FFT utilizing Modified SQRT CSLA and Proposed 5:3 & 9:4 Compressor

Avneesh Kumar Mishra
Dept. of Electronics Communication
Truba College of Science and Technology, Bhopal

Neeraj Jain
Dept. of Electronics Communication
Truba College of Science and Technology, Bhopal

Paresh Rawat, PhD
Dept. of Electronics Communication
Truba College of Science and Technology, Bhopal

ABSTRACT
While designing Fast Fourier Transform (FFT) cores, due to the use of multiplexers, memory, or ROMs, there is a substantial increase in power consumption and area. In order to increase speed and throughput, folding and pipelining methods have been approached by various existing designs. But the prime disadvantage of those architectures is the use of multipliers for twiddle multiplications. This present work has proposed fast fourier transform using compressors based multiplier. Both parallel and pipelining techniques have also been used in the proposed designs.

Carry Select adder is known to be the fastest adder among the Conventional adder structures. This work uses an efficient Carry select adder by sharing the binary to excess-1 converter (BEC) term. After a logic simplification, we only need one XOR gate, one AND gate and one inverter gate for carry and summation operation. Through the multiplexer, we can select the correct output according to the logic states of the carry in signal. These all design and experiments were carried out on a Xilinx 14.1i Spartan 3e device family.

Keywords
Fast Fourier Transform (FFT), Regular 16-bit SQRT CSLA, Modified SQRT CSLA

1. INTRODUCTION
Because of the need of rapid information transmission, today versatile information transfers industry confronts the issue of giving the innovation that have the capacity to bolster a mixed bag of administrations running from voice correspondence with a bit rate of a couple kbps to remote media in which bit rate up to 2 Mbps [1]. The FFT is one of the most commonly used digital signal processing algorithm. Since FFT is done in the advanced area, there are a few techniques to execute the framework. One of the strategies to actualize the framework is utilizing ASIC (Application Specific Integrated Circuit) [2-4]. ASICs are the quickest, littlest, and least power approach to execute FFT into equipment. The primary issue utilizing this technique is rigidity of configuration procedure included and the more drawn out time to market period for the planned chip [5].

We have proposed outline for executing the quick Fourier change (FFT) utilizing modified SQRT CSLA and proposed 5:3 and 9:4 compressors operation in equipment keeping the objective of a force effective structural engineering. These plans are checked utilizing different equipment recreating devices.

2. 5:3 AND 9:4 COMPRESSORS
2.1 5:3 Compressor
5:3 compressors are capable of adding 4 bits and one carry, in turn producing a 3 bit output. The 5-3 compressor has 5 inputs V_1, V_2, V_3, V_4, P_{CRY} and 3 outputs C_{SUM}, N_{CRY} and N_{OUT} along with a previous Carry (P_{CRY}), nest carry (N_{CRY}) and compressors sum (C_{SUM}) as shown in Figure 1. The input P_{CRY} is the output from the previous lower significant compressor.

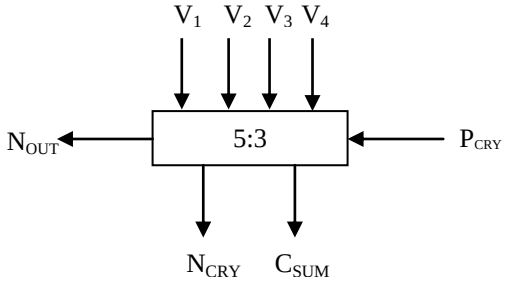

Figure 1: Block Diagram of 5:3 Compressors

The N_{CRY} is the output to the compressor in the next significant stage. The critical path is smaller in comparison with an equivalent circuit to add 5 bits using full adders and half adder.

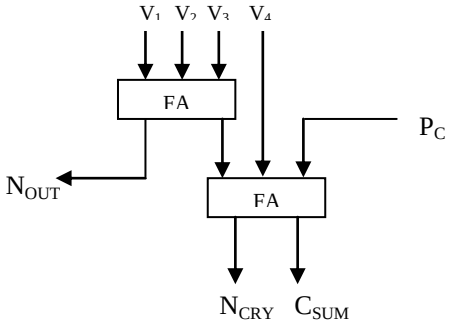

Figure 2(a): Design of 5:3 compressor using Full Adder

The 5:3 compressors is governed by the basic equation

$$V_1 + V_2 + V_3 + V_4 + P_{CRY} = C_{Sum} + 2*(P_{CRY} + N_{CRY}) \quad (1)$$

The standard implementation of the 5:3 compressors is done using 2 Full Adder cells as shown in Figure 2(a). The full adder consist of 2-XOR, 3-AND and 2-OR gate, it can be observed that the overall delay is equal to 4*XOR [6-8]. The block diagram in Figure 2(b) shows the 5:3 compressors using XOR and multiplexer with a 28 cost and 9 delay. The equations governing the outputs in the existing architecture are shown below

$$C_{SUM} = V_1 \oplus V_2 \oplus V_3 \oplus V_4 \oplus P_{CRY} \quad (2)$$

$$N_{OUT} = (V_1 \oplus V_2).V_3 + \overline{(V_1 \oplus V_2)}.V_1 \quad (3)$$

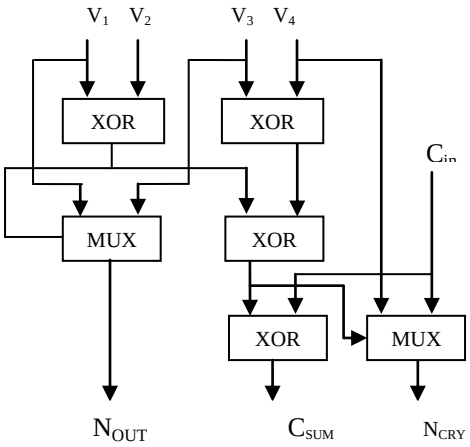

Figure 2(b): 5:3 Compressor using XOR and Multiplexer

$$N_{CRY} = (V_1 \oplus V_2 \oplus V_3 \oplus V_4).P_{CRY} + \overline{(V_1 \oplus V_2 \oplus V_3 \oplus V_4)}.V_4 \quad (4)$$

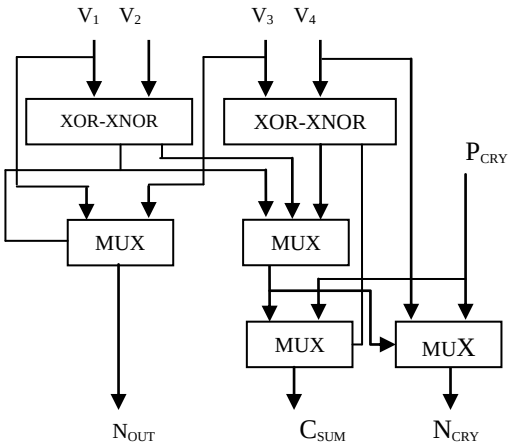

Figure 2(c): 5:3 Compressors using XOR-XNOR Gate

Thus replacing some XOR blocks with multiplexer's results in a significant improvement in delay. Also the MUX block at the SUM output gets the select bit before the inputs arrive and thus the transistors are already switched by the time they arrive. This minimizes the delay to a considerable extent. This is shown in Figure 2(c). The equations governing the outputs in the 5:3 compressors using XOR-XNOR gate and multiplexer are shown below

$$C_{SUM} = (V_1 \oplus V_2).(V_3 \oplus V_1) + (V_1 \oplus V_2).P_{CRY} \quad (5)$$

$$N_{OUT} = (V_1 \oplus V_2).V_3 + \overline{(V_1 \oplus V_2)}.V_1 \quad (6)$$

$$N_{CRY} = (V_1 \oplus V_2 \oplus V_3 \oplus V_4).P_{CRY} + \overline{(V_1 \oplus V_2 \oplus V_3 \oplus V_4)}.V_4 \quad (7)$$

2.2 9:4 Compressor

Similar to its 5:3 compressor counterpart, the 9:4 compressors as shown in Figure 3, is capable of adding 7 bits of input and 2 carry's from the previous stages, at a time [9]. In our implementation, we have designed a novel 9:4 compressor utilizing two 5:3 compressors, two full adder and one half adders. The architecture for the same has been shown in Figure 4.

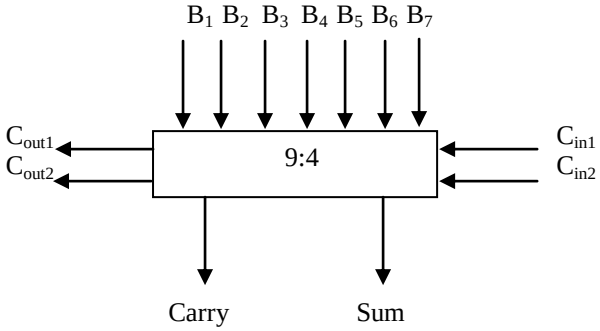

Figure 3: Block Diagram of 9:4 Compressors

$$Sum1 = S_1 \oplus S_2 \quad (8)$$

$$Carry1 = S_3 \oplus C_1 \oplus C_{21} \quad (9)$$

$$C_{out1} = C_3 \oplus C_2 \oplus C_{22} \quad (10)$$

$$C_{out2} = C_3C_2 + C_{22}C_2 + C_3C_{22} \quad (11)$$

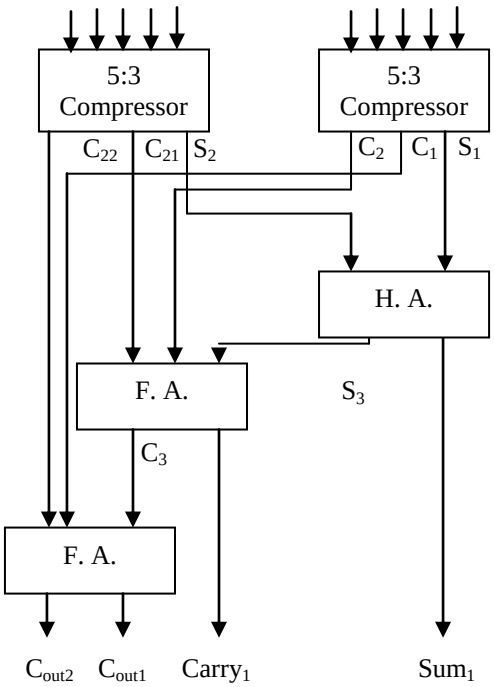

Figure 4: 9:4 Compressor using 5:3 Compressor

3. REGULAR 16-BIT SQRT CSLA USING RCA

The regular 16-bit SQRT CSLA is comprises of two swell convey adders and a multiplexer. Including two n-bit numbers with convey select viper is finished with two adders (in this way two swell convey adders) keeping in mind the end goal to perform the computation twice, one time with the presumption

of convey being zero and the other accepting one. After the two outcomes are ascertained, the right aggregate, and the right convey, is then chosen with the multiplexer once the right convey is known [10].

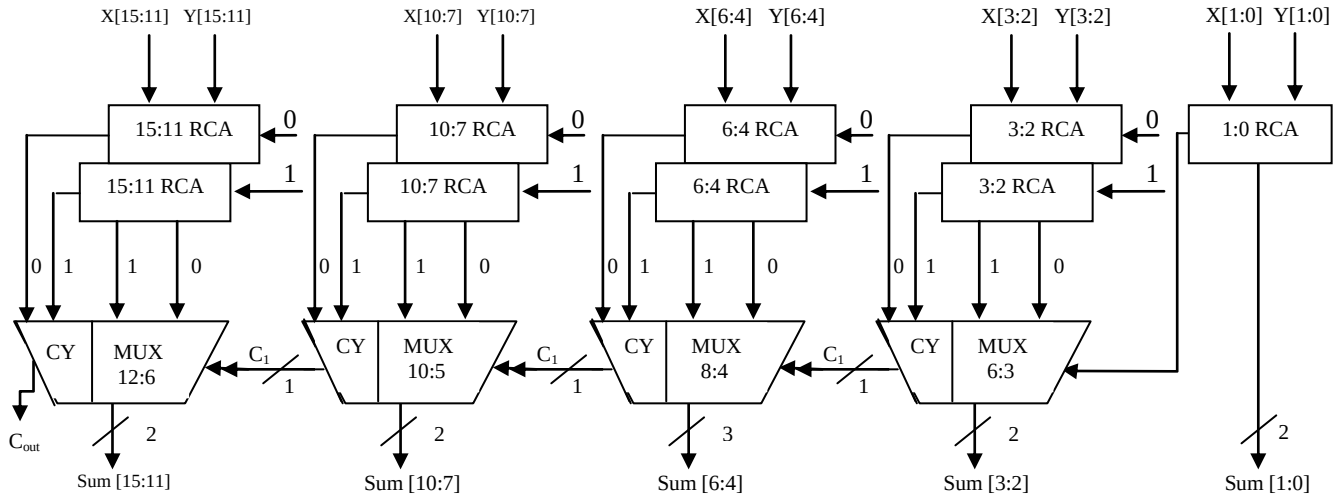

Figure 5: Block Diagram of 16-bit Regular 16-bit SQRT CSLA using RCA

The do of the Group 0 which goes about as the choice data to mux which is in gathering 1, chooses the outcome from the relating RCA (Cin=0) or RCA (Cin=1). Additionally the remaining gatherings will be chosen relying upon the Cout from the past gatherings.

3.1 Modified Sqrt CSLA

The Binary to excess one Converter (BEC) replaces the ripple carry adder with Cin=1, in order to reduce the area and power.

A 16-bit convey select snake can be created in two unique sizes to be specific uniform piece size and variable square size. Swell convey adders are the easiest and smaller full adders, however their execution is restricted by convey that must proliferate from the minimum noteworthy bit to the most-critical bit. The pace of a convey select viper can be enhanced up to 40% to 90%, by performing the increases in parallel, and diminishing the most extreme convey delay.

Figure 5 demonstrates the Regular structure of 16-bit SQRT CSLA. It incorporates numerous swell convey adders of variable sizes which are separated into gatherings. Bunch 0 contains 2-bit RCA which contains stand out swell convey snake which includes the info bits and the information convey and results to entirety [1:0] and the do.

Consumption of the regular CSLA, The modified16-bit CSLA using BEC is shown in Figure 6. The structure is again divided into five groups with different bit size RCA and BEC. The group 2 of the modified 16-bit CSLA is shown Figure 6.

3.2 Binary To Excess-1 Converter (BEC)

BEC is a technique to reduce the number of partial products in 1-bit increase. The binary to excess-1 converter is used to reduce the area and power consumption in CSA. Figure 7 shows the basic structure of 3-b BEC. The Boolean expressions of the 3-b BEC is as

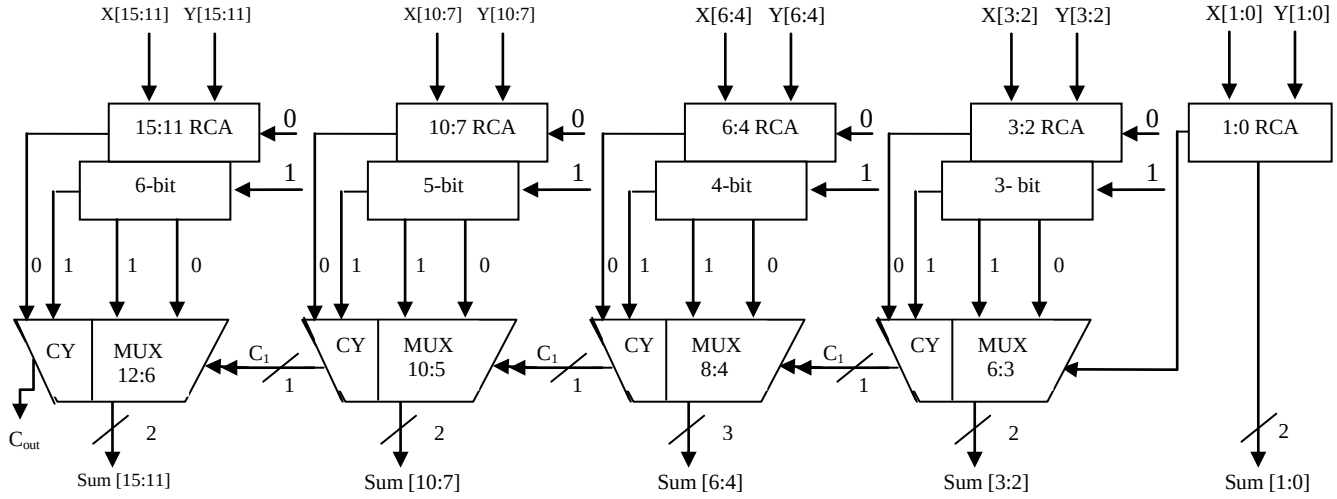

Figure 6: Block Diagram of Regular 16-bit SQRT CSLA using RCA with BEC

(12) X0 = ~ B0

(13) X1 = B0^B1

(14) X2 = B2^ (B0 & B1 &B2)

Table 1: Function Table of 4-bit BEC

Binary[3:0]	Excess- 1[3:0]
0000	0001
0001	0010
0010	0011
|	|
|	|
|	|
1110	1111

By physically tallying the quantity of entryways utilized for gathering 2 is 43 (full viper, half snake, multiplexer, BEC). One info to the mux runs from the RCA with Cin=0 and other information from the BEC. Looking at the gathering 2 of both customary and changed CSLA, it is clear that BEC structure diminishes the territory and force. In any case, the weakness of BEC technique is that the deferral is expanding than the consistent CSLA. The primary thought of double to exces-1 converter rather than the RCA with RCA (swell convey snake) and Cin =1 keeping in mind the end goal to lessen the zone and force utilization of the 16-B SQRT CSLA.

3.3 DELAY AND AREA METHODOLOGY

The delay and area evaluation methodology considers all gates to be made up of AND, OR, and Inverter (AOI), each having delay equal to 1 unit and area equal to 1 unit.

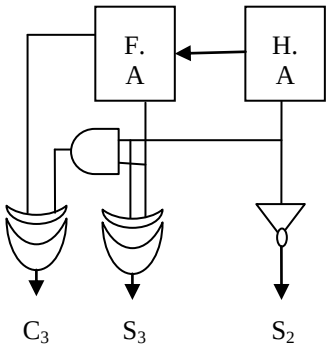

Figure 7-bit Booth Encoder

We then add up the number of gates in the longest path of a logic block that contributes to the maximum delay. The area evaluation is done by counting the total number of AOI gates required for each logic block [11-12].

Table 2: Delay and Area count of the basic block of CSLA

Adder Block	Delay	Area
XOR	3	5
2:1 MUX	3	4
Half Adder	3	6
Full Adder	6	13

Table 3: Delay and Area Count of Regular SQRT CSLA and Modified SQRT CSLA Groups

Regular SQRT CSLA			Modified SQRT CSLA		
Group	Delay	Area	Group	Delay	Area
Group-1	9	19	Group-1	9	19
Group-2	11	57	Group-2	13	43
Group-3	13	87	Group-3	16	61
Group-4	16	117	Group-4	19	84
Group-5	19	147	Group-5	22	107

4. FAST FOURIER TRANSFORM

The decimation, however, causes shuffling in data. The entire process involves $v = \log 2 N$ stages of decimation. Fast Fourier Transform (FFT) is one of the most efficient ways to implement Discrete Fourier Transform (DFT) due to its reduced usage of arithmetic units. DFT is one of those primary tools that are used for the frequency analysis of discrete time signals and to represent a discrete time sequence in frequency domain using its spectrum samples. The analysis (forward) and synthesis (inverse) equations of an N point FFT are given below.

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \tag{15}$$

$$x[n] = \frac{1}{N} \sum_{n=0}^{N-1} X[k] W_N^{-kn} \tag{16}$$

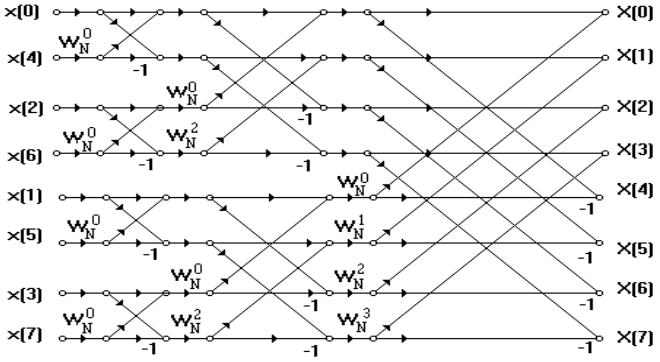

Figure 8: IFFT Signal Flow Graph.

Graphically the operation can be view utilizing FFT stream chart demonstrated as a part of figure 8. From this figure, the FFT calculation is refined in three stages. The X (0) until X (7) variable is indicated as the data worth partitioned as even and odd terms for FFT calculation and X (0) until X (7) is arrives in a grouping structure as the yield. There are two operations to finish the calculation in every stage. The upward bolt will execute expansion operation while descending bolt will execute subtraction operation. The subtracted worth is increased with twiddle element esteem before being prepared into the following stage. This operation is done simultaneously and is known as butterfly process. Noticed that each of the butterfly process is performed simultaneously empower it to execute FFT calculation process in a quick system.

5. SIMULATION RESULT

All the planning and examination in regards to calculation that we have said in this paper is being created on Xilinx 14.1i

overhauled form. Xilinx 14.1i has couple of the striking components, for example, low memory prerequisite, quick troubleshooting, and ease. The most recent arrival of ISETM (Integrated Software Environment) outline device gives the low memory prerequisite surmised 27 rate low. By the guide of that product we investigate the project effectively. Additionally included is the most current arrival of the chip scope Pro Serial IO Tool unit, giving rearranged investigating of fast serial IO plans for Spratan-3 FPGAs.We functionally verified each unit presented in this paper including modified SQRT CSLA and proposed compressor based multiplier. We have been found from the results shown in Table 2 to Table 4 respectively.

Table 4: Device utilization summary (VERTEX-7) of Fast Fourier Transform (FFT)

Design	No. of slices	No. of 4 input LUTs	MCPD (ns)
FFT using Modified SQRT CSLA and Compressor based Multiplier	327	576	22.269
FFT using Modified SQRT CSLA and Compressor based Multiplier	171	302	16.369
FFT using Modified SQRT CSLA and Proposed Compressor based Multiplier	149	261	13.965

6. CONCLUSION

Fast Fourier change (FFT) is utilized to change over intricate and genuine qualities into genuine and complex ones. It obliges rot of data into stages using butterfly like DFT. Yet the butterfly used as a piece of FFT is really unmistakable with respect to coefficients or multipliers.In this thesis three design are discussed i.e. Design FFT using modified SQRT CSLA and compressor based multiplier, Design FFT using modified SQRT CSLA and modified compressor based multiplier and Design FFT using modified SQRT CSLA and proposed compressor based multiplier. Among all three designs, proposed compressor based multiplier provides the least amount of Maximum combinational path delay (MCPD).

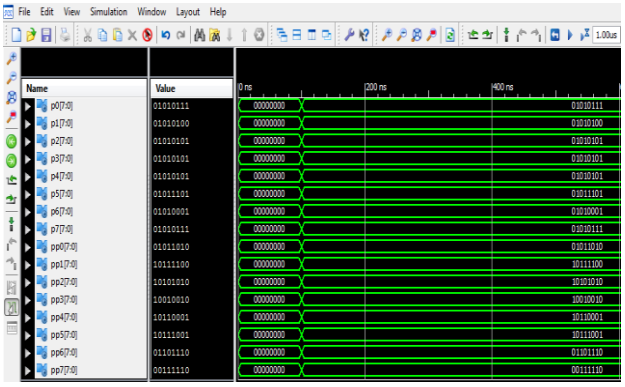

Figure 9: Output Waveform of Proposed Design

7. REFERENCES

[1] Sushma R. Huddar and Sudhir Rao, Kalpana M., "Novel High Speed Vedic Mathematics Multiplier using Compressors", 978-1-4673-5090-7/13/$31.00 ©2013 IEEE.

[2] S. S. Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A, "Implementation of Vedic multiplier for Digital Signal Processing", International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011, Proceedings published by International Joural of Computer Applications® (IJCA), pp.1-6.

[3] Himanshu Thapaliyal and M.B Srinivas, "VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics", Center for VLSI and Embedded System Technologies, International Institute of Information Technology Hyderabad, India.

[4] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics: Sixteen simple Mathematical Formulae from the Veda", Delhi(2011).

[5] Sumit Vaidya and Depak Dandekar. "Delay-power performance comparison of multipliers in VLSI circuit design". International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010.

[6] P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Eng. Edu, Vol.8, No.2, 204, UICEE Published in Australia.

[7] S. Correa, L. C. Freitas, A. Klautau and J. C. W. A. Costa, "VHDL Implementation of a Flexible and Synthesizable FFT Processor", IEEE LATIN AMERICA TRANSACTIONS, VOL. 10, NO. 1, JAN. 2012.

[8] Y. Kim and L. -S. Kim, "16-bit carry-select adder with reduced
area, "Electron. Lett. vol. 37, no. 10, pp. 614-615, May 2001.

[9] B. Ramkumar, H. M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," *Eur. J. Sci. Res.*, vol. 42, no. 1, pp. 53–58, 2010.

[10] T. Y. Ceiang and M. J. Hsiao, "Carry-select adder using single ripple carry adder," *Electron. Lett.*, vol. 34, no. 22, pp. 2101–2103, Oct. 1998.

[11] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," *Electron. Lett.*, vol. 37, no. 10, pp. 614–615, May 2001.

[12] J. M. Rabaey, *Digital Integrated Circuits—A Design Perspective*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[13] Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for low power applications, "in *Proc.IEEE Int. Symp. Circuits Syst.*, 2005, vol. 4, pp. 4082–4085