

DCLAB final report

b08901091顏柏聖、b08901185曾啟睿、b08602049黃奕榜

1. File structure

- | -DS2-115

- | -mixer.mif

- | | -VGA_Audio_PLL

- | | -I2C_AV_Config

- | | -screen

- | | -screen_up

- | | -screen_pic

- | | -screen_down

- | | -ps2_keyboard

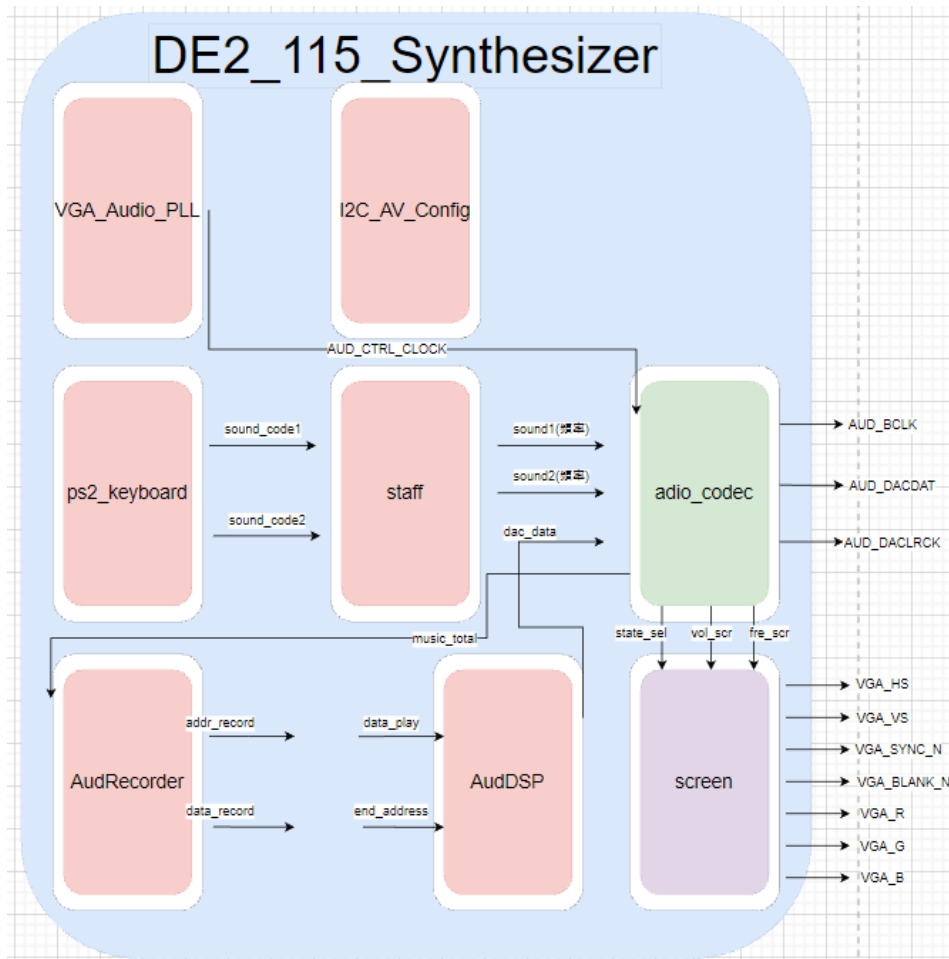
- | | -staff

- | | -adio_codec - wave_gen

- | | -AudRecorder

- | | -AudDSP

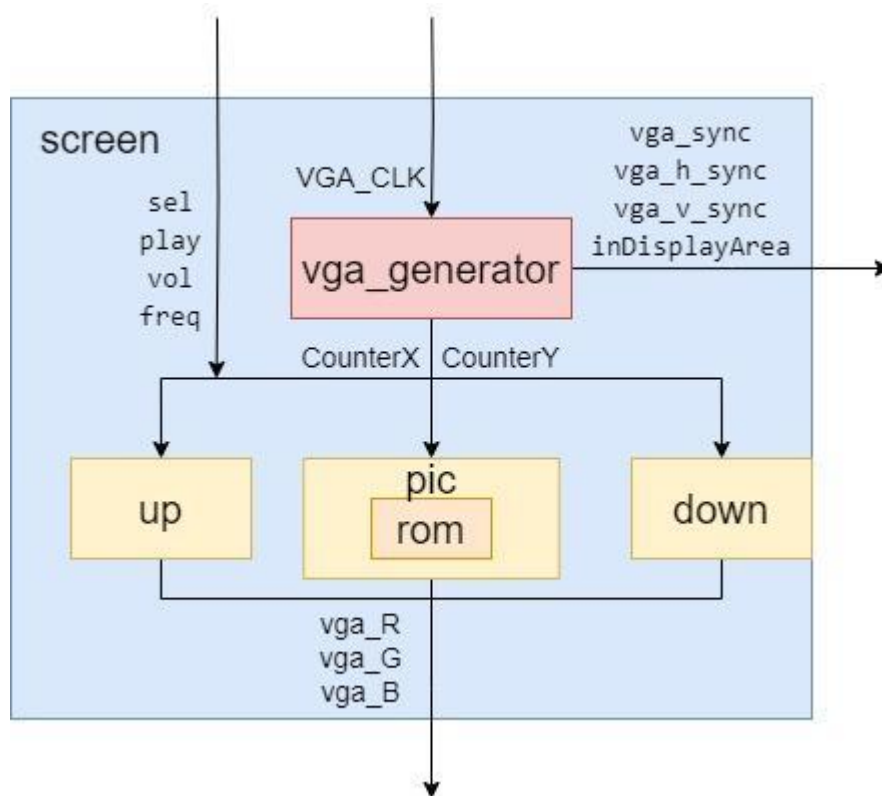
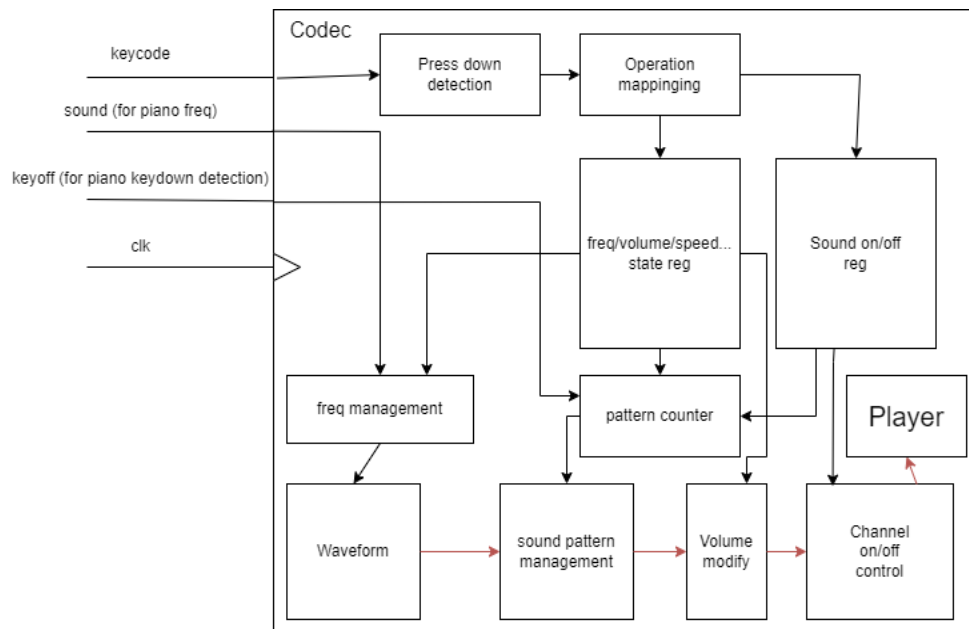
2. System structure



keyboard_ps2: 把PS2協議掃描出的keycode輸出

staff: 將輸入的keycode(鋼琴鍵盤)decode成各自對應的頻率後輸出，也輸出key是否為按下的狀態

codec: 將輸入的鍵盤狀態來對應不同的動作，再依此改變裡面state register的頻率、音量... 狀態，以此合出不同的音效送進player，其datapath大致如下



3. Hardware scheduling

sound

在選擇頻道時使用一個register來儲存正在選取哪個頻道，藉由按鍵來改變所選取的頻道，而在混音器中使用的是把每個音效的狀態各自存起來的register，例如:音效1音量，音效1音調，音效2音量...，若偵測到特定的按鍵便改變選取頻道中對應的register內容，比較像FSM的部分應該就是這些。

screen

screen裡面將vga_generator產生的counterx和country給up, down, pic，指定counter的RGB輸出顏色。

vga_generator

我們使用640*480的螢幕解析度，clk為25MHz，根據vga的protocol送出水平同步hs和垂直同步vs和blank，水平和垂直皆分為四個階段disp, fporch, sync, bporch，disp的大小就是水平的640和垂直的480，各自sync的階段hs和vs會分別為0，水平方向的運作會跟著VGA_clk，垂直方向則會跟著hs，blank只有在display的階段是1，水平和垂直的blank皆為1時，輸出的blank為1，水平的blank為1時，counterx會根據自己的clk增加，country同樣。

screen_up

螢幕上半部分輸出音量的波型，分為20個間隔，每次移動2到20的間隔就會往前，第20個間隔是新的輸出，用顏色區分不同的頻率，這裡使用另一個register儲存音量和頻率，如果頻率是最低的，接下來的8個間隔都會是同樣的音量和顏色，如果頻率是最高的，接下來只會有1個間隔代表現在的音量和顏色。

screen_pic

把bmp圖片轉換成rom可以存放的mif，圖片和VGA同樣都是24色的，所以可以直接透過matlab提取rgb，將圖片從rom取出後，輸出在畫面中央。

screen_down

螢幕下半部分根據音量的大小和是否播放來顯示，顏色在每個clk隨機選取。

4. Fitter summary

Fitter Summary	
Fitter Status	Successful - Sat Jun 17 21:55:23 2023
Quartus II 64-Bit Version	15.0.0 Build 145 04/22/2015 SJ Full Version
Revision Name	DE2_115_Synthesizer
Top-level Entity Name	DE2_115_Synthesizer
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	20,174 / 114,480 (18 %)
Total combinational functions	19,495 / 114,480 (17 %)
Dedicated logic registers	2,252 / 114,480 (2 %)
Total registers	2252
Total pins	518 / 529 (98 %)
Total virtual pins	0
Total memory bits	1,771,392 / 3,981,312 (44 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	1 / 4 (25 %)

5. Timing analyzer

Slow 1200mV 85C Model Setup Summary			
	Clock	Slack	End Point TNS
1	CLOCK_50	15.509	0.000
2	altera_reserved_tck	41.461	0.000
3	u1 altpll_component auto_generated pll1 clk[1]	44.935	0.000

6. 遇到的問題與解決方法

鍵盤

鍵盤掃描的訊號部分主要是使用DE2-115的sample code，但只能輸出是否正在按住某個按鍵，而若要偵測是否為按下的動作，則要比較這個cycle的scan code與上個cycle的scan code是否不同，而判斷所使用的cycle為bclk還是daclrck則以要控制的動作來決定，若要用來控制counter(以daclrck更新)，就使用daclrck來控制鍵盤判斷；若用來控制音量，就用bclk或daclrck來控制鍵盤判斷都可以。

音效

利用從DE2-115的sample code挖出的波形檔來合成出樂器聲，但每個波型只有單一變化，為了聽起來像樂器，可將不同的頻率波形疊合，形成不同的音色，再控制震幅的變化以形成樂器聲。

為了形成音效的變化，使用counter進行循環，而在任意的時間區段內可編輯音量與頻率變化(用counter代表的時間區段控制)，以達成循環變化的音效。

鋼琴鍵盤

使用不同琴鍵來對應不同的波形取樣頻率與波形來做出不同的音色與音調，若要使音效像真正的鋼琴一般按下之後漸弱，就要再使用counter來控制震幅大小，每有按下的動作就重製counter，使震幅回到最大，隨著counter增加就使振幅漸小，使音效達成逐漸消失的效果；此外也可以藉由調整counter與振幅大小的關係靈敏度來做出拉出長音的鋼琴聲。鼓聲按鍵做法也同理，只是也同時以counter來控制頻率大小以做出上述提到的頻率漸弱。

其他音效調整方法

音量：使振幅依比例改變

音調：調整波形的頻率取樣

速度：將控制變化的counter移動速度加快

音效對齊/小位移：直接更改控制變化的counter裡面的值，使其能夠以自己想要的方式對齊，將每個音效的以上狀態各自存起來，這樣當操作完其他音效後回來時便能直接使用上次操作完的狀態。

混音

基本上以線性疊加即可，但有時太多音量大的波形疊加後會出現爆音/雜音的現象，目前尚未明瞭此現象的原因，因為某些波形特別會發生此現象，有些不會。有試過疊加後用除法降低振幅，但這也容易使音質損毀，而目前的解決方法為讓容易爆音音效的振幅縮小一些，把喇叭調大聲一點，盡量不讓訊號振幅太大。

螢幕

畫面一開始用testbench來測試rgb是否有輸出，但是不容易看出來實際的畫面，所以後來都是直接顯示在螢幕上debug，雖然要compile比較慢，但是可以直接看到問題在哪裡，一開始打算用sdram作為暫存圖片的方法，雖然可以直接使用，但是要理解sdram的整體架構有點複雜，因為檔案不大可以用rom，在轉換檔案的時候有碰到一些問題，但還好參考網路上的方法後順利解決。

7. 心得

黃奕榜：final protject一開始在想題目的時候想了很久，每種硬體都想嘗試看看，但是又擔心會太複雜，過程中遇到許多問題，都想辦法找出了替代的方法，雖然sdram和sd card沒有做出來很可惜，但還是有把基本的部分完成，覺得可以做自己想嘗試的東西很有趣。

顏柏聖：我這次負責的是音效的部分，雖然一開始想說用DE2-115光碟的sample code為模板進行改進，但發現與我們想要的功能還是有很大的距離，所以把整個發音的部分解構後審視每個部分，把還能用的東西進行改進，若需要實作完全新的功能的話就上網找參考資料然後試著自己解決(例如鼓聲與鋼琴聲的部分)，最後把這些部分依照需求重新組裝成我心目中的混音器。除了自己的部分以外，與隊友的溝通也很重要，例如與隊友溝通讓他實作螢幕顯示時可以配合我的混音器狀態進行顯示。可惜的是錄音功能到最後未成功運作，可能是因為SRAM的操作有一些問題。

曾啟睿：因為我們原本希望能播放好幾首較長的歌曲，搭配混音器讓大家都能成為夢想中的DJ，有嘗試將SD卡內所存儲的音樂讀出以進行播放(SDRAM的儲存量128MB太小)，然而卻在實作完成後發現問題出在fifo ip core，SD卡僅能成功送出一小段字節到fifo這個module，而我們從fifo裡面讀出的卻是空的東西，且後面幾個sector也卡住無法繼續將資料送進fifo，由於尚未完成debug的部分，在demo時便沒有展出。