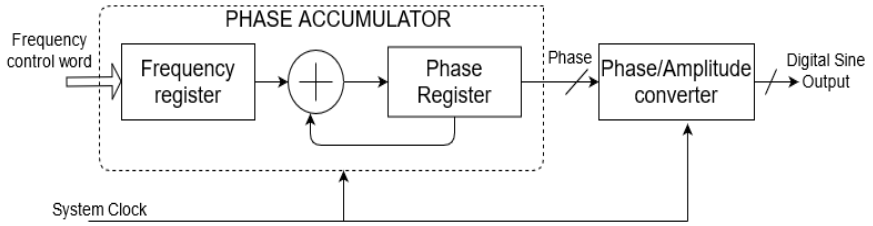
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
CHENNAI - 600036

Implementation of LUT-Based Direct Digital Frequency Synthesizer in FPGA



A Project Report

Submitted by

SRIKANTH GARLAPATI

In the partial fulfilment of requirements

For the award of the degree

Of

MASTER OF TECHNOLOGY

July 2021

DEDICATION

To my family and friends for their love and support

CERTIFICATE

This is to undertake that the Project report titled **IMPLEMENTATION OF LUT-BASED DIRECT DIGITAL FREQUENCY SYNTHESIZER IN FPGA** , submitted by me to the Indian Institute of Technology Madras, for the award of M.Tech, is a bona fide record of the research work done by me under the supervision of Dr. Janakiraman Viraraghavan. The contents of this Project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Chennai 600 036

Date: 5th July 2021

Srikanth Garlapati
Graduate Student

Dr. Janakiraman Viraraghavan
Project Guide

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere thanks and deepest regards to my project guide Dr. Janakiraman Viraraghavan for their attention, guidance, and insight during my project report. I would also like to extend my thanks my faculty advisor Dr. Nitin Chandrachoodan for providing encouragement and support at all times. I acknowledge the Department of Electrical Sciences of the Indian Institute of Madras for providing me with financial support during my graduate studies. Finally, I am forever indebted to my family and friends for being the source of energy, understanding, and care during these times.

ABSTRACT

KEYWORDS: DDFS; High-speed synthesizer; FPGA;LUT;BIST.

A method for the design as well as implementation of FPGA-based Direct digital Frequency synthesizer (DDFS) for Digital sine wave generation is firstly presented and analysed in this paper. A Digital part DDFS is designed based on optimized ROM LUT using the symmetry of the sine wave one fourth period compared with sine waveform. ROM LUT with only one fourth of the values of sin waveform is needed to develop the whole sin wave. The simulations for the designed model is carried out with the help of VIVADO 2019.1 software in field programmable gate array (FPGA) xc7z020clg484-1 with Verilog HDL. The Implemented model is synthesizable in FPGA. At last, an error analysis is made to the result of simulation using MATLAB.

LIST OF FIGURES

Figure	Title	Page
1.1	Basic DDFS Architecture	1
2.1	Phase Accumulator[3]	5
2.2	Digital Phase Wheel[1]	7
2.3	Phase/Amplitude Conversion table [4]	8
3.1	RTL schematic	11
3.2	Hardware Utilization	11
3.3	On-chip Power consumption	11
4.1	Sine output waveform for 1MHz frequency	13
4.2	Sine output waveform for 5MHz frequency	14
4.3	Max error vs number of output bits	15
4.4	RMS error vs number of output bits	16
4.5	Different Error vs Number of Output bits	17

ABBREVIATIONS

ADC	Analog to Digital Converter
IP	Intellectual property
BIST	Built-in Self Test
LUT	Look-up Table
ROM	Read Only Memory
DAC	Digital to Analog Converter
NCO	Numerically controlled Oscillator
PA	Phase Accumulator
FCW	Frequency Control Word
MSB	Most Significant Bit
FPGA	Field Programmable Gate Array
RTL	Register Transfer Level
CORDIC	Co-ordinate Rotation Digital Computer

CHAPTER 1

INTRODUCTION

1.1 Background

Direct Digital Frequency Synthesis(DDFS) is a digital technique for generating any analog waveform i.e, sinusoidal,ramp etc..It has been widely used in communication systems.Unlike conventional analog oscillator structures,DDFS can be applied where fast frequency switching,frequency resolution, and a coherent phase relationship among sinusoidal waves are required.The most traditional DDFS system follow the general architecture as shown in Fig 1.1. There are many ways to implement the sine generator.There is the polynomial approximations-based DDFS that requires a ROM to store the coefficients of the polynomials and the polynomial evaluation hardware with multipliers. Taylor's expansion method is too complex to use, and the computing speed is constrained in hardware implementation to some extent and also there is Co-ordinate Rotation Digital Computer(CORDIC) which has several modes such as circular,linear, hyperbolic.Other Simplest way is to use ROM lookup table (LUT) where a large ROM is needed.There are several Approches made in order to reduce LUT memory usage. When higher bit precision is required, the benefits of CORDIC architectures can be apparent.In comparison to CORDIC architecture, increased bit accuracy requires a larger ROM, which consumes more power and is slower.

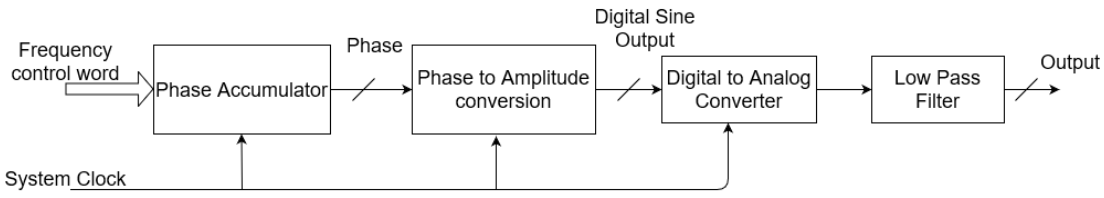


Fig. 1.1: Basic DDFS Architecture

1.2 Motivation

BIST is the most preferred technique with ADC applications to perform parallel processing with limited resource requirements.It produces a high precision mixed signal test response by offering a solution for signal integrity through on-chip stimulus generation and output response analysis.

The purpose of the project is to generate a sampled sinusoidal signal with minimum error so as to use it for testing and debugging an ADC.This process is achieved by using LUT based sine generating method. LUT based sine genration has some Advantages over other Architectures. The ROM LUT Implementation is easy beacuase of the simplicity of ROM ciruit.

1.3 Contents

In this report, a DDFS architecture based on the ROM Look-up table method is presented.In chapter four the results are presented and analyzed and the conclusion is the last part of the report.

The material presented in this report is organised as follows.In chapter two DDFS, Phase Accumulator, ROM/Look-up table and Phase/Amplitude conversion are reviewed briefly. In chapter three the Implementation of proposed DDFS architecture for LUT based method is presented. Subsequently we shall also take a perfunctory glance into the design flow and the top level design for this algorithm to get the fell of how it works after the sine wave is generated.The final part involves the simulation results the error analysis of the output.

CHAPTER 2

Direct Digital Frequency Synthesis

2.1 Brief History of DDFS

J.Tierney first suggested the DDFS concept in 1971 [2]. Direct Digital Synthesis (DDS) is an electronic method of synthesising arbitrary waveforms and frequencies digitally from a single, fixed source frequency.Since the 1980s, direct digital frequency synthesizers have been widely employed in wireless transceivers due to the advancement of VLSI technology and the requirements of modern communication systems.Direct Digital Frequency Synthesis (DDFS), also known as Numerically Controlled Oscillator (NCO), is a technique for generating signal waveforms that are repeating in nature using digital data and mixed/analog signal processing blocks.Over a large frequency range, a DDFS can perform quick frequency switching in small frequency steps. It also offers linear phase and frequency shifting with excellent spectral purity.A DDFS is particularly useful for producing an accurate, high-frequency, and phase-tunable output.An accumulator, a ROM/lookup table, a DAC, and certain reconstruction filters make up a standard DDFS architecture.A direct digital synthesizer (DDS) has the advantage of being able to modify its output frequency, phase, and amplitude accurately and quickly using a digital processor[6].Other DDFS features include the capacity to tune with extremely fine frequency and phase precision, as well as the ability to quickly "hop" between frequencies.The use of digital functional blocks allows for a high level of system integration to be achieved in DDFS. Cable modems, measuring equipment, arbitrary waveform generators, cellular base stations, and wireless local loop base stations are among the applications addressed by the DDFS.

2.2 Overview of DDFS

Direct Digital Frequency Synthesis(DDFS) is a digital technique for generating any analog waveform i.e, sinusoidal,ramp etc..It has been widely used in communication systems.Unlike conventional analog oscillator structures,DDFS can be applied where fast frequency switching,frequency resolution, and a coherent phase relationship among sinusoidal waves are required.The most traditional DDFS system follow the general architecture as shown in Fig 2.1.It depicts the architecture which consists of a phase accumulator(block of ADDER and register),a sine generator(Phase/Amplitude conversion unit),a digital to analog converter(DAC), and a low-pass filter. The section of the the phase accumulator and the sine/cosine generator is called a numerically controlled oscillator(NCO).The function of the NCO is to transform a linear phase word into a digital sine/cosine word.As the two inputs reference clock(F_{clk})and frequency control word(FCW) are used the phase accumulator integrates FCW to produce an angle in the interval of [0,2π), and the sine/cosine generator computes the sinusoidal values digitally and followed by a DAC and low-pass filter gives an analog outputs. Coordinate Rotation Digital Computer(CORDIC) is a simple and hardware efficient algorithm that evaluates various elementary functions,including sine and cosine of functions.As the implementation only requires simple adders and shifters,CORDIC has been used in high speed application[7]

2.3 Building Blocks of DDFS

DDFS has a few components like a Phase Accumulator, Phase to Amplitude converter(a sine look up table), DAC and a Filter. This section discusses about Phase Accumulator and Phase/Amplitude converter Blocks in detail which is mostly involved a part DDFS Architecture for Digital sine wave generation.

2.3.1 Phase Accumulator

We use DDFS for producing a sine wave of required frequency. The frequency of produced sine wave often depends on two factors, the sampling clock frequency and the fre-

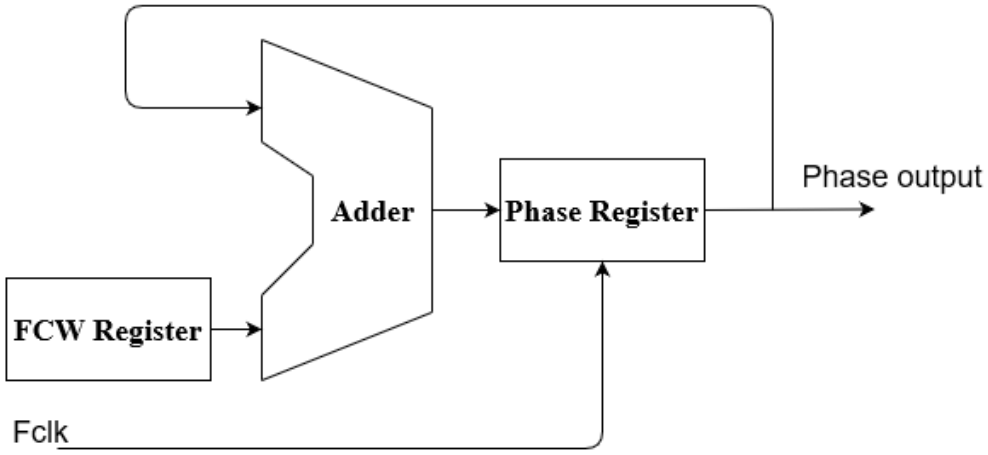


Fig. 2.1: Phase Accumulator[3]

quency control word which could be a binary number that is programmed into a phase register.Phase Accumulator acts as a means to Phase to Amplitude conversion.Phase Accumulators main role is to generate a phase (angle) address which is used to pick up corresponding sine amplitude from the look up table. Let us understand phase accumulator role in generating sine samples. If we use a look-up table which is preloaded with sine values in it. The Phase Accumulator computes corresponding Phase address for the look up table which gives corresponding digital output to that of phase angle. For generating a particular frequency of sine wave, we give a constanst phase increment value (a constant binary number) which is stored in a phase register to the Phase accumulator and therefore, The phase accumulator value will be incremented with each clock cycle as shown in fig 2.1[3]. The phase accumulator is actually a modulo-M counter that increments its stored number each time it gets a clock edge.Phase increment will effectively sets how many points to skip around the phase wheel[1].The larger the jump size, the faster the PA overflows and completes its corresponding sine-wave cycle.The resolution of the phase accumulator (N), which controls the tuning resolution of the DDS, determines the number of discrete phase points included in the wheel.

For Example, if phase increment value(frequency control word) is 000...01 and Phase accumulator width is 26 bits then PA will increment it's value by 1 for every clock cycle and it will overflow after 2^{26} clock cycles. The PA will overflow after only

two reference-clock cycles if the FCW value is changed to 0111...1111. The Number of Points on Digital wheel of PA for different PA width is shown in Table 3.1. For Example, If the N is 4 and FCW value is 2,The number of samples on digital phase wheel would be 16 and Jump size in Phase wheel with each clock edge would be same as illustrated in fig. 3.1. The address to a sine lookup table/ROM/phase-to-amplitude converter comes from the phase accumulator's output. Generally,Each address in the LUT corresponds to a sine wave phase point ranging from 0 to 360 degrees. But Here, Only 1st Quadrant values are stored in LUT. If the Phase Increment is large, then the PA will go quickly through the sine look-up table and Therefore, generates a higher frequency sine wave. Similarly, If the phase increment is lower then the PA takes more steps to cover the sine look-up table. Therefore, resulting in a lower frequency sine wave. If F_{clk} is the clock frequency, then the frequency of the output sine wave is equal to:

$$F_{out} = \frac{F_{clk}}{2^N}.FCW \tag{2.1}$$

Above equation is known as the DDFS "tuning equation."

Where, FCW = Frequency control word

F_{clk} = reference clock frequency (system clock)

N=length of the phase accumulator, in bits

Table 2.1: Number of Points on Phase wheel

Number of PA bits	Number of Points
8	256
12	4096
16	655535
20	1048576
24	16777216
26	67108864
28	268435456
32	4294967296

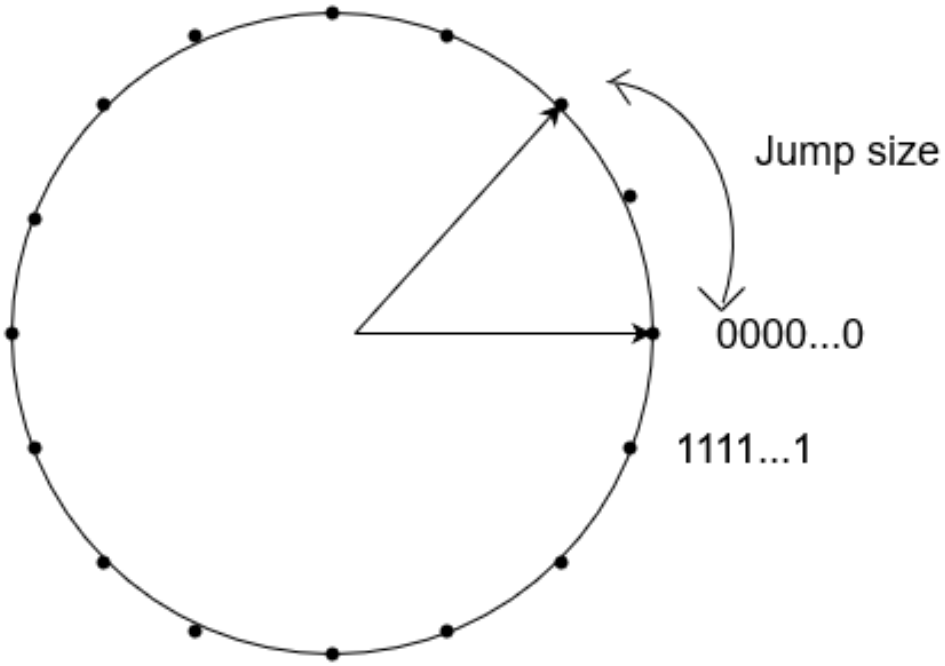


Fig. 2.2: Digital Phase Wheel[1]

2.3.2 Phase/Amplitude Conversion

Phase/Amplitude converter is another key component in generation of digital sine wave. In this Model, A ROM Memory is used for storing sine Look-up Table. This will generate Sine output amplitude from PA's Digital Phase output.The accumulator output represents the phase of the wave as well as an address to a word, which is the corresponding amplitude of the phase in the LUT. In this model, the lookup memory contains quarter cycle of the sine values.The size of the LUT is 2^n words.LUT converts digitally truncated phase information into quantized numerical waveform samples. LUT used in this method consists of one fourth period of sine values.

Let,The PA is 26 bit Accurate Register. The output of PA is a saw tooth wave. To directly convert 26 bits of phase to corresponding Amplitude would be need a 2^{26} in a LUT. If each sample is stored with 16 bit accuracy, then the required ROM memory to store LUT sine samples would be 0.125 giga-bytes. It is very hard to develop such a design.Therefore, the solution could be using a fraction of the most significant bits of PA output as the phase information. For example, Consider a PA of 26 bit wide LUT of 10 bit address. The Upper most two digits of PA are used for identifying quadrant of

Two Upper bits in Accumulator	Phase	Sine Wave Conversion
00	[0,π/2]	No Phase Conversion, No Amplitude Conversion
01	[π/2,π]	Phase Conversion, No Amplitude Conversion
10	[π,3π/2]	No Phase Conversion, Amplitude Conversion
11	[3π/2,2π]	Phase Conversion Amplitude Conversion

Fig. 2.3: Phase/Amplitude Conversion table [4]

sine wave to be generated.The other last 10 bits are used as Phase information to locate corresponding sine value from LUT based on Quadrant of operation.For Example, To generate a sine wave of 1MHz output frequency with 100MHz reference frequency 100 clock cycles are required for one complete cycle of output sine wave.For 1st quadrant sine values, the two last MSB's are 00 and other last 10 bits are used as phase address to the LUT with out any changes and the corresponding LUT output is taken as it is for the sine output. In 2nd Quadrant, After completion of 25 clock cycles The last two MSB's of PA are 01 and the other last 10 bits is taken as phase information and subtracted from 2^{10} which will be used as a address for the LUT and the corresponding values are taken as sine output. Where as in 3rd Quadrant, The last two MSB's are 10 and the Phase information is used as the Phase address input to the LUT.Here the sine output will be negative value of LUT output. Finally, For the fourth Quadrant the two MSB's are 11. Here, both Phase conversion and Amplitutde conversion is performed. The phase information is subtracted from 2^{10} and is used as Phase address input to the LUT. The Sine output is negative of LUT output.Therefore, The complete sine wave is generated with use of Phase or/and Amplitude conversion as shown in fig 2.3 for one cycle of operation, The PA will overflow and again it increments from initial value.

•

CHAPTER 3

Implementation

This section discusses the implementation of Digital part of DDFS that is up to producing digital sine output. Phase Accumulator part and Phase/Amplitude conversion part to get digital sine output is discussed in this chapter. the designed model contains a Phase register(PR), Phase Accumulator(PA) and a Look-up table and Phase/Amplitude conversion unit.

3.0.1 Hardware Implementation

This method has been modelled using Verilog and simulations and RTL analysis are carried out using VIVADO 2019.1. This Verilog is synthesizable in FPGA. The FPGA used in this Implementation is xc7z020clg484-1 which is available in VIVADO tool. A verilog module is written such that it takes clock,reset and 24 bit data "fcw" as input and 16 bit output "sine_out" and also a Phase Accumulator of 26 bit register "accu" is used in this module. The fcw which is a Phase increment to PA is calculated using frequency tuning equation(2.1) for generating 1MHz output sine wave with 100MHz reference clock as discussed earlier. fcw is 671088 which is converted to Hexadecimal format "A3D70" and given as input to the Phase register "fcw". A memory file is created and added to the Project. The Sine samples are generated from MATLAB using sin function and these are double precision floating point numbers converted to fixed point 16 bit accuracy samples. These 16 bit samples are stored in the created memory file. A ROM memory "rom_memory" is instantiated in the module with "ROM_WIDTH" and ROM_ADDRESS_BITS are given as Parameters inside the module. In this case ROM_WIDTH = 16 and ROM_ADDRESS_BITS = 10 that means a $16x2^{10}$ memory is created and the values from memory file is stored in this rom_memory with the use of $readmemh function in verilog.

Quadrant	rom_out
[0,$\pi/2$] (00)	rom_memory[lut_index]
[$\pi/2, \pi$] (01)	rom_memory[(2**ROM_ADDR_BITS)-1)-lut_index]
[$\pi, 3\pi/2$] (10)	-rom_memory[lut_index]
[$3\pi/2$,2π] (11)	-rom_memory[(2**ROM_ADDR_BITS)-1)-lut_index]

Table 3.1: Phase/Amplitude conversion for sine generation

For every positive clock edge, The module checks if reset is high then the accu is assigned zero else, fcw is added with accu. A two bit register "count" is also initialized in the module such that it assigned with last two Most significant bits of accu for every clock edge. The count is used for checking case statement to recognise quadrant of sine wave to be generated. Based on case statements the Phase/Amplitude conversion is made with LUT output and lut_index. Here, lut_index is a 10 bit register which is assigned from other last 10 bits of accu i.e, accu[23:14].The phase/amplitude conversion verilog case statements are shown in Table 3.1 for each quadrant. In Case 01,11 condition, for the quadrant two and four The phase conversion is made on this lut_index and resultant is used as a address for rom_memory . lut_index is directly used as a phase address to rom_memory for the other two quadrants. Amplitude conversion is made to LUT output to get corresponding sine wave output. This also done based on Quadrant of operation. For Quadrant one and three, The rom_memory output is directly taken out as sine_output. whereas, for other two quadrants, The output sine values is negative of rom_memory output. These statements are executed based on count value. It is 1st Quadrant for 00 and second quadrant for 01 and Third quadrant for 10 and fourth quadrant for 11. Finally,The rom_out is assigned to sine_out which is a 16 bit 1MHz sine wave output observed in VIVADO simulation.

The fig 3.1 shows RTL schematic of the implemented model.It shows a RTL memory block of 10 address lines and 16 bit wide and a 26 bit RTL Adder for PA, RTL Subtractors for Phase/Amplitude conversion and several RTL MUXs in the design. The designed model is synthesizable and it uses 24 LUT, 26 FF and 68 I/O Ports and

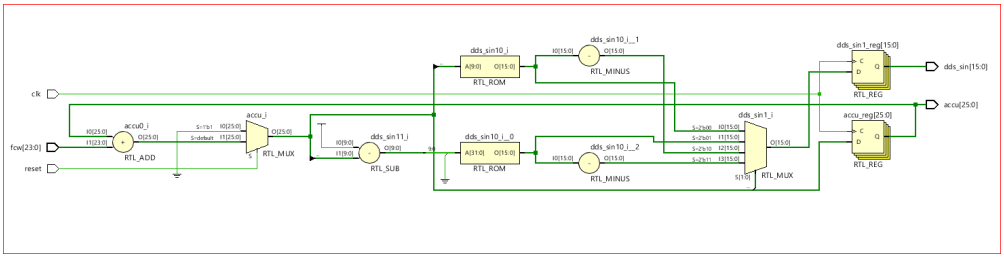


Fig. 3.1: RTL schematic

also a BUFG that are available in the xc7z020clg484-1 FPGA after VIVADO sysnthesis.The Utilization Reports are shown in Fig. 3.2 and total on-chip power consumption is 0.163W is shown in fig 3.3

Resource	Utilization	Available	Utilization %
LUT	24	53200	0.05
FF	26	106400	0.02
IO	68	200	34.00
BUFG	1	32	3.13

Fig. 3.2: Hardware Utilization

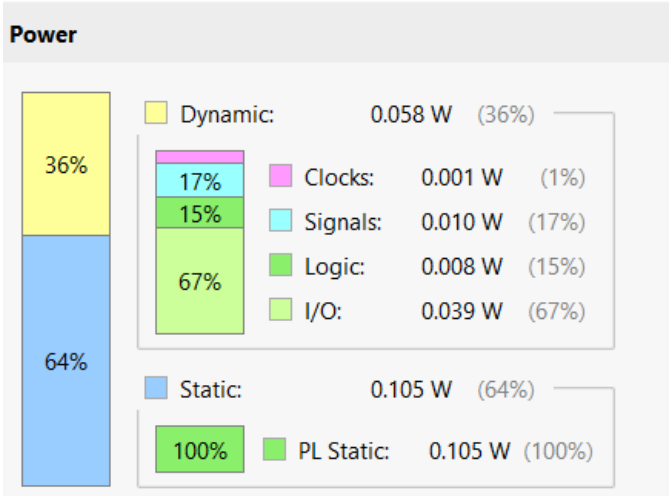


Fig. 3.3: On-chip Power consumption

CHAPTER 4

Results and Analysis

This section discusses about sine output and Its Error analysis. Finally, will come to a optimal conclusion in usage of number of output bits and LUT size in the design for bare minimum error estimate of output sine wave.

4.1 Simulation Results

The simulation for the designed model is carried out with help of VIVAVO 2019.1 softwave with xc7z020clg484-1 FPGA in it and the sine output waveform and Phase accumulator output is observed and analysed as discussed below. Here, 2^{10} number of MATLAB generated sine values for the 1st Quadrant are stored in a memory file with 16 bit Accuracy.The number of Samples stored in LUT is 1024 with 10 bit address. a Phase Accumulator of 26 bit wide is chosen for the model and a Phase register of 24 bit is chosen to store the Frequency Control Word (FCW) a binary constant number which will will be used as a phase increment given to the PA. The PA is initialized with zero. For every clock edge, the PA will be incremented by FCW value.The output of PA will be served as a address of the LUT and corresponding sine value is read out of ROM memory.Actually, the last two Most significant bits of PA will be used for recognizing quadrant and other last 10 bits will be used for generating address to the LUT. remaining least significant bits in PA will be truncated. The overflow of PA depends upon the bit width of PA and FCW. The output frequency of digital sine wave is directly proportional to the FCW. Therefore, larger the frequency control word, Higher will be the output frequency and sooner the PA overflows. This is shown with the help of VIVADO simulations of the designed model.The carried out simulations helps in understanding the signal flow graph through the design and the overflow of PA and the relation between the output frequency and PA. The output frequency is determined by the PA's overflow rate and the frequency control word determines the overflow rate. To generate

an output frequency of 1MHz with a reference clock of 100MHz, a frequency control word 671088 is stored in a Phase register of 24 bits wide. The value of the Frequency control word is calculated using frequency tuning equation (2.1). This frequency control word is added to PA with every clock cycle. After performing Phase/Amplitude conversion the corresponding sine values are generated from LUT using PA bits. In the fig 4.1 it can be seen that for a frequency control word of 671088, The waveforms of PA and sine output are shown for the first 200 clock cycles. It is observed that sine wave takes 1 micro second for one complete cycle. Hence output frequency of generated sine waveform is calculated to be 1MHz which Justifies the Frequency Tuning equation 2.1. PA overflows slightly more than two times resulting in a sine wave output with lower frequency.

If the output frequency generated is increased to 5MHz, the new FCW would be

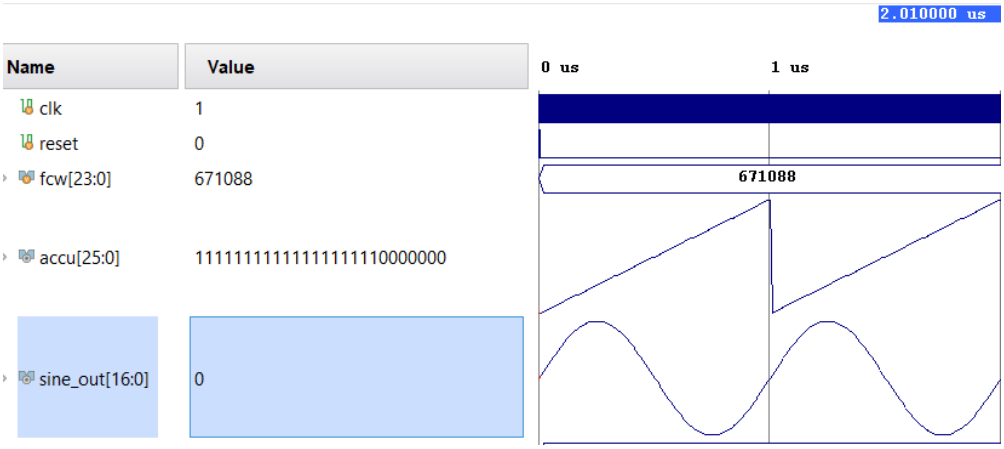


Fig. 4.1: Sine output waveform for 1MHz frequency

3355443 which is stored in a 24 bit Phase register.This FCW is added to previous value of PA on every clock edge. The figure 4.2 shows the output waveform of Accumulator and sine wave with frequency of 5MHz. For the first 200 clock cycles shown, PA overflows slightly more than 10 times resulting in a sine wave output with higher frequency. It can be inferred from fig 3.1 and fig 3.2 that For small FCW, PA overflows slower than larger FCW case. similarly, the output sine frequency is lower in fig. 3.1 than in fig 3.2

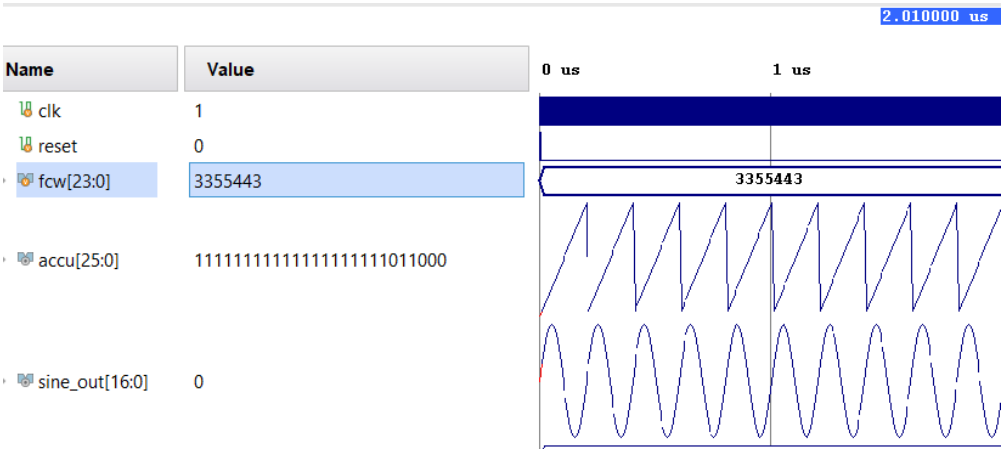


Fig. 4.2: Sine output waveform for 5MHz frequency

4.2 Error Analysis

LUT contains the samples of 1st Quadrant sine wave.These samples are in Quantised form (or) in binary form. These samples corresponds to different phase angles of sine wave. In this design, the Sine output using LUT Samples of 8,16 and 32 bit are analyzed and errors are estimated to recognise optimal number of output bits to be used in sine LUT. Errors are calculated for the output frequency of 1KHz with reference clock of 100MHz. The Parameter, number of samples stored in LUT also has significant affect on error of output sine wave. The sine wave output values are generated in VIVADO for 1KHz frequency by choosing corresponding FCW. This FCW is calculated using frequency tuning equation 2.1. The output samples generated are taken out and converted them in the range of (-1,1) double precision fixed point numbers. Original Double precision fixed point sine values are generated for particular frequency from MATLAB and compared with VIVADO generated sine values. Fig. 4.3 shows Variation of Maximum error with respect to number of Output bits drawn for different sizes of LUT such as 256,1024 and 8192. Similarly, Fig. 4.4 shows Variation of RMS error with respect to number of output bits drawn for different sizes of LUT such as 256,1024 and 8192. It is observed that LUT with 256 values produces more error compared to that of LUT with 1024 and 8192 samples. It is also observed that the LUT with 1024 and LUT with 8192

samples produces error in same range in the order of 10^{-4}. Therefore, LUT with 1024 sine wave samples are chosen for design in order to save the ROM memory required to store the values. If we take LUT with 8192 samples, then Memory required would be more.

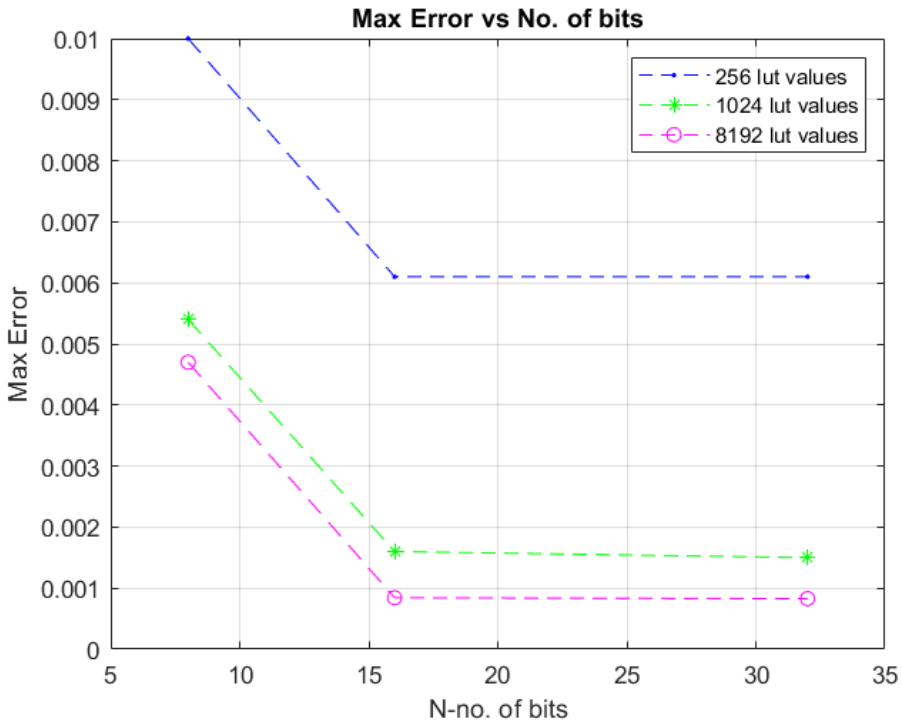


Fig. 4.3: Max error vs number of output bits

Now, 1KHz of output sine wave with 1024 LUT samples is considered for further analysis on varying number of output bits. Figure 4.5 shows the plot between Different type of Errors and number of output bits varied from 8 to 32 in the intervals of four. From the below figure 4.5, it is observed that Max Error, Mean Error and RMS Errors are Higher in case of 8 bits and 12 bits output. Where as the less difference pronounced in various errors for 16 to 32 bit output cases in the order of 10^{-4} which is preferably accepted. So, the optimal bit width of sine wave Samples to be stored in LUT is chosen to be 16 bit binary numbers. A the number of output bits increases the error decreases and also as the number of LUT samples increases the error produced is decreases. The optimal number of outputs bits chosen for the design is 16 bits with LUT 1024 samples has max error produced is 0.0016 and RMS error is 0.00063.

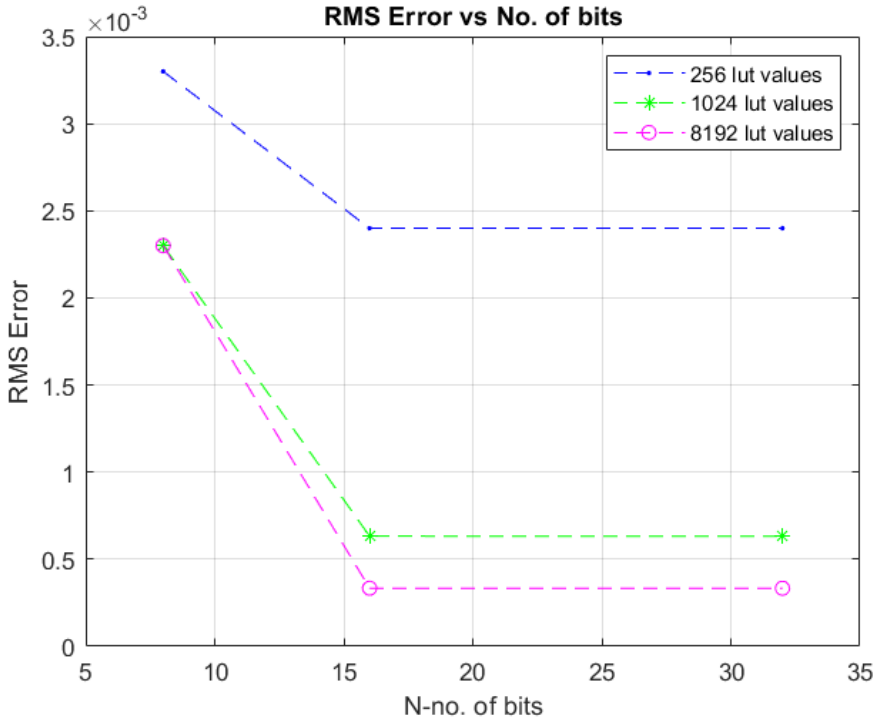


Fig. 4.4: RMS error vs number of output bits

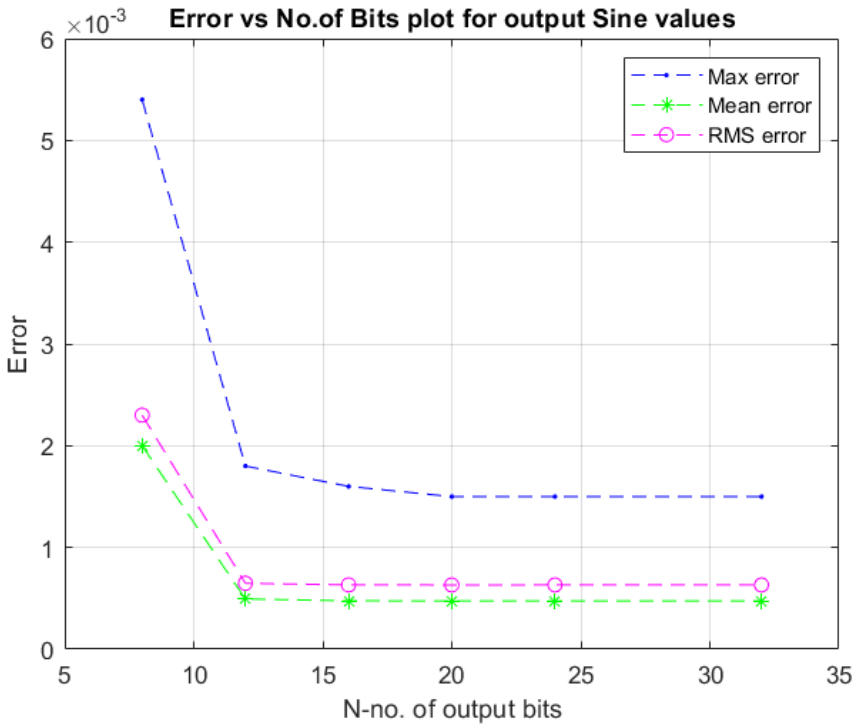


Fig. 4.5: Different Error vs Number of Output bits

CHAPTER 5

Conclusion

The Digital part of DDFS model have been implememnted and simulated using VIVADO 2019.1 simulator. The Implemented model illustrates the reduction in ROM/-LUT usage as the sine wave is generated by storing only First Quadrant sine values. The Error Analysis for different combination of number of LUT samples and number of output bits are performed and then LUT with 1024 samples and 16 bit accuracy is chosen for this model for min Error and optimal memory usage and has max error produced is 0.0016 and RMS error is 0.00063. As the size of LUT increases the error in the output decreases but the memory required increases. The Verilog module presented in this thesis is synthesizable on FPGA. Sine wave generation can also be implemented using ROMless architecture using quarter wave symmetry or some different algorithm, to reduce the hardware, power dissipation and increase the efficiency.

REFERENCES

1. (2004). Ask the application engineer—33: All about direct digital synthesis. URL https://www.analog.com/en/analog-dialogue/articles/all-about-direct-digital-synthesis.html.

2. **J. Tierney, J. R.** and **Gold** (1971). A digital frequency synthesizer. *IEEE Transactions on Audio and Electroacoustics*, **19**(1), 48–57.

3. **J.M.P. Langlois, D. A.-K.** (2002). A low power direct digital frequency synthesizerwith 60 dbc spectral purity.

4. **L, A.** (2018). Sinus wave genarator with verilog and vivado. URL https://miscircuitos.com/sinus-wave-generation-with-verilog-using-vivado-for-a-fpga/.

5. **Ming-gang Gan, L. C.-T. Y., Jie Chen** (2009). The design of direct digital frequency synthesis based on rom lookup table. *International conference on Information Engineering and Computer science*.

6. **Vankka, J.** (2001). Direct digital synthesizers: Theory, design and applications.

7. **Ying-Shen Juang, J.-E. C. T.-Y. S. a. H.-C. H., Lu-Ting Ko** (2012). Optimization and implementation of scaling-free cordic-based direct digital frequency synthesizer for body care area network systems. *Computational and Mathematical Methods in Medicine*, **2012**(8), 9.