

**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 Using 180nm Technology

Priyanka Dhayal, Yashika Saini

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

Assistant Professor, RIET, Jaipur, India

ABSTRACT: 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.

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]. 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

FFT computation is the useful technique to convert the domains i.e the time domain to the frequency domain and the reverse for the implementation on digital hardware is the DFT. For N point DFT of a complex data sequence $x(n)$ is defined in Eq. 1:

$$X(K)=\sum_{n=0}^{N-1} x(n)W_N^{kn}, k = 0,1, \dots \dots \dots, N - 1 \quad (1.1)$$

Where:

$x(n)$ and $X(k)$ are the Complex numbers

$W_N^k = e^{-j2\pi/N}$ is the Twiddle factor

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

FFT algorithm first divides the system and then calculates i.e it divides the N coefficient points into smaller blocks in different stages. The first stage is implemented with groups of two coefficients, producing N/2 blocks, with each stage calculating the addition and subtraction of the coefficients multiplied by the respective twiddle factors, known as a butterfly. The outputs of these are then used to calculate the next N/4 blocks, which in turn tie up the outputs of two previous blocks, combining four coefficients at this stage. The process is repeated till one main block is produced, with a final calculation of all N coefficients [21].

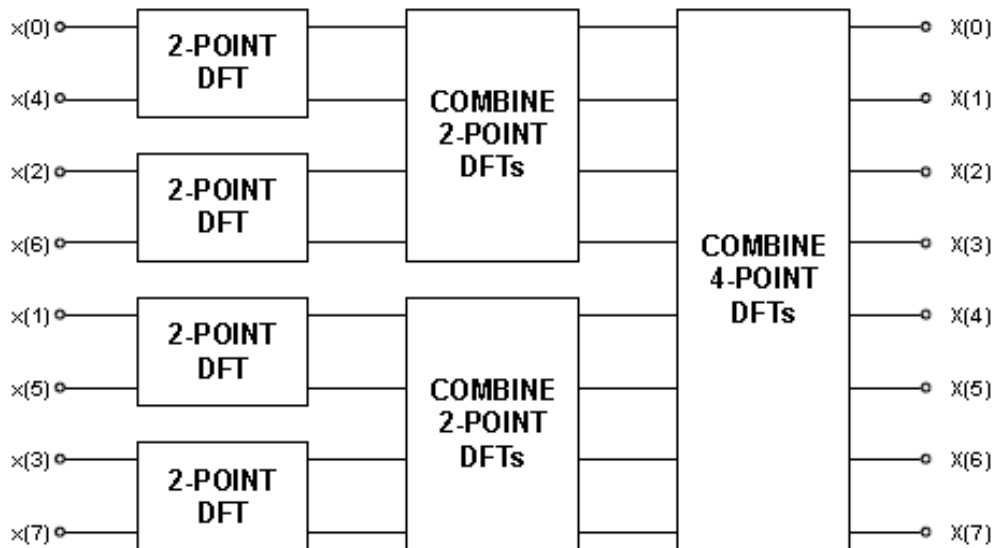


Figure 1 Computation of an 8-Point DFT in Three Stages using Decimation-In-Time .

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

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

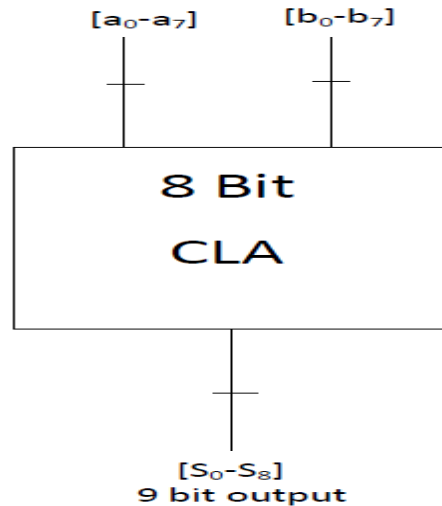


Figure 3: Block Diagrams of 8 Bit Carry Lookahead Adder.

In order to reduce the time to transfer the carry, the carry look-ahead adder is computed by calculating first the carry signals, as per the input signals. Carry Look Ahead adder block diagram of 4 bit input is shown in Fig.2. Fig 3 shows the block diagram of 8 input Carry look ahead adder.

The Boolean expression is manipulated according with full adder in order to understand how the CLA works. The both signals will be valid after one gate delay and propagate and generate signals depend only on the input bits. The Propagate P and generate G in a full-adder, is given as:

$$P_i = A_i \oplus B_i \quad \text{Carry propagate} \quad (1)$$

$$G_i = A_i + B_i \quad \text{Carry generate} \quad (2)$$

The output carry and sum expression is given by:

$$C = G_i + P_i C_i \quad (3)$$

$$S = P_i \oplus C_i \quad (4)$$

The general expression of carry signal is

$$C_{i+1} = G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \dots + P_i P_{i-1} \dots P_2 P_1 G_0 + P_i P_{i-1} \dots P_1 P_0 C_0 \quad (5)$$

The above expression clears that the output carry will compute without waiting for the previous carry. Taking an example of 4-bit adder as shown in figure 4.11 and apply these equations to it, the output carries C_1 , C_2 , C_3 and C_4 (last Stage Carry) will be compute as:

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

It is clear from above equations that carries C_2 , C_3 and C_4 not depend on the previous carry. Hence, the next carry need not to wait for the previous carry like C_4 will not wait for C_3 to propagate. As soon as C_0 is calculated, C_4 can reach steady state. Similarly, it is true for C_2 and C_3 . Fig 4.13 (a), and (b) shows the schematic of 4 bit carry look ahead adder and its output waveform respectively.

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

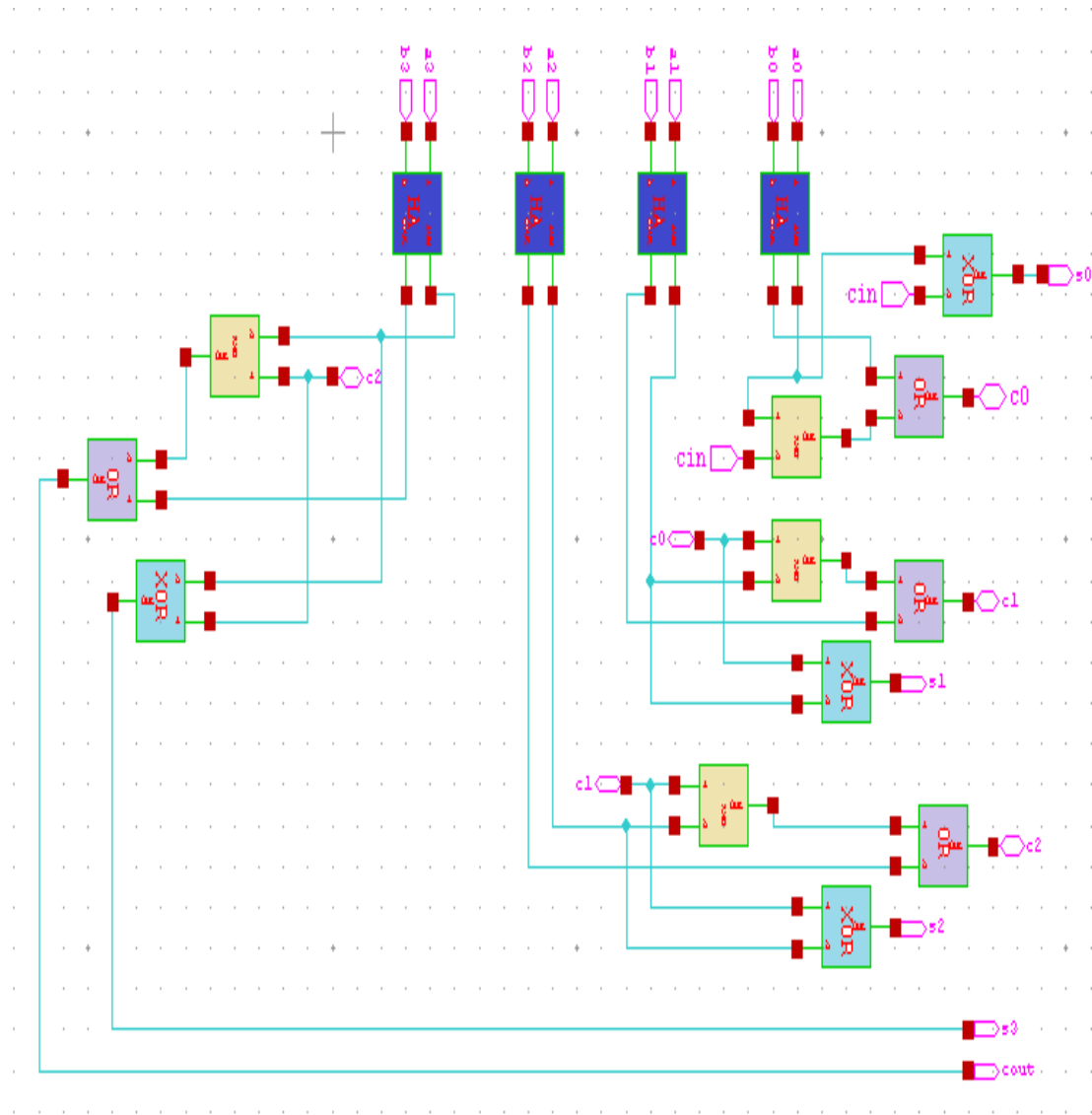


Figure 4 (a): Schematic diagram of 4 Bit Carry Lookahead Adder.

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

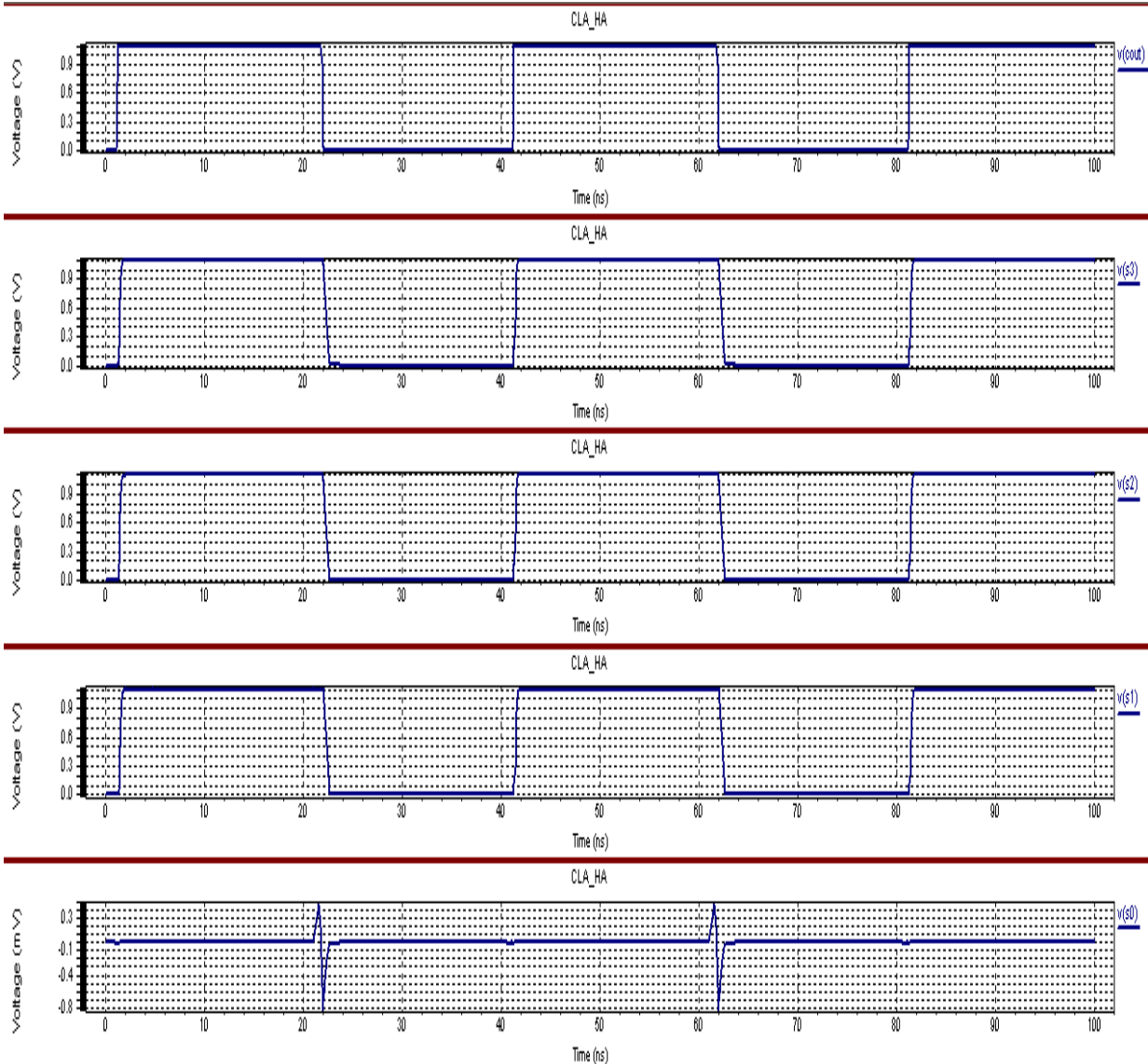


Figure 5(b): Output waveform of 4 bit Carry Lookahead Adder

According to above fig the 4 bit Carry Look Ahead Adder Schematic contains four half adders and various AND, OR, and XOR logic operations. The 5 bit output is produced after the simulation process. The 8 input carry look ahead adder schematic diagram is shown in fig 4(a) with its output waveform resulted from T spice simulation is given in fig 5(b).

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.

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

Table 6.1: Result Analysis

Resources	Reported Design 1 [1]	Reported Design 2 [5]	Proposed Design
Average Power(watt)	94×10^6	1.2812	4.10×10^{-5}
Time Delay(sec)	8.50×10^{-11}	6.40×10^{-8}	1.9×10^{-8}
Power Delay Product	.00799	8.19923×10^{-8}	7.7×10^{-13}

VI. 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 0.18um 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.