RESEARCHERID
THOMSON REUTERS

[Potharaju, 4(11): November 2017]
DOI- 10.5281/zenodo.1067984

ISSN 2348 – 8034
Impact Factor- 4.022

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

FPGA IMPLEMENTATION OF DIRECT DIGITAL SYNTHESIZERUSING VHDL

Vidya Sagar Potharaju*
*Associate Professor, Department of Electronics and Communication Engineering,
Vignana Bharathi Institute of Technology, Hyderabad, Telangana,INDIA

ABSTRACT

Signal generators are heavy and large in size and are limited to particular set of analog wave forms,creation of arbitrary wave forms are not possible. The available Digital signal generators nowadays are incapable of creating all type of waveforms and more ever they are not reconfigurable. In this paper I am proposing an efficient method called Direct Digital Synthesis (DDS) to realize all the hardware parts of signal generator called Direct Digital Synthesizer in FPGA using VHSIC Hardware Description Language (VHDL). DDS has many advantages over its analog counterpart and improved phase noise. It has precise control of the output phase across frequency switching transitions.The output of FPGA will be digital and this digital data is given to Digital-to-Analog Convertor (DAC). DAC converts the digitally created signal to analog form thus creating a versatile, flexible signal generator. The DAC and FPGA share a single clock source. Since we are using FPGA as a hardware part, with simple alteration to FPGA design signal generator upgradation is possible without any additional cost by altering the FPGA design. It is used in all sorts of places such as frequency hopping, signal synthesis, medical imaging systems, radio receivers, Phase Locked Loop (PLL), test equipment.

Keywords: *Direct Digital Synthesis (DDS),VHSIC Hardware Description Language (VHDL), Digital-to-Analog Convertor (DAC), Phase Locked Loop (PLL), Field Programmable Gate Array (FPGA)..*

I. INTRODUCTION

DDS is a method of generating timing signals from a clock source with programmable frequency.Hence this can be easily implemented in FPGA using VHDL.The DDS structure was first proposed in 1971 by Tierney, Josepf, Charles Rader, and Bernard Gold.In this paper, we are generating various waveforms such as square wave, triangle wave, saw tooth wave, sine wave. The most critical part is the generation of sine wave.
Basic block diagram contains three main units (**Figure 1**):

1. Clock Source
2. FPGA
3. DAC

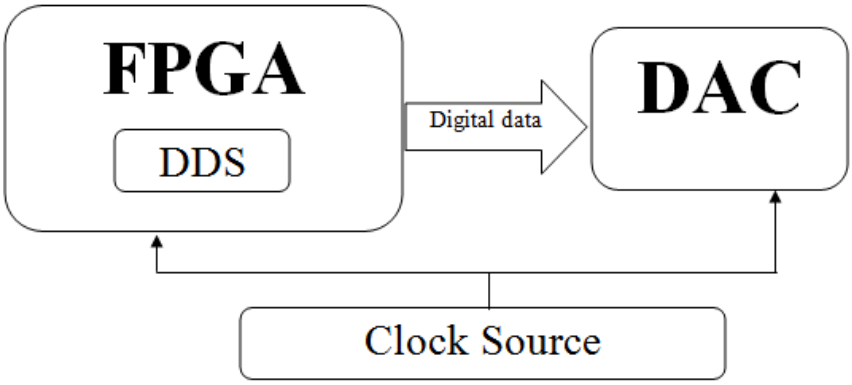

Figure 1:Basic Block Diagram

ISSN 2348 – 8034
Impact Factor- 4.022

The implementation of DDS is divided into two distinct parts as shown in **Figure 2,** a discrete time phase generator (the accumulator) outputting a phase value ACC, and a phase to waveform converter outputting the desired DDS signal.

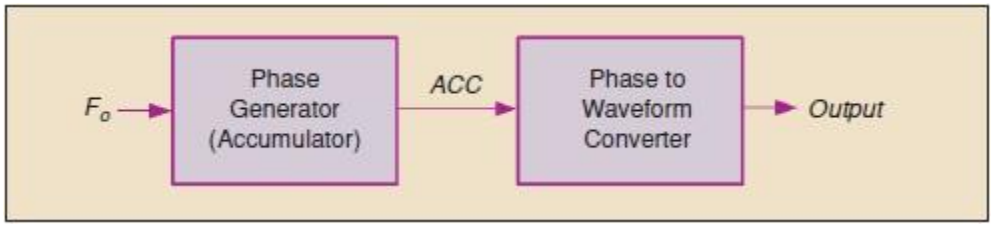

Figure 2:Fundamental Design

The accumulator is a running counter which stores the current phase value of the generated waveform. The rate at which the accumulator is updated and the accumulator increment value determine the frequency of the generated waveform. When the accumulator phase value reaches the maximum (360 degrees) it rolls over and starts again at 0 degrees. The current accumulator (phase) value is used to perform a lookup operation in a lookup table of the reference waveform to determine the next output value. The lookup table contains one cycle of the waveform to be generated and typically contains 1024 to 8192 sample points which represent the waveform. Because the accumulator value typically has a lot more resolution than the reference waveform, limited by the number of samples, the lookup operation may also perform an interpolation between two samples in the reference waveform. This is done based on the extra resolution in the accumulator value and returns a more accurate update value which provides better frequency control and less harmonic distortion in the generated signal.

Phase to waveform converter converts the phase value (count) into the wave form amplitude. This is also called phase to amplitude converter. This block will vary for each wave. Input to phase to waveform converter is phase 'count' (16 bit) and 'n' value. And the output is amplitude value 'wave_out' (8 bit).

II. BASIC DDS DESIGN

The basic method works as follows: (seeFigure 3 Basic DDS Design) The phase register is loaded with a value which is proportional to the required frequency. See Equation 1 Output Frequency. Then at each rising clock edge the phase register is added to the accumulator. After enough clocks and additions the accumulator rolls over and starts again from the remainder. That's all there is for the Numeric Control bit, the rest is output formatting. The top p bits of the accumulator are used to address a lookup table. The lookup table outputs q data bits to the DAC. Which generates the analog output. The anti-aliasing filter removes all the lumps and bumps, and is fixed at around a third the clock frequency.

Output Frequency*:*

$$F_{(OUT)} = \frac{(phase * F_{(clk)})}{(2^{(n)})} Hz \qquad \textbf{Equation 1}$$

Where n is the number of accumulator bits.

Frequency Resolution:

$$F_{res} = \frac{F_{(clk)}}{(2^{(n)})} Hz \qquad \textbf{Equation 2}$$

[Potharaju, 4(11): November 2017] ISSN 2348 – 8034
DOI- 10.5281/zenodo.1067984 Impact Factor- 4.022

Where n is the number of accumulator bits,with accumulator sizes of 32 bits or it is easy to get resolutions smaller than 1/2 Hz, even at high frequencies.

Max frequency:

$$F_{max} = \frac{F_{(clk)} * 2^{(m)}}{(2^{(n)})} Hz$$

Equation 3

Where m is the number of phase bits, n is the number of accumulator bits. For reasons of practical output filter design, the maximum output frequency is normally limited to Fclk/3.

This is so that the max frequency is within the lower passband of the of output anti-aliasing filter. At higher frequencies the signal gets distorted as the output filter peaks before transitioning into the stop band.If perfect filters were available Fclk/2 would be usable.

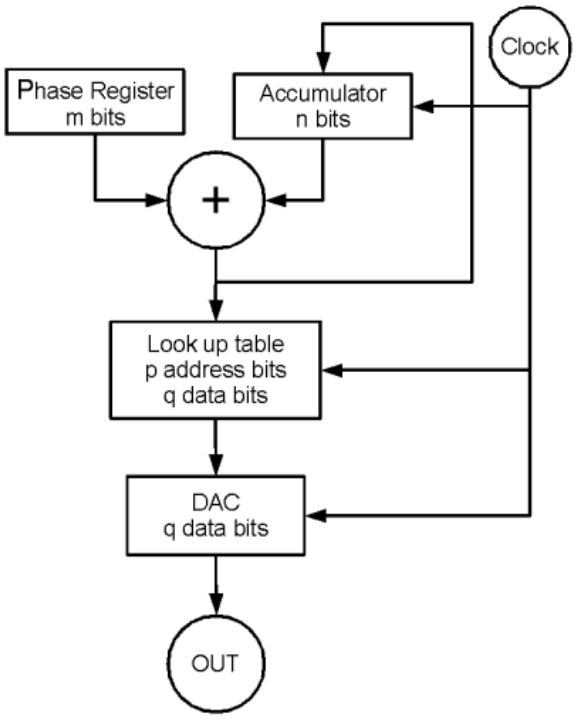

Figure 3: Basic DDS Design

Max Phase Jitter:
The phase jitter is basically determined by the frequency of the system clock. Although it can be reduced by a neat trick in the analog domain.

$$Jitter = \frac{1}{F_{(CLK)}} s$$

Equation 4

Signal to quantization noise (SQR):
This is basically a function of how many bits the DAC has and how much of the range is used.

[Potharaju, 4(11): November 2017] ISSN 2348 – 8034
DOI- 10.5281/zenodo.1067984 Impact Factor- 4.022

$$SQR = 1.76 + 6.02 * q + 20 * \log\left(\frac{signal}{(MAX\ SIGNAL)}\right) dB$$ **Equation 5**

where q = number of bits in the DAC signal is the current signal output level and Max Signal is the full scale output of the DAC.

Over sampling
Oversampling is the use of higher than required sample rates. Minimum rate is defined by Nyquist at 2x the required output frequency, any rate above that is oversampling. Can be used to reduce the effects of quantization noise. Oversampling reduces the effects of quantization noise because quantization energy is spread equally across the bandwidth of the output DAC. By only using a proportion of the output bandwidth, the quantization noise in the required band is reduced.

$$SQR = 1.76 + 6.02 * q + 20 * \log\left(\frac{signal}{(MAX\ SIGNAL)}\right) + 10 * \log\left(\frac{F_{(SAMPLE)}}{F_{(NYQUIST)}}\right) dB$$

Equation 6 Reduction in Quantization Noise due to Oversampling
where q = number of bits in the DAC signal is the current signal output level and Max Signal is the full scale output of the DAC.

Accumulator Truncation
Errors are added to the output signal due to the truncation of the accumulator before the look up table. Causes spurs in the frequency domain at multiples of the output frequency. Spurs vary from 0 to maximum truncation error shown below. Assume accumulator is n bits. Look up table address is p bitsIf n – p is >=4 then

$$Max\ truncation\ error \approx -6.02 * p\ dB$$

III. WAVE FORM GENERATION PROCEDURE

Square wave
The amplitude of the wave will be high for n/2 samples. Here we took high value as '127' corresponding to all ones. After n/2 cycles, the amplitude will be made low ('0').

Saw tooth wave
The amplitude should be increased to maximum for n samples and then reset to low. Maximum value is '127' and the low value is '0'. The rate of increase is inversely proportional to n value.

Triangular wave
The amplitude has to incremented to maximum till n/2 cycles (from all zeros to all ones). After n/2 cycles, the value has to be gradually decremented to all zeros again.

Sine wave
The generation of sine wave is the most critical part in creating a DDS signal generator. There are many methods available to generate sine wave. They are

1. Taylor series Approximation
2. Table lookup.

There are other methods available which are too complicated to implement in FPGA.

1. **Taylor series Approximation**

The Taylor series of the sine function is: sin(x) = x - (x^3)/3! + (x^5)/5! - (x^7)/7! + ...
Where x is in radians. Since x value is in radians, x should be a floating value. Since we approximate the value, if we take more terms of this series, we get less error. For low values of x, we need only few terms. For higher values of x we need more terms. This kind of approach can be realized by hardware or FPGA. A serious disadvantage of this method is that it requires large no of multiplications to be done. So the hardware implementation is very large. So the cost of implementation is also high.

2. **Look up table method.**

In this method, we first generate the all the sine values for n samples and store it in a memory. In this method, we use the symmetric properties of sine wave. So we need to save only the values of sine wave from 0 to pi/2. This method yields very less error compared to other methods and the hardware realization is also simple.

Memory requirement:

So we chose this method to implement in the hardware. We chose n=512. Therefore we need to store 512/4= 128 sample sine wave values. Because 2*pi/4= pi/2. So we need to store values from 0 to pi/2 only. Other values are generated by symmetric properties of sine wave. For intermediate values we can interpolate between two samples. The no of bits used to store the sample values are 8 bits. So memory needed 8x128 bits= 1024 bits = 128 bytes.
Here we used separate blocks for generation of each wave and a signal selector is used to select the required output waveform using select lines.

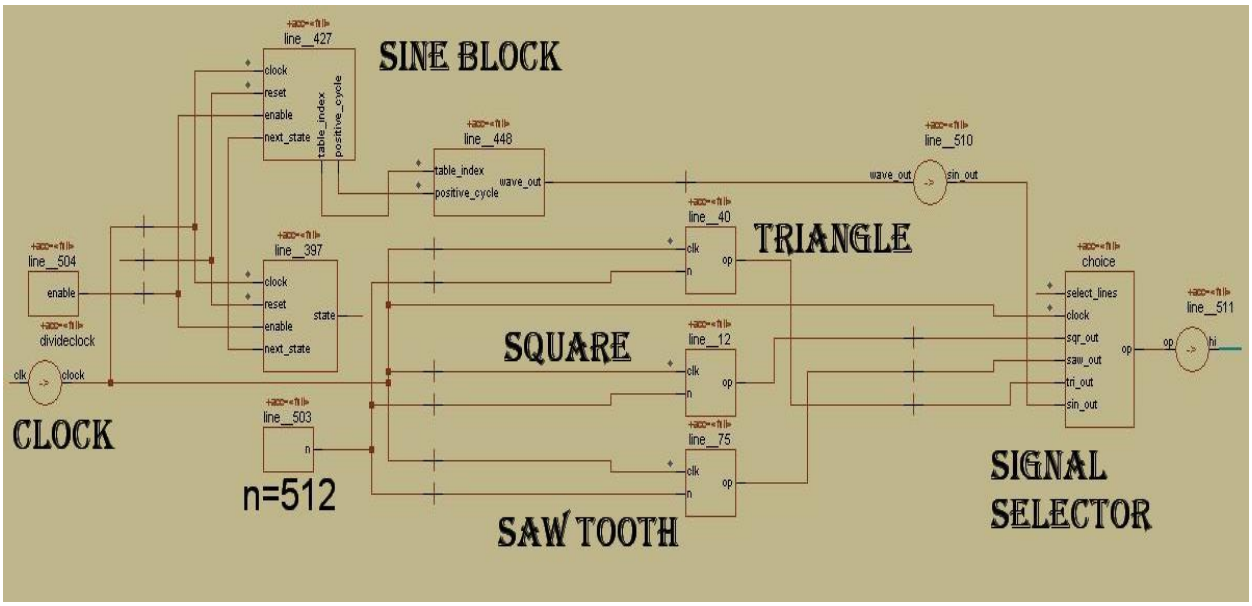

Figure 4: Block Design of Each wave form

IV. RESULT & DISCUSSION

The DDS block coding is done in VHDL, simulationand synthesis is done using Xilinx ISE. The individual waveformsare shown in Fig.5,Fig.6,Fig.7,Fig.8,Fig.9,Fig.10,Fig.11,and Fig.12, Overall output inFig.13 and arbitrary waveform in Fig.14. The code is downloaded into theFPGA. Amplitude values for sine and cosine signals are stored in rom. It generates the requiredfrequency from a reference clock.

ISSN 2348 – 8034
Impact Factor- 4.022

Result Square wave:

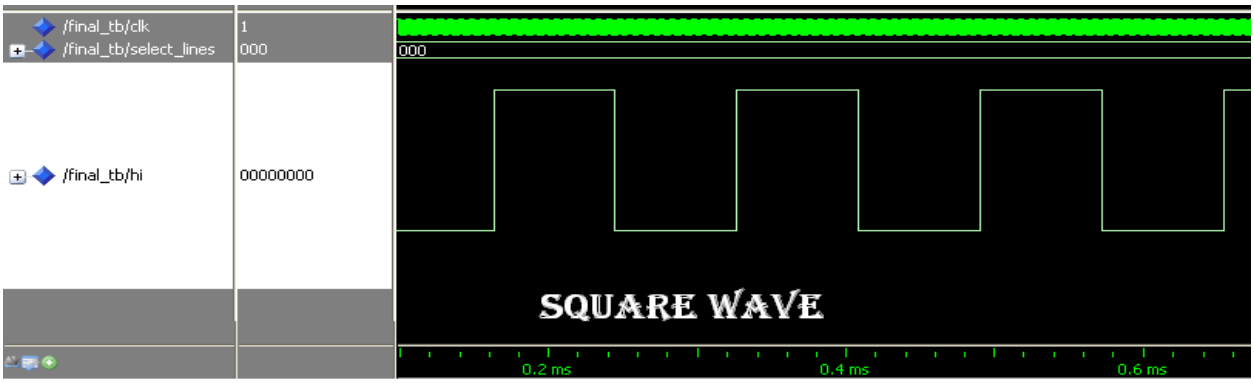

Fig.5

Saw tooth wave:

Fig.6

Triangular wave

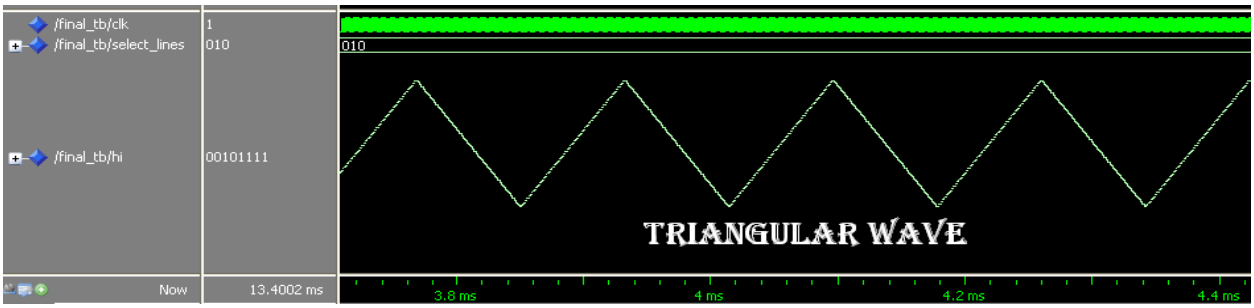

Fig.7

[Potharaju, 4(11): November 2017]
DOI- 10.5281/zenodo.1067984
ISSN 2348 – 8034
Impact Factor- 4.022

Sinewave:

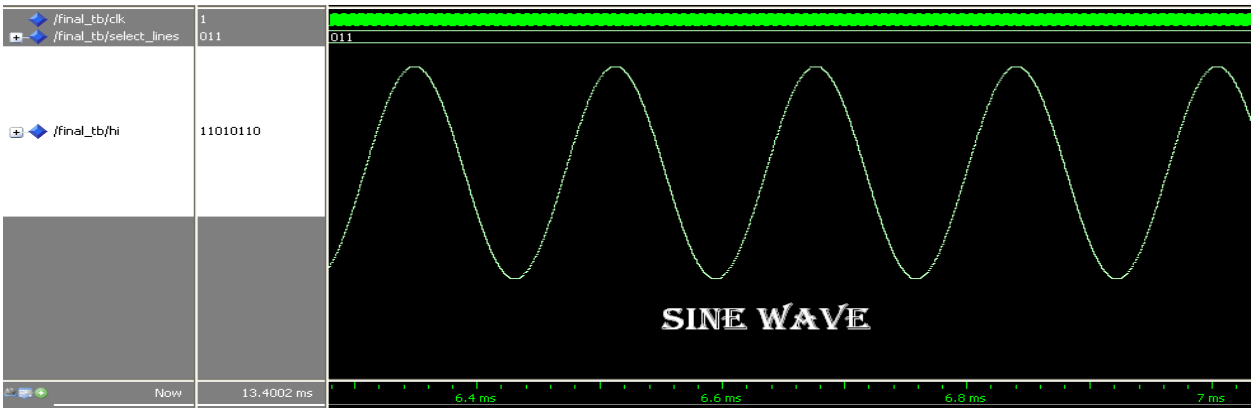

Fig.8

Mixing of two signals(Square wave + Saw tooth wave):

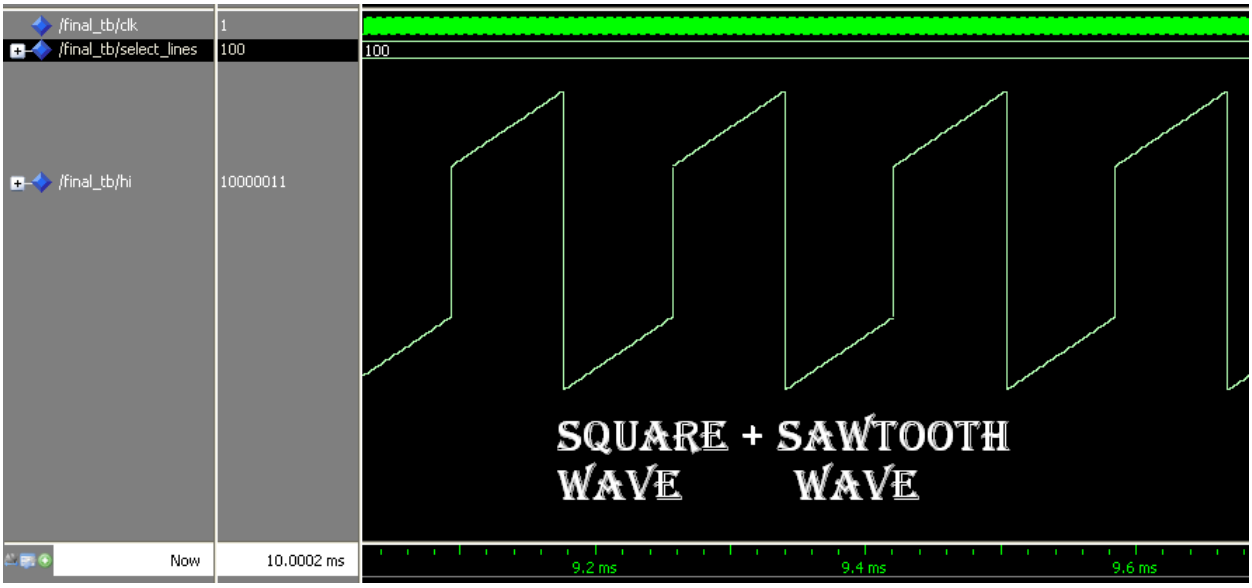

Fig.9

ISSN 2348 – 8034
Impact Factor- 4.022

Square wave + sine wave:

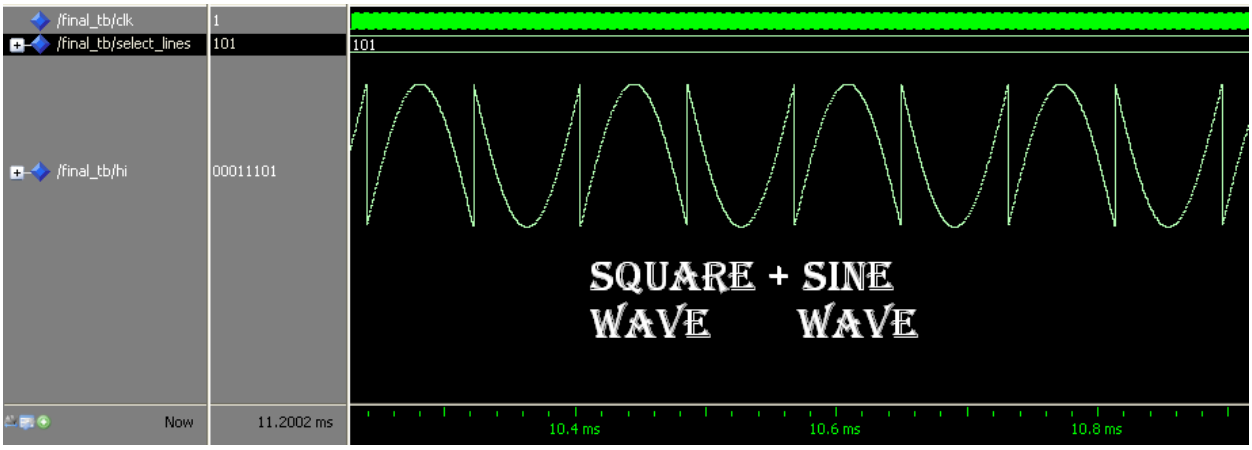

Fig.10

Square wave + triangular wave:

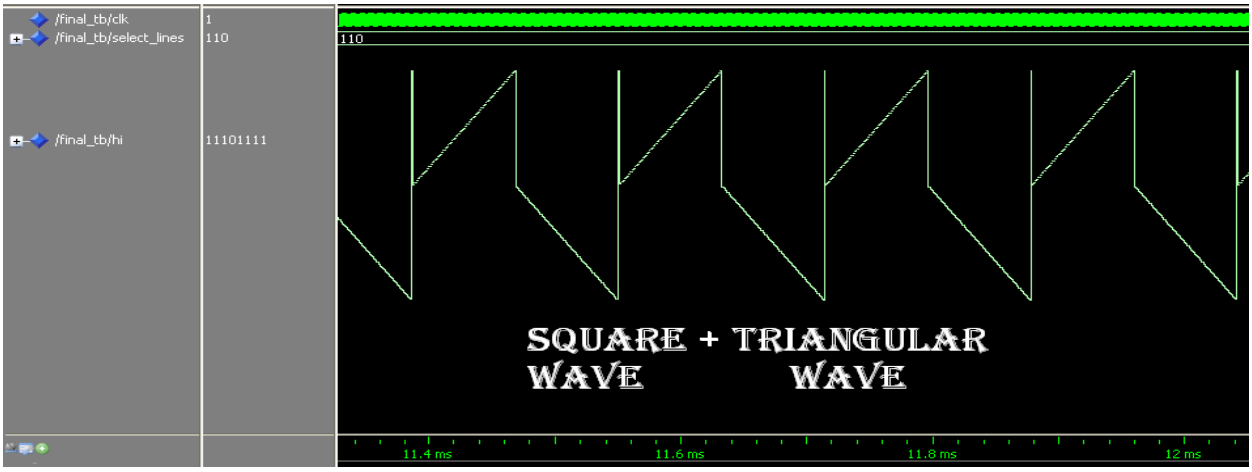

Fig.11

[Potharaju, 4(11): November 2017] ISSN 2348 – 8034
DOI- 10.5281/zenodo.1067984 Impact Factor- 4.022

DC signal:

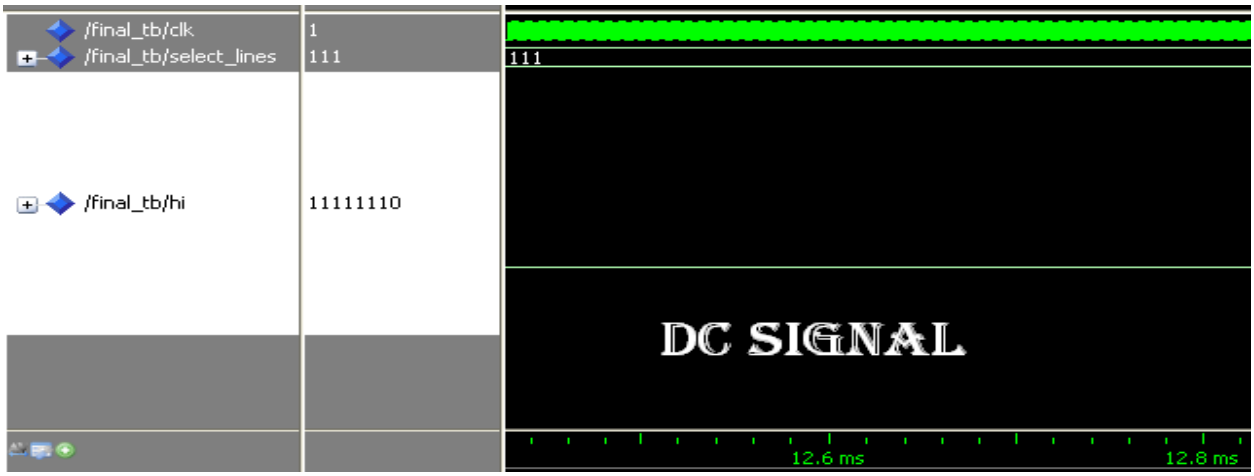

Fig.12

Overall output:

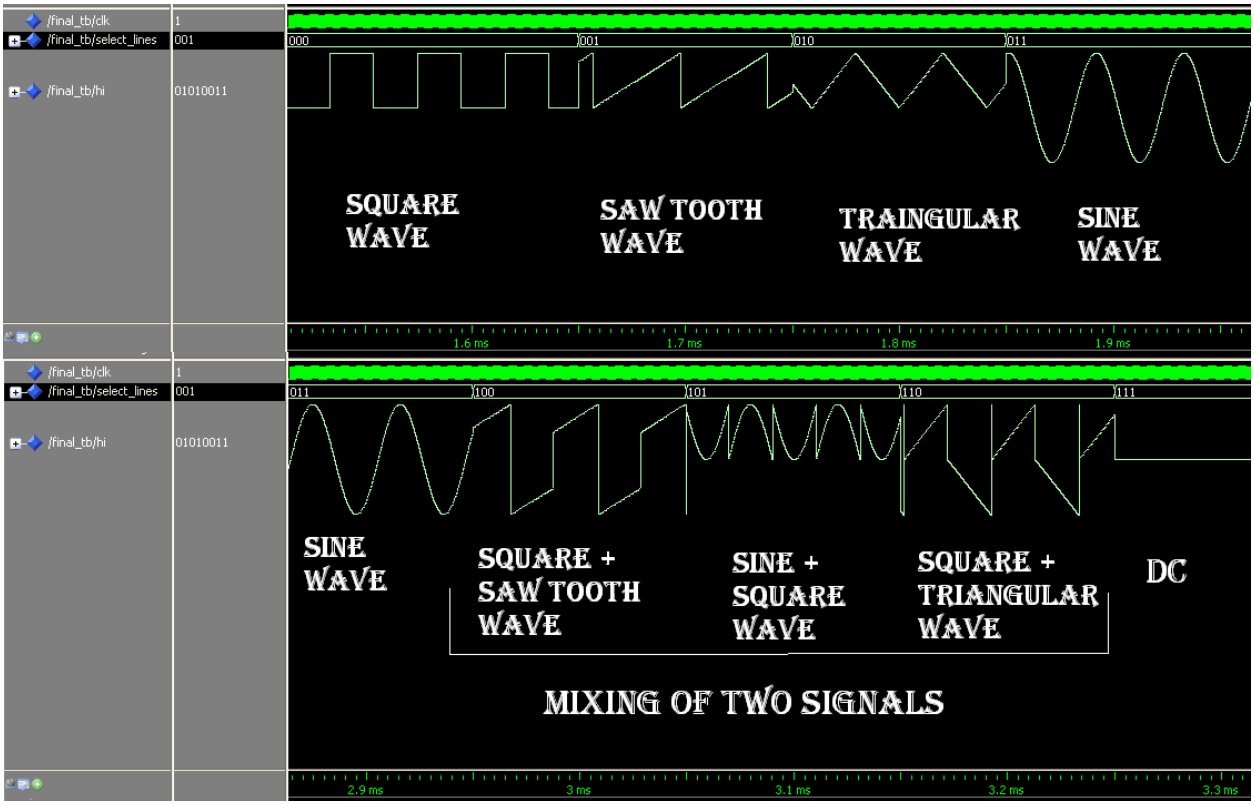

Fig.13

Merits of DDS signal generator:

The tuning resolution can be made arbitrarily small to satisfy almost any design specification. The phase and the frequency of the waveform can be controlled in one sample period. The DDS implementation is always stable, even

with finite-length control words. There is no need for an automatic gain control. The phase continuity is preserved whenever the frequency is changed (a valuable tool for tunable waveform generators). But in conventional signal generators, the phase of the waveform will change if the waveform is tuned or changed.DDS consumes very less power compared to conventional signal generator. Because the FPGA and DAC consumes only very less power compared to analog signal components in the analog signal generator.

Inferences:

1. We first implemented sine wave generation using Taylor series method. But this method does not give exact values of sine signal for higher phase values. This method also needed more no of multiplications. So we adopted another method called look up table method. Though it needed a little memory, it is an effective method to generate sine wave.
2. Not only the standard signals such as sine wave, square wave can be generated but also any kind of periodic wave form can be achieved using DDS. Even we can create any arbitrary waveform we like. We just need to generate the waveform in the computer and feed it into FGPA lookup table values. This kind of arbitrary waveform generation is impossible in conventional function generators.
3. We can implement any kind of modulation and mixing of signals using DDS signal generator. When we use a analog circuit or other components to perform modulation or mixing of two or more signals, the device noise will be added to the output waveform. But in DDS there is no chance of noise because all the modulation and mixing are done digitally and the output waveform is noise-free.

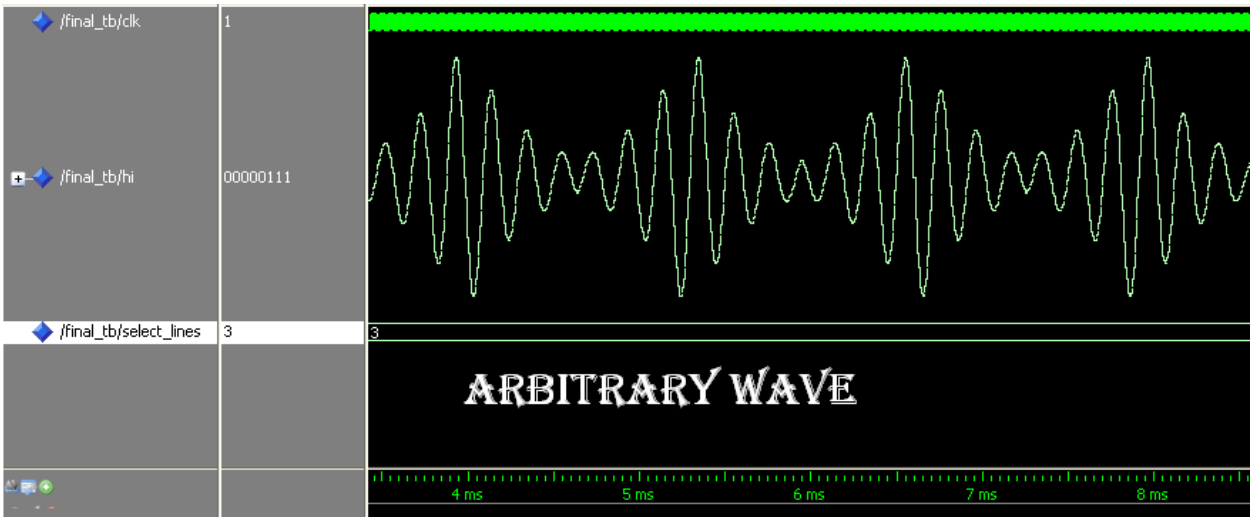

Fig.14

4. Any kind of signal processing techniques can be implemented in FPGA. Before sending the digital values to DAC, we can perform digital signal processing which gives more flexibility and power to the FPGA based design of signal generator.
5. The DAC chip should be operated at the range of MHz because the output of the FPGA is in Mega Samples per second. The advised D/A convertor chips are AD9761, AD9783, AD9780, AD5433, AD5440, AD768 etc.
6. The amplitude of the output waveform is controlled by an amplifier which can be tuned using a potentiometer. It is very easy to implement this type of signal generator.
7. Frequency control: the frequency of the signal can be controlled by three ways.
   a) By varying the clock of the FPGA externally by hardware.
   b) By varying the 'n' value the length of the waveform varies and hence the frequency of the signal changes.
   c) By implementing a frequency divider in the vhdl code of the FPGA. This method is very helpful if you want to decrease the frequency in large scale. In this code, we implemented this kind of approach to vary the frequency.

When we implement both methods b) and c) we can achieve any signal with any frequency. This is the major advantage of DDS FPGA signal generator.

V. CONCLUSION

Thus, Direct Digital synthesis is used to generate any waveform in FPGA. DDS is a valuable technique that can be applied to generate any waveform at any frequency. When implemented using DDS along with Signal processing techniques, we can modulate, encode, encrypt, any signal digitally and generate it. Moreover, it can also be interfaced with computer, which enables the user to view or access the waveforms generated by the FPGA. This FPGA design can be implemented to ASICs and the power consumed will be reduced further. Thus DDS turns to be effective technique based on signal quality, phase noise, very low device noise, low power consumption, size, etc.

VI. ACKNOWLEDGEMENTS

I sincerely thank FIST (Fund for Improvement of Science and Technology Infrastructure) Sponsored by Department of Science and Technology (DST)at Vignana Bharathi Institute of Technology, Hyderabad.Underwhich the present work was possible.

REFERENCES

1. *Fang Yi-yuan, Chen Xue-jun, "Design and simulation of DDS based on Quartus II" .Copmuter Science and Automation Engineering (CSAE), 2011 IEEE international conference on June 2011: 357-360.*
2. *Manoj Kollam, S.A.S.KrishnaChaithanya, Nagarajukommu, "Design and Implementation of an Enhanced DDS Based Digital Modulator for Multiple Modulation Schemes" International Journal of Smart Sensors and Ad HocNetworks (IJSSAN), Volume-1, Issue-1, 2011.*
3. *Wenmiao Song, Qiongqiong Yao. "Design and Implementation of QPSK Modem Based on FPGA" Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE international conference on July 2010: 599-601.*
4. *Zhong Wei-jie, JIANG Lei, LIU Yao-ying. "Design of DDS Based on VHDL Editing Routine", Shipboard ElectronicCountermeasure .2007, 30(2):102-105.*
5. *Pao-Lung Chen, Ching-Che Chung, and Chen- Yi Lee,A Portable Digitally Controlled Oscillator Using Novel Varactors*
6. *Duo Sheng, StudentMember, IEEE, Ching-CheChung, Member, IEEE, and Chen-Yi Lee, Member, IEEE An Ultra-Low-Power and Portable Digitally Controlled Oscillator for SoC Applications*
7. *Parameshwara M. andSrinivasaiah H. A Novel ROM based DDFS Architecture for Portable and wide band Communication.*
8. *Lionel Cordesses, DDS- A tool for periodic wave generation.*
9. *Volnei A. Pedroni, Circuit Design with VHDL.*
10. *Bai Juxian. Direct Digital Synthesis. Xi'an: Xi'an JiaotongUniversity Press, 2007..*