## SYSTEM STRUCTURE



本實驗會先創立一個 DE2-115 Project，再依序加入各個 system verilog 檔案。

此實驗要完成四大部分：

1. PLL，包含 clk100k.sv, clk12m.sv
2. Initialize，包含 I2cSender.sv, I2cInitialize.sv
3. Signal processing，包含 top.sv
4. Display，包含 SevenHexDecoder.sv, LCDsender.sv, FPGA\_2\_LCD.v

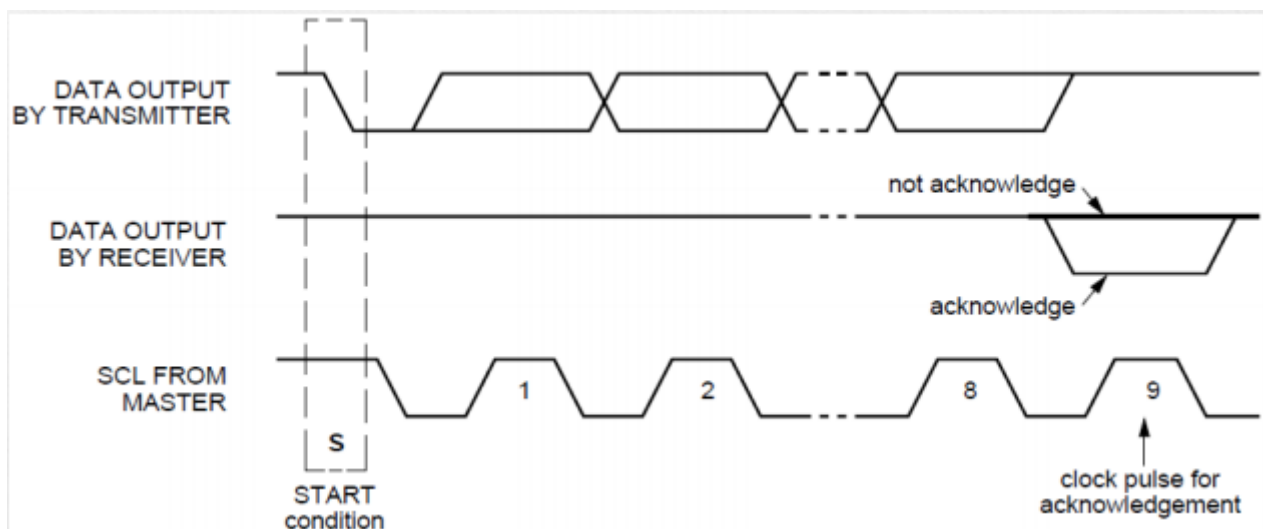
## PLL

我們使用 Quartus 內附的 PLL IP 模組來將 50MHz 的 clock 變成 100kHz 和 12MHz 的 clock。其中與 I2C 有關的功能需要 100kHz 的 clock；其他部分則需要 12MHz 的 clock。

## INITIALIZE

使用 WM8731 處理音訊前，必須先使用 I2C 的格式初始化其內部 registers，分別為 Left Line In、Right Line In、Left Headphone Out、Right Headphone Out、Analogue Audio Path Control、Digital Audio Path Control、Power Down Control、Digital Audio Interface Format、Sampling Control、Active Control。

I2C 格式，主要將初始化資料(共 24 個 bits)依序傳入上述 register，register 每接收到 1byte，會回傳一個 ack，如下圖所示。



初始化的目的，主要在於訂定傳送格式(I2S Format)、資料精確度(16 bits)、sampling rate(32kHz)、訊號強度加強等等。

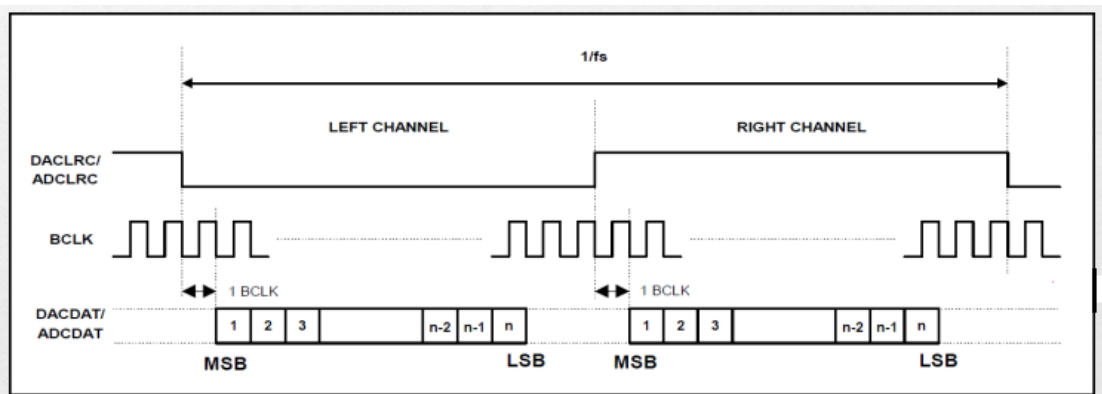
## SIGNAL PROCESSING

此部分是此實驗主要的部分，top 這個 module 會和很多 module 溝通，包括通訊晶片 WM8731、DE2-115 裡的 SRAM 記憶體、輸出錄音機狀態的 LED 和 LCD。

## WM8731

WM8731 有 ADC 和 DAC 的功能，錄音時用 ADC，把接收到的類比訊號轉成數位傳給 top，top 會把訊號存到 SRAM 裡，放音時 DAC，把 top 從 SRAM 裡取出的數位訊號轉成類比再輸出。

Top 和 WM8731 的溝通形式為 I2S，如下圖：



SRAM 的容量為 2M， $2M/32k(\text{sample frequency})/32(\text{sec})=16$ ，等於我們如果要存 32 秒，一次取樣只能存 16 bits，也就是左聲道和右聲道我們只能取一個。

ADC 和 DAC 的運作方式一樣，只是一個為輸入一個為輸出，CLRC 為 0 傳送左聲道的資料，為 1 傳送右聲道的資料，所以 ADC 時我們只在 CLRC 為 0(也可為 1)取資料，但 DAC 時則是 0 和 1 都要。

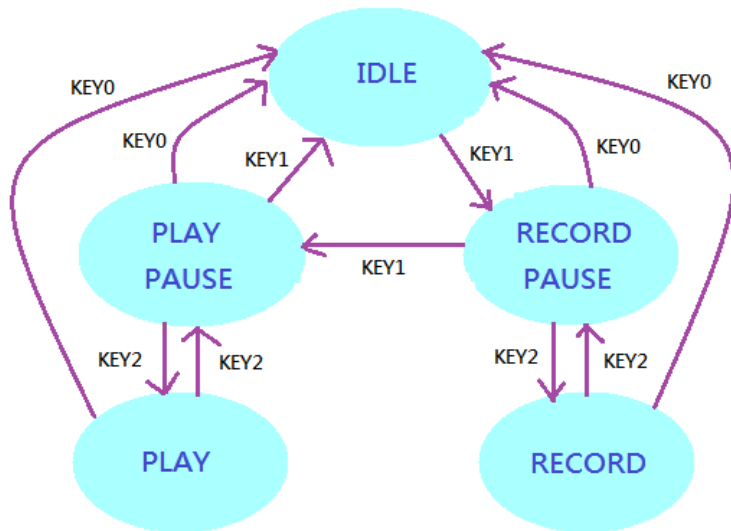
## SRAM

溝通方式如下：

Mode	WE	OE	OE	LB	UB	I/O PIN		V <sub>DD</sub> Current
						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	

此實驗 LB<sub>n</sub> 和 UB<sub>n</sub> 可直接 assign 為 0，因為我們一次存和取都是 16bit，存的時候拉低 WE<sub>n</sub> 和 CE<sub>n</sub>，取的時候拉高 WE<sub>n</sub>，拉低 CE<sub>n</sub> 和 OE<sub>n</sub>，sram\_address 的話就是要存和去的位置，比較需要注意的是 sram\_data，是 inout port，在我們的設計中，存入時由 top 的模組控制，取的時候由 SRAM 模組控制，沒有要控制的一方要放開控制，將 sram\_address 設為 High impedance。

## FINITE STATE MACHINE



---

## IDLE

顧名思義就是閒置的 state，按下 KEY1 可轉至 RECORD mode，再按一次 KEY1 可轉至 PLAY mode，再一次則可回到 IDLE mode，此時所 assign `dacdata = adcddata` 的話，在把 line in 接上音樂，錄音機就可擁有卡拉 OK 的功能！

---

## RECORD PAUSE

為了不讓錄音機剛轉到 RECORD mode 時就開始錄音，我們可以先設成暫停狀態，按 KEY2 可以轉至 RECORD，開始錄音，此時按 KEY2 可暫停，再按就繼續，但只有在 RECORD PAUSE 時按 KEY1 離開 RECORD mode 轉至 PLAY mode，離開時把 `sram_address` 歸 0，使之可以從頭開始播放，按 KEY0 可回到 IDLE。

---

## RECORD

開始錄音，持續接收 WM8731 的訊號(address 增加)，存入 SRAM 中，按 KEY2 可暫停，在按一次可以原 address 繼續錄，按 KEY0 回到 IDLE。

我們可以設了一個 `adccount`，在 `negedge` 時歸零(只取左聲道)，往後每一個 clock 加 1，讓一個 16bit 的 buffer 的第 15-`adccount` 個 bit 等於 `adcddata`，如此就能得到 WM8731 的一筆資料，重複動作進行錄音。

---

## PLAY PAUSE

為了不讓錄音機剛轉到 PLAY mode 時就開始錄音，我們可以先設成暫停狀態，按 KEY2 可以轉至 PLAY，開始播放，此時按 KEY2 可暫停，再按就繼續，但只有在 PLAY PAUSE 時按 KEY1 離開 PLAY mode 轉至 IDLE，離開時把 `sram_address` 歸 0，使之可以從頭開始錄音，按 KEY0 也可回到 IDLE。

---

## PLAY

開始播放，持續從 SRAM 拿訊號資料(address 增加)，送到 WM8731，轉成類比訊號輸出，按 KEY2 可暫停，在按一次可以原 address 繼續放，按 KEY0 回到 IDLE

我們可以設了一個 `adccount`，在 `posedge` 和 `negedge` 時歸零(左、右聲道都放)，往後每一個 clock 加 1，讓一個 16bit 的 buffer 的第 15-`daccount` 個 bit 等於 `dacdata`，如此就能得到 SRAM 的一筆資料，重複動作進行播放。

加速播放其實很簡單，要 2 倍最快的方法就是一個資料播，一個資料不要播，要 8 倍就是一個資料播，七個資料不要播，PLAY 時把原本 sram\_address 加 1 的地方改成加 2~8 即可，但這個方法會造成音調上升的問題。

這裡我們用的是第二種方法，一次撥一段資料(500up)，再一跳一段資料，再撥一段.....，經過嘗試可以使音調不變高。例如 8 倍時 sram\_address 就連續 511 次加 1，再加 3584，重複。

---

### 減速撥放

0 次內插：再設一個 counter，要讀 sram 時+1，若要放慢成 1/2 倍，就當 counter 等於 1 時 address 才加 1，並將 counter 降回 0；若要放慢成 1/8 倍，就當 counter 等於 7 時 address 才加 1，並將 counter 降回 0。

1 次內插：必須再創一個 buffer 來存上一個 data。我們分別把兩個數做以下處理再相加：取正值，乘以比率，使其回到原來正負狀態。例如-3 可能會變為 3，乘以比率後變為 2，再回到原先小於 0 的-2。這種最後再相加的方式可以避免 overflow 並且適合任何正負號的二補數，算是一種簡單的實作方式。

我們可以將加速、減速、0 次內插、1 次內插的控制設在 DE2-115 的 SW0~SW17 上，如此可以很容易的操控，若將 2 倍和 1/2 一起開啟，還可以出現速度不變、音調不變的效果！

---

### 倒轉播放

原本播放時 address 是逐漸增加的，改成用減的即可達到此功能。

---

### 回音功能

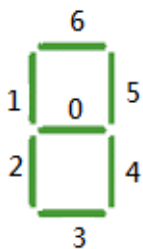
播放時向 SRAM 取資料原本一次只取一個，若要聽起來回音的話，需再向前 1/8 秒、1/4 秒、3/8 各取一個，並乘以 1/2(原資料)、1/4、1/8、1/16 倍進行疊加，原資料也要乘以一半是防止 overflow，這個功能要注意資料是 signed，乘以 1/2 的倍數時容易出錯，我們是先將其轉成 unsigned，除 2 後再轉回來。

## DISPLAY

這部分用來顯示錄音機的當前狀態，分為 SevenHexDecoder 與 LCD 兩個部分。

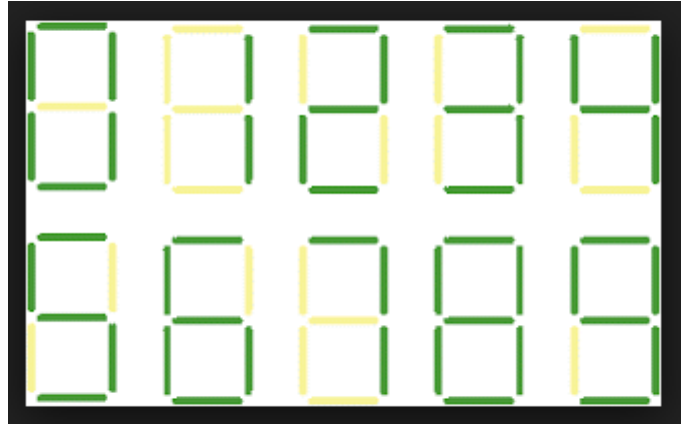
---

### (1).SEVENHEXDECODER.SV



如圖所示，7 個 bit 代表不同位置的燈，要亮就那個 bit 等於 0，不亮則讓他為 1，依照此規則，可以推出現是 0~9 分別要輸出的數。

**0 = 7'b1000000**  
**1 = 7'b1111001**  
**2 = 7'b0100100**  
**3 = 7'b0110000**  
**4 = 7'b0011001**  
**5 = 7'b0010010**  
**6 = 7'b0000010**  
**7 = 7'b1011000**  
**8 = 7'b0000000**  
**9 = 7'b0010000**

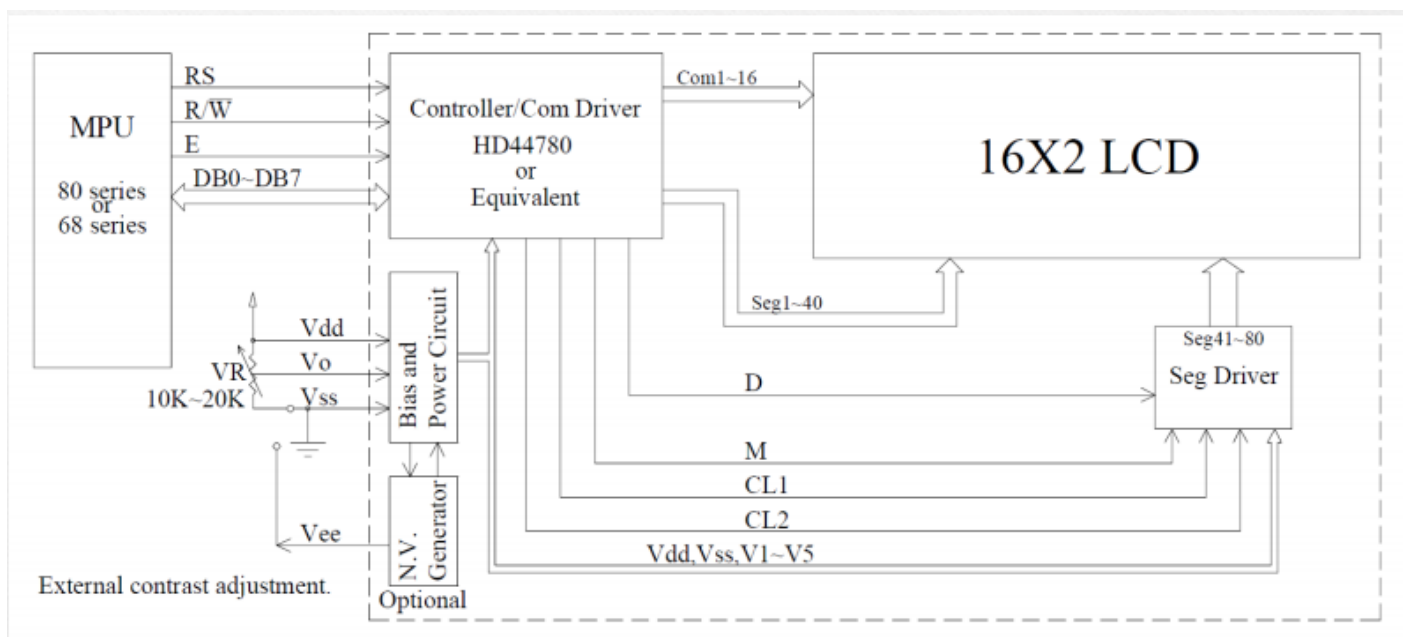


這個 module 的作用就是當接收到一個 2 位數的數字時，正確的輸出兩個分別代表十位數和個位數的 7bit 數，例如 input 等於 31 時，輸出 7'b0110000 和 7'b1111001 給 DE2-115，版子就會顯示 31。

而 Input 來自 top 裡的 sram\_address 的前 5 個 bit，因為我們現在的取樣頻率為 32kHz ->  $2^{15}$  bps，所以 sram\_address 除以  $2^{15}$  後就可以得知現在正在處理的 data 是第幾秒的資料，將這個數丟掉 SevenHexDecoder 裡，DE2-115 上的 LED 就會顯示秒數。

## (2)LCD

使用 LCD 的目的，在於顯示目前的狀態與時間。使用前，與 WM8731 相同，需要先初始化其使用格式(本組使用 8bits, 2lines, 5\*8dots)，須注意每個 instruction 執行所需要的最低時間，系統必須等待其完成指定工作。



使用方法簡單，只須將所需要顯示之資料，傳入 LCD 即可(注意需在 En 下降前設定好 data)，傳入資料的對應表如下圖所示。

Upper 4 bit Lower 4 bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)			0	1	2	3	4					5	6	7	8
LLLH	(2)		!	1	2	3	4	5					6	7	8	9
LLHL	(3)		"	2	3	4	5	6					7	8	9	0
LLHH	(4)		#	3	4	5	6	7					8	9	0	1
LHLL	(5)		\$	4	5	6	7	8					9	0	1	2
			%	5	6	7	8	9					0	1	2	3

## (2)LED

使用 LED 以顯示目前播放音訊之音量，並將之視覺化顯示於 FPGA 板子上。其中，每 1/8 秒變換開啟 LED 之數量，由平均 1 秒音訊音量決定。

