RESEARCH ARTICLE OPEN ACCESS

FPGA Implementation of DDS for Arbitrary wavegeneration

Aswathi Anil
PG Scholar,Dept. of Electronics Mar Athanasius College of Engineering
Kothamangalam,Kerala,686666

Athira Prasad
Assistant Professor,Dept. of ElectronicsMar Athanasius College of Engineering
Kothamangalam,Kerala,686666

ABSTRACT
Direct digital synthesis is a method to generate waveforms directly in the digital domain. Arbitrary Waveform Generator (AWG) is an important signal generation module in many applications such as communication and mixed signal testing. This project proposes an AWG based on Direct Digital Synthesizer (DDS).The common way to generate arbitrary wave- form is based on the look-up ROM.But due to the limitation of ROM size, it is difficult to realize high resolution and high precision DDS by lookup table method. So in this work Cordic algorithm is used for that purpose. CORDIC is a versatile algorithm widely used for VLSI implementation of digital signal processing applications. CORDIC algorithm provides fast and area efficient computations of sine and cosine functions without using ROM LUTs. The proposed DDS is implemented on FPGA using Verilog. Direct Digital Synthesizer using Cordic approach, increase the speed with minimum area requirement in FPGA.

***Index Terms*—**Direct Digital Synthesizer(DDS), Arbitrary wavegeneration, Cordic , FPGA, ROM LUT, PAC

Date of Submission: 29-06-2021 Date of Acceptance: 13-07-2021

I. INTRODUCTION

Arbitrary Waveform Generator (AWG) is an important signal generation module in many applications such as communication and mixed signal testing.There are many cases where certain signals can cause your device to malfunction.So we have to properly test a product during its development.Testing an electrical/electronic device or system implies applying the right stimuli to the device under test (DUT) and analyzing the resulting behavior. In some cases, stimuli come from the real world but in most situations, a set of signals must be supplied by appropriate instrumentation.So the AWG is a form of test equipment that can generate a broad variety of stimuli so that the operating range of the DUT may be established and validated.It can determine system or product noise,timing problems,signal level abnormalities,bandwidth loss,harmonic distortion etc.Arbitrary waveform generators are very similar to function generators, but offer much greater levels of flexibility in terms of waveform generation and they are generally more sophisticated and hence more costly.

The common way to generate arbitrary waveform is based on the look-up ROM.However, a ROM usually operates at a frequency one order of magnitude lower than the accompanying logic.Furthermore, the size of the required ROM increases exponentially with the bit number of phase accumulator,thus becomes unacceptable even for moderate resolution.Thus a more beneficial technique called Direct Digital Frequency Synthesis(DDS) is used in this work to design an AWG.DDS is a technique of generating analog waveforms using digital methods.DDS technique has the following features such as extremely fast "hopping speed" in tuning output frequency (or phase),eliminates the need for the manual system tuning and tweaking,easy implementation with FPGA.

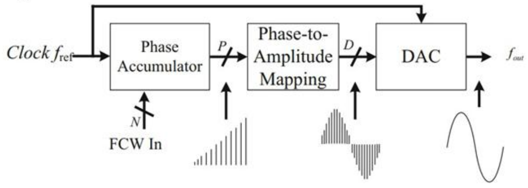


Fig. 1. Basic DDS architecture

A simplified DDS architecture is shown in Figure1.It consists of three blocks: the phase accumulator (PA), the phase-to amplitude converter (PAC) and a Digital to Analog Converter(DAC). An external reference is provided to both the phase accumulator and the DAC. The phase accumulator may be thought of as a numerically controlled oscillator, which derives its output from the reference clock.The PA integrates and periodically wraps the value of FCW, frequency control word, which is the input of DDS to determine the frequency of the generated sinusoid. The output of PA is“angle of phase”, which will be converted to the sinusoidvalue by PAC. The output of PAC is the phase angle of the sinusoid and is digital in form, which will be converted into the analog form by DAC.

The frequency of the output sinewave is given by theequation:

$$f_{out} = \frac{FCW * f_{ref}}{2^N} \qquad (1)$$

FCW is the binary number programmed into register, N is the bit number of PA and fref is the clock frequency.

A. *Frequency tuning equation*

A sine wave is generally expressed as a(t) = sin (ωt)which is non-linear and not easy to generate except through con- structing it from pieces. However, the angular information is linear because the phase angle rotates through a fixed angle for each unit of time. Thus, the angular rate depends on the frequency of the signal described as ω = 2πf , where,ω is the angular frequency. As shown in Figure 2, the phase increases linearly from 0 to 2π over one complete cycle of the sine wave.

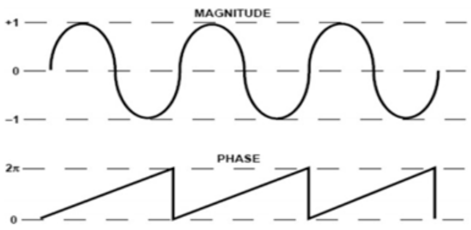


Fig. 2. Sine magnitude and phase representation

Knowing that the phase of a sine wave is linear and that it depends on a reference clock period, with clock frequency ,the phase rotation (Δp) for that period can be determined by

$$\Delta p = \omega \Delta t \qquad (2)$$

ω is angular frequency of wave,Δt is small change in time. Solving for ω in Equation 2, gives

$$\omega = \frac{\Delta p}{\Delta t} = 2\pi f \qquad (3)$$

The overflowing accumulator (PA) clocked with .f_{ref} generates the phase value sequence,

$$f_{ref} = \frac{1}{\Delta t} \qquad (4)$$

Solving for, from Equation 3 and substituting the reference clock frequency for the reference period in Equation 4, specifies the frequency of the output signal:

$$f_{out} = \frac{\Delta p * f_{ref}}{2\pi} \qquad (5)$$

Finally, for an n-bit accumulator the output signal will have the frequency specified

$$f_{out} = \frac{\Delta p * f_{ref}}{2^n} \qquad (6)$$

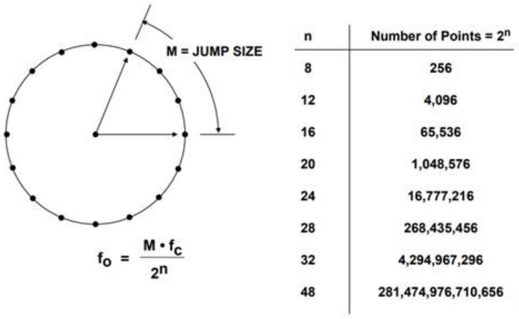
where, Δp(in degree) is the phase increment word or fre-quency control word or frequency tuning word and f_{ref} is the clock frequency, n is the length of accumulator. This phase value Δp is generated using the modulo 2^n overflowing property of an n-bit PA. The rate of the overflow is the output frequency given by Equation 6. Δp , is an integer, thereforethe frequency resolution is found by setting Δp = 1,

$$\Delta f = \frac{f_{ref}}{2^n} \qquad (7)$$

B. *Phase accumulator*

The heart of the system is the phase accumulator whose contents is updated once each clock cycle.The PA consists of an N bit adder and phase register.Each time the phase accumulator is updated, the digital number, M or FCW , stored in the frequency register is added to the number in the phase accumulator register. Assume that the number in the delta phase register is 00...01 and that the initial contents of the phase accumulator is 00...00. The phase accumulator is updated by 00...01 on each clock cycle. If the accumulator is 32-bits wide, 232 clock cycles (over 4 billion) are required before the phase accumulator returns to 00...00, and the cycle repeats.The output of phase accumulator serves as the input to the PAC.

To understand this basic function, consider the sine-wave os- cillation as a vector rotating around a phase circle as shown in figure 3.Each designated point on the phase wheel corresponds to the equivalent point on a cycle of a sine wave. As the vector rotates around the wheel, visualize that the sine of the angle generates a corresponding output sine wave. One revolution of the vector around the phase wheel, at a constant speed, results in one complete cycle of the output sine wave. The phase accumulator provides the equally spaced angular values accompanying the vector's linear rotation around the phase wheel. The contents of the phase accumulator correspond to the points on the cycle of the output sine wave. For an N-bit phase accumulator there are 2^N possible phase points.Themagnitude of the increment is determined by the binary-coded input word (M or FCW). This word forms the phase step size between reference-clock updates; it effectively sets how many points to skip around the phase wheel. The larger the jump size, the faster the phase accumulator overflows and completes the equivalent of a sine wave cycle. The number of discrete phase points contained in the wheel is determined by the resolution of the PA (n-bits), which determines the tuning resolution of the DDFS. For example, for an n = 28- bit phase accumulator, M will have a value of 0000...0001, which would cause the phase accumulator to overflow after 228 reference-clock cycles (increments). If the value of M is changed to 0111...1111, phase accumulator will overflow after only 2 reference-clock cycles (the minimum required by Nyquist). This relationship can be seen in the basic tuning equation for DDFS architecture.



n	Number of Points = 2^n
8	256
12	4,096
16	65,536
20	1,048,576
24	16,777,216
28	268,435,456
32	4,294,967,296
48	281,474,976,710,656

Fig. 3. Digital phase wheel

C. Phase to amplitude converter

The phase-to amplitude converter(PAC) is used to convert the phase accumulator's instantaneous output value into the sine wave amplitude information that is presented to the D/A converter.DDS systems can be implemented with ROM or without ROM.Thus in this work PAC is implemented without using a ROM LUT.Here the PAC is implemented using Cordic algorithm.The different methods to implement PAC is discussed in the literature survey.

D. Digital to analog converter and filter

The phase accumulator computes a phase (angle) address for the PAC, which outputs the digital value of amplitude—corresponding to the sine of that phase angle—to the DAC. The DAC, in turn, converts that number to a corresponding value of analog voltage or current. The DAC and rest of the system run at the same reference clock for synchronization.

The DAC adds quantization error at the output to the sine wave. Ideally (sinx/x) is used to filter the output of the DAC. It removes the extra frequency components added to the sine wave and hence produces a smooth sine wave.

E. Literature Survey

Phase-to-amplitude conversion module is the focus of DDS research.There exist different methods of mapping from phase to amplitude.In conventional DDS architecture, the traditional Phase to Sine Amplitude Conversion (PAC) has been implemented by using a basic ROM look up table(LUT) method[1].Here the amplitude data of each phase are precalculated and stored in the ROM.The ROM LUT based method is simple and fast but the size of the ROM increases exponentially with number of phase bits,N.So when high resolution is required the ROM size get increased.Thus hardware resources will be consumed greatly.Area and power consumption increases and eventually the speed is reduced. Several ROM compression techniques have been proposed to optimize the ROM size.With the quarter-wave symmetry method[2], the ROM only needs to store one-quarter(the phase from 0 to pi/2) of the sine wave sequences.This method exploits the quarter wave symmetry of sine wave. Using this technique a ROM compression ratio of 4:1 is obtained.In the sine amplitude approximation method[3], the approximate sinusoid amplitude value is pre-calculated by approximation circuit. At the same time, an error compensation ROM is exploited to store the error value between the real value and the approximate value. Then, the real value is reconstructed by adding the approximate value and the error value with an adder.

In Polynomial Approximation method[4] the sine function is approximated with a polynomial.By doing so,higher order approximations can achieve high sample accuracy.But this method is rarely used because it requires multiplication,squaring and other power functions for high order cases and that will incraese hardware and computational complexity.In order to overcome the deficiencies of conventional DDFSs, a new topology that removes the hungry PAM block and replaces the linear DAC with a sine or cosine-weighted nonlinear DAC (NLDAC) has been developed [5].NLDAC converts digital phase information directly into analog output. Regular NLDACs are typically implemented using a single thermometer decoder with cosine/sine weighted current sources. The complexity of the decoder and the number of switches increases exponentially with the DAC resolution, resulting in a similar exponential growth in both the area and parasitic components in the DAC.NLDAC-based DDFSs are weak in terms of resolution and dynamic performance due to the limited DAC accuracy.

CORDIC (COordinate Rotational DIgital Computer) theorywas introduced by Volder in 1959. Because of the simplicityof the algorithm, it has got a wide spread acceptance and now a days it is commonly used for the calculation of mathematical functions. It is a rotation based algorithm[6].The key idea is continously rotate a specific angle through an iterative method,so that the sum of the cumulative rotation angles is close to the target angle.CORDIC uses only adder/subtractor and shifter for calculating functions such as trignometric,exponential etc.So it eliminates the use of complex hardware multipliers thus saving a lot of area and power consumption.In this work the DDS is implemented based on CORDIC algorithm[7].It improves the speed and resolution of the system.And also reduces the consumption of a lot of hardware resources.

II. IMPLEMENTATION OF CORDIC

CORDIC is a hardware efficient iterative method CORDIC is not a unidirectional rotational algorithm. For obtaining the desired angle, it is possible to rotate in both clockwise and anticlockwise directions. Hence requires an extra variable to represent the direction of rotation and variable 'd' is included for that. Variable 'd' can be

$$\begin{aligned} x_{i+1} &= (x_i - d_i y_i 2^{-i})K_i \\ y_{i+1} &= (y_i + d_i x_i 2^{-i})K_i \\ z_{i+1} &= (z_i - d_i \vartheta_i)K_i \end{aligned} \qquad (11)$$

d_i represents the direction of rotation and K_i is the scaling factor of ith iteration stage.For each iteration a comparison is done between the target angle and previous angle.Then the comparison sign(sign of z_i) is used to determine the direction of next rotation. For $z_i > 0, d_i = +1$ represents counterclockwise rotation and for $z_i < 0, d_i = -1$ represents clockwise rotation After completion of desired number of rotation,the result obtained is given by

$$\begin{aligned} x_n &= cosz_0 \\ y_n &= sinz_0 \\ z_n &= 0 \end{aligned} \qquad (12)$$

which use rotations to calculate a wide range of elementary functions. That means an initial vector (x_0, y_0) is chosen and is subjected to rotation.CORDIC is classified into two: Rotational mode and vectoring mode.In this work, rotational mode is used. That means, the initial vector (x_0, y_0) will rotates n times and hence obtain x_n and y_n.In general, it can be written as

$$\begin{aligned} x_2 &= x_1 \cos\vartheta - y_1 \sin\vartheta \\ y_2 &= x_1 sin\vartheta + y_1 cos\vartheta \end{aligned} \qquad (8)$$

The above equation contains four multiplications and two additions. Hence it is not feasible to use directly in the system. To make it more simple, we take the $cos\vartheta$ term outside.Thus Eq. (8) is simplified into Eq. (9)

$$\begin{aligned} x'_2 &= x_1 - y_1 tan\vartheta \\ y'_2 &= y_1 + x_1 tan\vartheta \end{aligned} \qquad (9)$$

The term $cos\vartheta$ is independent of iterations and having a con-stant value of 0.60725. In order to convert the multiplicationto addition, the term $tan\vartheta$ is restricted to 2^{-i}.So multiplication can be replaced with arithmetic right shift. Here i is an arbitrary integer number. Thus Eq. (9) is transformed into

$$\begin{aligned} x'_2 &= x_1 - y_1 2^{-i} \\ y'_2 &= y_1 + x_1 2^{-i} \end{aligned} \qquad (10)$$

either positive or negative.The value of d is decided by the rotation angle,z_i.Thus an iterative phase accumulating algorithm is obtained:

z_n = 0 represents the vector is basically coincident with the target vector by continuous rotation.

For the generation of sine or cosine wave by CORDIC based approach, phase accumulator output is initialized as the desired rotation angle,z_0. Along with that the initial x and y values as 1/K and 0 respectively, are given to CORDIC block. The sine and cosine values of the angles are simultaneously obtained in the x and y coordinates after completing the iterations.Since CORDIC algorithm converges around –pi/2 to pi/2, the two MSB bits of phase accumulator is used to generate wave for a complete cycle of period 0 to 2 pi by quadrant mapping. CORDIC is not a unidirectional rotational algorithm. For obtaining the desired angle, it is possible to rotate in both clockwise and anticlockwise directions. Hence requires an extra variable to represent the direction of rotation and variable 'd' is included for that. Variable 'd' can be either positive or negative.The value of d is decided by the rotation angle,zi , as shown in the flowchart in the figure 4.i denotes the number of iterations.Based on the direction of rotation adders/subtractors are utilized. zn = 0 represents the vector is basically coincident with the target vector by continuous rotation.

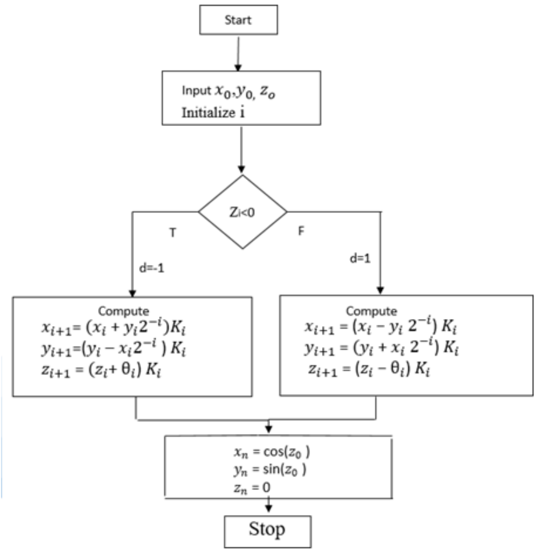

Start
Input x_0, y_0, z_o
Initialize i
Zi<0
T
F
d=-1
d=1
Compute
$x_{i+1} = (x_i + y_i 2^{-i}) K_i$
$y_{i+1} = (y_i - x_i 2^{-i}) K_i$
$z_{i+1} = (z_i + \theta_i) K_i$
Compute
$x_{i+1} = (x_i - y_i 2^{-i}) K_i$
$y_{i+1} = (y_i + x_i 2^{-i}) K_i$
$z_{i+1} = (z_i - \theta_i) K_i$
$x_n = \cos(z_0)$
$y_n = \sin(z_0)$
$z_n = 0$
Stop

Fig. 4. Design flow of cordic algorithm

III. SYSTEM ARCHITECTURE

The proposed system implements arbitrary waveform generation based on Direct Digital Frequency Synthesis and Cordic algorithm.It is implemented on FPGA SPARTAN-3E board using verilog HDL.The phase accumulator will control the frequency of output signal.The FCW provides the main input to the phase accumulator.The FCW corresponding to required output frequency is calculated and the binary number obtained is stored in a frequency register. The phase accumulator consists of an N bit adder and phase register.At each positive edge of the reference clock cycle the FCW is added to the value previously held in the phase register inorder to generate appropriate phase increment.So at any instant the value in the PA represents the phase of the sinusoid.The output of PA is given to the PAC.

Here the PAC is implemented using Cordic. Hence when an overflow is occured in the accumulator, the phase value will be given to the cordic section.The accumulator output is

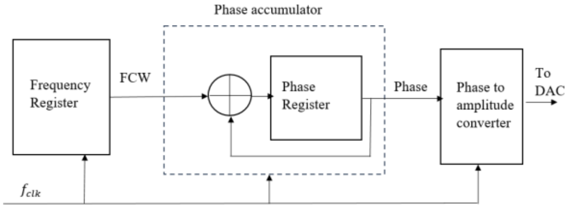


Fig. 5. Circuit architecture of Phase accumulator

initialized as the desired rotation angle,z_0. Along with that the initial x and y values as 1/K and 0 respectively, are given to CORDIC block. The cosine and sine values of the angles are simultaneously obtained in the x and y coordinates after completing the iterations.Since CORDIC algorithm converges around –pi/2 to pi/2, the two MSB bits of phase accumulator is used to generate wave for a complete cycle of period 0 to 2 pi by quadrant mapping.There is adders/subtractors,shift registers, mux and an LUT for storing rotation angles.The LUT in cordic contains fixed angle constants. Combining all these modules the cordic algorithm is implemented.And by combining the design of phase accumulator and cordic, the sine and cosine corresponding to the desired frequency can be generated.

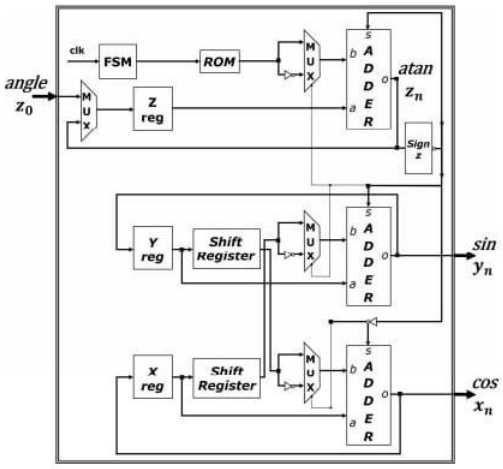


Fig. 6. RTL architecture of cordic

The STM32F103C8T6 module is used to give user inputs(FCW) to FPGA.A UART is included to input some inbuilt arbitrary functions into the system so that we can generate any arbitrary wave. The phase accumulator and Cordic algorithm is implemented on FPGA SPARTAN-3E.Thus the phase accumulator generates the phase corresponding to input frequency.After that the CORDIC algorithm calculates the sine and cosine function corresponding to the phase angle.After the processing of FPGA a time-varying signal in digital form is generated and then performing a digital-to-analog conversion the desired output is obtained.

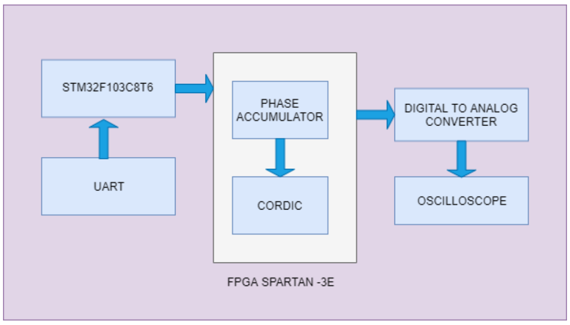


Fig. 7. Detailed view of RTL schematic of PA

A. DESIGN STEPS

STEP 1: Calculate the FCW corresponding to desiredoutput frequency using equation 1.

STEP 2: Store the binary format of calculated FCW into a register.

STEP 3: Initialize the value in phase register of PA to zero.When reset goes low with each clock pulse the FCW in the register is added to the value previously held in the phase register.This goes on until the accumulator overflows and the cycle starts again.Each time the accumulator overflows thePA outputs the phase corresponding to the sine amplitude.

STEP 4: The output of PA serves as input to the cordic section. Initialize $x_0 = 1/K$ and $y_0 = 0$ (K is the scaling factor and it is approximated to 0.60725) and z_0 is initialized to the target angle(phase from PA).

STEP 5: At first the initial vector is rotated by 45°.

STEP 6: Compare $z_0$and 45° and store the result in z_1.

STEP 7: If $z_1 > 0$,the direction of rotation d_1 is taken as +1 which implies counterclockwise rotation.If $z_1 < 0$,the direction of rotation d_1 is taken as -1 which implies clockwise rotation.

STEP 8: Input to shift registers are the given initial coordinates or the input to a particular iteration step:x_i,y_i.Now depending on the iteration step (value of i), $x_i$andy_i are right bit shifted and then added or subtracted depending upon the value of the decision vector z_i, with $x_i$andy_i respectively to generate $xi+1$and$yi+1$.

STEP 9:The value of z_1 is added to the next stored angle (22.5°) or the next stored value (22.5°) is subtracted from z_1, depending on the value of z_1. The result generated is stored in variable z_2.

STEP 10: Steps 7 to 9 are repeated for the required number of iterations to get the desired result.Finally y coordinate of the vector will have sine and x coordinate ofthe vector will have cos value.

IV. IMPLEMENTATION AND RESULTS

The circuit architecture of phase accumulator is shown in figure 5.It is coded using Xilinx ISE and simulated in Modelsim.The detailed RTL schematic of phase accumulator generated from Xilinx ISE simulator is shown in figure 8 and figure 9 shows the corresponding technology schematic of PA.

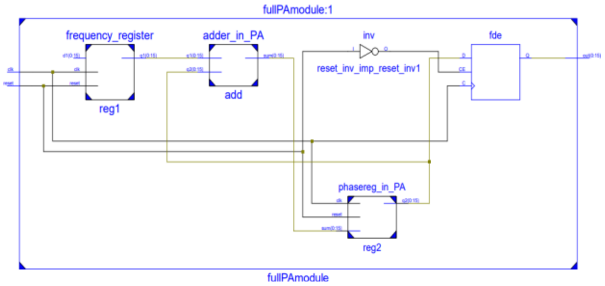


Fig. 8. Detailed view of RTL schematic of PA

To generate a fixed-frequency sinusoidal wave, a constant value phase increment that is decided by the FCW (binary number) is added to the phase accumulator with every clock cycle.The phase accumulator having an N- bit adder accumu- late the phase angle value by frequency control word recur- sively.Frequency control word is loaded into a register by usingclk and reset signal.When reset goes low the calculated FCW value(using equation 1) corresponding to required frequency is loaded into the register.The phase accumulator content is initialized to zero by using reset signal.When reset goes low with each clock pulse the FCW is added to the previous PA content.This goes on until the accumulator overflows.The output of phase accumulator simulated in modelsim is shown in figure 9.

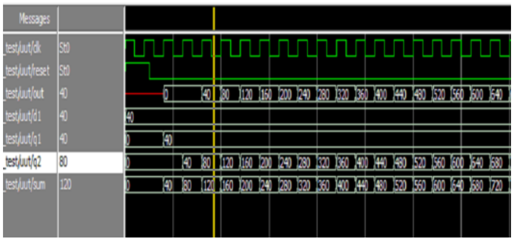


Fig. 9. Output of Phase accumulator

Figure 6 shows the RTL design of Cordic.The programme is designed based on this.There are adders/subtractors,shift registers, mux and an LUT for storing rotation angles.The LUT in cordic contains fixed angle constants. Combining all these modules the cordic algorithm is implemented.Hardware implementation for CORDIC arithmetic requires three regis- ters for x, y and z, two shifter to supply the terms $2^{-i}x_i$ and $2^{-i}y_i$ to the adder/subtractor units and a look up table to store the values of $\vartheta_i = \tan^{-1} 2^{-i}$.The d_i factor (-1 and 1) selects the shift operand or its complement. The initial inputs to the architectures are X0=1/K, Y0=0.

The structure requires a pre-processing unit to converge the input angles to the desired range and a post processing unitto fix the sign of outputs depending on the initial angle quadrants. The pre-processing unit takes in angles of any range and converges it to the interval [-pi/2, pi/2]. It keeps record of the quadrant of the input angle which may be used in the post-processing unit to fix the sign of outputs. These two blocks are inevitable for any application as the input range cannot be predicted always.And by combining the design of phase accumulator and cordic, the sine and cosine corresponding to the desired frequency can be generated.This model has been implemented using Verilog and simulated in ModelSim.Figure 10 shows snapshot of RTL schematic of CORDIC processor(top) generated from Xilinx ISE simulator. Here inputs are angle (binary input), clk (clock), start and outputs are sine (binary output), cosine (binary output), done.

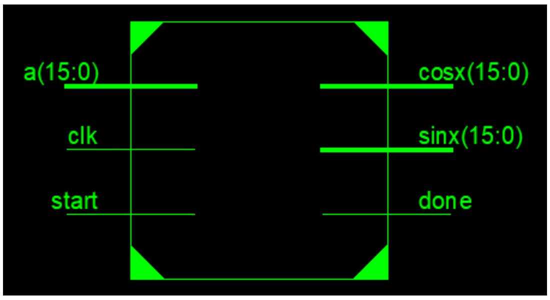


Fig. 10. RTL schematic of cordic processor

The phase to amplitude converter module is the focus of DDS technique.The PAC can be implemented using different methods.Here a comparison is made between PAC implemented using ROM LUT and Cordic algorithm. If a sine look-up table is used as PAC, the amplitude data of each phase are pre-calculated and stored in the LUT.The PA computes a phase (angle) address for this look-up table. Each address in the LUT corresponds to a phase point on the sine wave from 0° to 360°. The LUT contains the corresponding digital amplitude information for one complete cycle of a sine wave. The LUT outputs the digital value of amplitude— corresponding to the sine of that phase angle—to the DAC.The output of DDS based on LUT method is shown in figure 11.It is simulated using Modelsim SE 6.5. Inorder to generate an output frequency of 7.8 MHz with 50MHzclock frequency , the FCW is calculated to be 40.It's binary form is loaded into frequency register.When reset goes low with each clock pulse the FCW is added to the value previously held in the phase register.This continues until the phase accumulator overflows and the cycle starts again. The PA used is 8 bits.So the PA overflows after 2^8 reference clock cycle increments.Each time the PA overflows, the LUT outputs sampled values of the sine wave.

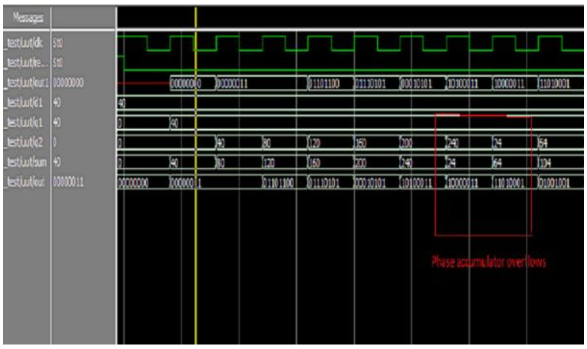
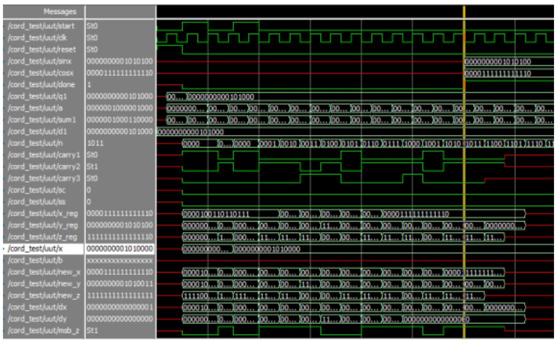


Fig. 11. Simulation result of DDS based on ROM LUT

In the previous simulation, the PA was 8-bit in length. Therefore, The LUT has 2^8 entries i.e. 256 values and each entry is 8-bit in length. Therefore, the size of the LUT is256*8 i.e. 65536. Thus the size of the LUT ROM increases exponentially with the bit number of phase accumulator.The ROMLUT method occupies maximum memory space and it utilizes more number of resources on FPGAs and also reducesthe speed. In order to minimize the area utilization in FPGA, the CORDIC architecture is used in this design instead of ROMLUT.Using cordic, the sine and cosine correspondingto input phase is obtained without using massive LUTs.The simulated output is shown in figure 12.This is also simulated using Modelsim SE 6.5.The clock rate of the implemented design is 50 MHz.The generated frequency is 7.8 MHz.The number of iterations used in the programme is 11. Therefore the output is obtained after 11 iterations and as number of iterations increase the output will become more precise.

V. COMPARISON OF GENERATOR METHODS

The Performance of Rom Based Dds and Cordic Based DDS can be analysed from the total real time taken for Xst completion which is 101.00secs for ROM LUT and 20.00secs for cordic based DDS.Total CPU time taken for Xst completion of ROM based DDS is 100.49secs and it is 20.46 secs for cordic based DDS.The ROM LUT based method occupies maximum memory space and it utilizes more number of resources on FPGAs compared to cordic DDS.The performance in speed and area is shown in table I.As a result a comparison is made between DDS based on ROM LUT and Cordic algorithm. The advantages and disadvantages of both methods are summarized below in table II.



```
Messages
/cord_test/uut/start     St0
/cord_test/uut/clk       St0
/cord_test/uut/reset     St0
/cord_test/uut/sinx      0000000001010100
/cord_test/uut/cosx      0000111111111110
/cord_test/uut/done      1
/cord_test/uut/q1        0000000000101000
/cord_test/uut/a         0000001000001000
/cord_test/uut/sum1      0000001000110000
/cord_test/uut/d1        0000000000101000
/cord_test/uut/n         1011
/cord_test/uut/carry1    St0
/cord_test/uut/carry2    St1
/cord_test/uut/carry3    St0
/cord_test/uut/sc        0
/cord_test/uut/ss        0
/cord_test/uut/x_reg     0000111111111110
/cord_test/uut/y_reg     0000000001010100
/cord_test/uut/z_reg     1111111111111110
/cord_test/uut/x         0000000001010000
/cord_test/uut/b         xxxxxxxxxxxxxxxx
/cord_test/uut/new_x     0000111111111110
/cord_test/uut/new_y     0000000001010011
/cord_test/uut/new_z     1111111111111111
/cord_test/uut/dx        0000000000000001
/cord_test/uut/dy        0000000000000000
/cord_test/uut/msb_z     St1
```

Fig. 12. Simulation result of DDS based on cordic

Technique	Number of slices used	Number of 4 input LUTs	Total REAL time to Xst completion	Total CPU time to Xst completion	Total memory usage
ROM LUT DDS	165	226	101.00secs	100.49secs	5043356 Kb
CORDIC DDS	119	194	20.00 secs	20.46secs	4510608 Kb

TABLE I
SPEED AND AREA PERFORMANCE

METHOD	BENEFITS	DRAWBACKS
ROM LUT	• Easy to implement • Easy to set up • Simple calculations	• Low precision • Larger area on FPGA
CORDIC	• Better precision • High performance • Smaller chip area usage	• Requires more calculations • Complex program design

TABLE II
COMPARISON OF GENERATOR METHODS

VI. CONCLUSION

An Arbitrary waveform generator(AWG) is a form of test equipment that is capable of generating a broad variety of signals.The resulting waveforms can be injected into a device under test and analyzed as they progress through it, confirming the proper operation of the device.Different techniques for arbitrary waveform generation has been studied. In this work an AWG based on Direct Digital Synthesis technique is proposedto implement on FPGA.The proposed system's architecture has been designed in Altium designer.The design is coded in verilog and simulated in Modelsim. The proposed architecture has been compared with the prior architectures in simulations. This proposed design provides high speed,high resolution and minimum resource utilization compared to conventional DDS architectures.

REFERENCES

[1]. M. Gan, J. Chen, L. Cheng and T. Yu, "The Design of Direct Digital Frequency Synthesis Based on ROM Lookup Table," 2009 International Conference on Information Engineering and Computer Science, Wuhan, 2009, pp. 1-3, doi: 10.1109/ICIECS.2009.5363868.

[2]. H. Chen, J. Chen, Y. Li and Y. Wu, "A Quarter-ROM DDS Using Phase Accumulator with Subtraction," 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taichung, 2018, pp. 1-2, doi: 10.1109/ICCE-China.2018.8448686.

[3]. M. P. Langlois and D. Al- Khalili , "Phase to sinusoid amplitude conversion techniques for direct digital frequency synthesis," in IEE

Proceedings - Circuits, Devices and Systems, vol. 151, no. 6, pp. 519- 528, 15 Dec. 2004, doi: 10.1049/ip-cds:20040500.

[4]. C. Meenakarn and A. Thanachayanont, ”A sine-output ROM-less di- rect digital frequency synthesiser using a polynomial approximation,” Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS ’03., Bangkok, 2003, pp. I-I, doi: 10.1109/IS- CAS.2003.1205641.

[5]. S. Mortezapour and E. K. F. Lee, ”Design of low-power ROM-less direct digital frequency synthesizer using nonlinear digital-to-analog converter,” in IEEE Journal of Solid-State Circuits, vol. 34, no. 10, pp. 1350-1359, Oct. 1999, doi: 10.1109/4.792601.

[6]. Li, J. Fang, B. Li and Y. Zhao, ”Study of CORDIC algorithm based on FPGA,” 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, 2016, pp. 4338-4343, doi: 10.1109/CCDC.2016.7531747.

[7]. Y. SU, Z. KUANG, G. ZHANG and L. SUN, ”A 14-bit DDS Circuit Design Based on CORDIC Algorithm,” 2019 IEEE International Confer- ence of Intelligent Applied Systems on Engineering (ICIASE), Fuzhou, China, 2019, pp. 321-323, doi: 10.1109/ICIASE45644.2019.9074119.