

■ 層級架構

檔案層級架構

```
lab3
|_ ALTPLL
|   |_ synthesis
|       |_ ALTPLL.qip
|       |_ ALTPLL.v
|_ DE2_115
|   |_ DE2_115.qsf
|   |_ DE2_115.sdc
|   |_ DE2_115.sv
|   |_ Debounce.sv
|   |_ SevenHexDecoder.sv
|_ AudDsp.sv
|_ AudPlayer.sv
|_ AudRecorder.sv
|_ I2cInitializer.sv
|_ Top.sv
```

模組層級架構

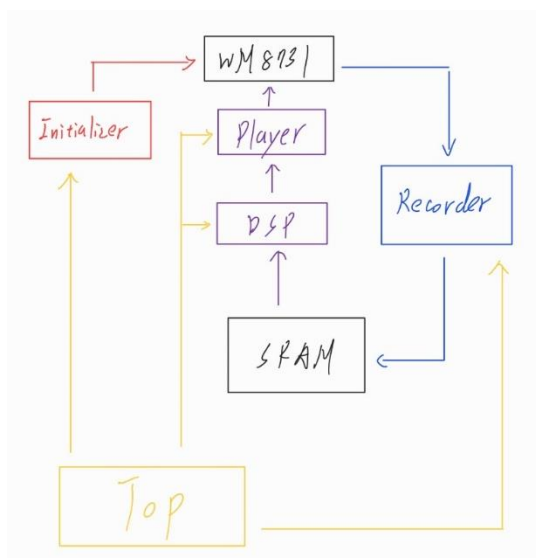
```
DE2_115
|_ pll0 (ALTPLL)
|_ deb0 ~ deb2 (Debounce)
|_ seven_dec0, seven_dec2 (SevenHexDecoder)
|_ top0 (Top)
    |_ init0 (I2cInitializer)
    |_ dsp0 (AudDSP)
    |_ player0 (AudPlayer)
    |_ recorder0 (AudRecorder)
```

■ What We Have Done

baseline : ALL Done

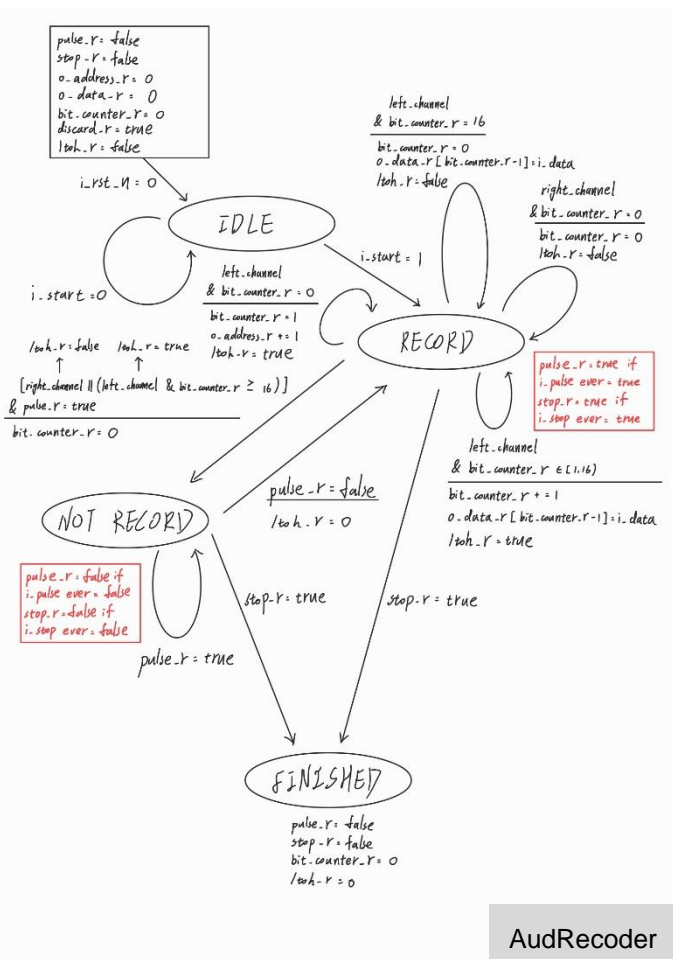
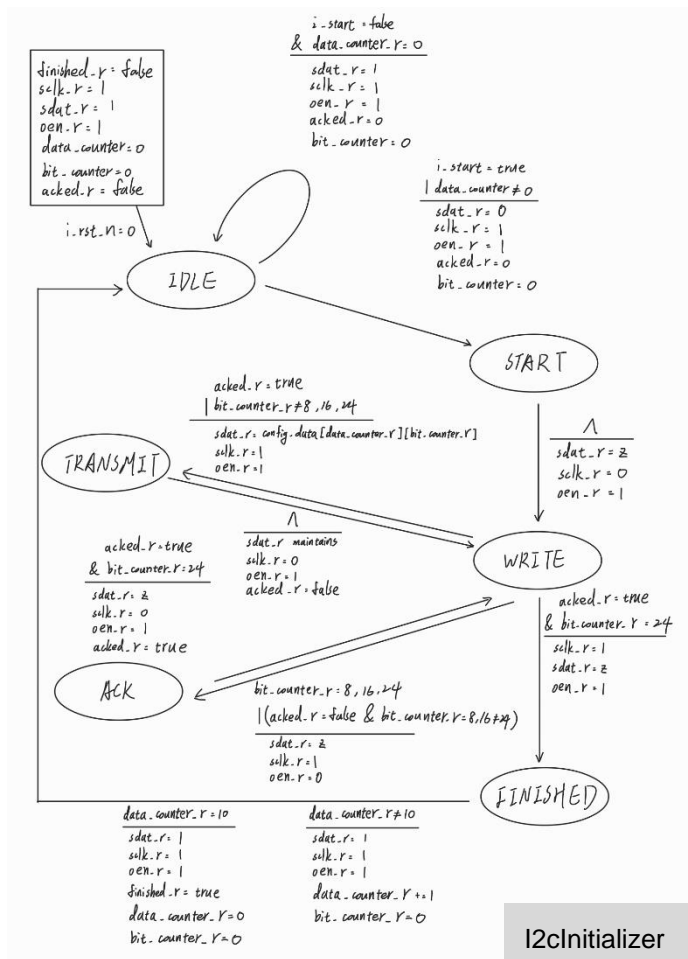
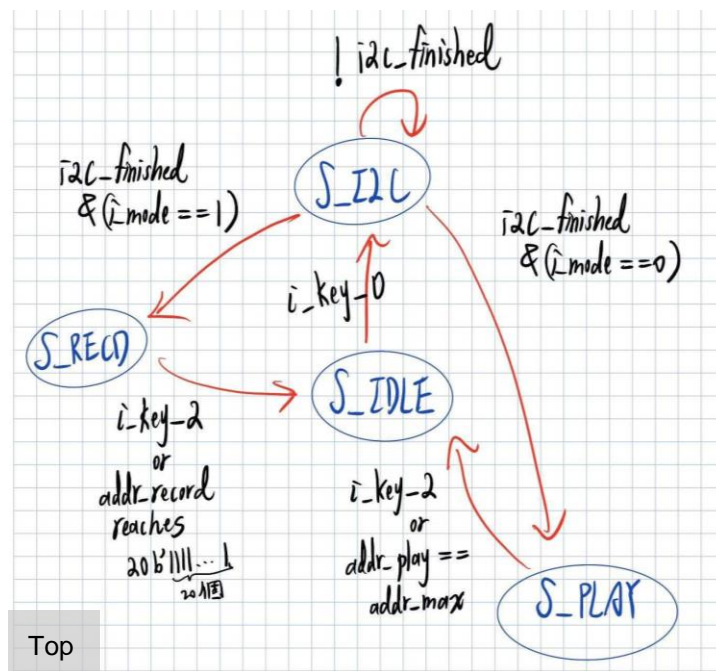
bonus : 倒轉、用七段顯示器顯示狀態和秒數

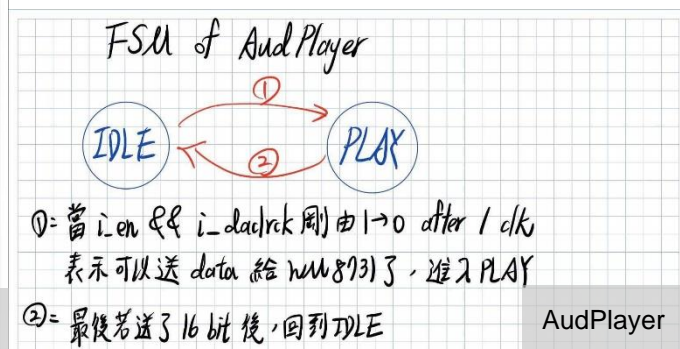
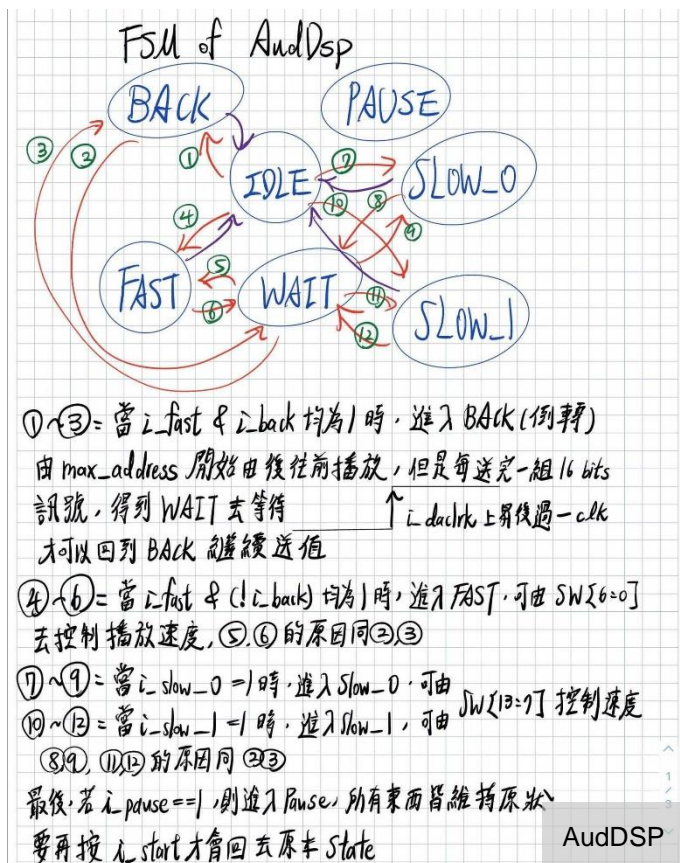
■ Block Diagram



1. Top 負責控制 I2cInitializer、AudRecorder、AudDSP、AudPlayer 四個模組，針對 WM8731 晶片和 SRAM 進行操作。
2. I2cInitializer 首先以 I2C 協定設定 WM8731 的 11 項參數，設定完成後 WM8731 才能進入我們想要的工作模式。
3. WM8731 設定完成後，使用 AudRecorder 從 WM8731 讀入來自 Mic In 的數位訊號，並以 16-bits 為單位寫進 SRAM 直到停止錄音或 SRAM 用盡儲存空間。
4. 錄音完成後，AudDSP 從 SRAM 讀取資料，轉換成快速播放、慢速播放或倒轉的數位音訊號，傳給 AudPlayer 驅動 WM8731 播出。

FSM





Fitter Summary & Timing Analyzer

Fitter Summary

Fitter Summary	
Fitter Status	Successful - Wed Nov 02 23:03:55 2022
Quartus II 64-Bit Version	15.0.0 Build 145 04/22/2015 SJ Full Version
Revision Name	DE2_115
Top-level Entity Name	DE2_115
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	1,040 / 114,480 (< 1 %)
Total combinational functions	1,031 / 114,480 (< 1 %)
Dedicated logic registers	193 / 114,480 (< 1 %)
Total registers	193
Total pins	480 / 529 (91 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	1 / 4 (25 %)

Timing Analyzer

TimeQuest Timing Analyzer
Summary
Parallel Compilation
SDC File List
Clocks
Slow 1200mV 85C Model
Fmax Summary
Timing Closure Recommendations
Setup Summary
Hold Summary
Recovery Summary
Removal Summary
Minimum Pulse Width Summary
Worst-Case Timing Paths
Datasheet Report
Metastability Summary
Slow 1200mV 0C Model
Fast 1200mV 0C Model
Multicorner Timing Analysis Summary
Multicorner Datasheet Report Summary
Advanced I/O Timing
Clock Transfers
Report TCCS
Report RSKM
Unconstrained Paths
Messages

Slow 1200mV 85C Model Setup Summary			
	Clock	Slack	End Point TNS
1	pll0 altpll_0 sd1 pll7 clk[0]	-52.262	-1330.252
2	pll0 altpll_0 sd1 pll7 clk[1]	-5.482	-14.397
3	AUD_BCLK	-3.203	-80.379

Slow 1200mV 0C Model Setup Summary			
	Clock	Slack	End Point TNS
1	pll0 altpll_0 sd1 pll7 clk[0]	-47.266	-1199.664
2	pll0 altpll_0 sd1 pll7 clk[1]	-4.774	-12.506
3	AUD_BCLK	-3.134	-78.994

Fast 1200mV 0C Model Setup Summary			
	Clock	Slack	End Point TNS
1	pll0 altpll_0 sd1 pll7 clk[0]	-25.619	-663.690
2	pll0 altpll_0 sd1 pll7 clk[1]	-2.421	-6.240
3	AUD_BCLK	-1.642	-42.089

操作方式

mode	慢速播放內插	倒放	慢速播放倍率	快速播放倍率
[record/play]	[linear/constant]	[倒放/正放]	[i-th=1 代表 $\frac{1}{i-6}$ 倍]	[i-th=1 代表 i+1 倍]
SW17	SW16	SW15	SW13~SW7	SW6~SW0

reset	stop record/play	pause record/play	start record/play
KEY3	KEY2	KEY1	KEY0

■ 遇到的問題和解決法

- 問題：**不知道 Initializer 和 Recorder 的實際運行狀況，無從 debug。
解法：原因在於我原先不太知道如何寫 testbench，在詢問第四組的做法後，根據 lab1 的 testbench 自己寫出兩者的 testbench，最後根據波型修正不少 bug。
- 問題：**到很晚才發現 I2C 的 SCL 和 SDA 應該是錯開至少一個 clock cycle，可能面臨須要大改程式。
解法：讓 SDA 在 positive edge 改變、SCL 在 negative edge 改變，兩者錯開半個 cycle，既不需要大改程式且加速 Initialize WM8731 的時間。
- 問題：**Recorder 錄進的聲音放出來後變成噪音。
解法：檢查發現是 16-bit data 的順序相反，變成與 player 方向相反，更改 bus 的方向即可正常運作。
- 問題：**助教給的 template code 有幾個錯誤，例如時間的 k 大小寫錯誤，但編譯器抓不到錯誤。
解法：用肉眼檢查出來再狂罵三字經。
- 困難：**因為這次比較多模組，而且都是不同人寫的，所以在用 Top 連接各模組時有出現腳位對不上或是訊號給的方式出錯等等。
解法：再經過互相討論、一起除錯之後這問題就解決了。

■ 心情點滴

lab3 對我們三位組員都非常煎熬，由於遇到期中考週，因此我們的想法是等到期中考週的週末再開始寫 lab3，但當我們在認真研究後才發現時間似乎非常不夠。到我們將程式寫完並到 FPGA 上測試時已經是 Demo 當週星期二 18:30，我們當天將程式灌到 FPGA 後，一切都如預期的一樣不能運作，於是我們一直嘗試用肉眼抓 bug，但屢屢失敗。因為之前亦有聽說直接用 LCD debug 的方法，所以一開始我們是覺得用傳統 debug 的方法就夠了，但後來發現難度奇高，所以在可能 23:00 左右吧，我才決定去詢問別組的做法，從第四組那裡我了解到寫 Initializer testbench 的方法竟然比想像中簡單，因此著手製作所需的 testbench，直到 Demo 前一天的早上才確定改好。

之後我們遇到更棘手的問題：原本早上灌進去能動的程式上完三堂課後回來變成不能動了，當時已經是 Demo 前一晚的 18:00 了，我們簡直崩潰至極。因此我們決定吃完晚餐再回來面對壓力，之後經過 1~2 小時一頓神奇的操作後，突然一切就都恢復正常了，接著我們一步步解 bug，最後在 22:30 左右終於讓之前寫的 code 都能一起正常運作。

若從現在重新看這兩天不眠不休(是真的通宵)，會覺得當面對像 FPGA 無預警死機、無法解決的狀況時，心理素質非常重要，雖然我們三個人嘴上都喊著放棄，但實際上手仍然是動個不停，最後才能在午夜前完成。我想若能度過數電實驗，應該會有非常大的成長吧。同時也希望 Demo 時 FPGA 不要再死機。