ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

Design and Implementation of a low latency Radix-8 FFT using FPGA Architecture

[1]V.Lavanya, [2]B.Kiranmai
[1]Department of ECE, M.V.G.R College of Engineering, Vizianagaram
[2]Department of ECE, Lendi Institute of Engineering and Technology, Vizianagaram
[1]lavanyavadda@gmail.com, [2]kbabburu@gmail.com

Abstract:

In view of emerging networks low power consumption, low latency and accuracy are the most important issues. To address these issues, optimization is the important aspect that can solve these problems. In the communication channel the low latency is due to the conversion from time domain to the frequency domain at the receiver. The conversion of the time domain to frequency domain can be obtained by using FFT/IFFT which involves butterfly radix methodology. In this paper, a new algorithm is developed to reduce the computation operation in Radix-8 FFT implementation and which intern reduces the area, power consumption and latency. This optimized Radix-8 provides better performance compared with the previous Radix architectures.

Keywords: Radix, FFT, Low latency, Twiddle factor, Complex Multiplication

I. INTRODUCTION

The efficient computations and analysis of the spectrum are obtained in the fields of communications through conversion from time domain to frequency domain and which can be achieved by DFT/FFT [1]. Mainly FFT has been used in OFDM systems, Wireless communications, GPS [2]. In FFT algorithm, the computation time and complexity can be reduced by applying logarithmically [3]. In the literature there are number of algorithms were proposed to reduce the complexity and time taken to reduce the computation time [7].

The proposed algorithms for the Digital signal processing are performed excellent on MATLAB toolboxes [4]. In this context, a digital signal processor which converts time domain signals to frequency domain are the crucial and needs new approaches and applications for reducing the latency. So optimized and low computation FFT implementation is very essential for reducing the latency in DSP processors [5]. The VLSI technology has given enormous changes for implementing the DSP hardware design with low power, area and reduced latency [6]. In the present day scenario, the required and the desirable throughput are to be achieved with reduced time and power [8].

ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

In this context, to reduce the design complexity and time delay in higher radix FFT implementation because the higher radix requires the higher hardware design and complexity depends on the length of FFT implementation. A new algorithm has been proposed in this paper to achieve less complexity in design and reduced latency for conversion in 8 point FFT implementation.

II. Algorithm for proposed architecture

The radix format is the most efficient design for computing the DFT by using FFT. In this section the existed systems for Radix-2, Radix4, Radix-8 are discussed and also the proposed optimized Radix-8 architecture algorithm is presented here.

II.I RADIX-2 ALGORITHM

The basic fundamental method which was proposed by the Cooley-Tukey for the computation of the DFT is butterfly method. The radix-2 is that which requires inputs 2 and outputs2 at the end [9]. The equations 1 and 2 explain the computation of DFT using butterfly method.

$$x(2k) = \sum_{n=0}^{\left(\frac{N}{2}\right)-1}([x(n) + x(n + (\frac{N}{2}))]w_{\frac{N}{2}}^{kn})\text{----------} (1)$$

$$x(2k+1) = \sum_{n=0}^{\left(\frac{N}{2}\right)-1}([x(n) - x(n + (\frac{N}{2}))w_N^n]w_{\frac{N}{2}}^{kn})\text{---------} (2)$$

The above equations 1 and 2 are implemented with adder, subtractor and twiddle factor. The butterfly diagram for 8 point FFT in DIF (decimation in frequency) is shown in figure1

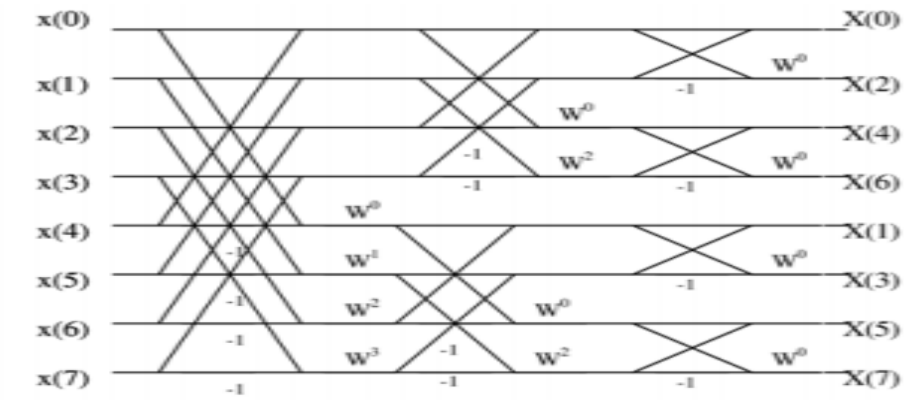

Figure 1: Radix-2 8-point butterfly flow diagram

ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

The figure 1 shows the butterfly diagram of the 8point FFT and the mid stage represents the twiddle factor, computed for converting the values.

II.II RADIX-4 ALGORITHM

The Radix-4 algorithm has computations for 4 inputs and will have 4 outputs at the end. The Radix -4 computations are represented by these following equations.

$$x(4k) = \sum_{n=0}^{\left(\frac{N}{4}\right)-1}([x(n) + x(n+(\tfrac{N}{4})) + x\left(n+\left(\tfrac{N}{2}\right)\right) + x\left(n+\left(\tfrac{3N}{4}\right)\right)]\, w_N^0)]w_{\frac{N}{4}}^{kn})\text{---3}$$

$$x(4k+1) = \sum_{n=0}^{\left(\frac{N}{4}\right)-1}([x(n) - j*x(n+(\tfrac{N}{4})) - x\left(n+\left(\tfrac{N}{2}\right)\right) + j*x\left(n+\left(\tfrac{3N}{4}\right)\right)]\, w_N^0)]w_{\frac{N}{4}}^{kn})\text{—4}$$

$$x(4k+2) = \sum_{n=0}^{\left(\frac{N}{4}\right)-1}([x(n) - x(n+(\tfrac{N}{4})) + x\left(n+\left(\tfrac{N}{2}\right)\right) - x\left(n+\left(\tfrac{3N}{4}\right)\right)]\, w_N^0)]w_{\frac{N}{4}}^{kn})\text{---5}$$

$$x(4k+3) = \sum_{n=0}^{\left(\frac{N}{4}\right)-1}([x(n) + j*x(n+(\tfrac{N}{4})) - x\left(n+\left(\tfrac{N}{2}\right)\right) - j*x\left(n+\left(\tfrac{3N}{4}\right)\right)]\, w_N^0)]w_{\frac{N}{4}}^{kn})\text{---6}$$

The equations from 3 to 6 are representing the computational complexity for calculating the twiddle factor for Radix4 and the butterfly diagram is shown in the figure with DIF.

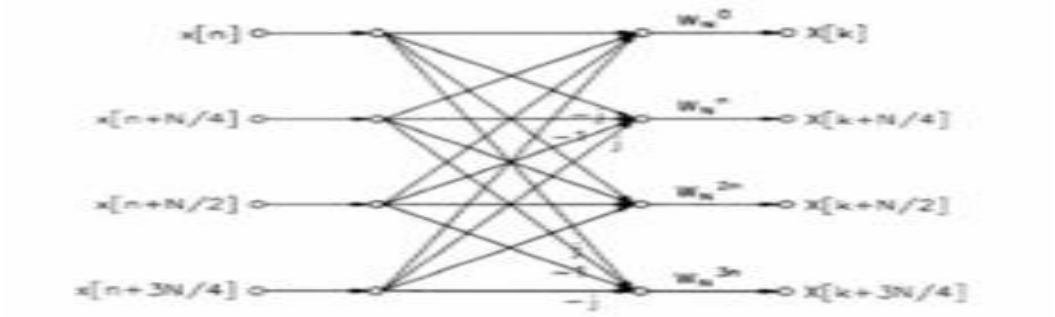

Figure 2: radix 4 4-point FFT butterfly diagram

To obtain the 8 point FFT for Radix 4 the further stages are coupled with Radix-2.

II. III RADIX-8 ALGORITHM

The Radix-8 algorithm requires the 8 inputs for computation and provides 8 outputs at the end. The computational complexity is very high as the Radix increases and that is shown in these following equations.

ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

$$X(k)=\sum_{m=0}^{\frac{N}{8}-1} x(8m)W_N^{k(8m)}+\sum_{m=0}^{\frac{N}{8}-1} x(8m+1)W_N^{k(8m+1)}+\sum_{m=0}^{\frac{N}{8}-1} x(8m+)W_N^{k(8m+2)}+\sum_{m=0}^{\frac{N}{8}-1} x(8m+3)W_N^{k(8m+3)}+\sum_{m=0}^{\frac{N}{8}-1} x(8m+)W_N^{k(8m+4)}+\sum_{m=0}^{\frac{N}{8}-1} x(8m+)W_N^{k(8m+5)}+\sum_{m=0}^{\frac{N}{8}-1} x(8m+6)W_N^{k(8m+6)}+\sum_{m=0}^{\frac{N}{8}-1} x(8m+7)W_N^{k(8m+7)}$$

$$=DFT_{N/8}^{[x(8n)]}+W_N^k DFT_{N/8}^{[x(8n+1)]}+W_N^k DFT_{N/8}^{[x(8n+2)]}+W_N^k DFT_{N/8}^{[x(8n+3)]}+W_N^k DFT_{N/8}^{[x(8n+4)]}+W_N^k DFT_{N/8}^{[x(8n+5)]}+W_N^k DFT_{N/8}^{[x(8n+6)]}+W_N^k DFT_{N/8}^{[x(8n+7)]}$$

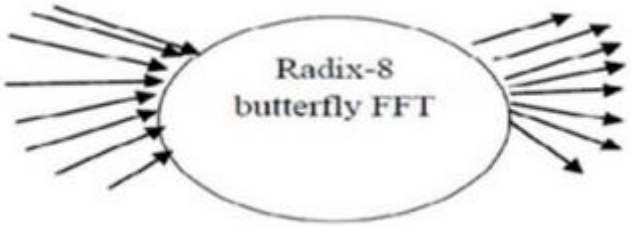

Fig 3: radix 8 8-point flow diagram

III Proposed Radix-8 algorithm architecture

The computational complexity and delay increases along with the higher Radix FFT implementation but higher throughput will be achieved with higher Radix FFT. In this proposed algorithm, to decrease the computational time and increase the throughput, the standard Radix -8 algorithms is analyzed and this can be achieved by assigning base to 8.

In the proposed algorithm, an N point DFT is separated into groups of iteratively associated Radix 'R' point transformation and the input x(n) be is powers of R. For Radix 8 algorithm ' R' value is 8 and the Decimation in frequency (DIF) Radix-8 FFT can be obtained by repetitively partitions DFT with 'R' point transformation i.e into eight length DFT with individual eighth sample . In this context the distributed outputs of smaller FFTs are can be again utilized for computing the overall outputs because of this the computational complexity reduces and the latency will be reduced effectively.

Hence the complex computations that are carried at the first stage of FFT implementation will be pushed to the final stage and thus reducing the multiplication time required, area, the delay and the power utilization at the same time it improves the maximum throughput.

ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

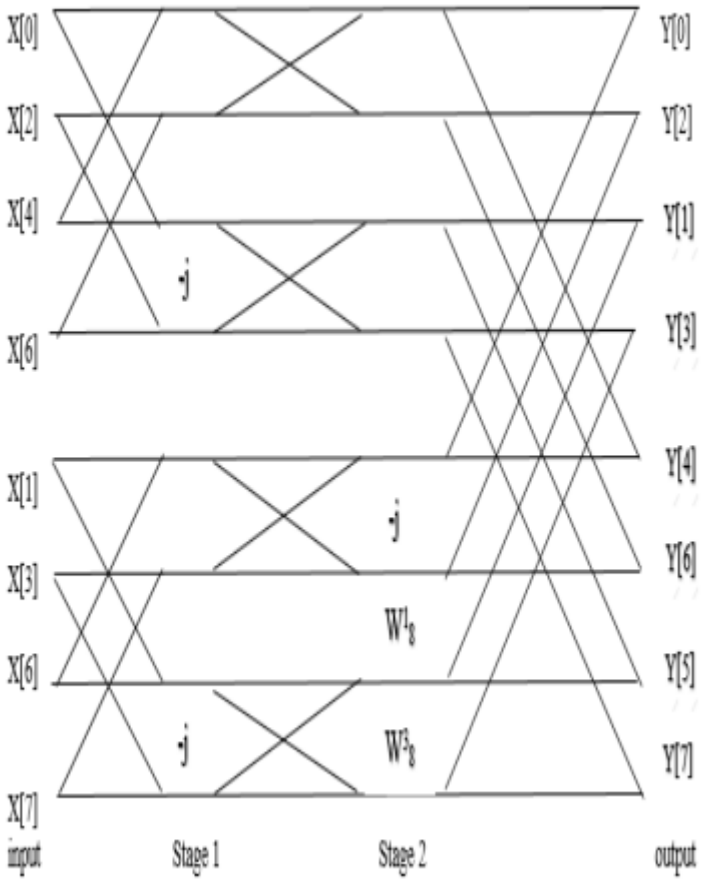

Figure 4: Proposed optimized Radix 8 FFT flow diagram

The figure 4 shows the proposed Radix 8 FFT diagram, in this algorithm the stage 1 has less computations as it takes the input values and processes to the stage 2. With this less computation the input values are given to the stage 2 so the computational speed increases and the design complexity reduces. Hence reduction in multiplications and complexity, improves the system design, power consumption and latency.

The proposed architecture of Radix-8 algorithm consists of the output as complex and real terms. In this proposed system the given 8 bit inputs are taken as real data. This data is further preceded with using butterfly Radix-2 implementation which reduces the delay path. This Radix 2 module is calculated and every time instances it will be called for small computations. Hence the simplified computations continued up to the final stage calculations which improve faster system calculations and efficient final value in frequency domain.

In this proposed system the twiddle factor have been calculated will be stored in the register this stored value in the register multiplied with the stage 1 calculations. This

ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

process will be continued for the each until the final stage is obtained. Hence the final stage computations in the 8 point Radix-8 FFT reduced and hence it is a high speed improved technique.

IV. Result

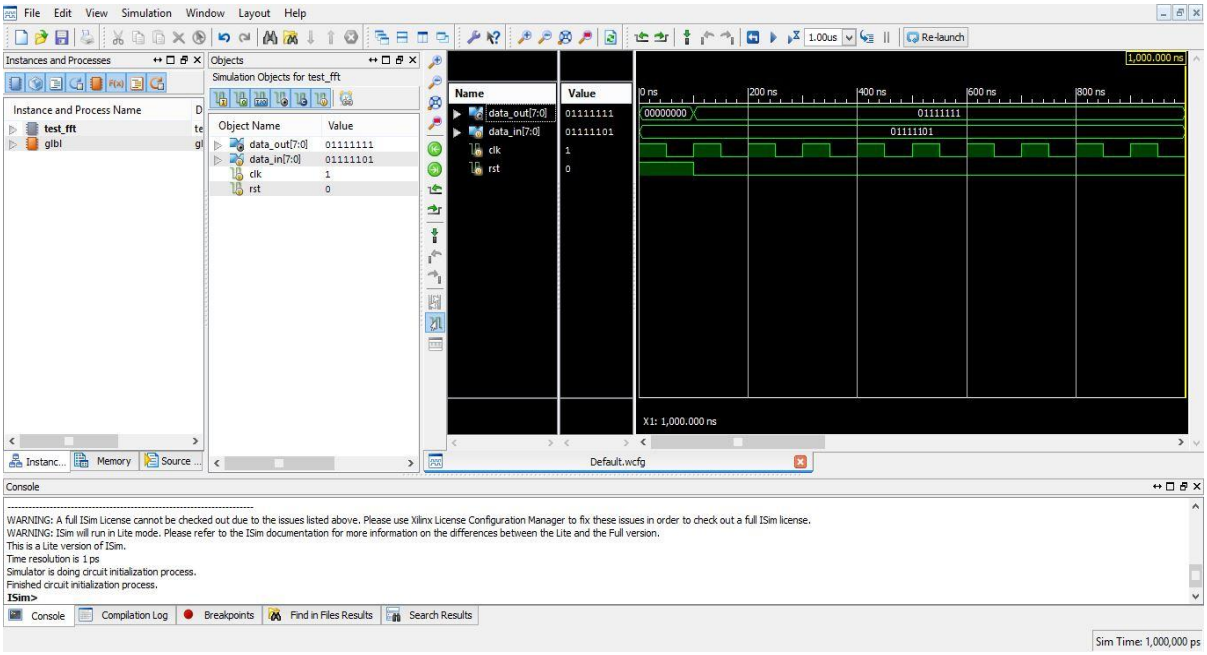

Fig 5: Simulation result of the existing system

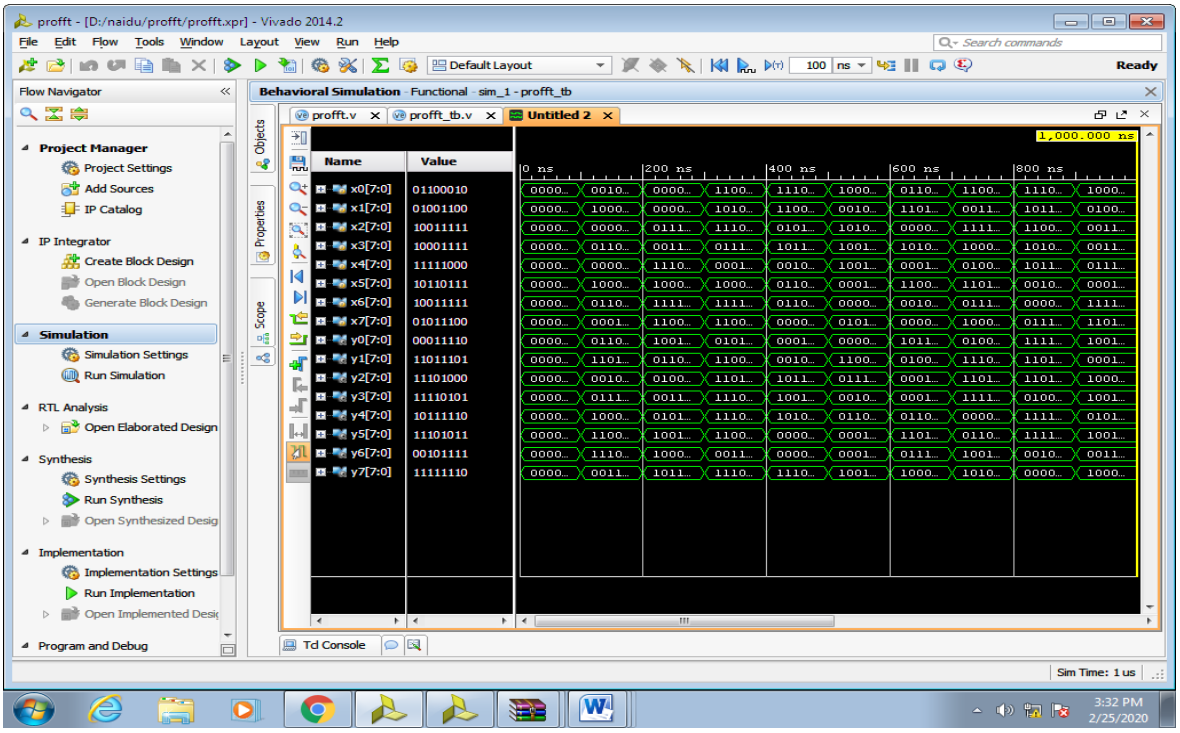

Fig 6: Simulation result of the proposed system

ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

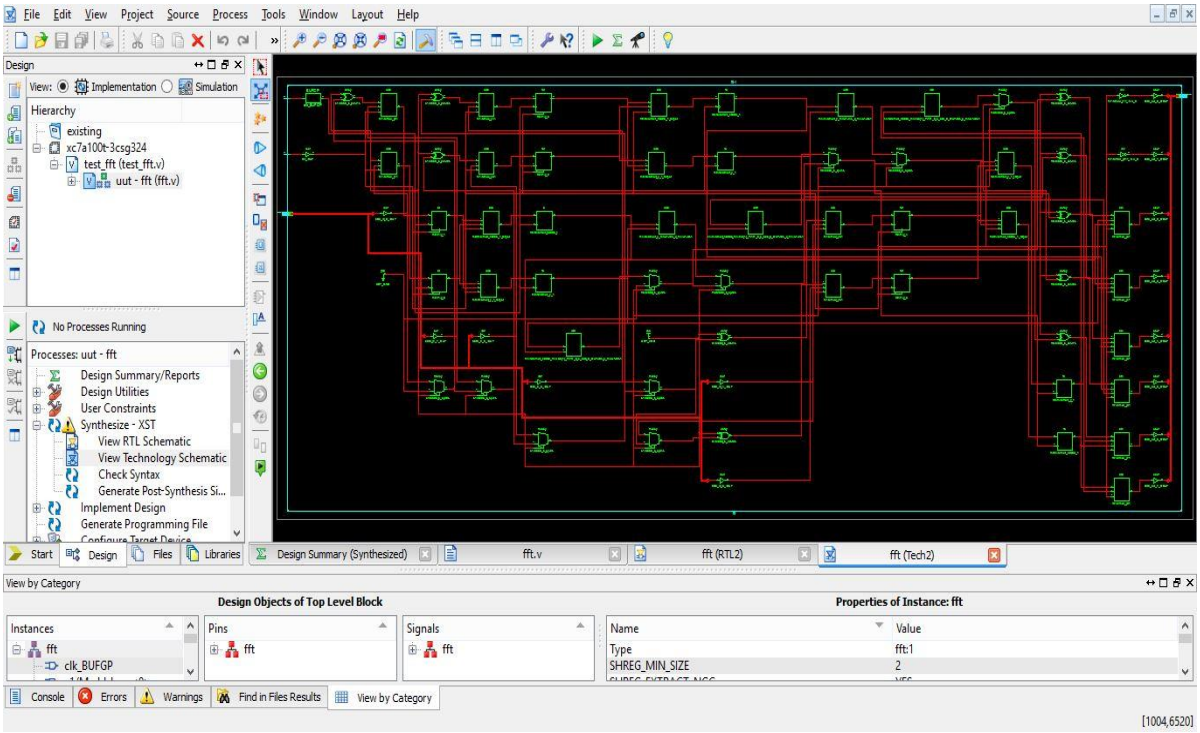

Fig 7: Technology schematic of the existing system

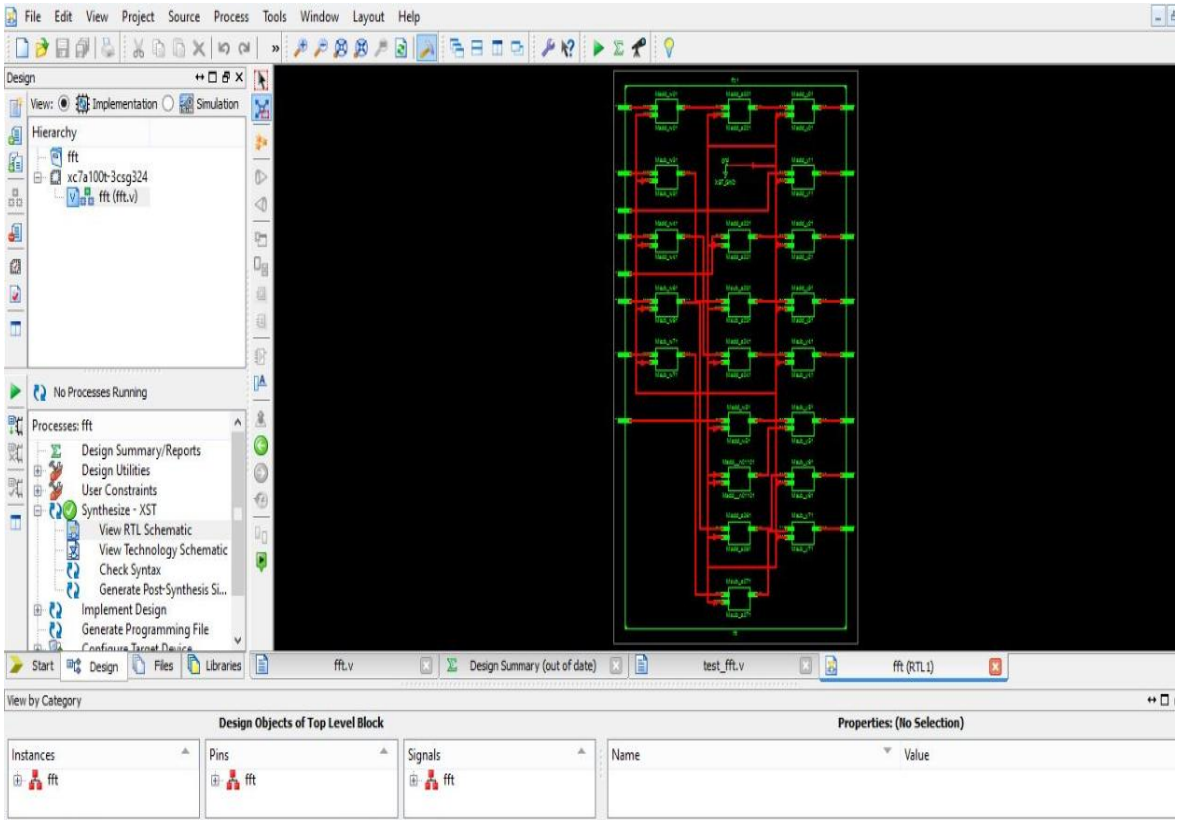

Fig 8: Technology schematic of the proposed system

ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

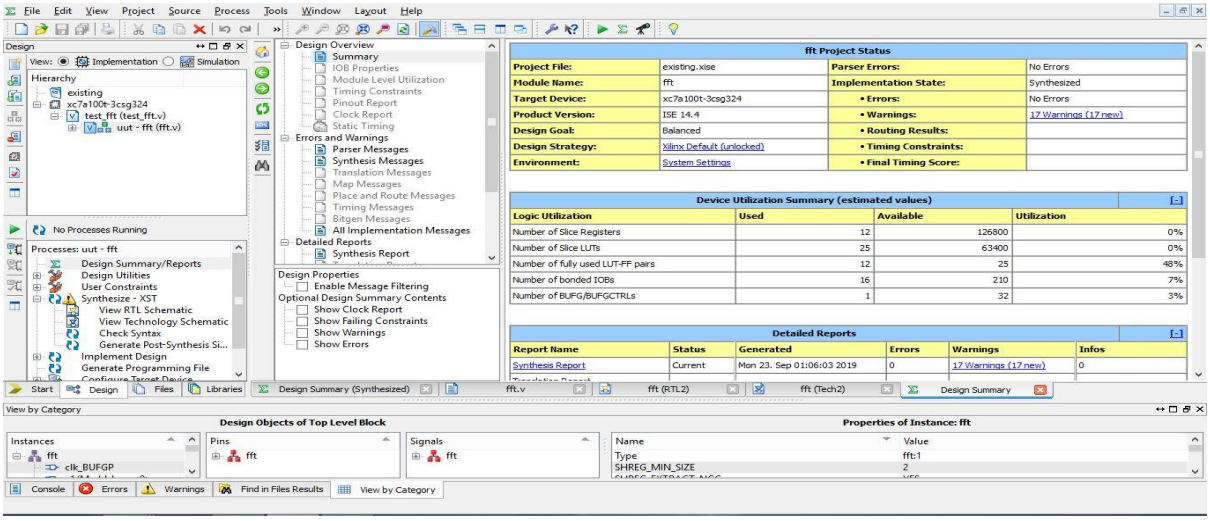

Fig 9: Summary report of the existing system

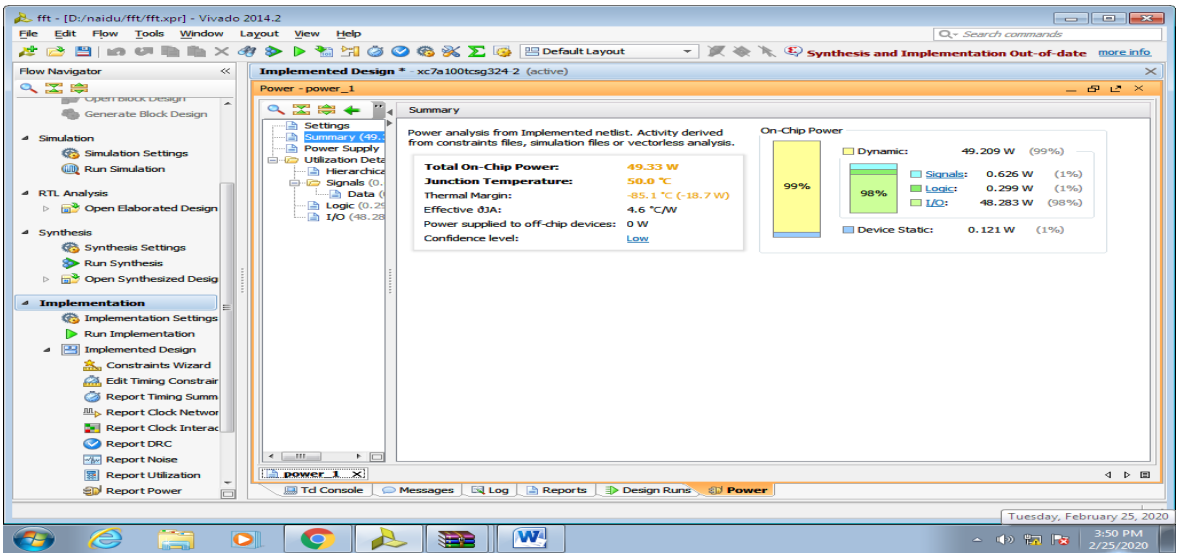

Fig 10: Power report of the existing system

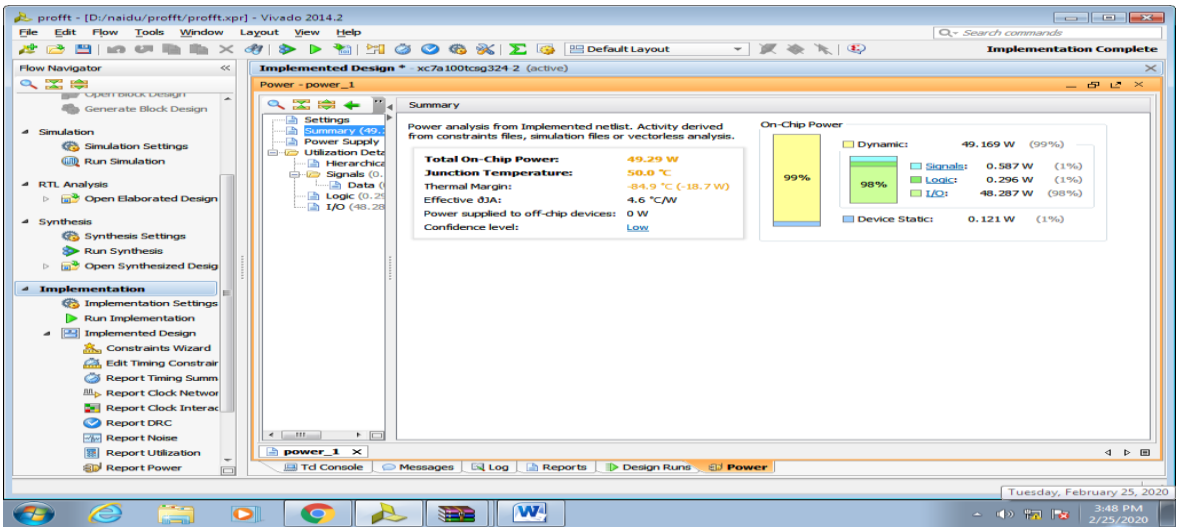

Fig 11: Power report of the proposed system

ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

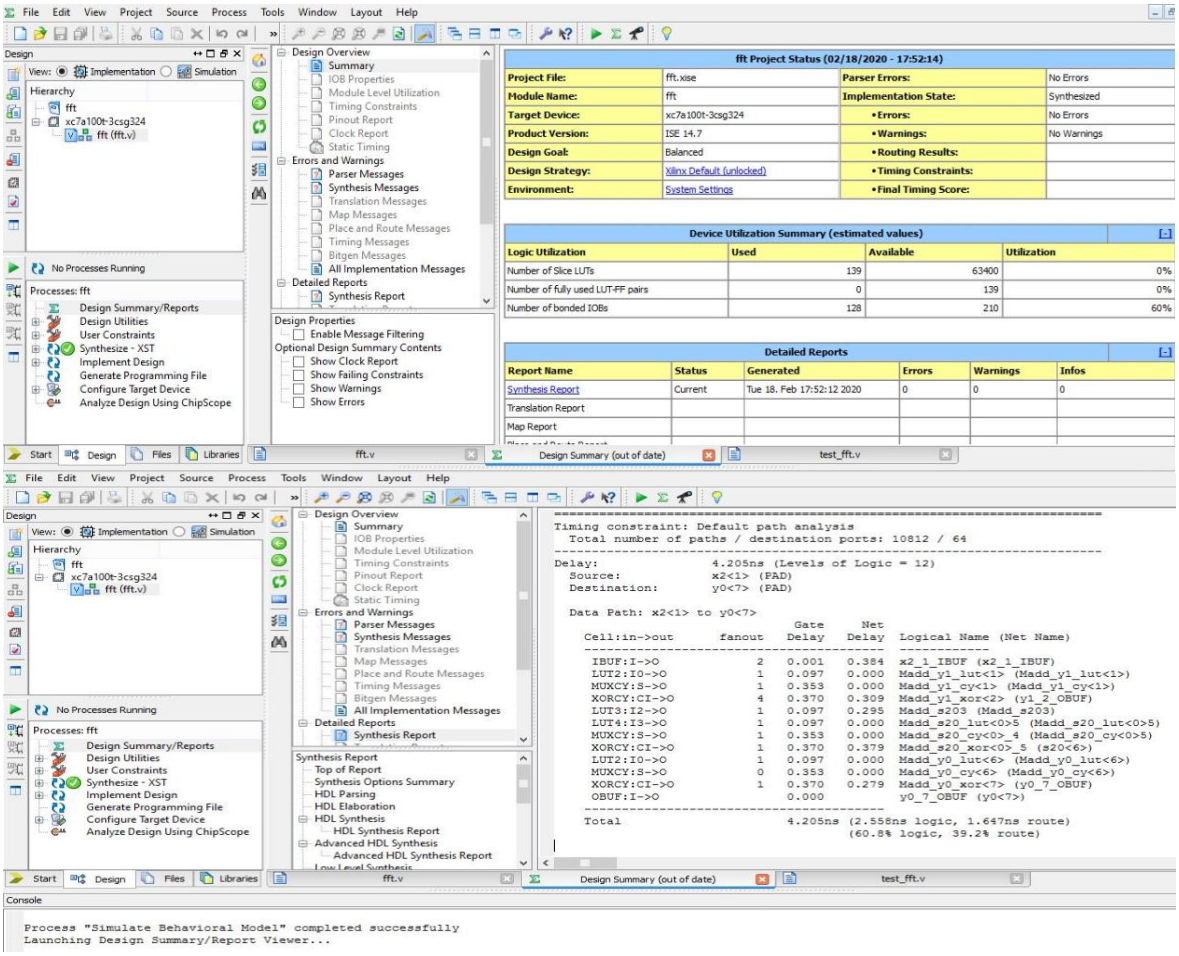

Fig 12: Summary report of the proposed system

Table 1: comparison of Radix-8 FFT architectures

	LUT	Delay	Memory usage
Existing (8- point radix 2)	138	4.955 ns	385292 kb
Proposed (Optimized Radix 8)	139	4.205 ns	386348 kb

V. Conclusion and Future work

The proposed Radix 8 optimized system technique improves the high throughput and area required. Power consumption and delay can be reduced. The FFT implementation with reduced latency implementation has enormous applications in the wireless communications where the OFDMA implementation is purely depends on FFT /IFFT modules. The latency in the wireless communications can be reduced with this proposed algorithm.

ISSN: 2278-4632
Vol-10 Issue-5 No. 5 May 2020

Further this work can be enhanced with an Application Specific Integrated Chip (ASIC) for reducing the latency and improving the throughput in wireless communication systems

References

[1] T. Widhe, "Design of Efficient Radix-8 Butterfly PEs for VLSI," IEEE Trans. Audio and Electro-acoustics, vol. AU-17, pp. 109 - 119, 2013.

[2] M. Rajasekhar, K. Manjula, "Design and Simulation of 512 Point FFT using RADIX-8 Algorithm," International Journal of Applied Sciences, Engineering and Management ,ISSN 2320 – 3439, Vol. 05, No. 04, pp. 30 – 33, 2016.

[3] LihongJia, YonghongGao and HannuTenhunen, "Efficient VLSI Implementation of Radix-8 FFT Algorithm," IEEE 7 803 -5 582-2/99, 2009.

[4] Mingxi Sun, LiyuTian and Dongmin Dai, "Radix-8 FFT Processor Design Based on FPGA," 5th International Congress on Image and Signal Processing, 2012.

[5] Shousheng He and Mats Torkelson," Designing Pipeline FFT Processor for OFDM (de) Modulation" URSI International Symposium on Signals, Systems, and Electronics. Conference Proceedings, PISA, ITALY, 2008.

[6] Saad Bouguezel, M. Omair Ahmad," A New Radix-2=8 FFT Algorithm for Length-q 2m DFTs" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 51, NO. 9, 2004.

[7] Kala S, Nalesh S, ArkaMaity, S K Nandy," High Throughput, Low Latency, Memory Optimized 64K Point FFT Architecture using Novel Radix-4 Butterfly Unit", IEEE International Symposium on Circuits and Systems, 2013.

[8] MengjunLi,Yongxin Zhu, Xu Wang, Tian Huang, Weida Chen, Bin Liu, Yishu Mao," Evaluation of Variable Precision Computing with Variable Precision FFT Implementation on FPGA" International Conference on Field-Programmable Technology, 2016.

[9] MounirArioua, Said Belkouch, Mohamed Agdad," VHDL implementation of an optimized 8-point FFT/IFFT processor in pipeline architecture for OFDM systems," International Conference on Multimedia Computing and Systems, 2011.

[10] Johan Lofgren and Peter Nilsson," On Hardware Implementation of Radix 3 and Radix 5 FFT Kernels for LTE systems" International Conference on Multimedia Computing and Systems, 2011.

[11] Vinay Kumar M, David Selvakumar A, Sobha P M," Area and Frequency optimized 1024 point Radix-2 FFT Processor on FPGA", International Conference on VLSI Systems, Architecture, Technology and applications, 2016.

