Lab8_1

Design Specification

✓ For a speaker:

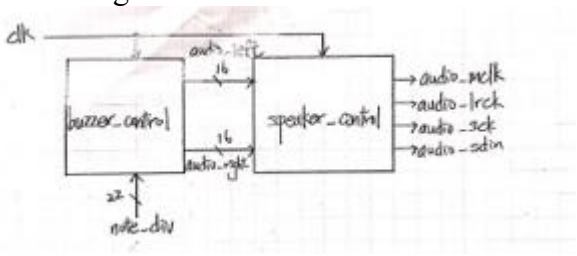
Input:

clk, rst_n

Output:

audio_mclk,
audio_lrck
audio_sck,
audio_sdin

✓ Draw the block diagram of the design.



Design Implementation

✓ 本題由 buzzer_control 與 speaker_control 組成。

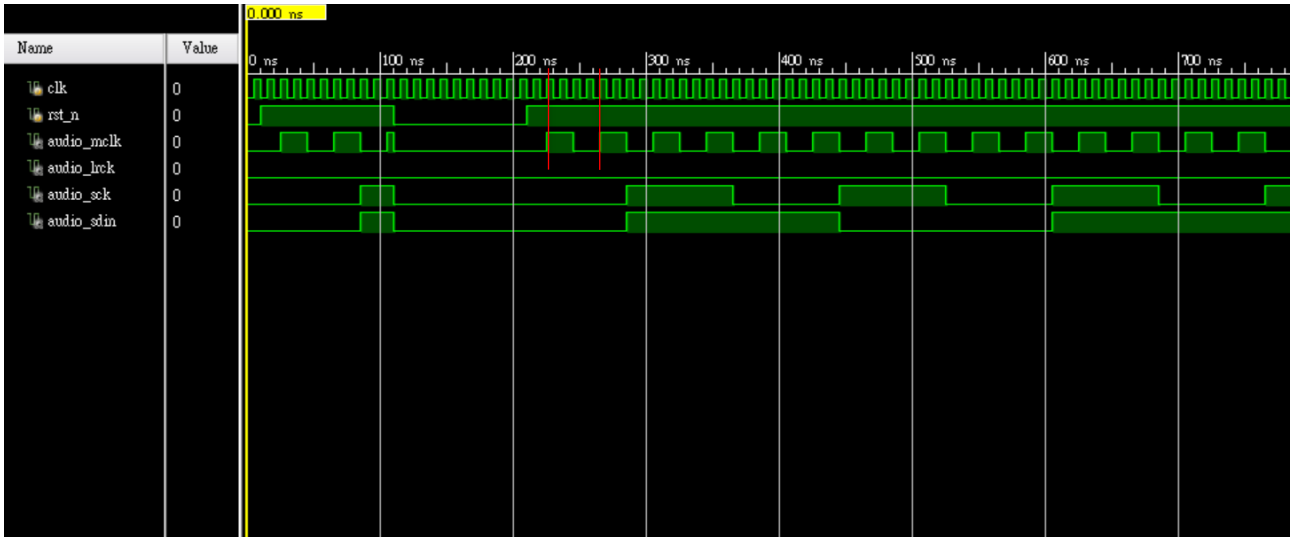
✓ buzzer_control

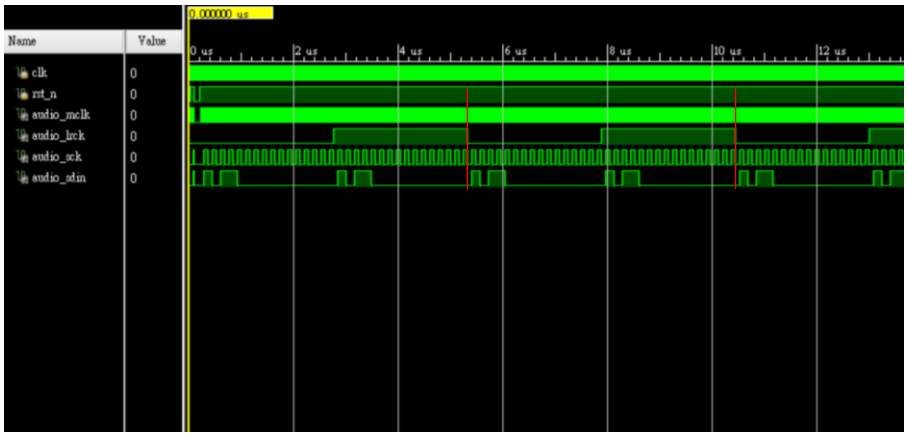
此 module 的功能為控制蜂鳴器的聲音頻率與振幅。由於第一題中的數值為給訂好的，因此其詳細功能將在第二題中說明。

✓ speaker_control

此 module 的功能為將 buzzer_control 輸出的各 16bits 的聲音頻率分為一次輸出 1bit。其中的 audio_mclk, audio_lrck, audio_sck, 皆為用除頻器所得的結果；audio_lrck 是控制 audio_right 與 audio_left 的 clock，audio_sck 則是控制 audio_sdin 的。在這邊為了將 audio_right 與 audio_left 的值給 audio_sck，並一次輸出 1 bit，我利用了 flip flop 來完成此部分。而由於 audio_right 與 audio_left 為各 16 bits 的數值，因此 audio_sck 的頻率為 audio_lrck 的 32 倍。

✓





由圖可看出，audio_mclk 的頻率為 clk 的 1/4 倍；audio_sck 的頻率為 audio_lrck 的 32 倍。

Lab8_2

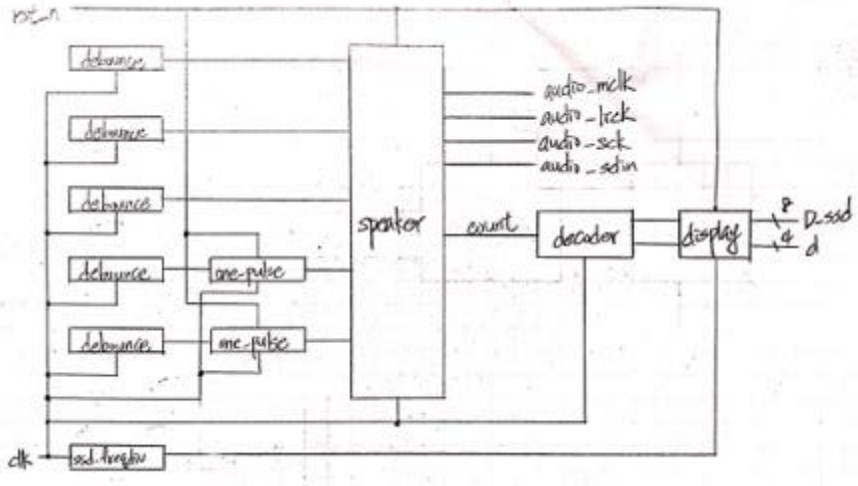
Design Specification

✓ For a speaker:

Input: clk,
rst_n,
left,
center,
right,
up,
down

output:
audio_mclk,
audio_lrck,
audio_sck,
audio_sdin,
[7:0]D_ssd,
[3:0]d

✓ Draw the block diagram of the design.



Design Implementation

✓ 第二題為第一題的實作，在第一題中 buzzer_control 的頻率與振幅是給定值，在第二題中的音頻則分別是 do、re、mi，分別對應到左中右按紐；振幅(音量)則是由上下按鈕控制。在這裡我做了一個 MUX 來選擇輸出的頻率為哪個音。振幅則是將其做成 16 段的調節器，每當 up = 1，down = 0 時音量則加 1，反之則減 1。由於若只將音量的數值加一並沒有明顯區別，因此將其乘上 1280(任一較大的變數)，才可以聽出其音量的改變。另外，為了讓它可

以像鋼琴一樣，因此沒有在控制頻率的按鈕加 one_pulse。

- 音頻

left	center	right	note_div
1	0	0	22'd151699
0	1	0	22'd170241
0	0	1	22'd191131
default			0

✓ I/O pin

I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
VOC	W7	W6	U8	V8	U5	V5	U7	V7
I/O	d[3]	d[2]	d[1]	d[0]	audio_mclk	audio_lrck	audio_sck	audio_sdin
VOC	W4	V4	U4	U2	A14	A16	B15	B16
I/O	rst_n	clk	left	center	right	up	down	
VOC	V17	W5	T17	U18	W19	T18	U17	

Discussion

在 speaker_control 中，我原本想用 ring counter 讓 audio_sdin 可以一次輸出一個 bit，但卻發現這樣會出問題，程式也會變得很冗長。所以我改用一般的 flip flop，並設一個變數去控制輸出的值為 audio_left 與 audio_right 合起來的第幾個 bit。

在控制 do、re、mi 時，原本的控制是當一個按鈕按下後便會發出聲音，沒有排除其他可能(例如兩個按鈕一起按)，所以會同時聽到兩個以上的音。為避免這種狀況，我把 MUX 改為只有一個按鈕按下時才會有聲音。

Conclusion

這次的實驗運用到了新的概念，一開始會有點不瞭解為什麼 buzzer_control 的結構，但經由第二題自己改過參數後就會比較了解他的原理了。