邏輯設計實驗 Lab09 結報

104060012 邱怡庭

1 Please design an audio-data parallel-to-serial module to generate the speaker control signal with 40MHz system clock, 5MHz bit clock and (5/32) MHz stereo 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.

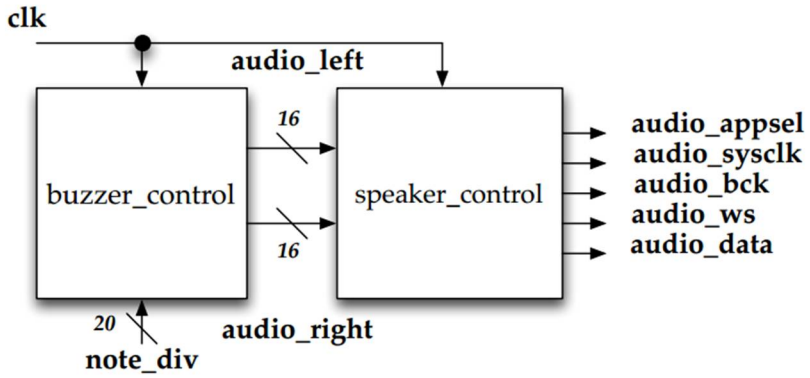
Design Specification

output : audio_appsel, // playing mode selection
audio_sysclk, // control clock for DAC (from crystal)
audio_bck, // bit clock of audio data (5MHz)
audio_ws, // left/right parallel to serial control
audio_data

input : clk, // clock from crystal
rst_n, // active low reset

wire : [15:0] audio_in_left, audio_in_right;

block diagram:

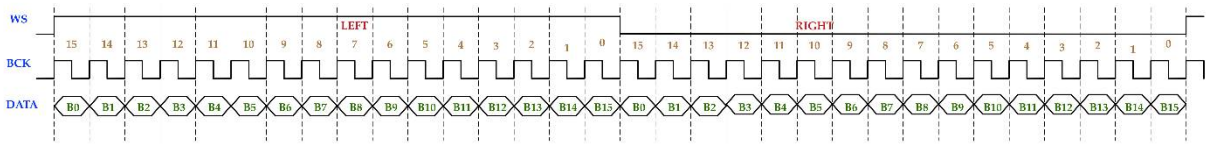


Design Implementation

logic function / logic diagram:

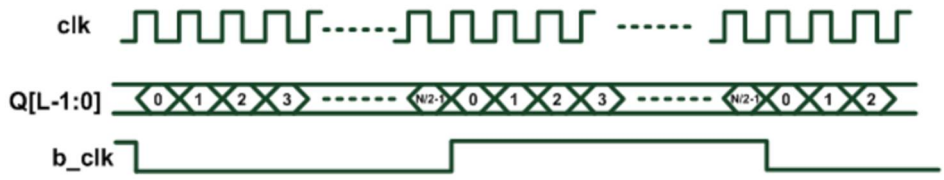
speaker_ctl:

- *Frequency dividers*
  - *audio_bck*
  - *audio_ws*
- *Parallel to serial module*
  - *To re-formulate the audio sequence*
    - *Right first, then left*
    - *MSB first*

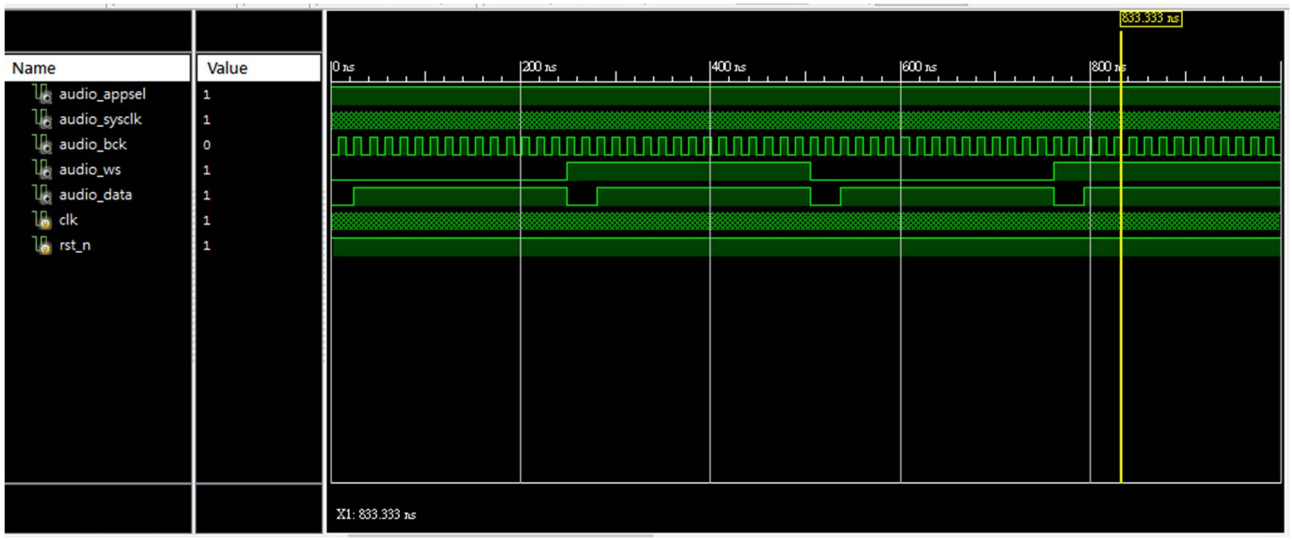


buzzer_ctl:

- *The buzzer frequency is obtained by dividing crystal frequency 40MHz by N.*
- *The buzzer clock (b_clk) is periodically inverted for every N/2 clock cycles. (determine the sound)*



The Final Result:



Discussion:

在 speaker_ctl 中設置一個 3bits 的 counter，取第 3 個 bit 為 *audio_bck，使其輸出頻率為 5MHz。當 audio_ws 為 0 時存入 right data，當 audio_ws 為 1 時存入 left data；並設置另一個 4bits 的 counter，當數到第 15 時，使 audio_ws 變為~audio_ws。*

2 Speaker control

2.1 Please produce the buzzer sounds of Do, Re, and Mi by pressing buttons (S1,S2,S3)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 volumn of the sound by pressing button (S4) as increase and (S5) and decrease the volumn. Please also quantize the audio dynamic range as 16 levels and show the current sound level in the 14-segment display.

Design Specification

output : audio_appsel // playing mode selection
audio_sysclk // control clock for DAC (from crystal)
audio_bck // bit clock of audio data (5MHz)
audio_ws // left/right parallel to serial control
audio_data
[3:0] ftsd_ctl
[14:0] display

input : clk // clock from crystal
rst_n // active low reset
Do
Re
Mi
increase //for volumn
decrease //for volumn

wire : clk_d;
[1:0] ftsd_ctl_en;
[15:0] volumn;
[4:0] level;

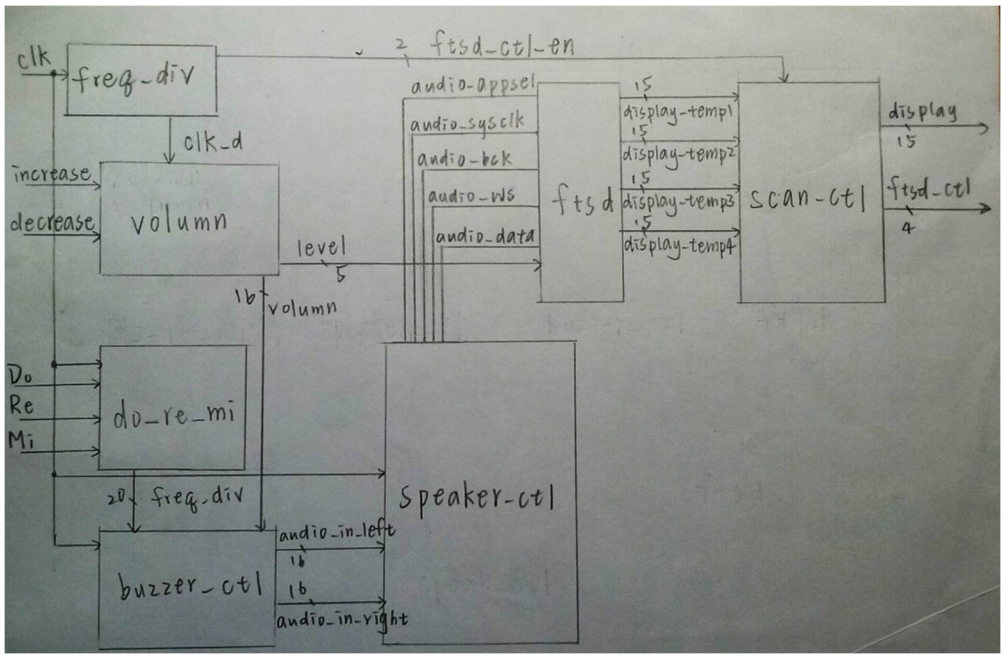
[19:0] freq_div;

[15:0] audio_in_left;

[15:0] audio_in_right;

[14:0] display_temp1,display_temp2,display_temp3,display_temp4;

block diagram:

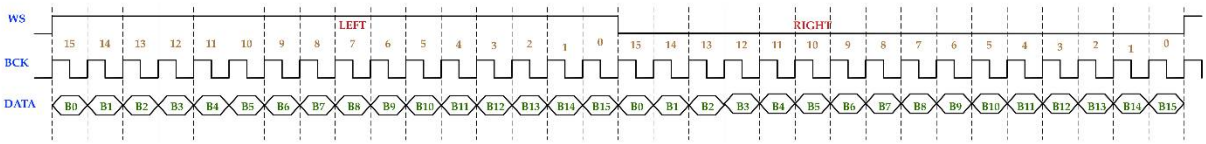


Design Implementation

logic function / logic diagram:

speaker_ctl:

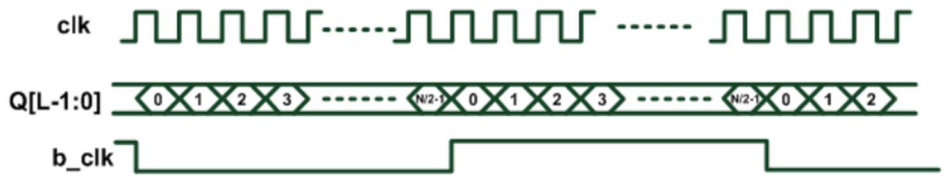
- *Frequency dividers*
  - *audio_bck*
  - *audio_ws*
- *Parallel to serial module*
  - *To re-formulate the audio sequence*
    - *Right first, then left*
    - *MSB first*



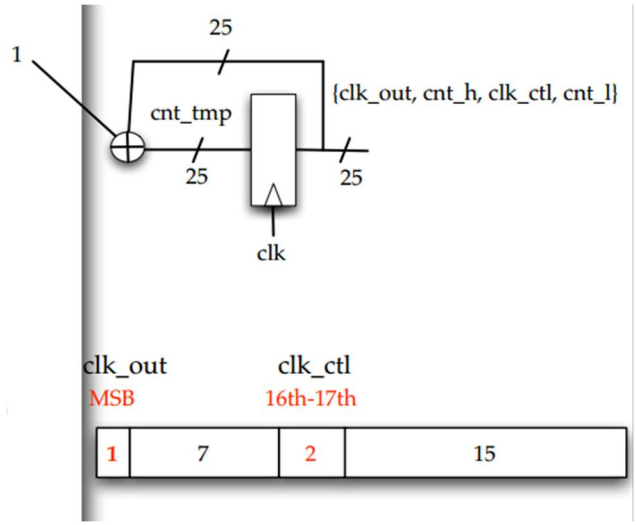
buzzer_ctl:

• The buzzer frequency is obtained by dividing crystal frequency 40MHz by N.

• The buzzer clock (b_clk) is periodically inverted for every N/2 clock cycles. (determine the sound)



freq_div:



scan_ctl

if ftsd_ctl_en=00

->控制第一個 14 段顯示器，顯示第一種狀況

if ftsd_ctl_en=01

->控制第二個 14 段顯示器，顯示第二種狀況

if ftsd_ctl_en=10

->控制第三個 14 段顯示器，顯示第三種狀況

if ftsd_ctl_en=11

->控制第四個 14 段顯示器，顯示第四種狀況

ftsd

當 level 大於 9，個位數 in1 與十位數 in2 分別控制 ftsd

當 bcd=4'd0: display = 15'b0000_0011_1111_111; //0

當 bcd=4'd1: display = 15'b1111_1111_1011_011; //1

當 bcd=4'd2: display = 15'b0010_0100_1111_111; //2
當 bcd=4'd3: display = 15'b0000_1100_1111_111; //3
當 bcd=4'd4: display = 15'b1001_1000_1111_111; //4
當 bcd=4'd5: display = 15'b0100_1000_1111_111; //5
當 bcd=4'd6: display = 15'b0100_0000_1111_111; //6
當 bcd=4'd7: display = 15'b0001_1111_1111_111; //7
當 bcd=4'd8: display = 15'b0000_0000_1111_111; //8
當 bcd=4'd9: display = 15'b0000_1000_1111_111; //9
default: display = 15'b1111_1111_1111_111; //DEF

do_re_mi

Mid Do: 261 Hz
Mid Re: 293 Hz
Mid Mi: 330 Hz

volumn_ctl

用 increase 和 decrease 控制 level 大小，並將 16 種 level 對應到 16 種不同的聲音大小。

I/O pin assignment:

```
NET "clk" LOC=R10;
NET "rst_n" LOC=N3;

NET "Do" LOC=P4;
NET "Re" LOC=P3;
NET "Mi" LOC=L6;

NET "increase" LOC=U1;
NET "decrease" LOC=T2;

NET "audio_appsel" LOC=H18;
NET "audio_bck" LOC=K16;
NET "audio_sysclk" LOC=H17;
NET "audio_ws" LOC=L15;
NET "audio_data" LOC=L16;

NET "display[14]" LOC = P6;
NET "display[13]" LOC = N4;
```

```
NET "display[12]" LOC = V5;
NET "display[11]" LOC = T5;
NET "display[10]" LOC = U7;
NET "display[9]" LOC = R3;
NET "display[8]" LOC = N5;
NET "display[7]" LOC = R5;
NET "display[6]" LOC = T3;
NET "display[5]" LOC = T4;
NET "display[4]" LOC = V4;
NET "display[3]" LOC = V7;
NET "display[2]" LOC = R7;
NET "display[1]" LOC = T7;
NET "display[0]" LOC = U5;

NET "ftsd_ctl<0>" LOC = V8;
NET "ftsd_ctl<1>" LOC = U8;
NET "ftsd_ctl<2>" LOC = V6;
NET "ftsd_ctl<3>" LOC = T6;
```

Discussion:

改變頻率挑整音高，改變震幅調整音量。

Conclusion:

一開始對於頻率轉換有些摸不著頭緒，只要了解 buzzer 的運作，再配合不同 state 的控制，即可產生有趣音階變化。

report(38/47)

9-1 (14/16)
Design Specification (2/2)
block diagram of the design or Logic Diagram (4/4)
Discussion +Conclusion (2/3)
Function explanation (3/4)
verify the design with simulation results (3/3)

9-2 (19/21)
Design Specification (3/3)
block diagram of the design or Logic Diagram (5/5)
I/O pin assignment (3/3)
Discussion +Conclusion (4/5)
Function explanation (4/5)