International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 10 | Oct -2016 www.irjet.net p-ISSN: 2395-0072

Design and Implementation of Modified CORDIC based FFT using Vedic Multiplication Techniques

K Nagapramodh[1]and Hemanth Kumar K.S[2]

[1]*M.Tech Student, Dept. of E.C.E., UTL Technologies Ltd, Karnataka, India*
[2]*Assistant Professor, Dept. of E.C.E., UTL Technologies Ltd, Karnataka, India*

Abstract –*Time management is a very important issue in today's high speed day-to-day human life. Most essential thing in the time management is the uses of high speed devices which reduces the human effort and required time. The processor speed is the main requirements of any high speed devices. The speed is the main issue in any DSP processors due to the uses of large amounts of hardware resources. The main component cause this delay or frequency issue is the uses of conventional multipliers. In this paper, we proposed the implementation of FFT processor using modified CORDIC and Vedic multiplier to reduce the delay. The simulation and synthesis result shows that the proposed architecture can be able to process data faster than existing algorithms.*

Key Words: Fast Fourier Transform, Vedic Multiplication, Modified CORDIC algorithm,Butterfly diagram, FPGA Implementation.

1. INTRODUCTION

In today's high speed era the trends is all day-to-day life gadgets must operate at high speed and the area of the device is less. Reducing area also indirectly effects battery life of the corresponding devices. To overcome from this problem the hardware and software of the device is designed in compact way that it does not require large amount of resources and operating frequency is high.

Normally DSP systems uses large amount of resources and it's operating frequency also lower. This is because all the DSP calculations involve complex mathematics which will requires large amounts of hardware resources. Also all those DSP calculations consists of a large amount of multipliers. This is a major component of most DSP systems. Normally hardware multiplier architectures require large amount of resources [1] in-terms of basic gates. As a result the number of multipliers increase will directly effects on the maximum operating frequency of the device and increase the area requirements. So, in most DSP system reducing multiplier with other elements is a main key feature to increase frequency and reduce area. But it is not suitable for all scenarios occurred in DSP systems. As a result a large number of multiplier remains after fine tuning of the algorithm.

To overcome that problem removing of multiplier with equivalent elements does not work along. With this removing we must have to find some approach for implementing multiplier with reduced logic utilizations. The types of multipliers can be broadly divided into three parts [2] in electrical systems as Binary Multiplier, Analog Multiplier, Frequency Multiplier etc.

Since normally digital systems are used to implement all DSP processor so, we consider only digital multipliers. Digital multipliers are mainly divided into two categories [3] as Based on Sign and Based on Architecture.

Based on sign we can divide into two way as

I. *Signed Multiplier:* In this case both input and output can be either positive or negative binary numbers.

II. Unsigned Multiplier: In this case both input and output only positive binary numbers. It won't support negative binary values.

Based on on the verity of architectures we can classify multipliers as

I. Baugh-Wooly Multiplier

II. Wallace-Tree Multiplier

III. Dadda Multiplier

IV. Booth Multiplier

V. BKM Algorithm

VI. Modular Multiplication etc.

Also we can be able to change the performance of the multipliers by changing adder structures. Different types of adder structures are given below [3] as

I. RCA (Ripple Carry Adder)

II. CLA (Carry Look-ahead Adder)

III. CSA (Carry Save Adder)

IV. CScipa(Carry Skip Adder) etc.

2. LITERATURE SURVEYS

Prasant and Venkat [1] proposed design of Vedic multiplier compatible for FPGA Implementation. They used adder blocks as compression of partial products. By using this compressor block first they design 4x4 and 8x8 Vedic multiplier with reduced hardware and then 16x16 bits

multiplier is designed by using those predefined block. Parallel addition is used to increase the operating frequency of the design. To design efficient multiplier "Urdhva Tiryakbhyam" sutra is used. The proposed architecture is implemented on Spartan-3 FPGA board. Sreelekshmi et al., [2] uses Vedic multiplier to construct high speed MAC unit. To increase the speed of the MAC unit they replace normal multiplier by high speed Vedic multiplier. They design 32x32 bit multiplier using "Urdhva Tiryakbhyam" sutra. Also the normal adders structures are replaced by CSA (carry save adders). The proposed design is implemented using Xilinx ISE 10.1 and the coding is done by using HDL language. The structure produce 24.1 ns delay.Gaurav Sharma et al., [3] compare the delay introduced by different types of adders present in Vedic multiplier architecture. To implement 4x4 Vedic multiplier "Urdhva Tiryakbhyam" sutra is used. The conventional adder present in the architecture is replaced by CSA (carry save adder), CLS (carry lookahead adder) and RCA (ripple carry adder). Each architecture is implemented using VHDL language and the delay introduced by those architecture is measures on Xilinx Spartan-3 FPGA board. The result shows RCA present in Vedic multiplier shows minimum delay than others.Premananda B.S. et al., [4] proposed implementation of 8x8 Vedic multiplier. Here the Vedic mathematics is mainly used to generate partial products faster than conventional method. They propose carry skip addition techniques to increase the speed of multiplication. They implement 8-bit Vedic multiplier using 4x4 multiplier using both carry skip and ripple carry adder. The implementation is done in Xilinx Spartan-3 FPGA and the coding is done using HDL language. The proposed multiplier can be able to operate at higher frequency than existing.Pusphalata Verma [5] shows efficient way to implement Vedic multiplier (4x4) using Electronic Design Automation (EDA) tool. To implement 4x4 Vedic multiplier they used Xilinx 12.1i EDA tool and the coding is done by using VHDL language. They compared the proposed Vedic multiplier with conventional multiplier by synthesizing the proposed model in Spartan-3 FPGA board. Vedic multiplier can be able to operate much faster than conventional multiplier using less hardware resources.Poornima et al., [6] shows efficient hardware implementation of multiplier using Vedic mathematics. The proposed method used ripple carry adders (RCA) to generate partial products. They implement 8x8 multiplier using two 4x4 multipliers. The proposed design is implemented on Spartan-3 FPGA board and VHDL language is used to write coding part. The proposed multiplier produce 28.27 ns delay. Irine and Suchitra [7] propose Vedic multiplication based on floating point arithmetics. Due to this floating point calculations the accuracy of this multiplier increase in-terms of truncation errors. The proposed structure used ripple carry adders to generate intermediate products. The total multiplication procedure is divided into calculation unit and control unit. In calculation unit mantissa and exponent calculation is done. Along with this sign control also taken care by this unit. The architecture is implemented on Virtex-2 FPGA board.Keerthi and Manoj [8] proposed Vedic multiplier design using modified Gate Diffusion Input (GDI) technique. The proposed multiplier is designed using 45 nm technology. They use Cadence Virtuso EDA tool to implement the proposed models. They proposed two multiplier design for low power applications. The synthesis results of both designs shows decrements in power consumption of proposed circuits than conventional circuits.Sumit and Deepak [9] compares vital VLSI matrix corresponding to different types multipliers in-terms of area, power and performance. They compare array multiplier, Wallace-tree multiplier, Booths multiplier and Vedic multiplier in-terms of various hardware implementation parameters. They found the Vedic multiplier will give best performance of most of the hardware aspects than conventional multipliers. As a result Vedic multiplier can be used in the system to achieve low power.Virendra [10] proposed arithmetic operation using intelligent Vedic multiplier circuit. In this paper the author is trying to built some of the basic operation using the Vedic multiplier circuit used in DSP cases. The multiplier is designed by using VHDL language and synthesize the code for Spartan-3 FPGA board. The proposed multiplier shows 20.499 ns delay.Lakshmi et al., [11] proposed Vedic multiplication using the concept of reversible logic gates. Due to the uses of reversible logic gate the power consumption of the proposed Vedic multiplier is much less than the normal Vedic multiplier. They used BPPG and PG reversible gates to construct the reversible Vedic multiplier. To maintain speed normal adder is replace by ripple carry adder. The design is implemented on Xilinx Spartan-3 FPGA. The result shows major decrease in power consumption of proposed circuit than conventional circuit. Sheshavali and Niranjan [12] proposed new methods of implementing Vedic multiplier. They used memory based approach to construct Vedic multiplication. In the paper they design 16x16 multiplier using Vedic methods. The proposed structure is synthesized in Cyclotone-II board. The proposed architecture can be able to perform operation 1.5 times faster than normal architecture. Toni and Rotake [13] proposed fast adder based Vedic multiplier to increase the operating frequency. They use barrel shifter and fast adder to implement 8 bit Vedic multiplier. The propagation delay is 6.781 ns for proposed design.Nirmal and Ingole [14] use different types of available adder to decrease the power consumption of multiplier circuits. The normal adder is replaced by carry lookahead adder, koggy stone adder and ripple carry adder. Each architecture is implemented on FPGA. The koggy stone adder uses less power than other adders.Pranali et al., [15] proposed low power ALU based on Vedic multiplier. To reduce power required by multiplier present normal ALU the author is using Vedic multiplier. The code for 64 bit ALU is written using verilog.Elakkiya and Mathan [16] surveys the performances of Vedic multiplier with conventional multiplier. They shows that the multiplier designed using any of the corresponding sutras can be able to perform better than normal conventional multipliers.Suresh Kumar et al, [17] proposed hardware implementation of FFT architecture for the case of higher order. They implement 64 and 256 point FFT architecture. In the paper they

compare the performance of both radix-2 and radix-4 architecture. The proposed architecture is implemented on Spartan-3 board. The result shows reduction in hardware resources in the case of radix-4 structure.Aniket and Mayuresh [18] perform comparisons between various techniques used to implement FFT in hardware. For this comparisons they consider 256 and 1024 point FFT and to implement those in hardware they use radix-2, radix-4 and rader-brenner algorithm. VHDL language is used for coding and Xilinx tool is used for synthesis. The result shows small decrease in hardware requirements in the case of rader-brenner algorithm.Abhishek Gupta et al., [19] proposed FFT processor where normal multiplier is replaced with Vedic multiplier to increase the operating frequency and reduce area requirements. In the paper they modified the structure to achieve high frequency. The synthesis report prove that the operating frequency is more than existing.Subha Sri et al., [20] surveys different types of CORDIC implementation to understand advantages and disadvantages of each architectures. For this comparison they used folded structure, unfolded structure and parallel structure. The parallel structure shows high speed operation with the cost of area requirements.

3. BASIC THEORIES

In this section we describe all basic theory used to implement the total architecture.

3.1. Fast FourierTransforms

FFT is a form of Discrete Fourier Transform (DFT). The FFT reduce the time and computational complexity arises in DFT. This is a domain transform technique. It converts time domain information to frequency domain information and vice-versa. The equation of forward FFT is given below

$$X(K) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi Kn}{N}} \quad (1)$$

Since we implement all stage parallel so it is parallel FFT.

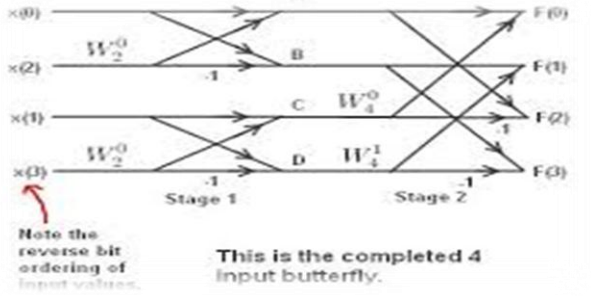

Fig -1: Butterfly diagram of 4 point FFT

3.2. Basic CORDIC Algorithm

The CORDIC (CO ordinate Rotation Digital Computer) [1] is based on the convention of 2D geometry. Jack E. Volder reported the first algorithm on the iterative reckoning functions of trigonometry, division and multiplication in 1959. The popularity of CORDIC was intensify there after primarily due to economical implementation of applications which includes the trigonometric, logarithmic, multiplication of complex number, inverse of matrix ,computation of Eigen-value, linear systems solution and singular value decomposition (SVD) for signal processing, image processing, and general scientific computation. Some other prominent and forthcoming applications are: 1) Robot manipulation: Direct and Inverse Kinematics figuring 2) Synthesis and communication of speech/music : Digital Modulation and coding , Synthesis of Direct Frequency 3) Graphics and Animation: Planar and 3D vector rotation. Although CORDIC technique may not be the fastest in performing these operations, it is attractive because of hardware implementation simplicity, as the same iterative algorithm should be make used in all these applications using the basic add and shift operations.This method is based on trial and error method for calculating the sine and cos value of corresponding input angles. From the algorithm we can write the rotation matrix R as

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (2)$$

$$R = (1 + tan^2\theta)^{-\frac{1}{2}} \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \quad (3)$$

We can rewrite the above equation as

$$R_c = \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \quad (4)$$

$$K = (1 + tan^2\theta)^{-\frac{1}{2}} \quad (5)$$

So, now

$$R_c(i) = K_i \begin{bmatrix} 1 & -\delta_i 2^{-i} \\ \delta_i 2^{-i} & 1 \end{bmatrix} \quad (6)$$

Where $k_i = \frac{1}{\sqrt{1+2^{-2i}}}$

By taking account all those equation we can write equation for CORDIC rotation mode

$$\begin{aligned} X_{i+1} &= X_i + \delta_i 2^{-i} Y_i \\ Y_{i+1} &= Y_i - \delta_i 2^{-i} X_i \\ \omega_{i+1} &= \omega_i - \delta_i \alpha_i \end{aligned} \quad (7)$$

The basic block diagram of CORDIC rotation mode is shown in Fig. 2

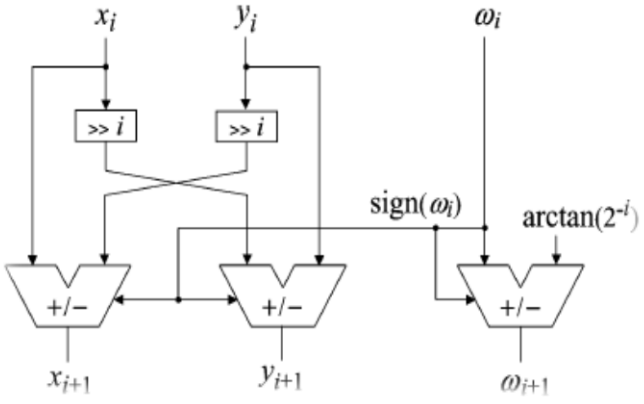

Fig -2: Hardware architecture of CORDIC algorithm

3.3. Vedic Multiplications

In Vedic mathematics there are two sutras, which are applicable for multiplication. They are Urdhva-Tiryakbyham and Nikhilam Navatascaramam Dasatah.

3.3.1. Urdhva-Tiryakbyham Sutra

This sutra can be able to perform all kinds of multiplication. In this case all partial products are generated and then the concurrent addition of these partial products can be done. This sutra is used to obtain parallelism of partial products and their summation. The algorithm can be generalized for n × n bit number. The algorithm calculates partial products and their sums in parallel manner. The same amount of time to calculate the product is needed by the multiplier and hence it is possible to maintain high clock frequency. The multiplication procedure of two decimal numbers 14 ×15 is given below. The digits on the both sides of the lines are multiplied and added with the carry generated from the previous step.

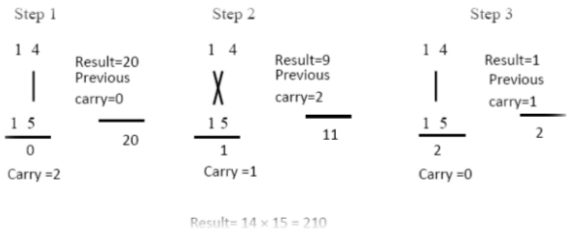

Fig -3: Multiplication of Decimal Numbers Using Urdhva Sutra

3.3.2. Nikhilam Navatascaramam Dasatah Sutra

The sutra basically means start from the leftmost digit and begun subtracting'9'from each of the digits; but subtract '10' from the last digit. To subtract 4679 from 10000, all the digits except the last digit are subtracted from 9 and the last digit is subtracted from 10, yielding 5321. This sutra is more efficient when the numbers involved are large.

One example of this sutra is given below

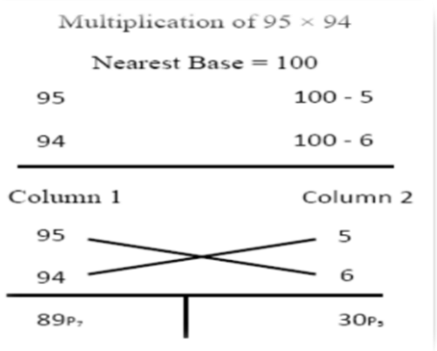

Fig -4: Multiplication of Decimal Numbers Using Nikhilam Sutra

4. PROPOSED ARCHITECTURE

The proposed CORDIC based FFT architecture is shown in Fig. 5 where proposed CORDIC based FFT block consists of Controller, Modified CORDIC Block and 4 point DIT-FFT. The controller is used to control the operation of total block by activating and deactivating each block at correct clock sequence. Each block is explained in brief below

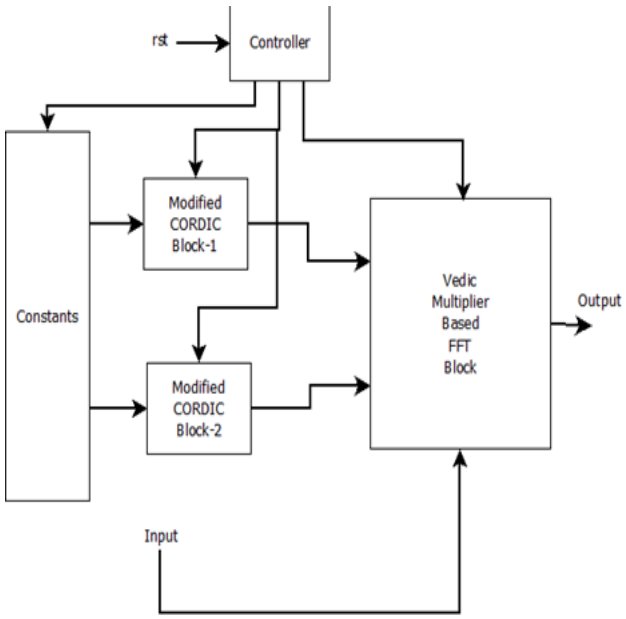

Fig -5: Proposed CORDIC based FFT

4.1. Modified CORDIC

The basic CORDIC equation consists of a large amount of constant multiplier. To increase the operating frequency and reduce area we replace the constant multiplication by corresponding shifts. So, the modified equation for CORDIC is given below

$$X_{i+1} = X_i + \delta_i(\gg i)Y_i$$
$$Y_{i+1} = Y_i - \delta_i(\gg i)X_i \quad (8)$$
$$\omega_{i+1} = \omega_i - \delta_i\alpha_i$$

Where, (>>i) is left shift by i^{th} position.

The proposed multiplier less CORDIC architecture is shown in the Fig 6. Also to increase parallelism in the structure we replace the ROM used to store predefined angle by fixed constants.

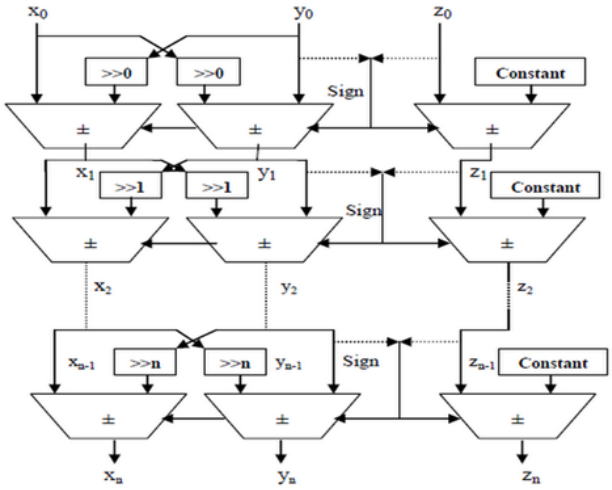

Fig -6: Internal Structure of Proposed CORDIC

Another problem in basic CORDIC is it can able to produce output from angle -99.9° to +99.9°, to overcome from this problem we use one preprocessing unit before the CORDIC block which is shown in Fig 7. Through this the unit can be able to process angle present in any one of the four quadrants.

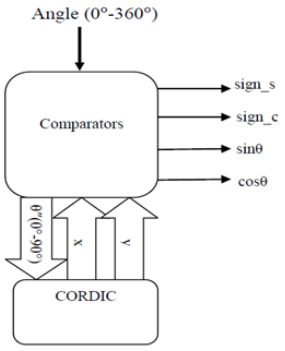

Fig -7: Modified CORDIC Architecture

4.2. Modified FFT

The radix-2 FFT algorithm is performed to generate N point DII-FFT. The twiddle factors are generated through CORDIC processor using N point samples as input. The Sinθ and Cosθ values of twiddle factors for variable angles are multiplied with an input to obtain N point FFT samples. The equation for FFT is given as

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi nk}{N}} = \sum_{n=0}^{N-1} x[n] W_N^{nK} \quad (9)$$

Let us consider θ=2πnk/N then

$$e^{-j\frac{2\pi nk}{N}} = e^{-j\theta} (10)$$

Decomposing exponential term as

$$e^{-j\theta} = \cos\theta - j\sin\theta \quad (11)$$

This Sinθ and Cosθ are generated by modified CORDIC algorithm. Now FFT equation is modified as

$$X[k] = \sum_{n=0}^{N-1} x[n]\cos\theta - j\sum_{n=0}^{N-1} x[n]\sin\theta \quad (12)$$

The FFT module implementation is done using equation where Sinθ and Cosθ are generated for predefined angles of N point FFT. The proposed FFT uses modified CORDIC to generate twiddle factors for predefined angles and observed that it avoids ROM based storage and Look Up Table method of storage. This results in high speed operation of FFT block with less hardware.

The multiplications present in FFT are replaced by Vedic multiplier to increase frequency.

5. FPGA IMPLEMENTATION

The proposed architecture is implemented on Xilinx ATLYS board (xc6slx45-2csg322) with the help of Simulink tools where the code is designed by VHDL language. The Simulink model of proposed technique is shown in Fig. 8

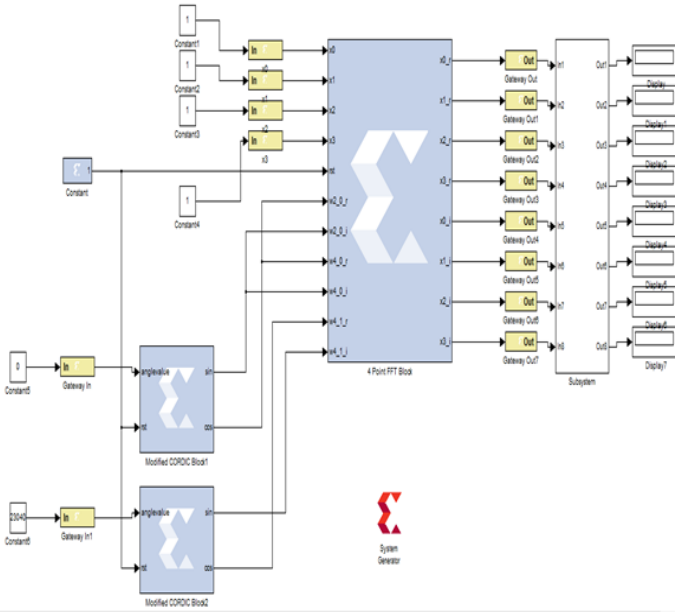

Fig -8: Simulink Model of CORDIC based FFT with Vedic Multiplier

The synthesis result of the proposed structure is given in Table 1

Table -1: Hardware Utilization of Proposed Architecture

Parameters	Hardware Utilization
Number of Slice Registers	259
Number of Slice LUTs	1325
Number of fully used LUT-FF pairs	164
Maximum Operating Frequency (MHz)	104.828

6. HARDWARE COMPARISONS

To check the performance of the propose architecture we compare the delay introduced by various conventional methods with proposed methods is shown in Table 2.

Table-2: Delay comparisons between existing and proposed method

Techniques	Authors	Delay (nsec)
Normal Array based FFT	----	32.01
Booth Multiplication based FFT	----	29.549
Modified Vedic Multiplier based FFT	Abhishek Gupta et al., [19]	19.467
Proposed Modified CORDIC based FFT with Vedic Multiplier	----	9.539

7. CONCLUSIONS

In this paper we propose parallel FFT architecture using modified CORDIC and Vedic multipliers. The proposed architecture is faster than existing architecture due to following reasons

i. The conventional multipliers are replaced by high speed MUX based Vedic multipliers.

ii. Parallel FFT architecture id used.

iii. Modified CORDIC architecture is used to generate twiddle factors instead of ROM based architecture.

REFERENCES

[1] Prashant D. Pawale, Venkat N Ghodke, "High speed Vedic multiplier design and implementation on FPGA", *International Journal of Applied Research*, Vol. 1, No. 7, pp. 239-244, 2015.

[2] Sreelekshmi M. S., Farsana F. J., Jithin Krishnan, Rajaram S and Aneesh R, "Implementation of MAC by using Modified Vedic Multiplier", *International Journal of Advanced Computer Research*, Vol. 3, No. 3, Issue. 12, pp. 11-15, 2013.

[3] Gaurav Sharma, Arjun Singh Chauhan, Himanshu Joshi and Satish Kumar Alaria, "Delay Comparison of 4 by 4 Vedic Multiplier based on Different Adder Architectures using VHDL", *International Journal of IT, Engineering and Applied Sciences Research*, Vol. 2, No. 6, pp. 28-32, 2013.

[4] Premananda B.S., Samarth S. Pai, Shashank B. and Shashank S. Bhat, "Design and Implementation of 8-Bit Vedic Multiplier", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 2, Issue. 12, pp. 5877-5882, 2013.

[5] Pushpalata Verma, "Design of 4x4 bit Vedic Multiplier using EDA Tool", International Journal of Computer Applications, Vol. 48, No.20, pp. 32-35, 2012.

[6] Poornima M, Shivaraj Kumar Patil, Shivukumar , Shridhar K P , Sanjay H, "Implementation of Multiplier using Vedic Algorithm", *International Journal of Innovative Technology and Exploring Engineering*, Vol. 2, Issue. 6, pp. 219-223, 2013.

[7] Irine Padma B.T and Suchitra. K, "Pipelined Floating Point Multiplier Based On Vedic Multiplication Technique", *International Conference on Innovations &Advances In Science, Engineering And Technology*, pp. 130-137, 2014.

[8] B.Keerthi Priya and R.Manoj Kumar, "A Novel Low Power Vedic Multiplier using Modified GDI Technique in 45nm Technology", *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 3, Issue 11, pp. 11721-11728, 2015.

[9] Sumit Vaidya and Deepak Dandekar, "Delay-Power Performance Comparison of Multipliers in VLSI Circuit Design", *International Journal of Computer Networks & Communications*, Vol.2, No.4, pp. 47-56, 2010.

[10] Virendra Babanrao Magar, "Intelligent and Superior Vedic Multiplier for FPGA Based Arithmetic Circuits", *International Journal of Soft Computing and Engineering*, Vol. 3, Issue. 3, pp. 31-36, 2013.

[11] P. Koti Lakshmi, B Santhosh Kumar and Rameshwar Rao, "Implementation of Vedic Multiplier using Reversible Gates", *International Journal of Applied Research*, pp. 125–134, 2015.

[12] C.Sheshavali and K.Niranjan kumar, "Design and Implementation of Vedic Multiplier", *International Journal of Engineering Research and Development*, Vol. 8, Issue. 6, pp. 23-28, 2013.

[13] Toni J.Billore and D.R.Rotake, "FPGA implementation of high speed 8 bit Vedic Multiplier using Fast adders", *IOSR Journal of VLSI and Signal Processing*, Vol. 4, Issue. 3, pp. 54-59, 2014.

[14] N.G.Nirmal and D.T.Ingole, "Novel Delay Efficient Approach for Vedic Multiplier with Generic Adder Module", *International Journal of Engineering Research and Applications*, Vol. 3, Issue 3, pp.1394-1396, 2013.

[15] Pranali Thakre1, Sanjay Dorle and Vipin Bhure, "Low Power 64-bit Multiplier Design by Vedic Mathematics", *International Journal of Application or Innovation in Engineering & Management*, Vol. 3, Issue. 4, pp. 393-396, 2014.

[16] Elakkiya J and Mathan N, "Survey on Performance of Vedic Multiplier", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, Issue 2, pp. 416-419, 2015.

[17] Suresh Kumar Dunna, B Vijaya Bhaskar and R Suryaprakash, "Implementation of Higher Order FFT Processor Using FPGA", *International Journal of Engineering, Reserch and Technology*, Vol. 1, Issue. 6, pp. 1-4, 2012.

[18] Aniket Shukla and Mayuresh Deshmukh, "Comparative Study Of Various FFT Algorithm Implementation On FPGA", International Journal of Engineering, Technology and Science, Vol. 1, Issue. 1, pp. 19-22, 2012.

[19] Abhishek Gupta, Amit Jain, Anand Vardhan Bhalla and Utsav Malviya, "Design Of High Speed FFT Processor Using Vedic Multiplication Technique", *International Journal of Engineering, Reserch and Applications*, Vol. 2, Issue 5, pp.1501-1504, 2012.

[20] T.Subha Sri , K.Seshu Kumar and R. Muttaiah, "Review of CORDIC Architectures", *International Journal of Engineering and Technology*, Vol. 5, No. 2, pp. 578-585, 2013.