Efficient VLSI Design of Fast Fourier Transform using Reversible Gate

Akash[1], Prof. Satyarth Tiwari[2]

M. Tech. Scholar, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal[1]

Guide, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal[2]

Abstract:- The FFT is enumerate is DFT and DFT is enumerate is consecutive way, it accomplishes continuous application with constant preparing when the information is persistently taken care of through the processor. Included paper, joined is radix-2 butterfly (R_2B), R_4B & R_8B components based single path delay feedback (SDF) technique and reversible gate, for diminishing the computational stages and for decreasing the equipment use than the R_2B and R_4B FFT. The implemented SDF technique has single delay commutators at one stage without exception. N/2 point is consecutive controlled in consequence of delay component. The proposed technique has less number of multipliers and the more modest number of computational stages and butterfly components than the Radix-2 & 4 FFT.

Keywords: - R_2B, R_4B, R_8B, SDF, FFT

I. INTRODUCTION

Electronic sign planning is the legitimate control of a data sign to modify or update it somehow. It is portrayed by the depiction of discrete time, discrete rehash, or other discrete region signals by a movement of numbers or pictures and the preparing of these signs [1].

The Fourier Transform is a widely used method in signal processing to estimate spectral content of any signal. The Fourier Transform when applied to an aperiodic discrete signal rather than a continuous signal is called Discrete Time Fourier Transform (DTFT). But DTFT of an aperiodic discrete signal is continuous in frequency domain. Hence to use DTFT in computers, we sample the DTFT. This sampled DTFT is called Discrete Fourier Transform (DFT) [2].

The Fast Fourier Transform (FFT) and its Inverse Fast Fourier Transform (IFFT) are essential in the field of digital signal processing (DSP), widely used in communication systems, especially in Orthogonal Frequency Division Multiplexing (OFDM) systems, wireless-LAN, ADSL, VDSL systems and Wi-MAX. Apart from the applications, the system demands high speed of operation, low power consumption, reduced truncation error and reduced chip size. By considering these facts, we proposed the ROM-less processor with Single-path Delay Feedback (SDF) pipeline architecture and modified booth width multiplier. The SDF pipelined architecture is used for the high throughput in FFT processor. There are three types of pipeline structures; they are Single path Delay Feedback (SDF), Single-path Delay Commutator (SDC) and Multi-path Delay Commutator. The advantages of Single path Delay Feedback (SDF) are (1) This SDF architecture is very simple to implement the different length FFT. (2) The required registers in SDF architecture is less than MDC and SDC architectures. (3) The control unit of SDF architecture is easier. We implement the processor in SDF architecture with radix-8 algorithm. There are various algorithms to implement FFT, such as radix-2, radix-4 and split-radix with arbitrary sizes, radix-2 algorithm is the simplest one, but its calculation of addition and multiplication is more than radix4's. Though being more efficient than radix-2, radix-8 only can process 8 n-point FFT. The radix-8 FFT equation essentially combines three stages of a radix-2 FFT into one, so that one third as many stages are required [3, 4].

II. RADIX FFT

Radix-2

A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and its inverse. Fourier analysis converts time (or space) to frequency and vice versa; an FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. As a result, fast Fourier transforms are widely used for many applications in engineering, science, and mathematics. Show the butterfly operations for radix-2 DIF FFT in figure 1 and figure 2. The radix-2 algorithms are the simplest FFT butterfly algorithm.

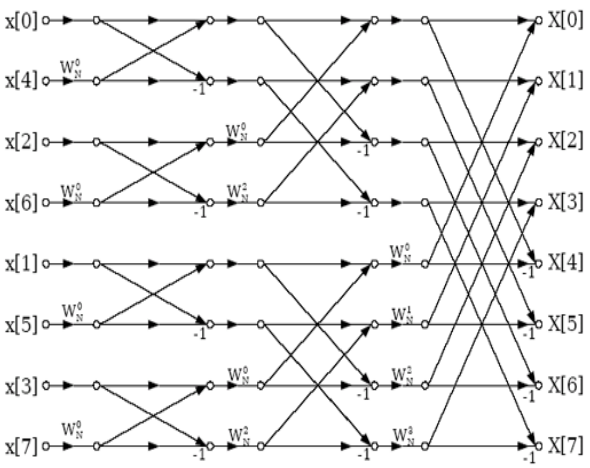

Figure 1: Radix-2 Decimation in Time Domain FFT Algorithm

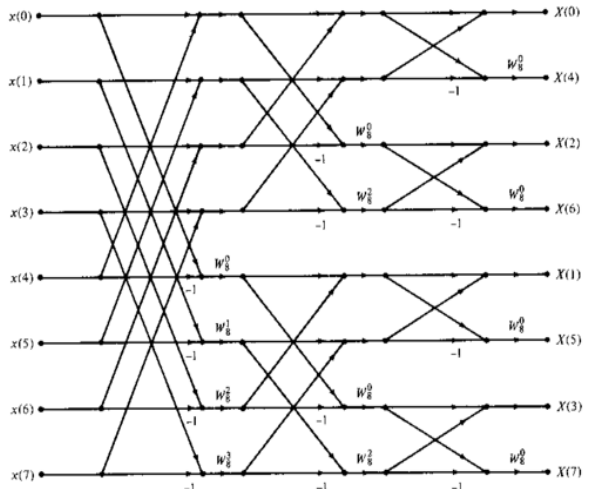

Figure 2: Radix-2 Decimation in Frequency Domain FFT Algorithm

Radix-4

In last three decades, various FFT architectures such as single-memory architecture, dual memory architecture, pipelined architecture, array architecture and cache memory architecture have been proposed. In order to improve the power reduction, we propose a radix-4 64-point pipeline FFT/IFFT processor. In order to speed up the FFT computations, more advanced solutions have been proposed using an increase of the radix. The radix-4 FFT algorithm is most popular and has the potential to satisfy the current need.

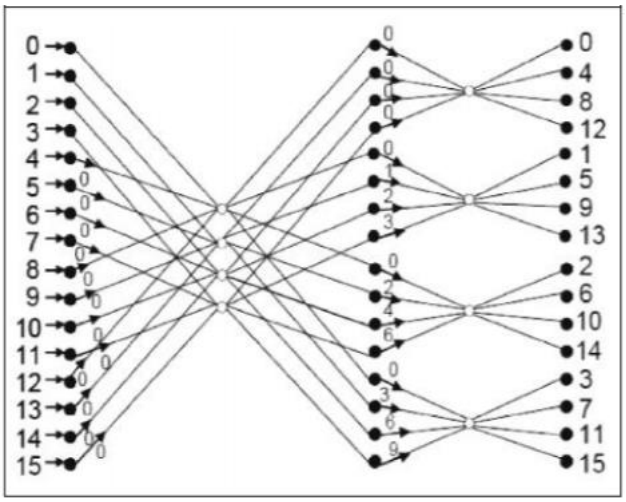

Figure 3: Flow Graph of a 16-Point Radix-4 FFT Algorithm

To calculate 16-point FFT, the radix-2 takes log216 = 4 stages but the radix-4 takes only log416 = 2 stages. A 16-point, radix-4 decimation-in-frequency FFT algorithm is shown in Figure 1. Its input is in normal order and its output is in digit-reversed order. It has exactly the same computational complexity as the decimation-in-time radix-4 FFT algorithm.

Radix-8

Radix-8 is one FFT algorithm which is analyzed and surveyed in order to enhance the throughput of operation by decreasing the calculation and computations; this will be achieved by applying the base to 8. Given same value if base incremented the power must drastically reduce. The value of stages is noticeable reduction by 75% since N=82 i.e., only two stages. By implementation of the FFT algorithm the functional and computational complexity decreases to where indicate the Radix ‘r’ FFT. Radix ’r’ FFT can be progressively obtained from DFT by decomposing the N point DFT into groups of iteratively associated r-point transformation and x(n) is powers of r. In case of Radix-8 algorithm, r value 8. The DIF Radix-8 FFT repetitively partitions a DFT into eight chuck-lengths DFT of sets of individual eighth sample. The outputs of distributed and smaller FFTs are reutilized into computing several outputs, which significantly bring down overall computational expense. The Radix-8 Decimation In Time (DIT) and Decimation In Frequency(DIF) Fast Fourier Transform (FFTs) obtain their pace by reutilizing the result of distributed smaller, intermediary calculation and computation to achieve total and multiple DFT results.

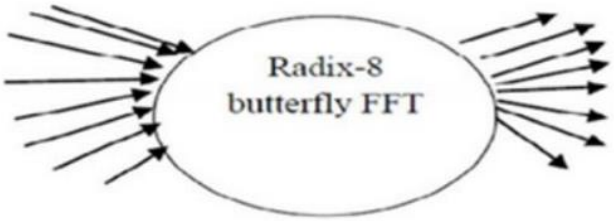

Fig. 4: Radix-8 Butterfly FFT

III. PROPOSED METHODOLOGY

The consolidated R_2B, R_4B & R_8B based SDF FFT has been planned in this proposed work. The consolidated Radix of FFT design has a lesser measure of computational way and furthermore improves the exhibitions of FFT processor. SDF design, the info information successions are going through one single way. The butterfly preparing component plays out the calculation on the information. The expansion and deduction activity is done in butterfly components. The changed convey select viper circuit is utilized for snake activity in this engineering. This snake structure is extremely productive in this design. The structure of joined R_2B, R_4B & R_8B FFT is appeared in Fig. 5. The engineering of 16 point SDF FFT is appeared in Fig. 6.

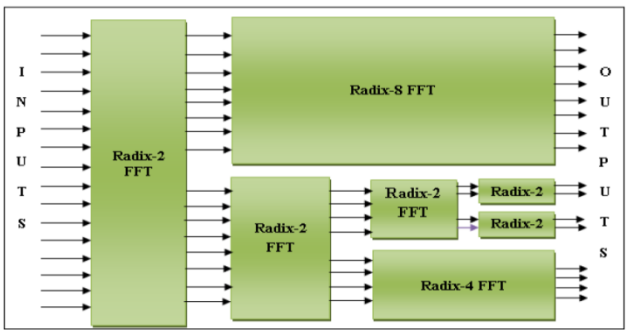

Fig. 5: Structure of Combined R_2B, R_4B & R_8B FFT

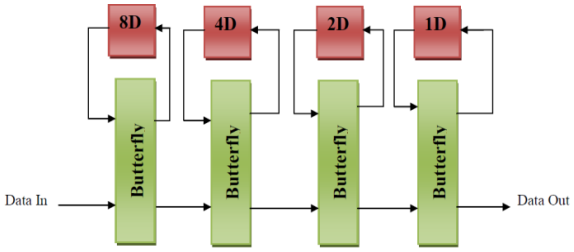

Fig. 6: Architecture of 16 Point SDF FFT

The architecture of improved Radix-8 algorithm involves the resolution of the output terms as complex and real. The inputs are taken as 8-bit vector value for the implementation for the case of real data. The inputs are forwarded using butterfly Radix-2 implementation which helps to improve the combinational delay path. The Radix-2 butterfly module is implemented and instances are called every time for the small computation [2]. For the simplicity of computation, the variables have been scaled up to overcome the decimal notation wherein the analysis is continued in software wherein the simple division by base 10 will be much faster and efficient in the computation of the final value of in frequency domain [3]. Each stage final values are updated to register and the twiddle factor have been previously computed and stored in register are invoked for the finding the value. This invoked value is multiplied with the value obtained by the Radix-2 instantiation. Each stage value is carried onto the further stage until the final stage is obtained. At the final stage we obtain the complete 8-point Radix-8 FFT for the computation of DFT in optimized and high speed technique. The multiplication is made in the stages of the addition to reduces the area utilization. The complex output is provided as the individual value apart from real values. The final staged values are driven to software or the next macro for further computation and analysis.

Reversible DKG Gate:-

The RDKG is presented by figure 7. RDKG is 4×4 system representation. RDKG is working on full adder and full sub-tractor when first input assume '0' and '1'.

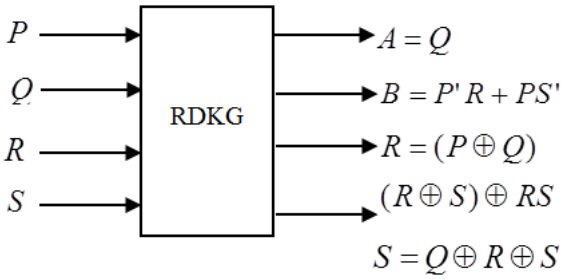

Fig. 7: RDKG Gate

IV. RESULT

The proposed improved optimized Radix-8 algorithm is designed using Verilog Hardware Description Language and Behavioral style of coding. The design and analysis are done for 8-point 8-bit inputs. It is simulated using Xilinx XST for Spartan 6 family, XC6SLX4 device with speed -3 and package TQG144 FPGA. The functionality is verified for the inputs given below and timing is being validated. Inputs are created by Verilog Hardware Description Language (HDL) test bench.

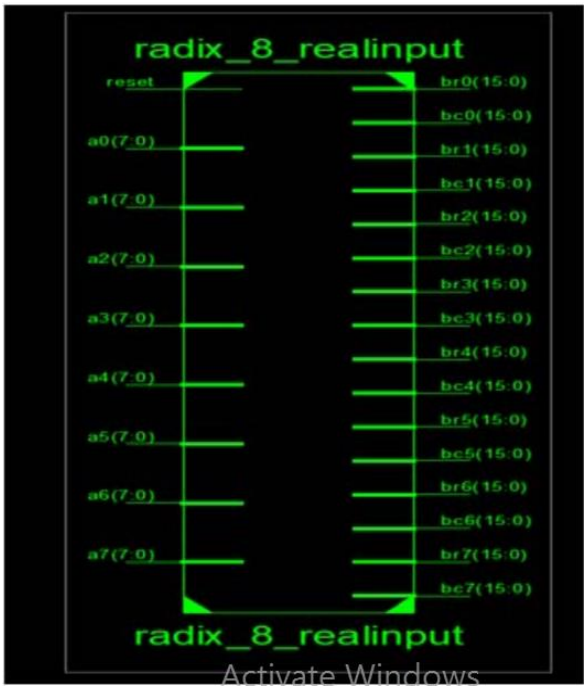

Fig. 8: Simulation result of proposed combined R_2B, R_4B and R_8B based SDF FFT

Case :1 Input :x(n) = {1,2,2,2,1,0,0,0}

Output :X(k)= {8, -4.84i,0, -0.84i,0,0.84i,0,4.84i}

Case 2: Input :x(n) = {1,1,1,1,1,1,1,1}

Output :X(k)= (8,0,0,0,0,0,0,0}

Case 3: Input :x(n) = {1,2,3,4,5,6,7,8}

Output :X(k)= {36, -4+9.68i, -4+4i, -4+1.68i, -4, -4-1.68i, -4-4i, -4-9.68i}

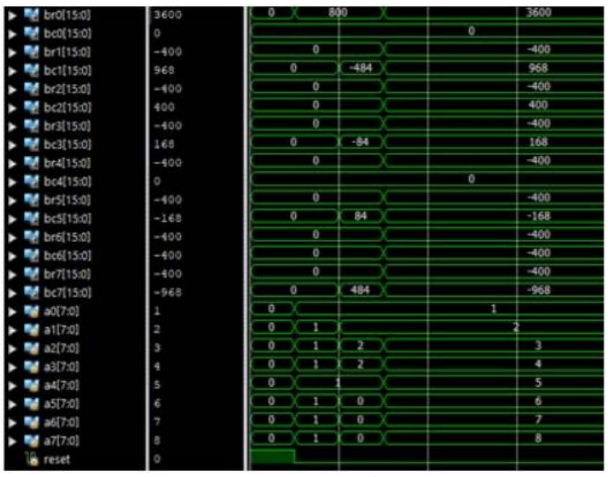

Fig. 9: Output Waveform

The view technology schematic (VTS) of 8-point FFT using DKG gate is shown in figure 10.

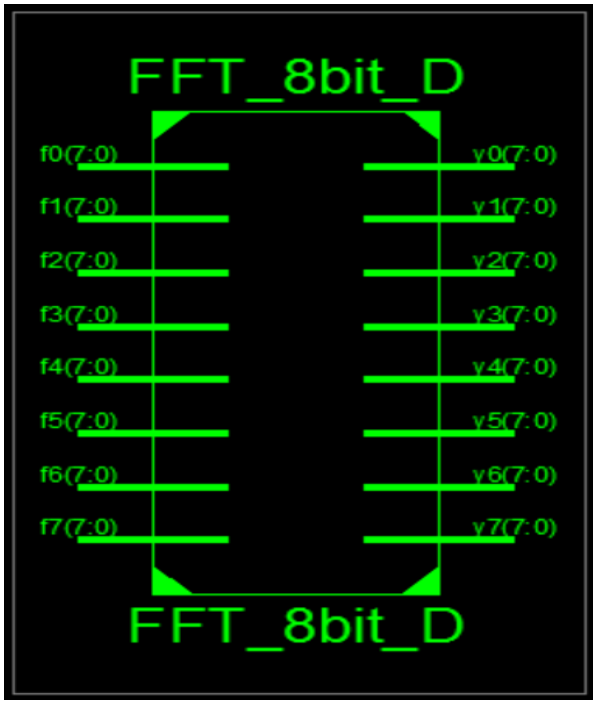

Fig. 10: VTS of 8-point FFT using DKG Gate

This figure 11 is the view technology schematic (VTS) of 8-point FFT. In this, the input is f0-f7 and the outut y0-y7.

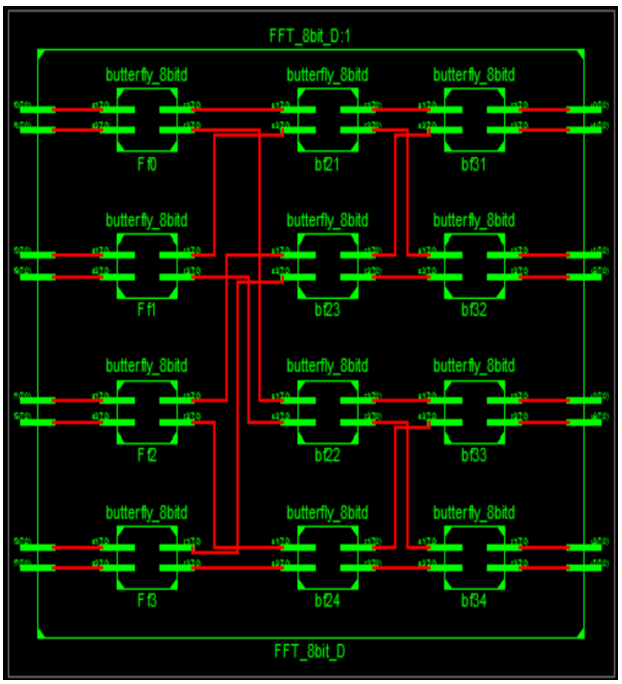

Fig. 11: RTL View of DIT 8-point FFT using DKG Gate

This figure 5.11 is the RTL Schematic of 8-point FFT it depends on the view technology. It shows all the components inside the VTS.

Fig. 12: VTS of DIT 8-point FFT using DKG Gate

This is the VHDL test-bench (VTS) of 8-point FFT. This figure 5.28 shows the waveform.

V. CONCLUSION

The pipelined Radix-2 Single Path Delay Feedback (SDF) – Single Path Delay Commutator (SDC) FFT using Modified Carry Select Adder has been proposed. The carry select adder was modified by reducing the full adder structure to reduce the hardware slices, delay and power consumption. The reduced full adder is designed using less number of gates compared to the conventional Full adder. These reduced adders are applied to the carry select adder to analyze and improve the performance. The modified carry select adder is incorporated into Radix-2 SDF-SDC FFT processor. The main goal of this research work is to reduce the processing time and improve the speed of the FFT processor.

REFERENCES

[1] Syeda Farhat Sultana and Basavarja Patil, “Area efficient VLSI architecture for reversible radix_2 FFT algorithm”, International Conference on Emerging Smart Computing and Informatics (ESCI), IEEE 2021.

[2] Navin Nadar, Moksha Mehta, Kaushal Bhat, Joshua Mendes and Dr. Ravindra Chaudhari, “Hardware Implementation of Pipelined FFT using Polyphase Decomposition”, International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), IEEE 2021.

[3] Shashidhara. K. S and H.C. Srinivasaiah, “Low Power and Area efficient FFT architecture through decomposition technique”, International Conference on Computer Communication and Informatics (ICCCI -2017), Jan. 05 – 07, 2019, Coimbatore, INDIA.

[4] Fahad Qureshi, Jarmo Takala, Anastasia Volkova, Thibault Hilaire, “Multiplier-less Unified Architecture

for Mixed Radix-2/3/4 FFTs", 2017 25th European Signal Processing Conference (EUSIPCO), IEEE 2019.

[5] Guilherme Ferreira; Leandro M. G. Rocha; Eduardo Costa and Sergio Bampi, "Combining m=2 Multipliers and Adder Compressors for Power Efficient Radix-4 Butterfly", 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE 2019.

[6] S. Aseeri "State-of-the-Art FFT: Algorithms Implementations and Applications" Reflections on the 2018 SIAM Conference on Parallel Processing for Scientific Computing pp. 1-3 Jun 2018.

[7] E. Konguvel and M. Kannan "A Survey on FFT/IFFT Processors for Next Generation Telecommunication Systems" Journal of Circuits Systems and Computers vol. 27 no. 03 pp. 1830001 2018.

[8] W. Choi Kang and J. Park "Embedded DRAM- Based Memory Customization for Low-Cost FFT Processor Design" IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2017.

[9] Fahad Qureshi, Muazam Ali, and Jarmo Takala, "Multiplierless Reconfigurable Processing Element for Mixed Radix-2/3/4/5 FFTs", 978-1-5386-0446-5/17/$31.00 ©2017 IEEE.

[10] R. H. Neuenfeld M. B. Fonseca E. A. C. da Costa and J. P. Oses "Exploiting addition schemes for the improvement of optimized radix-2 and radix-4 fft butterflies" 2017 IEEE 8th Latin American Symposium on Circuits Systems (LASCAS) pp. 1-4 2017.

[11] B. Silveira G. Paim B. Abreu M. Grellert C. M. Diniz E. A. C. da Costa et al. "Power-Efficient Sum of Absolute Differences Hardware Architecture Using Adder Compressors for Integer Motion Estimation Design" IEEE Transactions on Circuits and Systems I: Regular Papers vol. PP no. 99 pp. 1-12 2017.

[12] M. Najem T. Bollengier J. C. Le Lann and L. Lagadec "A Cost Effective Approach for Efficient Time-Sharing of Reconfigurable Architectures" 2017 International Conference on FPGA Reconfiguration for General-Purpose Computing (FPGA4GPC) May 2017.

[13] M. Z. A. Khan and S. Qadeer "A new variant of Radix-4 FFT" 2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN) pp. 1-4 2016.

[14] C. A. Arun and P. Prakasam "A mathematical approach on various radix- 2i FFT algorithms" 2016 International Conference on Electrical Electronics and Optimization Techniques (ICEEOT) pp. 1040-1045 2016.

[15] A. Rauf M. A. Pasha and S. Masud "A Novel split radix fast fourier transform design for an adaptive and scalable implementation" 2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS) pp. 116-121 2016.