ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Performance and Analysis for OFDM Application Using Pipeline Architecture of Point-16 Radix-4 with CFFT

Sonali R. Ghonge[1], Prof. Amol Boke[2]

P.G. Student, Department of Electronics and Communication, G.H.R.A.E.T., Nagpur, India[1]

Assistant Professor, Department of Electronics and Communication, G.H.R.A.E.T., Nagpur, India[2]

ABSTRACT: In this paper, an optimized implementation of the 16-point Radix-4 FFT algorithm with Cyclotamic Fast Fourier Transform (CFFT) will be presented. Cyclotomic FFT is type of FFT algorithm over finite fields. This algorithm decomposes a DFT into several circular convolutions and then derives the DFT results from the circular convolution. In previous paper, the pipeline architecture has been implemented for radix-**2^**k FFT. This paper shows trade-off between area and performance. Thus, Cyclotomic FFT will help to reduce this trade-off over FFT. In this paper, the Radix-4 16-FFT algorithm with CFFT were designed using VHDL (Very High Speed Integrated Circuit Design Hardware Description Language) and this design is useful for OFDM application. In this paper the analysis of FFT with radix-4 designed using VHDL and their performance are analysed.

KEYWORDS: Fast Fourier Transform (FFT), radix-4, Discrete Fourier Transform (DFT), Cyclotomic FFT(CFFT), OFDM (Orthogonal Frequency Division Multiplexing), VHDL, DIT(Decimation in Frequency).

I. INTRODUCTION

The fast Fourier transform (FFT) class of algorithmsis widely used in communication and digital signal processing. The FFT algorithm is considered one of the basic algorithms in many DSP projects. Nowadays, FFT is the key building block for the mobile communications; especially for the Orthogonal Frequency Division Multiplexing (OFDM) transceiver systems.Implementation of FFT of different architectures, for fast and efficient computational schemes, has attracted many researchers. The methodology of FFT simulation, implementation, and verification plays a key role in the industrial or consumer electronics areas, for example, the FFT image processing, encoding and decoding, harmonic analysis in renewable energy and so on. The FFT is a typical computation where the memory access intensively and the high parallelism is needed. VLSI realization of FFT algorithm, should have pipelined architecture and/or parallelism, be regular and modular. At algorithm level, it should achieve the multiplicative complexity as low as possible. At the architecture level, use the delay-feedback buffering strategy to minimize the memory size. It should have modular and regular modules, local routing, and low control complexity [6]. The discrete Fourier transform (DFT) X(k) of an N-point sequence x(n) is defined by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad (1)$$

$$W_N^{nk} = \cos\left(\frac{2\pi * nk}{N}\right) - j \sin\left(\frac{2\pi * nk}{N}\right) \quad (2)$$

In (2), the $W_N^{nk}$is usually referred to as twiddle factor [6].

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

The discrete Fourier transform is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fieldsbut computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly: computing the DFT of N points in the naive way, using the definition, takes O(N2) arithmetical operations, while a FFT can compute the same DFT in only O(N log N) operations. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to N / log(N). This huge improvement made the calculation of the DFT practical, FFTs are of great importance to a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers [11].

II. BACKGROUND

II.1 RADIX - 4 FFT

The decimation-in-time (DIT) radix-4 FFT recursively partitions a DFT into four quarter-length DFTs of groups of every fourth time sample. The outputs of these shorter FFTs are reused to compute many outputs, thus greatly reducing the total computational cost. The radix-4 decimation-in-frequency FFT groups every fourth output sample into shorter-length DFTs to save computations. The radix-4 decimation-in-time algorithm rearranges the discrete Fourier transform (DFT) equation into four parts [6]. High-radix FFT algorithms, such as radix-8, often increase the control complexity and are not easy to implement. And to radix-2 FFT, there is the increase in the number of butterfly elements compared with radix-4. So the radix-4 was selected in this paper [6].The DFT formula is split into two summations [6]:

$$X(K) = \sum_{n=0}^{\frac{N}{4}-1} x(n)\cdot W_N^{nk} + \sum_{n=\frac{N}{4}}^{N-1} x(n)\cdot W_N^{nk} = \sum_{n=0}^{\frac{N}{4}-1} x(n)\cdot W_N^{nk} + \sum_{n=0}^{\frac{N}{4}-1} x\left(n+\frac{N}{4}\right)\cdot W_N^{(n+\frac{N}{4})k}$$

$$X(K) = \sum_{n=0}^{\frac{N}{4}-1} x(n)\cdot W_N^{nk} + \sum_{n=\frac{N}{4}}^{\frac{N}{4}-1} x\left(n+\frac{N}{4}\right)\cdot W_N^{nk}\cdot W_N^{\left(\frac{N}{4}\right)k}$$

And $W_N^{\frac{N}{2}k} = (-1)^k$

$$X(K) = \sum_{n=0}^{\frac{N}{4}-1} [x(n) + (-1)^k\cdot\left(n+\frac{N}{4}\right)]\cdot W_N^{nk} \qquad (1)$$

X (K) can be decimated into even and odd index frequency sample.

$$X(4K) = \sum_{n=0}^{\frac{N}{4}-1}\left[x(n) + \left(n+\frac{N}{4}\right)\right]\cdot W_N^{2n\cdot k} = \sum_{n=0}^{\frac{N}{4}-1}\left[x(n) + \left(n+\frac{N}{4}\right)\right]\cdot W_{\frac{N}{4}}^{n\cdot k} \qquad (2)$$

$$X(4K+1) = \sum_{n=0}^{\frac{N}{4}-1}\left[x(n) - \left(n+\frac{N}{4}\right)\right]\cdot W_N^{2n\cdot k} = \sum_{n=0}^{\frac{N}{4}-1}\left[x(n) - \left(n+\frac{N}{4}\right)\right]\cdot W_{\frac{N}{4}}^{n\cdot k} \qquad (3)$$

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

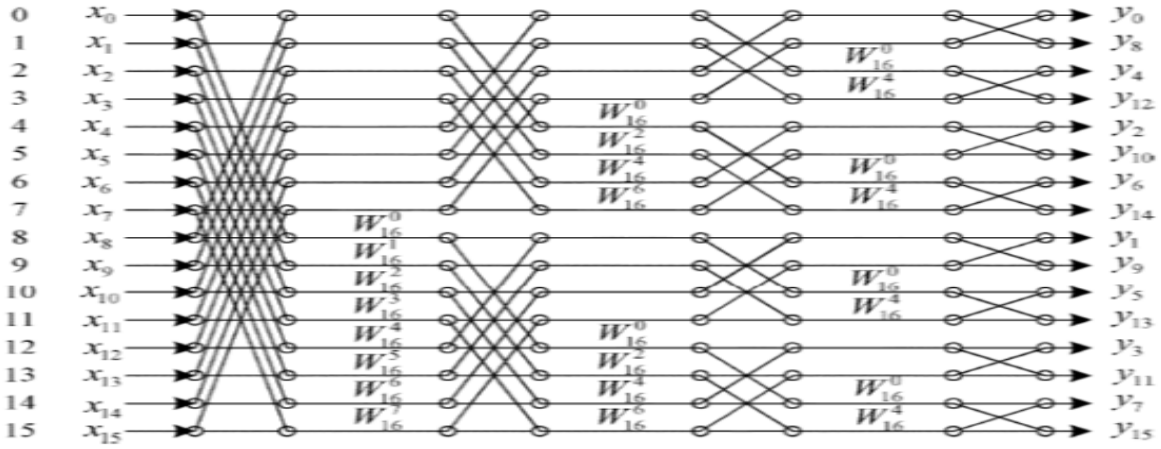

Fig1: - Signal flow graph of radix-4 16-point DIT FFT

Above figure (1) shows the Signal flow graph of radix-4 16-point DIT (Decimation in Time) FFT. In this figure all sixteen inputs are given and radix-4 is applied on these structure therefore output sequences are sampled by 4 and gives output with their twiddle factors.

II.2 CYCLOTOMIC FAST FOURIER TRANSFORM (CFFT)

The Cyclotomic Fast Fourier Transform is a type of Fast Fourier Transform algorithm over finite fields. This algorithm first decomposes a DFT into several circular convolutions, and then derives the DFT results from the circular convolution results. When applied to a DFT over GF (2^m), this algorithm has a very low multiplicative complexity.The discrete Fourier transform over finite fields find widespread application in the decoding of error-correcting codes such as BCH codes and Reed-Solomon code.
Generalized from the complex field, a discrete Fourier transform of a sequence $\{f_i\}_0^{N-1}$over a finite field GF(p^m) is defined as

$F_i = \sum_{i=0}^{N-1} f_i \propto^{ij}, 0 \leq j \leq N\text{-}1,$

Where $\propto$ is the N-th primitive root of 1 in GF(p^m). If we define polynomial representation of $\{f_i\}_0^{N-1}$ as

$f(x) = f_0 + f_1 x + f_2 x^2 + \ldots\ldots f_{N-1} x^{N-1} = \sum_0^{N-1} f_i x^i,$

It is easy to see that F_j is simply$f(\propto^j)$. That is, the DFT of sequence converts it to a polynomial evaluation problem. There are efficient algorithms which can be applied to perform the cyclic convolution efficiently and then reduce the computation complexity of the FFT [4].

Fast Fourier transforms (FFTs) based on the prime-factor algorithm [3] and the Cooley-Tukey algorithm have been proposed for DFTs over the complex field. When FFTs based on the prime-factor algorithm are adapted to DFTs over finite fields [3], they still have high multiplicative complexities. In contrast, recently proposed Cyclotomic FFTs (CFFTs) are promising since they have significantly lower multiplicative complexities.

CFFTs have two issues. First, they rely on efficient algorithms for short cyclic convolutions, which do not always exist. Second, CFFTs have very high additive complexities when directly implemented, which can be reduced by techniques such as the common sub-expression elimination (CSE) algorithm[4].

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

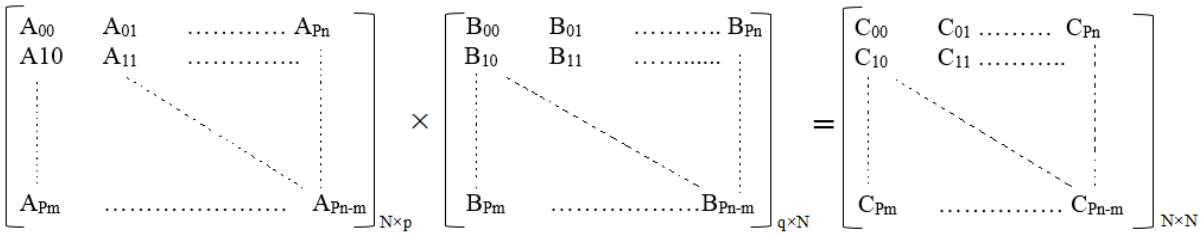

Fig 2: - Multiplication of two different size of matrix

In the above figure the multiplications of two matrices are performed. In this, first matrix of row and column are multiplied with second matrix of row and column respectively and those multiplied matrix are formed in output matrix with N- length.

II.3 Memory Design

The memory (RAMs) in the architecture, which store the input values of data and Co-efficient, and the output value of the result. The RAM is designed to store one full set of data for 16-point FFT computation. Both the real and imaginary parts of the data are stored in fixed point representation as two different numbers. So for a radix-4 butterfly operation we have 4 numbers (2-real, 2-imaginary) to be stored and each number is 16-bit long. The data can be read serially to be processed in the radix computation element [6].

III. PROPOSED WORK

Cyclotomic FFT is type of FFT algorithm over finite fields. For FFT structure, first we have need some adders and multipliers to design the structure with complex conjugate and get the output using basic structure. This computations can be used in whole deign of the CFFT. This paper proposes the additive and multiplicative complexities of FFT with area efficient and delay using CFFT using N× N matrix which gives m-matrix. In this matrix, first matrix requires N × p length and second matrix requires q × N length, the multiplication of those both matrix will applied on butterfly FFT structure. The image of two matrix multiplication is shown in fig2 above.Fig 3 shows the block diagram of FFT with CFFT. In this block diagram, matrix is applied to the all stages of FFT with butterfly.

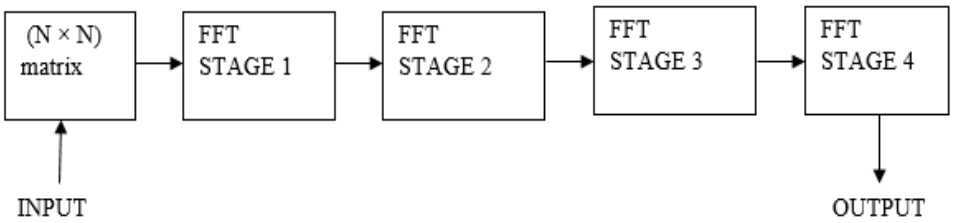

Fig 3:- Block structure of different stages of FFT with CFFT

IV. RESULT ANALYSIS

In Fig 4, it shows simulated result of all four stages of FFT with CFFT using serial multiplier. 16-bit radix-2^k serial multiplier has been implemented using Virtex family in VHDL. In this, each stages of FFT are simulated with serial multiplier which requires 64 clock cycle i.e., in serial multiplier each stage of FFT requires 16 clock cycle. Fig 5 shows the simulation result of complete FFT structure. In this block the pipeline multiplier gives output in 16 clock cycle.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

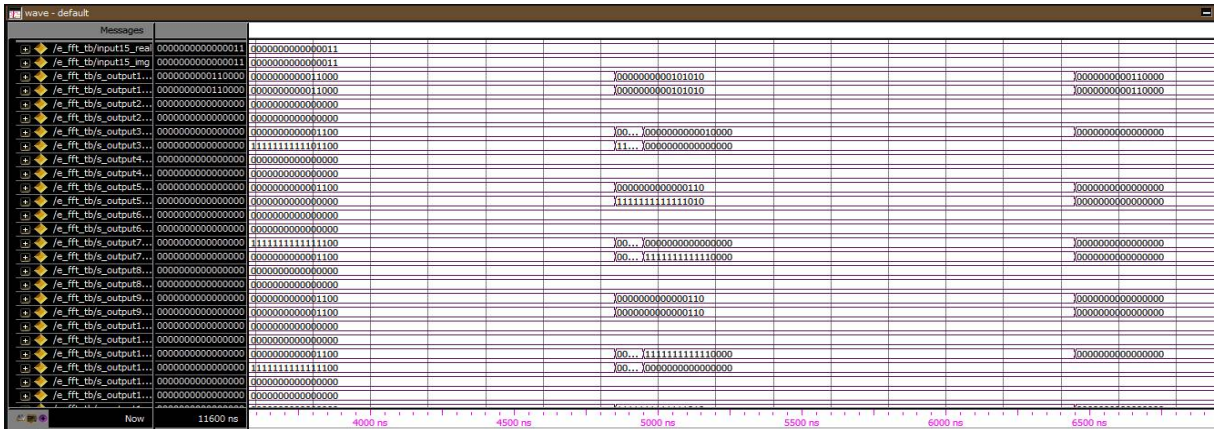

Fig 4: - Simulated result of all four stages of FFT with CFFT

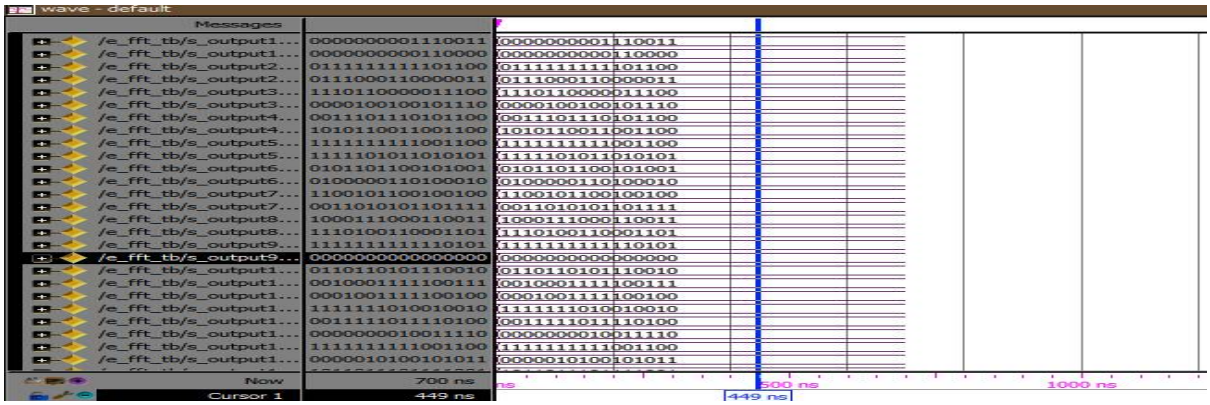

Fig 5: - Simulated result of complete FFT

Fig 6 shows the RTL view of computed overall FFT and Fig 7 shows the internal block structure of FFT stages. In this structure the number arithmetic operators of addition, subtraction, multiplication and shifters are connected to each other and these blocks are uses to get the simulation of all FFT stages.

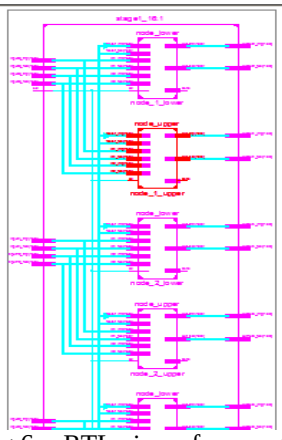

Fig 6: - RTL view of computed overall FFT.

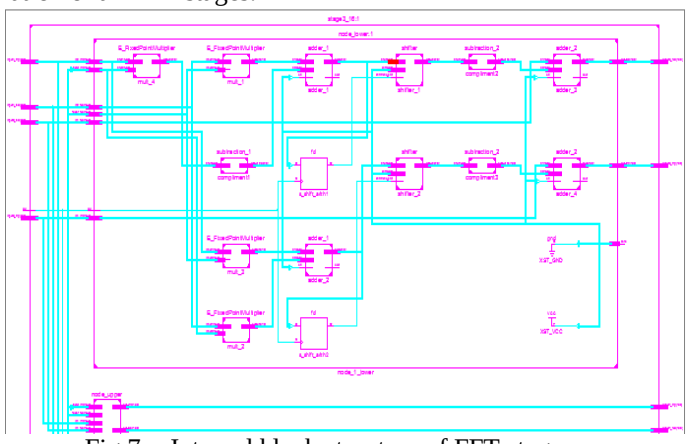

Fig 7: - Internal block structure of FFT stage

The comparison of previous paper and proposed paper which shows area of previous paper is more than the proposed method. In the previous paper the FeedForword FFT were used on butterfly structure therefore there latency of structure i.e., 0.026μs is more than proposed FFT and latency of proposed FFT is 0.019 μs. The presented FFT has been

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

synthesized with HDL by Virtex5 xc5vfx30t-3ff665 using fixed point arithmetic.This architecture needs parallel multipliers to optimized lowest delay.

V. CONCLUSION

In this paper, the algebraic method called Cyclotomic FFT were introduced and computing Fast Fourier Transform with radix-4 16-point DIT FFT in terms of multiplicative and additive complexities. This architecture has been modelled with Hardware Description language VHDL and ModelSim tool with generic parameters using fix point arithmetic operators. The simulation results shows that the proposed method performs CFFT with butterfly FFT. The proposed method requires 6400ns of clock for simulation of full FFT with serial multiplier and 100ns of clock for simulation of full FFT with parallel multiplier.

REFERENES

1. Mario Garrido, Member, IEEE, J. Grajal, M. A. Sánchez, and Oscar Gustafsson, "Pipelined Radix- 2^k Feed forward FFT Architectures", Senior Member, IEEE; IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 21, NO. 1, JANUARY 2013.
2. MounirArioua, Said Belkouch, Mohamed Agdad, "VHDL implementation of an optimized 8-point FFT/IFFT processor in pipeline architecture for OFDM systems", 978-1-61284-732-0/11/$26.00 ©2010 IEEE.
3. Jerônimo da Silva Jr. and R. M. Campello de Souza "Cyclotomic Basis for Computing the Discrete Fourier Transform" The 7th International Telecommunications Symposium (ITS 2010), Department of Electronics and SystemsUFPE, CP7800, 50711-970Recife PE, Brazil.
4. Xuebin Wu, MeghanadWagh, Ning Chen, Member, IEEE, Ying Wang, Member, IEEE, andZhiyuan Yan, Senior Member, IEEE, "Composite Cyclotomic Fourier Transforms With Reduced Complexities", IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 59, NO. 5, MAY 2011.
5. Y.Jung, H.Yoon, J.Kim, "New Efficient FFT Algorithm and Pipeline Implementation Results for OFDM/DMT Applications," IEEE Transactions on Consumer Electronics, vol.49, No.1, 2003, pp. 14-20.
6. Jiang Wang and Leif Arne Ronningen, "An Implementation of Pipelined Radix-4 FFT Architectureon FPGAs" Journal of Clean Energy Technologies, Vol. 2, No. 1, January 2014.
7. P.Verma, H. Kaur, M.Singh, M, B.Singh, "VHDL Implementation of FFT/IFFT Blocks for OFDM," In Proc. of Intern. Conf. on Advances in Recent Technologies in Communication and Computing, pp. 186-188, PI. 978-1-4244-5104-3, Kerala, 2009.
8. H.L.Lin, H.Lin, Y.C. Chen and R.C. Chang, "A Novel Pipelined Fast Fourier Transform Architecture for Double Rate OFDM Systems," IEEE workshop on signal processing systems design and implementation, pp. 7-11, ISBN 03-8504-7, Texas, USA, 2004
9. Manushree Bhardwaj1, Arun Gangwar2, Devendra Soni3, "A Review on OFDM: Concept, Scope & its Applications", IOSR Journal of Mechanical and Civil Engineering (IOSRJMCE)ISSN : 2278-1684 Volume 1, Issue 1 (May-June 2012), PP 07-11www.iosrjournals.org
10. Marwan A. Jaber, Daniel Massicotte and Youssef Achouri, "A Higher Radix FFT FPGA Implementation Suitable for OFDM Systems", Laboratory of Signal and System Integrations, 978-1-4577-1846-5/11/$26.00 ©2011 IEEE.
11. N. Amarnath Reddy, D. Srinivasa Rao and J. VenkataSuman, "Design and Simulation of FFT Processor Using Radix-4 Algorithm Using FPGA", International Journal of Advanced Science and Technology, Vol.61, (2013), pp.53-62, http://dx.doi.org/10.14257/ijast.2013.61.06.