Experiment: Music Synthesizer

Prakhar Mittal Roll Number 190070046

EE-214, WEL, IIT Bombay

April 07, 2021

## Overview of the experiment:

|  |
| --- |
| The purpose of the experiment was to generate a looping song using behavioural style VHDL on the Krypton board using state transitions. The looping song plays a certain sequence of tones with corresponding LEDs glowing for each tone. If the reset switch is closed, the song stops and LEDs stop glowing. When the reset is opened, the song starts playing from the start again. |

## Approach to the experiment:

|  |
| --- |
| For generating the song, I created an FSM that keeps track of the count. There are 6 states, Silent, Sa, Re, Ga, Ma, Ni, each corresponding to 0.25s of tone. While reset is 1, the state remains Silent. When it goes to 0, start counting in intervals of 0.25 seconds and take appropriate state transitions every 0.25s. When the count reaches 32, count is reset to 0, and state goes back to silent. This loops. |

## Design document and VHDL code if relevant:

|  |
| --- |
| toneGenerator: It takes switch (std\_logic\_vector<7 downto 0>), 50MHz clk, resetn as input, and plays the corresponding tone on the board.  Example of toneGenerator:  if(clk='1' and clk' event) then  --Sa1  if(switch="00000001") then  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');  Silent and a tone state:  when Silent=>  y\_next\_var := sa;  n\_count := 1;  --assign the signal for switch which will be the input of toneGenerator component    switch <= "11000010";  WHEN sa => --if the machine in Sa state  if((count = 1) or (count = 5) or (count = 9)) then  y\_next\_var := sa;    elsif((count = 2) or (count = 6) or (count = 10)) then  y\_next\_var := ga;    elsif((count = 16) or (count = 31)) then  y\_next\_var := ni;  end if;  n\_count := count + 1;  --assign the signal for switch which will be the input of toneGenerator  switch <= "00000001";  State Transition:  if (clock\_music = '1' and clock\_music' event) then  if (resetn = '1') then  y\_present <= Silent;  count <= 1;    else  y\_present <= y\_next\_var;  count <= n\_count;    end if;  end if; |

## State View:

|  |
| --- |
|  |

## RTL Simulation:

|  |
| --- |
|  |

## Krypton board\*:

|  |
| --- |
|  |

## Observations\*:

|  |
| --- |
| When switch mapped to reset is kept open, the sing plays out and the LED corresponding to each note lights up. When switch is closed, the song stops playing, and the leds stop glowing. On opening the switch again, the song starts from the first tone. |

## References:

|  |
| --- |
| Nil |

\* To be submitted after the tutorial on ”Using Krypton.