International Journal of Research in Advent Technology, Vol.7, No.4S, April 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Design and Implementation of 16-Bit Radix-4 IFFT by Inverse Decimation in Time (IDIT) Using Verilog

B. Anil Kumar[1],M.Saipreethi [2], M.Yeshwanth Chowdhary[3], K. Jayashankar Goud [4]
Dept. of ECE [1],Dept. of ECE[2], Dept. of ECE[3], Dept. of ECE[4],MRIET[1], MRIET[2], MRIET[3], MRIET[4]
(Email: banilkmr301@gmail.com[1] , preethireddy14598@gmail.com[2] , rajyamyash.143@gmail.com[3], jayashankargoud19@gmail.com[4] *)*

Abstract- In the Inverse Decimation in Time (IDIT) computation, the butterfly theory plays a central role, since they allow the calculation of complex terms. The IDIT is the foremost algorithm used in the field of Digital Signal Processing. In this project, the different and dedicated structures for IDIT radix 4 FFT algorithm with 16-bytes are implemented, where the main goal is to minimize the number of arithmetic operators. This calculation, involving multiples of input data with appropriate co-efficients, the optimization of the butterfly can contribute for the reduction of the power consumption of the FFT architectures. Verilog was used as a description language. The proposed algorithms are used in radix- 4 butterfly in all stages. In IDIT, we firstly compute multiplier then adder. It is mostly achieved by simple scaling of twiddle factor using a special scaling factor. The efficient computation of IDIT is an important issue as it is used in almost all fields of engineering for signal processing. FFT can achieve less time delay, beat down the area complication and, also reach cost dominant execution with minimum grow up time.

Index Terms- Xilinx, Verilog, IFFT, Radix-4,IDIT

1. INTRODUCTION

Since the development of Fourier Transforms the implementation of DFT is wide as there are lot of complexities and performance issues, FFT is used with the reduction of number of complex additions and multiplications using butterfly structures.

To implement this Verilog is used and it helps in proper coding of the structures respectively, Verilog is a Hardware Description Language(HDL) , As it helps in describing a digital system like a network switch or electronic devices. The radix-4 IDIT IFFT divides an N-point discrete Fourier transform(IDFT) into two N/4-point IDFTs, then into 16 N/16 -point IDFTs,and so on. In the radix-2 DIT FFT, the DFT equation is expressed as the sum of two calculations. One calculation sum for the first half and one calculation sum for the second half of the input sequence. Similarly, the radix-4 IDIF fast Fourier transform (IFFT)expresses the IDFT equation as two summations, then divides it into two equations, each of which computes every two output sample.

2. LITERATURE SURVEY

Initially Radix-2 FFT is used and 16-point data sequence is considered.

But due to the implications that arise in complexity and computation time, the overall performance is delayed , to overcome this issue Radix-4 FFT is used in the next level of implementation.

Comparison between Radix-2&Radix-4

2.1.Radix-2

As we are using 16-point data, the number of stages are limited to four and the main factor for choosing FFT is its information size, the information size depends on the requirement of the application.

To find the number of stages mathematically the equation is

$$\mathbf{n=log^{N}{}_{2}}$$

N= number of samples

n= number of stages

$$\mathbf{n=log_2{}^{16}}$$
$$\mathbf{n=4log_2{}^{2}}$$

so, number of stages(n)=4.

2.2.Radix-4

In Radix -4, the number of stages are limited and reduced to two stages as the complexity is less when compared to Radix -2 ,the performance stastics are high. One Radix-4 butterfly structure contains four Radix-2 butterfly sturctures and it can store more efficiently which reduces the computation time.

$$n=\log_4^N$$
$$n=\log 416$$
$$n=2\log_4^4$$

so, number of stages (n)=2.

3.BLOCK DIAGRAM

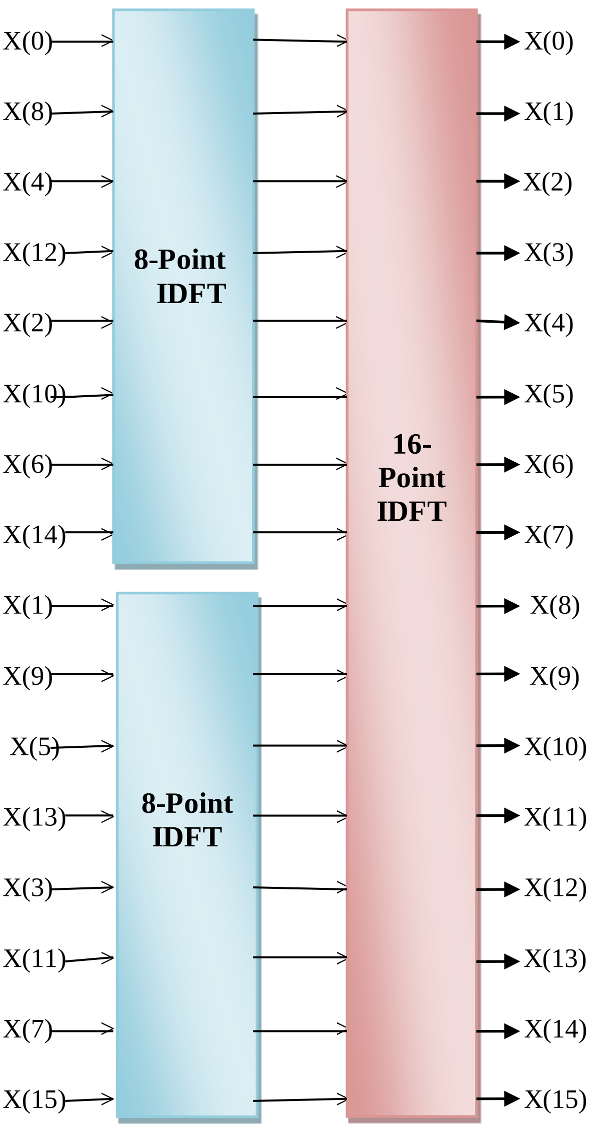


Figure1: Block diagram of Radix-4 structure

In the radix-4 FFT computation,for the odd indexed outputs, a radix-4 decomposition the computational efficiency is increased, because the four-point DFT has the largest multiplication free butterfly. This is ecause the radix-4 FFT is more efficient than the radix-2 FFT from the multiplication complexity point of view. Consequently, the DFT computation uses different radix FFT algorithms for odd and even indexed outputs. This reduces the number of complex multiplications and additions/subtractions.

4.DESCRIPTION

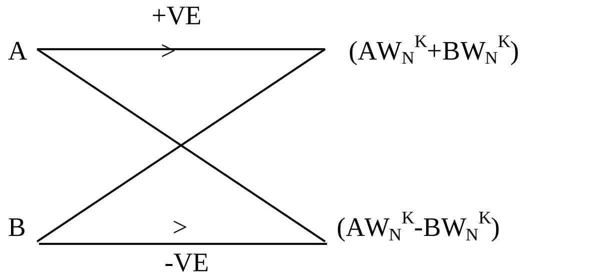
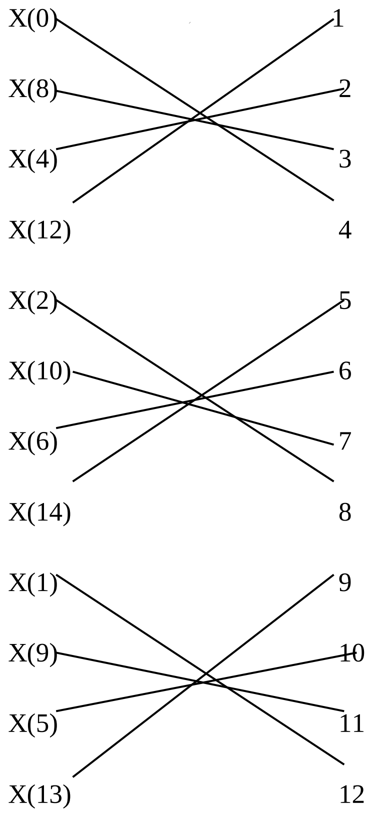


Figure2:flow graph of Radix-4 butterfly structure

BUTTERFLY STRUCTURE OF RADIX-4



E-ISSN: 2321-9637
Available online at www.ijrat.org

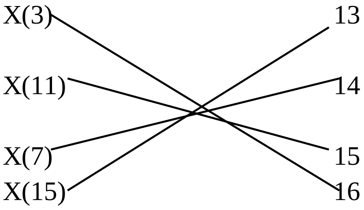


Figure3:STAGE-1 OF RADIX -4

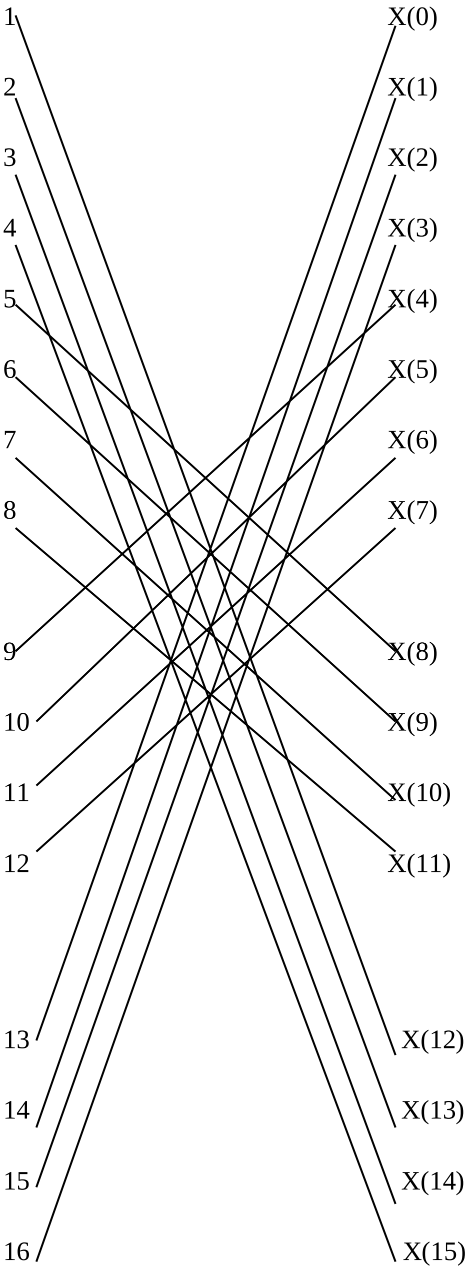


Figure4:STAGE-2 OF RADIX-4

We can find the signal flow graph(SFG) for DFT computation with IDIF computation is transposition of that with IDIT algorithmHence, many properties for IDIT and IDIF are same. For instance, the computation workload for the IDIT and IDIF are the same. The unscrambling processes are required for both IDIT and IDIF algorithms.However, there are clear differences between IDIF and IDIT algorithms,

Example: The position of twiddle factor multiplications.The IDIF algorithms have the twiddle factor multiplications after the DFT's and the IDIT algorithms have the twiddle factor multiplications before the DFT's.

And also in IDIF there is no need of applying the conjugate at the input side and bit reversal too, whereas in IDIT there is a process of applying the conjuate at the input side and the input should be in bit reversal state.

In a Radix-4 FFT algorithm, the addition and substraction operations are used for realizing the butterfly operations and the complex multiplications.Since substraction has the same complexity as addition, So we consider substraction as addition.The additions for the butterfly operations is the larger part of the addition complexity.

CALCULATION OF TWIDDLE FACTORS

Consider 16-point,then N=16

$\mathbf{W_N^K = W_{16}^K}$

$\mathbf{W_{16}^K = e^{-j(2\pi/16)K} = e^{-j(\pi/8)K}}$

If K=0, $\mathbf{w^0_{16} = e^{-j(0)} = 1}$

If K=1, $\mathbf{w^1_{16} = e^{-j(\pi/8)} = 0.92 - j0.38}$

If K=2, $\mathbf{w^2_{16} = e^{-j(2\pi/8)} = 0.707 - j0.707}$

If K=3, $\mathbf{w^3_{16} = e^{-j(3\pi/8)} = 0.38 - j0.92}$

If K=4, $\mathbf{w^4_{16} = e^{-j(4\pi/8)} = -j}$

If K=5, $\mathbf{w^5_{16} = e^{-j(5\pi/8)} = -0.38 - j0.92}$

If K=6, $\mathbf{w^6_{16} = e^{-j(6\pi/8)} = -0.707 - j0.707}$

If K=7, $\mathbf{w^7_{16} = e^{-j(7\pi/8)} = -0.92 - j0.38}$

If K=8, $\mathbf{w^8_{16} = e^{-j(8\pi/8)} = -1}$

If K=9, $\mathbf{w^9_{16} = e^{-j(9\pi/8)} = -0.92 + j0.38}$

If K=10, $\mathbf{w^{10}_{16} = e^{-j(10\pi/8)} = 0.707 + j0.707}$

If K=11, $\mathbf{w^{11}_{16} = e^{-j(11\pi/8)} = -0.38 + j0.92}$

If K=12, $\mathbf{w^{12}_{16} = e^{-j(12\pi/8)} = j}$

If K=13, $\mathbf{w^{13}_{16} = e^{-j(13\pi/8)} = 0.38 + j0.92}$

If K=14, $\mathbf{w^{14}_{16} = e^{-j(14\pi/8)} = 0.707 - j0.707}$

If K=15, $w^{15}_{16} = e^{-j(15\pi/8)} = 0.92 - j0.38$

A twiddle factor , in fast fourier transform(FFT) algorithms, is any of the trigonometric constant coefficients that are multiplied by the data in the course of the algorithm.
More efficiently, "twiddle factors" originally reffered to the root of unity complex multiplicative constants in the butterfly operations,used to recursively combine smaller Discrete Fourier Tranforms(DFT).
This remains the terms most common meaning, but it may also be sed for any data independent multiplicative constant in an IDIT-FFT.

5. TABLE

COMPARISON OF FFT WITH DFT FOR N=16

S.no	Number of complex additions	Number of complex multiplications
1.	DFT $N(N-1)=16(16-1)$ $=240$	FFT $N^2=16^2=256$
2.	DFT $NLog_4{}^N=16log_4{}^{16}$ $=32$	FFT $N/2Log_4{}^N=16/2log_4{}^{16}$ $=16$

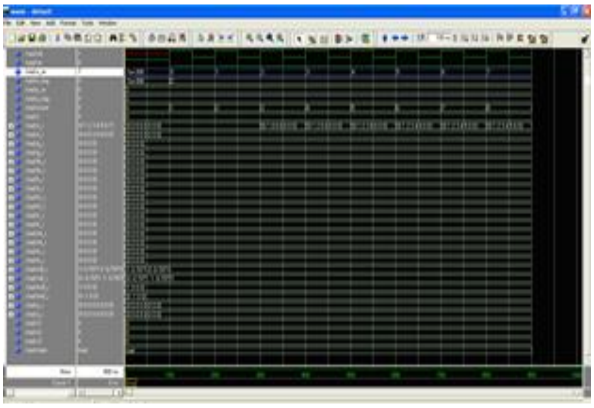
6.RESULTS

Input of stage1= X(K) = (**2,0,2,0,0.293+0.707i,0.707-0.293i,0.707+1.707i,1.707-0.707i,4014+2.76i,0.639-0.98i,4.88+3.24i,1.86-2.76i,0.08-0.38i,1.087+0.213i,-0.327+1.627i,1.92+0.38i**)

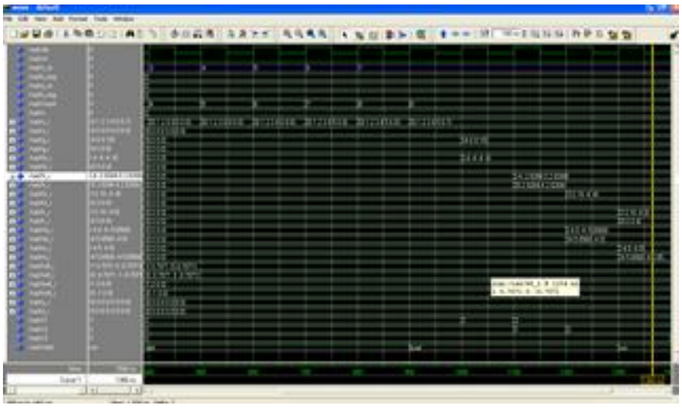
Output of stage1=**{2,0,-2,2,2,1.414+1.414i,1.414i, -1.414+10414i,6,5.519+2.26i,-4.2414.22i,2.28+5.52i, 2,0.76+1.84i,1.414+1.414i,-1.84-0.76i**}
Overall ouput Result of **IDIT** for **N=16** samples is Obtained as **x(n) = {1,1,1,1,1,1,1,1,2,2,2,2,0,0,0,0}**

The simulation of this whole project has been done using the Xilinx of version 16.2. Xilinx is a simulation tool for programming {VLSI} {ASIC}s, {FPGA}s, {CPLD}s, and {SoC}s. Xilinx provides a comprehensive simulation and debug environment for complex ASIC and FPGA designs. Support is provided for multiple languages including Verilog, SystemVerilog, VERILOG and SystemC.

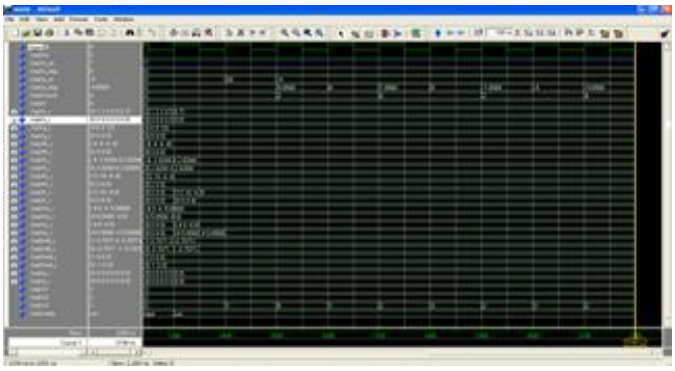
Simulation Result Obtained For Start State



Simulation Results Obtained For Load State



Simulation Results Obtained For Run State



CONCLUSION AND FUTURE SCOPE

CONCLUSION

This project describes the efficient use of VERILOG code for the implementation of radix-4 based FFT architecture and the wave form result of the various stages has been obtained successfully. Compared to previous method it requires only 16 clock cycles for performing the butterfly process and also, the accuracy in obtained results has been increased with the help of efficient coding in VERILOG. The accuracy in results depends upon the equations obtained from the butterfly diagram and then on the correct drawing of scheduling diagrams based on these equations

FUTURE SCOPE : The future scopes of this project are to implement the proposed FFT architecture using Field-Programmable Gate Arrays (FPGAs) and also obtain the Discrete In time (DIT) algorithm of FFT.
The FFT (Fast Fourier Transform) processor plays a critical part in speed and power consumption of the Orthogonal Frequency Division Multiplexing (OFDM) communication system. Thus the FFT block can be implemented in OFDM with RADIX-8.

REFERENCES

[1].Bringham, E. O.The Fast Fourier Transform.Prentice-Hall, Englewood Cliffs, NJ, 1974.
[2]. 9. V. J. Alberto, R. R. Jesus, and O. Alejandro, 2005. VERILOG core for 16-point radix-4 FFT computation: Proc. of the 2005 International Conference on Reconfigurable Computing and FPGAs (ReConFig 2005).
[3]. W. Hussain, F. Garzia, J. Nurmi, “Evaluation of Radix-4 FFT Processing on a Reconfigurable Platform”, 13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems, 2010.
[4]. Douglas L. Jones “Decimation-in-Time (DIT) Radix-4 FFT Algorithms” Connexions module: m12016 .
[5]. A. Saidi, “Decimation-in-Time-Frequency FFT Algorithm,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. III: 453-456, April 19-22 1994.
[6]. Lihong Jia, Yonghong GAO, Jouni Isoaho, and Hannu Tenhunen, "A New VLSI-Oriented FFT Algorithm and Implementation", IEEE ASIC Conf., Sep.1998, pp. 337-341.
[7]. John G.Proakis and Dimitris G.Manolakis, “Digital Signal Processing: Principles, Algorithms, and Applications”, Fourth Edition, Pearson Education, Inc., 2007.
[8]. James W.Cooley, Peter A.W.Lewis and Peter D.Welch, “The Fast Fourier Transform and its Applications” IEEE Transactions as Education, Vol. 12. No. 1, March,1969.
[9]. S.Palani and D.Kalaiyarasi, “Digital Signal Processing”, First Edition, Ane Books Pvt. Ltd, New Delhi, 2010.
[10]. 9. Takahashi D(2001),“An Extended Radix-4 FFT Algorithm ”,IEEE Signal Processing Letters,Vol.8,No.5, pp.145-147.

Mr. **B. ANIL KUMAR** , M.Tech working as Assistant professor ,in the **Department of Electronics and communication , Mallareddy Institute of Engineering and Technology, Hyderabad**. He studied **B.Tech** in **Electronics and Communications Engineering** from JNTU college of Engineering, JNTUH, Hyderabad, Telangana and **M.Tech** in **VLSI SYSTEM DESIGN From Aurora college of engineering**, JNTUH, Hyderabad, Telangana.

M.SAIPREETHI is studying B.Tech in (Electronics & Communication Engineering) at Mallareddy Institute of Engineering & Technology (MRIET), Hyderabad. Telangana.

M.YESHWANTH CHOWDHARY is studying B.Tech in (Electronics & Communication Engineering) at Mallareddy Institute of Engineering & Technology (MRIET), Hyderabad. Telangana.

K.JAYASHANKAR GOUD is studying B.Tech in (Electronics & Communication Engineering) at Mallareddy Institute of Engineering & Technology (MRIET), Hyderabad. Telangana.