DESIGN OF FFT ARCHITECTURE USING KOGGE STONE ADDER

Rambabu Nusullapalli,
Department of Electronics and Communication Engineering,
Pace Institute of Technology and Sciences,
Ongole, Andhra Pradesh, India
rambabu_n@pace.ac.in

Vaishnavi N,
Department of Electronics and Communication Engineering,
Pace Institute of Technology and Sciences,
Ongole, Andhra Pradesh, India
vaishnavi_n@pace.ac.in

Abstract: An efficient Fast Fourier Transform (FFT) algorithm is used in the Orthogonal Frequency Division Multiplexing (OFDM) applications in order to compute the discrete Fourier transform. Also, a Single Path Delay Feedback (SDF) which is pipeline FFT architecture is used for faster performance to achieve high throughput. In conventional method, the FFT design has high delay and power due to time taken by the multiplication part. To decrease the delay, Kogge Stone Parallel Prefix Adder (KSPPA) is used with booth multiplier. As SDF is a simpler approach to realize FFT in different length, 64-point Radix-4 SDF-FFT algorithm using KSPPA in the booth multiplier is discussed in this study. The system is implemented in Xilinx 12.4 ISE and simulated using MODELSIM 6.3c. Results show that the system reduces the delay and power.

Keywords: FFT, SDF, Kogge stone parallel prefix adder, booth multiplier, Xilinx.

I. INTRODUCTION

The fast execution of OFDM is required in many real time applications such as radar and biomedical instrumentation in military domain. Memory based architecture with pipelined structure is described in [1]. A Radix-2 for 8 point structure is designed using SDF. The design of FFT using multiple radix algorithm is described in [2]. Discrete In Time (DIT) domain FFT structure is used. Expandable

A generalized mixed radix algorithm is described in [3]. It supports the structure of high radix by reversing the decomposition order without memory conflict. A generalized reconfigurable high radix FFT is described in [4] within a single clock domain. It carries continuous data flow. The pipeline SDF FFT architecture for Radix 22 is described in [5]. To process the radar data pipeline FFT is used with serial storage and multiple arithmetic units is described in [6]. This is used to offer spectral analysis for tracking the pulse Doppler radar. Pipeline mixed pipeline radix-2 SDF and radix-8 MDC architecture is described in [7]. Also, to improve the energy and area efficiency a dual optimized multipath multiplication approach is intended.

FFT using Vedic multiplication is described in [8]. Urdhva Tiryakbhyam is used in the Vedic computation. A mixed radix-4 and 8 FFT using SDF-Single Delay Commutator (SDC) is described in [9]. A modified borrow select adder is

used in to minimize the area. A design of FFT architecture for 4-point radix-2 is described in [10]. It is based on KSPPA to reduce the delay of FFT computation. An overview of the FFT processor is described in [11]. A comparison of different FFT structure is also made.

A Radix-4 FFT using SDF-MDC is described in [12]. A modified bit parallel multiplier is used In the place of twiddle factor multiplication. It offers good performance in terms of high speed. A Radix-2 FFT using serial rapid single flux quantum multipliers-adders is described in [13]. It uses the architecture of parallel flow which consists of carry save serial adders and single bit wide serial multipliers.

A design of FFT using fast adders is described in [14]. Fast adders are used in the digital image processing to reduce the loss in the image quality and also improve the processing time. A carry free addition block is used in the fast adder module. A pipeline architecture using adder compressors with new XOR gate structure for Radix-2 DIT-FFT is described in [15]. It reduces the number of critical path structures as well as real multipliers.

In this study, 64-point radix-4 FFT architecture using KSPPA in booth multiplier is presented. The organization of this paper is as follows. The methods and materials of the design of FFT using KSPPA in booth multiplier are discussed in section 2. Section 3 explains the results of the system and section 4 describes conclusion of the work.

II. METHODS AND MATERIALS

Figure 1 shows the 64-point radix-4 FFT using SDF structure. Radix-4 SDF utilizes the register more efficiently by storing one output of each butterfly in the feedback shift registers. It has the same number of multipliers and butterfly units as in radix-2 MDC but reduces the memory registers requirement by (N-1). Thus, the Radix-4 SDF-FFT structure occupies less area. The Radix-4 butterfly structure is made up of four outputs and four inputs and its length is 4O where O is the number of stages.

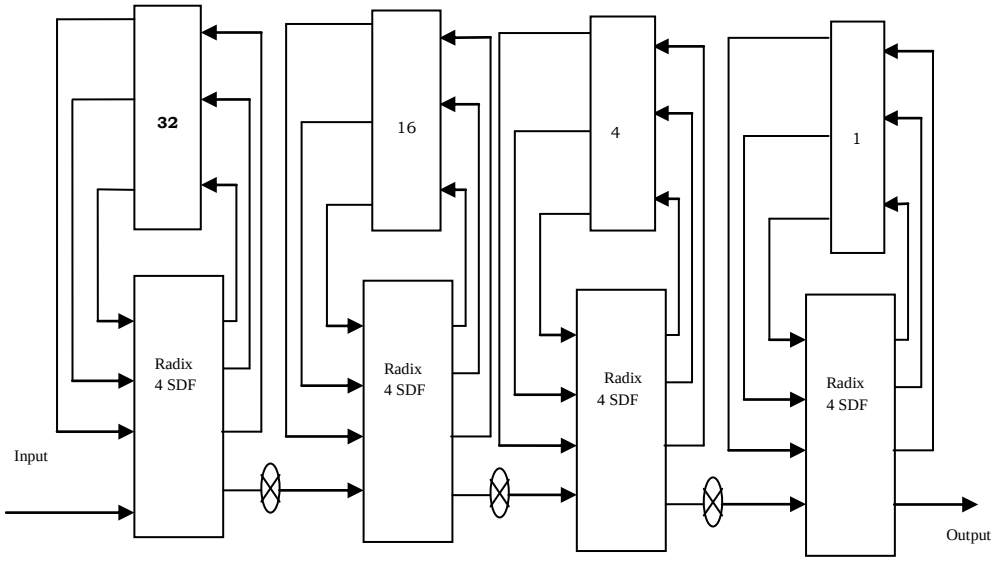

Fig. 1 64-point Radix-4 SDF FFT structure

The frequent operation of signal processing algorithms is the multiplication of two complex numbers. Let us consider the two complex numbers $(w+jx)$, $(y+jz)$ and their multiplication is defined by

$$(w+jx)(y+jz)=(wy-xz)+j(wz+xy) \quad (1)$$

The multiplication of two complex numbers produces the partial products and with the help of the subtractor and adder, the sum and difference between the partial products are calculated. The flow chart of the booth multiplier with KSPPA is shown in Fig. 2.

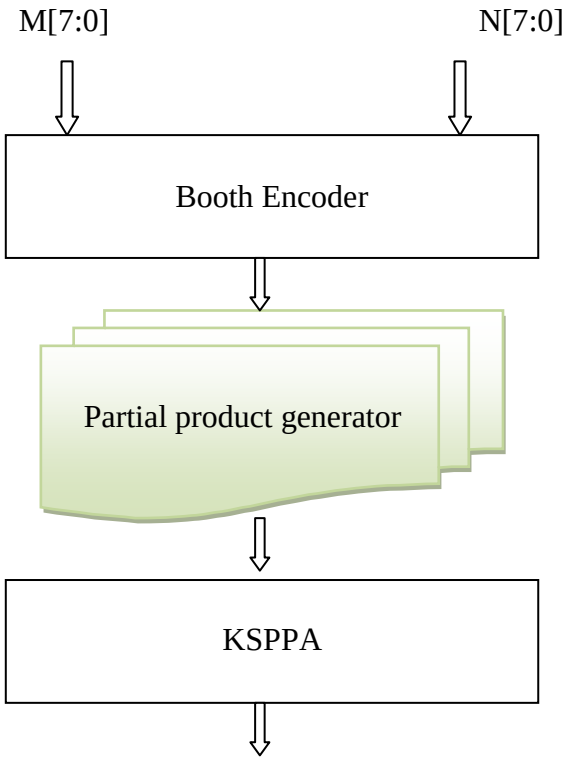

Fig. 2 Flow chart of Booth multiplier with KSPPA

The function of the booth multiplier with KSPPA is as follows. At first, the multiplier bits are encoded by the booth encoder. A partial product is formed by multiplying *M* and *N* where *M* and *N* are the 8-bit inputs. By using KSPPA the partial products are added which provides less delay and area. Figure 3 shows the 16-bit structure of the KSPPA. This adder is the form of carry look ahead adder. It provides the fewer fan out which supports the carry chain realizations. Each vertical stages produces the generate bit and propagate bit. In the pre processing stage, for propagate the Mi ^ Ni and Mi & Ni for generate is used.

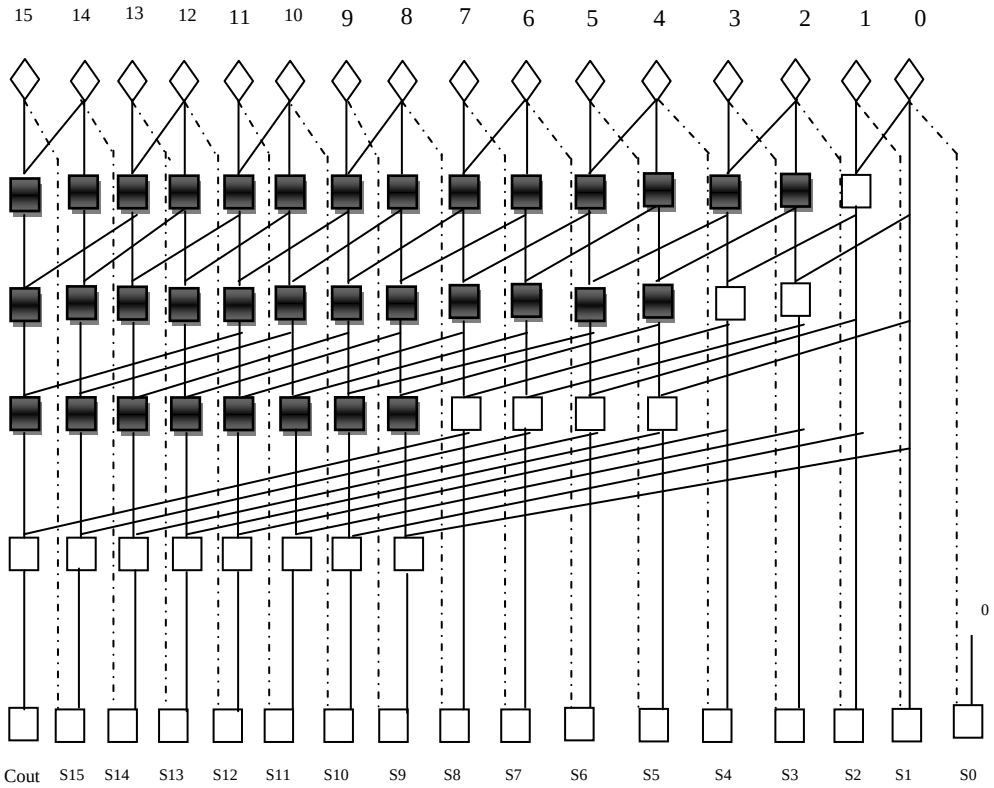

Fig. 3 16-bit structure of KSPPA

The next stage involves the calculation of carry equivalent to each bit. Finally, in the post processing stage, the sum is calculated. It produces carry in with $O(\log n)$ time with high performances where n is the input size.

III. RESULTS AND DISCUSSION

The 64-point Radix-4 SDF-FFT using booth multiplier with KSPPA is implemented using Xilinx 12.4 tool (Family Spartan 3, device XC3S50, package PQ208 and speed -5) and simulated using the MODELSIM 6.3c. The codes for the design are written using Verilog hardware description language. Figure 4 shows the simulation waveform for the KSPPA.

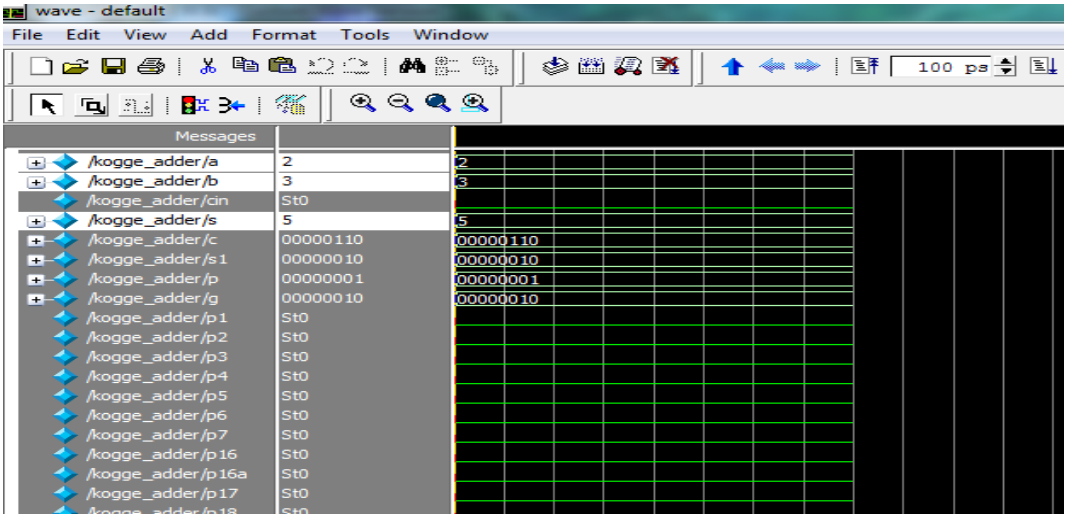

Fig. 4 Simulation waveform of the KSPPA

In Fig. 4, *a* and *b* are the inputs and *s* is the sum of *a* and *b*. The output of KSPPA is validated by the assigning any values to *a* and *b*. The waveform shows the output (s = 5) for the inputs (a = 2 and b = 3). Figure 5 shows the simulation waveform for the Modified booth multiplier.

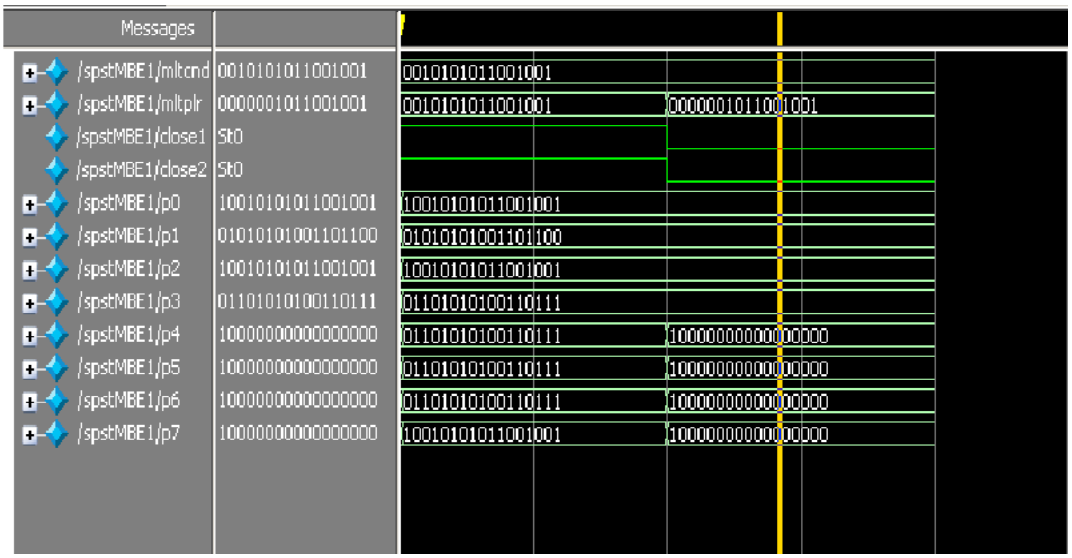

Fig. 5 Simulation Waveform of Booth multiplier with KSPPA

The multiplier and multiplicand are the two 8 bit inputs and the output is 16 bit. Figure 6 shows the simulation waveform of the 64-point radix-4 FFT using booth multiplier with KSPPA. The clock and reset are the default inputs and *data_real_in* and *data_imaginary_in* are the two 8 bit inputs, similarly *data_real_out* and *data_imaginary_out* are the outputs for the 64-point Radix-4

SDF-FFT. Figure 7 shows the performance analysis of the system in terms of area and delay in comparison with conventional FFT [16].

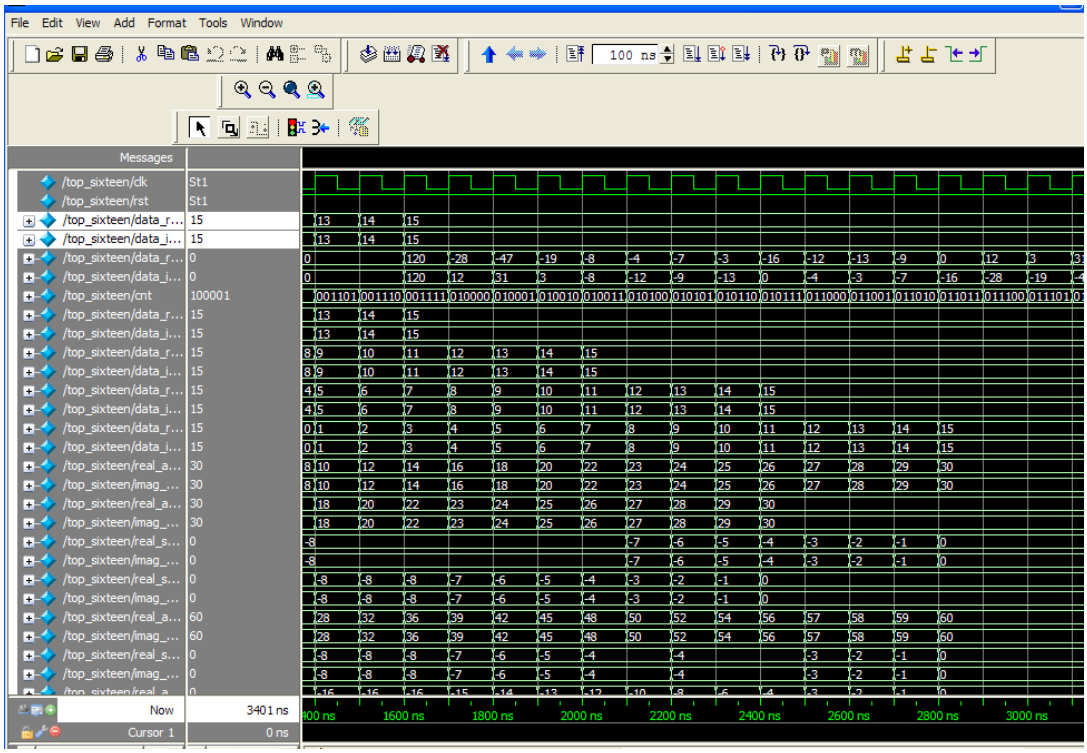

Fig. 6 Simulation waveform of the 64-point Radix-4 SDF-FFT architecture using booth multiplier with KSPPA

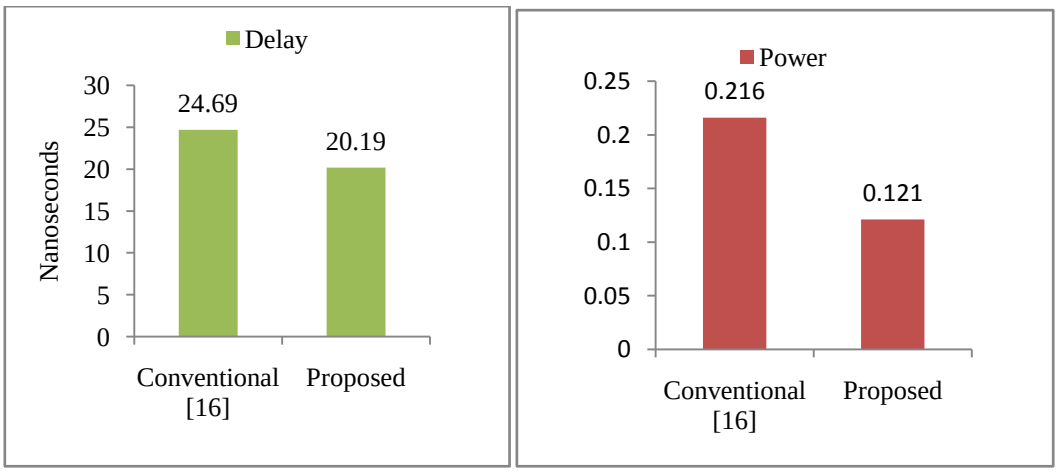

Fig. 7 Performance analysis of the system in terms of delay and Power (mw)

From the Figure 7, it is inferred that the delay and power of conventional method are high (24.69 ns and 0.216 mw) when compared to the system using booth multiplier with KSPPA.

IV. CONCLUSION

In this study, 64-point radix-4 SDF-FFT using booth multiplier with KSPPA is presented and implemented using Xilinx ISE 12.4 tool. In the booth multiplier design, KSPPA is used to reduce the delay. When compared to the conventional method, the system offers 18.22% reduction in delay and 43.98 %reduction in power. Results show that the overall multiplication time is reduced which in turn increases the speed of computation of FFT. In future, 64-point radix-4 SDF-FFT using booth multiplier with KSPPA can be realized in the OFDM transmitter and receiver design and their performances will be analyzed.

REFERENCES

[1]. B.C. Lin, Y.H. Wang, J.D. Huang, and J.Y. Jou, "Expandable MDC-based FFT architecture and its generator for high-performance applications". IEEE International SOC Conference, 2010, pp. 188-192.

[2]. M.P.R. Bhonde, R.D. Ghongade, and R.D. Sushir, "Design and Simulation of 32 and 64 Point FFT Using Multiple Radix Algorithms". Conference on Advanced Computing & Communication Technologies, Vol. 168, 2012, pp. 171.

[3]. C.F. Hsiao, Y. Chen, and C.Y. Lee, "A generalized mixed-radix algorithm for memory-based FFT processors", IEEE Transactions on Circuits and Systems II: Express Briefs, Vol.57, No.1, 2010, pp.26-30.

[4]. K. Ravi, and J. Lingaiah, "Hardware Efficient Mixed Radix-25/16/9 FFT for LTE Systems", International Journal of Scientific Engineering and Technology Research, Vol.5, No.38, 2016, pp.7960-7965.

[5]. C. Ingemarsson, P. Källström, F. Qureshi, and O. Gustafsson, "Efficient FPGA mapping of pipeline SDF FFT cores", IEEE Transactions on Very Large Scale Integration Systems, Vol.25,No.9, 2017, pp.2486-2497.

[6]. H.L. Groginsky, and G.A. Works, "A pipeline fast Fourier transform", IEEE Transactions on Computers, Vol.100,No.11, 1970, pp.1015-1019.

[7]. N.D. Vasumathy, and T. Vigneswaran, "A 128-Point FFT/IFFT Processor for MIMO-OFDM Transceivers–a Broader Survey", International Journal of Engineering and Technology, Vol. 8,No.5, pp.2154-2160.

[8]. J. Agarwal, V. Matta, and D. Arya, "Design and implementation of FFT processor using Vedic multiplier with high throughput", International journal of emerging technology and advanced engineering, Vol.3, No.10, 2013, pp.207-211.

[9]. G. Manikandan, and M. Anand, "Mixed Radix 4 & 8 Based SDF-SDC FFT Using MBSLS for Efficient Area Reduction", Indian Journal of Public Health Research & Development, Vol. 9,No.10, 2018,pp.1112-1116

[10]. A.K. Singh, and A. Nandi, "Design of Four point Radix-2 FFT structure on Xilinx", International Conference on Intelligent Computing and Control, 2017, pp. 1-4.

[11]. S.M. Joshi, "FFT architectures: a review", International Journal of Computer applications, Vol.116, No. 7, 2015, pp.0975-8887.

[12]. K. Thabitha, and D. JayaKumar, "VLSI based 64-point pipelined FFT using radix-4 combined SDF-MDC", International Journal of Pure and Applied Mathematics, Vol. 118, No. (20), pp.3509-3516.

[13]. O.A. Mukhanov, and A.F. Kirichenko, "Implementation of a FFT radix 2 butterfly using serial RSFQ multiplier-adders", IEEE Transactions on Applied Superconductivity, Vol. 5, No.2, 1995, pp.2461-2464.

[14]. A. Rajaram, S. Saravanan, and R. Vijaysai, "Novel Design and Implementation of Fast Fourier Transform (FFT) Using Fast Adders",Vol.5,No.2,2013,pp.542-546

[15]. M.B. Fonseca, E.A.C. da Costa, and J.B. Martins, "Design of power efficient butterflies from Radix-2 DIT FFT using adder compressors with a new XOR gate topology", Analog Integrated Circuits and Signal Processing, Vol. 73,No. 3, 2012,pp.945-954.

[16]. K.H. Rao, and C. Paul," Efficient Implementation of Radix 4 single path delay feedback (SDF) FFT Processors". Indian Journal of Sciences and Technology, Vol. 9, No.12, 2016, pp.1-13.