DIRECT DIGITAL SYNTHESIS BASED CORDIC ALGORITHM: A NOVEL APPROACH TOWARDS DIGITAL MODULATIONS

Prajakta J. Katkar[1], Yogesh S. Angal[2]

[1]PG student with Department of Electronics and telecommunication, BSIOTR, Wagholi, Pune, Savitribai Phule University, Pune, Maharashtra, India

[2]Head of Department of Electronics and telecommunication, BSIOTR, Wagholi, Pune, Savitribai Phule University, Pune, Maharashtra, India

Abstract

Modulation is the technique in which carrier signal varies according to amplitude of modulating signal. A brilliant solution for realizing digital modulators is CORDIC (CO-ordinate Rotation Digital Computer) algorithm. Rotation mode and vector mode are two modes in which this algorithm is used. Here rotation mode is used to convert the coordinates from polar mode to rectangular mode. This paper presents the implementation of different communication subsystems like ASK, FSK, PSK, BPSK, QPSK, 4 QAM, 16 QAM that can be found in software defined radio by using CORDIC algorithm. The focus of this paper is to analysis and simulation of modulation scheme using Direct Digital Synthesizer having CORDIC algorithm.

Keywords: *Software Defined Radio, CORDIC algorithm, DDS, ASK, FSK, PSK, BPSK, QPSK, 4-QAM, 16-QAM.*

--***--

1. INTRODUCTION

The Coordinate Rotation Digital Computer (CORDIC) was introduced in 1959 by Volder [1]. It is an easy-to-implement and versatile algorithm widely used for digital signal processing applications. It calculates the rotation of a two-dimensional vector using only add and shift operations. CORDIC also used to implement different communication subsystems found in a digital radio: direct digital synthesizers; Quadrature phase shift keying(QPSK), amplitude shift keying (ASK),phase shift keying (PSK), frequency shift keying (FSK), and Quadrature amplitude modulation(QAM) ,Binary phase shift keying (BPSK) modulators.

2. FUNDAMENTAL CONCEPTS OF CORDIC

For an easy understanding of how to use the CORDIC algorithm in the implementation of digital intermediate frequency (IF) communications systems, CORDIC is presented only as a computational resource with three inputs (X_0, Y_0, and Z_0) and three outputs (X_N, Y_N, and Z_N) that allows performing the following operations[3]as shown in Figure 1.

A generic scheme that shows how to use RM CORDIC to implement different digital communication. The scheme is composed of an RM CORDIC where signals I and Q are connected into X_0 and Y_0inputs, and the phase term q connected into Z_0 input is $q\ =\ (\sum[fc+\ fm]\ )\ +\ \O m\ )\ \times\ \pi$.

Direct digital synthesis is the process of generating sine waveforms directly in the digital domain. DDS consists of a phase accumulator and a phase-to-waveform converter. The phase-to-waveform converter could be realized by an RM-CORDIC as shown in Figure 4. The cosine and sine waveforms are obtained respectively by the CORDIC outputs X_N and Y_N.

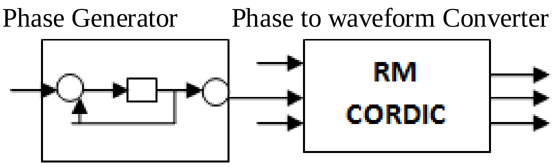


Fig 1 CORDIC-based Direct Digital Synthesizer $\theta = \sum f_c * \pi$.

CORDIC computes a pseudo-rotation of a two-dimensional vector instead of perfect rotation. This means that the orignal vector is rotated by an angle q, and its magnitude is enlarged by a constant factor *K*.

3. CORDIC APPLICATIONS USED IN COMMUNICATION SYSTEM

3.1 Direct Digital Synthesis

To generate waveforms directly in the digital domain is a method direct digital synthesis. The sine wave and cosine waves are important in communications systems. A DDS is combination of a phase accumulator and a phase-to amplitude converter as shown in Fig. 2 a. The CORDIC algorithm configured in RM can behave as a quadrature phase-to-amplitude converter that directly generates sine and cosine waveforms [4]. The main advantage of using CORDIC-based DDS with respect to LUT based methods is that it can achieve both high phase resolution and high precision with lower hardware cost [5]. A difference between both methods is that the phase accumulator generates an integer value that addresses an LUT in the LUT-based method, while it generates an angle in CORDIC-based DDS. Thus, in the last case a ramp signal in the interval [–π, π] must be obtained by the accumulator, as shown in Fig. 2c. This accumulator is easily implemented with an *N*-bit adder. A two's complement fractional numeric format (only one integer bit) is considered; hence, a ramp in the interval [–1, 1] is generated, and a multiplier by *π* is introduced to achieve the desired range.

To generate sine and cosine waveforms of a digital frequency f_c with the scheme based on CORDIC of Fig. 3, the parameters f_m, Φ_m, and *Q* must be zero and I = 1/*K*. The oscillation frequency is controlled by giving a fixed value to *fc*. In such a case CORDIC generates directly the cosine and sine waveforms ($s_i(n) = cos(f_c.\ \pi\ n)$ and $s_q(n) = sin(f_c\ .\ \pi\ .n)$) through X_N and Y_N outputs, respectively.

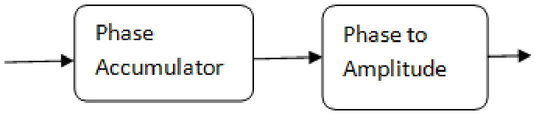


Fig 2 a) DDS block diagram; b) waveforms of the LUT-based method; c) waveforms of the CORDIC-based method

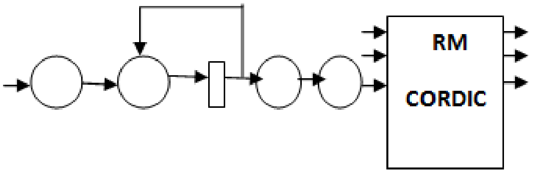


Fig 3 Generic Scheme to use CORDIC in Rotation Mode

3.2 Frequency, Phase and Amplitude Modulators

The CORDIC scheme of Fig. 3 can be used to directly generate in the digital domain at IF the binary modulations ASK, PSK, and FSK.

The CORDIC scheme of Fig. 3 can be used to directly generate in the digital domain at IF Considering *m*(*n*) as the modulator signal, ASK can be implemented by selecting in Fig. 3 carrier frequency f_c, using the input *Xo* as modulator signal $I = m(n)/K$, with f_m, $Ø_m$, and *Q* are zero.. In ASK signal ($s(n) = m(n) \times cos(f_c \times \pi \times n)$) is generated through X_N CORDIC output. For PSK signal the terms f_m and *Q* are zeroed, the input X_o is fixed to $I = 1/K$, and the phase modulator signal is $Ø_m = m(n)$. Then the PM signal ($s(n) = cos(f_c \times \pi \times n + m(n) \times \pi$)) is obtained with a carrier frequency f_c, through the X_N output. For FSK signal the terms $f_m = m(n)$, the carrier frequency is a fixed value f_c, the terms $Ø_m$, and *Q* are zero and the X_0 input is $I = 1/K$. The FSK signal ($s(n) = cos(f_c \times \pi \times n + (\sum m(n)) \times \pi)$) is also obtained by the output X_N.

4. RESULTS AND DISCUSION

VHDL Simulation of Direct Digital Synthesis

The Digital modulation schemes such as of amplitude-shift keying (ASK), frequency shift keying (FSK) and phase-shift keying (PSK), Quadrature phase shift keying(QPSK), Quadrature amplitude modulation(QAM), binary phase-shift keying (PSK), was designed separately using Xilinx ISE 13.2 tool and its simulated results are shown below.

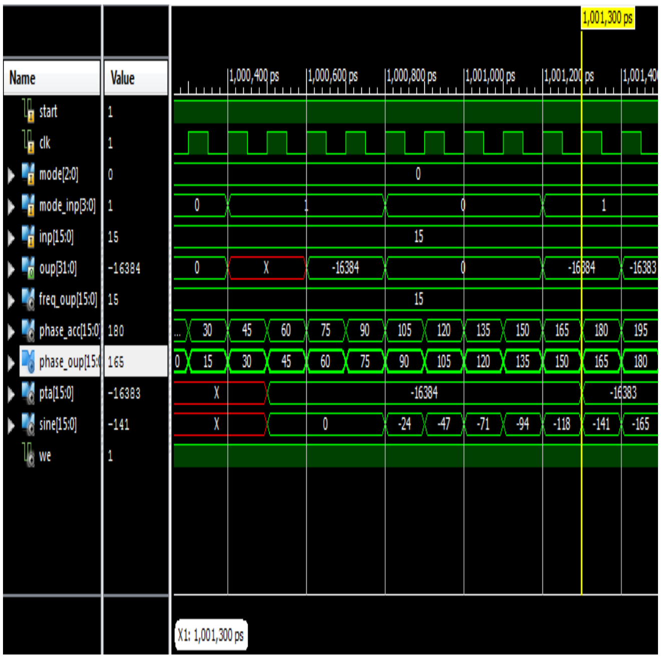
Following figure shows the Xilinx simulation of ASK modulated signal. This system is known as multimode system because depending on different 3 bit modes selected any one of the seven modulatio technique is selects.

Mode input is 4 bit input, for binary 1 "0001" and for binary 0 "0000" is selected. Input is 16 bit. We can select any input from 216=256 combinations. According to that we got the 32 bit output. Phase accumulator adds the inputs at every clock cycle. Phase output is delayed version of phase accumulator.

For ASK, the input and outputs are given bellow and are clearly shown in figure 8.
Mode=000
Mode input= 0000 to 0001
Input= 0000000000001111
Output= 16384 for binary 1 and 0 for binary 0.

Phase accumulator output= For input 15 its adds 15 in each clock cycle ie. 15, 30, 45,60 etc.

This output given as an input to the Phase to amplitude converter as Phase Output with delay of one clock cycle. Output of phase to amplitude converter is multiplied with amplitude modulation control and final output is obtained.



Name	Value
start	1
clk	1
mode[2:0]	0
mode_inp[3:0]	1
inp[15:0]	15
oup[31:0]	-16384
freq_oup[15:0]	15
phase_acc[15:0]	180
phase_oup[15:	165
pta[15:0]	-16383
sine[15:0]	-141
we	1

Fig 4 ASK modulated output in VHDL simulation

For FSK, the input and outputs are given bellow and are clearly shown in figure 8.
Mode=001
Mode input= 0000 to 0001
Input= 0000000000001111

Output= whenever binary 0000 to 0001 transition occurs the phase output gets changed.

Phase accumulator output= For input 15 its adds 15 in each clock cycle ie. 15, 30, 45,60 etc.

This output given as an input to the Phase to amplitude converter as Phase Output with delay of one clock cycle.

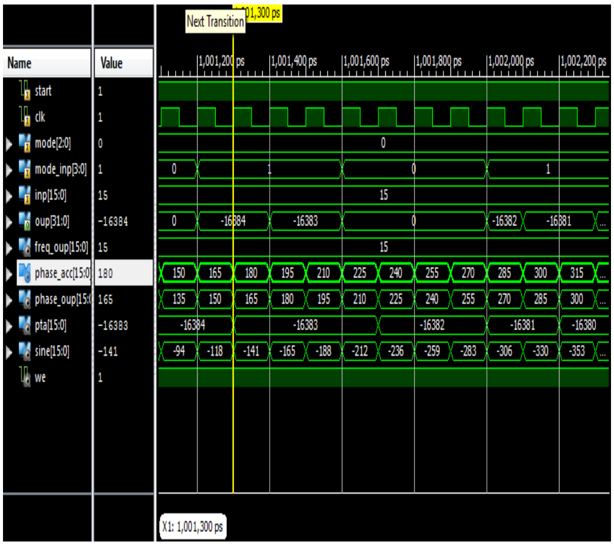


Fig 5 FSK modulated output in VHDL simulation

For PSK, the input and outputs are given bellow and are clearly shown in figure 8.
Mode=010
Mode input= 0000 to 0001
Input= 0000000000001111
Output= whenever binary 0000 to 0001 transition occurs the phase output gets changed.

When 0001 comes then 45º phase shift occurs.

Phase Output= (45º*2^{16})/360º=8192. This is added with 30, 8192+30=8222 which is as shown in following figure.

When output changes from 0001 to 0000 phase output is same as output of phase accumulator with delay of one clock cycle. For input 15 its adds 15 in each clock cycle i.e. 15, 30, 45, 60 etc.

This output given as an input to the Phase to amplitude converter as Phase Output with delay of one clock cycle. At Phase to amplitude converter, it takes cosine of phase output multiplied with $2^{14.}$ Here Cos (0)*2^{14}=16384 is the output. Then angles goes on increasing from 0 to 180 output goes on changing. AT 45°, Cos(45)*$2^{14=}$11585 as shown in following figure. And finally this output is given as final output with delay of one clock cycle.

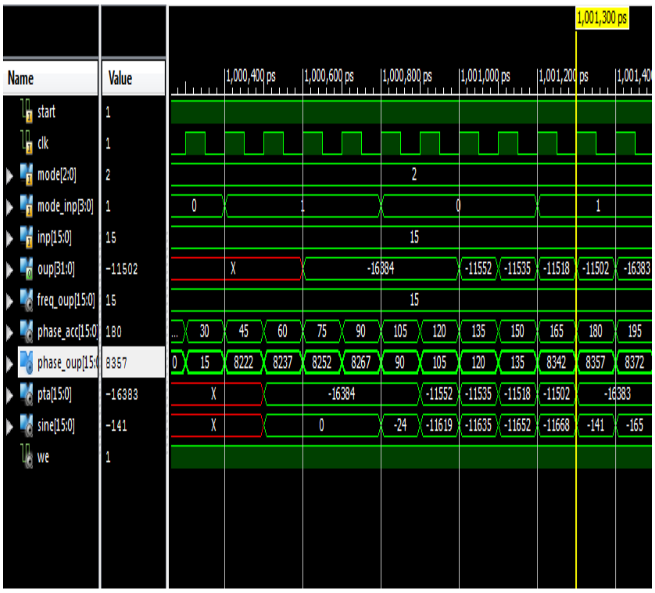


Fig 6 PSK modulated output in VHDL simulation

For BPSK, the input and outputs are given bellow and are clearly shown in figure 8.
Mode=011
Mode input= 0000 to 0001
Input= 0000000000001111
Output= whenever binary 0000 to 0001 transition occurs the phase output gets changed.

When 0001 comes then 180º phase shift occurs.

Phase Output= (180º*2^{16})/360º=32738. This is added with 30, 32738+30=32768 which is as shown in following figure.

When output changes from 0001 to 0000 phase output is same as output of phase accumulator with delay of one clock cycle. For input 15 its adds 15 in each clock cycle i.e. 15, 30, 45, 60 etc.

This output given as an input to the Phase to amplitude converter as Phase Output with delay of one clock cycle.

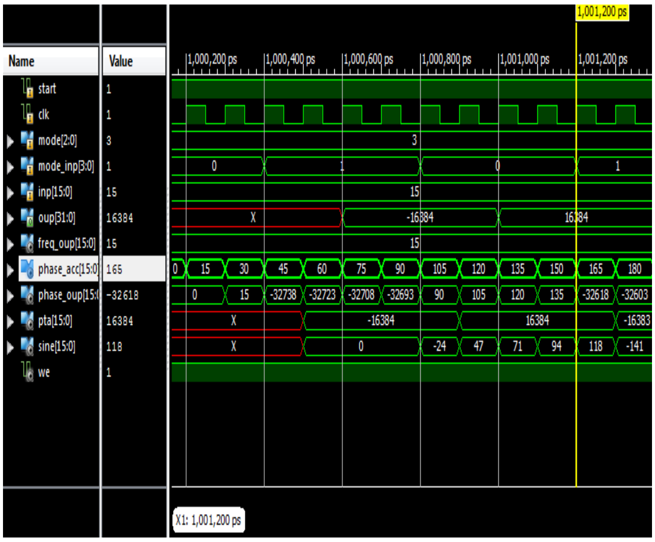


Fig 7 BPSK modulated output in VHDL simulation

For QPSK, the input and outputs are given below and are clearly shown in figure 8.
Mode=100
Mode input= 0000 to 0001
Input= 0000000000001111
Output= whenever binary 0000 to 0001 transition occurs the phase output gets changed.

When 0001 comes then 45º phase shift occurs.

Phase Output= (45º*2^{16})/360º=8192. This is added with 30, 8192+30=8222 which is as shown in following figure. Then 8222 is added with 15, 8222+15=8237 and so on up to Mode input is 0001. Whenever it changes from 0001 to 0000 phase output is same as output of phase accumulator with delay of one clock cycle. Again when mode input changes from 0000 to 0001, 45° phase shift occurs, due to that 8192 is added with previous phase accumulator output. In following figure 8192 is added with 150 to get 8342 as an output. This output given as an input to the Phase to amplitude converter as Phase Output with delay of one clock cycle.

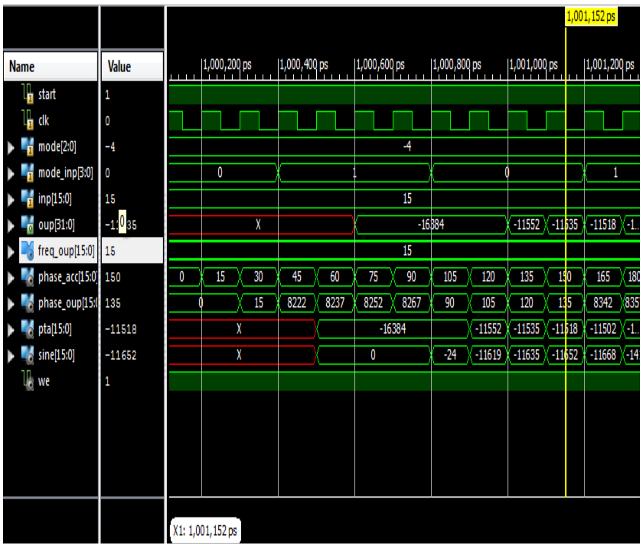


Fig 8 QPSK modulated output in VHDL simulation

For 4-QAM, the input and outputs are given below and are clearly shown in figure 8.
Mode=101
Mode input= 0000 to 0001
Input= 0000000000001111

Whenever output changes from 0000 to 0001 there is 45° phase shift occurs.

Phase Output= (45º*2^{16})/360º=8192. This is added with 15, 8192+15=8207 which is as shown in following figure.

Whenever output changes from 0001 to 0000 there is 135° phase shift occurs.

Phase Output= (135º*2^{16})/360º=24576. This is added with 30, 24576+30=24606 which is as shown in following figure. Then 24606 are added with 15, 24606+15=24621 and so on up to Mode input is 0001.

Phase accumulator output= For input 15 its adds 15 in each clock cycle i.e. 15, 30, 45, 60 etc.

This output given as an input to the Phase to amplitude converter as Phase Output with delay of one clock cycle.

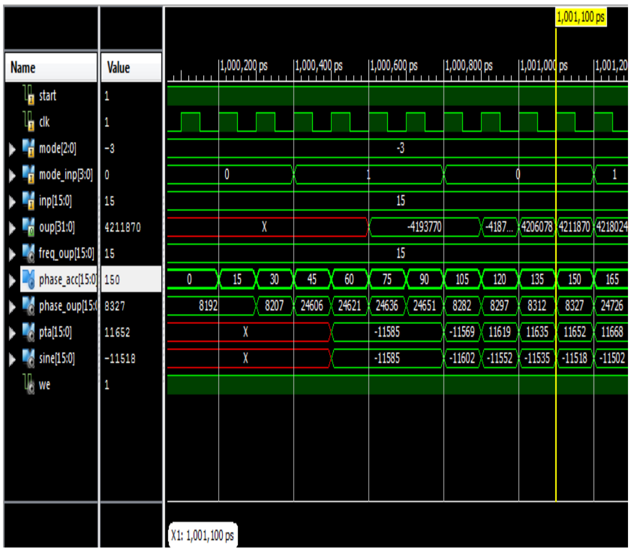


Fig 9 4-QAM modulated output in VHDL simulation

For 16- QAM, the input and outputs are given below and are clearly shown in figure 8.
Mode=110
Mode input= 0000 to 0001
Input= 0000000000001111

Whenever output changes from 0000 to 0001 there is 135° phase shif t occurs.

Phase Output= (135º*2^{16})/360º=24576. This is subtracted with 15, 8192-15=24561 which is as shown in following figure.

Whenever output changes from 0001 to 0000 there is 108° phase shift occurs.

Phase Output= (108º*2^{16})/360º=19739. This is added with 30, 24576+30=24606 which is as shown in following figure. Then 24606 is subtracted with 30, 19739-30=19709 and so on up to Mode input is 0001.

Again when mode input changes from 0000 to 0001, 135° phase shift occurs, due to that 24576 is added with previous phase accumulator output. In following figure 90 is subtracted from 24576 to get 24486 as an output. This output given as an input to the Phase to amplitude converter as Phase Output with delay of one clock cycle.

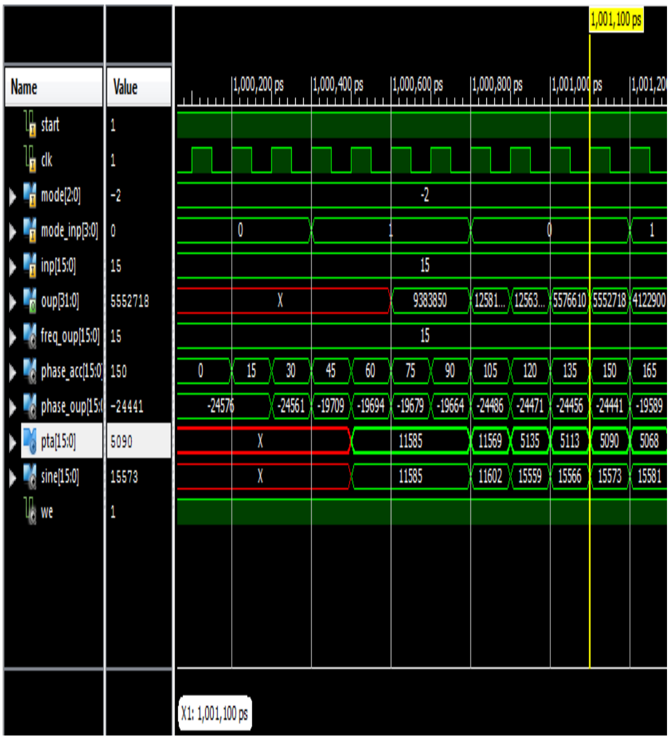


Fig 10 16- QAM modulated output in VHDL simulation

For Normal Waveform, the input and outputs are given below and are clearly shown in figure 8.
Mode=111
Mode input= 0000 to 0001
Input= 0000000000001111

Phase accumulator output= For input 15 its adds 15 in each clock cycle ie. 15, 30, 45,60 etc.

This output given as an input to the Phase to amplitude converter as Phase Output with delay of one clock cycle. At Phase to amplitude converter, it takes cosine of phase output multiplied with $2^{14.}$ Here Cos (0)*2^{14}=16384 is the output. Then angles goes on increasing from 0 to 1 output goes on decreasing. And finally this output is given as final output with delay of one clock cycle.

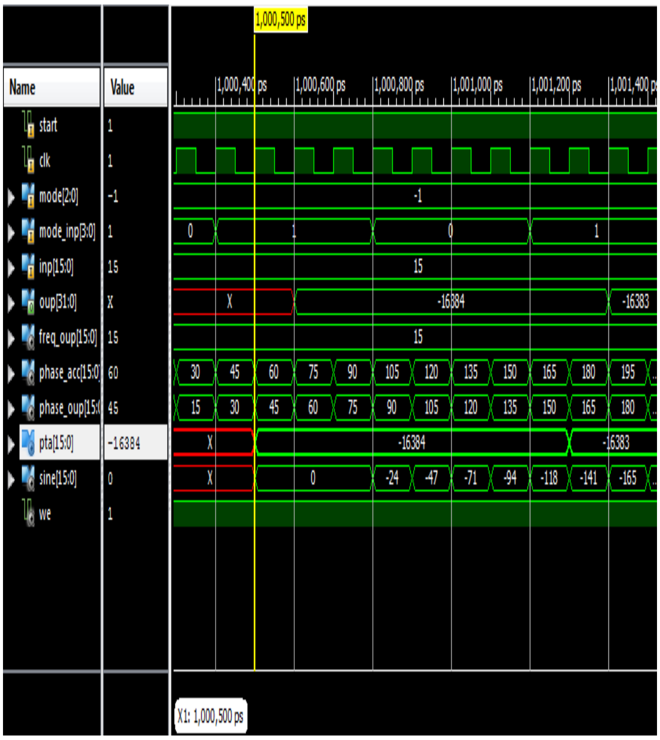


Fig 11 Normal Waveform Output in VHDL simulation

The design is interfaced with onboard LED display. The code was then successfully implemented on Spartan3E board after successful completion of translate, map, place and route processes. And the outputs can be viewed on LED display.

5. CONCLUSION

This paper shows the use of CORDIC algorithm for DDS. This algorithm is an useful technique for phase to sine amplitude conversion. CORDIC is implemented by a simple hardware through repeated shift-add operations. This paper reviews CORDIC architectures The proposed CORDIC design is based on Pipeline data path Architecture. This paper is focused on the Direct Digital Synthesizer using CORDIC approach, to increase the speed with minimum area requirement in FPGA. The features of CORDIC has made it an attractive choice for a wide variety of applications in communication system as in Direct Digital synthesizer; Analog and Digital modulation subsystems. Also this paper reveals that how to use the CORDIC algorithm to implement different communications systems like ASK, PSK, FSK, BPSK, QPSK, 4-QAM, 16-QAM etc digital modulators

REFERENCES

[1]. J. E. Volder, “The CORDIC trigonometric computing technique,” *IRE Trans. Electronic Computing*, volume EC-8, pp 330 – 334, 1959.
[2]. Pramod K. Meher, Javier Valls,Tso-Bing Juang, K. Sridharanand KoushikMaharatna “50 Years of CORDIC: Algorithms, Architectures, and Applications” *Ieee Transactions On Circuits And Systems—I*: Regular Papers, Vol. 56, No. 9, pp 1893-1907,September 2009

[3]. J. Valls, T. Sansaloni, A. Perez-Pascual, V. Torres, and V. Almenar, “The use of CORDIC in software defined radios: A tutorial,” *IEEE Commun. Mag.*, vol. 44, no. 9, 2006.

[4].Y.H. Hu, “Pipelined CORDIC architecture for the implementation of rotational based algorithm,” in *Proceedings of the International Symposium on VLSI Technology, Systems and Applications*, p. 259, May 1985.

[5]. J. E. Meggitt, “Pseudo division and pseudo multiplication processes,” *IBM Journal*, vol. 6, no. 2, pp. 210–226, 1962.

[6]. L. Cordesses, “Direct digital synthesis: A tool for periodic wave generation (part

[7]. J. Vankka *“Digital Synthesizers and Transmitters for Software Radio.*” Dordrecht, Netherlands: Springer, 2005.

[8]. J. Vankka, "Digital Modulator for Continuous Modulations with Slow Frequency Hopping," IEEE Transactions on Vehicular Technology, vol. 46, pp. 933-940, Nov. 1997.

[9]. M. Kosunen, J. Vankka, M. Waltari, L. Sumanen, K. Koli, and K. Halonen, "A CMOS Quadrature Baseband Frequency Synthesizer/Modulator", Analog Integrated Circuits and Signal Processing, Vol. 18, No. 1, pp. 55-67, Jan. 1999.

[10]. R.Andraka, “A survey of CORDIC algorithms for FPGA based computers, Proceedings of ACM/SIGDA sixth International Symposium on field Programmable Gate Arrays”, 1998,pp.191-200