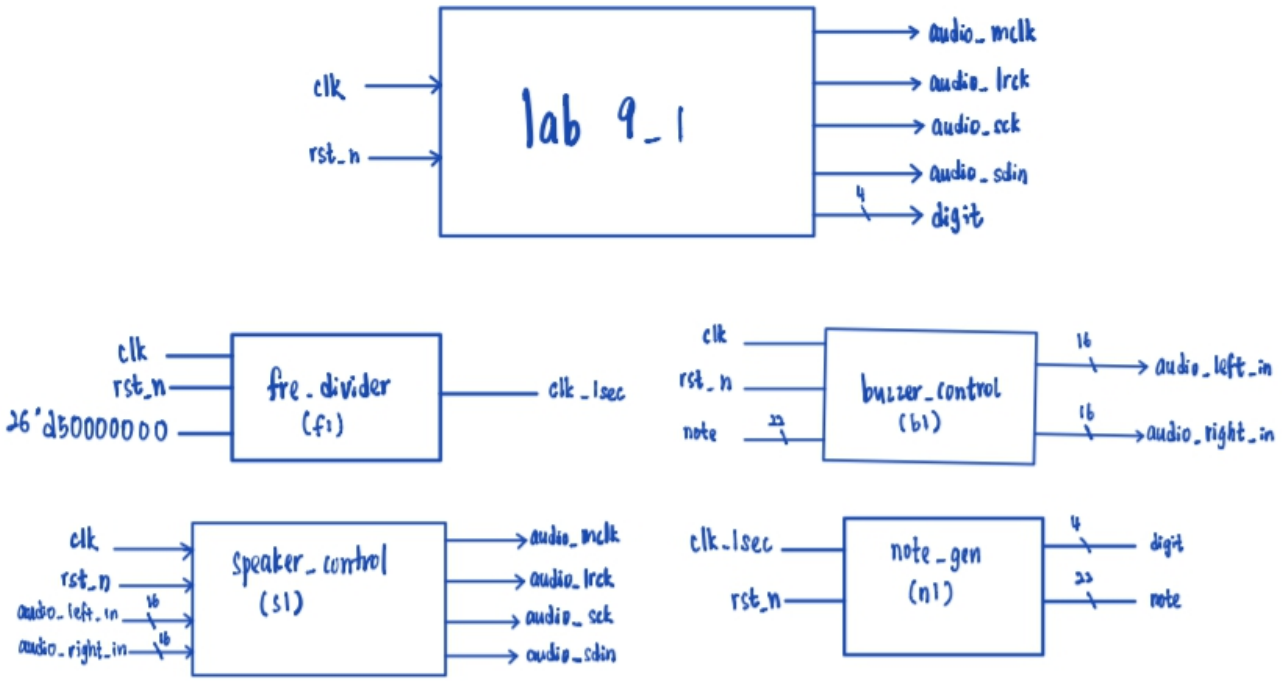
# Lab 9 Report

## Design Specification

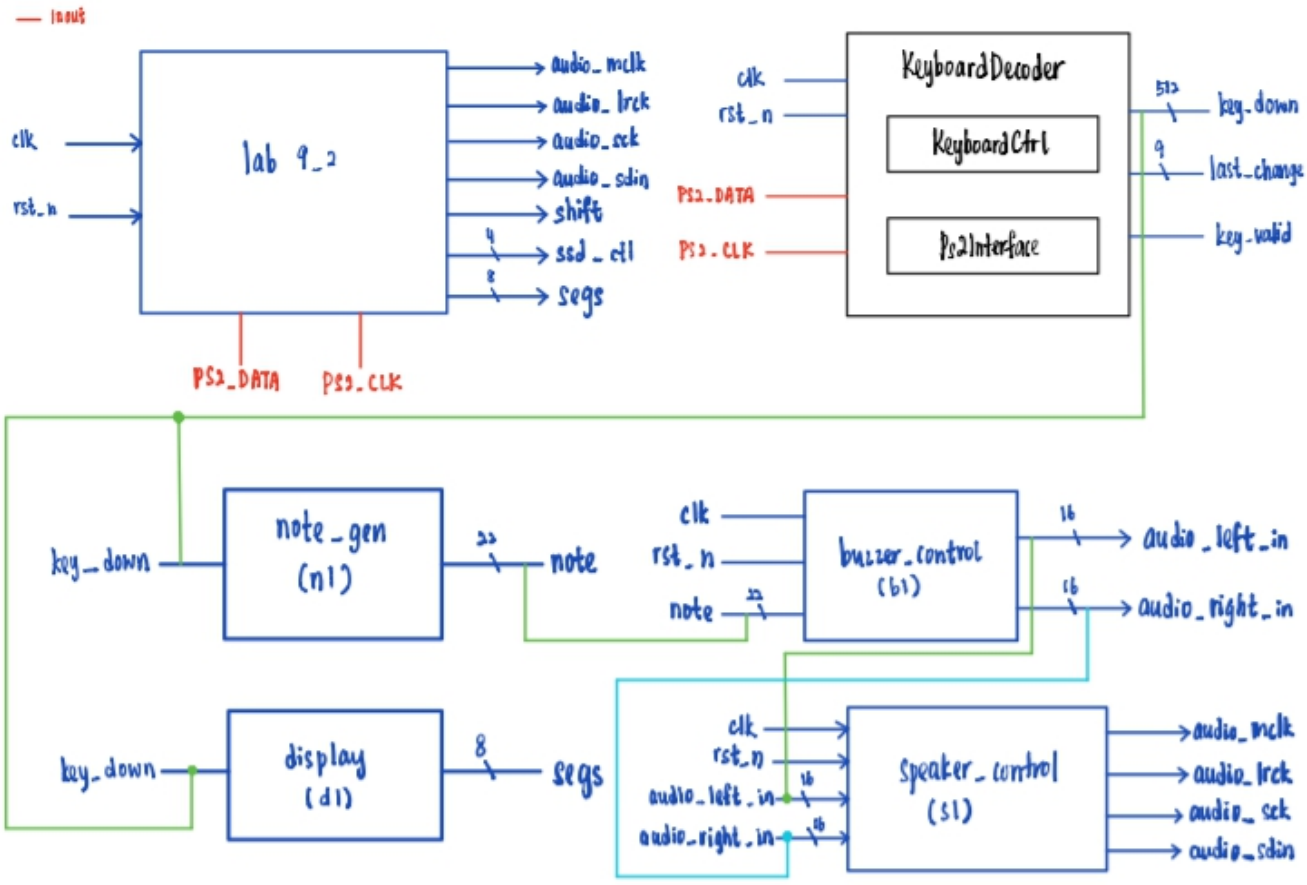
### Lab9_1:

Input is clk and rst_n.
Output is : audio_mclk, audio_lrck, audio_sck and audio_sdin, which are use for the sound, the speaker.
The other output is a 4-bits digit, which act as the input to the led, use for telling the user of the tone is changing now.



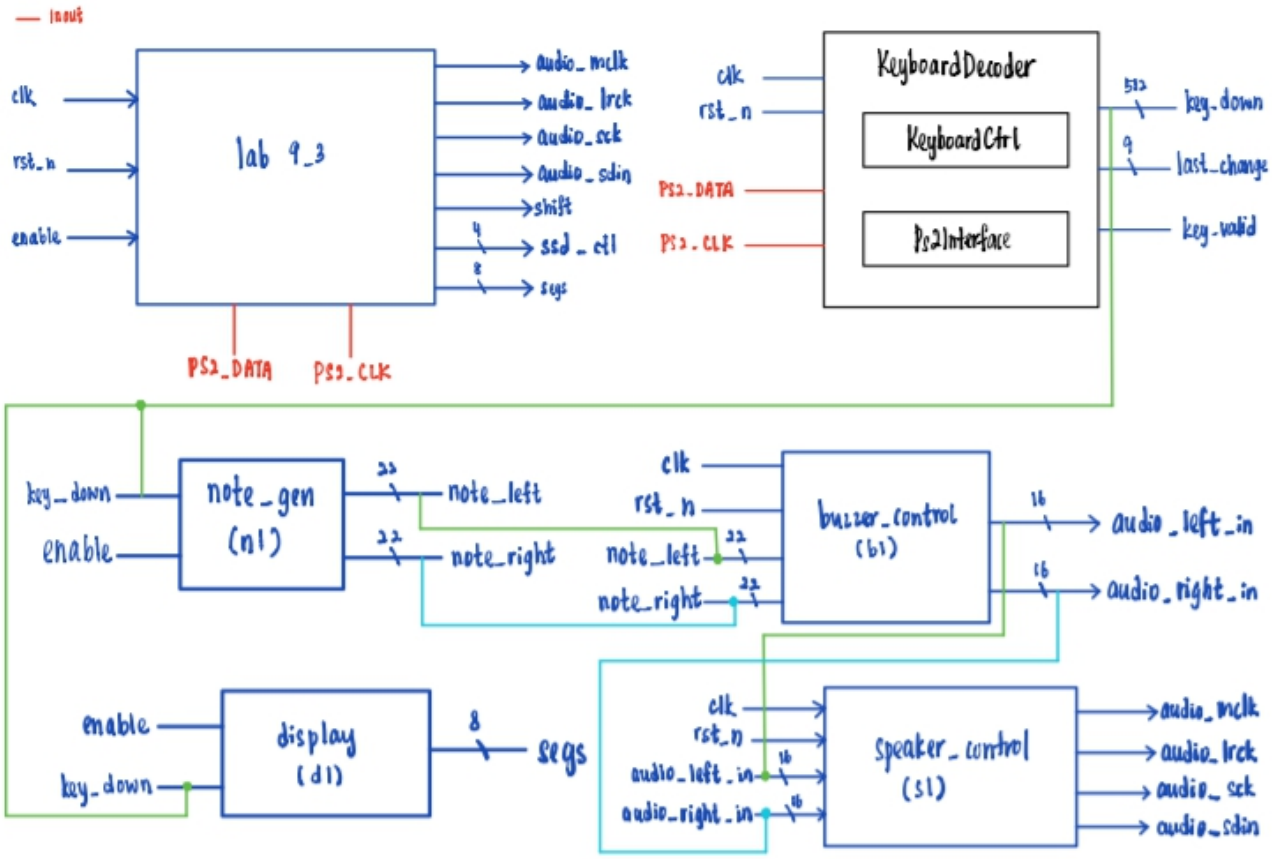
## Lab9_2:

input is clk and rst_n.
inout is PS2_DATA, PS2_CLK, which is use for the keyboard.
output is audio_sdin, audio_mclk, audio_lrck, audio_sck, shift, which are use for the speaker.
The next output is a 4-bits ssd_ctl, which is use for the seven segment decoder.
The other output is a 8-bits segs, which is used to light up the related segments on the seven segment decoder.
The last output is a 1-bit shift, which is a led used to show that the shift button on the keyboard is pressed, or let's say the higher frequency of the tone is played.



## Lab9_3:

input is clk and rst_n.
The other input is enable, which is function as the switch for the double tone(I called mixer).
inout is PS2_DATA, PS2_CLK, used for keyboard.
output is audio_sdin, audio_mclk, audio_lrck, audio_sck, shift, which are use for the speaker.
The next output is a 4-bits ssd_ctl, which is use for the seven segment decoder.
The other output is a 8-bits segs, which is used to light up the related segments on the seven segment decoder.
The last output is a 1-bit shift, which is a led used to show that the shift button on the keyboard is pressed, or let's say the higher frequency of the tone is played.



## Design Implementation

The module KeyboardDecoder used in all the lab was provided by the professor

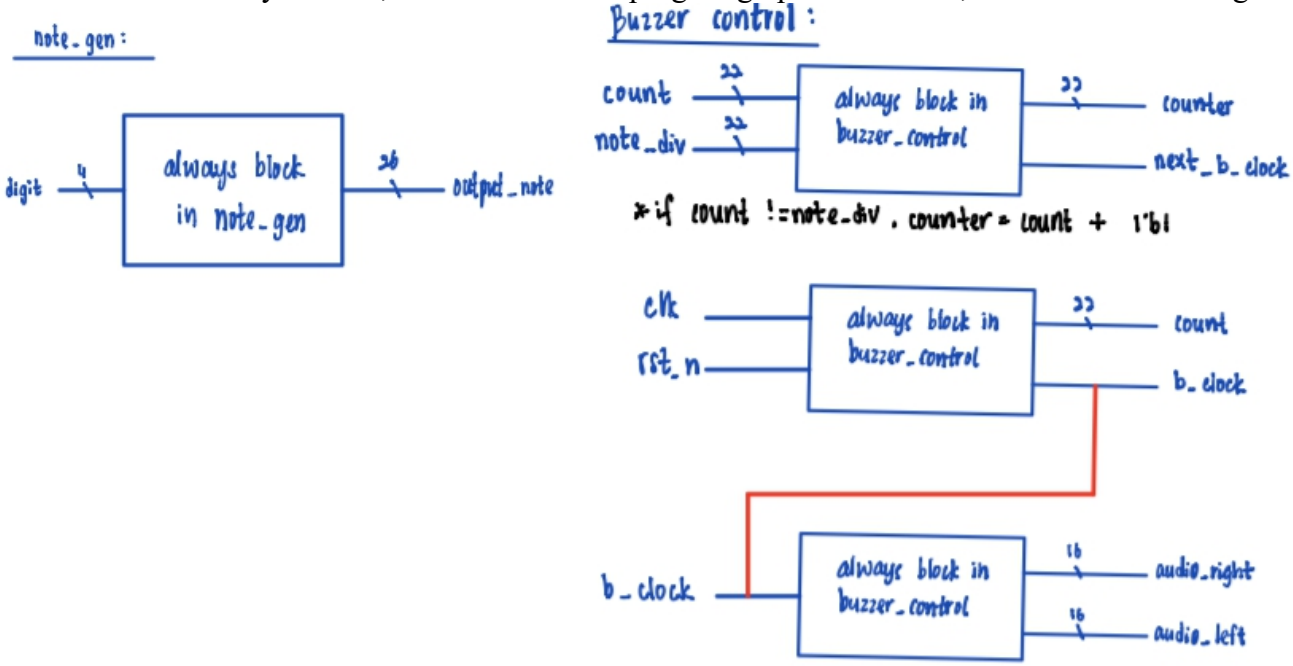
## Lab9_1:

The frequency divider used included the output of 1 second frequency divider, and the output of the different frequency to use for the speaker, which is the audio_mclk(Master Clock) synchronizes the audio data transmission, which is 25MHz. The LRCK (Left-Right Clock, or Word Select (WS) Clock, or Sample Rate (Fs) Clock) controls the sequence (left or right) of the serial stereo output, which is 25MHz/128, nearly 192kHz. The Serial Clock (SCK) controls the shifting of data into the input data buffers, which is 25MHz/4, 6.25MHz.

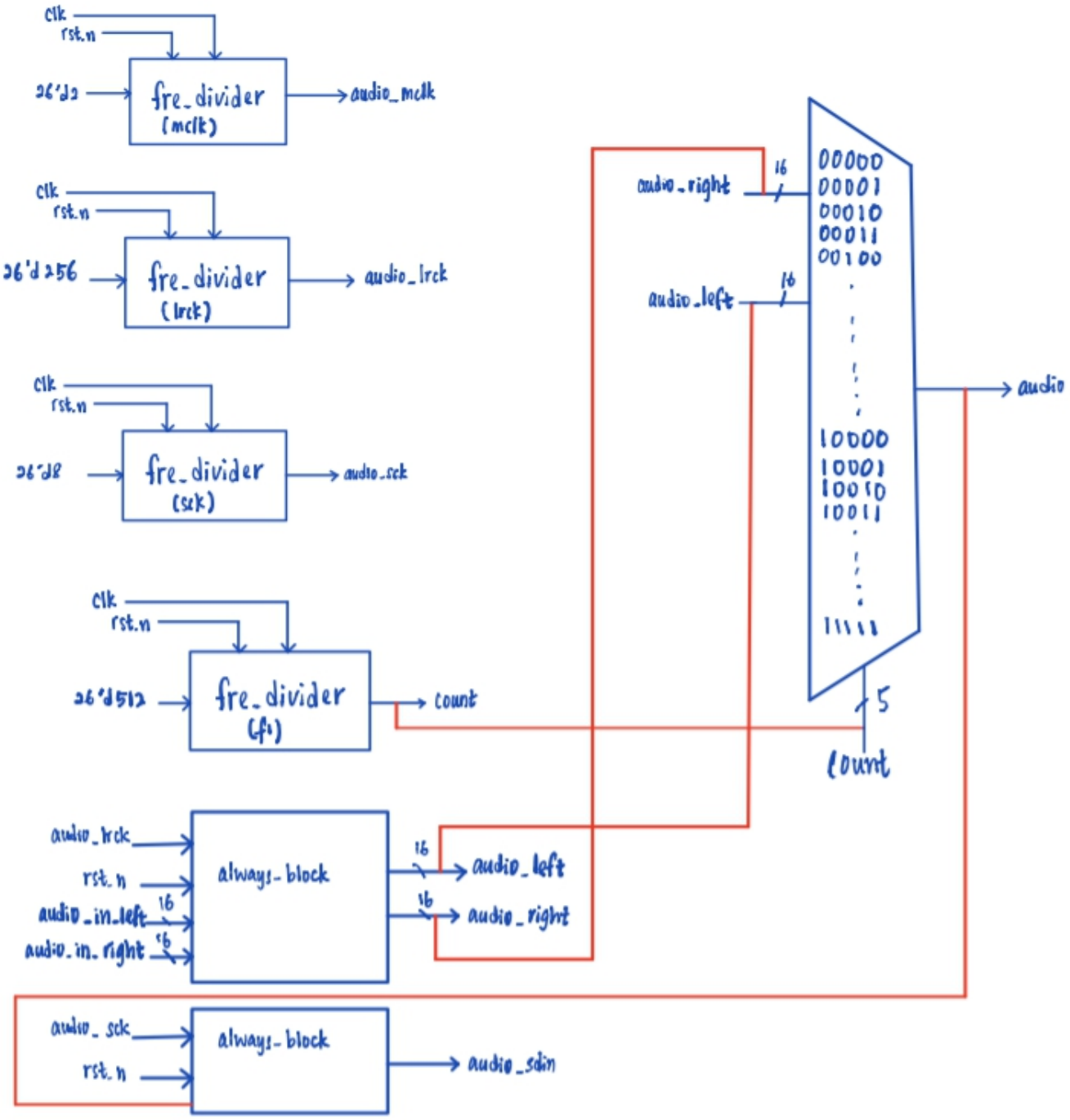
The buzzer_control is used to generate the input for speaker control, which are audio_left_in and audio_right_in, which then use to generate the sound for left and right ear. The note_gen is function as determine the frequency, either the middle or the higher frequency, of different tone, the Do, Re, Mi, Fa, So, La and Si.

There are 4 frequency divider used in speaker_control, as to generate the four different output, which are audio_mclk, audio_lrck, audio_sck, audio_sdin. The audio_mclk is generate by the 26'd2 counter, audio_lrck is generate by a 26'd256 counter, the audio_sck, is generate by a 26'd8 counter while the audio_sdin is generate by the parallel to serial converter. The forth frequency divider uses 26'd512 counter to generate a count(output) to act as the select for the mux of the parallel to serial converter. The mux will generate the serial output called audio, while the audio_left(audio_in_left) and the audio_right(audio_in_right) will act as the input to the mux. The audio output is then pass to the audio_sdin.

There are no specific function for the led, just to show that the tone is really changing now. Because there are only 14 tone, so the led will keep lighting up from 0 to 13, and then back to 0 again.



Speaker control:



## Lab9_2:

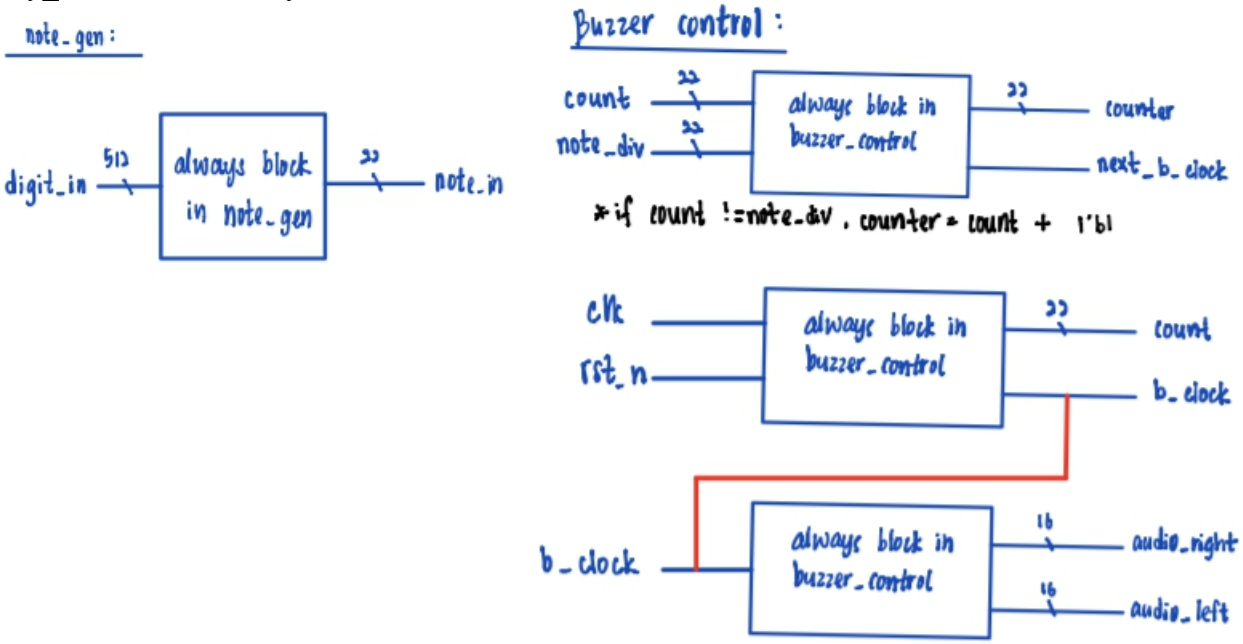
The output of the different frequency to use for the speaker, which is the audio_mclk(Master Clock) synchronizes the audio data transmission, which is 25MHz. The LRCK (Left-Right Clock, or Word Select (WS) Clock, or Sample Rate (Fs) Clock) controls the sequence (left or right) of the serial stereo output, which is 25MHz/128, nearly 192kHz. The Serial Clock (SCK) controls the shifting of data into the input data buffers, which is 25MHz/4, 6.25MHz.

The buzzer_control is used to generate the input for speaker control, which are audio_left_in and audio_right_in, which then use to generate the sound for left and right ear.

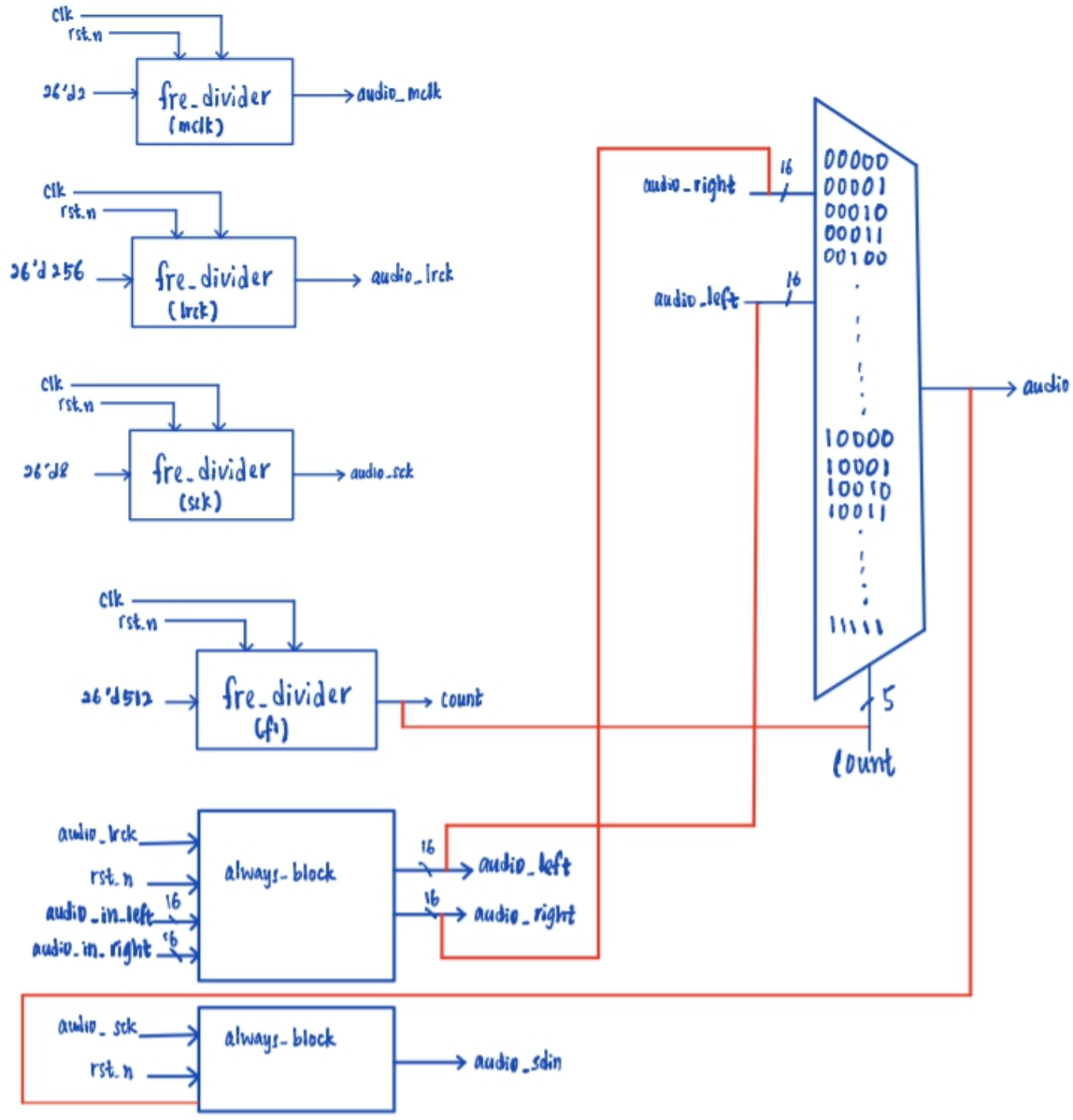
The note_gen is function as determine the frequency, either the middle or the higher frequency, according to the shift from the keyboard is pressed or not. If the shift is pressed, the higher frequency will be played according to the different key inserted(C, D, E, F, G, A, B), if shift is not pressed, the middle frequency will be played according to the different key inserted. The different tone, Do, Re, Mi, Fa, So, La and Si will be played according to the C, D, E, F, G, A and B.

There are 4 frequency divider used in speaker_control, as to generate the four different output, which are audio_mclk, audio_lrck, audio_sck, audio_sdin. The audio_mclk is generate by the 26'd2 counter, audio_lrck is generate by a 26'd256 counter, the audio_sck, is generate by a 26'd8 counter while the audio_sdin is generate by the parallel to serial converter. The forth frequency divider uses 26'd512 counter to generate a count(output) to act as the select for the mux of the parallel to serial converter. The mux will generate the serial output called audio, while the audio_left(audio_in_left) and the audio_right(audio_in_right) will act as the input to the mux. The audio output is then pass to the audio_sdin.

The ssd_ctl is used to determine which of the seven segment decoder to be used. The display will decide which segments are going to be used to display the digit required according to the input, the 512-bits of key_down from the keyboard decoder .



Speaker control:



## Lab9_3:

The output of the different frequency to use for the speaker, which is the audio_mclk(Master Clock) synchronizes the audio data transmission, which is 25MHz. The LRCK (Left-Right Clock, or Word Select (WS) Clock, or Sample Rate (Fs) Clock) controls the sequence (left or right) of the serial stereo output, which is 25MHz/128, nearly 192kHz. The Serial Clock (SCK) controls the shifting of data into the input data buffers, which is 25MHz/4, 6.25MHz.

The buzzer_control is used to generate the input for speaker control, which are audio_left_in and audio_right_in, which then use to generate the sound for left and right ear. Since, there are mixer involved in this lab, the buzzer_control was then designed as separate the counter for the left and the right of the sound, and then use different b_clk(b_clock), which are b_clk_left and b_clk_right. Then the output of audio_left_in and audio_right_in is then sent to the speaker control.

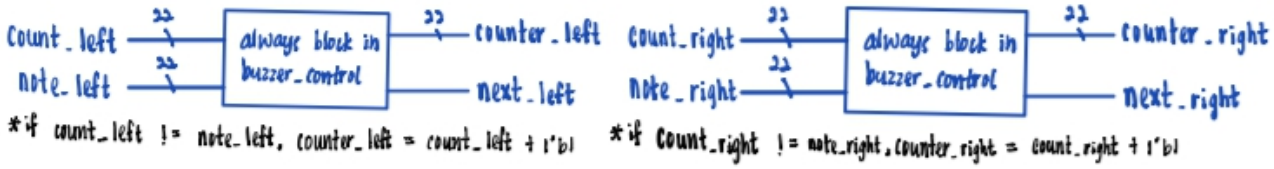
The note_gen is function as determine the frequency, either the middle or the higher frequency, according to the shift from the keyboard is pressed or not. If the shift is pressed, the higher frequency will be played according to the different key inserted(C, D, E, F, G, A, B), if shift is not pressed, the middle frequency will be played according to the different key inserted. The different tone, Do, Re, Mi, Fa, So, La and Si will be played according to the C, D, E, F, G, A and B.

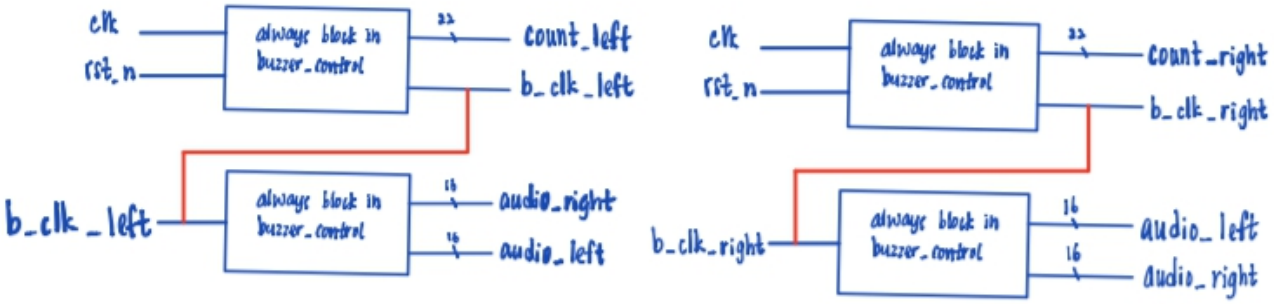
There are 4 frequency divider used in speaker_control, as to generate the four different output, which are audio_mclk, audio_lrck, audio_sck, audio_sdin. The audio_mclk is generate by the 26’d2 counter, audio_lrck is generate by a 26’d256 counter, the audio_sck, is generate by a 26’d8 counter while the audio_sdin is generate by the parallel to serial converter. The forth frequency divider uses 26’d512 counter to generate a count(output) to act as the select for the mux of the parallel to serial converter. The mux will generate the serial output called audio, while the audio_left(audio_in_left) and the audio_right(audio_in_right) will act as the input to the mux. The audio output is then pass to the audio_sdin.

The ssd_ctl is used to determine which of the seven segment decoder to be used. The display will decide which segments are going to be used to display the digit required according to the input, the 512-bits of key_down from the keyboard decoder .

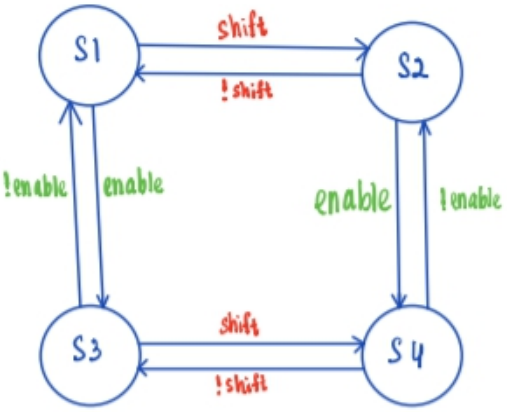
## Buzzer control :



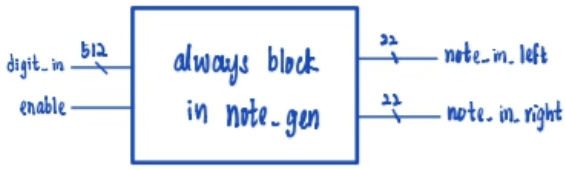



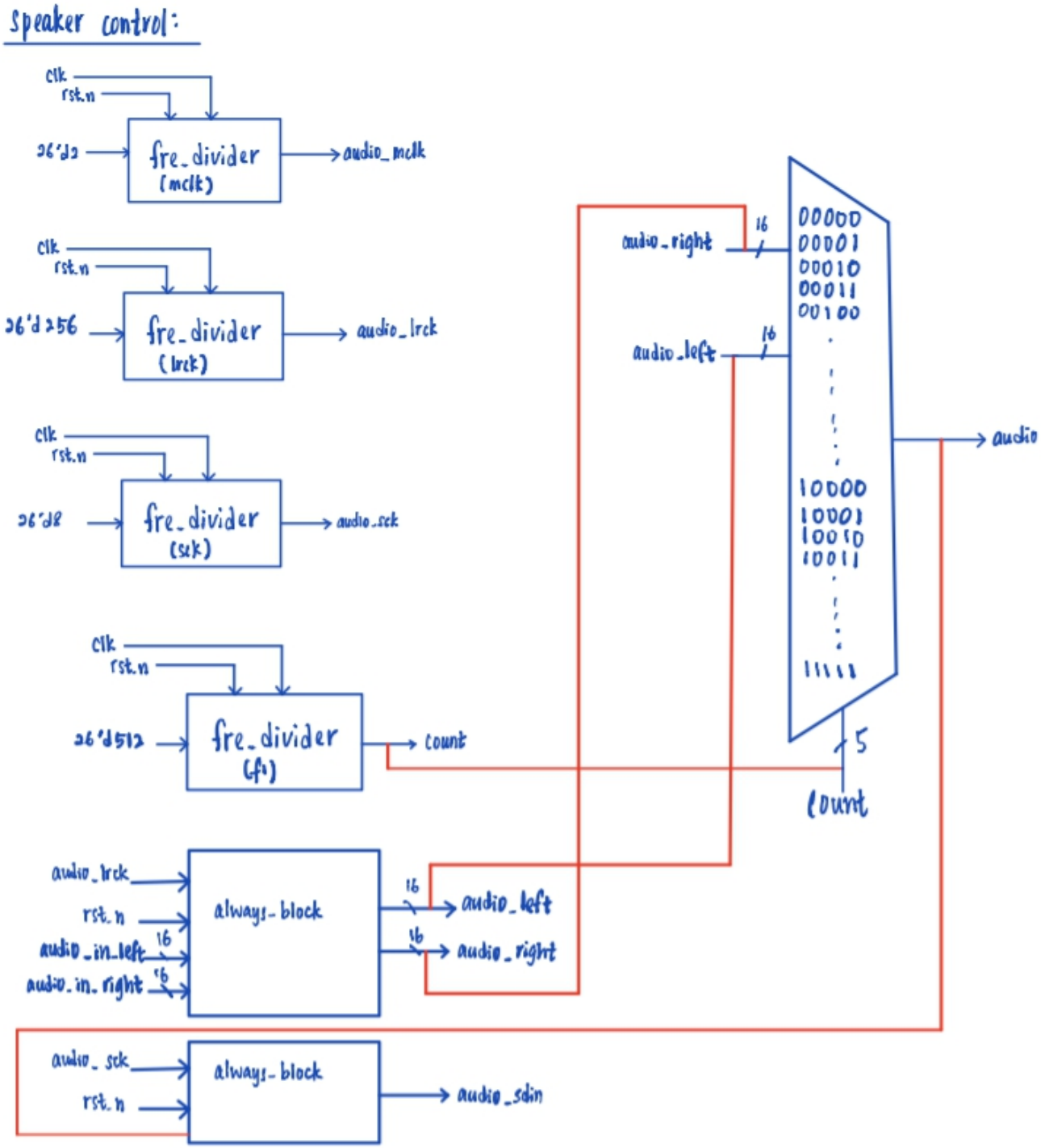

S1: mid
S2: higher
S3: mid mix
S4: higher mix

digit_in[9'h012] = shift
switch for mixer is on = enable




## note_gen :






## Discussion

The things that I did wrong was forgot to reset the counter in the buzzer_control to 0, and this cause a popping sound in my lab9_2, when demo the lab. But after that I did fixed it.

## Conclusion

From this lab, I had learned how to generate different tune of do to Si, and with different frequency, which is middle and higher frequency. The other things that I had learned is that how to implement the double tone from lab9_3, it is really fun.