Experiment 6
Music Synthesizer

Mayur Ware, 19D070070
EE-214, WEL, IIT Bombay

April 02, 2021

Overview of the experiment

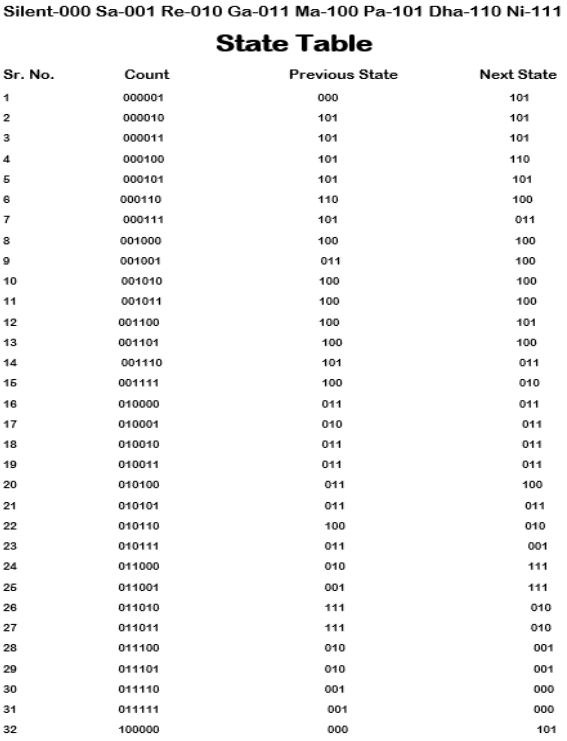
- The purpose of the experiment is to design a circuit which will playing a music using the notes which were introduced in the Tone Synthesizer experiment. The notes will play in a particular sequence like in a music.
- I designed a FSM to automate such that note will be played one after another in the given sequence and for a particular duration. I used toneGenerator design of previous lab as a component here. Then, I used Switch S1 as reset which will stop the music. 2 Notes
- The report contains State Transition Graphs (Handwritten and Quartus made), Sate table, notes table, some important extracts of VHDL code, the output waveforms, Krypton Board Observations and Circuit Images.

Approach to the experiment

First, I completed the given notes table which I will use in FSM to change the states sequentially. Each state has been assigned for a duration of 0.25s.
The counts for each Note are given in this table:

Pa	Pa	Dha	Pa	Ma	Ga	Ma	Ma	Pa	Ma	Ga	Re
0.5s	0.5s	0.25s	0.25s	0.25s	0.25s	0.5s	0.5s	0.25s	0.25s	0.25s	0.25s
1, 2	3, 4	5	6	7	8	9, 10	11, 12	13	14	15	16
Ga	Ga	Ma	Ga	Re	Sa	Ni	Re	Sa	Sa	Silent	Silent
0.5s	0.5s	0.25s	0.25s	0.25s	0.25s	0.5s	0.5s	0.25s	0.25s	0.25s	0.25s
17, 18	19, 20	21	22	23	24	25, 26	27, 28	29	30	31	32

Then, I made cases for each note according to its and count and the next note to be played.
Then, I made a 4Hz clock (for 0.25s time period) using the 50MHz Master Clock.
After completion of each state, at the end, program will increment it to the next state.
Following is the State Table using which I played the music.

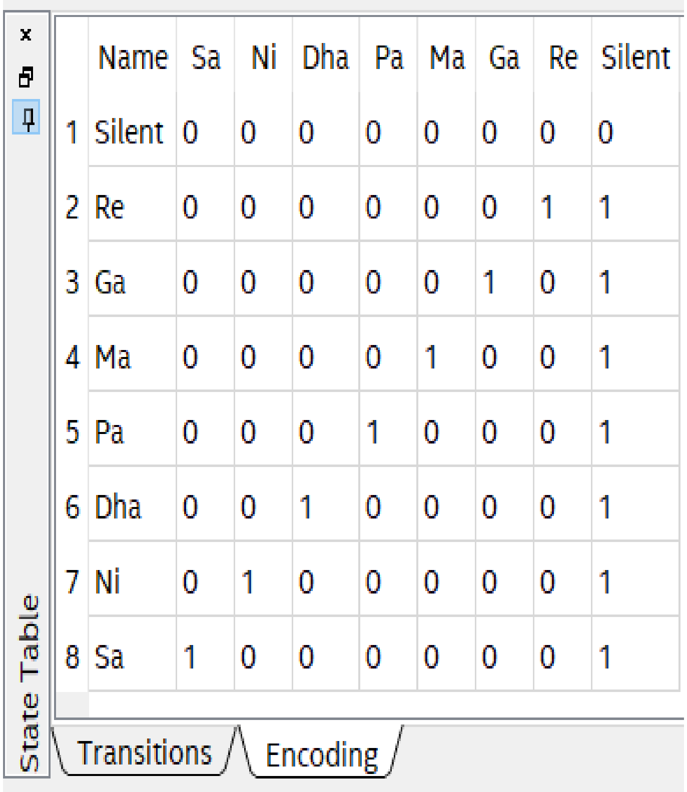


Silent-000 Sa-001 Re-010 Ga-011 Ma-100 Pa-101 Dha-110 Ni-111

State Table

Sr. No.	Count	Previous State	Next State
1	000001	000	101
2	000010	101	101
3	000011	101	101
4	000100	101	110
5	000101	101	101
6	000110	110	100
7	000111	101	011
8	001000	100	100
9	001001	011	100
10	001010	100	100
11	001011	100	100
12	001100	100	101
13	001101	100	100
14	001110	101	011
15	001111	100	010
16	010000	011	011
17	010001	010	011
18	010010	011	011
19	010011	011	011
20	010100	011	100
21	010101	011	011
22	010110	100	010
23	010111	011	001
24	011000	010	111
25	011001	001	111
26	011010	111	010
27	011011	111	010
28	011100	010	001
29	011101	010	001
30	011110	001	000
31	011111	001	000
32	100000	000	101

Table 1



	Name	Sa	Ni	Dha	Pa	Ma	Ga	Re	Silent
1	Silent	0	0	0	0	0	0	0	0
2	Re	0	0	0	0	0	0	1	1
3	Ga	0	0	0	0	0	1	0	1
4	Ma	0	0	0	0	1	0	0	1
5	Pa	0	0	0	1	0	0	0	1
6	Dha	0	0	1	0	0	0	0	1
7	Ni	0	1	0	0	0	0	0	1
8	Sa	1	0	0	0	0	0	0	1

Table 2

Design document and VHDL code if relevant

Main Entity :

```
entity music is
port (toneOut : out std_logic;
clk_50, resetn : in std_logic;
LED : out std_logic_vector(7 downto 0));
end entity music;
```

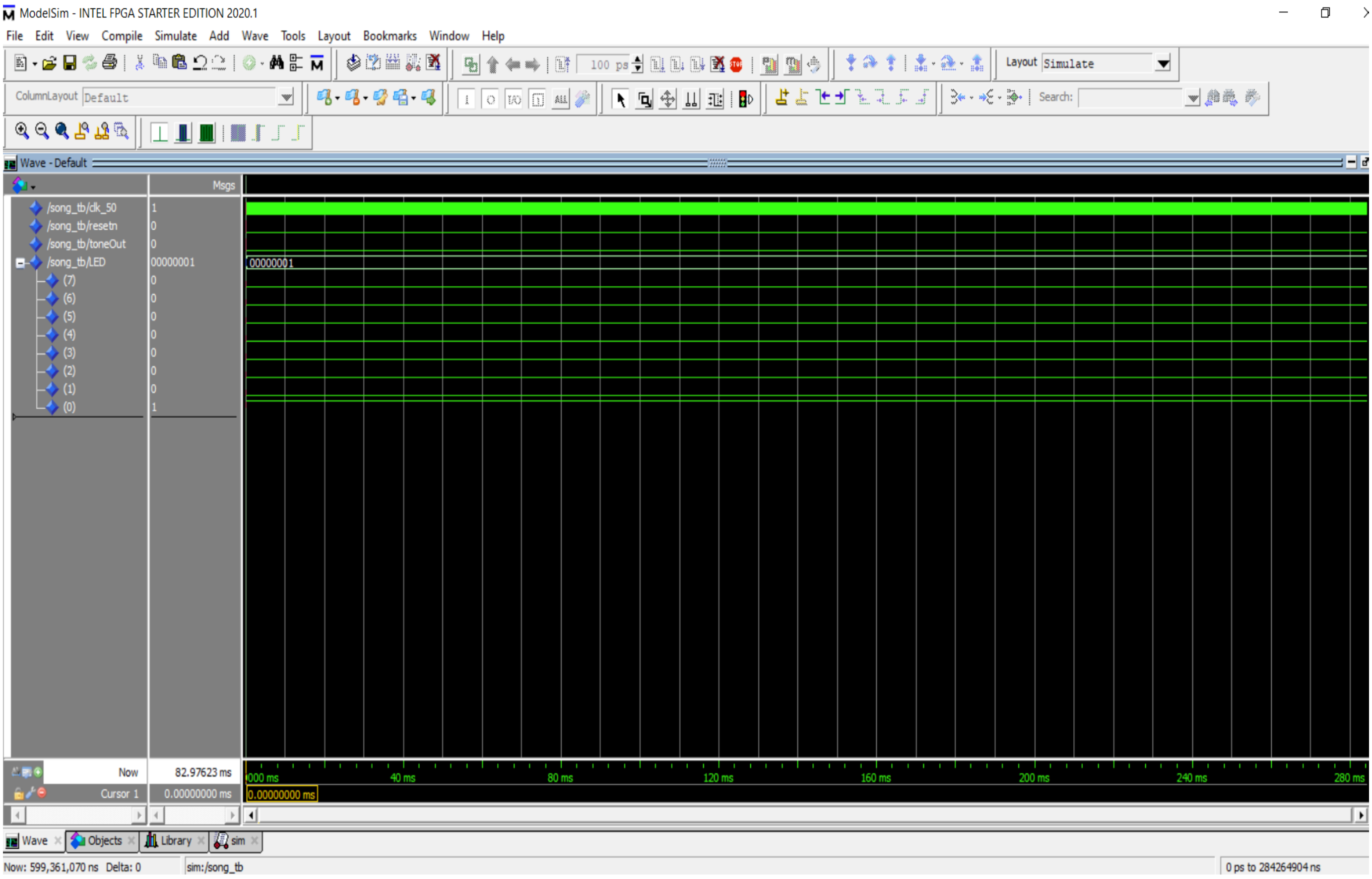
States :

```
case y_present is
WHEN Sa=>
if((n_count = 24)) then
y_next_var := Ni;
elsif((n_count=30)) then
y_next_var := Silent;
end if;
LED<=LED1;
toneOut<= Sa_1;
```

```
WHEN Re =>
if(n_count = 16) then
y_next_var := Ga;
elsif((n_count=23) or (n_count = 28)) then
y_next_var := Sa;
end if;
LED<=LED2;
toneOut< Re_1;
```

Similarly, for other states 'Ga', 'Ma', 'Pa', 'Dha', 'Ni'.

RTL Simulation



ModelSim - INTEL FPGA STARTER EDITION 2020.1

File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
100 ps
Layout Simulate
ColumnLayout Default
Search:
Wave - Default
Msgs

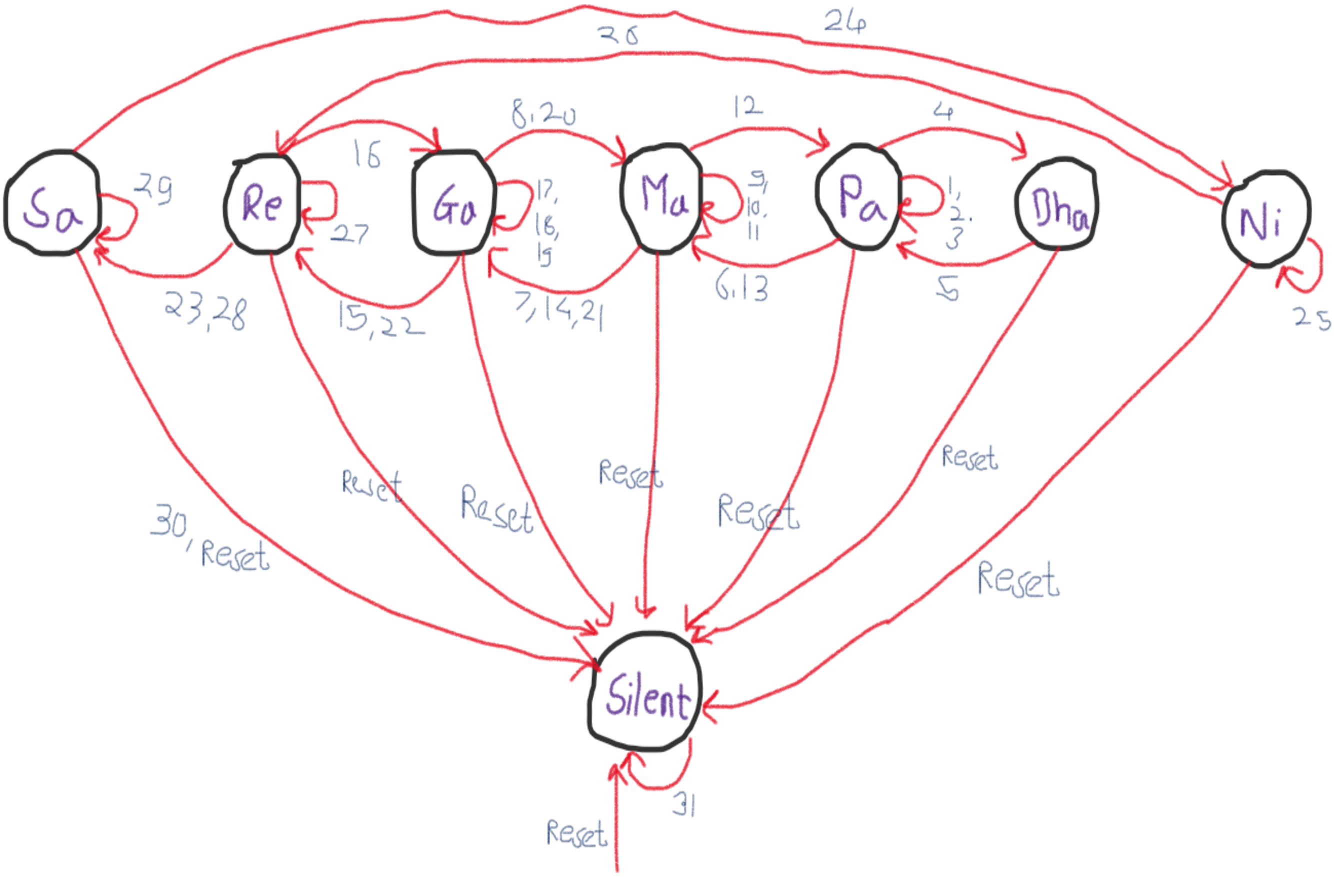
```
/song_tb/clk_50     1
/song_tb/resetn     0
/song_tb/toneOut    0
/song_tb/LED        00000001    00000001
  (7)               0
  (6)               0
  (5)               0
  (4)               0
  (3)               0
  (2)               0
  (1)               0
  (0)               1

Now        82.97623 ms
Cursor 1   0.00000000 ms   0.00000000 ms
```

Now: 599,361,070 ns Delta: 0 sim:/song_tb 0 ps to 284264904 ns

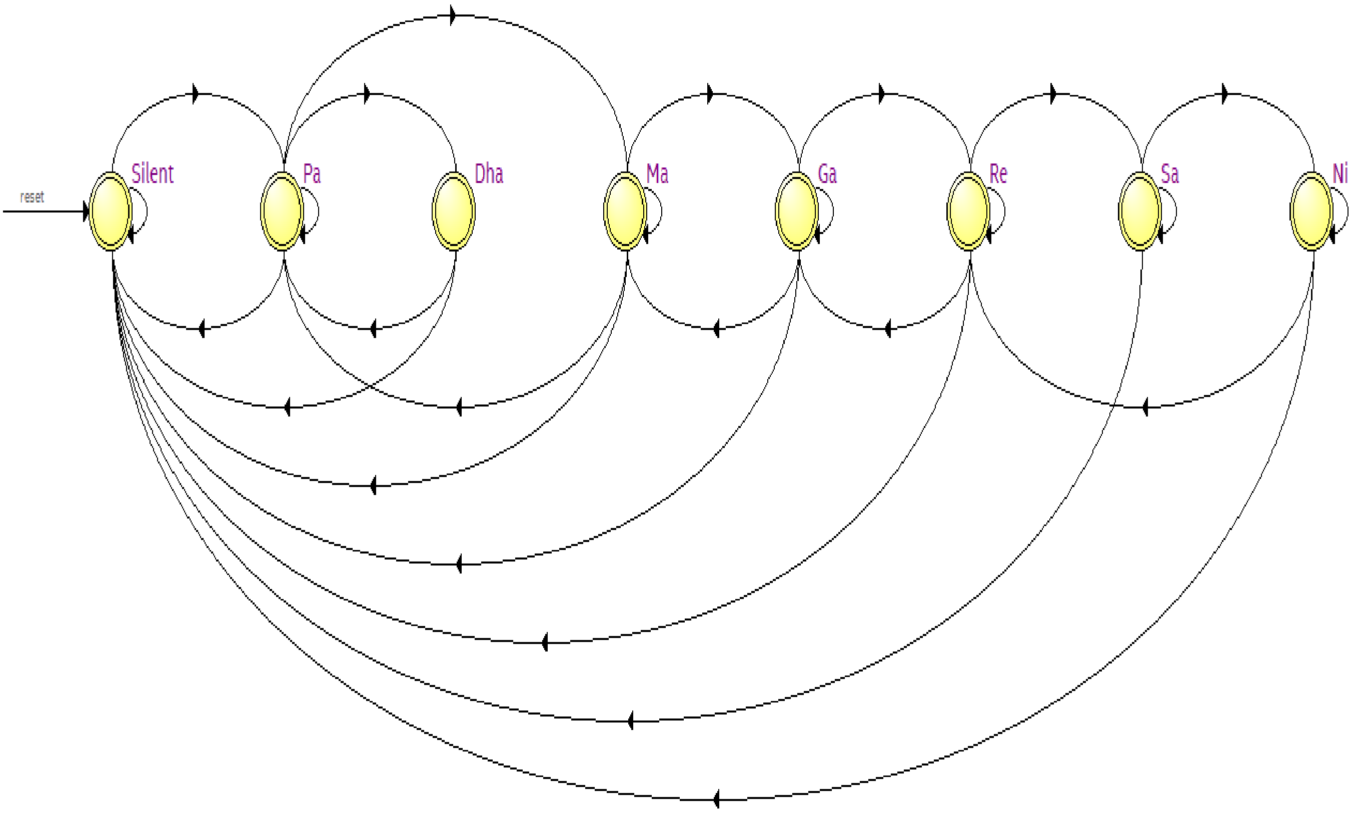
DUT Input/Output Format

```
toneOut : out std_logic;
clk_50, resetn : in std_logic;
LED : out std_logic_vector(7 downto 0));
```

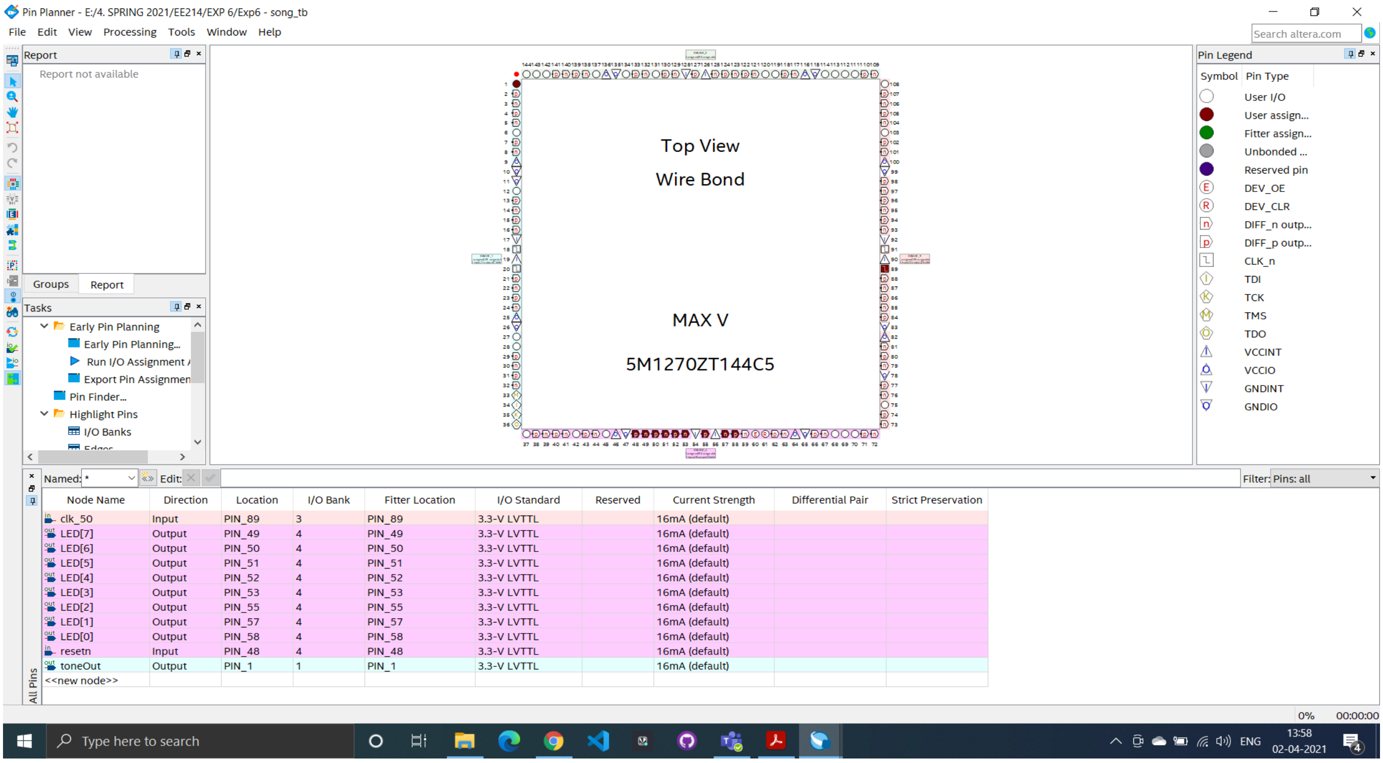
Handmade State Transition Graph



Quartus State Transition Graph



Krypton board

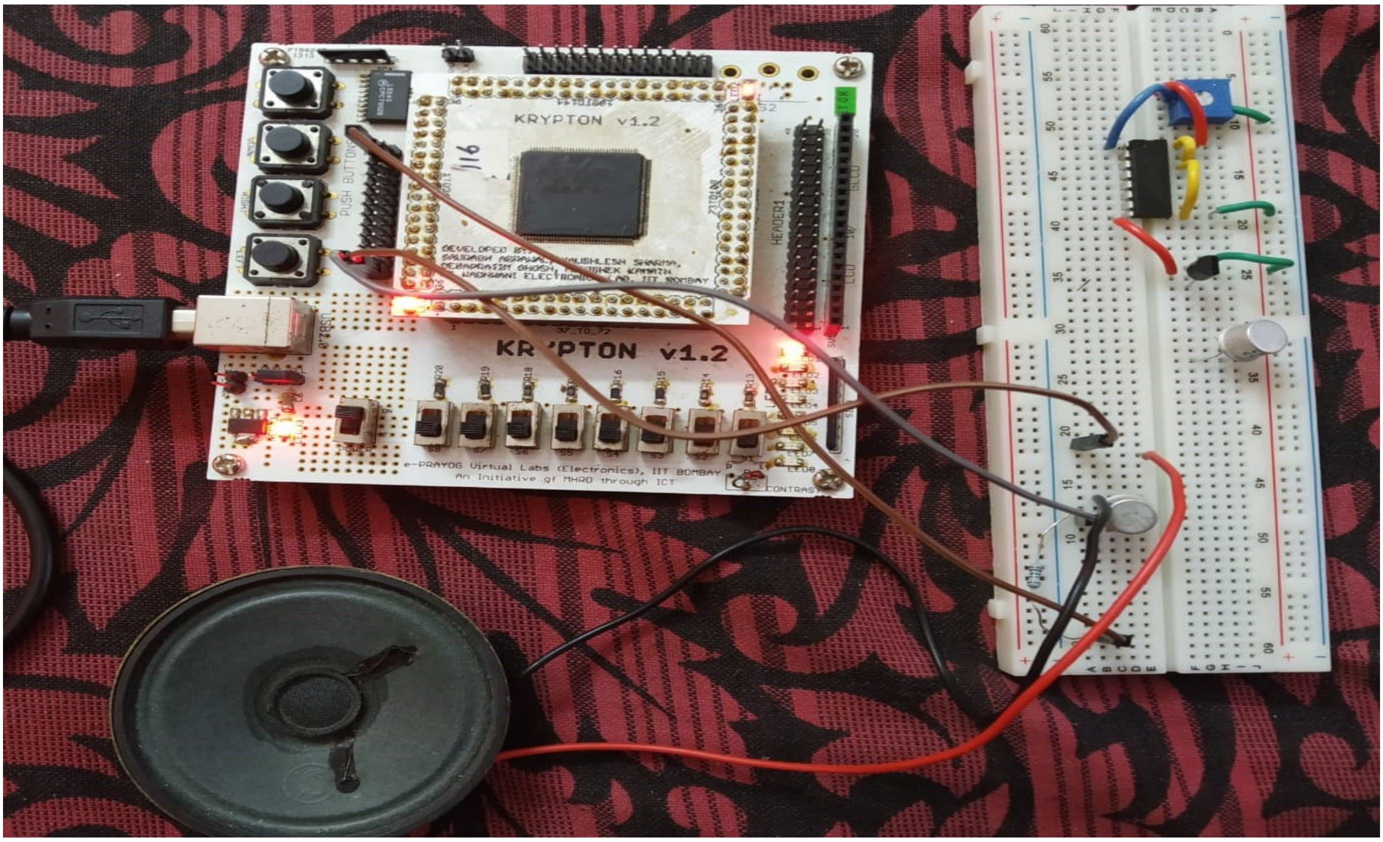


Pin Planner - E:/4. SPRING 2021/EE214/EXP 6/Exp6 - song_tb

Node Name	Direction	Location	I/O Bank	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair	Strict Preservation
clk_50	Input	PIN_89	3	PIN_89	3.3-V LVTTL		16mA (default)		
LED[7]	Output	PIN_49	4	PIN_49	3.3-V LVTTL		16mA (default)		
LED[6]	Output	PIN_50	4	PIN_50	3.3-V LVTTL		16mA (default)		
LED[5]	Output	PIN_51	4	PIN_51	3.3-V LVTTL		16mA (default)		
LED[4]	Output	PIN_52	4	PIN_52	3.3-V LVTTL		16mA (default)		
LED[3]	Output	PIN_53	4	PIN_53	3.3-V LVTTL		16mA (default)		
LED[2]	Output	PIN_55	4	PIN_55	3.3-V LVTTL		16mA (default)		
LED[1]	Output	PIN_57	4	PIN_57	3.3-V LVTTL		16mA (default)		
LED[0]	Output	PIN_58	4	PIN_58	3.3-V LVTTL		16mA (default)		
resetn	Input	PIN_48	4	PIN_48	3.3-V LVTTL		16mA (default)		
toneOut	Output	PIN_1	1	PIN_1	3.3-V LVTTL		16mA (default)		
<<new node>>									

Pin Planner

Observations



KRYPTON v1.2
116
PUSH BUTTONS
HEADER1
e-PRAYOG Virtual Labs (Electronics), IIT BOMBAY
An Initiative of MHRD through ICT

As visible in the picture, reset button was OFF, music was playing on the speaker and LED were glowing according to the Note being played.

References

Experiment 5 : Tone Generator
Clock Divider Tutorial by Teaching Assistant Mr. Sandesh Goyal
Quartus 20.1 Lite Software
EE214_Music_Synthesizer.pdf