ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Visit: www.ijirset.com*

**Vol. 8, Issue 11, November 2019**

# CMOS Implementation of Low Power High Performance Fast Fourier Transform

Priyanka Dhayal, Yashika Saini

Research Scholar M.Tech. (VLSI ), RIET, Jaipur, India

Assistant Professor, RIET, Jaipur, India

**ABSTRACT:** The implementation of high speed and low power consuming designs are of prime concern in current scenario. Low power devices are widely used in many signal processing systems and communication applications. In this thesis high performance energy efficient Fast Fourier Transform design is implemented with less power consumption and reduced delay . FFT algorithms are prime models in the design of processing signals. These are widely applied to various WLAN, image processing application, radar and multimedia communication services and spectrum measurements. In this work the design of Decimation in Time-Fast Fourier Transform is described with required improvements in the design as well as in modules used for computation in order to obtain needed low power and low delay results.

## I. INTRODUCTION

1 For digital signal processing systems, the Discrete Fourier Transform (DFT) is the widely used algorithm. These are not calculated directly, but instead are computed with the Fast Fourier Transform (FFT). The computation of N- sample input of this algorithm carries a large number of operations i.e. $N^2$ complex multiplications and N (N−1) complex additions. Since the DFT is based on computation technique, many changes have been proposed for implementing it efficiently and rapidly. It has been continuously applied in Ultra Wide Band (UWB), Radars, receivers and image processing system [3].

2 Thus a fast algorithm has been made known by Cooley- Turkey, namely Fast Fourier Transform (FFT) by decreasing the number of computing operations. Hence for fast computation of the Discrete Fourier Transform (DFT), the Fast Fourier Transform (FFT) is an efficient and widely accepted technique [1]. This algorithm can be applied in various systems such as fast convolution and correlation, spectrum estimation, signal modulation, etc. With the arrival of semiconductor technologies in VLSI system, FFT design realization need to rise steadily [5].

## II. FFT DESCRIPTION

This can be achieved by reducing the computation complexity of the FFT design. Various modules have been put into effect at different stages for reducing the complexity of the FFT algorithm. In current scenario fast arithmetic computation modules such as adders and multipliers in the VLSI designs is the basic requirement.

Also in order to increase the computation speed a ROM-free twiddle factor generator has been designed using simple shifters and adders only [8], hence the need to save all the twiddle factors in a large ROM space is removed. Thus a constant complex multiplier ($W_N$-multiplier) for calculating twiddle factor is proposed. Also it is advantageous to use arithmetic circuits for small ranges of the twiddle factor multipliers rather than general multipliers [1].

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Visit: www.ijirset.com*

**Vol. 8, Issue 11, November 2019**

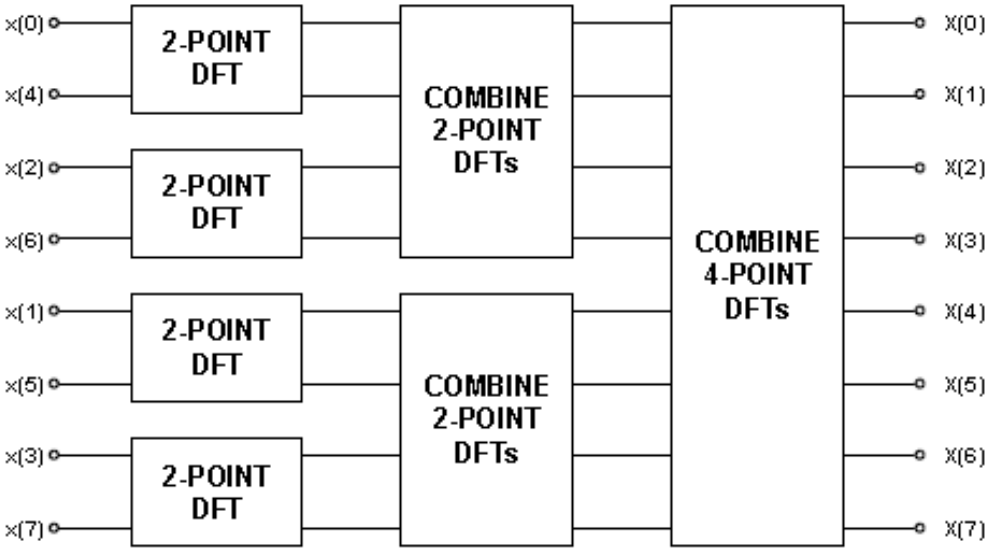


Figure 1

Computation of an 8-Point DFT in Three Stages using Decimation-In-Time .

## III. OBJECTIVE

The main motive of this research is VLSI implementation of FFT architectures, and circuits with the high-performance and energy-efficiency. This includes low power dissipation with high throughput and less propagation delay of the circuit. The main goal of the thesis is as follows:

1. FFT Implementation in 180NM technology.
2. Design of constant Complex multiplier for generation twiddle factor and using Vedic Multiplier for multiplying purpose, which shows various advantages over general multipliers.
3. To design 2 point, 4 point , 8 point FFT butterfly architecture and comparing with already proposed designs with same 180NM technology

Thus with these results and we have tried to make high performance FFT structure with less complexity.

## IV. TWIDDLE FACTOR

This is the factor by which the input 2-points are multiplied in the butterfly. The value of N and k depends on the stage of the algorithm as shown in Figure 3.1. They are either stored or calculated on the fly. Some twiddle factors requires no multiplication, but just negation or multiplication by j, like , , $W_N^0$, $W_N^{N/2}$, $W_N^{N/4}$.

Here decimation-in-time algorithm was chosen to compute the FFT. Although the number of multiplications required for an FFT algorithm by no means gives a complete picture of its complexity, it does give a reasonable first approximation. Although the number of multiplications decreases monotonically with increasing radix, the number of additions reaches a minimum and then increases. Further reductions in computational complexity are possible by simplifying trivial multiplications by ±1 or ±j. With questionable gains, multiplications by π/4, 3π/4, 5π/4, and 7π/4 can also be modified to require fewer actual multiplications [13].

Butterfly is a part of the computation that tie up the results of smaller Fourier transforms into a larger FT, or splitting a larger FT into sub transforms. The name "butterfly" appears from the shape of the data-flow diagram. Usually, the term "butterfly" appears with reference to the FFT algorithm, which continuously divides a DFT of composite size n = rm

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 11, November 2019

into r smaller transforms of size m where r is the "radix" of the transform [3]. These smaller DFTs are then combined via size-r butterflies, which themselves are DFTs of size r (performed m times on corresponding outputs of the sub-transforms) pre-multiplied by roots of unity (known as twiddle factors). This is the "decimation in time (DIT)".

**4.1 Implementation of 2 Point FFT**

In 2 point FFT algorithm, butterfly is basic block which is shown in Fig. 5.1.

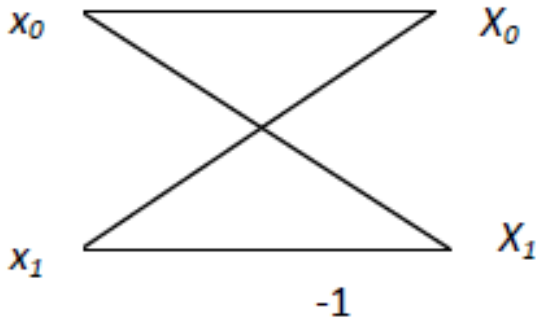


Figure 2: 2-Point Butterfly FFT.

Butterfly is basically a DFT of size-2 that consists of two inputs ($x_0$, $x_1$) and gives two outputs ($X_0$, $X_1$) by the formula (not including twiddle factors):

$A_0 = B_0 + B_1$

$A_1 = B_0 - B_1$

On implementing data-flow diagram for this pair of operations, the ($x_0$, $x_1$) to ($X_0$, $X_1$) lines cross and shows the wings of a butterfly, hence specifies the name. A decimation-in-time FFT algorithm on n = 2 p inputs with respect to a primitive $n^{th}$ root of unity $W_n^k$ rely on O (n log n) butterflies of the form [1]:

$x_0 = X_0 + X_1 W_n^k$

$x_1 = X_0 - X_1 W_n^k$

where k is an integer depending on the part of the transform being computed. In general N-point DFT of a sequence x(n) is given by eq. 3.10

The radix-2 algorithms [3] are the easiest FFT algorithms. The 2 point decimation-in-time (DIT) FFT divides a DFT into two equal length DFTs of the even-term and odd-term time samples. Hence the total computational cost is reduced by reusing the outputs of these shorter FFTs to compute many outputs. The decimation-in-time and decimation-in-frequency fast Fourier transforms (FFTs) are the basic FFT algorithms. The schematic shows that one carry look ahead adder and one subtractor unit is used for calculation. Four bit input is applied to the design. The calculation is done as follows:

$S_n = a_n + b_n$

$di_n = a_n - b_n$

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Visit: www.ijirset.com*

**Vol. 8, Issue 11, November 2019**

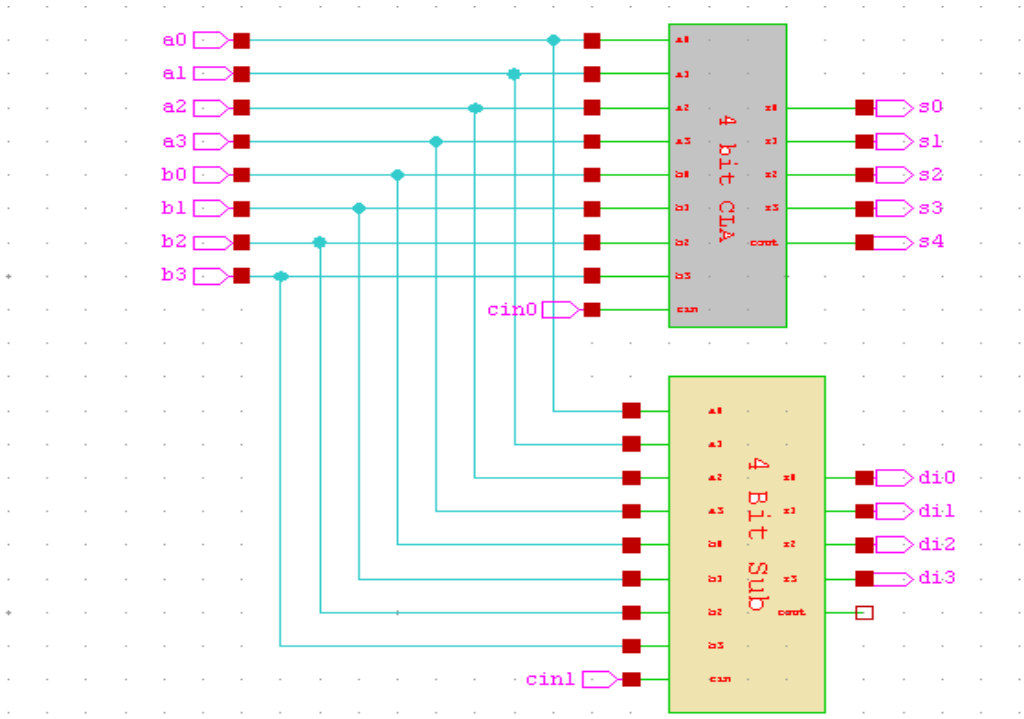


Figure 3(a): Schematic Diagram of 2- Point butterfly FFT.

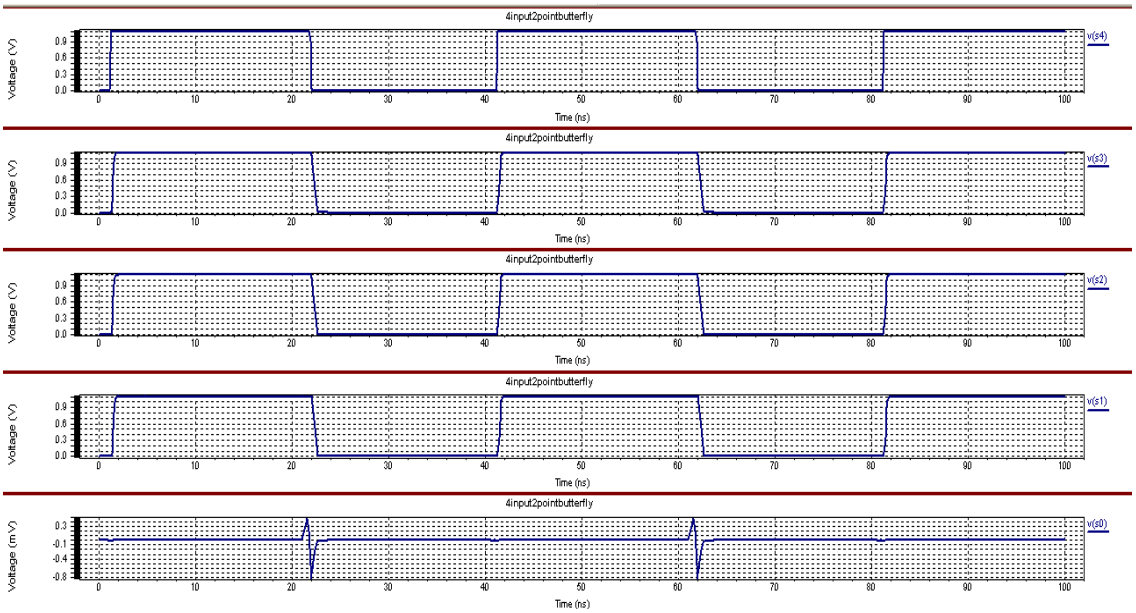


Fig 4(b): Output waveform of 2- Point Butterfly FFT.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 11, November 2019

**Implementation of 4 Point FFT**

4-point transform can be divided to two 2-point FFT's one for even terms, one for odd terms. The odd one will be multiplied by $W_4^k$. According to diagram this can be represented as two levels of butterflies as shown in fig 5.3. All the multipliers can be expressed as powers of the same $W_N$ using the identity $W_{N/2n} = W_{N2n}$.

$W_N^k = e^{\left(-\frac{i2\pi}{N}\right)k}$
$W_4^0 = 1$
$W_4^1 = -j$

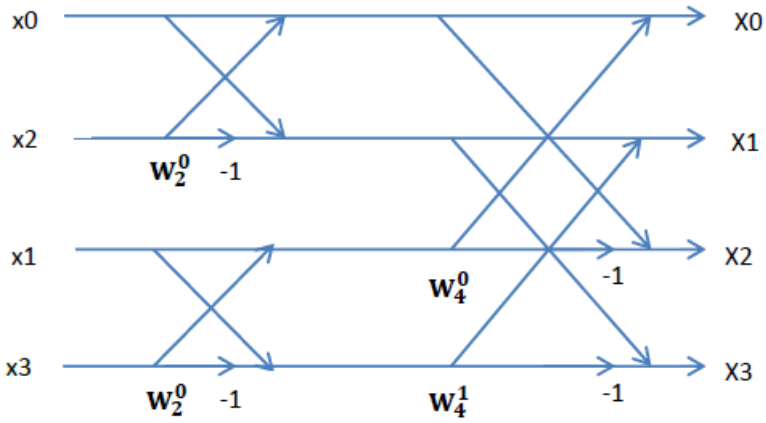


Figure 4: 4-Point Butterfly FFT.

Considering example x(n) = (0,1,2,3) and calculating 4 point DIT FFT algorithm, as shown in fig 8.4.
Here, N=4
$W_4^0 = 1$
$W_4^1 = -j$
Using DIT FFT algorithm, X(k) can be calculated from the sequence x(n), shown in fig 5.4.
Hence , X(k) = {6,-2+j2,-2,-2-j2}.

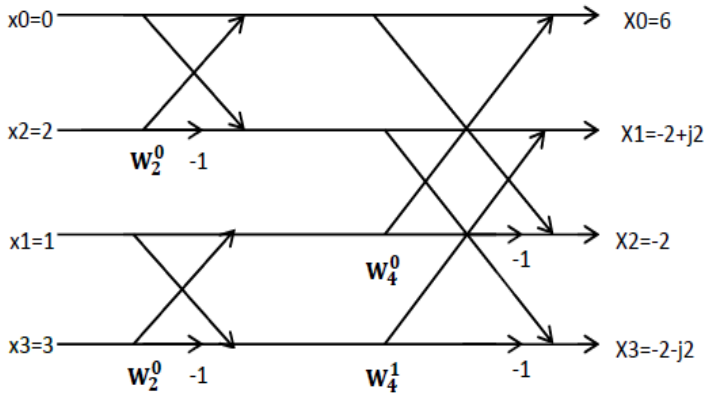


Figure. 5: Illustrated Example of 4-Point Butterfly FFT.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 11, November 2019

Fig 5 shows the schematic design of proposed 4 point butterfly FFT and the resulted waveform of 4 point butterfly respectively. It contains 2 2-Point Butterfly, Vedic Multiplier, and six adder/ subtractor units.

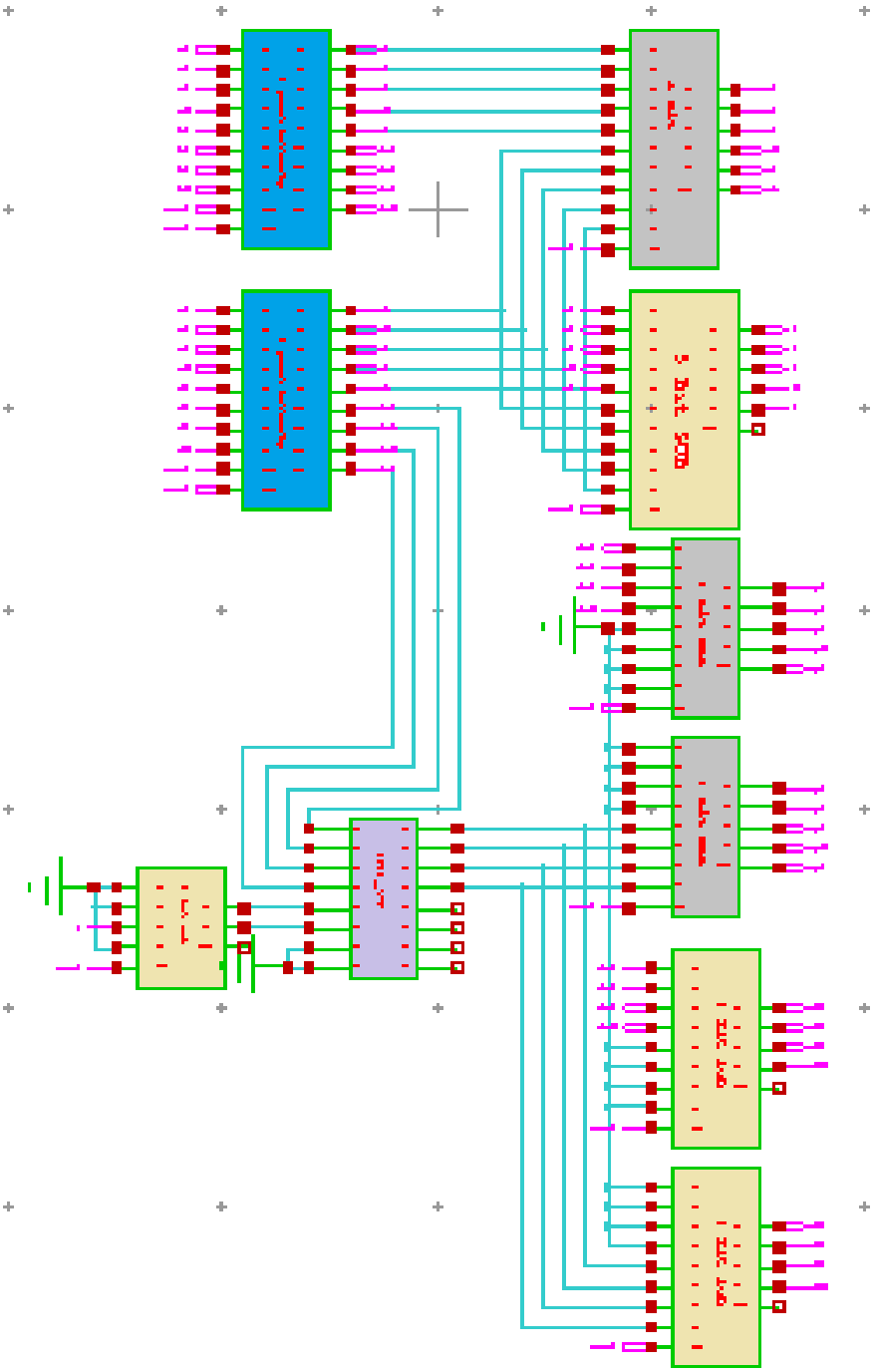

Figure 6 (a) Schematic Diagram of Proposed 4-Point Butterfly FFT.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Visit: www.ijirset.com*

**Vol. 8, Issue 11, November 2019**

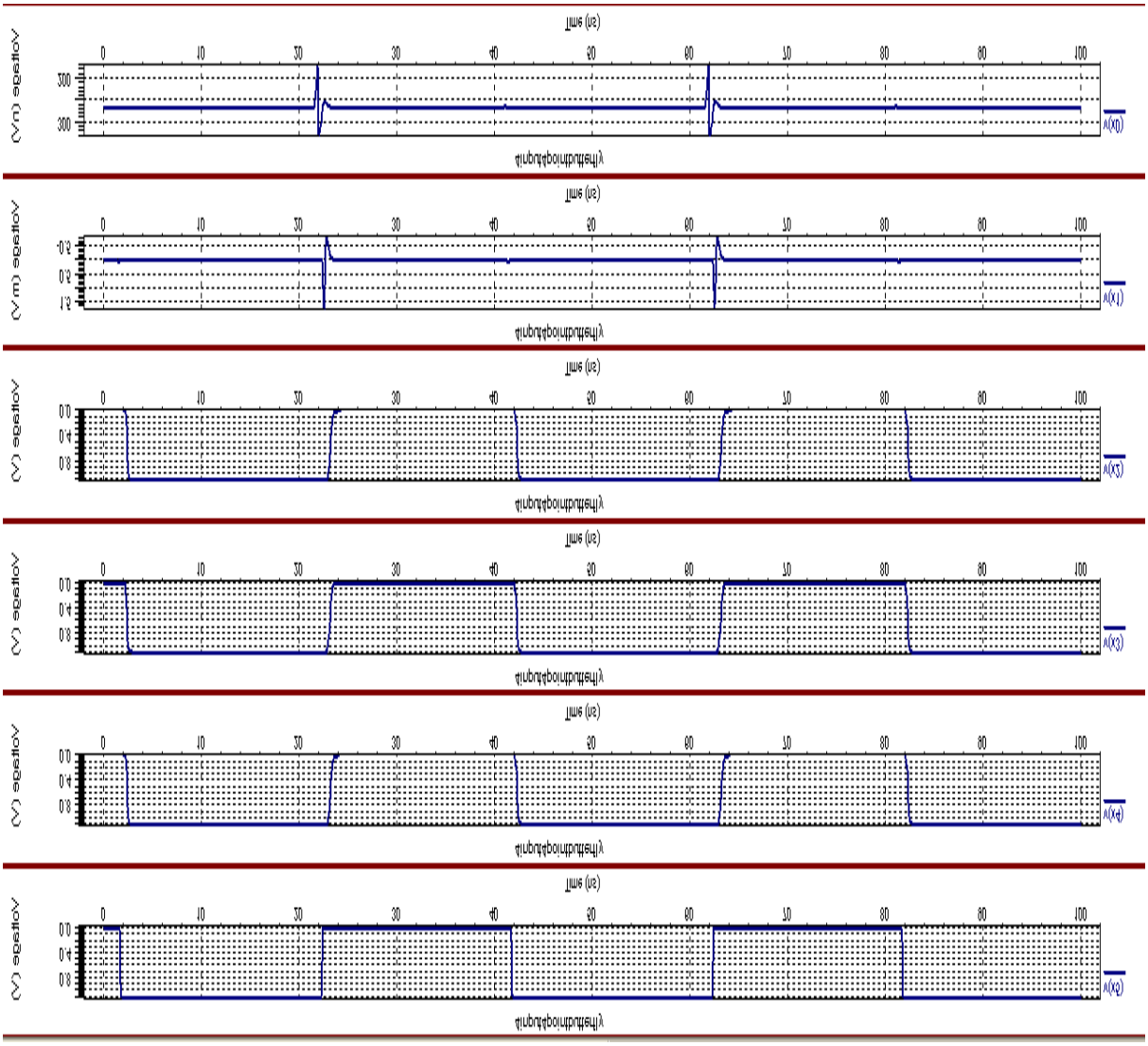

Fig. 7 (b): Output Waveform of 4-point Butterfly FFT.

The proposed 8 point FFT architecture is realized on Tanner EDA tool version 14.11 with 180NM CMOS technology library files. The design was built and tested for high performance and energy efficiency. The T spice simulation results verified the performance of proposed design. CMOS implementation is done with operation voltage of 1.1 V. Table 1 shows the, delay and power delay product of the proposed architecture after simulation.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 11, November 2019

Table1: Result Analysis

| Resources | Reported Design 1 [1] | Reported Design 2 [5] | Proposed Design |
|---|---|---|---|
| Average Power(watt) | $94x10^{6}$ | 1.2812 | $4.10x10^{-5}$ |
| Time Delay(sec) | $8.50x10^{-11}$ | $6.40x10^{-8}$ | $1.9x10^{-8}$ |
| Power Delay Product | .00799 | $8.19923x10^{-8}$ | $7.7x10^{-13}$ |

## V. CONCLUSION

Simulation results shows that the proposed Fast Fourier transform architecture represents a better and efficient architecture with lesser number of multiplications. This work proposes complex multiplier with reduced number of multipliers which in turn reduces the complexity of the overall design. Also a Constant complex multiplier is designed using adding and shifting operations for twiddle factor generation and multiplication required in Fast Fourier Transform algorithm. This multiplier increases the speed of the FFT designs and contributes to ROM less twiddle factor generators.

The Proposed 8 point Fast Fourier transform architecture is implemented on 180NM CMOS technology library files using tanner Tool. As Low power systems are of great need in current scenario, the T spice simulation results shows that the proposed FFT design offers low power and less delay as compared to reported literature [1] and [5], where total power obtained is 94MW. This is basically due to simplification of the mathematic algorithm in Multiplier operations, which provides the better computation of FFT Architecture.

## REFERENCES

[1] Dora Suarez, Renato J. Cintra, F´abio M. Bayer, Arindam Sengupta, Sunera Kulasekera, Arjuna Madanayake, "Multi-Beam RF Aperture Using Multiplierless FFT Approximation", electronic journal, volume 50,issue 24,2014/DOI-10.1049/el.2014.3561.

[2] Naman Govil,Shubhajit Roy Chowdhury "High Performance and Low Cost Implementation of Fast Fourier Transform Algorithm based on Hardware Software Co-design" , IEEE Region 2014

[3] Deepa Susan George & V.Sarada, "Low Power ROM less FFT Processor**"** ISSN (Print) : 2319 – 2526, Volume-2, Issue-2, 2013

[4] Dhanabal R, Bharathi V, Sujana D.V., Shruthi Udaykumar, Johny S Raj,Aravind Kumar V.N "Low Power Feed Forward FFT Architectures Using Switch Logic",Journal of Theoretical and Applied Information Technology, Vol. 62 No.3 30th April 2014.

[5] Rekha Masanam ,B.Ramarao "Area Efficient FFT/IFFT Processor for Wireless Communication", IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 4, Issue 3, Ver. III (May-Jun. 2014), PP 17-21.

[6] R.K. Bathija, R.S. Meena, S. Sarkar , Rajesh Sahu TINJRIT "Low Power High Speed 16x16 bit Multiplier using Vedic Mathematics", International Journal of Computer Applications (0975 – 8887) Volume 59– No.6, December 2012

[7] Akanksha Mandowara, Mukesh Maheshwari "Performance Analysis of Different Parallel CMOS Adders and Effect of Channel Width at 0.18um", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 8, October 2012.

[8] Siva Kumar Palaniappan and Tun Zainal Azni Zulkifli "Design of 16-point Radix-4 Fast Fourier Transform in 0.18µm CMOS Technology", American Journal of Applied Sciences 4 (8): 570-575, 2007 ISSN 1546-9239 © 2007 Science Publications

[9] Yuke Wang, Yiyan (Felix) Tang, Yingtao Jiang, Jin-Gyun Chung, Sang-Seob Song, Member, and Myoung-Seob Lim, "Novel Memory Reference Reduction Methods for FFT Implementations on DSP Processors", IEEE Transactions On Signal Processing, Vol. 55, No. 5, May 2007.