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

Implementation of 64-Bit Radix-8 FFT by Decimation in Frequency (DIF) Using Verilog

B. Anil Kumar[1], K.Rujwala[2], V.Satyaarjun[3],P.Parvathiswara Rao[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] ,rujwalakatakam797@gmail.com[2], arjunchowdary823@gmail.com[3], pogiriparvathiswararao@gmail.com[4])

Abstract-Always technical designers choice includes algorithms, flowcharts, programming etc and the end users requires given input and application output. Based upon this view this paper focus on the advancement of Fast Fourier Transform(FFT) by doing design and observing the performance analysis of 64 point FFT, using Radix 8 algorithm. The algorithm is developed by Decimation In Frequency(DIF) of FFT,using VHDL as design entity and synthesis ar performed in Xilinx. In this architecture the number of stages are reduced to 75%.

Index Terms-FFT, DIF, Verilog,,XILINX..

1. INTRODUCTION

A fast Fourier transform (FFT) is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. There are many distinct FFT algorithms involving a wide range of mathematics, from simple complex number arithmetic to group theory and number theory.
A DFT decomposes a sequence of values into components of different frequencies. This operation is useful in many fields (see discrete Fourier transform for properties and applications of the transform) but 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 a DFT of N points in the naive way, using the definition, takes O(N2) arithmetical operations, while an FFT can compute the same result in only O(N log N) operations. The difference in speed can be substantial, 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 many DFT based algorithms 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.
The most well known FFT algorithms depend upon the factorization of N, but there are FFTs with O(N log N) complexity for all N, even for prime N. Many FFT algorithms only depend on the fact that $e^{-2\prod i/N}$ is anth primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number theoretic transforms. Since the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a 1/N factor, any FFT algorithm can easily be adapted for it.

2. LITERATURE SURVEY

J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series.
Mathematics of Computation, 1965. A fast algorithm for computing the Discrete Fourier Transform . (Re)discovered by Cooley & Tukey in 19651 and widely adopted thereafter has a long and fascinating history.Explained the design of pipelined structure from Radix-8 FFT with DIF algorithm using efficient butterfly structure. The different and dedicated structures for the 64 bit-width pipelined radix-8 DIF butterfly structure are implemented. The main goal of this paper is to minimize the number of real multipliers of the architectures. This is done by varying the structure of the complex multipliers and applying them into the butterflies. These structures are widely used in fast and low power multiplier architectures. In pipelined Radix-8 FFT structures have been developed with the help of Feed forward structures. Feed forward structure provides 26ns for performing 8-point FFT.

3. EQUATIONS

The main reason for going with Fast Fourier Transform is to reduce the complexity and make mathematical calculations easier compared to that of DFT.The formulae what we use in DFT fails for higher complex stages where the calculations become unperformable, So here in FFT we go for higher complex stages with reduced number of calculations and complexity. The main difference between the calculations performed in DFT and FFT is logarithm.
In this paper we are trying to implement the FFT using DIF with log base 8i.e.,Radix 8 structure implementing 64 bits of data.

General Equations

$$X(K)=\sum_{N=0}^{M-1} x(n)WN \quad \text{---------- (1)}$$

$$X(K)=\sum_{n=0}^{N/2-1} x(n)Wnkn+\sum_{n=N/2}^{N-1} x(n)Wnkn \quad \text{-----(2)}$$

4.**BLOCK DIAGRAM**

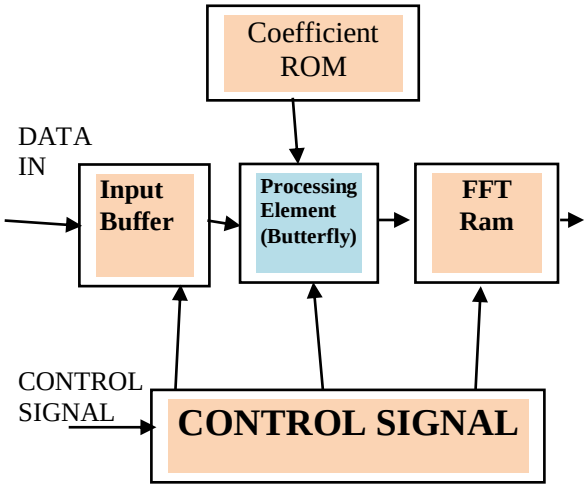


Figure 1

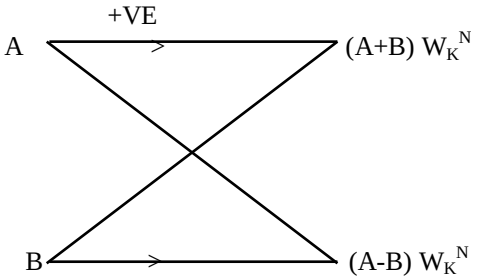


Figure 2 Butterfly Structure of RADIX-8

In the radix 8 FFT first we align the data and it is given to the input buffer which controls the data and performs the distribution property which allows data to perform their operations in a sequence.The next block refers to the processing element i.e.,Radix 8 Butterfly structure,Where the number of stages reduced to 75%.The next and main block of the structure is control signal which performs data control and processing control.The next block COEF ROM allows the processing element t perform the calculations and store the result temporarily for future years.

DESCRIPTION

Compare to that of radix 2 and radix 4 here in this paper we perform radix 8 operations by using the twiddle factors. The radix 8 butterfly structure helps us to carry out the complex calculations in the easier way.

In the butterfly structure of Radix 8 point FFT we have 2 stages.The number of stages are obtained by reducing the complexity using the FFT.The number of stages are obtained as follows below.

CALCULATIONS

To find the number of stages mathematically the equation is

$$n=\log_N 8$$

N= number of samples

n= number of stages

$$n=\log_8 64$$

$$n=2\log_8 8$$

so,number of stages(n)=2.

S.no	Number of complex additions	Number of complex multiplication
1.	DFT N(N-1)=64(64-1) =4095	FFT N2=642=4096
2.	DFT NLog8N=64log864 =128	FFT N/2Log8N=64/2log864 =64

From the above table we can conclude that ,In DFT for the higher stages the complexity increases where in FFT the complexity is reduced .

5. PROCESS OF DECIMATION

First step of decimation is splitting a sequence in a smaller sequences.A sequence of 64 Bit can be splitted in 8 sequences of 4 blocks.Here on the first stage carries out the butterfly operation by applying the twiddle factor.

In the second stage the output from the 64 Bit FFT is split into sequence of 8 equal parts.and the initial 32Bits are performed by addition and twiddle factor multiplication and then followed 32 Bits perform subtraction and multiplied with twiddle factor.

Twiddle factors :

$$W_N^K = e^{-j(2\Pi/N)K}$$

For 64 points,the twiddle factor is represented as,

N=64,

$$W_{64}^{0} = e^{-j(2\Pi/64)0} = 1$$

$$W_{64}^{1} = e^{-j(2\Pi/64)1} = 0.995 - j0.098$$

$$W_{64}^{2} = e^{-j(2\Pi/64)2} = 0.980 - j0.195$$

$W_{64}^{3} = e^{-j(2\Pi/64)3} = 0.956\text{-}j0.290$
$W_{64}^{4} = e^{-j(2\Pi/64)4} = 0.923\text{-}j0.382$
$W_{64}^{5} = e^{-j(2\Pi/64)5} = 0.881\text{-}j0.471$
$W_{64}^{6} = e^{-j(2\Pi/64)6} = 0.831\text{-}j0.555$
$W_{64}^{7} = e^{-j(2\Pi/64)7} = 0.773\text{-}j0.634$
$W_{64}^{8} = e^{-j(2\Pi/64)8} = 0.707\text{-}j0.707$
$W_{64}^{9} = e^{-j(2\Pi/64)9} = 0.634\text{-}j0.773$
$W_{64}^{10} = e^{-j(2\Pi/64)10} = 0.555\text{-}j0.831$
$W_{64}^{11} = e^{-j(2\Pi/64)11} = 0.471\text{-}j0.881$
$W_{64}^{12} = e^{-j(2\Pi/64)12} = 0.382\text{-}j0.923$
$W_{64}^{13} = e^{-j(2\Pi/64)13} = 0.290\text{-}j0.956$
$W_{64}^{14} = e^{-j(2\Pi/64)14} = 0.195\text{-}j0.980$
$W_{64}^{15} = e^{-j(2\Pi/64)15} = 0.098\text{-}j0.995$
$W_{64}^{16} = e^{-j(2\Pi/64)16} = -j$
$W_{64}^{17} = e^{-j(2\Pi/64)17} = -0.098\text{-}j0.995$
$W_{64}^{18} = e^{-j(2\Pi/64)18} = -0.915\text{-}j0.980$
$W_{64}^{19} = e^{-j(2\Pi/64)19} = -0.290\text{-}j0.956$
$W_{64}^{20} = e^{-j(2\Pi/64)20} = -0.382\text{-}j0.923$
$W_{64}^{21} = e^{-j(2\Pi/64)21} = -0.471\text{-}j0.881$
$W_{64}^{22} = e^{-j(2\Pi/64)22} = -0.555\text{-}j0.831$
$W_{64}^{23} = e^{-j(2\Pi/64)23} = -0.634\text{-}j0.773$
$W_{64}^{24} = e^{-j(2\Pi/64)24} = -0.707\text{-}j0.707$
$W_{64}^{25} = e^{-j(2\Pi/64)25} = -0.773\text{-}j0.634$
$W_{64}^{26} = e^{-j(2\Pi/64)26} = -0.831\text{-}j0.555$
$W_{64}^{27} = e^{-j(2\Pi/64)27} = -0.881\text{-}j0.471$
$W_{64}^{28} = e^{-j(2\Pi/64)28} = -0.923\text{-}j0.382$
$W_{64}^{29} = e^{-j(2\Pi/64)29} = -0.956\text{-}j0.290$
$W_{64}^{30} = e^{-j(2\Pi/64)30} = -0.980\text{-}j0.195$
$W_{64}^{31} = e^{-j(2\Pi/64)31} = -0.995\text{-}j0.098$
$W_{64}^{32} = e^{-j(2\Pi/64)32} = -1$
$W_{64}^{33} = e^{-j(2\Pi/64)33} = -0.995+j0.098$
$W_{64}^{34} = e^{-j(2\Pi/64)34} = -0.980+j0.195$
$W_{64}^{35} = e^{-j(2\Pi/64)35} = -0.956+j0.290$
$W_{64}^{36} = e^{-j(2\Pi/64)36} = -0.923+j0.382$
$W_{64}^{37} = e^{-j(2\Pi/64)37} = -0.881+j0.471$
$W_{64}^{38} = e^{-j(2\Pi/64)38} = -0.831+j0.555$
$W_{64}^{39} = e^{-j(2\Pi/64)39} = -0.773+j0.634$
$W_{64}^{40} = e^{-j(2\Pi/64)40} = -0.707+j0.707$
$W_{64}^{41} = e^{-j(2\Pi/64)41} = -0.634+j0.773$
$W_{64}^{42} = e^{-j(2\Pi/64)42} = -0.555+j0.831$
$W_{64}^{43} = e^{-j(2\Pi/64)43} = -0.471+j0.881$
$W_{64}^{44} = e^{-j(2\Pi/64)44} = -0.382+j0.923$
$W_{64}^{45} = e^{-j(2\Pi/64)45} = -0.290+j0.956$
$W_{64}^{46} = e^{-j(2\Pi/64)46} = -0.195+j0.980$
$W_{64}^{47} = e^{-j(2\Pi/64)47} = -0.098+j0.995$
$W_{64}^{48} = e^{-j(2\Pi/64)48} = j$
$W_{64}^{49} = e^{-j(2\Pi/64)49} = 0.098+j0.995$
$W_{64}^{50} = e^{-j(2\Pi/64)50} = 0.195+j0.980$
$W_{64}^{51} = e^{-j(2\Pi/64)51} = 0.290+j0.956$
$W_{64}^{52} = e^{-j(2\Pi/64)52} = 0.382+j0.923$
$W_{64}^{53} = e^{-j(2\Pi/64)53} = 0.471+j0.881$
$W_{64}^{54} = e^{-j(2\Pi/64)54} = 0.555+j0.831$
$W_{64}^{55} = e^{-j(2\Pi/64)55} = 0.634+j0.773$
$W_{64}^{56} = e^{-j(2\Pi/64)56} = 0.707+j0.707$
$W_{64}^{57} = e^{-j(2\Pi/64)57} = 0.773+j0.634$
$W_{64}^{58} = e^{-j(2\Pi/64)58} = 0.831+j0.555$
$W_{64}^{59} = e^{-j(2\Pi/64)59} = 0.881+j0.471$
$W_{64}^{60} = e^{-j(2\Pi/64)60} = 0.923+j0.382$
$W_{64}^{61} = e^{-j(2\Pi/64)61} = 0.956+j0.290$
$W_{64}^{62} = e^{-j(2\Pi/64)62} = 0.980+j0.195$
$W_{64}^{63} = e^{-j(2\Pi/64)63} = 0.995+j0.098$

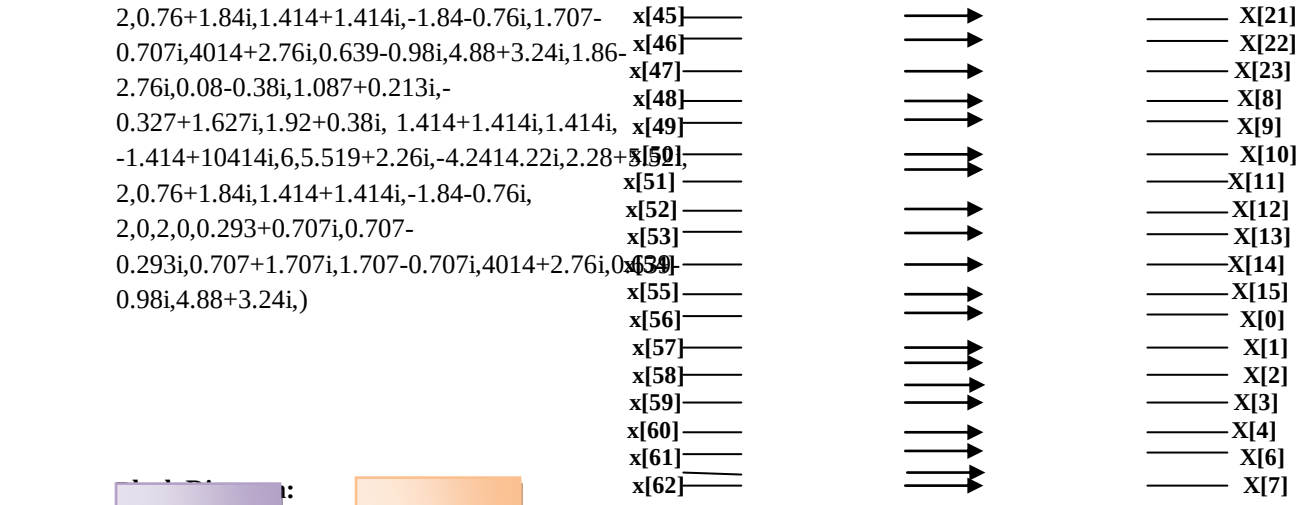
Inputs at stage1:-

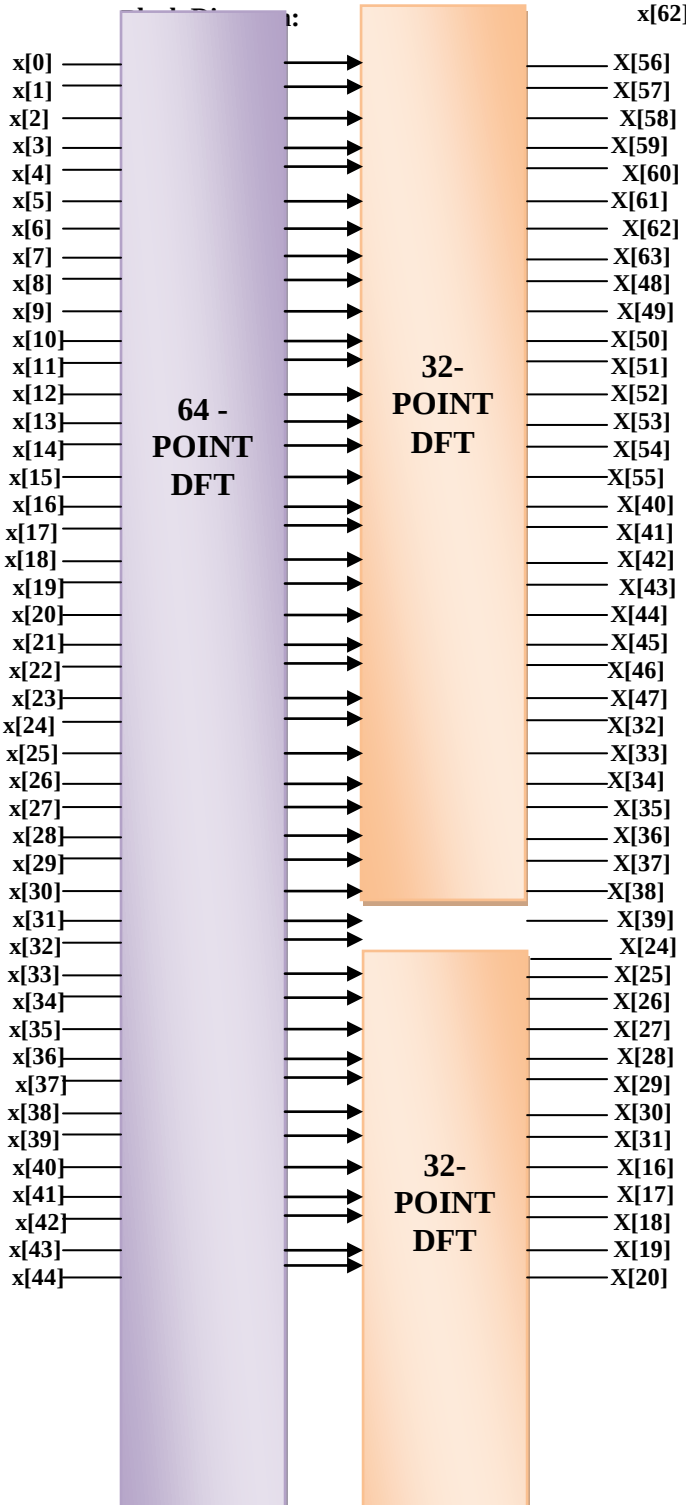
x(n)=(1,1,1,1,1,1,1,1,2,2,2,2,0,0,0,0,
1,1,1,1,1,1,1,1,2,2,2,2,0,0,0,0,
1,1,1,1,1,1,1,1,2,2,2,2,0,0,0,0,
1,1,1,1,1,1,1,1,2,2,2,2,0,0,0,0)

Outputs of stage :- 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, 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,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, 1.414+1.414i,1.414i, -1.414+10414i,6,5.519+2.26i,-4.2414.22i,2.28+3.24i, 2,0.76+1.84i,1.414+1.414i,-1.84-0.76i, 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,)





6. SOFTWARE SIMULATION AND RESULTS

The proposed FFT block of signal length 64 is been simulated and synthesized using the Xilinx Design Suite 16.1. The RTL block thus obtained for the decimation in time domain radix -8 Fast Fourier transform algorithm is shown The RTL view of the butterfly structure obtained after the simulation of the 64-point FFT block, Decimation in time domain is shown next and also the internal architecture of the butterfly block is shown.

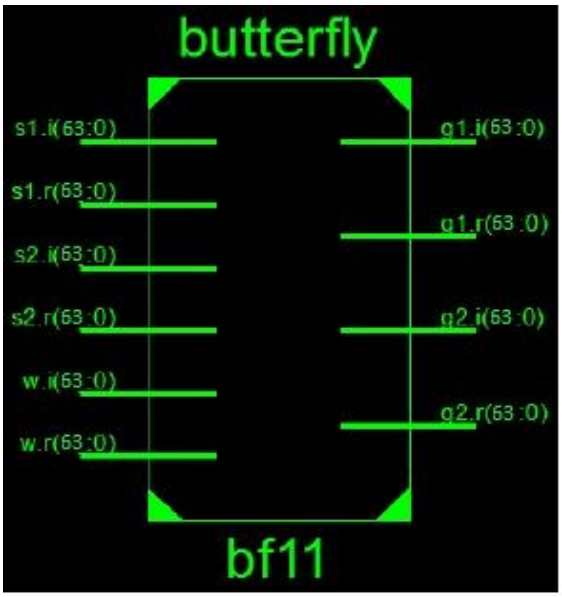


Figure3: RTL View Of A Butterfly Component Used In 64-Point FFT

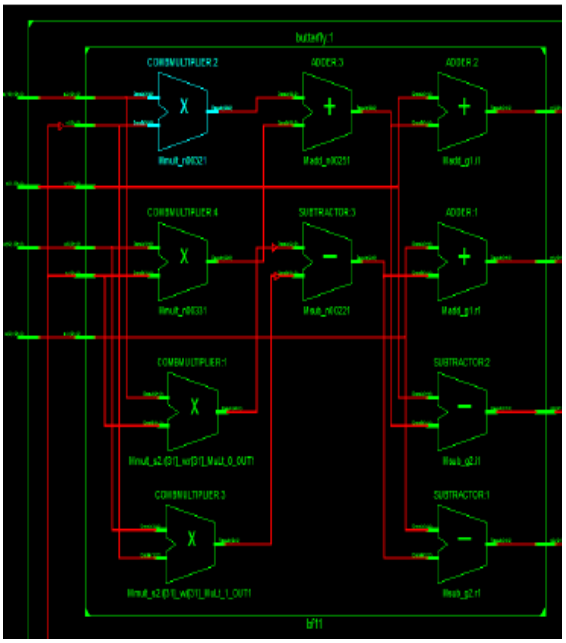


Figure 4: Internal Architecture of The Butterfly Component

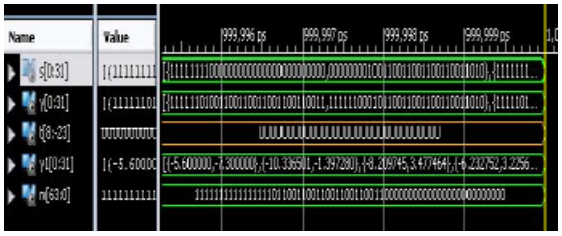
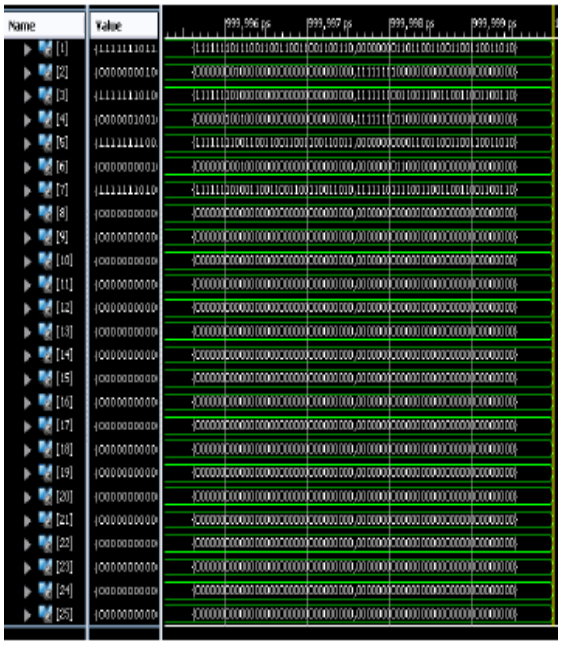
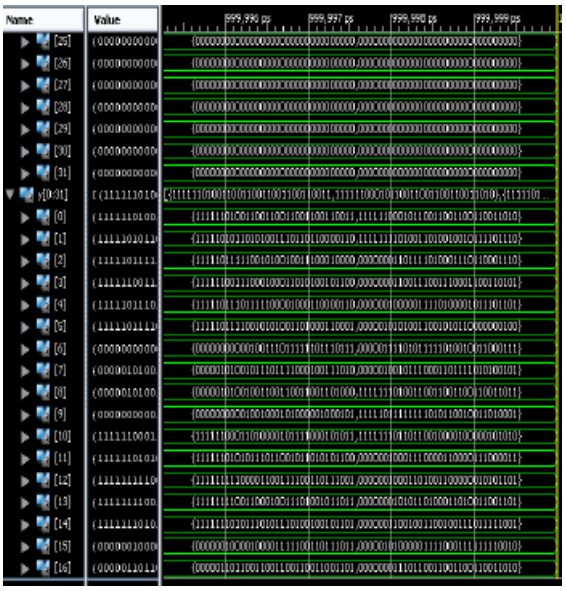


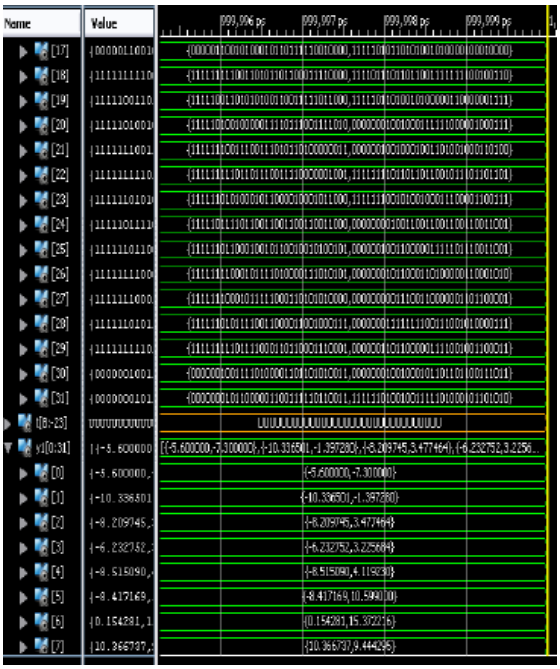
Figure5: Simulation results of the 64-point FFT

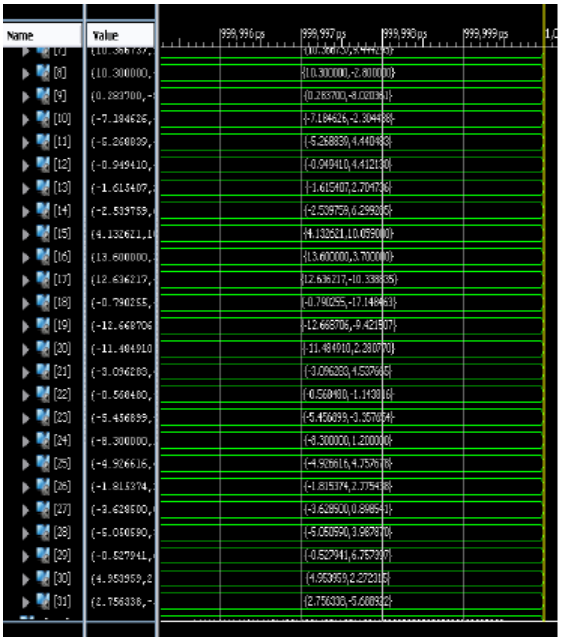




Acknowledgments

This section should come before the References. Funding information may also be included here.





CONCLUSION

This project describes the efficient use of VLSI for the implementation of radix 8 based FFT pipelined architecture and the wave form result of the various stages has been obtained successfully. Compared to previous method 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. 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 Frequency (DIF) algorithm of FFT.

REFERENCES

[1] Bharat, K.; Broder, A. (1998): A technique for measuring the relative size and overlap of public Web search engines. Computer Networks, **30**(1–7), pp. 107–117.

[2] Broder, A.; Kumar, R.; Maghoul, F.; Raghavan, P.; Rajagopalan, S.; Stata, R.; Tomkins, A.; Wiener, J. (2000): Graph structure in the Web. Computer Networks, **33**(1–6), pp. 309–320.

[3] Chakrabarti, S. (2000): Data mining for hypertext: A tutorial survey. SIGKDD explorations, **1**(2), pp. 1–11.

[4] Saad Bouguezel, M. Omair Ahmad,IMPROVED RADIX-4 AND RADIX-8 FFT ALGORITHMSIEEE. Department of Electrical andComputer Engineering Concordia University 1455 de Maisonneuve Blvd.West Montreal, P.Q., Canada.

[5] Ali Saidi , DECIMATION-IN-TIME-FREQUENCY FFTALGORITHM Motorola Applied Research, Paging and Wireless DataGroup Boynton Beach.

[6] Rizalafande Che Ismail and Razaidi Hussin High Performance ComplexNumber Multiplier Using Booth-Wallace Algorithm School of Microelectronic Engineering Kolej University Kejuruteraan UtaraMalaysia.

[7] J.G.Proakis and D.G.Manolakis, Digital Signal Processing,Principles,algorithms and applications Prentice Hall India Publication.

[8] J.A.Hidalgo,V.Moreno-Vergara,O.Oballe, A Radix-8 Multiplier UnitDesign For Specfic Purpose,Dept of the Electronica ,E.T.S.I.Industriales

[9] C. S. Burrus and T. W. Parks,DFT/FFT and Convolution Algorithms,New York,NY : John Wiley,1985.

[10] A. Saidi , Generalized FFT algorithm,Proc. ICC,pp.227-231,May 1993.[8]

[11] Robert Polge and Brooks Lawence, Comparion of New Multiple RadixFast fourier Number Theoetic Transform with FFT Algorithm in Termsof Performance and Hardware Cost, ECC Department, University of Alabama Huntsville, AL35899,IEEE : 744-749.

(A.1) 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.

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

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

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