EE214 Tone Synthesizer

Soumyajit Langal

March 2021

1 Designing of the Synthesizer

In this experiment, you will generate the seven major **notes** in the Indian classical music named **sa, re, ga, ma, pa, dha, ni** and **Sa (upper octave)**. You will use the 8 slide switches on the krypton Board to play these notes. The frequency for the above mentioned notes is to be generated using clock divider introduced in the previous session.

2 Background

The seven major notes in the Indian scale are called Sa, Re, Ga, Ma, Pa, Dha, Ni and Sa (upper octave). The frequency of these notes are fixed by musical considerations in the Indian scale as shown in the **Table 1**.

To generate each note, you will generate the square wave of frequency corresponding to each note. Though the square wave will have the harmonics other than the fundamental tone, the resultant note is pleasant to our ears and simple to generate.
Table 1 lists the Shuddha notes in an octave, in Indian classical music Sa, Re, Ga, Ma, Pa, Dha, Ni, Sa (upper octave). Octave refers to one set of 7 major notes Sa, Re, Ga, Ma, Pa, Dha, Ni. The next octave begins from the Upper Sa. In lower octave, all the notes will have the frequency half the corresponding frequency in the next octave. This means, the **frequency range of this octave is from 240Hz to 480Hz** with all the intermediate frequencies as written in the Table 1. For lower octave, they will vary from 120 Hz to 240Hz.

3 Problem Statement

Table 1: Frequency Table

Note	**Frequency(Hz)**	**Count Value**
Sa	240	104168
Re	270	
Ga	300	
Ma	320	
Pa	360	
Dha	400	
Ni	450	
Sa (upper octave)	480	

- Fill up the **Count Value** column in Table 1. You need to calculate the count value for each note by taking the **master clock as 50 MHz**(Krypton's On-board clock). **NOTE:** If the count value comes in fraction take the nearest integer value.
- Map the **8 Slide Switches** of the Krypton board like the keys of a Piano/Keyboard for playing the notes (one at a time). The note will play till the corresponding slide switch is ON.

- Corresponding to each note(slide switch on), an LED on Krypton 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).
- The GPIO pins of the CPLD can not drive a Speaker directly because of current limitation, therefore, use **SL100 Transistor to drive the Speaker**. The Circuit schematic and the Krypton GPIO pins to be used for this purpose are illustrated in **Figure 1**.
- Refer to the **sample code in section 3.1** for I/O Ports in the Entity and write the VHDL description to meet the specifications mentioned above.
- Simulate it using **Testbench (new one, NOT Generic)** provided to you, and show it to your TA.
- Generate SVF file, **program the Krypton board and demonstrate the notes**(one at a time) to your TA.
- **Upload LAB work and Report on Moodle** as per the respective deadlines. The report should consist of Table 1 completely filled, include the calculation of count value in your report. Add VHDL files in the folder including Testbench and screenshot of the Output Waveforms (ModelSim).

3.1 Circuit Diagram

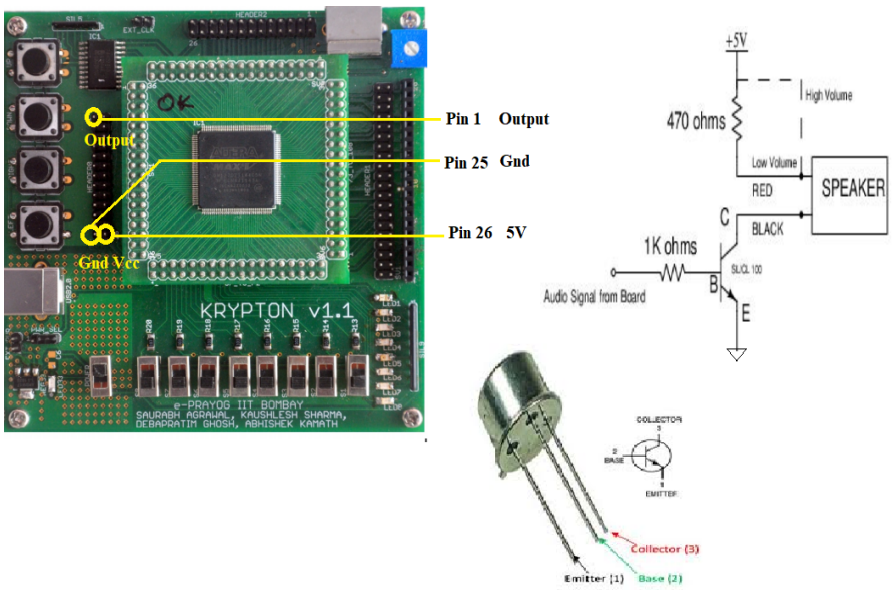

Figure 1: Circuit Diagram

3.2 PIN Map for Krypton Board

- **50 MHz Clock:** PIN 89
- **Slide Switch(S1-S8):** PIN (48, 45, 44, 43, 42, 41, 40, 39)
- **LEDs (LED1-LED8):** PIN (58, 57, 55, 53, 52, 51, 50, 49)
- **Music Output:** PIN 1
- **VCC (+5V):** PIN 26
- **GND (0V):** PIN 25

4 Appendix

4.1 An example of a simple counter (clock divider)

Suppose we need to generate $f = 5\ MHz$ from 50 MHz master clock. For this, we need a counter such that the clock out remains **HIGH for 5 Input(master) Clock Cycles and LOW for next 5 Clock Cycles**. In order to do this, we set-up a counter that starts from 1 and increments at every positive edge of the Input Clock(master) till the count reaches its maximum value which is 5 in this case.

$$count = 50MHz/(2 * f) = 5$$

After the count reaches 5, count will be initialized back to 1. And clock output will go LOW till count reaches maximum again.
Note: *Here we are counting from 1 to maximum count. (Not from 0 to maximum count -1).*

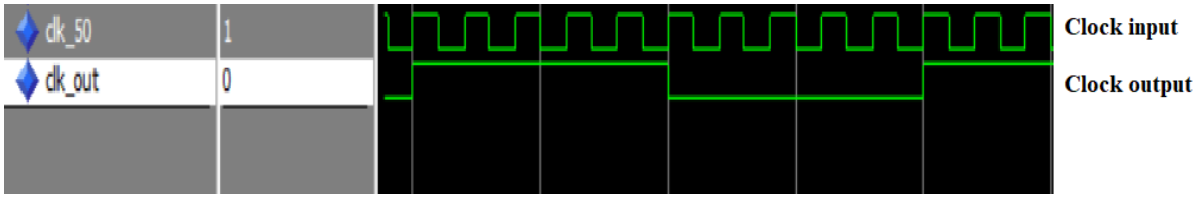

Figure 2: Waveform

4.2 I/O Port for Entity and Example code for Sa(240 Hz)

```
entity toneGenerator is
port (toneOut : out std_logic; --this pin will give your notes output
    clk : in std_logic;
    LED : out std_logic_vector(7 downto 0);
    switch : in std_logic_vector(7 downto 0));
end entity toneGenerator;
--..............Start the architecture here.....................
-------------------CODE
-------------------CODE
if (count_sa1 = 104168) then--240Hz
    count_sa1 := 1;
    sa1 := not sa1;
else
    count_sa1 := count_sa1 + 1;
end if;
toneOut <= sa1;
LED <= (0 => '1', others => '0');
-------------------CODE
-------------------CODE
```

5 Make a video

Make a video of the presentation and upload in Moodle. Please make the resolution 360p so that size is less than 50MB. Try to keep the duration of the video small.