Kareem Ahmad

EECS151

Lab 3 - Tone Generator

Questions

Question 1 - Square Wave Calculations

a) The period of the clock signal is 8ns

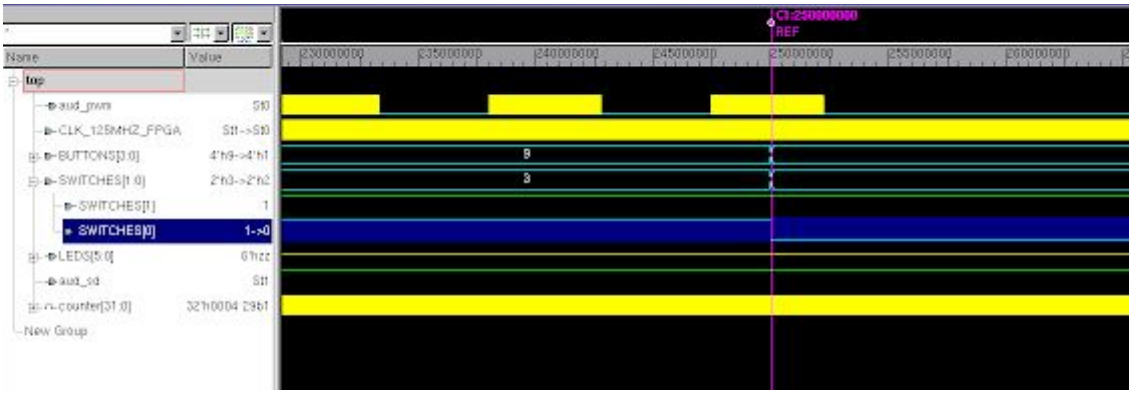
b) The period of a 220Hz wave is 4.5ms

c) 568181 clock cycles fit in one period of the square wave (though it rounds up to 568182).

Question 2 - Testbench Observations

a) If I increase the clock frequency from 125MHz, I would expect the tones generated by my tone generator to be of a higher frequency if no other changes are made. This is because the counter that serves as the basis for the tone counts a fixed number of clocks before inverting the output. With a higher clock frequency, each clock has a shorter period, and thus the counter will reach its target in shorter time. Thus its frequency is higher.

b) Output enable goes from 1 to 0 at the pink line (output enable is the line highlighted in blue), the top signal is the aud_pwm.



Effect of output enable

Question 3 - Verify Volume Adjustment

a) In the z1top_testbench file, volume can be tested by changing the code that controls the test duration (the first initial begin) to change the switches halfway through:

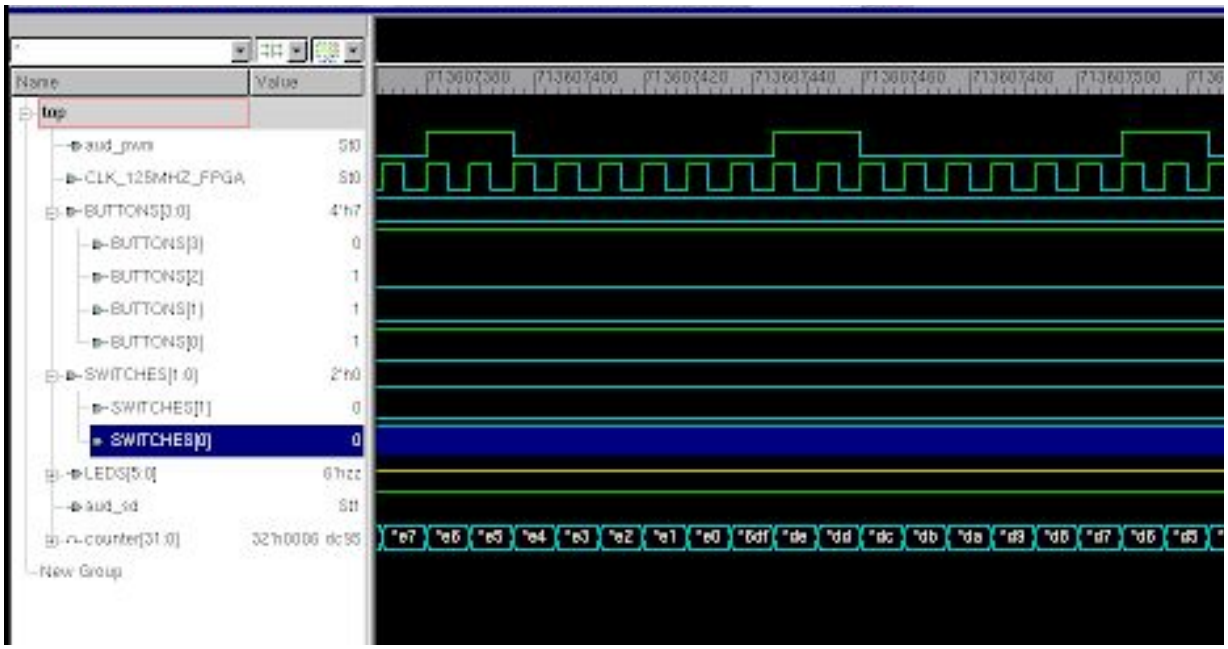
Original

```
27  initial begin
28    `ifdef IVERILOG
29      $dumpfile("z1top_testbench.fst");
30      $dumpvars(0,z1top_testbench);
31    `endif
32    #(200 * `MS);
33    $finish();
34  end
```

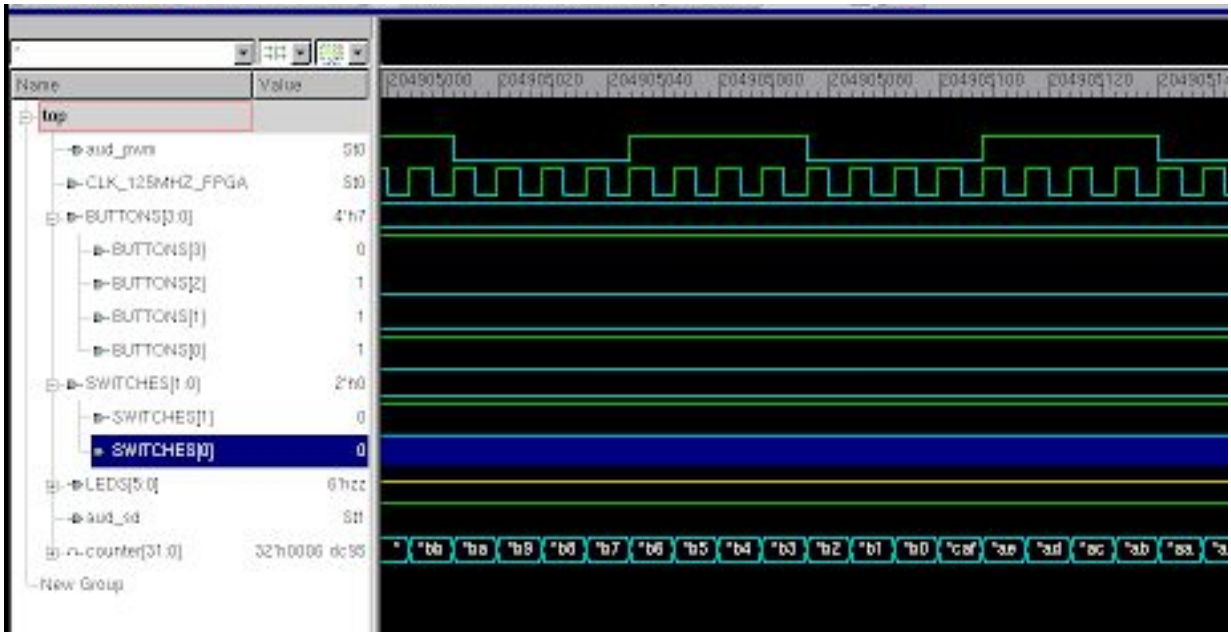
With volume test code

```
27  initial begin
28    `ifdef IVERILOG
29        $dumpfile("z1top_testbench.fst");
30        $dumpvars(0,z1top_testbench);
31    `endif
32    switches <= 2'b11;
33    #(200 * `MS);
34    switches <= 2'b01;
35    #(200 * `MS);
36    $finish();
37  end
```

b) See screenshots below:



Volume = 0



Volume = 1

Question 4 - Verify Configurable Frequency

a) Testbench created. As for volume, this can be done simply by editing the first initial begin block:

```
27    initial begin
28        `ifdef IVERILOG
29            $dumpfile("z1top_testbench.fst");
30            $dumpvars(0,z1top_testbench);
31        `endif
32        repeat (5) begin
33            buttons <= 4'b0001;
34            #(50 * `MS);
35            buttons <= 4'b0011;
36            #(50 * `MS);
37            buttons <= 4'b0111;
38            #(50 * `MS);
39            buttons <= 4'b0001;
40            #(50 * `MS);
41            buttons <= 4'b1001;
42            #(50 * `MS);
43            switches <= switches - 1;
44        end
45        //#(200 * `MS);
46        $finish();
47    end
```

b) When tone_switch_period is 0 (all buttons set to 0), there is no output:

Tone frequency goes to zero at the pink line

Final Code

tone_generator.v

```
1  `timescale 1ns/1ns
2  module tone_generator (
3      input clk,
4      input output_enable,
5      input [23:0] tone_switch_period,
6      input volume,
7      output square_wave_out,
8      output [31:0] _counter
9  );
10     reg [31:0] counter = 32'h0000_0001; // true min is 20 for the default settings
11     reg square_out = 0;
12     reg pwm_out = 0;
13
14     assign square_wave_out = square_out && pwm_out;
15     assign _counter = counter;
16
17     always @(posedge clk) begin
18         // square wave generator
19         if (counter <= 32'h0000_0001) begin
20             if (output_enable == 1'b1) begin
21                 square_out <= ~square_out;
22             end else begin
23                 square_out <= 0;
24             end
25             //A
26             //counter <= 32'h0000_0001;
27             counter[23:0] <= tone_switch_period;
28             //counter <= 284091;
29         end else begin
30             //counter <= counter + 1;
31             counter <= counter - 1;
32         end
```

```
33          // volume duty cycle
34          if (volume == 1) begin
35              pwm_out <= counter[2];
36          end else begin
37              pwm_out <= counter[2] && counter[1];
38          end
39      end
39  endmodule
```

z1top.v

```
 1  `timescale 1ns/1ns
 2
 3  module z1top (
 4      input CLK_125MHZ_FPGA,
 5      input [3:0] BUTTONS,
 6      input [1:0] SWITCHES,
 7      output [5:0] LEDS,
 8      output aud_pwm,
 9      output aud_sd
10  );
11      wire [31:0] counter;
12
13      assign aud_sd = 1;
14
15      tone_generator dut(.clk(CLK_125MHZ_FPGA), .square_wave_out(aud_pwm),
                .tone_switch_period(BUTTONS[3:0] << 16), .output_enable(SWITCHES[0]),
                .volume(SWITCHES[1]),     ._counter(counter));
16  endmodule
```

z1top_tesbench.v

The only lines changed in this file are shown in question 4.