Design and Implementation of Radix-4 IFFT by Inverse Decimation in Frequency by using Verilog

B. Anil kumar[1], R. Ravali[2], T. Teja Kumar[3],K. Sheshasai[4]

[1] *Assistant Professor, Dept of ECE, Malla Reddy Institute Of Engineering And Technology, Hyd., TS, India.*

[2]*B.Tech Student, Dept of ECE, Malla Reddy Institute Of Engineering And Technology, Hyd., TS, India*

[3]*B.Tech Student, Dept of ECE, Malla Reddy Institute Of Engineering And Technology, Hyd., TS, India*

[4]*B.Tech Student, Dept of ECE, Malla Reddy Institute Of Engineering And Technology, Hyd., TS, India*

Abstract-The Fast Fourier transform (FFT) is one of the rudimentary operations in the fields of DSP and DIP. The FFT estimates the spectral content (the harmonic content) of a time-domain sequence of digital signal samples. Ever since its inception in the mid 1960'S, the fast Fourier transform has revolutionized the field of real –time digital signal processing.The IFFT is a process to convert frequency-domain samples back to time-domain samples. Many algorithms were proposed, following the essential idea of decimation, either in the frequency or in the time domain. Infact, there has been no efficient IDIF algorithm for IFFT until now. This paper concentrates on the development of IFFT based on IDIF domain, radix-4 algorithm, by using Verilog as a design entity and the synthesis is done by Xilinx tool. The synthesis shows that the computation for calculating the 16-point IFFT is efficient in terms of speed, reduces the computational complexity and memory requirement.

Index Terms-radix-4, IDIF, IFFT,Xilinx,Verilog.

I. INTRODUCTION

Fast Fourier Transform (FFT) is a technique usually used to compute technique Discrete Fourier Transform (DFT). However, Discrete Fourier Transform (DFT) has certain drawbacks in computing. So, in order to overcome the drawbacks of DFT, Cooley and Tukey instigated the Fast Fourier Transform. The technique behind the efficient computation of DFTs in FFT is based on divide and conquer tactic. FFT algorithm can be implemented mainly in two methodologies i.e., one of them includes decimation in time and other includes decimation in frequency. FFT is used to speed up the computations of DFT and diminishes the computation time requires to compute the DFT and ameliorate the performance by factor 100 or more.

II. LITERATURE SURVEY

2.1 RADIX-2:

FFT can be mostly designed in radix-2, radix-4 and radix-8 format. Radx-2 is the basic format used to compute the DFTs in which the computation are done in four stages for data samples of N=16. idea of algorithms is to divide the N-point FFT into smaller ones until only two-point FFT is obtained.

2.2 RADIX-4:

In order to ameliorate the performance and increase the efficiency radix-4 is elucidated, where the computations are made only in two stages for N=16 data samples. The main idea of this algorithms is to divide the N-point FFT into smaller ones until only two-point FFT is obtained. Higher radix algorithms such as radix-4, radix-8, etc. can be employed to diminish the complex multiplications but the butterfly structure becomes more multiplex when compared to the radix-2 butterfly structure. Mostly the FFT computations are implemented in forward process i.e., FFT-DIF, till now there has been no efficient algorithm in reverse process i.e., IDIF. This paper concentrates on implementing the FFT algorithm in IDIF which is a reverse process in implementing FFT. This configuration s modeled on hardware description language VERILOG to physically realize the proposed butterfly. The size of FFTs under test ranges from 8 points, 16 points, 32points to 64 points. The result of synthesis tool and the timing analysis using the Xilinx simulator indicates a maximum operating frequency of 450MHz. It exhibited lower hardware complexity as compared to the radix-2. It achieved lower latency compared to the original radix-2 in DIF-FFT architecture with reasonable increase in hardware complexity.

III. BLOCK DIAGRAM

The block diagram of a radix-4 algorithm consists of 16 inputs and 16 outputs(see figure1).The FFT length i.e., data samples is 4^M, where M is represents as number of stages in radix-4. So the number of stages n radix -4 is 2 i.e., 4^2 where M=2.A radix-4 IDIF FFT has 16 N/16 samples in irst stage and has four N/4 samples in second stage. DIF-FFT is the forward process while IDIF-FFT is the reverse process. In the initial stage of radix-4 IDIF-FFT the conjugate is applied to input and output is represented in bit reversal. In the radix-2 IDIF 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 DIF fast Fourier transform (FFT) expresses the DFT equation as four

summations, then divides it into four equations, each of which computes every fourth output sample.

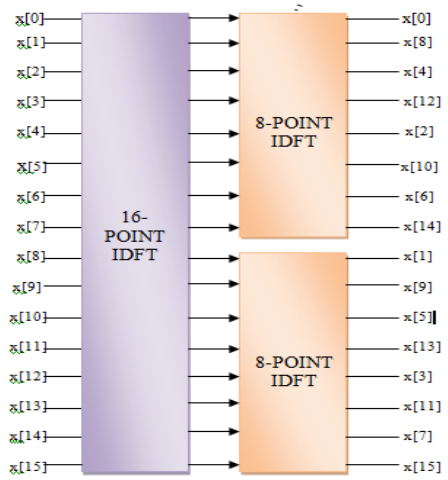


Figure 1: Block Diagram of Radix-4 IDIF.

IV. DESCRIPTION

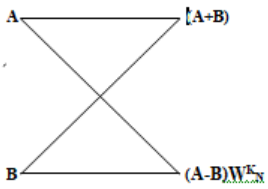


Figure 2: Basic Butterfly Structure of DIF-FFT.

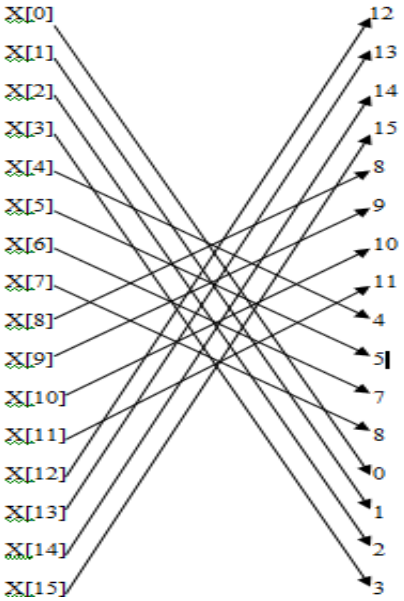


Figure 1. Second Stage of radix-4 IDIF-FFT

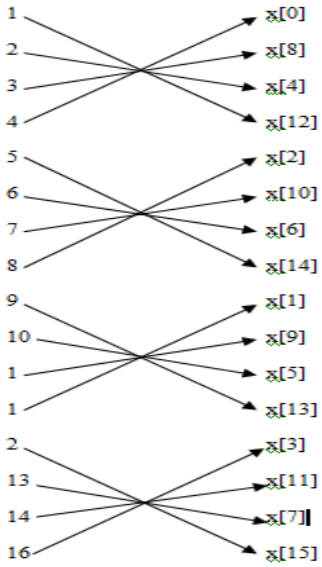


Figure 2

Number of Points, N	Complex Multiplications in Direct computations, N^2	Complex Multiplication in FFT/IFFT Algorithm, $(N/2)\log_4 N$	Speed improvement Factor
4	16	2	4.0
16	256	16	5.3
64	4096	96	8.0
256	65536	512	12.8
1024	1048576	2560	21.3
4096	16777216	12288	36.6

Table 1

4. TABLE 2

V. COMPARISON OF FFT WITH DFT

FOR DATA SAMPLES N=16

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

CALCULATIONS OF TWIDDLE FACTOR

Consider 16-point,then N=16

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The architectural design consist of data inputs, control unit, register, twiddle factor and the data output. The register may be of the array of four or eight variable in the type of real. The FFT implementation in VERILOG consists of two states such as start, load and run. The initial state is start where the eight input of the FFT are given through single clock cycles. The serial input of the input variable is declared as real. The input data is serially sent to the register memory and each input is given through a clock cycle and is used as an array in further stages through the internal register. The second state is load where the butterfly process is carried out. The butterfly process requires two clock cycles for completing a single stage of the butter fly stage. Totally it requires six clock cycles to complete the butter fly process.

In the outputs of FFT are obtained by one by one. The output will also requires a clock cycle and obtained by starting from x (0) up to x (15). As the output consist of real values, two registers are used for declare the values. The controlling unit carried in the FFT process is by clock and reset input in the program. The clock input is used to drive the pipeline stage the FFT architecture and reset is carried for resetting the initial state. The twiddle factor value is fixed and it defined inside the code itself.

VI. RESULTS

Input of stage1= X(K) =
(2, -0.23-j0.707, 4+j2.76, 0.08-j0.38, 0, -0.707+j0.293, 0.64-j0.98, 1.08+j0.21, 2, -0.707-j1.707,4.88+j3.26, -0.32+j1.62, 0, -1.707+j0.707, 1.86-j2.76, 1.92+j0.38,)

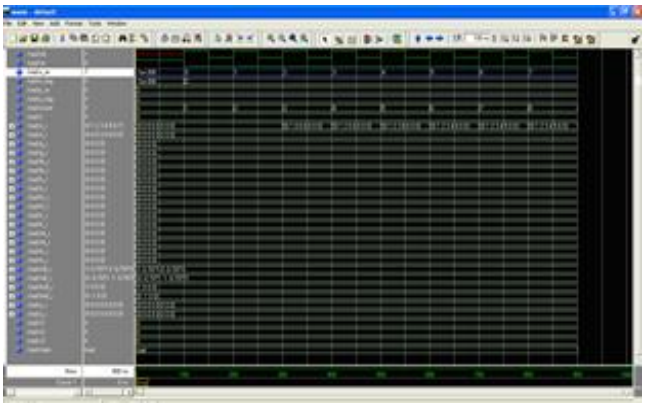
Output of stage1={2, -2, 6, 2, 0, 2, -1.83-j2.76, 5.51-j2.29, 1.97-j3.8, -2, 1.414+j4.24, -1.98+j7.07, 2, -0.76-j1.83, 2.29-j5.51, 1.98}

Overall ouput Result of IDIT for N=16 samples is

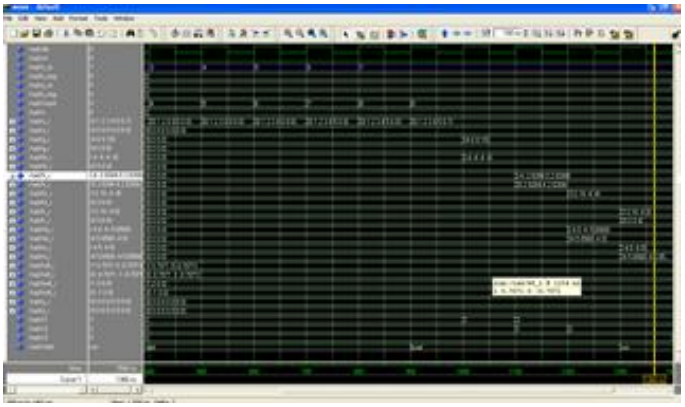
Obtained as x(n) = {1,1,1,1,1,1,1,1,2,2,2,2,0,0,0,0}

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.

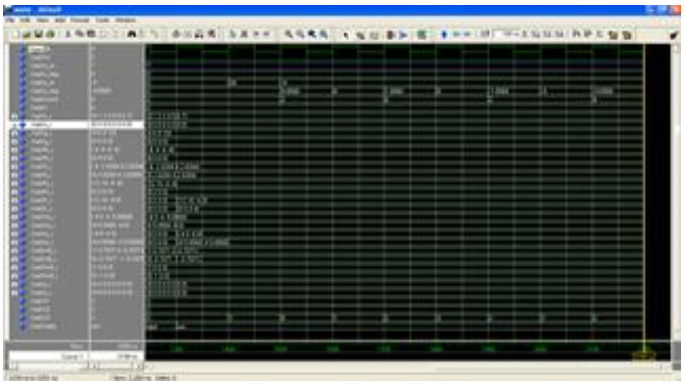
Simulation Result For Start State



Simulation Results For State



Simulation Results Obtained For overall State



VII. 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 Frequency (DIF) 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.

VIII. REFERENCES

[1]. Bouguezel S, Ahmad M O and Swamy M NS(2004), "Improved Radix-4 and Radix-8FFT Algorithms", IEEETransactionson Digital Object Identifier, Vol. 3,May, pp.561564.

[2].https://cse.google.com/cse?cx=partner-pub 8036109189802438%3A7790813904&ie=UTF8&q=16+point+radix+4+fft+algorithm+using+Verilog&sa=Search&siteurl=yourtv.link%2F#gsc.tab=0&gsc.q=16%20point%20radix%204%20fft%20algorithm%20using%20Verilog&gsc.page=1

[3]. 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.

[4]. J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series, Math.Computat" vol. 19, pp. 297-301, 1965.

[5]. S. Xin, Z. Tiejun, and H. Chaohuan, 2004. A Highperformance power-efficient structure of FFT (Fast Fourier Transform) processor: International Conference of Signal Processing: 555-558.

[6]. J G. Proakis and D. G. Manolakis, 1996. Digital Signal Processing Principles, Algorithm, and Applications. Prentice Hall, pp: 448-475.

[7]. K.Harikrishna 1, T. Rama Rao 2, Valadimir A. Labay: 'An Efficient FFT
Architecture for OFDM Communication Systems'. in Conference on Convergent Technologies for Asia-Pacific Region (TENCON '06), Vol. 1, pp. 95-99, 2006.

[8]. C. Gonzalez-Concejero, V. Rodellar, : 'An FFT/IFFT design versus Altera and Xilinx cores '. in conference on Reconfigurable Computing and FPGAs, 2008. ReConFig '08. International Conference.

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.

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

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

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