106010006 電機 21 黃詩瑜

Lab 8: Speaker

1. Please design an audio-data parallel-to-serial module to generate the speaker control signal with 100MHz system clock, 25 MHz master clock, (25/128) MHz Left-Right clock (Fs), and 6.25 MHz (32Fs) sampling clock.

1.1 Design a general frequency divider to generate the required frequencies for speaker clock.

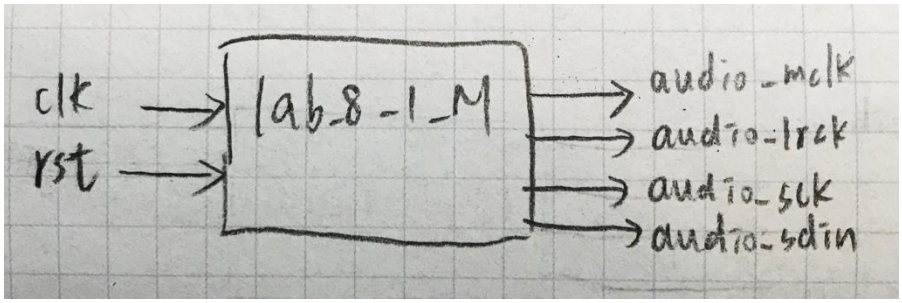
1.2 Design a stereo signal parallel-to-serial processor to generate the speaker control signals.

Please use Verilog simulation waveform to verify your control signal.

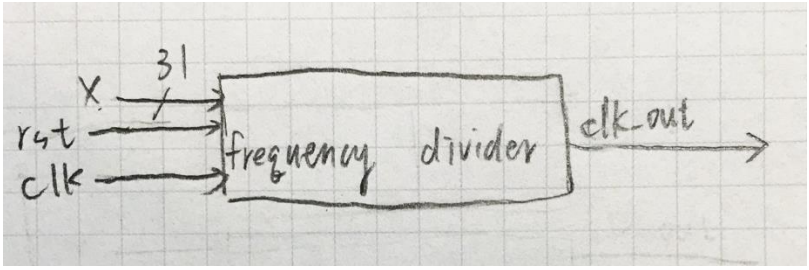
Specification:

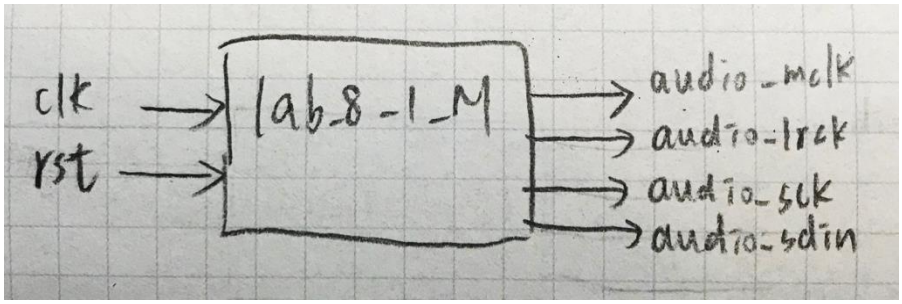
Input: clk, rst

Output: audio_mclk, audio_lrck, audio_sck, audio_sdin

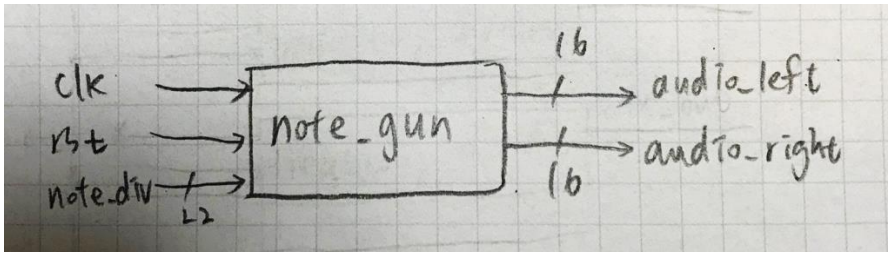


Implementation:





第一題需要做出 25 M, (25/128) M, 6.25 MHz 的 clock，就利用 frequency divider 除出相對應的頻率後，再將 clk_out 接給 lab_8_1_M 的 audio_mclk, audio_lrck, audio_sck 這 3 個 outputs。

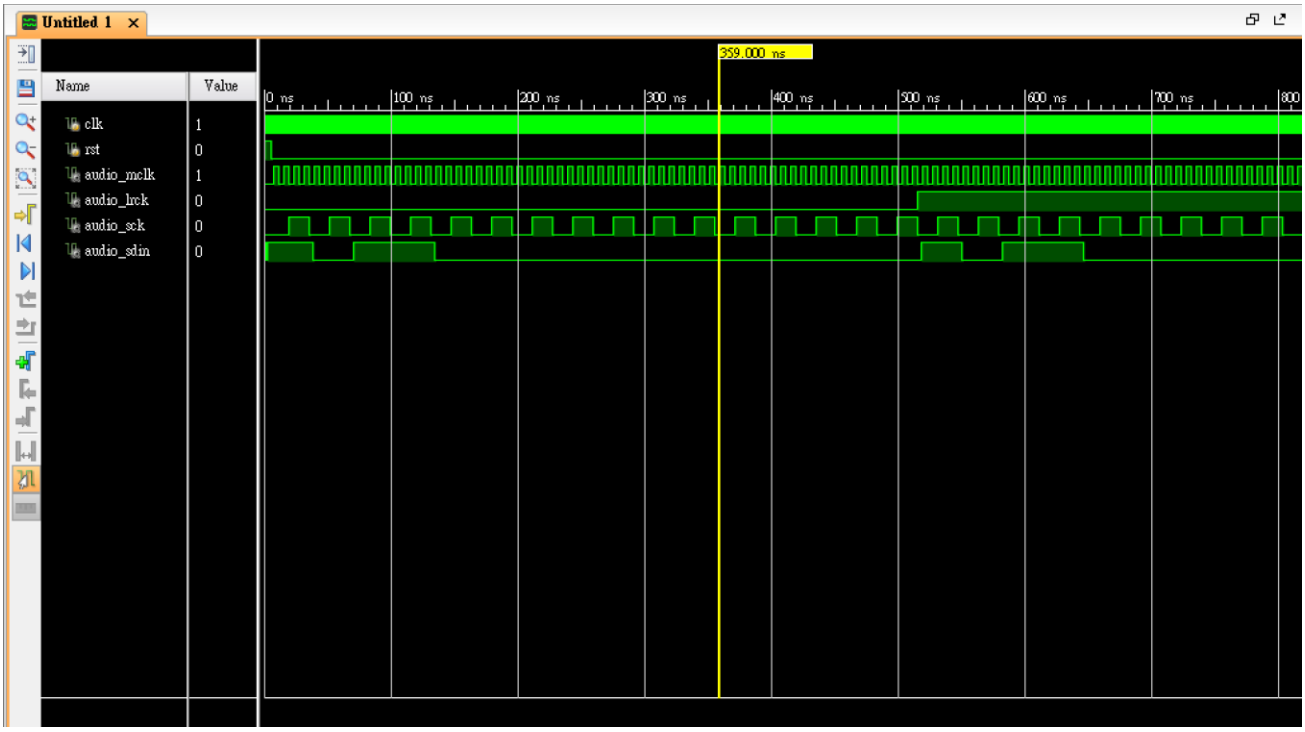


note_gun 這個 module 跟講義上的一樣，把 input note_div 設為 22'd191571，就會產生對應的 output audio_left, audio_right。

要將 audio_left, audio_right 一個一個 bit 的給 audio_sdin，

當 audio_lrck 為 0 時，先傳 audio_right，當 audio_lrck 為 1 時，再傳 audio_left，audio_left 和 audio_right 都是 16 個 bits，所以設一個 4-bit audio_s，它變化的頻率是 audio_lrck 的 16 倍，當 audio_s 由 0 數到 15 時，將 16-bit audio_right/audio_left 傳給 audio_sdin，這樣就設計完成了。

波形圖：



討論：

這個實驗並不難，note_gun 的部分只要複製講義上的，並瞭解裡面的設計，再將 parallel-to-serial processor 設計出來就行了。

2. Speaker control

2.1 Please produce the buzzer sounds of Do, Re, and Mi by pressing buttons (Left, Center, Right) respectively. When you press down the button, the speaker produces corresponding frequency sound. When you release the switch, the speaker stops the sound.

2.2 Please control the volume of the sound by pressing button (Up) as increase and (Down) and decrease the volume. Please also quantize the audio dynamic range as 16 levels and show the current sound level in the 7-segment display.

Specification:

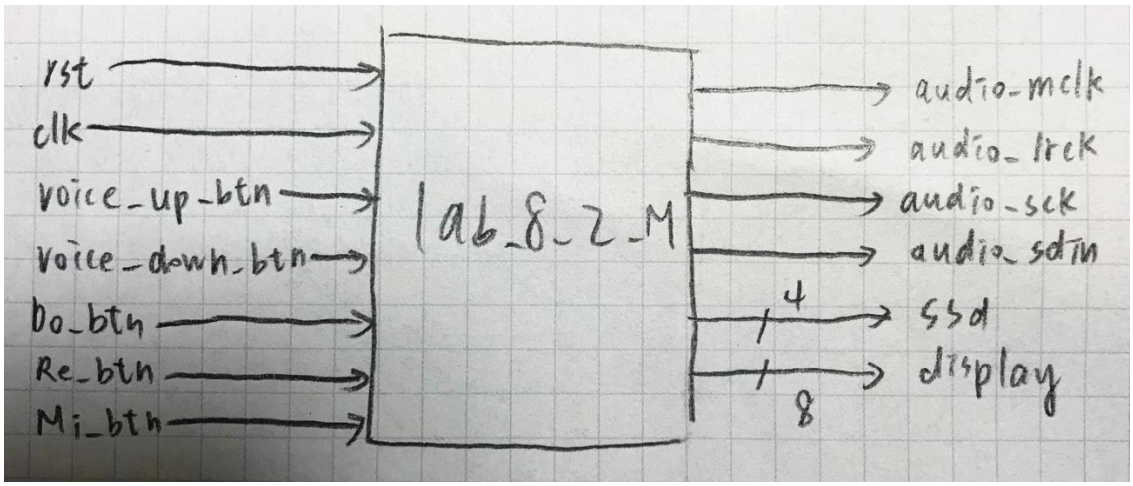
Input: clk, rst, voice_up_btn, voice_down_btn, Do_btn, Re_btn, Mi_btn

Output: audio_mclk, audio_lrck, audio_sck, audio_sdin, display[7:0], ssd[3:0]

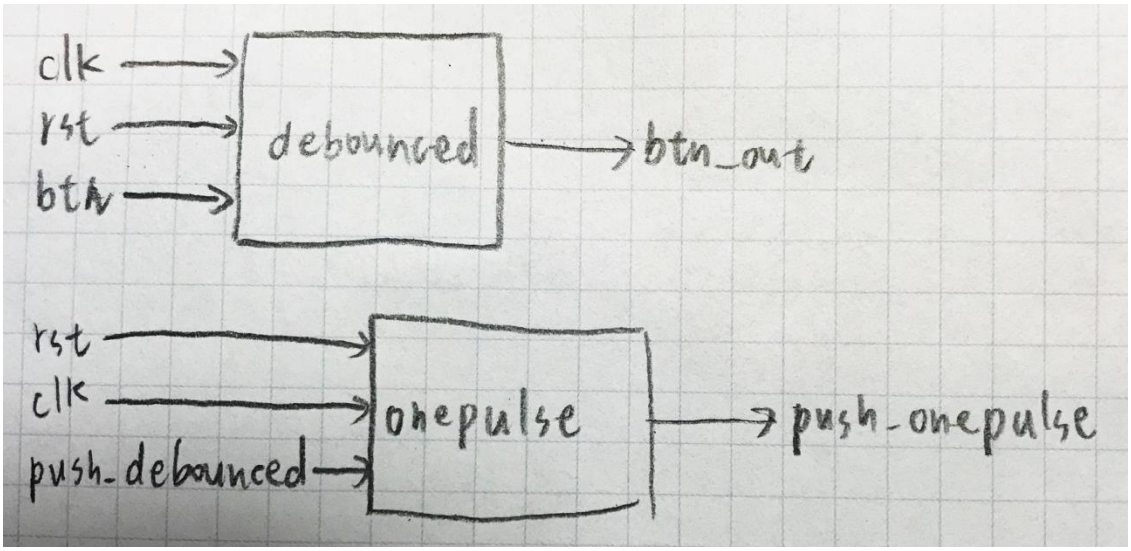
I/O	clk	rst	voice_up_btn	voice_down_btn	Do_btn	Re_btn	Mi_btn
LOC	W5	V17	T18	U17	W19	U18	T17

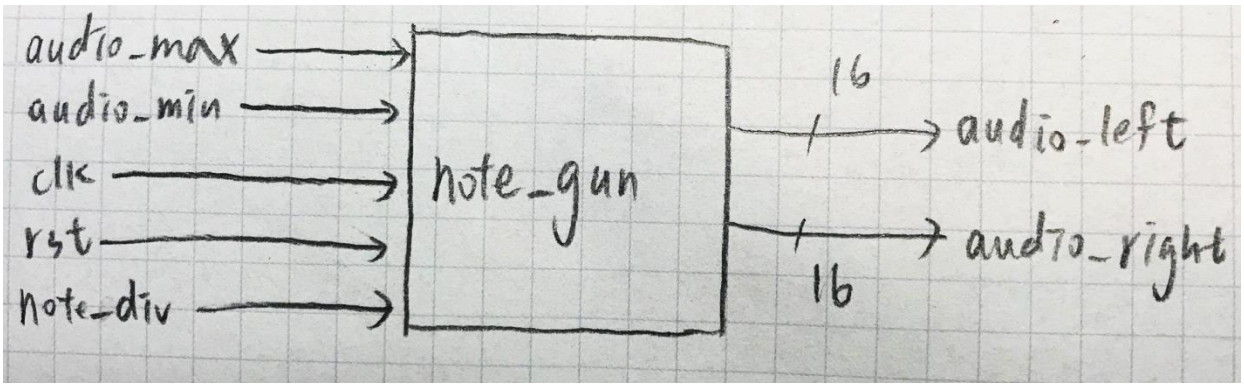
audio_mclk	audio_lrck	audio_sck	audio_sdin	ssd [0]	ssd [1]	ssd [2]	ssd [3]
A14	A16	B15	B16	U2	U4	V4	W4

display[0]	display[1]	display[2]	display[3]	display[4]	display[5]	display[6]	display [7]
V7	U7	V5	U5	V8	U8	W6	W7



Implementation:





第二題和第一題的差別主要是在設計按鈕控制的部分，先讓按鈕經過 debounce 和 one pulse 的處理，選擇 Do/Re/Mi 的設計是用多功器，當哪個按鈕按下就將 note_gun 的 input note_div 設定成對應的 22'd191571/22'd170648/ 22'd151515，note_gun 多了兩個 16-bit inputs audio_max, audio_min，設計一個 flip-flop，一開始 audio_min_next 是 16'h5FFF，audio_max_next 是 16'hB000，還有一個 digit_next 是 4'd15，當按下 voice up button 且 digit < 4'b1111 時，將 audio_min 減 16'd1280，digit 加 4'b0001，voice down button 按下時，將 audio_min 加 16'd1280，digit 減 4'b0001，這樣調整聲音大小的部分就設計完成，再把 digit 的值顯示在 7-seg LED 燈上就行了。

討論：

設計第二題的時候沒有遇到什麼問題，很順利，只要知道 note_gun 裡面如何控制聲音的大小，就沒問題了。

結論：

這次實驗跟以前比起來真的比較簡單，很快就完成了，只是我的按鈕訊號都有經過 debounce 和 one pulse，助教在 demo 的時候說按鈕有點不穩定，我也不清楚為什麼。