Experiment #3

Function Generator

Kamyar Rahmani

810199422

Mohammad Mashreghi

810199492

***Abstract*—In this document, we are going to design an Arbitrary function generator (AFG) with different waves, frequencies, and amplifier, and after that we are going to use Quartus and then use FPGA and uploaded it.**

***Keywords*— Waveform Generator, Frequency, ROM, Selector, Amplitude Selector, PWM, DDS, Counter, Function Generator, Arbitrary Function Generator**

I. INTRODUCTION

In this document, we are going to design an Arbitrary Generator that consists of awaveform generator, Amplitidude selector, DAC, and frequency selector.

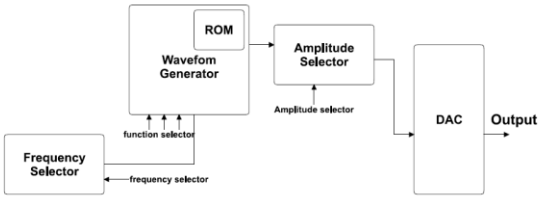

Fig. 1 Block diagram of the Arbitrary Generator

II. WAVEFORM GENERATOR

In Waveform generator we designed 6 different weveforms and we use a ROM for showing another waveform that we've got before.

func[2:0]	Function
3'b000	Reciprocal
3'b001	Square
3'b010	Triangle
3'b011	Sine
3'b100	Full-wave rectified
3'b101	Half-wave rectified
3'b111	DDS

Fig. 2 Function selection

For generating Square, Triangle, and Reciprocal we use a 8-bit counter inside the wave generator

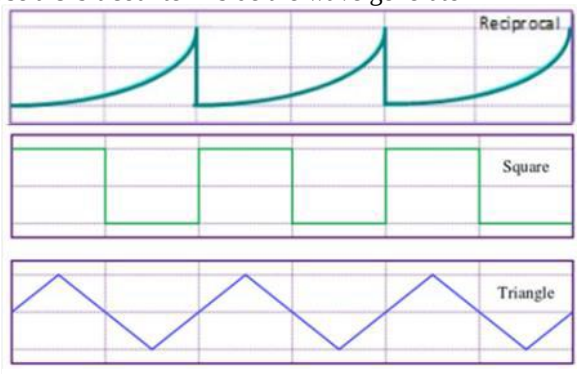

Fig. 3 wave of Square, Triangle and Reciprocal

And for the next three waves, at first we need to know how to create sin and after that we create full and half wave rectified.

For generating sin we use 2nd ODE formula and waveform clock as you can see:

$$\sin(n) = \sin(n-1) + \frac{1}{64}\cos(n)$$

$$\cos(n) = \cos(n-1) + \frac{1}{64}\sin(n)$$

$$Initial\ Value: \sin(0) = 0\ \&\ \cos(0) = 3000$$

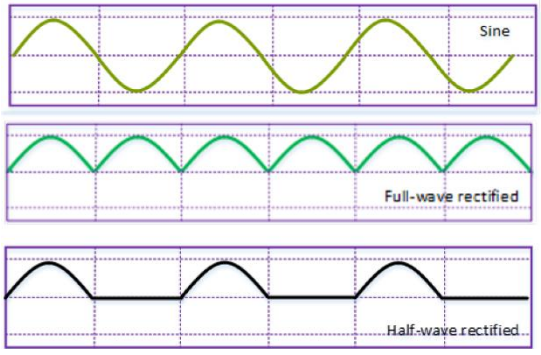

Fig. 3 wave of sin, half wave rectified and full wave rectified

Testbench result:

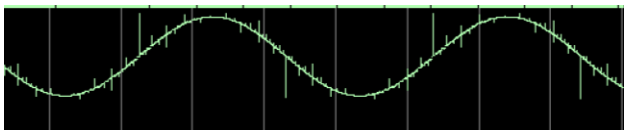

Fig. 4 wave of sin in testbench

Fig. 5 wave of half wave rectified in testbench

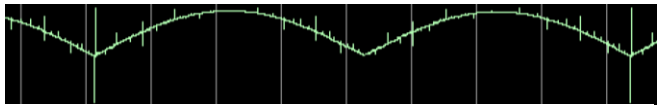

Fig. 6 wave of full wave rectified in testbench

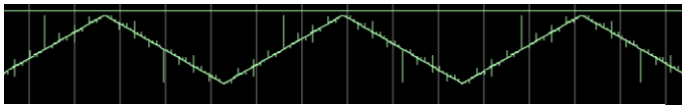

Fig. 7 wave of triangle in testbench

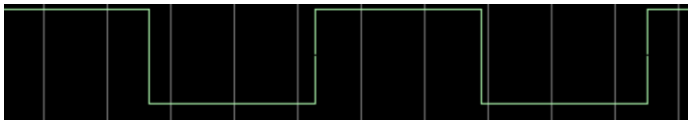

Fig. 8 wave of square in testbench

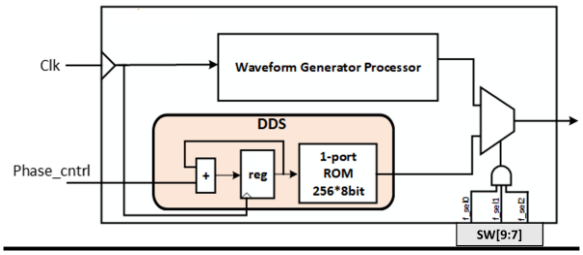

Fig. 9 Block diagram of waveform generator

Another way to generate wave is Direct Digital Synthesis (DDS) in this method we use adder , reg and ROM that used saved data and we can change its period with phase cntrl.

III. PWM

In this part we need a digital to analog conversion (DAC) and one of the cheapest methode is Pulse Width Modulation that we use and it use 8-bit counter inside it.

- When the input signal is larger than counter the output is **1.**
- When the input signal is less than counter the output is **0.**

$$duty\ cycle = \frac{T_{on}}{T_{on} + T_{off}}$$

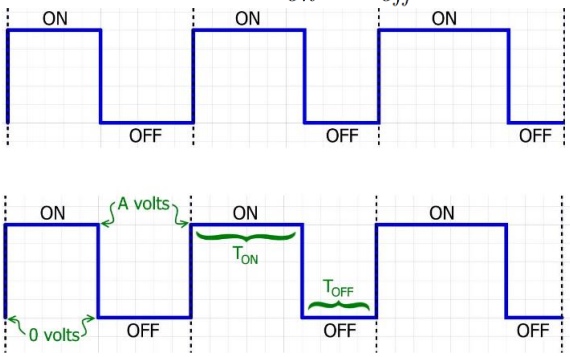

Fig. 10 Pulse Width Modulation (PWM)

IV. FREQUENCY SELECTOR

Arbitrary Generator usually have frequency selector to change its wave frequency according to the need of user.

In this part we use 9-bit counter and we set the first three bit for the change frequency and other bits our define before.

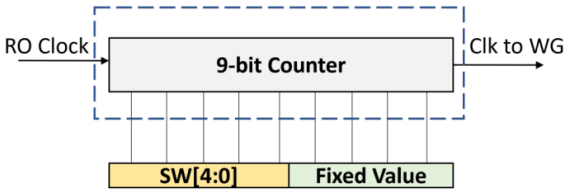

Fig. 11 Block diagram of frequency selector

V. AMPLITUDE SELECTOR

Another option of Arbitrary Generator is a amplitude selector that scale down the amplitude of the waveforms.

SW[6:5]	Amplitude
2'b00	1
2'b01	2
2'b10	4
2'b11	8

Table 2: Amplitude selection

VI. ACTIONS

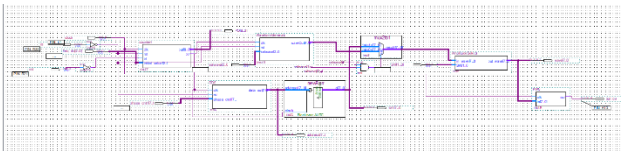

Fig. 12 Final block diagram of design

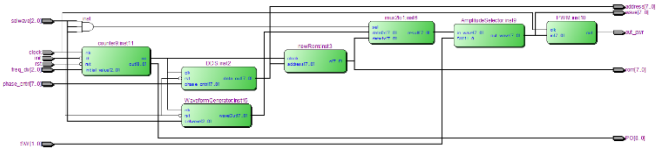

Fig. 13 RTL view

Flow Summary	
Flow Status	Successful - Tue Jan 03 20:45:56 2023
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	EXP3
Top-level Entity Name	block_diagram
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	284 / 18,752 (2 %)
Total combinational functions	276 / 18,752 (1 %)
Dedicated logic registers	99 / 18,752 (< 1 %)
Total registers	99
Total pins	44 / 315 (14 %)
Total virtual pins	0
Total memory bits	2,048 / 239,616 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

Fig. 14 Flow Summary

VII. TOTAL DESIGN

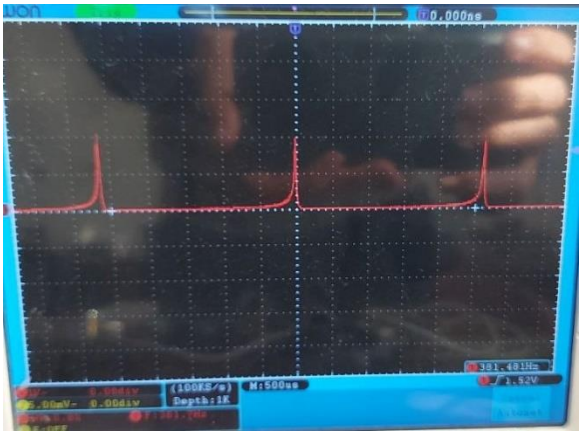

Fig. 15 Reciprocal

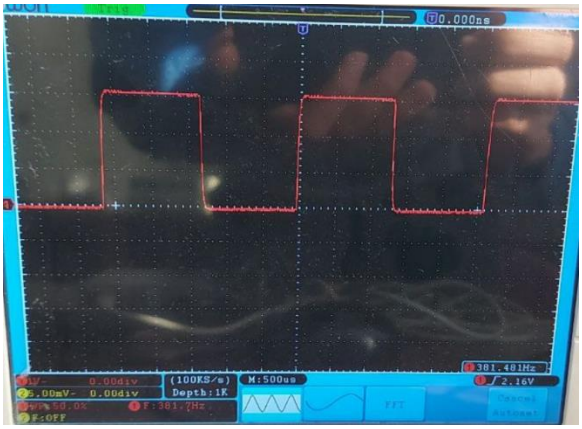

Fig. 16 Square

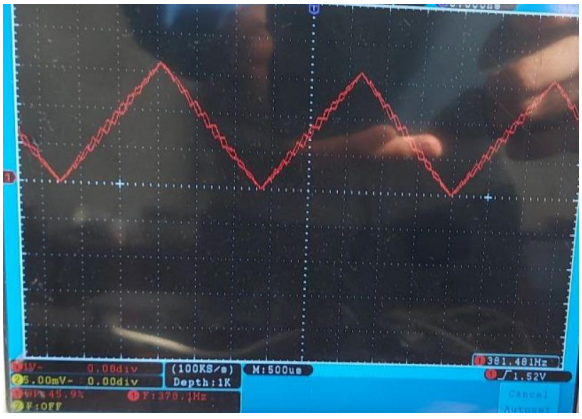

Fig. 17 Triangle

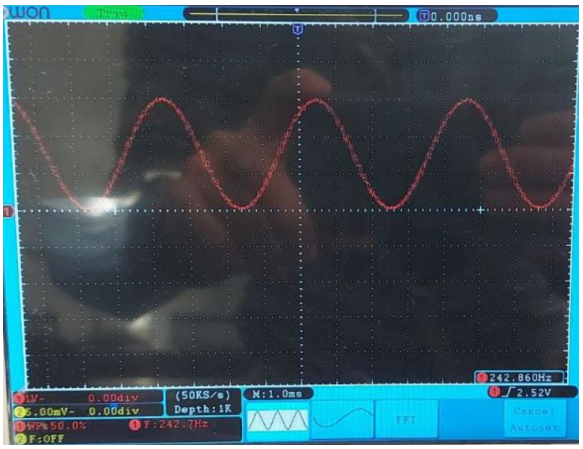

Fig. 18 Sine

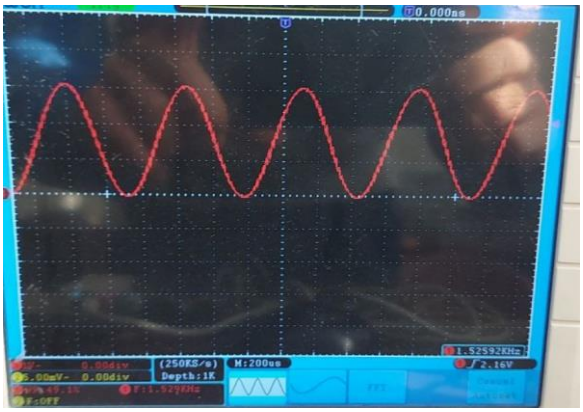

Fig. 21 DDS phase = 0

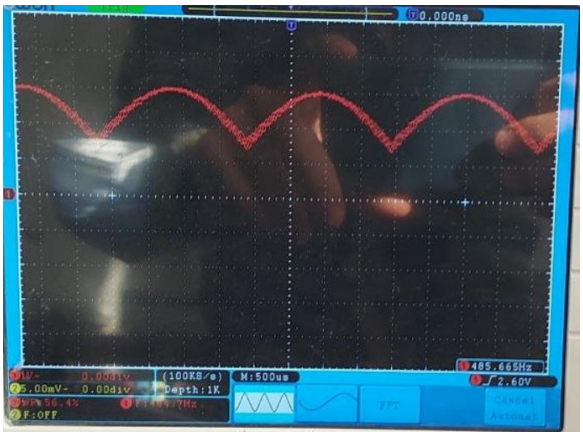

Fig. 19 Full Wave

Fig. 22 DDS phase = 1

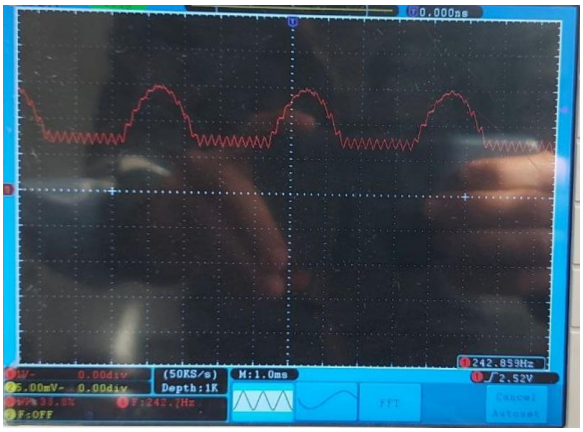

Fig. 20 Half Wave

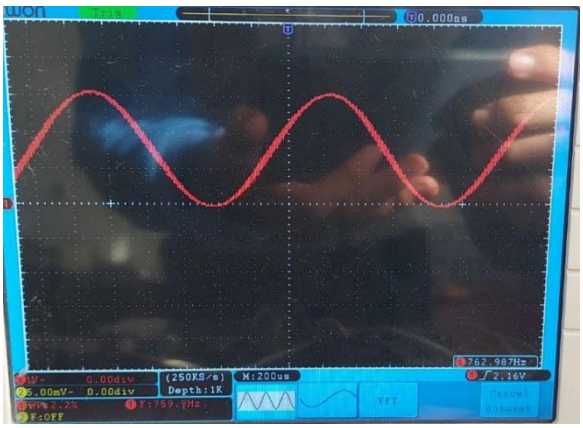

Fig. 23 DDS phase = 2

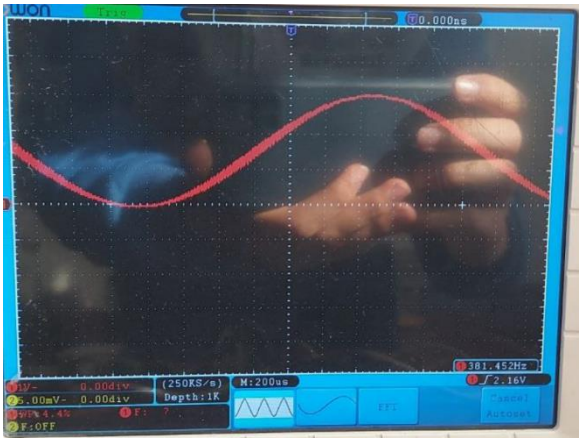

Fig. 24 DDS phase = 3

VII. **Conclusion**

In this experiment, we programmed a **Waveform Generator** with ModelSim and Quartus II and we designed different waves and change their domain and frequency, at last, we created a PWM to convert digital to analog.