RESEARCH ARTICLE OPEN ACCESS

Design and Simulation of 32-Point FFT Using Mixed Radix Algorithm for FPGA Implementation

Harpreet Kaur [1], Tarandip Singh [2]

Harpreet Kaur, Department of Electronics Engineering
Sri Guru GranthSahib World University
Fatehgarh Sahib – India

ABSTRACT

This paper focus on the development of the fast Fourier transform (FFT), based on Decimation-In-Time (DIT) domain by using Mixed-Radix algorithm (Radix-4 and Radix-8).Fast Fourier transforms, popularly known as FFTs, have become an integral part of any type of digital communication system and a wide variety of approaches have been tried in order to optimize the algorithm for a variety of parameters such as area, time delay, power and speed. In this paper a comparison of area and minimum time delay are drawn between the proposed design of 32 point FFT by using Mixed-Radix algorithmwithRadix-2 algorithm. Here our goal is to implement Mixed Radix 32-point FFT by using hardware language (VHDL). Simulation and synthesis of design is done using Xilinx ISE 14.7. The result shows significant reduction in terms of area complexity and minimum time delay.

Keywords:- Fast Fourier transform (FFT), Discrete Fourier transform (DFT), DIT, Radix-4, 8, VHDL, FPGA.

I. INTRODUCTION

Currently in the field of signal processing for communications, there is a rapid development in FFT algorithms which act as a key in designing a system. This paper explains the implementation and simulation of 32-point FFT using mixed-radix algorithm. Due to radix-4 and radix-8, FFT can accomplish minimum time delay, reduce the area complexity and also achieve cost effective performance with less development time [1].

The FFT is an efficient class of computational algorithms of the DFT. The Discrete Fourier Transform (DFT) is one of the most popular transformations representing discrete signals in the frequency domain. Signals can be transformed between the time and the frequency domain through various transforms [2]. It is necessary to devise efficient algorithms to compute the DFT as the DFT is an important component in many practical applications. FFT algorithms optimize the DFT by eliminating redundant calculations [3]. FFT algorithms eliminate redundant calculations in the computation of the DFT and are therefore much faster .Cooley and Tukey developed FFT algorithm to reduce the computations of the Discrete Fourier Transform (DFT) from N^2 to N/2 log2N multiplications and N(N-1) to N log2N additions[4].

Several algorithms are developed to reduce the computational complexity, which includes Radix-2, Radix-4, Mixed-Radix, Split-Radix. All these algorithms are developed on one method, that is, Divide and Conquer method [5]. Fast Fourier Transform (FFT) is based on decomposition and breaking the transform into smaller sequences and at last again combining into one transform.

In this design the coding is done in VHDL& their synthesis and simulation is done using Xilinx ISE Design Suite 14.7 [6].

FFT Algorithm: The N-point discrete Fast Fourier Transform(DFT) is defined as [7]:

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

$0 \leq n \leq N-1$, $0 \leq k \leq N-1$

(1)

Where X(k) is the frequency domain and x(n) is time domain of sequence and W_N^{nk} is called Twiddle factor.

II. RADIX-8 FFT ALGORITHM

In Fig.1the basic butterfly structure for Radix-8 is shown which is used in first stage of FFT block diagram. The numbers of over flow lines indicates the twiddle factor is to be multiplied with the samples [8][11].In this work the number of stages are reduced to 2 since N= 4*8 that is; only 2 stages. The following will explains the functioning of Radix-8 FFT Algorithm.

In this Mixed Radix the length of N=32-points is recursively partitions into four parts DFTs of groups of every eighth sample. Due to the presence of eighth sample in every DFT parts for computational the DIT Radix-8 FFT is used. The outputs of these shorter FFTs are reused to compute many outputs, which are greatly reduced the total computational cost. In this Mixed Radix the DFT equation is rearranges into four parts: sums over all groups consists of every eighth discrete-time index n=[0,4,8,.....N-4], n=[1,5,9,...N-3], n=[2,6,10…N-2], n=[3,7,11…N-1] ,(This works only when the FFT length is multiple of eight),further mathematical manipulation shows that the length-N DFT can be computed as the sum of the even-indexed and odd-indexed discrete-time samples, respectively, where all the time samples are multiplied by twiddle factors W_N^{k}, W_N^{2k}, W_N^{3k}.

The following equations illustrate Radix-8 DIT-FFT, which the N-point input sequence splits into four subsequence's, x(4n), x(4n+1), x(4n+2) ,x(4n+3), n=0,1,... .Each subsequence consists a group of eight samples in order to computations these samples the Radix-8 DIT-FFT is used. Equation (1) can be written as follows:

$$X(k) = \sum_{n=0}^{N/4-1} x(4n) W_N^{4nk} + \sum_{n=0}^{N/4-1} x(4n+1) W_N^{(4n+1)k} + \sum_{n=0}^{N/4-1} x(4n+2) W_N^{(4n+2)k} + \sum_{n=0}^{N/4-1} x(4n+3) W_N^{(4n+3)k} \quad (2)$$

$$X(k) = X_1(k) + X_2(k) W_N^{k} + X_3(k) W_N^{2k} + X_4(k) W_N^{3k} \quad (3)$$

Where $X_1(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n) W_N^{4nk}$, $X_2(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n+1) W_N^{4nk}$

$X_3(k) = \sum_{n=0}^{N/4-1} x(4n+2) W_N^{4nk}$, $X_4(k) \sum_{n=0}^{N/4-1} x(4n+3) W_N^{4nk}$

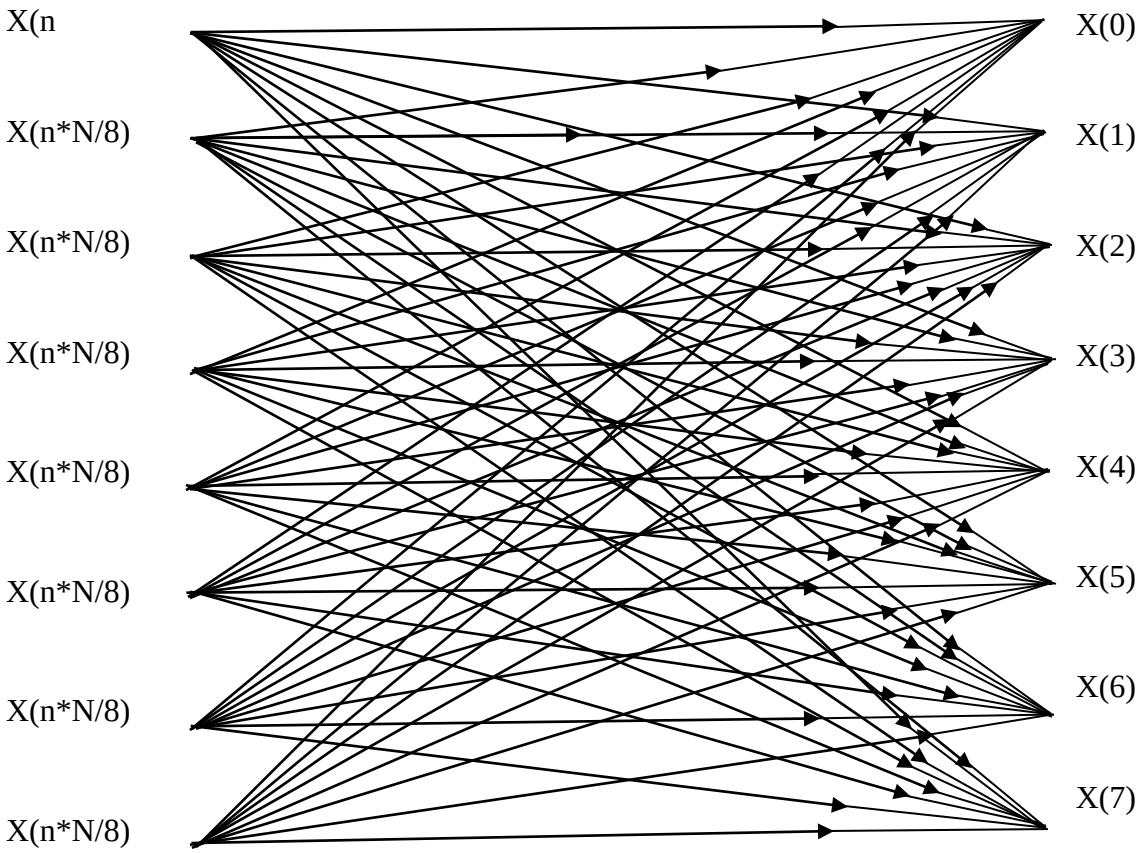

Fig.1.The basic butterfly for radix-8 FFT algorithm

III. RADIX-4 FFT ALGORITHM

In Fig. 2(a) and (b) shows the basic butterfly operation of radix-4 which have four inputs and four outputs; inputs are as x(n), x(n + n/4), x(n + n/2) and x(n + 3n/4) outputs are in normal order X(k). FFTs in which N = r*k, and where the butterflies used in each stage are the same, so called radix-r algorithms.

Radix-4 algorithms have a computational advantages over radix-2 algorithms because one radix-4 butterfly does the work of four radix-2 butterflies, and the radix-4 butterfly requires only three complex multipliers compared to four complex multipliers of four radix-2 butterflies [10].

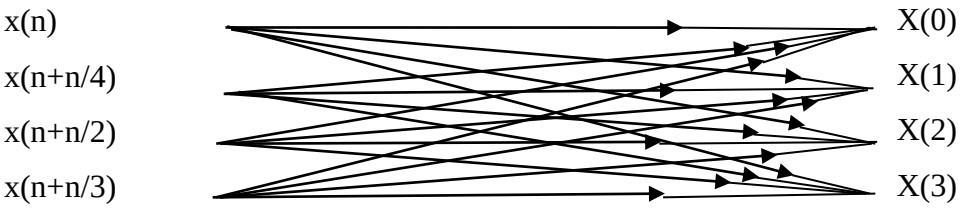

Fig. 2(a). The basic butterfly for radix-4 FFT algorithm

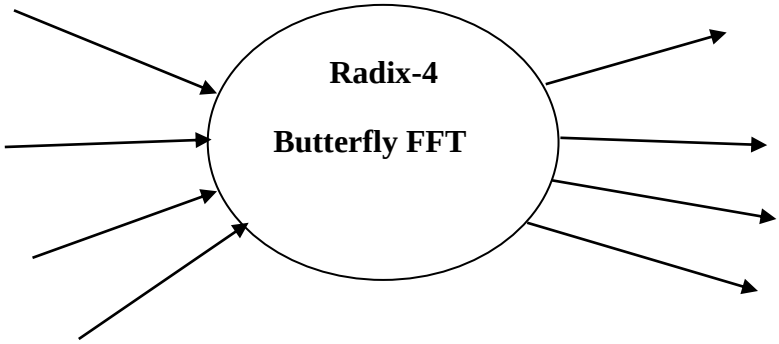

Fig. 2(b). The basic structure for radix-4 FFT

In this Mixed Radix algorithm for N=32-point the arithmetic operation of radix-4 DIT FFT is defined as-

$$X(k) = X_1(k) + X_2(k)\, W_N^{k} + X_3(k)\, W_N^{2k} + X_4(k)\, W_N^{3k} \qquad (3)$$

As shown in Fig.3 the block diagram of Mixed Radix (Radix-4 & 8) decimation in time (DIT) the method used for N=32-points FFT algorithm. In FFT blocks inputs are in digit-reversed order while the outputs are in normal order. Radix-4 decimation in time (DIT) is used in second stage of FFT of N=32-point length. The outputs of first stage of FFT are used as inputs which are processed with radix-4 FFT.

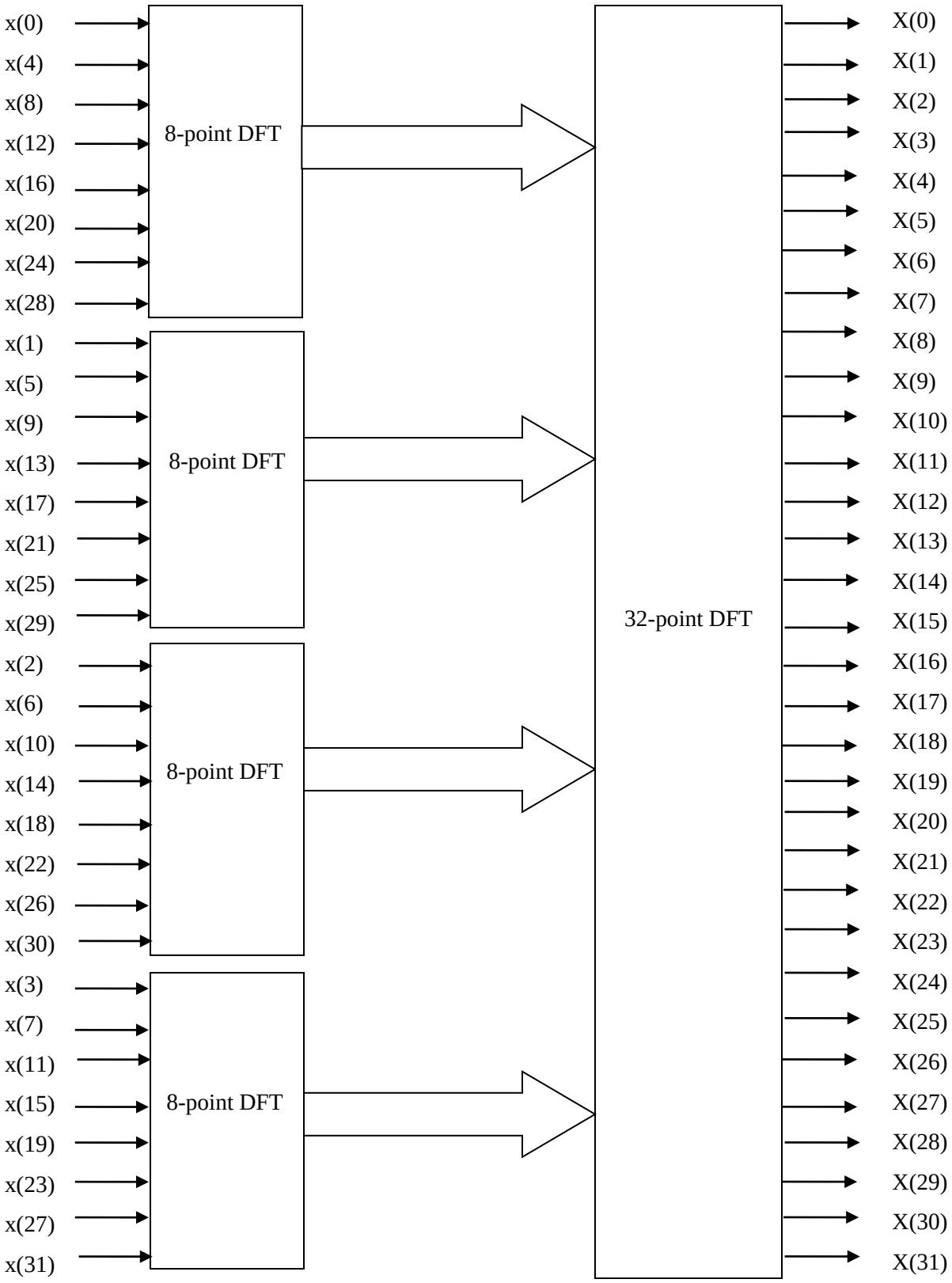

Fig. 3. Block diagram of Mixed Radix (Radix-4 & 8) Decimation in Time Domain FFT Algorithm for length of 32 Signals

IV. SIMULATION AND RESULTS

The RTL block thus obtained for the decimation in time domain by using Mixed Radix (radix-4 and radix-8) fast Fourier transform algorithm is shown in Fig.4. The internal architecture of the Butterfly Component is shown in Fig.5.

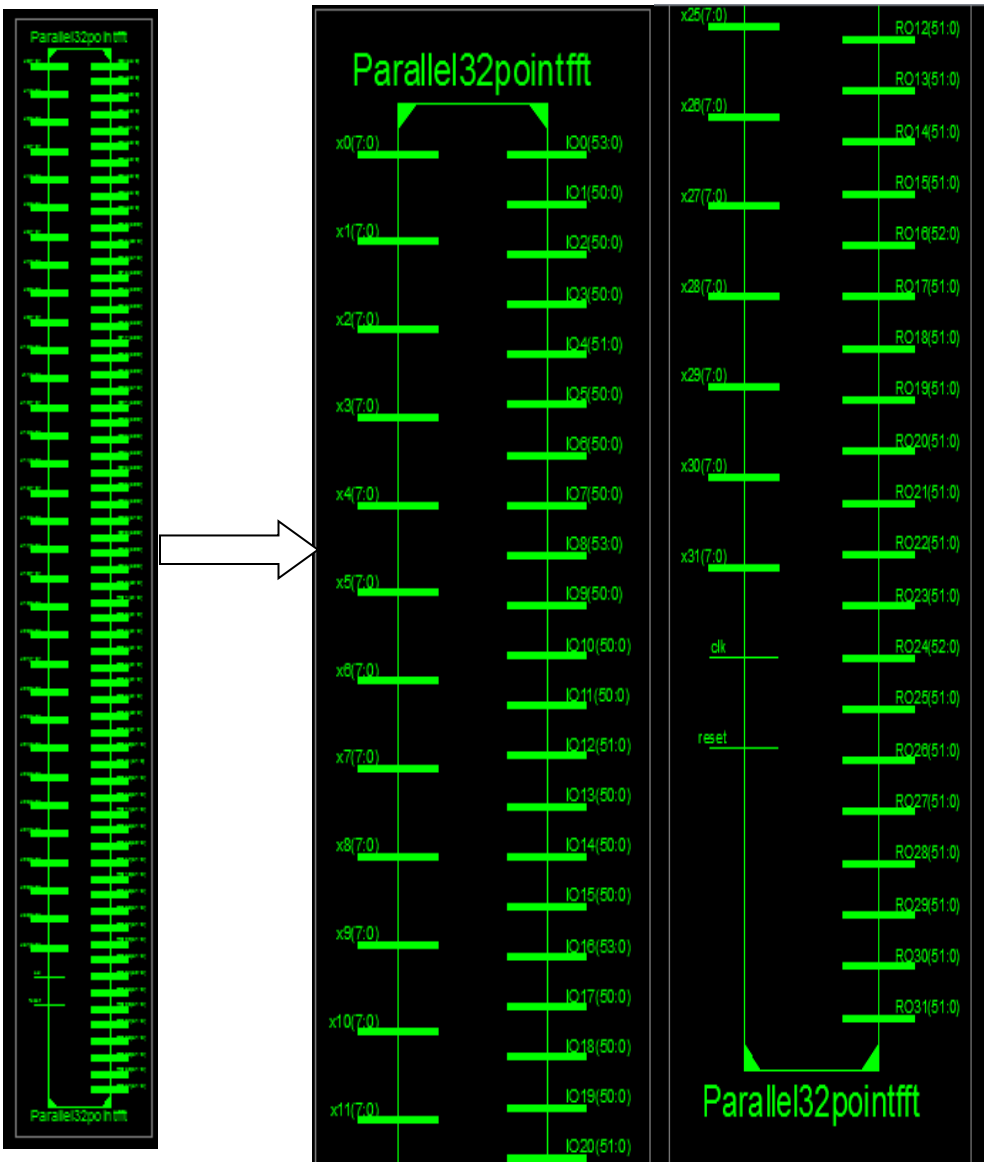

Fig.4. RTL view of 32 Point FFT

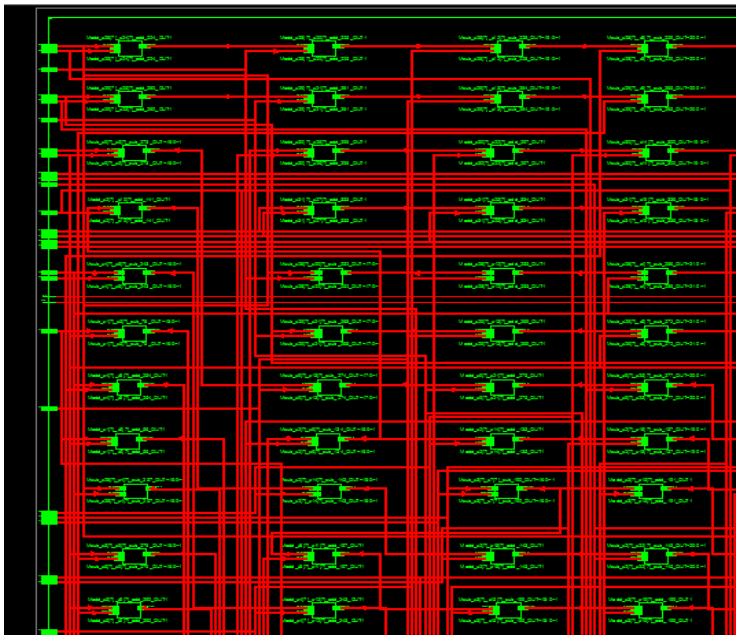

Fig.5. Internal Architecture of Butterfly Component

The proposed FFT block of signal length 32 has been simulated and synthesized using the Xilinx and the waveforms obtained after simulating are shown in Fig.6,7,8.

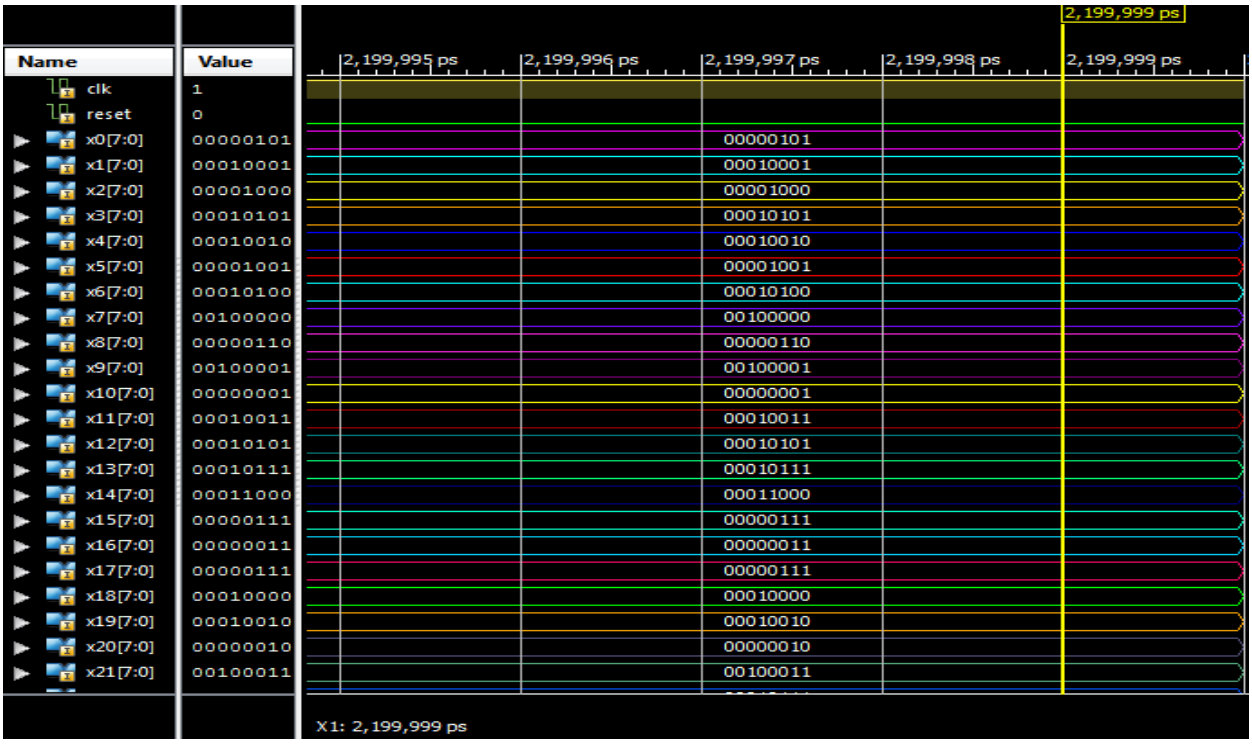

Fig.6. Simulation results of the 32-point FFT

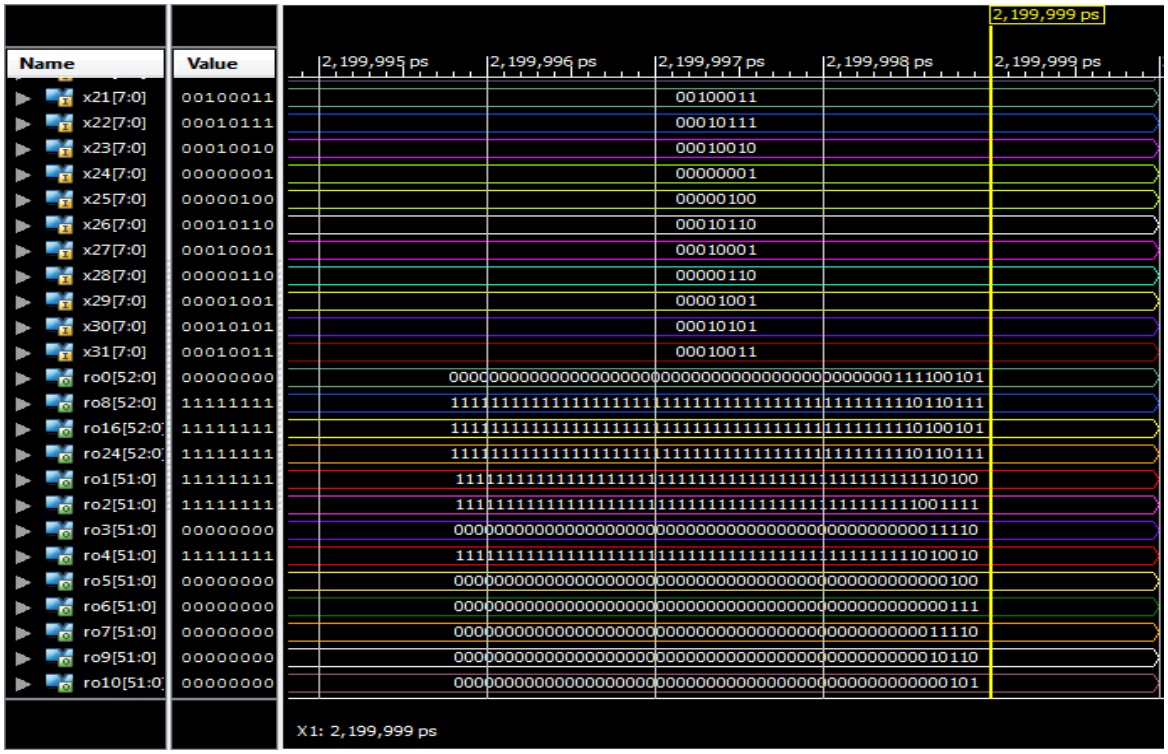

Fig.7.Simulation results of the 32-point FFT

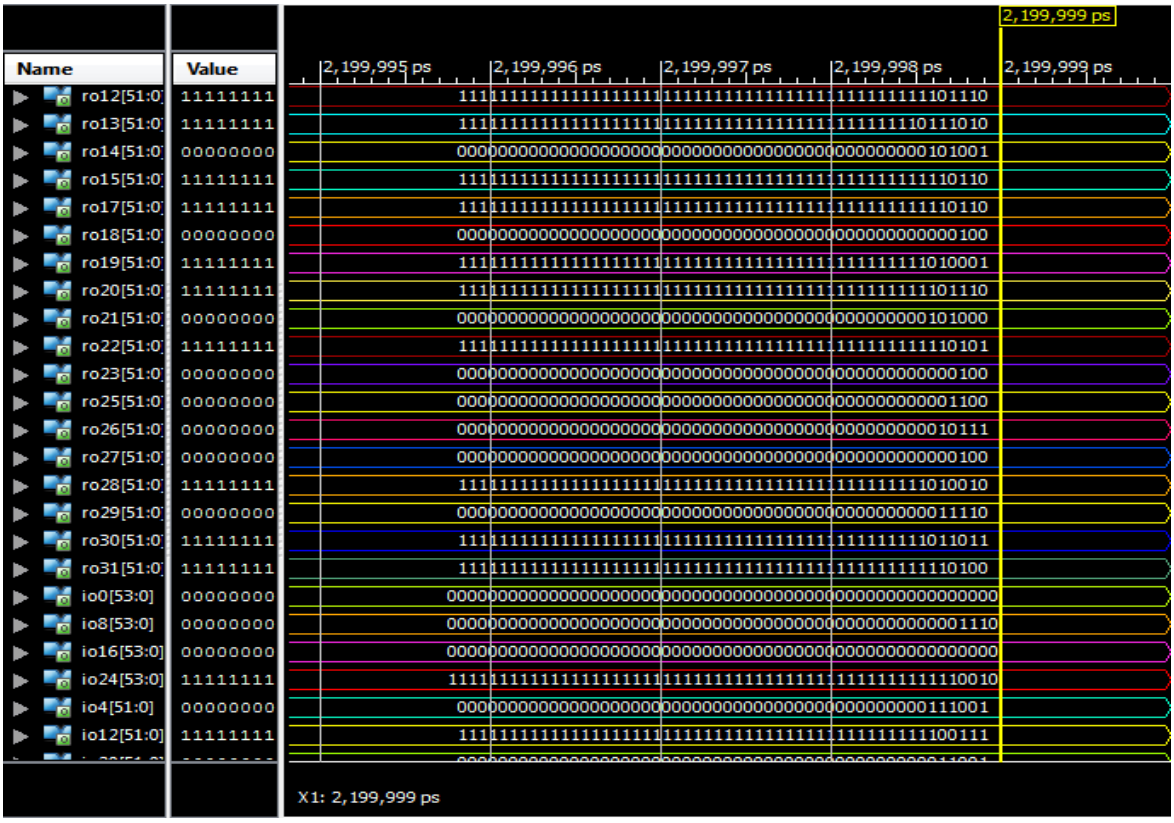

Fig.8.Simulation results of the 32-point FFT

The design summary of the system is as shown in table 1.The design summary shows that the proposed design utilizes 2394 (4%) number of slices registers, 26905 (98%) number of slice LUTs, 1723 (21%) number of fully used LUT-FF pairs, 3107 (1049%) number of bounded IOBs.

The final synthesis report is shown using the table below:

Table 1: Device utilization summary for 32 point FFT using Mixed Radix Algorithm

Device Utilization Summary for 32 Point FFT using Mixed Radix (Radix-4 & 8) Algorithm			
Logic Utilization	used	Available	Utilization
No. of Slice Registers	2394	54576	4%
No. of Slice LUTs	26905	27288	98%
No. of Fully Used LUT-FF Pairs	1723	8053	21%
No. of Bonded IOBs	3107	296	1049%

The timing summary of the proposed Design with Radix-4 &8 Algorithm shows the minimum period is 18.869ns and in Previous Design with Radix-2 Algorithm is 30.783ns. So here the comparison of the proposed design with previous work in terms of slice LUTs, bounded IOBs and time delay is as shown in table 2 and table 3.

Table 2: Comparison of device utilization of 32 point FFT

Logic utilization	Proposed Design with Radix-4 &8 Algorithm	Previous Design with Radix-2 Algorithm[7]
No. of Slice LUTs	98%	158%
No. of Bonded IOBs	1049%	1706%

Table 3: Comparison of Time delay of 32 point FFT

Parameter	Proposed Design with Radix-4 & 8 Algorithm	Previous Design with Radix-2 Algorithm[7]
Minimum Delay(ns)	18.869ns	30.783ns

V. CONCLUSION

In this paper we have designed 32 point FFT design using Radix-4 & 8 algorithm and their simulation and synthesis are done in VHDL using Xilinx synthesis tool on Spartan-6 SP605 Evaluation Platform. The Performance analysis is done by comparing the results of proposed design of 32-point FFT by using Mixed Radix (Radix 4 & 8) with Radix 2 FFT algorithms by taking various parameters into consideration. It is founded that, mixed radix algorithm having better performance over radix-2 algorithm in term of reduction of area complexity and minimum time delay.

REFERENCES

[1] Ahmed Saeed, M. Elbably, G. Abdelfadeel, and M. I. Eladawy; "Efficient FPGA implementation of FFT/IFFT Processor"; International journal of circuits, Systems and Signal Processing,Issue 3, Volume 3, 2009.

[2] SaadBouguezel, M.Omair Ahmad, "Improved Radix-4 and Radix-8 Algorithms", IEEE Department of Electrical and Computer Engineering Concordia University 1455 de MaisenneuveBlvd west Montreal, P.q.,Canada,vol.78,2013.

[3] J. D. Bruguera and T. Lang, "Implementation of the FFTbutterfly with redundant arithmetic," IEEE Trans. Circuits Syst. II,vol. 43, pp. 717–723, May 1996.

[4] J. W. Cooley and J. W. Tukey, "An algorithm for the machine computationof complex Fourier series," *Math.Comput.*, vol. 19, pp. 297–301,1965.

[5] SnehaN.Kherde, MeghanaHasamnis, " Efficient Design and Implementation of FFT", International Journal of Engineering science and Technology(IJEST), ISSN:0975-5462 NCICT Special Issue Feb 2011.

[6] Peter J. Ashenden, "The Designer's Guide to VHDL, Third Edition (Systems on Silicon)", 2008, ISBN 0-1208-8785-1.

[7] AsmitaHaveliya, "Design and simulation of 32-point FFT using Radix-2 Algorithm for FPGA Implmentation",2012 second International conference on Advanced Computing and Communication Technologies.

[8] Young-jin Moon, Young-il Kim "A Mixed-Radix 4-2 Butterfly with Simple Bit Revering for Ordering the Output sequences," ICA0T2006 vol. 4, pp. 1772–1774, February 2006.

[9] Shi Jiangi; Tian Yinghui; Wang Mingxing; Yang Zhe; "A Novel design of 1024-point pipelined FFT processor based on CORDIC algorithm"; Intelligent System Design And engineering Application (ISDEA) 2012 second International Conference on Digital Object Identifier.

[10]Marti-Puig,P., ReigBolano,R., "Radix-4 FFT algorithms with ordered input and output data" 16th International Conference onDigital Signal Processing, 2009.

[11] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Improved radix-4and radix-8 FFT algorithms," in Proc. ISCAS'04: IEEE Int. Symp. CircuitsSyst., 2004, vol. 3, pp. 561–564.