

DESIGN AND COMPARISON OF RADIX-2 FFT WITH DIFFERENT ADDERS AND MULTIPLIER COMBINATIONS

Nakka Sai Lokesh Kumar

Department of Electronic and Communication Engineering, AKRG COLLEGE OF ENGINEERING AND TECHNOLOGY,

Nallajerla, West Godavari Dist., AP, India.

Mr. K. Anji Babu, M-Tech

*Assistant Professor, Department of Electronic and Communication Engineering, AKRG COLLEGE OF ENGINEERING AND
TECHNOLOGY, Nallajerla, West Godavari Dist., AP, India.*

-----***-----

Abstract: There are many advancements in the design and implementation of Fast Fourier Transform (FFT) to improve its speed, reduction in area and power dissipation in view of its applications in signal processing, image processing and communication systems. In this paper radix-2 Decimation in-Time (DIT) FFT architecture is implemented. In this work, the processing element is designed with different adders and multipliers to implement four point radix-2 FFT architecture. The proposed architecture is implemented in verilog code and synthesized using Xilinx ISE 14.7. The performance analysis of FFT architecture is evaluated in terms of number of LUT's, number of flipflops, total number of slices, maximum path delay and dynamic power dissipation. It is evident from the results that the power dissipation is low in the FFT architecture implemented with Vedic Multiplier (VM) and Carry Select Adder (CSLA). Delay and area is reduced if FFT is implemented using booth multiplier (BM) and kogg stone adder (KSA).

Keywords--- *FFT, Radix-2, DIT, Butterfly, Multiplier, Adder.*

-----***-----

I INTRODUCTION

In the digital signal processing straight shifting and Fourier changes are the most used operations. The main operations like convolution and transforms had revolutionized the field of signal processing. The Fast Fourier transform is one of the techniques which reduces the computations required and increases the speed of the operation. The approaches are mainly concentrated on divide and conquer strategy to decrease the impact of sub problems like computational complexity and memory access. It became an alternative to the Discrete Fourier transform and had overcome its computational problems. The origin of many FFT algorithms, development and their implementation issues were discussed [1]. Different FFT architectures like memory based, cache based, sequential, parallel, array and pipeline are presented [2]. The parallel pipeline architectures[3] for FFT which involves proposal of folding transformation and register minimization techniques which computes FFT for real and complex values. The outputs obtained are in scrambled orders which are to be processed again leading to increase in computational blocks, resulting in increase in latency. The parallel pipeline architectures [3] for FFT which involves proposal of folding transformation and register minimization techniques for real and complex values. The outputs obtained are in scrambled orders which are to be processed again leading to increase in computational blocks, resulting in increase in latency.

The low cost variable length FFT employing pipeline shared memory architecture is discussed for different radix FFT algorithms [5]. An application in advanced video broadcasting is examined and discussed. A superior FFT library utilizing radix-

2 DIF is utilized for Single Guidance Various Information permitting equal gliding procedure on coterminous information in memory [6]. A 128- point FFT based radix-2 DIF is executed on Intel design. The co-ordinate rotational digital computer (CORDIC) was proposed by Volder in 1959. FFT architecture was implemented by CORDIC and multiplier for calculation of twiddle factor multiplication [7]. But CORDIC consumes more area and power than conventional multiplier which relies on low area and power consumption. The complex FFT processor dependent on FPGA is implemented using the pipe-line structure in butterfly module and ping-pong in information stockpiling. The processor depends on installed M4K rationale components and completed rapid fixed-point FFT activity [8]. Tale architecture [9] for memory- based FFT for sign dependent on radix-2 to DIF calculation is presented. In this design four memory banks are used to store samples and intermediate information in the processor. New pipeline hardware architecture [10] is designed for the computation of real valued FFT. It reduces the number of operations with respect to complex FFT which requires less area with high throughput and lower latency. The scrambling problem is solved here by pipelined architecture. A parallel pipelined architecture [11] for real valued signal is implemented. The imaginary parts of butterfly are used in the place of repeated operation for radix 23 and 24 butterfly structures which reduces hardware complexity, area, power and delay with increase in throughput.

The effective memory management [12] plans for memory-based engineering is used. Information movement conspire is utilized for the combination of numerous banks to

bring down the zone prerequisite and force scattering of FFT models. Further augmentation of memory to cache memory-based FFT design brings about more noteworthy reduction in power consumption. A novel scalable architecture [13] for setting up FFT calculation for the real-valued signal is implemented which eliminates the excess activity from stream chart. This architecture requires a complex radix-4 butterfly comprising of twelve complex adders and three complex multipliers. The multimode memory-based FFT processor for a medical framework named as Fourier Space Optical Coherence Tomography is reported [14]. It permits high throughput multimode FFT activities in the region effective arrangement. The proficiency of FPGA based OCT framework was additionally confirmed. An epic approach [15] to create pipeline FFT models for genuine esteemed signs depending on changing the stream charts of the FFT calculation with the end goal of both real and complex information is reported. An efficient mapping [16] of pipeline single path delay feedback FFT architecture involved the implementation done in Xilinx vertex-4 and vertex-6 devices. They were compared and results were evaluated. For an FFT operation three major blocks are considered namely adder /subtract, multiplier and memory/ register [17-19]. By choosing optimum blocks the speed, area and power dissipation of architecture can be determined.

In this paper FFT is implemented using radix-2 DIF architecture. Processing element is designed with different adders and multipliers to implement four point radix-2 FFT architecture. The performance analysis of FFT architecture is evaluated in terms of number of LUT's, number of flipflops, total number of slices, maximum path delay and dynamic power dissipation.

II. METHODOLOGY

In the proposed radix-2 FFT architecture, 4-point implementation is introduced using various adders and multipliers. Figure 1 depicts the 4-point FFT architecture overview in which radix-2 DIT FFT method is used to compute the mathematical calculations of FFT process. The N-point DFT of input sequence $X(n)$ is given by

$$x = 0, 1, \dots, N-1 \text{ Where } W_N^k \text{ is the twiddle factor}$$

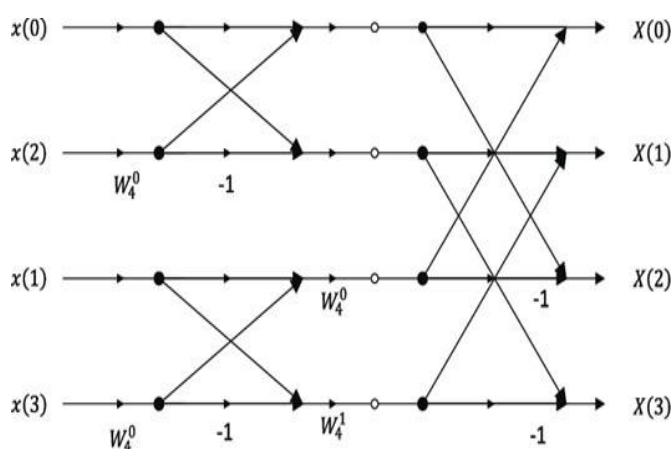


Figure 1: 4-Point radix-2 FFT Architecture

Operations carried out in the basic butterfly unit are represented in Figure 2. The butterfly unit mainly contains 2 operations addition/subtraction and multiplication. The second input in Figure 2 is multiplied with twiddle factor. Consider the twiddle factor $W = Ar + jBi$ and second element $B = Cr + jDi$ then multiplication operation results in Z forms one of the inputs to processing element along with A . Mainly addition and subtraction operations are performed on the inputs A and Z

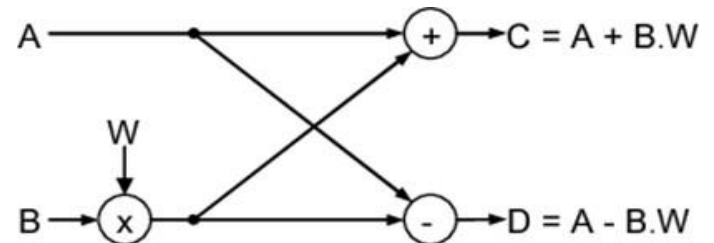


Figure 2: Basic Butterfly Unit

Structure of the processing element used in this work is given in Figure 3. In this structure different combinations of multipliers and adders are used to evaluate the performance of the proposed FFT. Multipliers used are Vedic Multiplier (VM), Booth Multiplier (BM) and Array Multiplier (AM). Carry Look Ahead Adder (CLA), Carry Select Adder (CSLA), Carry Skip Adder (CSA) and KoggStone Adder (KSA) are used.

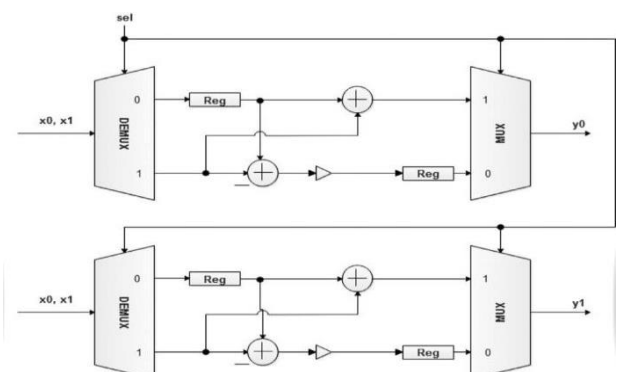


Figure 3: Processing Element

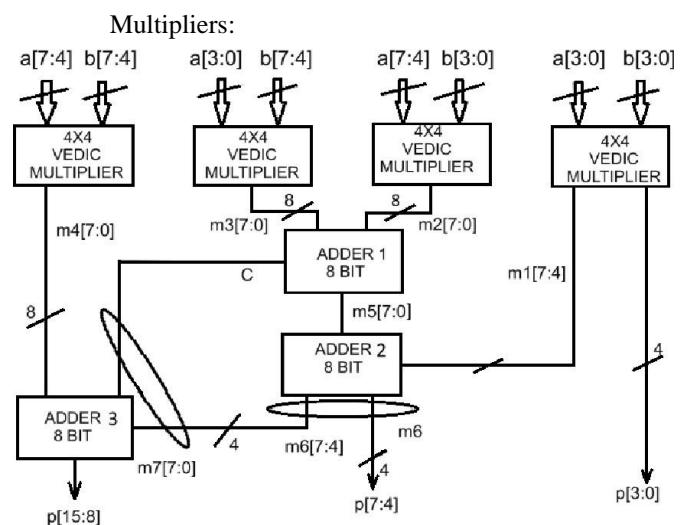
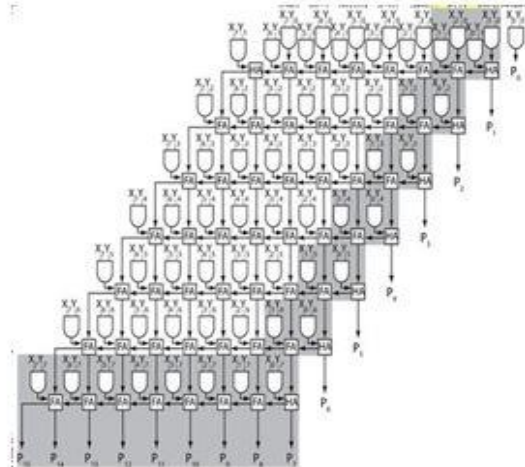
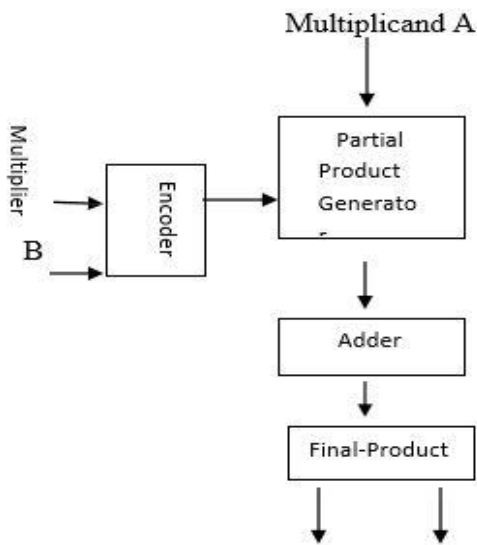


Figure 4: Vedic Multiplier

Multiplicand A

Product A*B



Product A*B |
Figure 5: Booth Multiplier

Figure 6: Array Multiplier

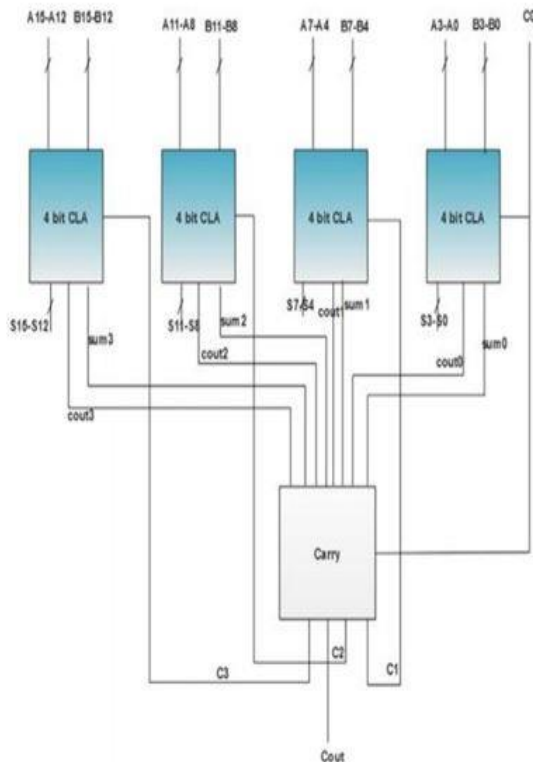


Figure 7: Carry Look Ahead Adder

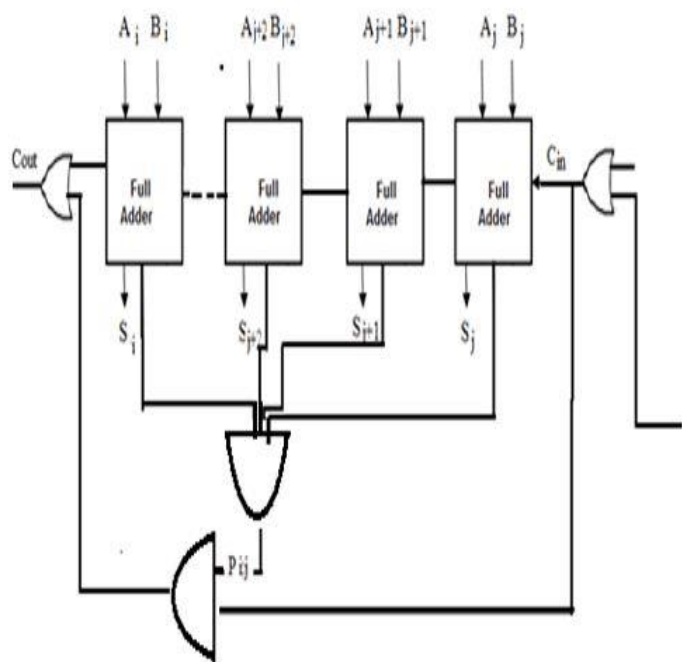


Figure 8: Carry Skip Adder

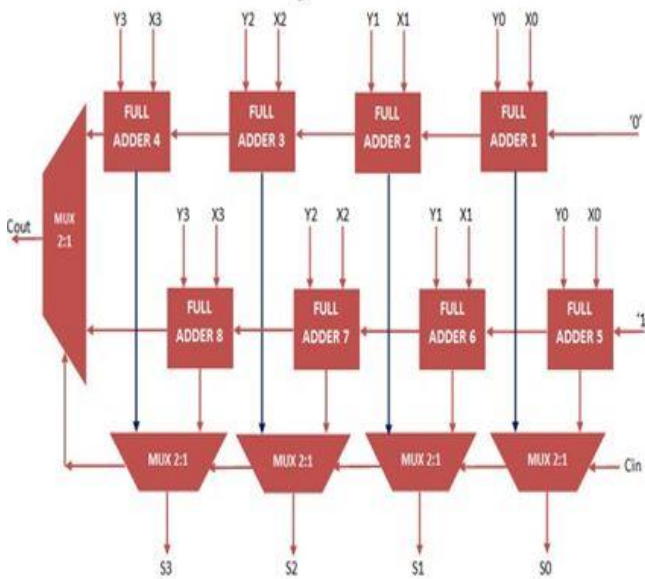


Figure 9: Carry Select Adder

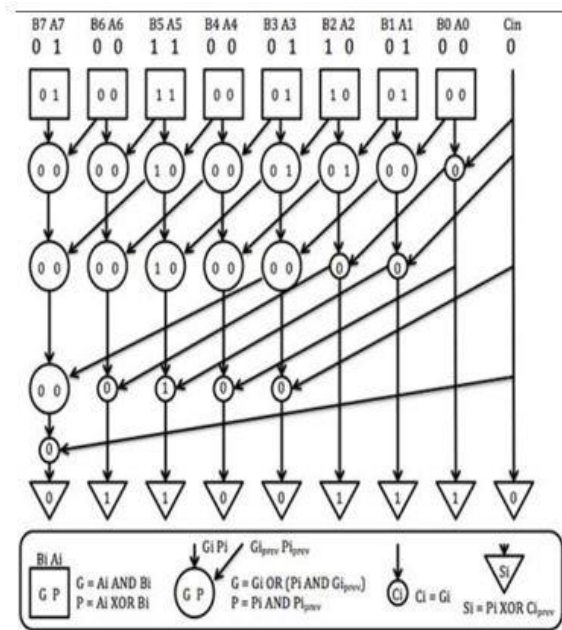


Figure 10: KoggeStone Adder

The basic structures of the multipliers used in the processing element are given in above Figure 4 to 6.

Adders:

The basic structures of adders implemented in the processing element for addition/subtraction operation are presented in Figures 7 to 10.

III.RESULTS AND DISCUSSION

The proposed radix-2 FFT four point architecture is implemented using Verliog code. Xilinx ISE

14.7 is used to synthesize the verilog code, verify the simulation diagram which is used to evaluate the performance characteristics like Maximum Path Delay, LUT's, Registers, Slices, Dynamic Power and Frequency. The performance characteristics of FFT for different combinations of multipliers and adders are tabulated in Tables. 1, 2, 3, 4. The RTL schematic of the 4-point FFT is shown in Figure11. The timing representation of the proposed method is shown in Figure 12 and 13.

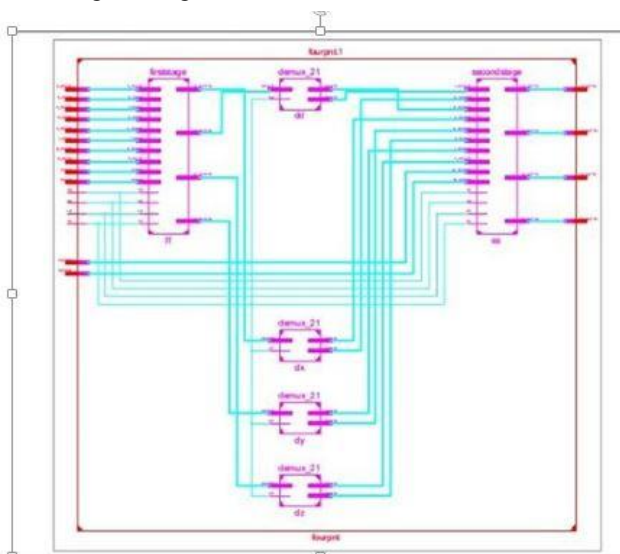


Figure 11: RTL schematic of 4-point

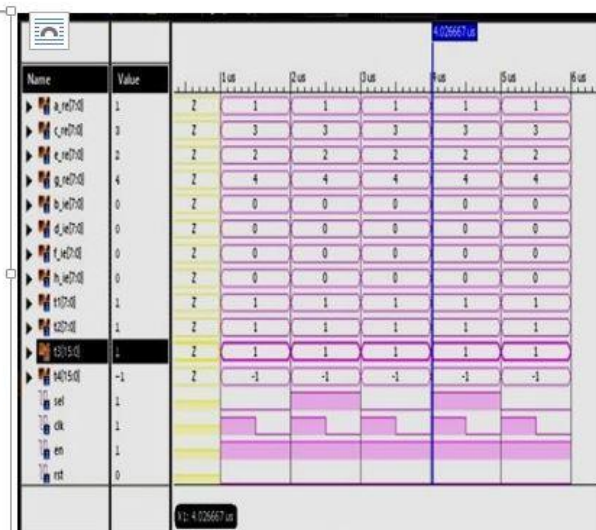


Figure 12: Timing Simulation

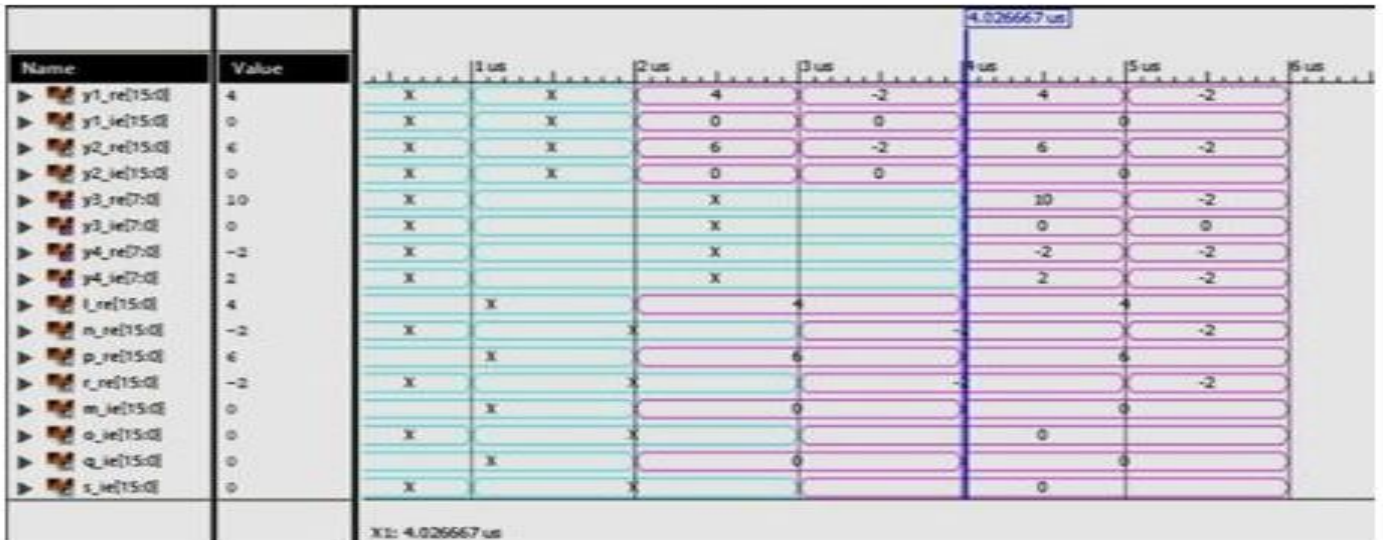


Figure 13 Timing Simulation (Continued)

Table 1 Comparison of FFT parameters for different multipliers using CSLA

Parameters	Multipliers		
	Vedic	Booth	Array
Maximum Path Delay	6.419 ns	6.406 ns	6.44 ns
Dynamic Power (in mw)	83.38	88.25	91.77
No of Occupied Slices	642/63,168	631/63,168	662/63,168
Total Input LUTs	1,045/1,26,336	1,031/1,26,336	1,059/1,26,336
No of Registers	341/1,26,336	333/1,26,336	352/1,26,336

Table 2 Comparison of FFT parameters for different multipliers using CLA.

Parameters	Multipliers		
	Vedic	Booth	Array
Maximum Path Delay	6.842 ns	6.829 ns	6.922 ns
Dynamic Power (in mw)	84.99	90.32	92.53
No of Occupied slices	592/63,168	587/63,168	616/63,168
Total Input LUTs	982/1,26,336	973/1,26,36	1002/1,26,36
No of registers	328/1,26,336	323/1,26,36	338/1,26,336

Table 3 Comparison of FFT parameters for different multipliers using KSA

Parameters	Multipliers		
	Vedic	Booth	Array
Maximum Path Delay	6.409 ns	6.345 ns	6.436 ns
Dynamic Power (in mw)	90.82	96.53	101.86
No of Occupied slices	664/63,168	652/63,168	679/63,168
Total Input LUTs	1,062/1,26,336	1,045/1,26,336	1,078/1,26,336
No of registers	361/1,26,336	352/1,26,336	374/1,26,336

Table 4 Comparison of FFT parameters for different multipliers using CSA

Parameters	Multipliers		
	Vedic	Booth	Array
Maximum Path Delay	6.439 ns	6.421 ns	6.459 ns
Dynamic Power (in mw)	86.02	91.24	93.68
No of Occupied Slices	649/63,168	643/63,168	664/63,168
Total Input LUTs	1048/1,26,336	1,036/1,26,336	1,064/1,26,336
No of registers	349/1,26,336	341/1,26,336	363/1,26,336

The below charts are pictorial representation of FFT parameters

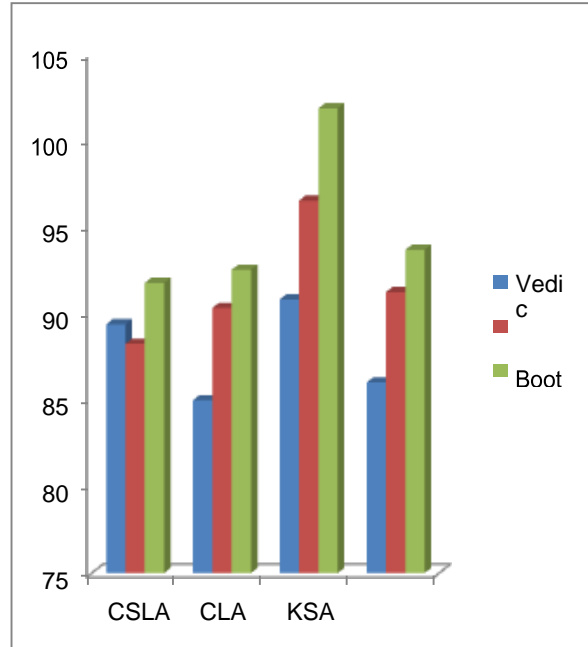
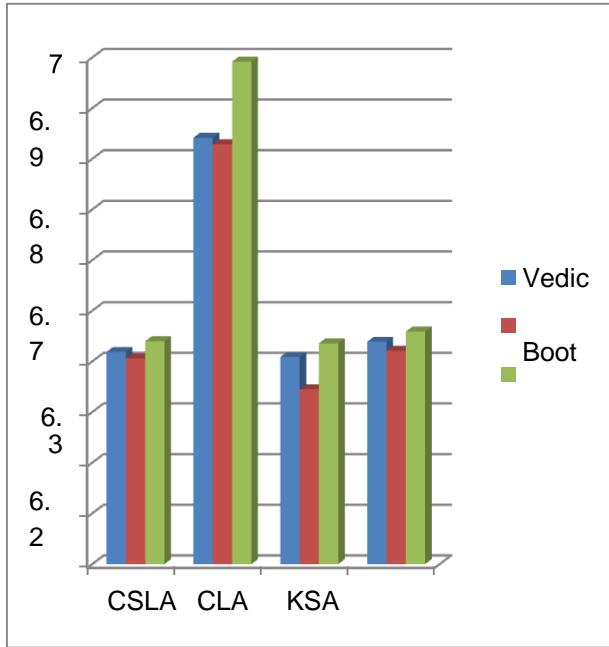


Figure14: Comparison of maximum path delay in ns **Figure 15: Comparison of Dynamic power in mW**

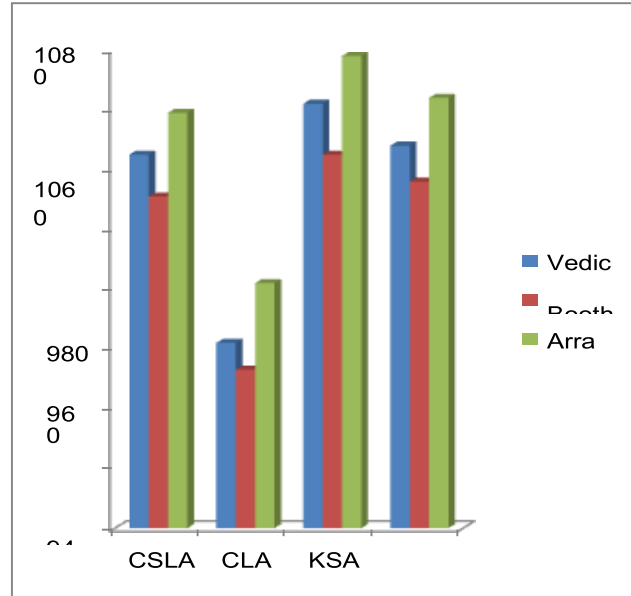
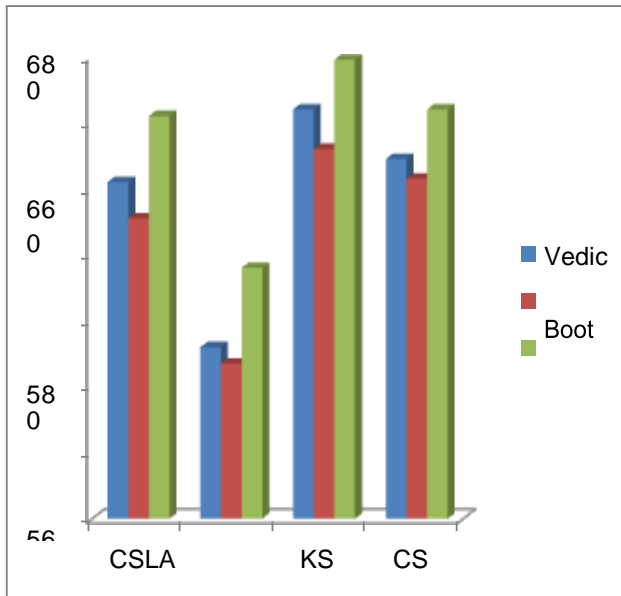


Figure 16: Comparison of Number of occupied slices

Figure 17: Comparison of Total Input LUT

AND ENGINEERING TRENDS

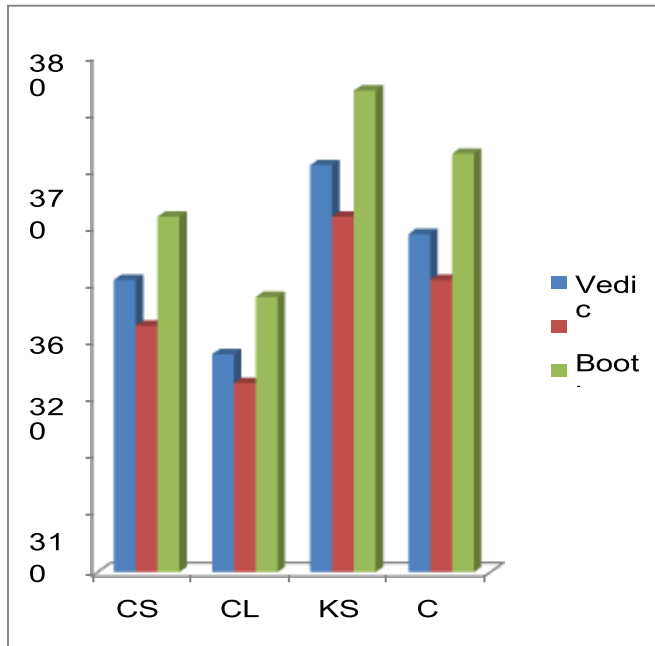


Figure 18: Comparison of Number of registers

Delay Profile:

Comparison of maximum path delay achieved for the proposed FFT architecture with different combinations of adders and multipliers is given in Figure 14. The FFT architecture implemented with the combination of Booth multiplier and KSA achieves the lowest delay of 6.345ns compared to all other combinations implemented.

Power Profile:

In Figure 15 comparison of dynamic power in mW consumed for FFT implementation is presented. It is evident from the results that Vedic multiplier with CSLA consumes least power of 83.38mW compared to all other combinations implemented. From the results obtained and presented four point FFT using the combination of KSA and booth multiplier can be preferred if speed is a major concern, combination of Vedic multiplier and CLA for low power dissipation and Booth multiplier in combination with CLA for less area.

Area Profile:

In Figure 16 comparison of number of slices occupied for FFT implementation is presented. It is evident from the results that Booth multiplier in combination with CLA occupies less number of slices i.e. 587 slices compared to all other combinations implemented.

Comparison of LUT's contained for the proposed FFT architecture with different combinations of adders and multipliers is given in Figure 17. The FFT architecture implemented with the combination of Booth Multiplier and CLA contains a least number of 973 LUT's compared to all other combinations implemented. The Number of registers for

the proposed FFT architecture with different combinations of adders and multipliers comparison is given in Figure 18. The FFT architecture implemented with the combination of CLA and Booth multiplier contains a least number of registers in order of 323 compared to remaining combinations.

IV CONCLUSION AND FUTURE SCOPE

In this paper, Radix-2 4-point FFT architecture is implemented using different combinations of adders like CLA, CSLA, CSA, KSA and multipliers like Vedic, Array, and Booth in Xilinx software using Verilog code. Various performance parameters for different combinations of adders and multipliers are evaluated. It was found that combination of KSA and Booth multiplier results in less delay, combination of Vedic multiplier and Carry look ahead adder for low power dissipation and Booth multiplier in combination with CLA for less area. This work can be further extended to implement higher order FFT Architectures using Complex adders and Multipliers to operate upon floating data.

REFERENCES

- [1] Duhamel and Vetterli.M “Fast Fourier transforms: a tutorial review and a state of the art”, Signal Process. 1990, 19, (4), pp. 259–299.
- [2] Shubhangi and M. Joshi “FFT Architectures: A Review” International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 7, April 2015
- [3] M.Ayinala, M.Brown, K.K.Parhi, “Pipelined parallel FFT architectures via folding transformation”, IEEE Trans. VLSI Syst. 20 (2012) 1068–1081.
- [4] P.Y.Tsai, C.Y.Lin, “A generalized conflict-free memory addressing scheme for continuous-flow parallel-processing FFT processors with rescheduling”, IEEE Trans. VLSI Syst. 19 (2011) 2290–2302.
- [5] Kisun Jung and Hanho Lee, "Low-cost variable-length FFT processor for DVB-T/H applications", Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on, IEEE, 2010.
- [6] Wang Xu, Zhang Yan and Ding Shunying, "A high performance FFT library with single instruction multiple data (SIMD) architecture", Electronics, Communications and Control (ICECC), 2011 International Conference on, IEEE, 2011.
- [7] R.Bhakhavatchalu, N.A.Kareem and J.Arya, “Comparison of Reconfigurable FFT Processor Implementation Using CORDIC and Multipliers”, Recent Advances in Intelligent Computational Systems (RAICS), 2011.
- [8] S. Zhou, X. Wang, J. Ji and Y.Wang, “Design and implementation of a 1024-point high-speed FFT processor based on the FPGA”, in Image and Signal Processing (CISP), 2013 6th International Congress on, 2, 2013, pp.

1112–1116.

- [9] Zhen-Guo Ma, Xiao-Bo Yin and Feng Yu, "A novel memory-based FFT architecture for real-valued signals based on a radix-2 decimation-in-frequency algorithm", IEEE Trans. Circuits Syst. II 62 (9) (2015) 876–880. P. Heckbert, Fourier transforms and the fast fourier transform (FFT) algorithm", Comput. Graph. 2 (1995) 15–463.
- [10] G. Mario, K.K. Parhi and J. Grajal, "A pipelined FFT architecture for real-valued signals", IEEE Trans. Circuits Syst. 56 (2009).
- [11] V.Suganya and C.Paramasivam, "Parallel Pipelined FFT Architecture for Real Valued Signals," IEEE WiSPNET 2016 conference.
- [12] B. Kim and S.H. Kong, "Design of FFT-based TDCC for GNSS acquisition", IEEE Trans. Wirel. Commun. 13 (2014) 798–2808.
- [13] H.F. Luo, Y.J. Liu and M.D. Shieh, "Efficient memory-addressing algorithms for FFT processor design", IEEE Trans. VLSI Syst. 23 (2015) 2162–2172.
- [14] S.N. Tang, F.C. Jan, H.W. Cheng, C.K. Lin and G.Z. Wu, "Multimode memory-based FFT processor for wireless display FD-OCT medical systems", IEEE Trans. Circuits-I 61 (2014) 3394–3406.
- [15] A. Manohar and K.K. Parhi, "FFT architectures for real-valued signals based on radix 2^3 and radix 2^4 algorithms", IEEE Trans. Circuits-I 60 (2003) 2422–2430.
- [16] C. Ingemarsson , P. Källström , F. Qureshi and O. Gustafsson , "Efficient FPGA mapping of pipeline SDF FFT cores", IEEE Trans.VLSI Syst. 25(9) (2017) 2486–2497 .
- [17] Rajasekhar Turakaa and Dr.SatyaSai Ram M. b "Low power VLSI implementation of real fast Fourier transform with DRAM-VM-CLA" Elsevier Microprocessors and Microsystems 69 (2019) 92–100.
- [18] Basavoju Harish, K. Sivani and M.S.S. Rukmini "Plan and Execution Correlation among Different kinds Topologies for Adder" Third Global Gathering on Analysing Ideologies and Correspondence (ICCMC 2019).
- [19] Khurajam Nelson Singh and H. Tarunkumar "A survey induced with detailed review on different multipliers plans with in detailed designs in VLSI" 2015 Yearly IEEE India gathering (INDICON).