FAST FOURIER TRANSFORM BY VERILOG

Conference Paper · November 2020

CITATIONS
0

READS
11,180

2 authors:

Hina Magsi
Sukkur IBA University
21 PUBLICATIONS **237** CITATIONS
SEE PROFILE

Ali Hassan Sodhro
Kristianstad University
101 PUBLICATIONS **3,480** CITATIONS
SEE PROFILE

FAST FOURIER TRANSFORM BY VERILOG

Hina Magsi
Department Of Electrical Engineering
Sukkur Institute of Business Administration, University
Sukkur, Pakistan
hina.mece17@iba-suk.edu.pk

Ali Hassan Sodhro
Department Of Electrical Engineering
Sukkur Institute of Business Administration, University
Sukkur, Pakistan and DISP LAB, University Lumiere
Lyon2, Lyon, France
ali.hassan@iba-suk.edu.pk, alihassan.sodrho@uni-lyon2.fr

***Abstract*—The paper presents the Verilog coding of Fast Fourier transform implementation on Vivado. The butterfly diagram used to design the Fast Fourier transform of given input signals. The FFT is useful and efficient tool in digital signal processing especially in the area of digital signal and image processing. It has also been used in new technologies like internet of medical things for extracting desired information from signal. The Verilog implementation of FFT which includes complex numbers addition and multiplication due to twiddle factor. It is an efficient and fast method which requires less computations and produces output in short time.**

Keywords— fast fourrier trasnsform(FFT) ,dicrete fourier transform(DFT),twiddle factor.

I. INTRODUCTION

The main purpose in digital signal processing is to reduce the requirements of hardware and increased the computation. The Fast Fourier transform (FFT) is an algorithm which samples the signal over some space and splits them into frequency points. This method is used to get the discrete Fourier transform of the signal and it converts the signal into its frequency domain. It has advantage over DFT because it reduces the complexity of DFT from O(N^2) to O(NlogN) where N represents the data points. The fast method to compute the DFT (Discrete Fourier transform) is the FFT (fast Fourier transform). It has many applications like spectral analysis, matched filtering, image processing and disease extractions. When FFT was developed the real time digital signal become realized in many fields such as radar and communications. There are many situations where FFT computation is required to face the real-time data like medical diagnosis and earthquake monitoring and so on [17]. The detailed information about spectrum of signal in frequencies

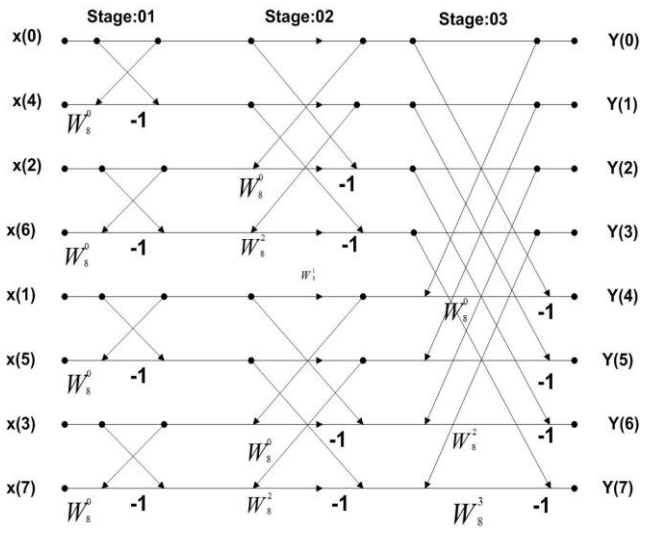

Fig. 1 Butterfly Representation [9]

and amplitude would be difficult to analyze in time domain [19]. Due to this reason the signal would be converted into frequency domain to get the complete information. In IoMT (Internet of Medical Things) FFT plays an important role. As EEG and ECG are highly noisy and difficult for processing. That's why extraction information form this signal is the main issue. To extract the detailed and meaningful information from noisy signal in order to get the best analytics FFT is used. It converts the noisy data into desired information and reduces the noise level. It provides the detailed information in frequency domain. The frequency domain representation of signals educes the computational complexity of various digital processing systems. In the field of ultrasonic echo signals FFT increased the computational speed and accuracy which makes the system performance better. The FFT is the center part in Doppler blood flow spectrum analysis and also used in Doppler imaging system [1]. The results of DFT and FFT are same only difference is the reduction in the calculations. FFT

Input	Address (binary)	Output (bit reversed)	Address (binary)
x(0)	000	x(0)	000
x(1)	001	x(4)	100
x(2)	010	x(2)	010
x(3)	011	x(6)	110
x(4)	100	x(1)	001
x(5)	101	x(5)	101
x(6)	110	x(3)	011
x(7)	111	x(7)	111

Fig: 2 Bit Reversal [11]

divides the data into smaller sets (even and odd) [12]. There are two types of division known division in time and division in frequency. The division in time requires bit reversed output and produces normal output order while division is frequency takes normal input order and generates bit reversed output [14]. This paper uses the division in time method. Each stages, the results of previous stage are combined. The Fig. 1 shows the butterfly diagram of 8-point FFT [9]. The butterfly is part of the FFT computation which combines smaller DFT point into larger points. The name butterfly implies that the shape of the flow of data as shown in Fig. 1. In every stage first inputs are multiplied by twiddle factor and add or subtract to other input. The points defines the stages of butterfly like 8-point have 3 stages, 16-point have 4 stages and so on. In FFT

calculation there is concept of bit reversal the inputs are to be bit reversed to produce the outputs bits in the normal order. The input bits are divided into even and odd parts and these parts are divided into even and odd parts. This process is continued till only two points left [11]. In the paper the input sequence is (x0,x1,x2,x3,x4,x5,x6,x7) and after bit reversed the sequence becomes (x0,x4,x2,x4,x1,x5,x3,x7) . This is shown in Fig. 2[11].

The main contribution of this paper is to design FFT algorithm using radix-2 Verilog code, implemented and verified on vivado. The block diagram implemented on Vivado is shown in Fig.3.

Remaining part of this paper is organized as follow. Section II discussed the literature review, Section III explained the proposed algorithm on Vivado, results are shown in Section IV and finally the paper is concluded in Section V.

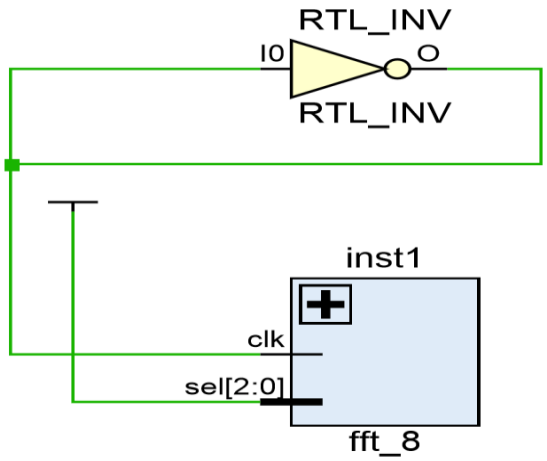

Fig: 3 RTL diagram

II. LITERATURE REVIEW

There are many researchers who have worked on FFT algorithm and implement the different methods. Some of the work are discussed below.

Zhou et al proposed the FFT processor for high speed and real time signal processing [1]. They present FFT design on field programmable gate array (FPGA) by using radix-2 algorithm and also pipeline structure for butterfly algorithm and Ping-Pone for memory unit. They compared the theoretical and practical values in their paper. This structure is very easy and expandable by changing the twiddle factor and some changes in timings of ping-pone operations.

Sridhanya et al presents the [2] application of FFT in MIMO-OFDM (multiple input multiple output orthogonal frequency division multiplexing) systems. They used radix Ns (number of data streams) butterflies for every stage. Memory scheduling and multipath delay commutator is used as hardware implementation which reduced the storage and provides efficient saving for power. They replaced the twiddle factor with complex multiplier and presents the advantages in terms of power consumption.

Song Yu et al proposed [3] the FFT algorithm by using rotation factor and data address of the node. They uses Verilog HDL platform for design and realization of data address and also compilation is done on PLD software. They concludes that FFT with FPGA is easy for expansion and it has real-time data capability.

Athira et al introduced the FFT application in telemetry data processing applications [4]. They uses vedic mathematics for multiplications. The Urdhva Tiryakbhyam sutra method is used because it reduces the computation complexity of multipliers. They proposed the 24 bit floating point implementation using IEEE754 multiplication on vedic mathematics and compare the results with conventional method. The carry adder or ripple carry adder improved the performance of addition part. The design was done on Verilog while simulations stage was carried out using Modelsim.

Zakir et al presents the Q format constant multipliers for FFT processors with implementation on hardware improves the speed of the multipliers [5]. The common sub expression elimination (CSE) method and canonical signed Digit (CSD) is used to reduce number of adder which also reduces hardware specifications. They presented the design for 8bits, 10 bits and 16bits using Verilog and implement this system on Altera device.

Ibrahim at al proposed the radix-2 FFT algorithm to be implemented and investigated on FPGA and Graphic Processing Units (GPU). [6] The Verilog HDL is used for FPGA and Open Computing Language is used for GPU. They compares both results and concludes that FFT of small sizes produces faster result on FPGA while GPU can used for larger sizes of FFT. Also FPGA implementations is faster than built-in IP core Xilinx.

Archna et al presents the [7] importance of Hilbert algorithm in digital signal processing like modulation, frequency analysis and audio production. They proposed the realization of this algorithm by radix 2^2 single path delay feedback (SDF) pipelined FFT processor which can be used in envelope detection. The system has implemented on Xilinx Verilog code.

Ravi et al proposed the bit-reversal part of FFT. The proposed system is simple and efficient for reordering of parallel data formation [8]. They used the parallel pipelined FFT processors. The system has been implemented using Verilog HDL and simulated on Modelsim.

Anup et al compares the [9] DFT and FFT in the field of digital signal processing. They said that FFT is efficient method to compute DFT with larger points. They implemented FFT algorithm on Verilog by using floating point. They concluded that the FFT reduced the cost and complexity of adders and multipliers.

Miao et al described the powerful tool FFT used in signal processing [10]. The FFT method reduces the computational complexity from O(n^2) to O(NlogN). It is important tool for real time data and used in many applications like IoMT. The dot product engine (DPE) method is used for computing FFT

and compared the results with real multiplication operation. The DPE cluster can be used for many applications like IoMT.

Vinodh et al proposed the implementation of FFT pruning algorithm on FPGA [13]. The FFT reduced the time for computation when zero valued number outstrips the non-zero valued due to excessive computations. This is achieved by pruning algorithm of FFT and is implemented in hardware. The computational time consumption is observed. The FFT implementation on FPGA would reduce the computation time and easily calculates the DFT [20].

Atin et al [15] proposed the area efficient architecture of radix-2 FFT processor. The algorithm reused the units of single stages of butterfly units which reduces the area. The simulation has been taken on VHDL. The system has been implemented and verified on FPGA. The FFT processor performance can be increased b recording the size and improving SNR of the system [18]. The proposed system provides the better results than conventional system of FFT processor.

Josue et al explained the 16 and 32 bit vertex 6 FFT algorithm on VHDL [16]. The design proposed had used short area and increase of input-output block can be reduced by using parallel in serial out and serial in parallel out shift registers.

$$Y(k) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi kn/N}$$

where x(n) represents the inputs and exponential part is known as twiddle factor(W_n^k) which includes complex numbers. N is the number of data points. The paper proposed implementation of DIT radix-2 algorithm Fast Fourier Transform using Verilog on VIVADO software. The design includes the implementation of DIT radix-2 8-point FFT butterfly algorithm. The butterfly comprises of three stages. Every stage is connected with each other to get final output. It takes input sequence and twiddle factor and produces the final desired output. The twiddle factor is calculated manually and initialize in Vivado. The design includes adders and multipliers of complex numbers. The Fig. 4 illustrated the FFT design radix-2. The RTL diagram for radix-2 FFT algorithm is shown in Fig. 6.The adder and multiplier of radix-2 FFT is shown in Fig.5. It takes 8 bit inputs as (x0,x1,x2,x3,x4,x5,x6,x7), initialize twiddle factor as real (wr) and imaginary part (wi) and produces the final output after three stages as 8 bit real and imaginary part($y0$, $y1$,y2,y3,y4,y5,y6,y7). The initial input values are x0=1, x1=2, x2=4, x3=8, x5=16, x7=64, x7=128, and the twiddle factors are calculated below and manual calculations are shown in Table. I.

TABLE I. Manual Calculations

Calculations		
Stage 1 output	***Stage 2 output***	***Stage 3 output***
$x10r = x(0) + \omega_8^0\, x(4)$=1+ (1)16=17.	$x20r = x(10r) + \omega_8^0\, x(12r)$=17+ (1)68=85.	$y0r = x(20r) + \omega_8^0\, x(24r)$=85+ (1)170=255.
$x11r = x(0) - \omega_8^0\, x(4)$=1-(1)16=15.	$x21r + x21i = x(11r) + \omega_8^2\, x(13r)$=-15+ (-j)-60=-15+j60.	$y1r + y1i = [x(21r) + x(21i)] + \omega_8^1 [x(25r) + x(25i)]$ = [-15+j60] + (0.707-j0.707) (-30+j120) = 48+j165.
$x12r = x(2) + \omega_8^0\, x(6)$=4+ (1)64=68.	$x22r = x(10r) - \omega_8^0\, x(12r)$=17- (1)68=-51.	$y2r = x(22r) + \omega_8^2\, x(26r)$=-51+ (-j) (-102) =-51+j102.
$x13r = x(2) - \omega_8^0\, x(6)$=4-(1)64=-60.	$x23r + x23i = x(11r) - \omega_8^2\, x(13r)$=-15-(-j)-60=-15-j60.	$y3r + y3i = [x(23r) + x(23i)] + \omega_8^3 [x(27r) + x(27i)]$ = [-15-j60] + (-0.707-j0.707) (-30-j120) =-78+j45.
$x14r = x(1) + \omega_8^0\, x(5)$=2+ (1)32=34	$x24r = x(14r) + \omega_8^0\, x(16r)$=34+ (1)136=170.	$y4r = x(20r) - \omega_8^0\, x(24r)$=85-(1)170=-85.
$x15r = x(1) - \omega_8^0\, x(5)$=2-(1)32=-30.	$x25r + x25i = x(15r) + \omega_8^2\, x(17r)$=-30+ (-j)-120=-30+j120.	$y(5)r = [x(21r) + x(21i)] - \omega_8^1 [x(25r) + x(25i)]$=]= [-15+j60]-(0.707-j0.707) (-30+j120) =-78-j45.
$x16r = x(3) + \omega_8^0\, x(7)$=8+(1)128=136.	$x26r = x(14r) - \omega_8^0\, x(16r)$=34-(1)136=-102.	$y6r = x(22r) - \omega_8^2\, x(26r)$==-51-(-j) (-102) =-51-j102.
$x17r = x(3) - \omega_8^0\, x(7)$=8-(1)128=-120.	$x27r + x27i = x(15r) - \omega_8^2\, x(17r)$=-30-(-j)-120=-30-j120.	$y7r + y7i == [x(23r) + x(23i)] - \omega_8^3 [x(27r) + x(27i)]$ = [-15-j60] + (-0.707-j0.707) (-30-j120) =48-j165.

III. PROPOSED FFT ALGORITHM

The mathematical representation of Fast Fourier transform [9] is given as:

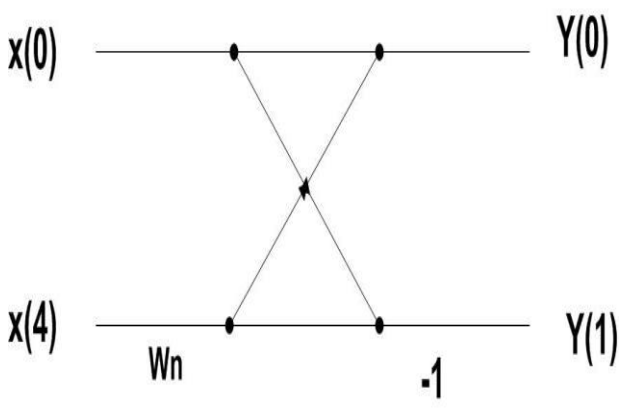

Fig. 4 DIT Radix-2 FFT

$W_n^k = e^{-j2\pi k/N}$ Where N=8

$$W_8^0 = e^{-j2\pi(0)/8} = e^0 = 1$$

$$W_8^1 = e^{-j2\pi(1)/8} = e^{-j\pi/4} = \frac{1-j}{\sqrt{2}}$$

$$W_8^2 = e^{-j2\pi(2)/8} = e^{-j\pi/2} = -j$$

$$W_8^3 = e^{-j2\pi(3)/8} = e^{-j3\pi/4} = \frac{-(1+j)}{\sqrt{2}}$$

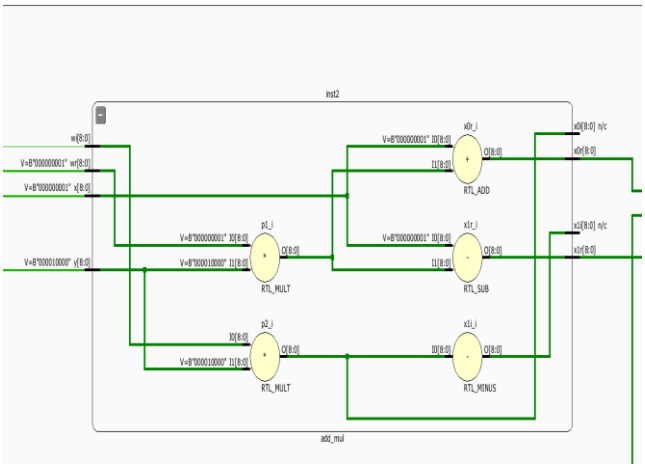

Fig. 5 Adder & Multiplier

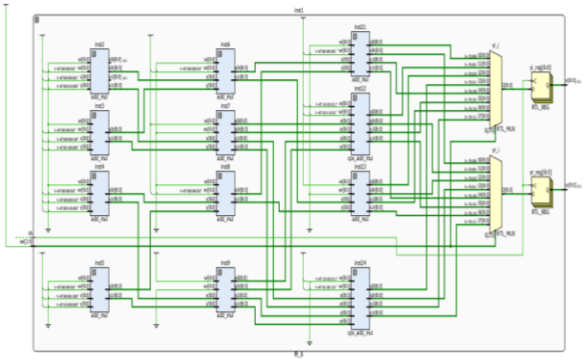

Fig. 6 RTL diagram of DIT radix-2 FFT

IV. RESULTS

TABLE III. Stagewise Output

//first stage outputs	//second stage outputs	//third stage outputs
sel=0, outputs:yr= 17 ,yi= 0 sel=1, outputs:yr= -15 ,yi= 0 sel=2, outputs:yr= 68 ,yi= 0 sel=3, outputs:yr= -60 ,yi= 0 sel=4, outputs:yr= 34 ,yi= 0 sel=5, outputs:yr= -30 ,yi= 0 sel=6, outputs:yr= 136 ,yi= 0 sel=7, outputs:yr=-120 ,yi= 0	sel=0,outputs:yr= 85 , yi= 0 sel=1,outputs:yr= -15 , yi=60 sel=2,outputs:yr= -51 , yi= 0 sel=3,outputs:yr=-15,yi= -60 sel=4,outputs:yr= 170,yi= 0 sel=5,outputs:yr= -30,yi= 120 sel=6,outputs:yr=-102 , yi= 0 sel=7,outputs:yr=-30,yi=-120	sel=0,outputs:yr= 255 , yi= 0 sel=1,outputs:yr=48 ,yi= 165 sel=2,outputs:yr= -51,yi= 102 sel=3, outputs:yr= -78,yi= 45 sel=4, outputs:yr= -85 ,yi= 0 sel=5, outputs:yr= -78,yi= -45 sel=6,outputs:yr= -51,yi=-102 sel=7,outputs:yr= 48,yi=-165

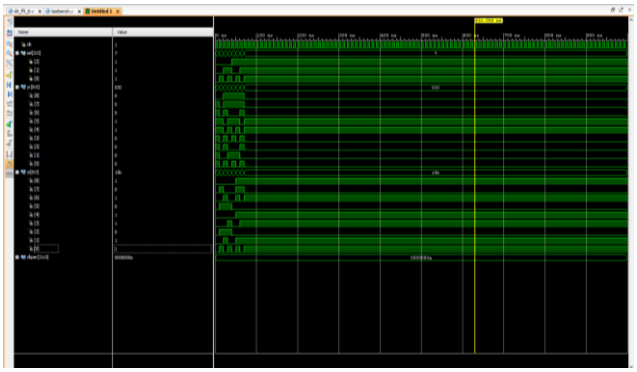

Fig. 7 Vivado Result

The Fig. 7 shows the waveform result and each stage output is shown in Table. II.The waveform have been shown which includes real and imaginary output.

V. CONCLUSION

FFT has the benefit over DFT because it has less computations. FFT takes inputs and through adding and multiplication process produces the output. While DFT has to make O (N^2) addition and multiplication for higher number of points. Implemented butterflies algorithm on VIVADO reduces calculations and makes system more efficient and fast. The FFT algorithm is used information extracting for ECG, EEG signals. It is also powerful tool for new technology IoMT. The future work will to reduce the complexity of multiplications in FFT to make system more efficient.

ACKNOWLEDGMENT

This work is funded by HEC Pakistan under the START-UP RESEARCH GRANT PROGRAM (SRGP) #21-1465/SRGP/R&D/HEC/2016, and Sukkur IBA University, Sukkur, Sindh, Pakistan.

REFERENCES

[1] Sheng Zhou, Xiaochun Wang, Jianjun Ji, and Yanqun Wang,"Design and Implementation of a 1024-point High-speed FFT Processor Based on the FPGA ", 6th International Congress on Image and Signal Processing (CISP), 2013.

[2] M. Sridhanya, and Mrs. G. Annapurna," Efficient design of FFT/IFFT processor using Verilog HDL", International Journal of Professional Engineering Studies (IJPRES), JULY 2015.

[3] Song Yu, Lin Xingye,and Zhai Shuang," Design and realization of FFT implementation unit based on FPGA ", Third International Conference on Multimedia Information Networking and Security, 2011.

[4] Athira Menon M S,and Renjith R J,"Implementation of 24 bit high speed floating point vedic multiplier", International Conference on Networks & Advances in Computational Technologies(NetACT), 20-22 July 2017.

[5] Md.Zakir Hussain, Kazi Nikhat Parvin ,and Zeba Fatima Mir Ilyas Ali," Q-Point Constant Multipliers for FFT Processors", International conference on Signal Processing, Communication, Power and Embedded System (SCOPES),2016.

[6] Muhammad Ibrahim, and Omar Khan, "Performance Analysis of Fast Fourier Transform on Field Programmable Gate Arrays and Graphic Card", 2016.

[7] Archna Rani1, Ram Mohan Verma,and Saurabh Jaiswal," FPGA implementation of Hilbert Transform via Radix-2 Pipelined FFT Processor",4th ICCCNT, July 4-6, 2013, Tiruchengode, India.

[8] Ravi L.S, Chithra .C .P ,and Farzana Parveen B.C," Design and Implementation of Parallel Bit Reversal on FFT by using Verilog HDL", Vol.7,Issue No.8,International Journal of Engineering Science and Computing, August 2017.

[9] Anup Tiwari,and Samir Pandey, "Implementation of Fast Fourier Transform in Verilog" International Journal of Engineering and Management Research, Vol.6, Issue-6, November-December 2016.

[10] Miao Hu, and John Paul Strachan, "Accelerating Discrete Fourier Transforms with Dot-product Engine",2016.

[11] Rubio, M., Gómez, P. and Drouiche, K.,"A new superfast bit reversal Algorithm", International Journal of Adaptive Control and Signal Processing, pp.703-707, 2002.

[12] Mehrotra, Mitul, Geetika Pandey, and Mandeep Singh Narula. "Implementation of FFT algorithm", International Journal of Engineering Sciences & Research Technology, 10 May 2017.

[13] Ch. Vinodh Kumar,and K.R.K Satry,"Design and Implementation of FFT Pruning Algorithm on FPGA", 7th International Conference on Cloud Computing, Data Science & Engineering, 2017.

[14] Debalina Ghosh , Depanwita Debnath , and Dr. Amlan Chakrabarti,"FPGA Based Implementation of FFT Processor Using Different Architectures",IJAITI,2012.

[15] Atin Mukherjee, Amitabha Sinha and Debesh Choudhury, "A Novel Architecture of Area Efficient FFT Algorithm for FPGA Implementation", 25 Feb 2015.

[16] Josue Saenz S., Juan J. Raygoza P., Edwin C. Becerra A. ,Susana Ortega Cisneros, and Jorge Rivera Dominguez," FPGA Design and Implementation of Radix-2 Fast Fourier Transform Algorithm with 16 and 32 Points", ROPEC,2015.

[17] J. G. Proakis, et al., "Digital Signal Processing", 4th ed. U.S.A.: Prentice-Hall, Inc., (2006).

[18] Aimei Tang, Li Yu, Fangjian Han, and Zhiqiang Zhang, "CORDIC-based FFT Real-time Processing Design and FPGA Implementation", 12th International Colloquium on Signal Processing & its Applications (CSPA2016), 4 - 6 March 2016, Melaka, Malaysia.

[19] Wu, R. and T. Tsao, 2003. "The Optimization of Spectrum Analysis for Digital Signals" IEEE Transactions on Power Delivery, 18(2): 398-405.

[20] Néstor Fernando Hortúa Díaz, Yuli Alexandra Velásquez Pulido, Dario Amaya Hurtado, "Calculation of The Fft Using The Radix 2 Algorithm In A Fpga Cyclone IV", 2016. Journal of Applied Sciences Research. 12(12); Pages: 46-53.