IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 2, Ver. II (Mar.-Apr. 2017), PP 28-33
www.iosrjournals.org

A Novel RTL Architecture for FPGA Implementation of 32- Point FFT for High-Speed Application

Kavitha MV[1],S.Ranjitha[2], Dr SG Hiremath[3] , Dr Suresh H N[4]
[1]Research scholar, Dept. Of E&IE, BIT,VTU ,Assistant professor, Gopalan college of Engineering and management , Bengaluru, Karnataka Bengaluru
[2]MTech,Bangalore , [3]Professor &Head ,EWIT, Bangalore
[4]Professor,[4]Dept. Of E&IE BIT,

Abstract: *In today's innovative technology, area, power and delay are the major parameters to design any kind of the algorithm on Field Programmable Gate Array (FPGA). In communication and Biomedical fields, Fast Fourier Transform (FFT) has the major role in obtaining the signal characteristics with minimum use of resources. Some of the algorithms have been proposed on FFT, such kind of algorithms were less effective in the performance parameters. To enhance this algorithm, in this paper, we have proposed a new algorithm with the area as well power efficient Low Area Carry Select Adder (CSLA). With this total area and power required for the algorithm has been minimized.*

Keywords: *FFT algorithm, FPGA, CSLA adder*

I. Introduction

Nowadays, Fast Fourier Transform (FFT) has got the much importance in the fields of signal processing and communication to analyze the features of any signal. The area, power, and speed are very important parameters to design any type of FFT processor. In VLSI technology also we have to reach optimum values of these parameters. In designing of FFT processor, the people should aware of the cost for that processor. FFT algorithm plays a vital role in the communications and signal processing operations involved in demultiplexing a Frequency Division Multiplexed (FDM) signal [1]. A fast radix-4 complex FFT implementation feasible for microprocessors with a 4-parallel SIMD instruction set has been proposed. The implementation loads consecutive 4 real or imaginary elements into a register at stages except the last. At the end stage, every 4 elements in a real or an imaginary array are loaded into registers by a matrix. This algorithm was implemented on the V830 processor, this processor will reduce the count of the clock but the number of processing elements will be increased [2]. The FFT implementation on a multi-core processor is proposed in [3], it uses the parallelism as well multi-core processor for better performance and good efficiency but it will increase the hardware. Scalable architecture for in-place fast Fourier transform (IFFT) computation for real-valued signals was presented. Their proposed computation was based on a modified radix-2 algorithm, which removed the redundant operations from the flow graph. The processing element (PE) was proposed using two radix-2 butterflies that can process four inputs in parallel. A conflict-free memory-addressing scheme was introduced to ensure the continuous operation of the FFT processor. Later on, the addressing way was extended to support multiple parallel PEs. The introduced real-FFT processor instantaneously requires fewer computation cycles and lower hardware cost compared to existing works. In this, the number of computation cycles was reduced proportionately with the increase in the number of PEs [4].

A novel ROM-less and low-power pipeline FFT/IFFT processor for OFDM applications, in this considering symmetry property of the twiddle factor and there, is no need of the ROM but for high point FFT will require much memory [5]. Another way of designing the FFT processor with low power complex multiplier for OFDM based wireless applications. In this canonical signed digital is used to implement the constant complex multiplications with the help of CSA tree [6]. A VLSI-oriented FFT algorithm radix-2/4/8, which can effectively minimize the number of complex multiplications. This algorithm can be implemented efficiently using a pipelined architecture. Based on the pipelined architecture, an 8K FFT ASIC is implemented for Digital Video Broadcasting (DVB) applications [7]. A new pipelined architecture for FFT processor implementation on hardware. In this paper, an optimized implementation of the 8-point FFT processor with the radix-2 algorithm in R2MDC architecture was proposed. The BU unit will minimize the multiplicative complexity of the FFT processor with the help of shift and add operations and it uses the less amount of the resource usage [8]. R2MDC architecture based FFT implementation will be minimizing the number of complex multiplications[9].

There are several methodologies for FPGA implementation of an FFT/IFFT module such as VHDL coding, System-level tools at macro block level and System-level tools at RT level. Here a comparison of all the three methods has been done and conclude that the first two methods were consuming much hardware and operating frequency. The last method gives the good results but internal data width cannot be controlled which is the major output error [9-10]. The CORDIC algorithm has only shift and adds operations, so it provides the better performance than compared to the MAC based FFT [10-11]. All above discussed methods has limitations with respect to parameters such area, power and execution time of the algorithms. To conquer these limitations we have proposed an efficient FFT processor design with the area as well power efficient CSLA adder.

FAST Fourier transform (FFT) is a basic block in orthogonal frequency division multiplexing (OFDM) systems. OFDM has been adopted in a very broad selection of applications from wired communication modems, like digital subscriber lines (xDSL) to wireless communication modems, like IEEE802.11 WiFi, IEEE802.16 WiMAX or 3GPP long run evolution (LTE), to method baseband knowledge. Inverse fast Fourier remodels (IFFT) converts the modulated data from the frequency domain to time domain for transmission of radio signals, whereas FFT gathers samples from the time domain, restoring them to the frequency domain. With multiple input multiple output (MIMO) devices, knowledge turnout will be increased dramatically. therefore MIMO-OFDM systems give promising rate and dependableness in wireless communications To handle "multiple" education streams, intuitively the purposeful blocks ought to be duplicated for the process the synchronal inputs. while not a correct style, the complexness of FFT/IFFT processors in MIMO systems grows linearly with the quantity of information streams.

II. Literature Survey

V. Arunachalam et.al [11] has introduced the efficient VLSI implementation of the FFT for Orthogonal Frequency Division Multiplexing (OFDM). As we know the complexity of the FFT is increases as it increases the performance of the system. Here the author has included multipliers and pass logics to enhance the performance improvement parameters but still multipliers are existed in the algorithm needs to eliminated.

Nandyala Ramanatha Reddy et.al [12] has introduced a split-radix algorithm for implementation of FFT. With the split-radix algorithm, the number of arithmetic operations were minimized. The importance of this article is to reduce the chip area, less execution time but it less effective in power consumption.

Nisha Laguri et.al [13] has introduced an efficient split-radix FFT based on Distributed Arithmetic (DA). The DA is the most important technique for implementation of minimized number of multiplier architectures for digital systems. The DA based algorithms were minimized overall arithmetic operations in FFT but increases the complexity and number of operations for each butterfly. With the split-radix FFT, less number of multiplications and additions were achieved but it introduces the overflow issue.

S. SreenathKashyap [14] has designed the several types of adder circuits such as ripple carry adder, Kogge stone adder, carry look ahead adder, Kogge stone adder, sparse Kogge stone adder and spanning carry look ahead adder. Also simulation was done on Xilinx by taking different bit widths and analyzed the power and delay parameters, but still these adders will be much power consuming.

Dongpei Liu et.al [15] has introduced a new CORDIC-based radix-4 FFT processor. CORDIC based FFT processor provide the Conflict-free parallel memory access scheme, pipelined CORDIC architecture and ROM free twiddle factors generator were employed for the FFT processor but not explained the variable length FFT/IFFT processing based on the radix-4 memory-based architecture.

III. Proposed Methodology

1].Radix-8 booth multiplier will be implemented as a complex multiplier for In place FFT architecture. And since this multiplier operates in parallel and requires less number of adders.2].Using Radix- 8 booth multiplier the computation time required will be less and also less power consuming technique.3].The proposed In place FFT architecture for real valued signals will be designed using verilog language in Xilinx ISE tool and the design will be synthesized and implemented to generate the area and power report. For generating power report X power analysis tool is executed after implementing the design on the FPGA device.

3.1 Proposed system

The proposed method has proposed the efficient FFT processor with radix-2 algorithm in Radix-2 Multipath Delay Commutator (R2MDC) architecture is the processing element. This butterfly- Processing Element (PE) used in the FFT processor to minimize the multiplicative complexity with real constant multiplication and also eliminates the multiplicative complexity by using add and shift operation in the proposed method.

The FFT algorithm is based on the fundamental principle of decomposing the computation of the Discrete Fourier Transform (DFT) of a sequence of length N into a smaller dfts.

$$z[k] = \sum_{n=0}^{N-1} x(n).W_N{}^{kn} --------(1)$$

$$\text{Where } W_N{}^{kn} = e^{\frac{-j2\pi kn}{N}}$$

Complex multiplications

The complex multiplication is a very expensive operation, to minimize the multiplicative complexity of the twiddle factor inside of the butterfly unit by calculating the only real multiplications, additions, and subtractions.

The twiddle factor multiplication

$$A + jB = (Y + jZ)(c + js) ---------- (2)$$

The complex multiplications can be simplified as

$$A = (C - S)Y + X ------(3)$$

$$B = (C + S)Z - X ------(4)$$

$$\text{Where } X = C(Y - Z)$$

Where C and S are pre-computed and stored in a memory. The three coefficients C, C + S, and C – S are necessary to store in a memory. The proposed algorithm of complex multiplication is shown in fig.3.

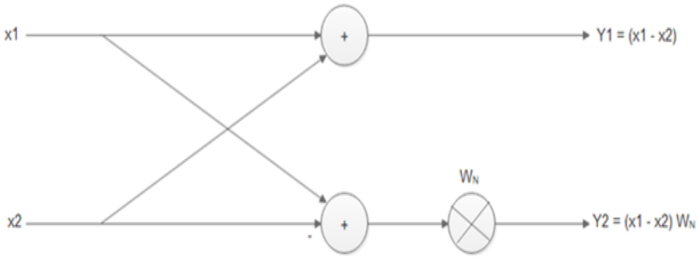

Fig.1. Signal flow graph of DIF- FFT

There are two types of algorithms in designing of FFT processor, which are Discrete in Time (DIT) FFT and Discrete in Frequency (DIF) FFT. In proposed method using the DITFFT algorithm, the major element of this is the BU which will take the input samples in bit reverse order and performs the internal operations of the algorithm. The proposed method will use the new AE-CSLA in the computations of these operations as it can be seen in fig 2 because it will reduce the area of the hardware and it also reduces the power required for the system. For example N-point FFT will take N^2 complex multiplications, N (N-1) additions, and $\log_2 N$ stages are required. The results obtained in the first stage are used as inputs to the next stage operation. After computation of the intermediate results of each stage of the FFT design with processing elements, these results are stored in

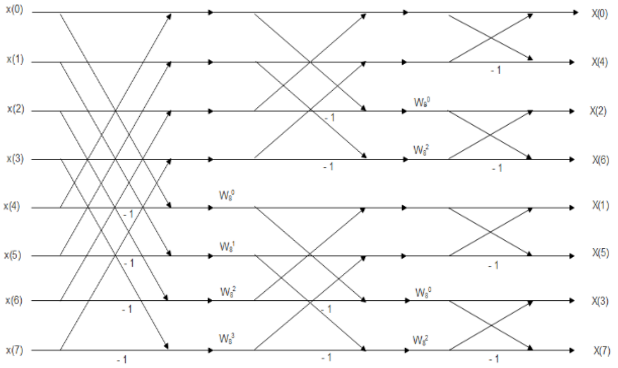

Fig.2. Butterfly diagram of DIT FFT algorithm

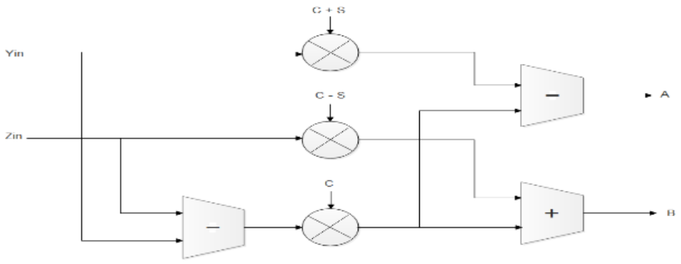

Fig.3. Implementation of complex multiplication

The fig.3 shows the implementation of the complex twiddle factors multiplication. As we can observe in the fig.3, the multiplication of the twiddle factors Y + jZ and C + jS are converted into a real multiplication represented in the eqn (3 & 4) and obtained the result of multiplication A and B in the fig.3. In this, we have one adder, two subtractors, and three multipliers. In our proposed method we replacing the normal adder to area efficient CSLA adder, to minimize the area and power consumption of the system.

The main proposed of this system is instead of using normal adder CSLA adder can be used, which is given in fig.8. This adder can achieve fast arithmetic operation in various data processing technique. In this adder mainly used for reducing area and power dissipation. CSLA is manipulating in many computational structures to cut the carry propagation delay. The elementary knowledge of this work is to habit BEC (binary to excess-1 convertor) instead of RCA (ripple carry adder) with Cin=1. By using fewer numbers of logic gates, we can derive BEC logic than n-bit FA (Full Adder).

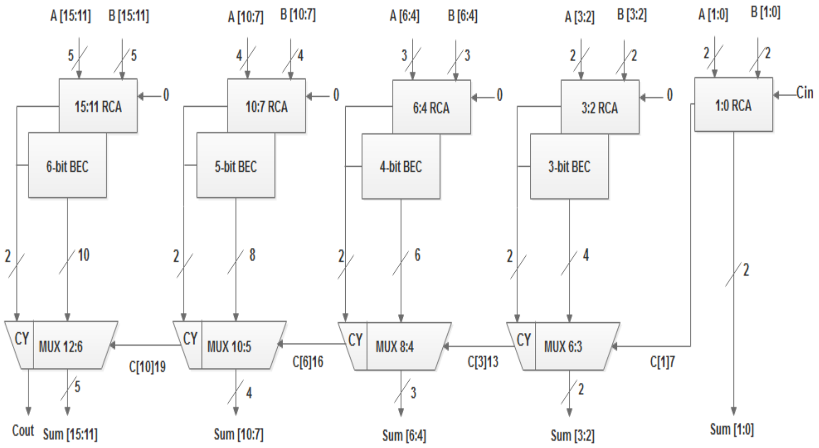

Fig.8. Proposed low area carry select adder

The assembly of the proposed system using BEC for normal adder or RCA with Cin =1 to hike the power and area. The 2-b RCA which has one full adder and one-half adder for Cin =1 where 3-b BEC is utilized which enhances one to the output from 2-b RCA. According to this consideration, the time delay has been reduced. It is the process on feedback values, which is the output of the mux is depended on the input of the mux. The input arrival time is lesser than the multiplexer selection input arrival time. By selecting the BEC output or the straight inputs, there are two possibilities is obtainable such as parallel and multiplexer rendering to the regulate signal Cin. While designing CSLA, the area will be reduced. The multiplexer delay and mux selection arrival time derived from the different kind of groups. Overall, the power consumption, delay, and area will be minimized in the proposed method with the support of CSLA adder.

3.2 FPGA Implementation of 32-point FFT

Fig 11 shows the block diagram for the implementation of 32-point FFT. In proposed method two RAMs are used to store the intermediate values. RAM2 is used to store the temporary value which is generated from intermediate stage. Input and output from RAM 1 and two is controlled by control logic block.

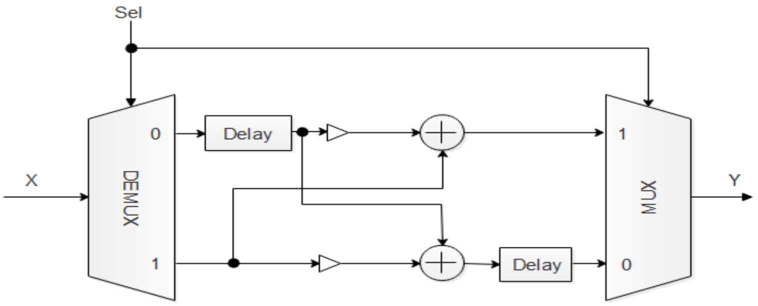

Fig.9. Structure of Processing Element (PE)

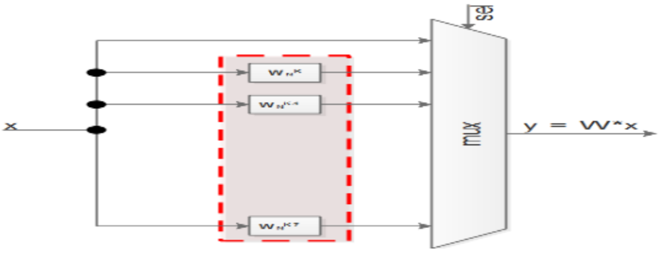

Fig.10. Weight Multiplication in PE

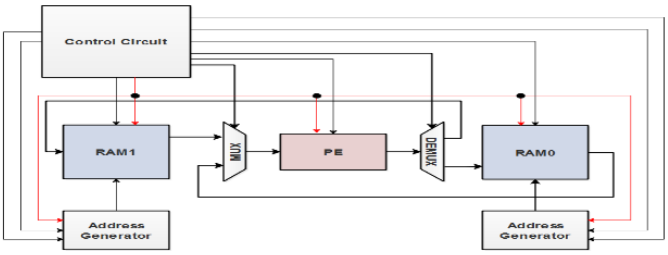

Fig.11. Block diagram for Proposed 32-point FFT

IV. Experimental Setup

The proposed method is implemented in Verilog version to obtain optimized area and power required for FFT processor implementation with low area CSLA adder. The complete work is done by using I_7 system with 8 GB RAM.

V. Results And Discussion

Table I. Comparison of Proposed and existing methods in Virtex-4 FPGA

Target FPGA	Circuit	LUT	Flip-flop	Slice	RAM	Frequency (MHz)
xc4vfx12-12sf363	Existing	1154/10,944	65/10,944	618/5,472	2	47.420
	Proposed	1,177/10,944	65/10,944	630/5,472	2	45.792

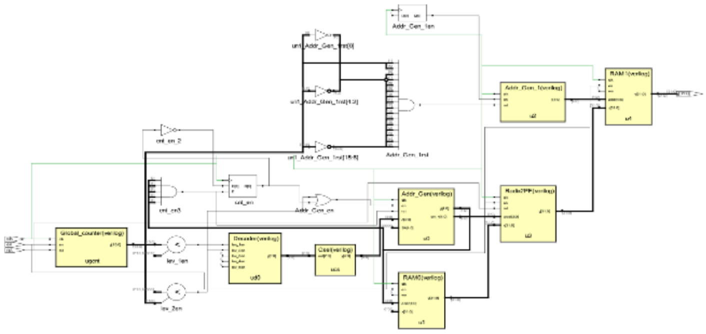

Fig.11. Block diagram for Proposed 32-point FFT

VI. Conclusion

In this paper, we have proposed a new kind of FFT processor with low area CSLA adder. This algorithm has improved the performance of parameters the FFT processor design. The algorithm has been analyzed by various numbers of parameters such as power, area, and delay. From these results, we can conclude that the proposed implementation of FFT processor is power and area efficient algorithm as we compared to the existing FFT processor implementations.

References

[1]. Wyche, M. K., "An efficient, microprocessor-based FFT implementation for FDM demultiplexing", Military Communications Conference, MILCOM'91, Conference Record, Military Communications in a Changing World, pp. 995-1000, IEEE, 1991.

[2]. Nadehara, Kouhei, Takashi Miyazaki, and Ichiro Kuroda., "Radix-4 FFT implementation using SIMD multimedia instructions", Acoustics, Speech, and Signal Processing, IEEE International Conference on. Vol. 4, pp. 2131-2134, IEEE, 1999.

[3]. Haofan Yang, Kedong Chen, Shengqiong Xie, Ming-e Jing, Zhiyi Yu, Xiaoyang Zeng., "Efficient implementation of 3780-point FFT on a 16-core processor", ASIC (ASICON), 10th International Conference on, pp. 1-4, IEEE, 2013.

[4]. Ayinala, Manohar, Yingjie Lao, and Keshab K. Parhi., “An in-place FFT architecture for real-valued signals”, IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 60, no.10, pp.652-656, 2013.

[5]. Indirapriyadarsini, K., S. Kamalakumari, and G. Prasannakumar., “VLSI Implementation of Pipelined Fast Fourier Transform”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol.1, no.4, pp-427, 2012.

[6]. Min Jiang, Bing Yang, Yiling Fu, Anping Jiang, Xin-an Wang, Xuewen Gan, Baoying Zhao, Tianyi Zhang., “Design of FFT processor with low power complex multiplier for OFDM-based high-speed wireless applications”, Communications and Information Technology, International Symposium on. Vol. 2, pp. 639-641. IEEE, 2004.

[7]. Lihong Jia, Yonghong Gao, Jouni Isoaho and Hannu Tenhunen., “A new VLSI-oriented FFT algorithm and implementation”, ASIC Conference, Proceedings. Eleventh Annual International, pp. 337-341, IEEE, 1998.

[8]. Mounir Arioua, Said Belkouch, Mohamed Agdad,Moha M’rabet Hassani., “VHDL implementation of an optimized 8-point FFT/IFFT processor in pipeline architecture for OFDM systems”, Multimedia Computing and Systems (ICMCS), International Conference on, pp. 1-5, IEEE, 2011.

[9]. Soundarya, G., V. Jagan Naveen, and D. Tirumala Rao.,”Implementation of 32-Point FFT Processor for OFDM System”, Computational Intelligence in Data Mining, Vol 3. Springer India, pp. 229-237, 2015.

[10]. J. Viejo, A. Millan, M. J. Bellido, E. Ostua, P. Ruiz-de-Clavijo, and A. Munoz., “Implementation of a FFT/IFFT Module on FPGA: Comparison of Methodologies”, Programmable Logic, 4th Southern Conference on, pp. 7-11, IEEE, 2008.

[11]. Karthick S1, Priya P2, Valarmathy S., "CORDIC Based FFT for Signal Processing System", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 1, Issue 6,pp.494-501, December 2012.

[12]. Arunachalam, V, and Alex Noel Joseph Raj., “Efficient VLSI implementation of FFT for orthogonal frequency division multiplexing applications”, IET Circuits, Devices & Systems, vol.8, no.6, pp.526-531, 2014.

[13]. Nandyala Ramanatha Reddy, Lyla B. Das, A.Rajesh, Sriharsha Enjapuri., “ASIC implementation of high speed fast fourier transform based on Split-radix algorithm”, Embedded Systems (ICES), International Conference on, pp. 22-27, IEEE, 2014.

[14]. Laguri, Nisha, and K. Anusudha., “VLSI implementation of efficient split radix FFT based on distributed arithmetic”, Green Computing Communication and Electrical Engineering (ICGCCEE), International Conference on, pp. 1-5, IEEE, 2014.

[15]. SreenathKashyap, S., “High Through-Put VLSI Architecture for FFT Computation”, Proceedings of the Third International Conference on Trends in Information, Telecommunication and Computing, pp. 3-13, Springer New York, 2013.

[16]. Dongpei Liu, Hengzhu Liu, Jianfeng Zhang, Botao Zhang and Li Zhou., “VLSI design of a hardware efficient FFT processor”, Future Information Technology, Application, and Service. Springer Netherlands, pp.61-70, 2012.