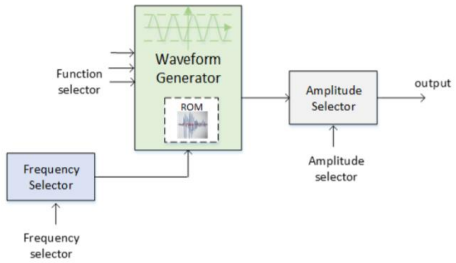
Experiment 3 – Frequency Generator

Mohadeseh
Azari,
810196404

***Abstract*— In this lab we will use Modelsim and Quartus II to build a frequency generator. we want to synthesis frequency generator using digital tools. For designing a frequency generator, we need to stablish three (3) main part; part1: wave generator. In this part we will make different kinds of waves like square, sine, saw tooth, ext., using Modelsim. Part2: frequency selector. In this part as in real function generators we want to build a FPGA to allow us to change the frequency. part3: amplitude selector. And finally simple but important part is ability to change the amplitude of our wave design.**

***Keywords*— wave function, counter, frequency**

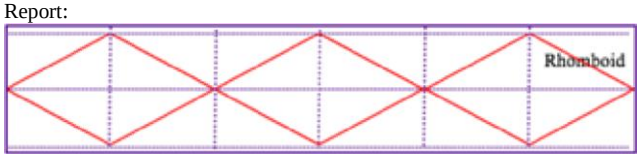
I. WAVEFORM GENERATOR



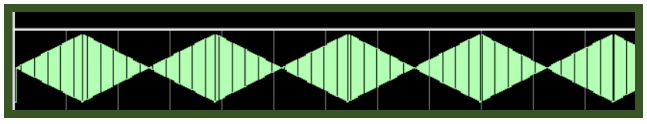
Block Diagram of an Arbitrary Function Generator(AFG)

This module is the heart of this design. Functions generated by this module have the fixed period of 256 clocks.

Report:

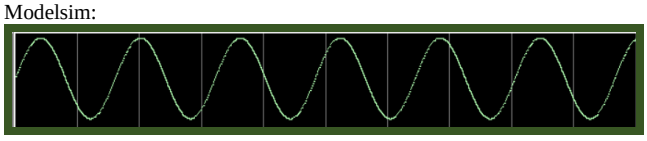


Modelsim:



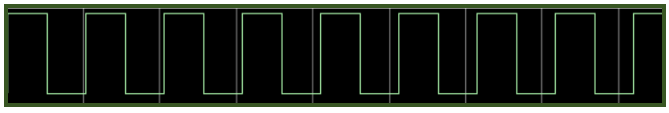
Report:

Modelsim:

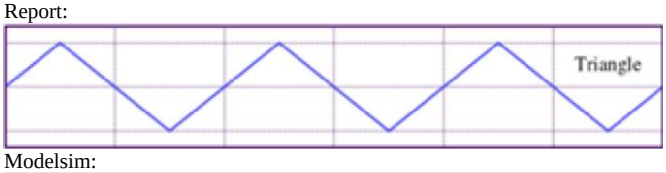


Report:

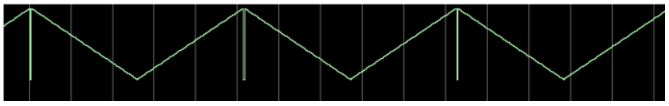
Modelsim:



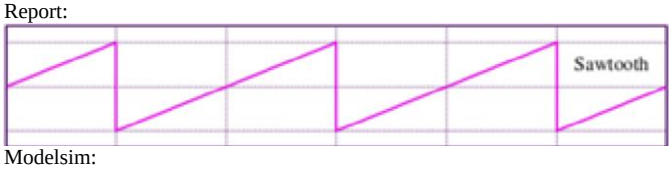
Report:



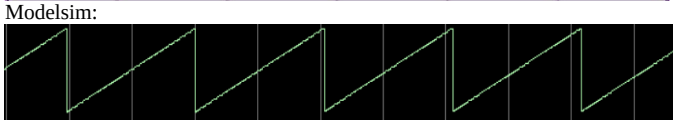
Modelsim:



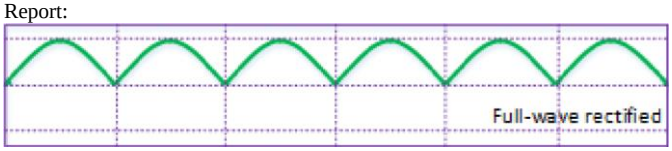
Report:



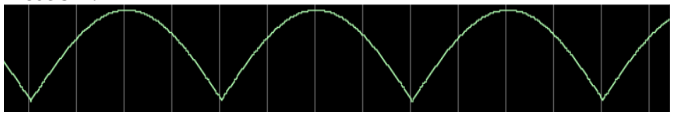
Modelsim:



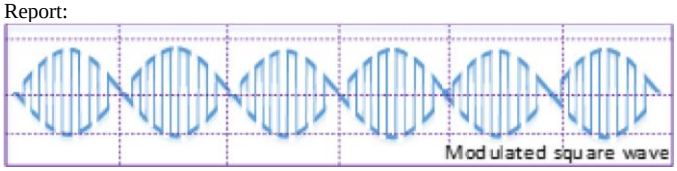
Report:



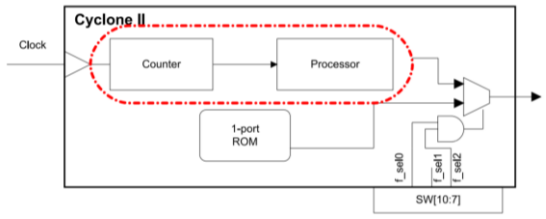
Modelsim:



Report:

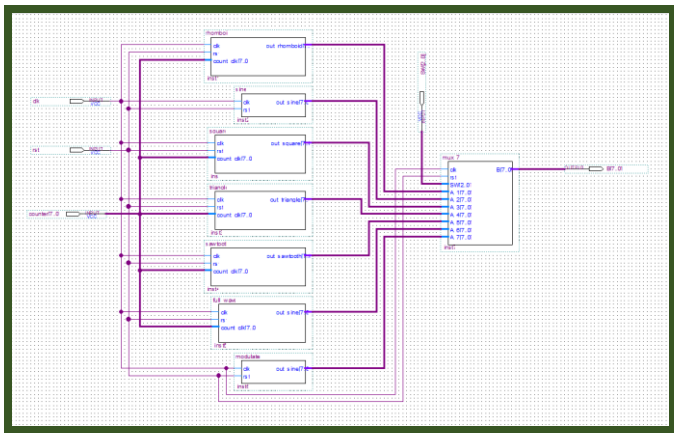


Modelsim:



Block Diagram of Waveform Generator

Processor:

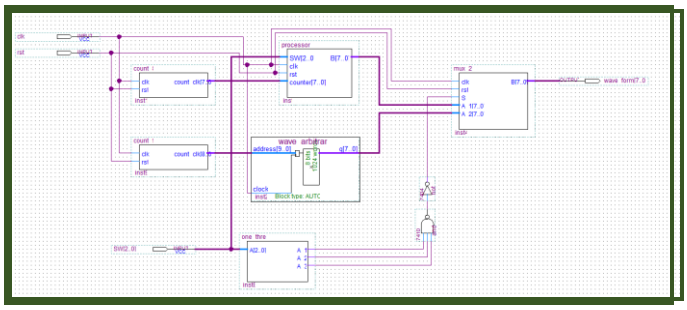


Function Selection:

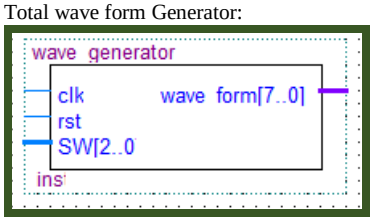
func[2:0]	Function
3'b000	Rhomboid
3'b001	Sine
3'b010	Square
3'b011	Triangle
3'b100	Saw-tooth
3'b101	Full-wave rectified
3'b110	Modulated sine wave
3'b111	Arbitrary

```
case(SW)
  3'b000 : B<=A_1;    //Rhomboid
  3'b001 : B<=A_2;    //Sine
  3'b010 : B<=A_3;    //Square
  3'b011 : B<=A_4;    //Triangle
  3'b100 : B<=A_5;    //Saw-tooth
  3'b101 : B<=A_6;    //Full-wave rectified
  3'b110 : B<=A_7;    //Modulated sine wave
endcase
```

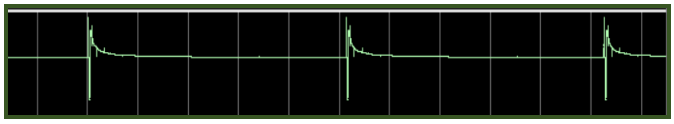
Wave Generator:



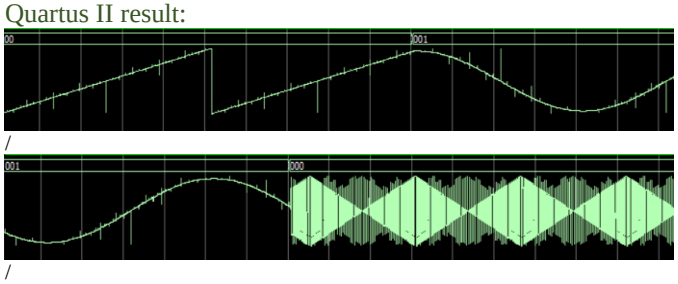
Total wave form Generator:



Arbitrary wave:
With 50MHz ☺



Quartus II result:

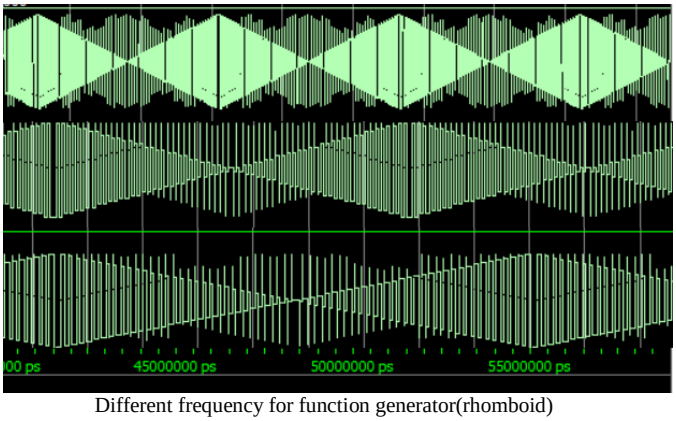


/

/

As you can see when the selecting signal SW [10:8] is changing from 3'b000 to 3'b001 the result will change from saw tooth to sine wave and as it changes to 3'b 000 our wave will be rhomboid.

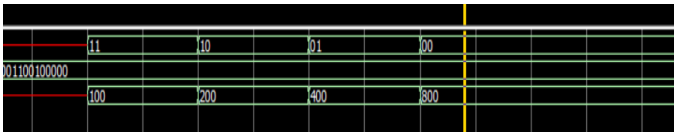
II. FREQUENCY SELECTOR



Different frequency for function generator(rhomboid)

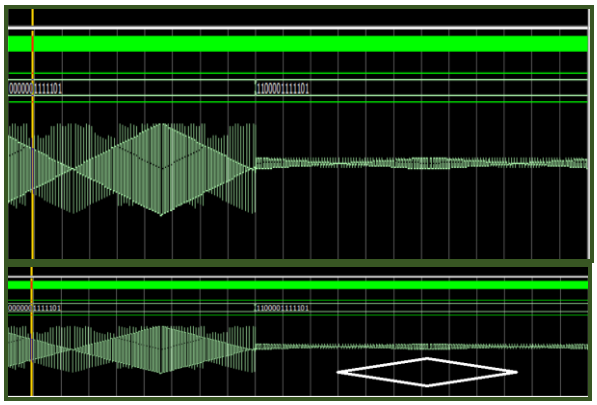
The responding frequencies are: 1. 50MHz
2. 400KHz
3. 200KHz

III. AMPLITUDE SELECTOR



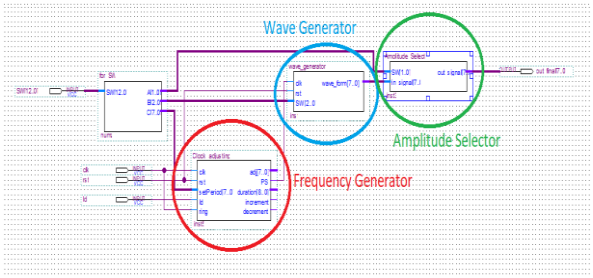
Here is a very small test bench to see the selection values and their results. As you can see they divide the input signal to the order of 100,200,400&800.

SW[12:11]	Amplitude
2'b00	1
2'b01	2
2'b10	4
2'b11	8



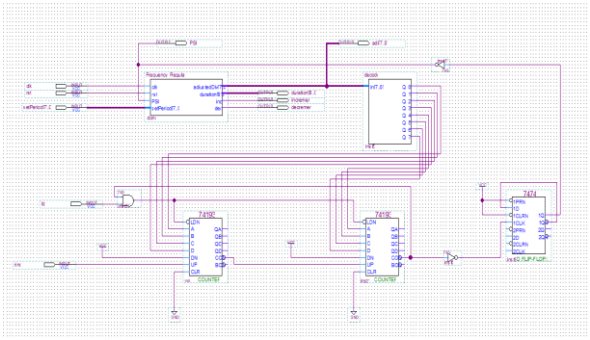
Here is another picture of Quartus II result in Modelsim

And you can see by the time the SW [12:11] changes with the clock from 2'b00 into 2'b11 the amplitude of our signal in scale down by a factor of 8.



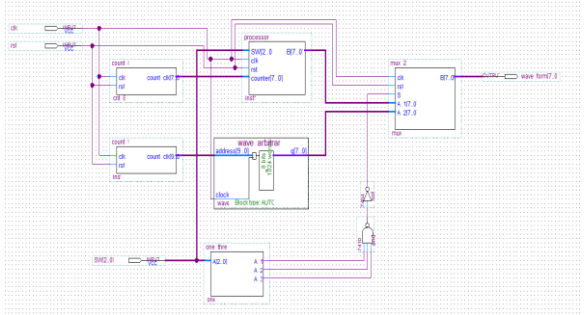
Total Design

Zoom in:



This is Frequency Generator

Zoom in:



This is wave form Generator

Zoom in:

```
1  `timescale 1ns/1ns
2  module Amplitude_Selector(input [1:0]SW,input[7:0] in_signal,output[7:0] out_signal);
3    wire [3:0]Divisor;
4    //multiplexer
5    assign Divisor = SW[1] ? (SW[0] ? 4'b1000 : 4'b0100) : (SW[0] ? 4'b0010 : 4'b0001);
6    //Dividor
7    assign out_signal=in_signal/Divisor;
8  endmodule
```

This is Amplitude Selector Code

IV. CONCLUSION

The important conclusion of this lab in my opinion is the ability to synthesis any complex circuit or any tools with simple modules and FPGAs using Modelsim and Quartus II. In the last previous labs, we see how we can easily build clock using different kinds of methods also we synthesis a frequency regulator in Quartus II using simple modules of lab 1 and its primitives and now in this particular lab we saw how simple synthesising a function generator is. Although the results are not analogue and we are not using resistors and capacitors but the learning aspects are always the same.