

Experiment 10: Tone Generator

Debasish Panda 21D070021

October 25, 2022

1 Overview of the experiment

In this experiment, we will generate the seven major notes in the Indian classical music named sa, re, ga, ma, pa, dha, ni and sa (upper octave). We will use the 8 slide switches on the Xenon Board to play these notes. The frequency for the above mentioned notes is to be generated using clock divider introduced in the last week's experiment.

The steps involved in designing the Tone Generator are:

i) First of all we need to calculate the count value for each note by taking the master clock as 50 MHz (Xenon's On-board clock).

(NOTE: If the count value comes in fraction take the nearest integer value.)

ii) Then we need to write the VHDL code for playing the notes Sa, Re, Ga, Ma, Pa, Dha, Ni, Sa(upper octave).

iii) The 8 Slide Switches of the Xenon board are mapped like the keys of a Piano/Keyboard for playing the notes. The note will play when corresponding slide switch gets ON.

(NOTE: Reset is not there for Tone Generator module.)

iv) Corresponding to each note(slide switch on), an LED on Xenon board should be mapped to indicate the note being played. The LED will glow when the respective slide switch is switched ON (all 8 LEDs will be used, one for each note/slide switch).

v) The GPIO pins of the Xenon Board can not drive a Speaker directly

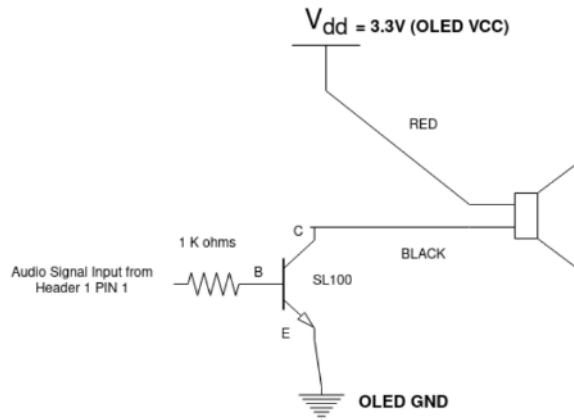
because of current limitation, therefore, a SL100 Transistor will be used to drive the Speaker. The output needs to be connected to the SL100 transistor taken from Header 1 PIN 1 written on the board.

vi) Sa (lower octave) need to be played when SW1 is ON, Re note should be played when SW2 and SW1 is ON, Ga played when SW3, SW2, SW1 is ON and so on for other notes.

vii) The supply voltages, +3.3V and GND for the speaker are to be taken from OLED VCC and GND written on the board.

The appropriate code for Behavioral-Dataflow modelling of the system is written using VHDL on Quartus Prime software. Next, we have to demonstrate the Tone Generator practically by using the Xen10 FPGA board. In order to run the VHDL code on the board, we also need to determine the input and output pins correctly as per the specifications of the circuit board.

2 Experimental setup



Speaker driver circuit

2.1 Design Documentation and VHDL Code

The VHDL code associated with the digital circuit is as follows:

```
library ieee;
use ieee.std_logic_1164.all;

entity tone_generator is
port(clk_out : out std_logic;
      led : out std_logic_vector(7 downto 0);
      switch : in std_logic_vector(7 downto 0);
      clk_50 : in std_logic);
end entity tone_generator;

architecture Struct of tone_generator is

    signal i0, i1, i2, i3, i4, i5, i6, i7: integer := 0;
    signal outp: std_logic_vector(7 downto 0) := "11111111";
    signal clock: std_logic_vector(7 downto 0) := "11111111";

begin
    clock_proc0: process(clk_50)
    begin
        if(clk_50='1' and clk_50'event) then
            if(i0=104168) then
                outp(0) <= not outp(0);
                i0 <= 1;
            else
                i0 <= i0+1;
            end if;
        end if;
        clock(0) <= outp(0);
    end process;

    clock_proc1: process(clk_50)
    begin
        if(clk_50='1' and clk_50'event) then
```

```

if(i1=92593) then
outp(1) <= not outp(1);
i1 <= 1;
else
i1 <= i1+1;
end if;
end if;
clock(1) <= outp(1);
end process;

```

```

clock_proc2: process(clk_50)
begin
if(clk_50='1' and clk_50'event) then
if(i2=83333) then
outp(2) <= not outp(2);
i2 <= 1;
else
i2 <= i2+1;
end if;
end if;
clock(2) <= outp(2);
end process;

```

```

clock_proc3: process(clk_50)
begin
if(clk_50='1' and clk_50'event) then
if(i3=78125) then
outp(3) <= not outp(3);
i3 <= 1;
else
i3 <= i3+1;
end if;
end if;
clock(3) <= outp(3);
end process;

```

```

clock_proc4: process(clk_50)
begin
if(clk_50='1' and clk_50'event) then
if(i4=69444) then
outp(4) <= not outp(4);
i4 <= 1;
else
i4 <= i4+1;
end if;
end if;
clock(4) <= outp(4);
end process;

```

```

clock_proc5: process(clk_50)
begin
if(clk_50='1' and clk_50'event) then
if(i5=62500) then
outp(5) <= not outp(5);
i5 <= 1;
else
i5 <= i5+1;
end if;
end if;
clock(5) <= outp(5);
end process;

```

```

clock_proc6: process(clk_50)
begin
if(clk_50='1' and clk_50'event) then
if(i6=55556) then
outp(6) <= not outp(6);
i6 <= 1;
else
i6 <= i6+1;
end if;

```

```

end if;
clock(6) <= outp(6);
end process;

```

```

clock_proc7: process(clk_50)
begin
if(clk_50='1' and clk_50'event) then
if(i7=52083) then
outp(7) <= not outp(7);
i7 <= 1;
else
i7 <= i7+1;
end if;
end if;
clock(7) <= outp(7);
end process;

```

```

final: process(switch)
begin
if switch = "00000001" then
clk_out <= clock(0);
led <= "00000001";
elsif switch = "00000011" then
clk_out <= clock(1);
led <= "00000010";
elsif switch = "00000111" then
clk_out <= clock(2);
led <= "00000100";
elsif switch = "00001111" then
clk_out <= clock(3);
led <= "00001000";
elsif switch = "00011111" then
clk_out <= clock(4);
led <= "00010000";
elsif switch = "00111111" then
clk_out <= clock(5);

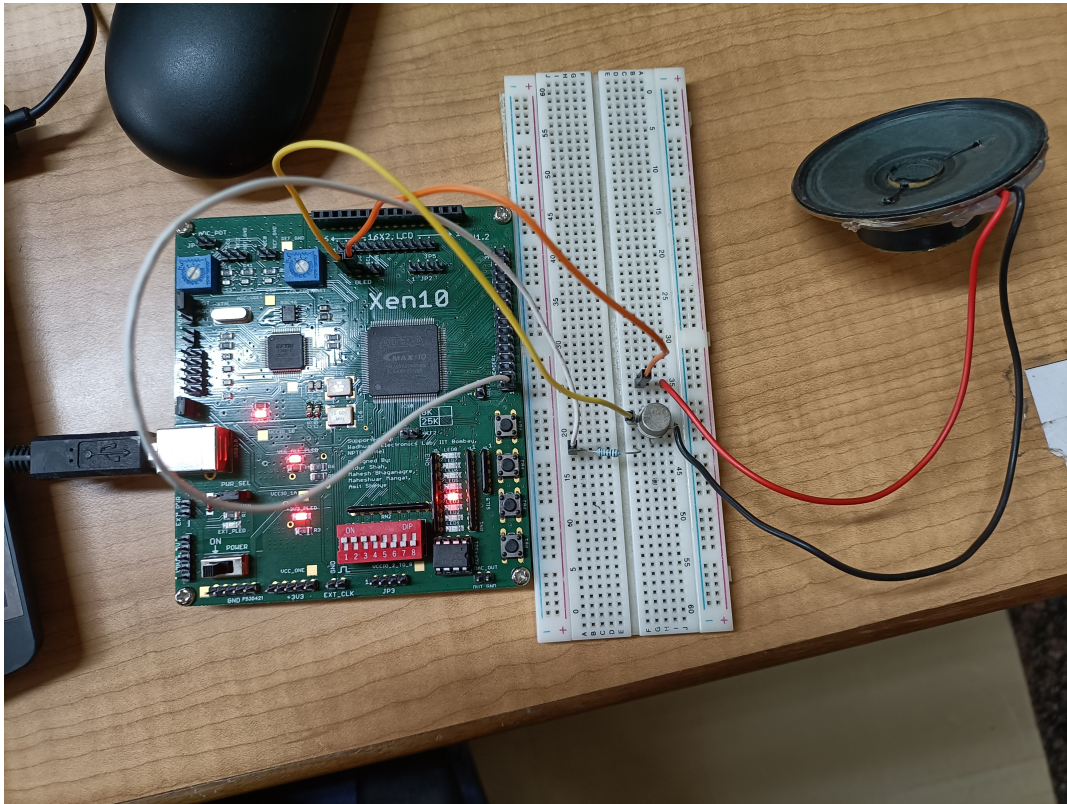
```

```
    led <= "00100000";
elsif switch = "01111111" then
    clk_out <= clock(6);
    led <= "01000000";
elsif switch = "11111111" then
    clk_out <= clock(7);
    led <= "10000000";
end if;
end process;

end Struct;
```

3 Observations

Following a successful compilation of the VHDL program, we need to execute the pin planning correctly and create an .svf file for the same, which is executable by the Xenon board. The experiment was conducted successfully and the Xenon board gave the desired result, with the tone generator working as per specifications. The circuit setup of the Tone Generator is shown below:



Tone Generator circuit diagram

4 References

- [1] [EE214 Github page](#)
- [2] [Overleaf LaTeX tutorial for beginners](#)
- [3] [Introduction to Xen10 Board](#)