FPGA Application Week 14 Exercise

CYEE 10828241 Chen Da-Chuan

May 26, 2023

Contents

List of Figures

List of Tables

1 Exercise 14-1 Use a state machine to play song automatically

1.1 Objective

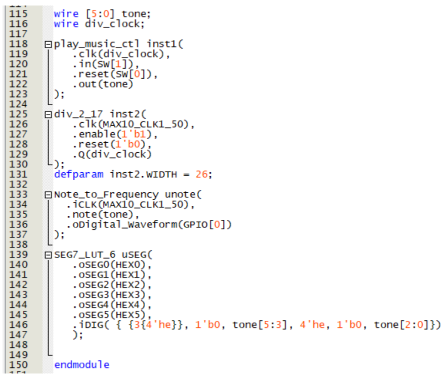
Each states in the state machine is going to play a note, and the state machine is going to enable us to program a sequential notes to play a song and automatically returns to the start.

1.2 Operation

Table 1: Exercise 14-1 Operation detail

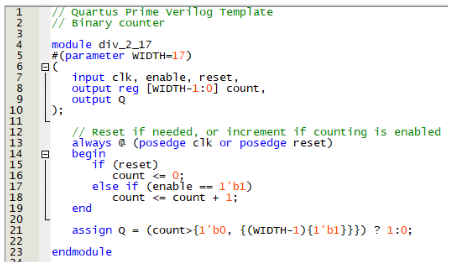
type	I/O	operation
input	MAX10_CLK1_50	$50MHz$ clock signal
input	SW[0]	reset state machine
input	SW[1]	play music
output	HEX0 + HEX3	tone (ex: C1 as 11)

1.3 Code



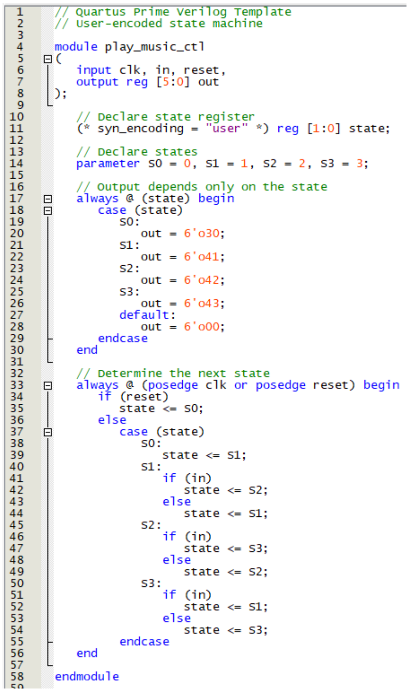
```
115    wire [5:0] tone;
116    wire div_clock;
117
118    play_music_ctl inst1(
119        .clk(div_clock),
120        .in(SW[1]),
121        .reset(SW[0]),
122        .out(tone)
123    );
124
125    div_2_17 inst2(
126        .clk(MAX10_CLK1_50),
127        .enable(1'b1),
128        .reset(1'b0),
129        .Q(div_clock)
130    );
131    defparam inst2.WIDTH = 26;
132
133    Note_to_Frequency unote(
134       .iCLK(MAX10_CLK1_50),
135       .note(tone),
136       .oDigital_Waveform(GPIO[0])
137    );
138
139    SEG7_LUT_6 uSEG(
140        .oSEG0(HEX0),
141        .oSEG1(HEX1),
142        .oSEG2(HEX2),
143        .oSEG3(HEX3),
144        .oSEG4(HEX4),
145        .oSEG5(HEX5),
146        .iDIG( { {3{4'he}}, 1'b0, tone[5:3], 4'he, 1'b0, tone[2:0]})
147        );
148
149
150    endmodule
```

Figure 1: de10_lite, top-level module



```
1     // Quartus Prime Verilog Template
2     // Binary counter
3
4     module div_2_17
5     #(parameter WIDTH=17)
6     (
7        input clk, enable, reset,
8        output reg [WIDTH-1:0] count,
9        output Q
10    );
11
12       // Reset if needed, or increment if counting is enabled
13       always @ (posedge clk or posedge reset)
14       begin
15          if (reset)
16             count <= 0;
17          else if (enable == 1'b1)
18             count <= count + 1;
19       end
20
21       assign Q = (count>{1'b0, {(WIDTH-1){1'b1}}}) ? 1:0;
22
23    endmodule
```

Figure 2: div_2_17, clock divider



```
1     // Quartus Prime Verilog Template
2     // User-encoded state machine
3
4     module play_music_ctl
5     (
6        input clk, in, reset,
7        output reg [5:0] out
8     );
9
10       // Declare state register
11       (* syn_encoding = "user" *) reg [1:0] state;
12
13       // Declare states
14       parameter S0 = 0, S1 = 1, S2 = 2, S3 = 3;
15
16       // Output depends only on the state
17       always @ (state) begin
18          case (state)
19             S0:
20                out = 6'o30;
21             S1:
22                out = 6'o41;
23             S2:
24                out = 6'o42;
25             S3:
26                out = 6'o43;
27             default:
28                out = 6'o00;
29          endcase
30       end
31
32       // Determine the next state
33       always @ (posedge clk or posedge reset) begin
34          if (reset)
35             state <= S0;
36          else
37             case (state)
38                S0:
39                   state <= S1;
40                S1:
41                   if (in)
42                      state <= S2;
43                   else
44                      state <= S1;
45                S2:
46                   if (in)
47                      state <= S3;
48                   else
49                      state <= S2;
50                S3:
51                   if (in)
52                      state <= S1;
53                   else
54                      state <= S3;
55             endcase
56       end
57
58    endmodule
```

Figure 3: play_music_ctl, music playing state machine

1. Figure 1 Line118-123: State machine, playing music with given clock signal.
2. Figure 1 Line125-131: Clock divider, divide the system clock speed ($50MHz$) to $0.745Hz$, which switches the state of state machine every $1.34s$.
3. Figure 1 Line133-137: Note to frequency converter, convert the note to frequency and use GPIO PIN to drive buzzer.
4. Figure 1 Line139-147: 7-segment display driver, display the current playing note.
5. Figure 2 Line4-23: Clock divider, the exact same design from the previous class with adjustable parameter.
6. Figure 3 Line17-30: Defines the output of each state of the state machine.
7. Figure 3 Line33-56: Defines how the state machine switches between states.

1.4 Result

The result is recorded and available at `https://youtu.be/xu-PDqTD7cE`.