ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 10, April 2020

# CMOS Implementation of 2, 4, 8 point FFT using modified complex multiplier

Sarita Chauhan
Assistant Professor, ECE, MLVTEC, Bhilwara, Rajasthan

***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 paper 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. A low-complexity design for multiplication is essential requirement in Fast Fourier implementation. In this work an optimized constant complex multiplier for twiddle factor multiplication is designed using Vedic multipliers. The proposed multiplier is slightly modified from the existing complex multipliers. The presented complex multiplier with minimum complexity provides much less delay and simulation time, which reduces overall speed when implemented in FFT. 2 point, 4 point and 8 point FFT Butterfly design is proposed in this paper work and is simulated with 45nmCMOS technology library files with the help of Tanner EDA tool version 14.11. These designs include complex multiplier, Vedic multiplier and carry look ahead adder for computation with a improvement in terms of power consumption and delay. The proposed design is compared with reported structures with same 45nm technology.***

***Keywords:*** **FFT, Vedic Multiplier, Complex multiplier, Tanner EDA tool.**

## I. INTRODUCTION

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].

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].

**Manuscript received: 26 March 2020**
**Manuscript received in revised form: 19 April 2020**
**Manuscript accepted: 08 May 2020**
**Manuscript Available online: 15 May 2020**

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. LITERATURE REVIEW

The [1] implements a low-complexity multiplier less approximation for the 8-point FFT using only 26 additions and being synthesized in 45 nm CMOS technology at 1.1 V supply, and verified on-chip using a Xilinx Virtex-6 Lx240T FPGA device. CMOS synthesis and FPGA implementations are done with estimated power 94.59mw. [2] Implements a high performance novel architecture of Fast Fourier Transform (FFT) algorithm which uses the notion of Hardware Software Partitioning, and ensures optimality in power, delay and area. The power consumption of the co-design has been found to be 0.072W at a supply voltage 3.3V.Thus in order to achieve higher system performance and design flexibility, the code sign methodology was used, and implemented on both microcontroller and FPGA together. The simulation result [3] shows the reduction in the number of slices and LUTs used with which reduction in area and hence power is obtained and hence can be used for OFDM applications. [4] Describes the design of an ASIC (Application Specific Integrated Circuit) CMOS FFT processor which is computed utilizing 0.18µm standard CMOS Technology. Compared to traditional radix-4 algorithm the architecture proposed results in 1.73% of power saving and 5.5% of area reduction. NC launch tool [5] is used for the compilation and simulation process of the design, utilizing 130 nm, 90nm, 45nm CMOS technologies using Cadence RTL compiler. The timing, power and area savings are of 26.2, 66, 23.4 percentage respectively for the proposed design.

## III. COMPUTATION MODULES USED IN THE DESIGN

Different 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 twiddle factor generators, adders and multipliers in the VLSI designs is the basic requirement.

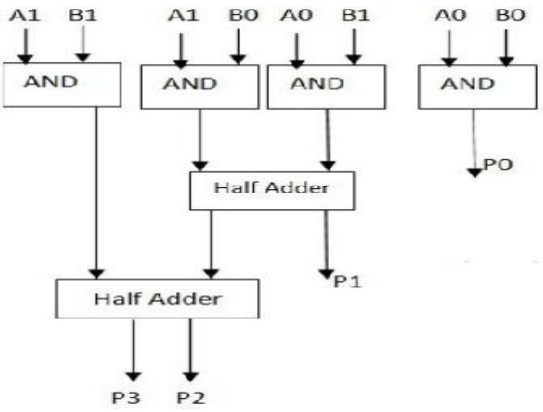
### *A. Urdhva-Tiryakbyham Multiplier*

Vedic Mathematics is the ancient system of Indian mathematics which has an exclusive technique of calculations based on 16 Sutras. Various multiplication schemes had been designed to increase the efficiency of the multiplier in algorithmic and structural levels, which

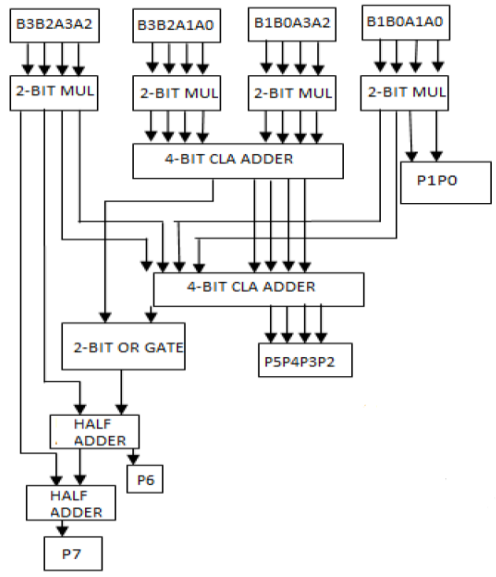
ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 10, April 2020

confronts the reduction of the partial products and the methods for their partial products addition, but the idea behind multiplication was same in all cases. Many multiplier design using "Urdhva-tiryakbyham" sutras, has been proposed which was adopted from the Vedas [7]. The complex multiplier includes multiplication using Urdhva Tiryakbhyam multipliers. The Sanskrit term means "Vertically and crosswise". Urdhva Tiryakbhyam is used in various multiplication techniques.

***1. Design of 4x4 UT Algorithm***




**(a)**




**(b)**

**Fig. 1 (a) 2x2 Urdhva Tiryakbhyam Multiplier [6]. (b) 4x4Urdhva Tiryakbhyam Multiplier [6].**

The design of 4x4 multiplier is using the combinations of input first given to the 2x2 multiplier block to produce the partial products. One way of designing 4x4 multiplier using ripple carry adders but this design covers large area and the delay time is also increases. Other way are using half adders assembling which reduces the delay time and the area? The design architecture is shown in figure4.4.The block diagram of 2bit, and 4 bit Urdhva-tiryakbyham multiplier is shown in fig 1(a), and fig 1(b) respectively. Here the partial products are generated along with addition of these in parallel.

***2. Design of 8x8 UT Multiplier***

The design of 8x8 multiplier is very much similar to 4x4 UT multiplier. The 8x8 is designed using 4x4 multiplier instead of 2x2 with the different input combinations with the half adder assembly. The partial products rows here added in an 8-bit carry look ahead adder optimally to generate final product bits. The structural architecture of 8x8 UT multiplier in figure 2.

The proposed Urdhva Tiryakbhyam Multiplier decreases the power consumption and reduces the delay of the overall circuit.

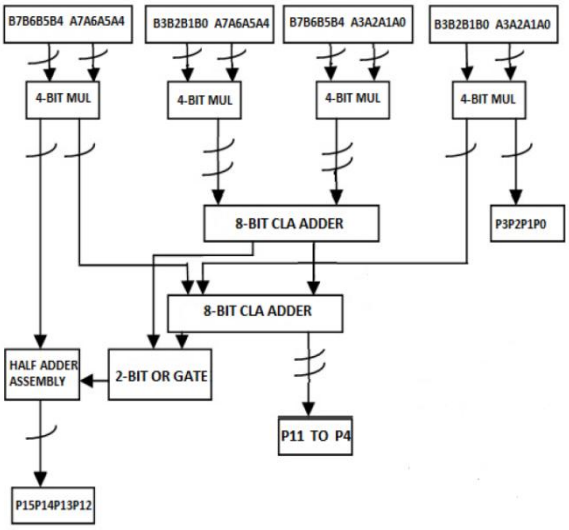



**Fig 2: 8x8 Urdhva Tiryakbhyam Multiplier [7].**

***B. Proposed Complex Multiplier Design***

Using Vedic mathematics, an N bit complex multiplication was characterized into three multiplications for final real and imaginary products. This sutra is employed for the multiplication, with minimal partial products, as compared with array multipliers. Transistor level implementation for power leakage, delay, and power consumed calculation of the proposed design was evaluated by tanner tool version 14.11 using 45 nm scales. This work resulted in simplified mathematical operations, in comparison with the other direct methods,

Multiplication is a very important part in complex multiplier design. The multiplication design based on Vedic sutra Urdhva Tiryakbhyam and carry look ahead adders, gives the efficient performance of multiplier. The Complex multiplier method as shown in fig 3 designed using this Vedic multiplier has high speed as well as low power consumption.

The complex multipliers [13] are the biggest combinational blocks in the design and lie-in the critical path. It is functionally very easy to compute the complex multipliers and reduce the critical path delay.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 10, April 2020

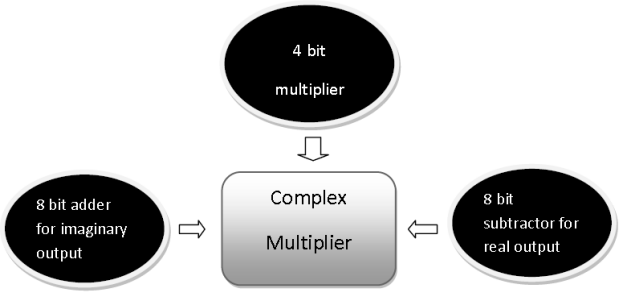



**Fig 3: Complex number multiplication method**

### *1. Architecture*

The complex multiplier implementation includes real part and imaginary part [13].

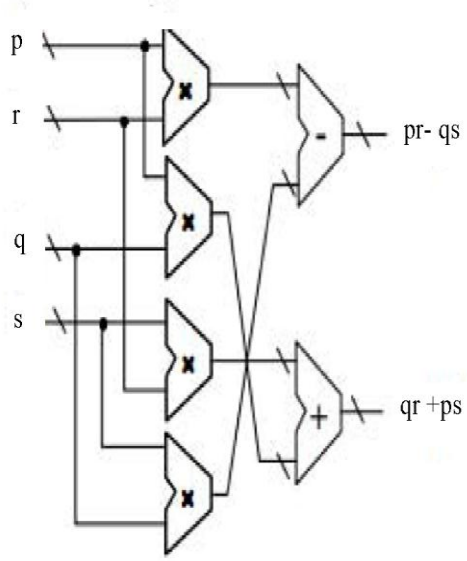



**Fig 4: Reported Complex Multiplier Design [13].**

p+jq)(r+js) = x+jy,

Where p+jq is one complex number and r+js is the another and, Eq 4.5 yields top

r + jps + jqr +$j^2$qs = x+jy,

pr + j(ps+qr) – qs = x+jy,

pr-qs + j(ps+qr) = x+jy,

On comparing right and left side equation x = pr-qs, and y = qr+ps, thus

According to eq, complex multiplier has been designed as shown in fig 4 with multiplications and addition/subtraction units. Thus complex number multiplication shows two results to calculate real and imaginary part. The real part of the output is obtained using pr-qs, and the imaginary part of the result can be obtained using qr+ps. Hence four multiplications and addition or subtraction are required for final output [1].

x = pr-qs + qr - qr, and y = qr+ps +pr – pr

x = (p+q)r –(s+r)q , and y = (p+q) r – (r-s) p

Real part, x = (p+q)r – (s+r) q

Img Part, y = (p+q) r – (r-s) p

This decreases the multipliers count from four to three and hence the overall speed of multiplier is enhanced. The proposed complex multiplier is shown in fig 5.

Obviously, using this factorization scheme, the system has some advantages. The number of real multiplications is reduced from four to three. And addition has less consumption than multiplication. So the system power consumption is also reduced.

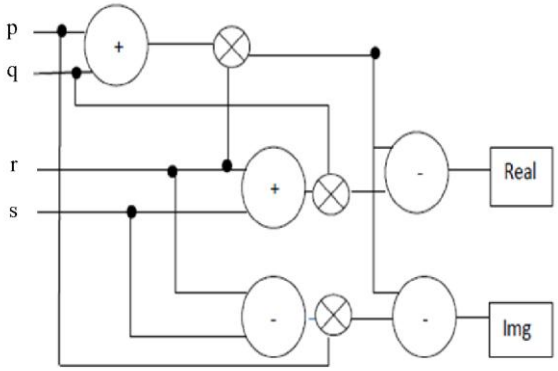



**Fig 5: Proposed Complex Multiplier Design**

### *C. Constant Complex Multiplier*

The signed digit (SD) system unlike the regular binary number system has is represented in three values 0,1, -1 (-1 is represented as 1). They are successful in implementing carry free adders or multipliers with less complexity. Since the complexity of the multiplier is typically estimated through the number of nonzero elements, which can be reduced by using SD numbers. The SD representation, unlike a 2C code, is no unique. We call a canonic digit system or CSD, the system that has least number of nonzero elements.

The proposed FFT architecture uses complex constant multipliers based on the canconical signed digit (CSD) representation for the complex multiplication arithmetic in stages 2 and 3.In our design, the complex CSD constant multiplier has been used for the twiddle factor $W_8$ multiplication. Also, the common sub-expressions sharing (CSS) technique reduces the complexity of the complex CSDconstant multipliers [10], as shown in Fig.6,7. The constant multiplier using the CSS technique is implemented using the common calculation patterns Y1, Y2, and Y3 [14].

The schematic view of proposed constant complex multiplier is shown in fig 8(a).Output waveform generated from T spice simulation is shown in fig 8(b). Multiplier used for Twiddle factor multiplication is shown in fig 6. The schematic and output waveform for this multiplier is shown in fig 9 (a) and (b) respectively.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 10, April 2020

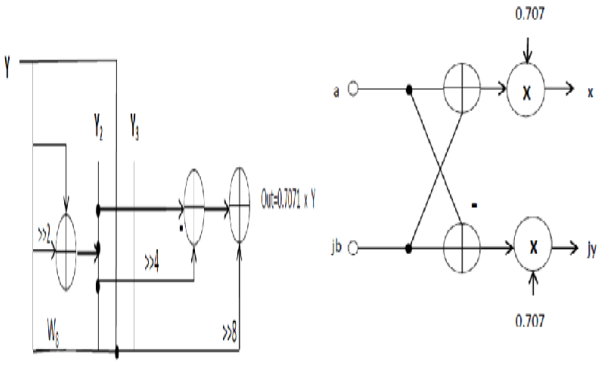



**Fig 6: Block diagram of the Complex Constant Multiplier**

**&**

**Fig 7: Multiplication by $W_N^{N/8}$.**

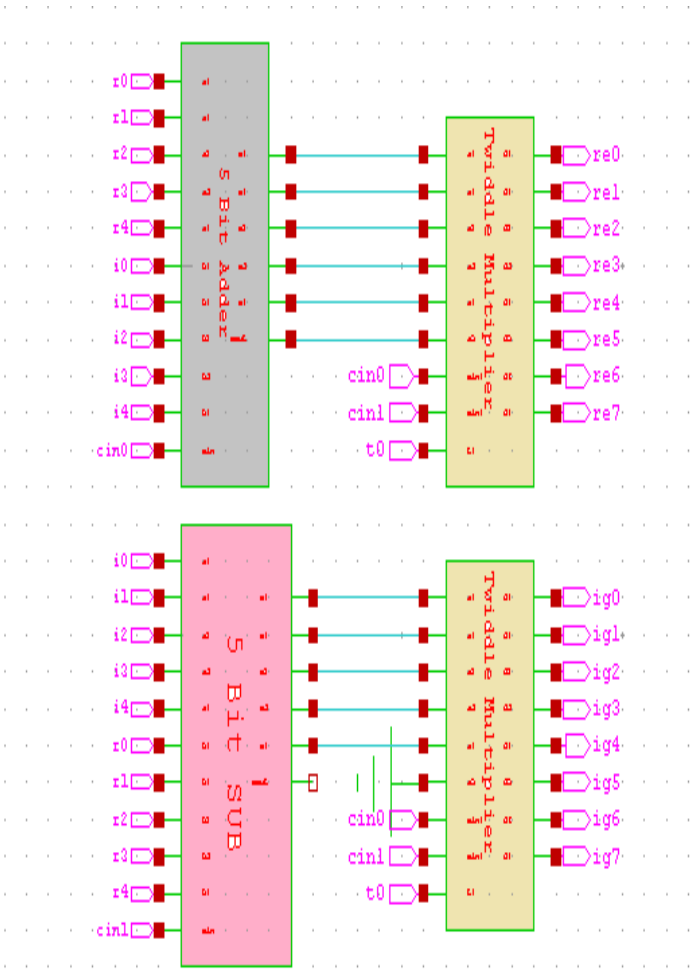



**Fig 8(a): Schematic of Proposed Constant Complex Multiplier.**

The schematic of complex multiplier shown in fig 8(a) uses two twiddle multiplier for twiddle factor multiplication with input, and two adder/subtractors. According to schematic of twiddle factor multiplier shown in fig 9 (a) three adders/ subtractors and right shifting operations are performed. The output gives the multiplication of twiddle factor 0.707 with input values as shown by output waveform.

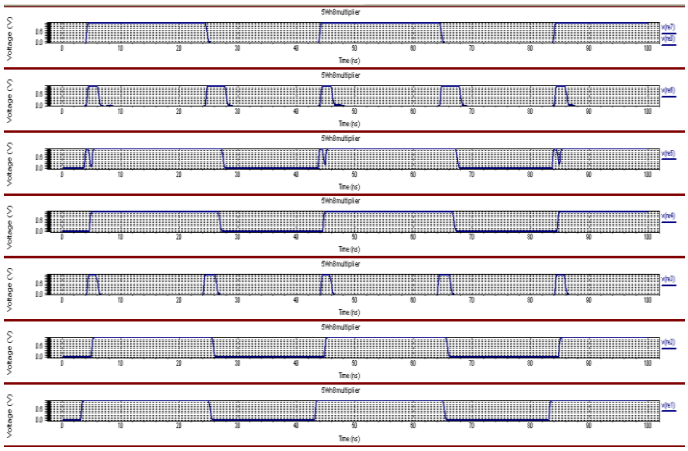


**Fig 8(b): Output Waveform of Proposed Constant Complex Multiplier.**

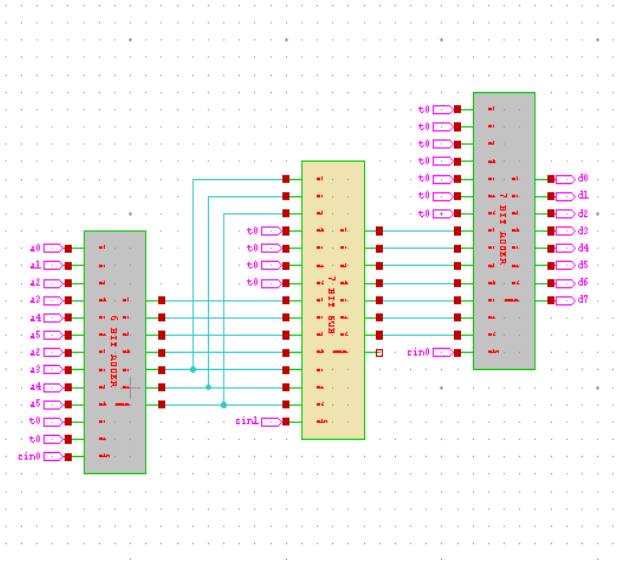


**Fig 9(a): Schematic diagram of Twiddle Factor Multiplication**

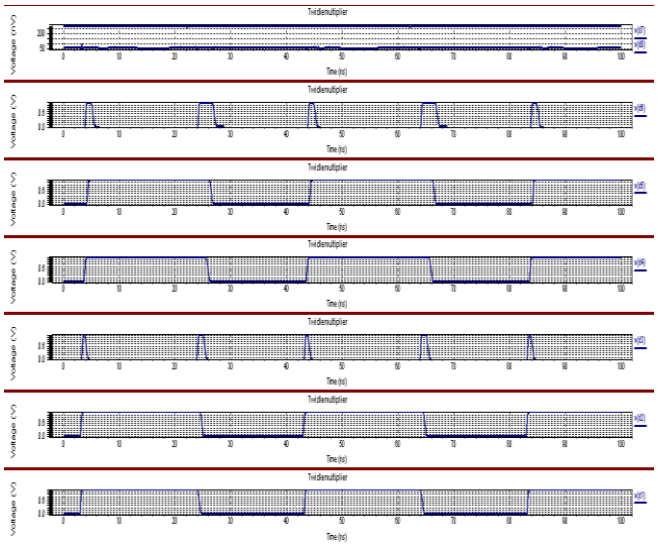


**Fig 9(b): Output Waveform of Twiddle Factor Multiplier.**

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 10, April 2020

## IV. PROPOSED FFT DESIGN

### *A. Introduction*

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 sizen = rm 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)".

### *B. Implementation of 2 Point FFT*

In 2 point FFT algorithm, butterfly is basic block which is shown in Fig. 10.

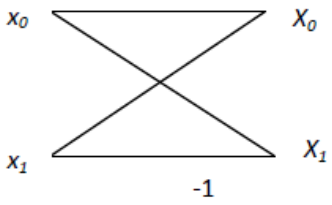



**Fig 10: 2-Point Butterfly FFT [1].**

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

$S_n = a_n + b_n$

$di_n = a_n - b_n$

Where the adding operation is performed by 4 bit Carry Look ahead adder and subtractions is done using 4 bit subtractor, as n take here is from 0 to 3 ( 4 bit).By reutilizing the results of smaller, intermediate computations to compute multiple DFT, these gain their speed, frequency outputs. Fig 11 (a) and (b) shows the schematic of 2 point FFT implementation and its output waveform generated from T spice simulation respectively.

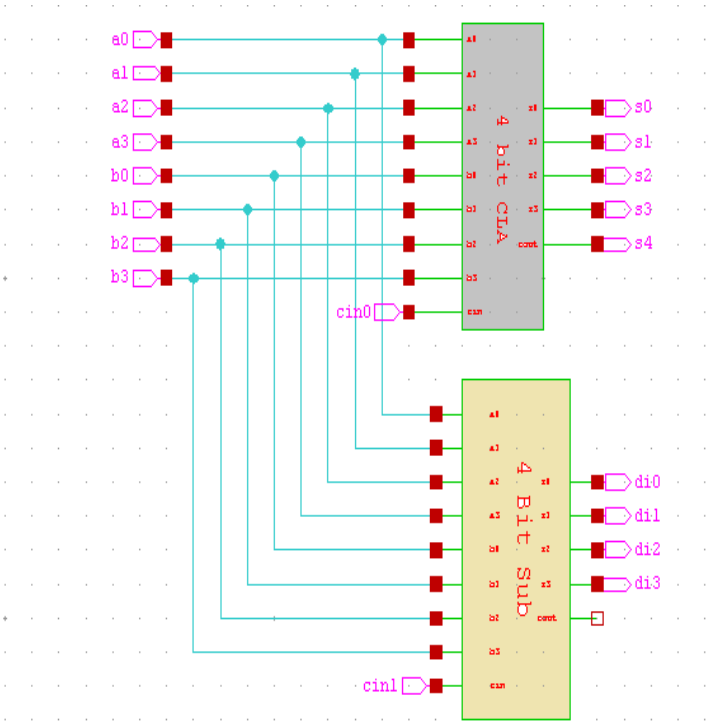



**Fig 11(a): Schematic Diagram of 2- Point butterfly FFT.**

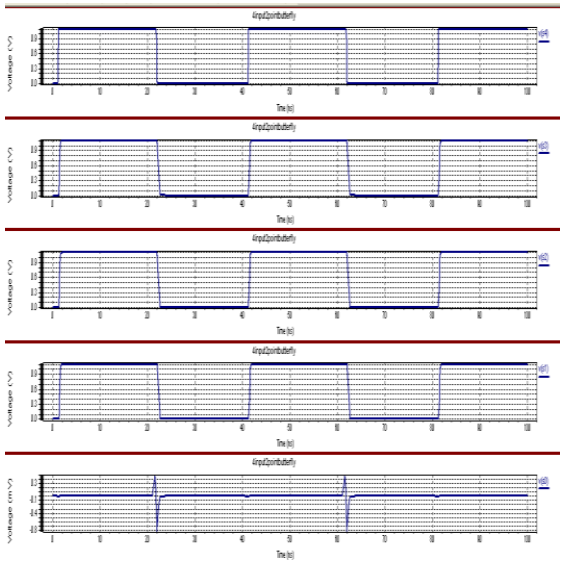


**Fig 11(b): Output waveform of 2- Point Butterfly FFT.**

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 10, April 2020

### *C. 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 fig5.3. All the multipliers can be expressed as powers of the same $W_N$ using the identity $W_{N/2n} = W_{N2n}$.

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

$W_4^0 = 1$

$W_4^1 = -j$

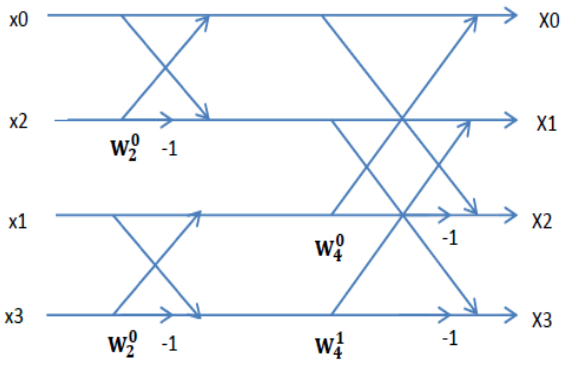



**Fig 12: 4-Point Butterfly FFT [4].**

Considering example x(n) = (0,1,2,3) and calculating 4 point DIT FFT algorithm, as shown in fig 12.

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 13.

Hence, X (k) = {6,-2+j2,-2,-2-j2}.

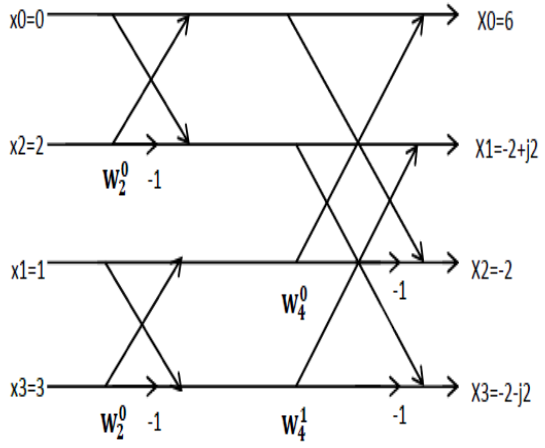



**Fig. 13: Illustrated Example of 4-Point Butterfly FFT [2].**

Fig 14(a) and (b) 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.

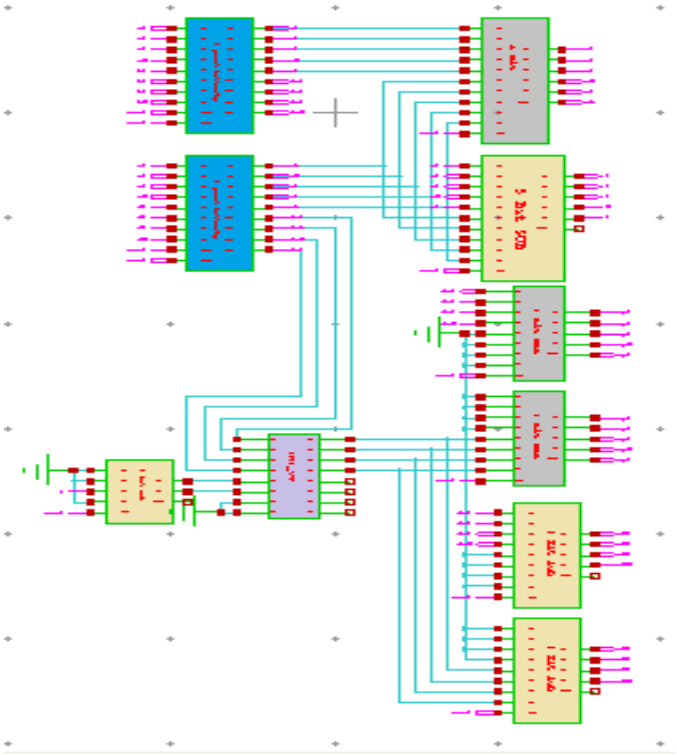


**Fig 14 (a) Schematic Diagram of Proposed 4-Point Butterfly FFT.**

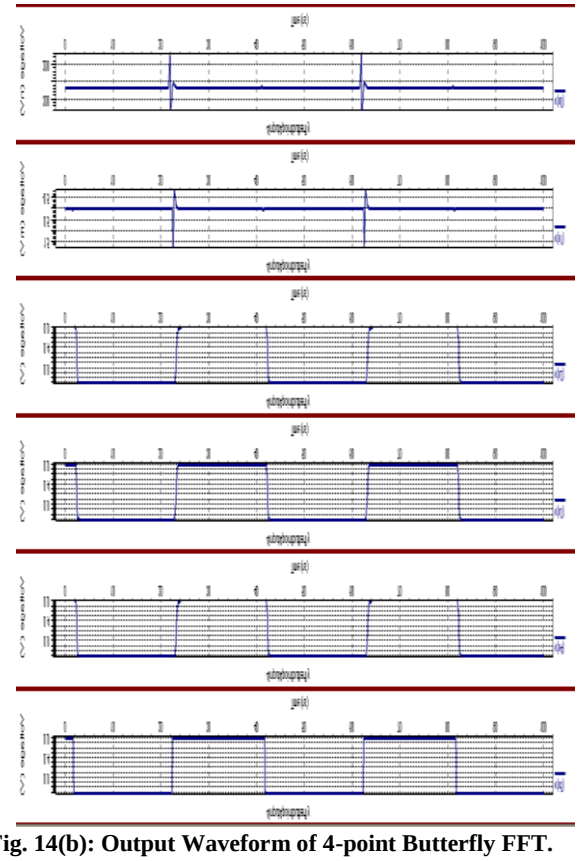


**Fig. 14(b): Output Waveform of 4-point Butterfly FFT.**

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 10, April 2020

### *D. Implementation of 8 point FFT*

In an 8 input butterfly diagram three are 12 2-input butterflies and thus 12x2 = 24 multiples as shown in fig 15. It can be shown as N Log N = 8 Log (8) = 24. A straight DFT has NxN multiples, or 8x8 = 64 multiples [5]. Hence this is saving of multiples for smaller samples. The savings are over 100 times for larger N thus increases as the number of samples increases. Fig 15 shows the 8-Point FFT dataflow diagram. Fig 16 (a) shows the schematic of 8 point butterfly unit and fig 16 (b) shows output waveform of the circuit.

The twiddle factor coefficient for 8- Point FFT is calculated as follows:

$W_8^0$ = 1

$W_8^1$ = 0.707 – j0.707

$W_8^1$ = -j

$W_8^2$ = - 0.707 – j0.707

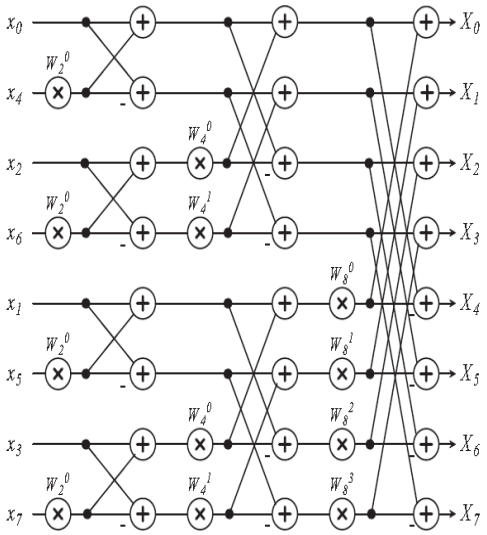


**Fig 15: 8-Point Butterfly FFT [3].**

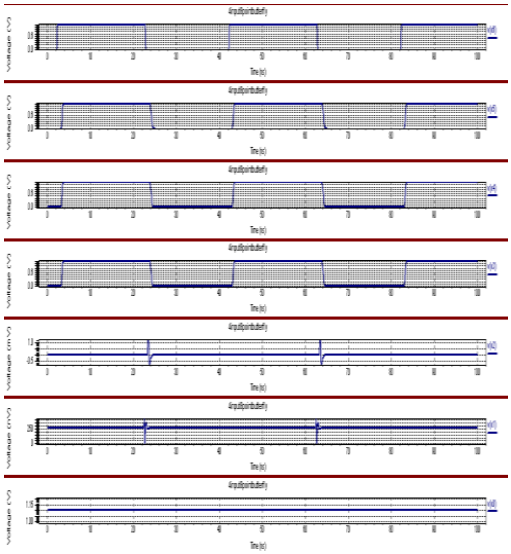


**Fig 16: output waveform of 8-Point Butterfly FFT**

## V. COMPUTAION RESULTS

The proposed 8 point FFT architecture is realized on Tanner EDA tool version 14.11 with 45nmCMOS 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.

The CMOS synthesis shows that the average power consumed is 41 microwatt and time delay of 19 ns is obtained. These low power and less delay results, in turn increases the efficiency of design in terms of speed and energy. This architecture also reduces the complexity of the

The proposed is compared with reported literature [1] and [5], the proposed consumed less power and of 41microwatt which low compared to reported design [1] and [5]. This shows that the overall low power delay product of proposed designs contributes to required high performance architecture.

**Table .1: 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 | $8.19923x10^{-8}$ | $7.7x10^{-13}$ |

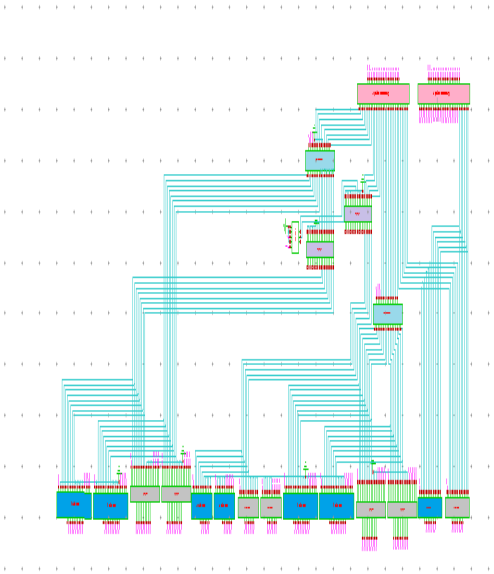


**Fig 17 Schematic of 8 point FFT butterfly**

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 10, April 2020

## VI. CONCLUSION AND FUTURE WORK

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 45nm 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.

Hence, the proposed architecture uses Constant Complex multipliers, Vedic multipliers, Carry Look ahead Adders which increase the speed of the design. On comparing with reported design the implemented structure works faster and utilizes lesser energy and thus gives good performance of the design.

***Future Scope***

This work sets a floor plan for implementing low power Fast Fourier Transforms. These designs are used where power, speed and complexity are concerned issues. The proposed Complex multiplier can be utilized in designs where less complexity is required. Using the proposed techniques larger points low power, less delay FFT's can be implemented.

## 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.

[10] Sergio Saponara, Nicola E. L'Insalata, Luca Fanucci , "Low-complexity FFT/IFFT IP hardware macrocells for OFDM and MIMO–OFDM CMOS transceivers", Microprocessors and Microsystems 33 (2009) 191–200.

[11] Adnan Suleiman, Hani Saleh, Adel Hussein, and David Akopian, "A Family of Scalable FFT Architectures and an Implementation of 1024-Point Radix-2 FFT for Real-Time Communications", 978-1-4244-2658-4/08©2008 IEEE.

[12] Anh T. Tran And Bevan M. Baas, "Design of an Energy-Efficient 32-bit Adder Operating at Subthreshold Voltages in 45-nm CMOS" International Conference On Communications And Electronics (ICCE), August 2010.

[13] Saha, P.; Banerjee, A. ; Bhattacharyya, P. ; Dandapat, A., "High speed ASIC design of complex multiplier using Vedic Mathematics", Students' Technology Symposium (TechSym), 2011 IEEE.

[14] Hari Chauhan, Student Member, Yongsuk Choi, Marvin Onabajo, "Accurate and Efficient On-Chip Spectral Analysis for Built-In Testing and Calibration Approaches", IEEE Transactions On Very Large Scale Integration (VLSI) Systems,2013".

## AUTHOR PROFILE

Smt Sarita Chauhan is working at MLVTEC ,Bhilwara in the capacity of Assistant Professor in ECE Department and has more than 17 years of experience .She has more than 20 publications in International Journals & Author of 2 books under RTU.Her field of interest are NT, VLSI,VHDL,ANFIS,NN, AE,EDA ,Image processing etc.