

FPGA PROJECT -

TONE GENERATOR USING DIFFERENT FREQUENCIES

BY :- SNEH SHAH (22BEC121)

INTRODUCTION:

- In the evolving world of digital electronics, Field-Programmable Gate Arrays (FPGAs) have emerged as important elements in the design and implementation of various digital systems, due to their versatility, reconfigurability, and high-performance capabilities.
- Among the many applications, the creation of tone generators using different frequencies showcases a majestic intersection of sound engineering and digital technology. This report deals with the design, implementation, and analysis of an FPGA-based tone generator, a device engineered to produce audio tones by generating waveforms at specific frequencies.



KEYWORDS

- 1. FPGA(Field-Programmable Gate Array)
- 2. DE2 Board
- 3. Waveform
- 4. Frequency
- 5. Amplitude
- 6. Digital Signal Processing (DSP)

- 1. Tone Generation
- 2. Polyphonic Sounds
- 3. Hardware and Software Integration
- 4. Audio Signal Processing
- 5. Pitch
- 6. Sound Synthesis

STATE OF THE ART TECHNOLOGY

State-of-the-art technology for Polyphonic Tone Generation on FPGAs includes:

1. High-performance FPGAs with dedicated DSP blocks.
2. High-Level Synthesis (HLS) tools for C/C++ code synthesis.
3. Custom IP cores for audio processing tasks.
4. Real-time operating systems (RTOS) for task scheduling.
5. Advanced audio codecs for high-fidelity input/output.
6. Open-source audio libraries for synthesis algorithms.
7. Machine learning for sound modeling.
8. Simulation and verification tools for design validation.
9. Cloud-based FPGA platforms for scalable deployment.

Some Indian companies using advanced FPGA technology for audio processing include Xilinx India, Intel India (Programmable Solutions Group), Texas Instruments India, Analog Devices India, and various startups and research institutions focusing on audio processing with FPGAs.

Some Indian speaker making companies that may use advanced technologies like FPGA for audio processing include Boat Lifestyle, JBL India (Harman International), Zebronics , Creative Technology, and Tranquil Audio.

LIMITATIONS AND DRAWBACKS

01

Complexity

02

Cost

03

Power
Consumption

04

Time
Consumption

05

Resource
Constraints

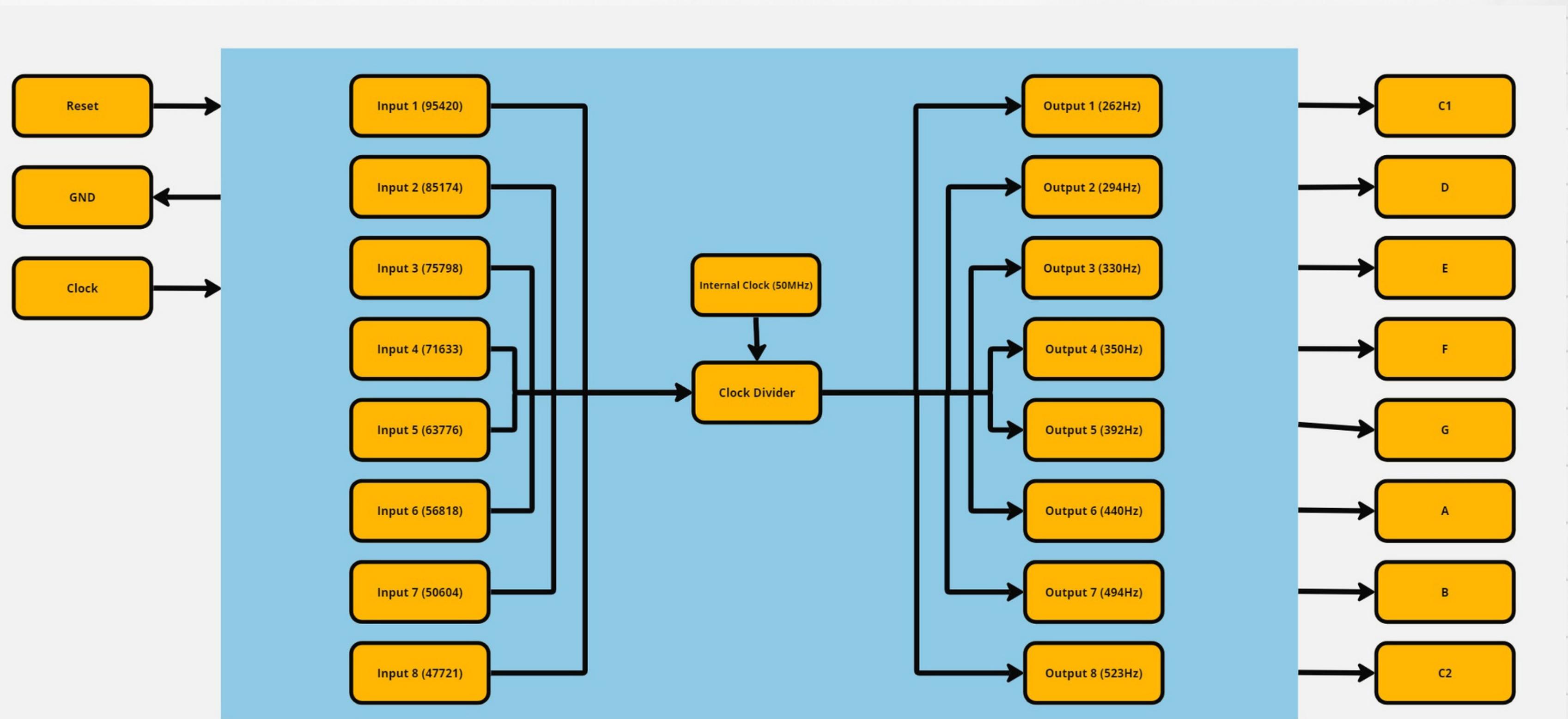
06

Latency

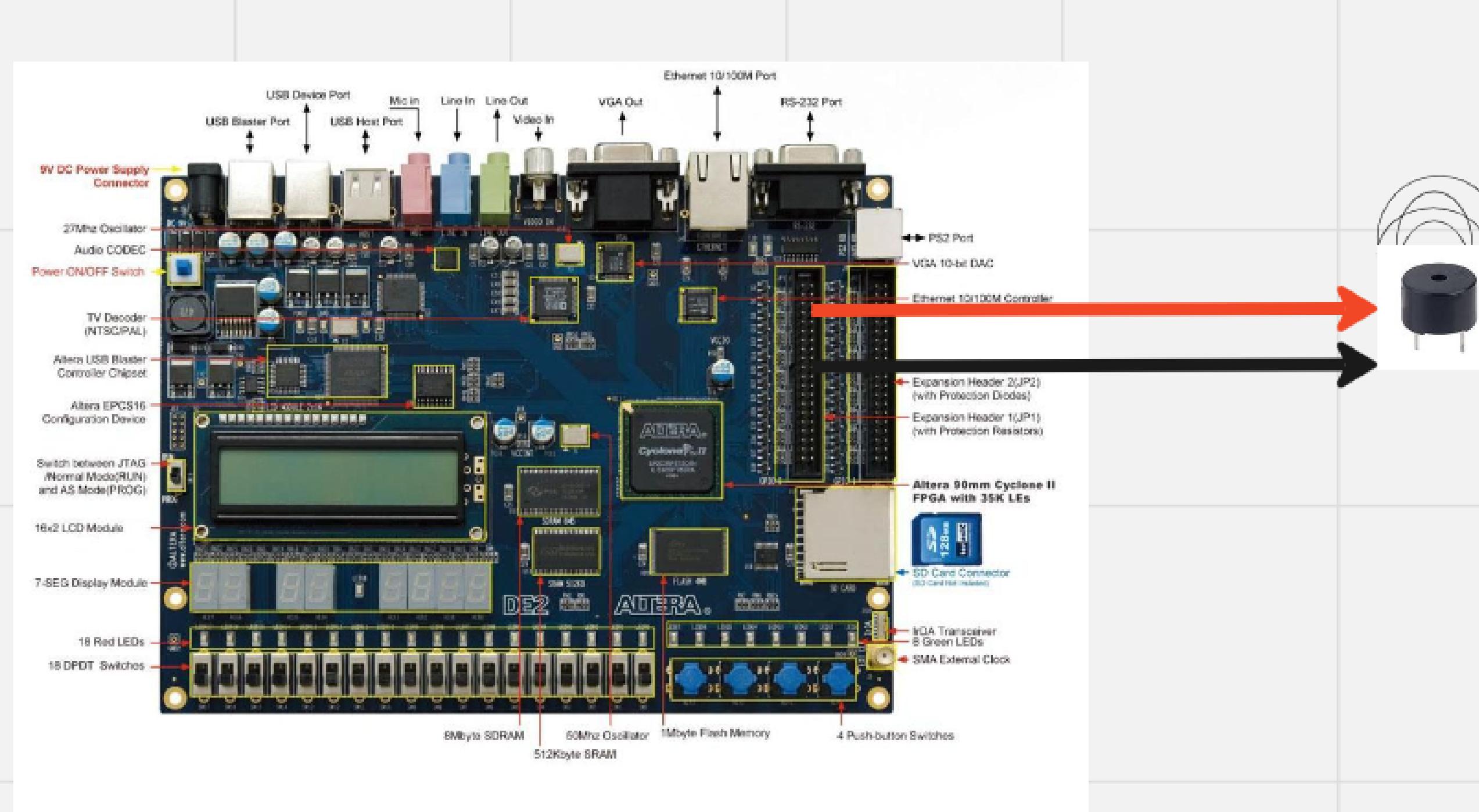
PROPOSED SOLUTIONS

- Modular Design and Simulation: Break the system into smaller parts for easier management and use simulation tools to validate each module before hardware implementation.
- High-Level Synthesis (HLS): Utilize HLS tools to write code in higher-level languages like C, which are then converted to FPGA logic which simplifies the design process.
- Resource Optimization: Apply techniques like time-multiplexing of DSP blocks and efficient coding practices to make the best use of FPGA.
- Power Management: Implement power-efficient designs to estimate and optimize power usage.
- Use of IP Cores: Apply pre-designed IP cores for common audio functions to reduce development time and ensure efficiency.
- Latency Management: Design with low latency in mind by employing pipelining and parallel processing to meet real-time requirements.

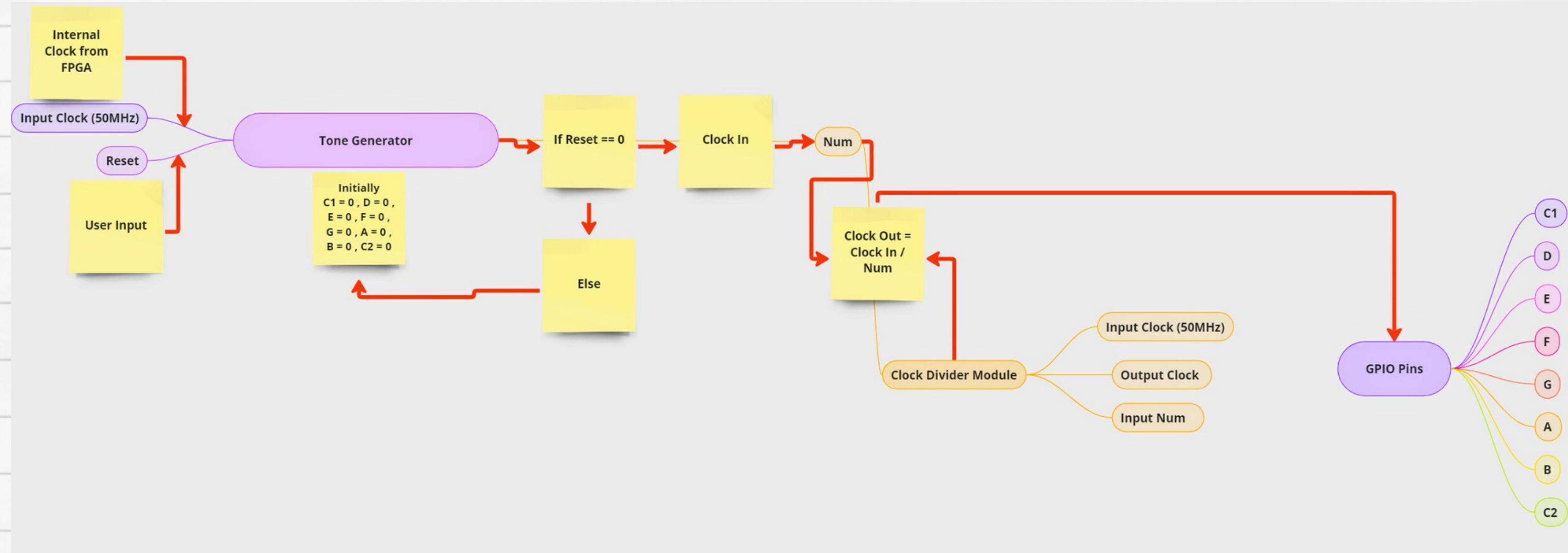
BLOCK DIAGRAM



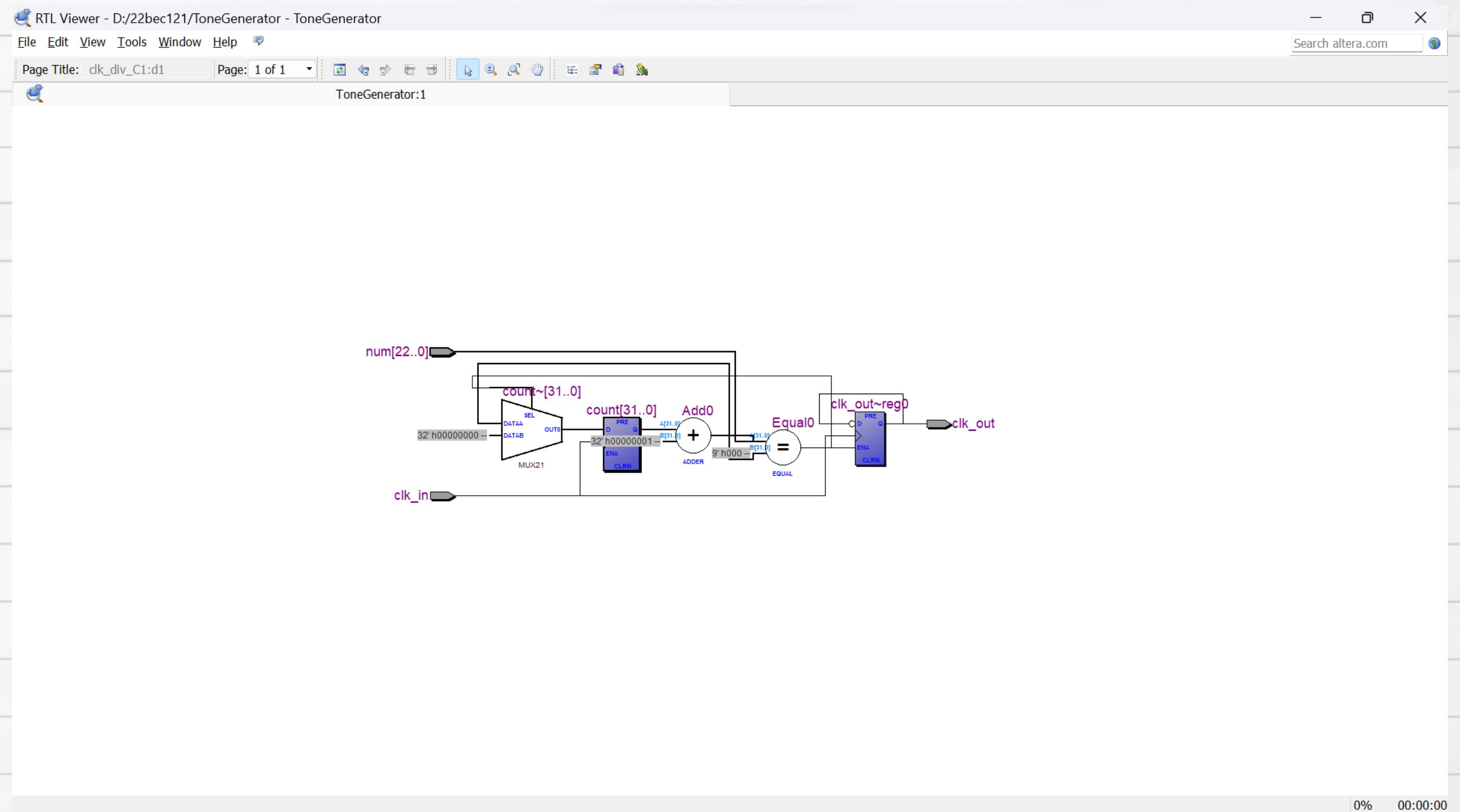
CIRCUIT DIAGRAM



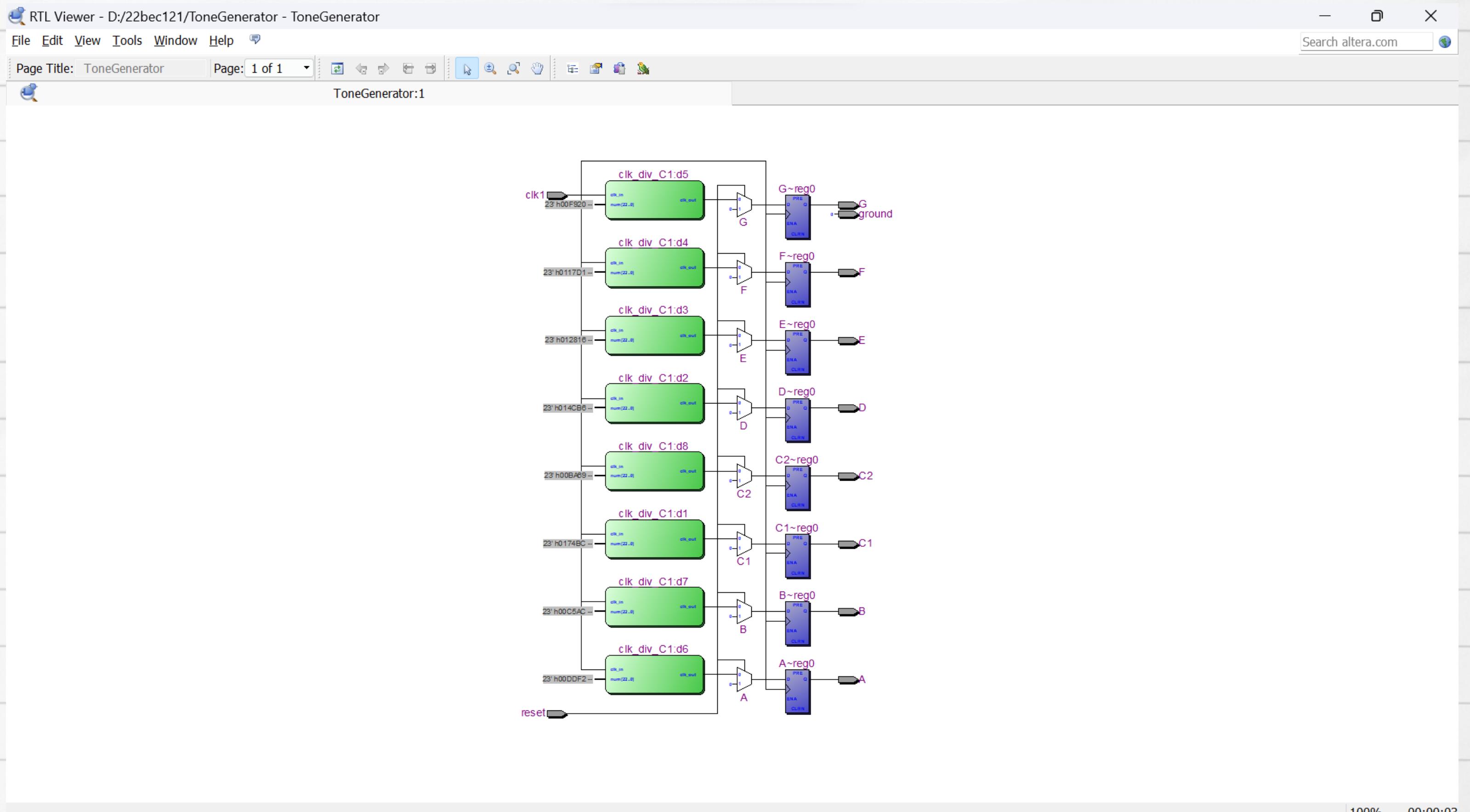
FLOW CHART



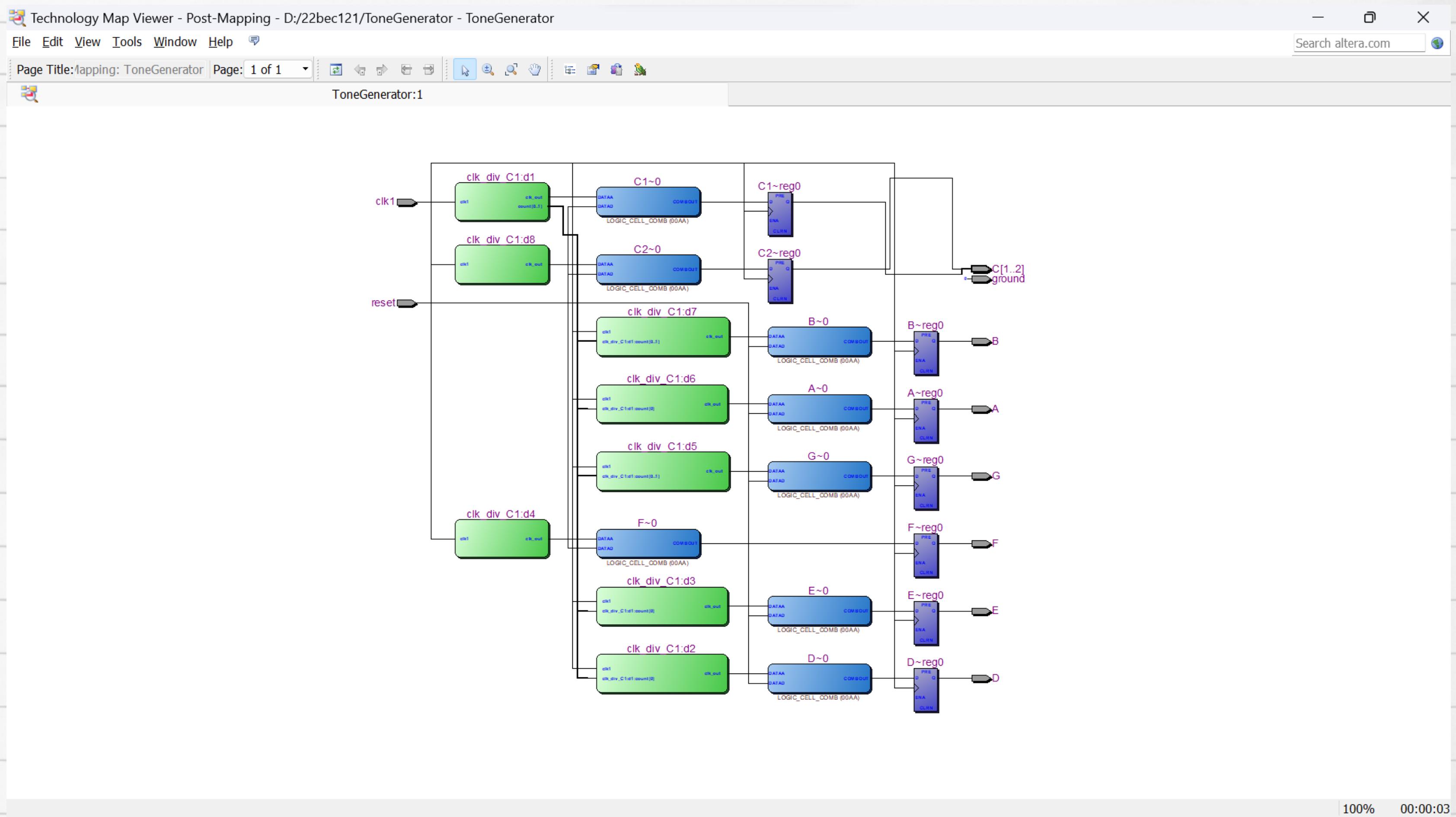
RTL VIEW (CLOCK DIVIDER MODULE)



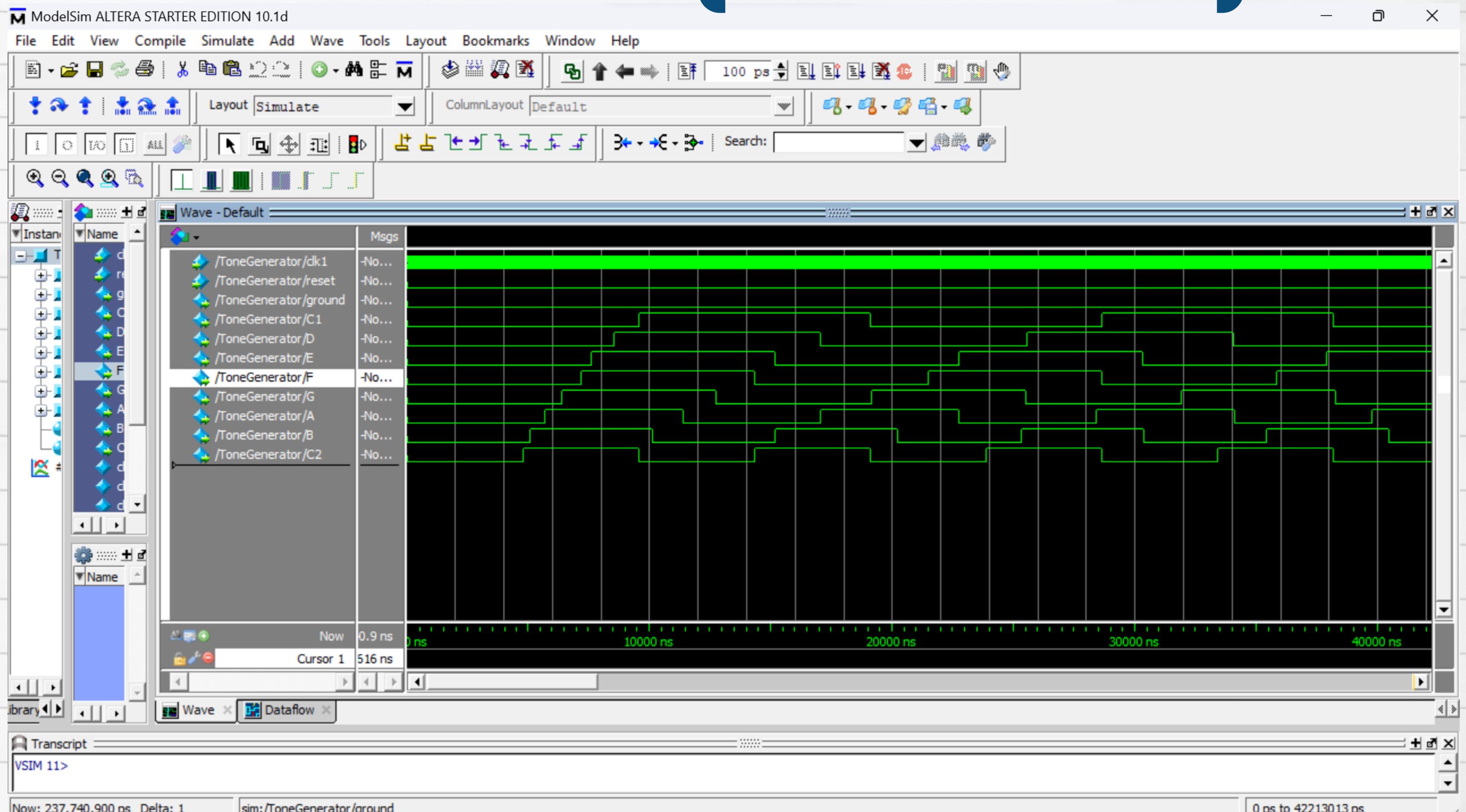
RTL VIEW (TONE GENERATOR MODULE)



TTL VIEW



SIMULATION (WAVEFORMS)



CONCLUSION

- In this project , I learnt a lot of new things , i.e. Clock Divider , Square Wave Generator , Generate different frequencies and finally Polyphonic Signals and their generation .
- This project also helped me to gain knowledge about the FPGA board and its power to do many different things .

REFERENCES

- https://www.academia.edu/7912472/Audio_Tone_Generator_Using_Verilog_HDL_Coding_Implementation_of_Audio_Tone_Generator_on_FPGA_Using_Verilog_HDL_Coding
- <https://www.fpga4fun.com/MusicBox.html>
- <http://www.innovateasia.com/asia/download/articles/2013/cn147.pdf>
- <https://www.youtube.com/watch?v=AANKGfbFYlk>

CODE

Quartus II 64-Bit - D:/22bec121/TG123 - TG123

File Edit View Project Assignments Processing Tools Window Help

Project Navigator

Cyclone II: EP2C35F67

TG123

```

1 module TG123 (
2     input clk1 ,
3     input reset ,
4     output ground ,
5     output reg C1 , D , E , F , G , A , B , C2
6 );
7
8     reg [15:0]n ;//1000Hz
9     reg [15:0]div ;
10    wire clk2 ; wire clk3 ; wire clk4 ; wire clk5 ; wire clk6 ; wire clk7 ; wire clk8 ; wire clk9 ;
11
12    clk_div_C1 d1(.clk_in(clk1) , .num(95785) , .clk_out(clk2)) ; //SA 0 //261Hz
13    clk_div_C1 d2(.clk_in(clk1) , .num(85324) , .clk_out(clk3)) ; //RE 1 //293Hz
14    clk_div_C1 d3(.clk_in(clk1) , .num(75987) , .clk_out(clk4)) ; //GA 2 //329Hz
15    clk_div_C1 d4(.clk_in(clk1) , .num(71633) , .clk_out(clk5)) ; //MA 3 //349Hz
16    clk_div_C1 d5(.clk_in(clk1) , .num(63776) , .clk_out(clk6)) ; //PA 4 //392Hz
17    clk_div_C1 d6(.clk_in(clk1) , .num(56818) , .clk_out(clk7)) ; //DHA 5 //440Hz
18    clk_div_C1 d7(.clk_in(clk1) , .num(50709) , .clk_out(clk8)) ; //NI 6 //493Hz
19    clk_div_C1 d8(.clk_in(clk1) , .num(47801) , .clk_out(clk9)) ; //SAA 7 //523Hz
20
21 initial
22 begin
23     n = 440 ;
24     div = 25000000 / n ;
25     C1 = 0 ; D = 0 ; E = 0 ; F = 0 ; G = 0 ; A = 0 ; B = 0 ; C2 = 0 ;
26
27     $display("General Count = %d" , div) ;
28 end
29
30 always @ (posedge clk1)
31 begin

```

Hierarchy

Tasks

Flow: Cor Customize...

Compile Design Analysis & S

Messages

All TD Message

'vme Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings
 293000 Quartus II Full Compilation was successful. 0 errors, 16 warnings

System (4) Processing (101)

Ln 24 Col 30 Verilog

Quartus II 64-Bit - D:/22bec121/TG123 - TG123

File Edit View Project Assignments Processing Tools Window Help

Project Navigator

TG123

```

begin
if(reset)
begin
    C1 = 0 ; D = 0 ; E = 0 ; F = 0 ; G = 0 ; A = 0 ; B = 0 ; C2 = 0 ;
end
else if (!reset)
begin
    C1 = clk2 ; D = clk3 ; E = clk4 ; F = clk5 ; G = clk6 ; A = clk7 ; B = clk8 ; C2 = clk9 ;
end
end
assign ground = 0 ;
endmodule

```

clk_div_C1

```

input clk_in ,
input [22:0]num ,
output reg clk_out
);

reg [31:0]count ; //17
initial
begin
    count = 0 ; clk_out = 0 ;
end
always @(posedge clk_in)
begin
    count = count + 1 ;

```

Messages

All TD Message

'vme Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings
 293000 Quartus II Full Compilation was successful. 0 errors, 16 warnings

System (4) Processing (101)

Flow: Cor Customize...

Compile Design Analysis & S

Messages

All TD Message

'vme Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings

System (4) Processing (101)

```

always @(posedge clk_in)
begin
    count = count + 1 ;
    if (count == num)
        begin
            clk_out = ~clk_out ;
            count = 0 ;
        end
    end
endmodule
//GND 12

```