

1. Play the 14 sounds repeatedly based on the sound table. Every sound is played for one second.

1. Specification

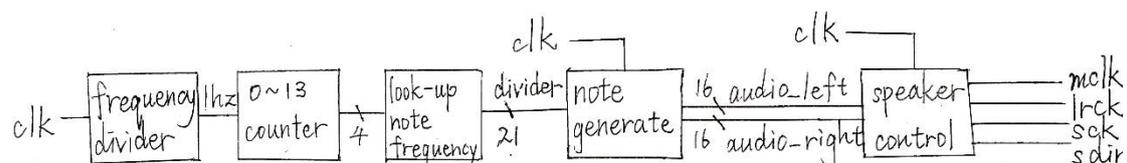
```

module LAB10_1_TOP(
    output audio_mclk, // master clock
    output audio_lrck, // left-right clock
    output audio_sck, // serial clock
    output audio_sdin, // serial audio data input

    input clk, // clock from the crystal
    input rst_n // active low reset
);

```

2. Block Diagram



先用除頻器製造出一赫茲的頻率，再用計數器數零到十三，再查表得到十四個音高的頻率 divider，再依據該頻率產生聲音訊號 audio left/right，再產生 DAC 所需的訊號 mclk、lrck、sck、sdin。

3. Finite state machine

無

4. Implement

先用一個二十七位元的計數器，作為除頻器，製造出一赫茲的頻率。如下。

```

15 reg [26:0] seconds_counter; // seconds_counter[26] is 1-second clock
16
17 always @(posedge clk or negedge rst_n) begin
18     if (~rst_n) seconds_counter <= 'b0;
19     else seconds_counter <= (seconds_counter < 27'd99_999_999)
20         ? (seconds_counter + 'b1)
21         : ('b0);
22 end

```

將一赫茲的時脈連接到四位元上數計數器。note_counter 每秒會增加一。如下。

```

24 reg [3:0] note_counter; // counts from 0 to 13
25
26 always @(negedge seconds_counter[26] or negedge rst_n) begin
27     if (~rst_n) note_counter <= 'b0;
28     else note_counter <= (note_counter < 4'd13)
29         ? (note_counter + 'b1)
30         : ('b0);
31 end

```

依據 note_counter，查表得到對應的音高頻率(note_frequency)。每秒改變一次音高。

```
37 reg [9:0] note_frequency;
38
39 always @* begin
40     case (note_counter)
41         4'h0: note_frequency <= 10'd261; // Do
42         4'h1: note_frequency <= 10'd293; // Re
43         4'h2: note_frequency <= 10'd330; // Mi
44         4'h3: note_frequency <= 10'd349; // Fa
45         4'h4: note_frequency <= 10'd392; // So
46         4'h5: note_frequency <= 10'd440; // La
47         4'h6: note_frequency <= 10'd494; // Si
48         4'h7: note_frequency <= 10'd524; // Do
49         4'h8: note_frequency <= 10'd588; // Re
50         4'h9: note_frequency <= 10'd660; // Mi
51         4'hA: note_frequency <= 10'd698; // Fa
52         4'hB: note_frequency <= 10'd784; // So
53         4'hC: note_frequency <= 10'd880; // La
54         4'hD: note_frequency <= 10'd988; // Si
55         default: note_frequency <= 'b0;
56     endcase
57 end
```

以 note_frequency 除 100M，得到 note_gen.v 的除數(note_div)。因為 note_gen.v 裡面用的是 T Flip-flop，所以要再除以二。

```
63 wire [15:0] audio_left;
64 wire [15:0] audio_right;
65
66 // a general frequency divider
67 // to generate the required frequencies for speaker clock
68 note_gen Ung(
69     .audio_left, // left sound audio
70     .audio_right, // right sound audio
71     .note_div(27'd100_000_000 / 2'd2 / note_frequency),
72     .clk, // clock from crystal
73     .rst_n // active low reset
74 );
```

最後，將 note_gen.v 產生的 audio left/right 連接到 speaker_control.v，就可以產生 DAC 所需的訊號(mclk、lrck、sck、sdin)。

```
82 speaker_control Usc(
83     // outputs
84     .audio_mclk, // master clock
85     .audio_lrck, // left-right clock
86     .audio_sck, // serial clock
87     .audio_sdin, // serial audio data input
88     // inputs
89     .audio_left, // left channel audio data input
90     .audio_right, // right channel audio data input
91     .clk, // clock from the crystal
92     .rst_n // active low reset
93 );
```

5. Conclusion

這一題只要把 LAB8 (speaker) 從三種音高增加十四種音高，並將控制方式從按鈕改成每秒加一的上數計數器即可。透過本題，可以熟悉使 DAC 產生特定音高的方法。

2. Electronic Organ

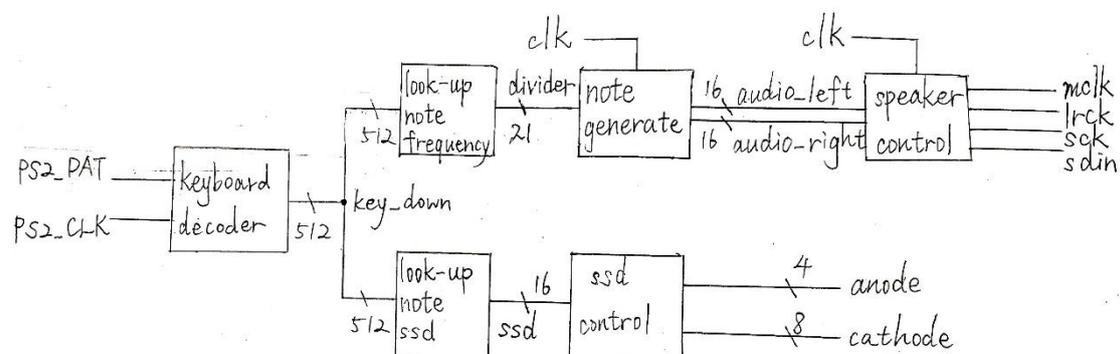
2.1 Integrate the keypad as the keyboard of the electronic organ. Keys c, d, e, f, g, a, b, C, D, E, F, G, A, B (two octaves from mid-Do) represent the sounds from low to high frequencies.

2.2 Display your playing sounds (Do, Re, Mi, Fa, So, La, Si) in the 7-segment LED

1. Specification

```
module LAB10_2_TOP(  
    // SSD  
    output reg [7:0] cathode,  
    output reg [3:0] anode,  
  
    // speaker  
    output audio_mclk,    // master clock  
    output audio_lrck,    // left-right clock  
    output audio_sck,     // serial clock  
    output audio_sdin,    // serial audio data input  
  
    // keyboard  
    inout PS2_DATA,  
    inout PS2_CLK,  
  
    input clk,            // clock from the crystal  
    input rst_n          // active low reset  
);
```

2. Block Diagram



從 KeyboardDecoder.v 的 key_down 可以判斷哪個鍵被按下。查表得到音高頻率和音高的 SSD 之後，分別連接到上一題用來產生聲音的模組和 SSD control。

3. Finite state machine

無

4. Implement

先利用 KeyboardDecoder.v 的 key_down，得知哪一個按鍵被按下。

```
24 wire [511:0] key_down;
25 wire [8:0] last_change;
26 wire key_valid;
27
28 KeyboardDecoder KeyboardDecoder(
29     .key_down,
30     .last_change,
31     .key_valid,
32     .PS2_DATA,
33     .PS2_CLK,
34     .rst(~rst_n),
35     .clk
36 );
```

查表，得到音高的頻率(note_frequency)。

```
42 reg [26:0] note_frequency;
43
44 always @* begin
45     case ('b1)
46         key_down[8'h1C]: note_frequency <= 27'd261; // Do
47         key_down[8'h1B]: note_frequency <= 27'd293; // Re
48         key_down[8'h23]: note_frequency <= 27'd330; // Mi
49         key_down[8'h2B]: note_frequency <= 27'd349; // Fa
50         key_down[8'h34]: note_frequency <= 27'd392; // So
51         key_down[8'h33]: note_frequency <= 27'd440; // La
52         key_down[8'h3B]: note_frequency <= 27'd494; // Si
53         key_down[8'h15]: note_frequency <= 27'd524; // Do
54         key_down[8'h1D]: note_frequency <= 27'd588; // Re
55         key_down[8'h24]: note_frequency <= 27'd660; // Mi
56         key_down[8'h2D]: note_frequency <= 27'd698; // Fa
57         key_down[8'h2C]: note_frequency <= 27'd784; // So
58         key_down[8'h35]: note_frequency <= 27'd880; // La
59         key_down[8'h3C]: note_frequency <= 27'd988; // Si
60         default: note_frequency <= 27'd100_000_000;
61     endcase
62 end
```

同上題，以 note_frequency 除 100M，得到 note_gen.v 的除數(note_div)。因為 note_gen.v 裡面用的是 T Flip-flop，所以要再除以二。

```
68 wire [15:0] audio_left;
69 wire [15:0] audio_right;
70
71 // a general frequency divider
72 // to generate the required frequencies for speaker clock
73 note_gen Ung(
74     .audio_left, // left sound audio
75     .audio_right, // right sound audio
76     .note_div(27'd100_000_000 / 2'd2 / note_frequency),
77     .clk, // clock from crystal
78     .rst_n // active low reset
79 );
```

再將 note_gen.v 產生的 audio left/right 連接到 speaker_control.v，就可以產生 DAC 所需的訊號(mclk、lrck、sck、sdin)。至此，按下鍵盤上的按鍵之後，就會使 DAC 產生對應的音高。接下來要試著讓鍵盤上的按鍵被按下之後，在七段顯示器上面顯示音高的名稱。

首先，查表，根據 key_down 產生對應的音高的名稱(note_name)，以 SSD 表示。

```
104 reg [15:0] note_name;
105
106 always @* begin
107     case ('b1)
108         key_down[8'h1C]: note_name <= 16'b10000101_11000101; // Do
109         key_down[8'h1B]: note_name <= 16'b11110101_01100001; // Re
110         key_down[8'h23]: note_name <= 16'b00101011_11011111; // Mi
111         key_down[8'h2B]: note_name <= 16'b01110001_00010001; // Fa
112         key_down[8'h34]: note_name <= 16'b01001001_11000101; // So
113         key_down[8'h33]: note_name <= 16'b11100011_00010001; // La
114         key_down[8'h3B]: note_name <= 16'b01001001_11011111; // Si
115         key_down[8'h15]: note_name <= 16'b10000101_11000101; // Do
116         key_down[8'h1D]: note_name <= 16'b11110101_01100001; // Re
117         key_down[8'h24]: note_name <= 16'b00101011_11011111; // Mi
118         key_down[8'h2D]: note_name <= 16'b01110001_00010001; // Fa
119         key_down[8'h2C]: note_name <= 16'b01001001_11000101; // So
120         key_down[8'h35]: note_name <= 16'b11100011_00010001; // La
121         key_down[8'h3C]: note_name <= 16'b01001001_11011111; // Si
122         default: note_name <= 16'hFFFF;
123     endcase
124 end
```

再把 note_name 連接到七段顯示器的控制模組即可。做法與以往相同，先產生一個適當的頻率。

```
131 reg [20:0] clk_f;
132
133 always @(posedge clk or negedge rst_n) begin
134     if (~rst_n) clk_f <= 'b0;
135     else        clk_f <= clk_f + 'b1;
136 end
```

依據該頻率，切換七段顯示器的陽極的 enable。

```
139 always @* begin
140     case (clk_f[20:19])
141         2'd0: anode = 4'b1110;
142         2'd1: anode = 4'b1101;
143         2'd2: anode = 4'b1011;
144         2'd3: anode = 4'b0111;
145         default: anode = 4'b0000;
146     endcase
147 end
```

並依據該頻率，同步傳送對應位數的 SSD 資料到七段顯示器的陰極。

```
152 always @* begin
153     case (clk_f[20:19])
154         2'd0: cathode = note_name[ 7:0];
155         2'd1: cathode = note_name[15:8];
156         2'd2: cathode = 8'hFF;
157         2'd3: cathode = 8'hFF;
158         default: cathode = 8'hFF;
159     endcase
160 end
```

5. Conclusion

這題只要在上一題的基礎上，將上一題的 0~13 計數器改成 LAB9 的鍵盤，再增加顯示器的功能即可。透過本題，可以熟悉鍵盤、DAC、七段顯示器等 IO 的使用方式。本題是這學期所遇到的，第一個需要同時使用這麼多外接儀器的題目。