International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

Efficient Design of Multiplier Using Adder Compressors

S.Srinivas & Emandi Jagadeeswararao
1PG Scholar, Department of ECE, Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh, India.
2Assistant Professor, Department of ECE, Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh, India.

***Abstract**:*

Multiplication is one of the most common arithmetic operations employed in digital systems such as FIR filters and DSP processors but multipliers are the most time, area, and power consuming circuits. Improvement in any of these parameters can be advantageous for improving the efficiency of the circuit. High-speed multiplier which uses the high-speed adder is designed based on the Wallace tree concept in this paper. In This paper first we presents an approach towards the reduction of delay in the Wallace tree multipliers by using 4:2 compressors along with full-adders and half-adders, in the partial product reduction stage. After, this paper proposes to design efficient 4-bit and 8-bit Wallace Tree Multiplier using 8-2 and 4-2 adder compressors. Verilog HDL is used for the programming and XILINX 14.7 for the synthesis and simulation. Finally, the proposed design is achieved to get very efficient results.

Keywords

Wallace Tree multiplier, compressors, adders, Xilinx and Verilog.

1. Introduction

Multiplications are important and tedious task among arithmetic operations. So multipliers are the major components in the various processors like arithmetic, signal, and image processors. There are many multiplication based functions like multiply and accumulate, convolution, filtering etc. in signal processing and image processing. The execution time for this process highly depends on the speed of operation of multiplier unit. In many DSP algorithms multiplication consumes more time compared to other basic operations, so the critical delay path for the complete operation is determined by the delay required for the multiplication unit and it substantiates the performance of the algorithm. Addition and multiplication are widely used operations in computer arithmetic; for addition full-adder cells have been extensively analysed for approximate computing [6-8].

Now a days available different types multiplication algorithms and each algorithm involves three basic steps such as partial product generation, partial product reduction and final summation. Some of the multiplication algorithms are serial multiplication, parallel multiplication and serial-parallel multiplication. Serial multiplication contains less hardware and less speed of operation [1].Parallel multiplication is used in high speed application and speed depends on number of partial products [2]. Now day's available different types parallel multiplier like array multiplier and tree multiplier [3]. Wallace tree multiplier is little bit fast among the available multipliers [4] and they use carry save algorithm for faster applications [5].

This paper is organized as follows. Section 2 is a review of existing schemes for Wallace tree multiplier. The two new designs of an approximate 4-2 and 8-2 adder compressor are presented in Section 3. Introduction of 4-bit and 8-bit in Section 4 and High sped adders in Section 5. Two proposed High speed multipliers i.e. 4-bit and 8-bit Wallace tree multipliers see in Section 6. Simulation results for multipliers with the approximate compressors are provided in Section 7. Section 8 concludes the manuscript.

2. Literature Survey

The approximation of athematic design focus on adder, but multiplier is one of the primary source reduce the power consumption digital signal processing such as FIR filters [9] . The majority of approximate arithmetic design focused not only adder but also the multiplier. So, In this paper we can design the multiplier using approximate adders. There are some research directions for designing approximate circuits. V.Gupta et al[10] made a reduction approximate multiplier at transistor level. For simplifying logic area D. Shinet al.[12] reduce the circuit area and more studies reduced the circuit delay by adjusting circuit architecture [11].Two well-known fast multipliers were presented by Dadda and Wallace and these multipliers use full adders and half adders in reduction phase. The modified Wallace tree reduces 80% of half adders. The partial products are also minimized. Finally, path carry select adder was used in final carry propagation [13].Fast column compression multiplication has been acquired using combination of two different designs. One is dividing the partial products into two portions for independent parallel column compression and acceleration is achieved using hybrid adder. The performance of the column compressed multiplier was examined by analyzing area, delay and power. The results demonstrated that 64-bit regular Dadda multiplier is 41.1% slower than fast column compression multiplier and also the power-delay product is considerably lower than fixed Dadda multiplier [14].Power management has developed as a critical anxiety due to its portable applications. Many procedures at different levels of design procedures were used to reduce power dissipation. High speed multiplication is a major problem in high performance computing systems. 8 8× hybrid tree multiplier is implemented by linking Wallace and Dadda methods and the results demonstrated 40 % of power reduction [15]. The modifications of Wallace/Dadda Multiplier use carry look ahead adders as a replacement of full adders to implement the reduction in bit product matrix. Each carry look ahead adder reduces the stages up to 9 partial products and it leads to a few reduction stages compared to conventional Wallace/Dadda Multiplier [16]. Swing Restored complementary Pass-transistor Logic (SR-CPL) was created using n-MOS transistor that is derived from Complementary Pass Logic (CPL) logic which can be applied to the arithmetic building block and it delivers high speed. DADDA multiplier was implemented using ripple carry and carry save adder and the simulations we are carried out by TANNER EDA tool [17].We proposed Wallace tree multiplier using different 4-2 and 8-2 adder compressors and it improves speed compare existing conventional Wallace tree multiplier.

Page should be of A4 size with normal margin. All printed material, including text, illustrations, and charts, must be kept within a print area. Do not write or print anything outside the print area. All text must be in a two-column format. Text must be fully justified. A format sheet with the margins and placement guides is available in Word files as <format.doc>. It contains lines and boxes showing the margins and print areas. If you hold it and your printed page up to the light, you can easily check your margins to see if your print area fits within the space allowed.

3. Adder Compressors

Compressors by far have been considered as the most efficient building blocks of a high speed multiplier. It provides an advantage of accumulation of partial products at an expense of least possible power dissipation. Rather than entirely summoning partial products with the help of CSA/Ripple adder tree, a structure of compressors would complete the same task in much lesser time and also will simultaneously eradicate the problems of large power consumption and optimization of the area. This addition of partial products when done using conventional method of implementing Full Adders and Half Adders cannot account as much to lessening of delay associated with the critical path as when counter or compressors are used. The reason for the apparent preference of compressors over counters is the advantages it provides in terms of power, number of transistors used and the delay associated with the critical path(comprising of XOR's mainly) [18]. The compressor design implemented in this paper prefers both MUX's and XOR's.

The internal structure of the 3-2 adder compressor is presented in Fig. 1-a. The maximum delay is given by two XOR gates. The final sum S of the 3-2 adder compressor is given in (1). The 3-2 adder compressor can also be used as a full-adder (i.e. mux-based full-adder) when the input C is used as a carry input.

$$S=Sum+2Carry \quad (1)$$

The internal structure of the 4-2 adder compressor is presented in Fig.1-b. It has a reduced critical path compared to conventional adders since the maximum delay is given by three XOR gates. The 4-2 compressor has five inputs (A, B, C, D, Cin), where Cin is the input carry, and three outputs (Sum, Carry and Cout). In this adder compressor, the carry output Cout is independent of the input carry (Cin), making it possible to implement this structure with higher performance. The final sum S result of the 4-2 adder compressor is given in (2).

$$S=Sum+2(Cout+Carry) \quad (2)$$

The internal structure of the 5-2 adder compressor is presented in Fig. 1-c. The maximum delay is given by six XOR gates. The final sum S of the 5-2 adder compressor is given in (3).

$$S=Sum+2(Cout1+Cout2+Carry) \quad (3)$$

The internal structure of the 7-2 adder compressor is presented in Fig. 1-d [19]. The maximum delay is given by ten XOR gates. The final sum S of the 7-2 adder compressor is given in (4).

$$S=Sum+2(Cout1+Cout2+Carry) \quad (4)$$

In this paper 8-2 adder design using 3-2,4-2,5-2 and 7-2. The internal structure of the 8-2 adder compressor is presented in Fig. 2(a,b,c,d) [20]. The final sum S of the 8-2 adder compressor is given in (5).

$$S=Sum+2(Cout0+Cout1+Cout2+Cout3+Cout4+Carry) \quad (5)$$

The main title (on the first page) should begin 1-3/8 inches (3.49 cm) from the top edge of the page, centered, and in Times 14-point, boldface type. Capitalize the first letter of nouns, pronouns, verbs, adjectives, and adverbs; do not capitalize articles, coordinate conjunctions, or prepositions (unless the title begins with such a word). Leave two blank lines after the title.

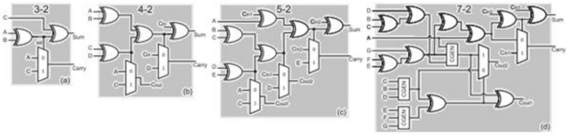


Fig.1. Adder compressors internal structures: (a) 3-2; (b) 4-2; (c) 5-2; (d) 7-2.

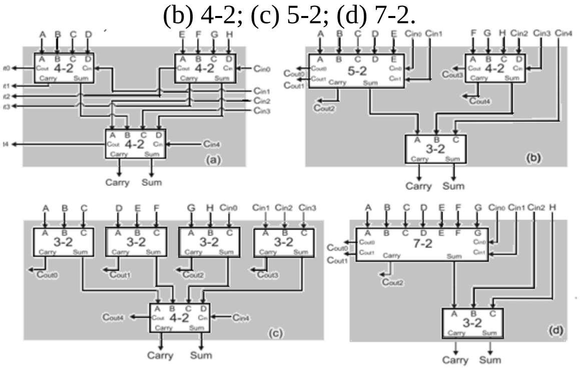


Fig.2. The structure of 8-2 adder compressor using: (a) Only 4-2 adder compressors; (b) Combination of 5-2, 4-2 and 3-2 adder compressors; (c) Combination of 3-2 and 4-2 adder compressors; (d) Combination of 7-2 and 3-2 adder compressors [20].

4. Wallace Tree Multiplier

A multiplier designed by using Wallace tree architecture is known as a Wallace multiplier. Wallace multiplier consumes less power and its switching speed is faster as compared to other multiplier architectures. Researchers have shown interest on Wallace multiplier, as result of which, different architectures are introduced to design a better Wallace multiplier architecture. A conventional Wallace multiplier and a reduced complexity Wallace multiplier are two architectures among them. In this paper design and performance analysis of a conventional Wallace multiplier and a reduced delay Wallace multiplier are discussed. Performance analysis is carried out by using Xilinx 14.7 synthesis tool.

The Wallace multiplier [21] exercises the Wallace tree which is an efficient and parallel multiplication algorithm by which to generate a result. The primary advantage of the Wallace tree is making an adding stage reduction by using half-adder and full-adder. The Wallace tree reduce the delay order of array multiplier from O(n) to O(log n).Fig.3(a) shows a 4×4 Wallace tree dot rotation. Also further reduce the delay Wallace tree multiplier using 4-2 adder compressor [22] shown in the Fig.3 (b). In this paper [22], design 4-2 adder compressor using mux in the place of XOR gates for increasing the speed.

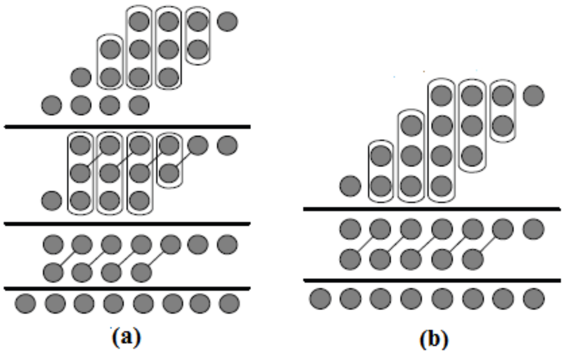


Fig. 3 A 4×4 Wallace multiplier dot rotation with (a) only half-adders and full-adder (b) only half-adder, full-adder and 4-2 adder compressor [22]

In this paper also develop the Verilog code for conventional 8×8 Wallace tree multiplier and further reduce the delay by designing Wallace tree with 4-2 adder compressor [23] and relevant diagram shown in Fig. 4.

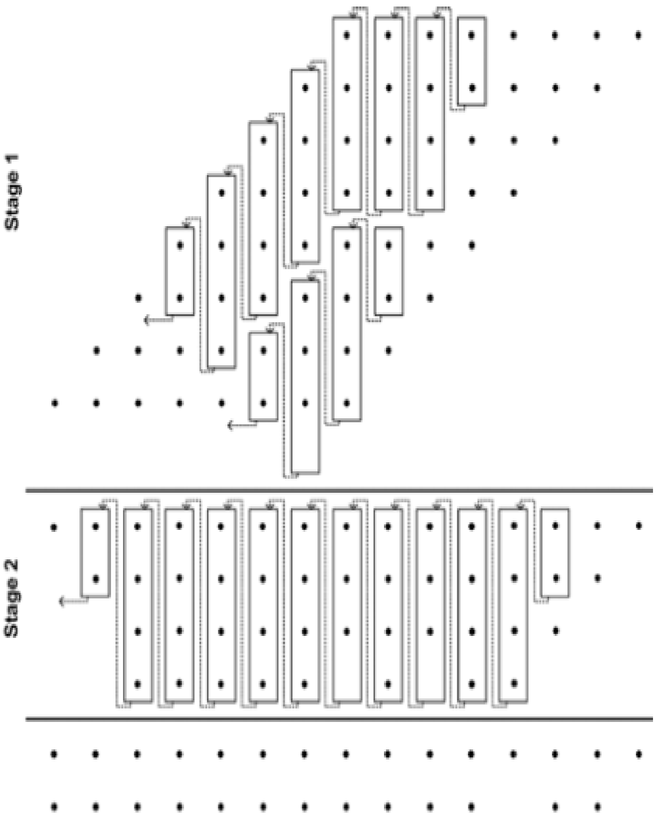


Fig. 4 Reduction circuitry of an *8×8* Dadda multiplier using 4-2 adder compressor [23]

5. High Performance Multipliers

For any multiplication algorithm contains three steps but in this summation of partial products is an important step to generate the final result. The performance of the multiplier depends on how fast partial products get added to obtain the final result. Many researchers can work in this area to achieve fast adders. The fundamental adder architecture is a Ripple Carry Adder and further develops number of adders such as Carry look a-head adder, Carry select adder, Carry save adder and Carry skip adder etc. In this ripple carry adder is well known for its regular structure and maximum delay because each step waits for the carry from the previous step. Carry look ahead adder have a minimum delay but area associated with these adders are maximum. Carry skip adder gives the more performance than ripple carry adder but it's consist of extra hardware circuitry to skip the carry generated [24]. Carry save adder gives the further addition by reducing addition there are number of three into two. The major drawback carry save adder consumes larger area [25]. Further carry select adder uses the two ripple carry adders and it doesn't wait for previous stage to execute. The carry select adder with higher bits exhibits excellent area and speed trade off compare with other adder architectures [26]. Many modifications can be dine in carry save adder for sacrificing its speed for area [27].

The 8-Bit Wallace tree multiplier is shown in Fig. 5. In this paper mainly developed the efficient multiplier using four 8-2 adder compressors already discussed in section 3. The red mainly indicates the 8-2 compressors and yellow box indicates the final sum shown in Fig. 5 and also combine 8-2 adder compressor and final sum and design single module [29]. In final sum is a combination of normal half adders and full adders. After designing the efficient multiplier

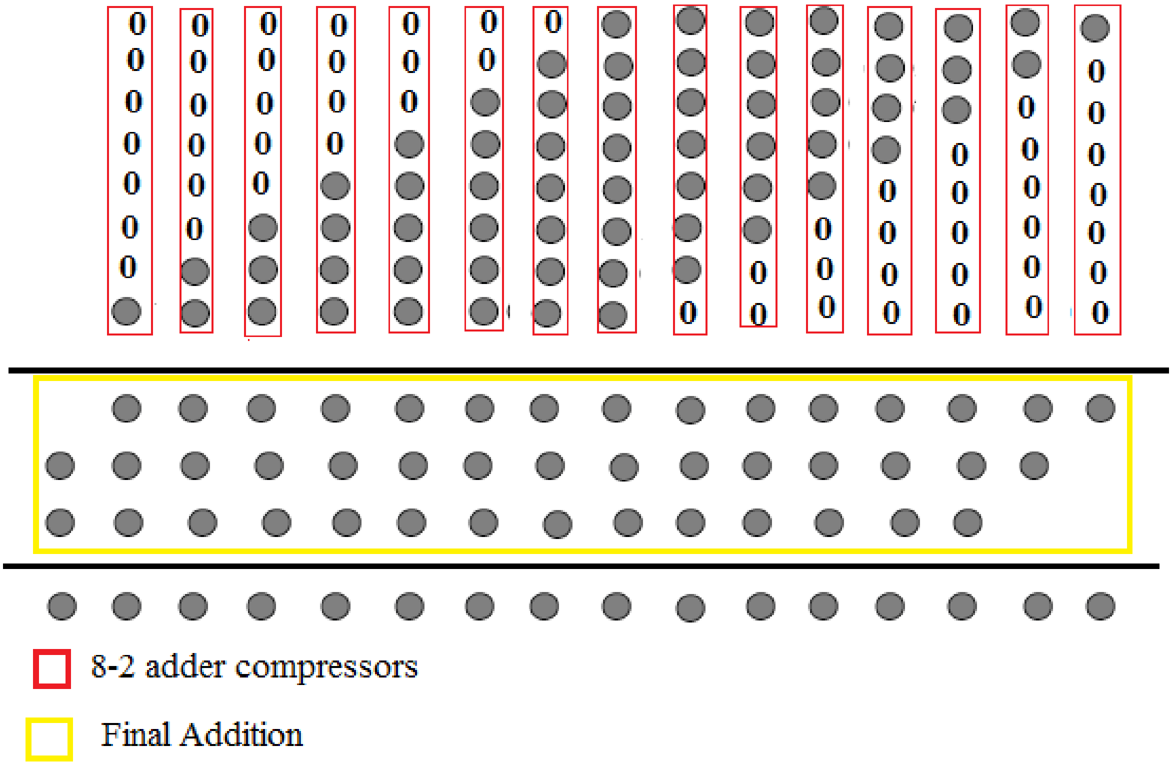


Fig. 5 8-Bit Wallace Tree Multiplier using different 8-2 adder compressors

Efficient 8-Bit Multiplier develop using Verilog code and simulate using Xilinx 14.7

Also design 4-bit multiplier using 4-2 adder compressor is shown in Fig. 6. In this Figure Blue box indicates the 4-2 adder compressor stage and yellow box indicates the final sum stage. In final sum stage is consists of normal half adders and normal full adders. Also combine both 4-2 adder compressors stage and final sum stage to design single module [28]. In this paper mainly focus on performance of multipliers also achieves to improve the performance of the multiplier using adder compressors. In following section discussed about results and design summary and simulation results of 4-Bit and 8- Bit Wallace Tree Multiplier.

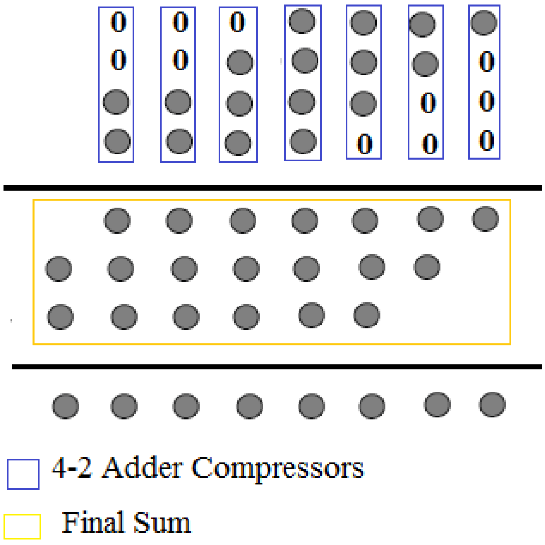


Fig. 6 4-Bit Wallace Tree Multiplier using 4-2 adder compressor

6. Simulation Results

The design was synthesized on Xilinx ISE and the functional verification of Wallace tree multiplier was done on Xilinx ISIM. The targeted device is of Spartan-3 of Spartan family. The grade speed of the design is set to -5. The following section contains the results obtained by synthesizing the design in XILINX ISE. Table I represents the device utilization summary of proposed 8-Bit Wallace Tree multiplier with four 8-2 adder compressors.

Table 1: Device Utilization Summary of 8-Bit Wallace Tree Multiplier

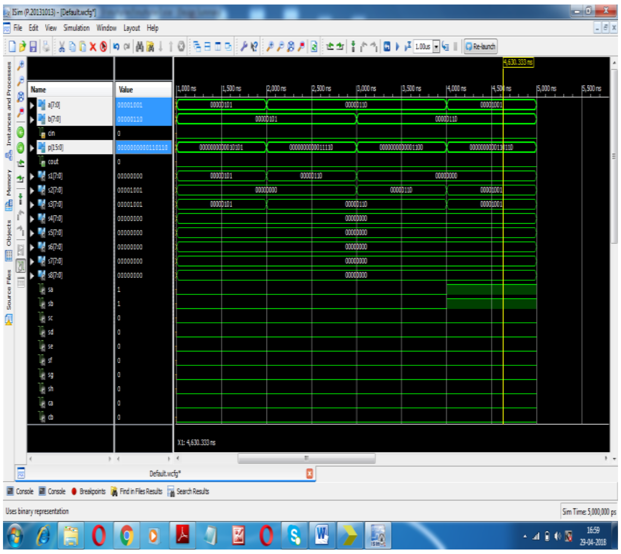
Logic Utilization	Method 1	Metho d 2	Method 3	Method 4
No. of slices	59	52	45	41
No. of 4 i/p LUT	113	99	88	79
No. of slices flip flops	64	64	64	64
No. of Bonded IOS	45	45	45	41
Maximum combinatio nal path delay	19.66 ns	10.69 ns	11.29 ns	8.19 ns

In Table 1 mainly tells about how much hardware used in Spartan 3. The Method 1 generally design using 8-2 adder compressor is shown in Fig 2(a), Method 2 using 8-2 adder compressor is shown in Fig 2(b), Method 3 using 8-2 adder compressor is shown in Fig 2(c) and Method 4 using 8-2 adder compressor is shown in Fig 2(d). All Methods are 8-Bit Wallace Tree Multipliers.

Table 2: Device Utilization Summary of 4-Bit Wallace Tree Multiplier

Logic Utilization	Method 1	Method 2	Method 3
No. of slices	14	10	15
No. of 4 i/p LUT	24	21	26
No. of slices flip flops	0	16	0
No. of Bonded IOS	16	20	17
Maximum combinational path delay	12.05 ns	8.07 ns	12.507 ns

Table 2 indicates the device utilization of three 4-bit Wallace Tree Multipliers. In Method 1 using 4-2 adder compressors and normal adders, Method 2 using half adders and full adders and method 3 using only 4-2 adder compressors in first stage and final sum using normal adders.



ISim (P.20131013) - [Default.wcfg*]

Name	Value
a[7:0]	00001001
b[7:0]	00000110
cin	0
p[15:0]	0000000000110110
cout	0
s1[7:0]	00000100
s2[7:0]	00001001
s3[7:0]	00001001
s4[7:0]	00000000
s5[7:0]	00000000
s6[7:0]	00000000
s7[7:0]	00000000
s8[7:0]	00000000
sa	1
sb	1
sc	0
sd	0
se	0
sf	0
sg	0
sh	0
ca	0
cb	0

X1: 4,630.333 ns

Uses binary representation

Fig. 7 8-Bit Wallace Tree Multiplier Simulation output

7. Conclusion

From the obtained results for the proposed design, It can be seen that the use of 4-2 and 8-2 adder compressors can enhance the performance of the system significantly. The Proposed multiplier results and simulation output are obtained. The proposed design enhances the performance and also increases the speed and uses very less area and gives the accurate results. Generally these multipliers used in DMA processors and Fir Filters and play major role in processors.

8. References

[1] S. Akhter and S. Chaturvedi, "Hdl based implementation of nx n bit serial multiplier, " in Signal Processing and Integrated Networks (SPIN), 2014 International Conference on. IEEE, 2014, pp. 470-474.

[2] S. Shah, A. AI-Khalili, and D. AI-Khalili, "Comparison of 32-bit multipliers for various performance measures," Microelectronics, 2000. ICM 2000, Proceedings of the 12th International Conference on, 2000.

[3] V. Sindhu and R. Kumar," Analysis simulation and comparison of different multiplier algorithms," International Journal of Management, IT and Engineering, vol. I, no. 3, pp. 146-156,2011.

[4] Rajaram, Srinath, and K. Vanithamani,"Improvement of Wallace multipliers using parallel prefix adders." In Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on, pp. 781-784. IEEE, 2011.

[5] Vinoth, C., VS Kanchana Bhaaskaran, B. Brindha, S. Sakthikumaran, V. Kavinilavu, B. Bhaskar, M. Kanagasabapathy, and B. Sharath. "A novel low power and high speed Wallace tree multiplier for RISC processor." In Electronics Computer Technology (ICECT), 2011 3rd International Conference on, vol. 1, pp. 330-334. IEEE, 2011.

[6] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," *Low Power Electronics and Design (ISLPED)* 2011 International Symposium on. 1-3 Aug. 2011.

[7]S. Cheemalavagu, P. Korkmaz, K.V. Palem, B.E.S. Akgul, and L.N. Chakrapani, "A probabilistic CMOS switch and its realization by exploiting noise," in *Proc. IFIP-VLSI SoC*, Perth, Western Australia, Oct. 2005.

[8] H.R. Mahdiani, A. Ahmadi, S.M. Fakhraie, C. Lucas, "Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 850-862, April 2010.

[9] S.K. Sangjin, S. Hong, M.C. Papaefthymiou and E. Stark,"Low Parallel Multiplier design for DSP applications through coefficient optimization", International ASIC/SOC conference pp.286-290,1999.

[10] V.Gupta, D.Mohapata,S.P.Park,A.Raghunathan and K.Roy,"IMAPCT: IM precise adder for low-power approximate and omputing", in low power electronics and design(ISLED) international symposium on pp.409-414, 2011.

[11] A.B. Kahng and S.Kang,"Accuracy-configurable adder for approximate arithmetic design",in design automatic conference(DAC),pp.820-825,2012.

[12] D.Shin nd S.K.unita," Approximate logic synthesis for error tolerance applications", in design, Automatic and test in Europe conference and exhibition(DATE), pp. 957-960,2010.

[13] Anju, S., &Saravanan, M. (2013). High Performance Dadda Multiplier Implementation Using High Speed Carry Select Adder. International Journal of Advanced Research in Computer and Communication Engineering, 2(3).

[14] Wang, Z., Jullien, G. A., & Miller, W. C. (1995). A new design technique for column compression multipliers. IEEE Transactions on Computers, 44(8), 962-970

[15] Anitha, P., &Ramanathan, P. (2014, February). A new hybrid multiplier using Dadda and Wallace method. In Electronics and Communication Systems (ICECS), 2014 International Conference on (pp. 1-4). IEEE.

[16] Chu, W., Unwala, A. I., Wu, P., &Swartzlander, E. E. (2013, November). Implementation of a high speed multiplier using carry lookahead adders. In Signals, Systems and Computers, 2013 Asilomar Conference on (pp. 400-404). IEEE.

[17] Samundiswary, P., &Anitha, K. (2013). Design and Analysis of CMOS Based DADDA Multiplier. International Journal of Computational Engineering and Management IJCEM, 1(16), 12-17.

[18] V. G. Oklobdzija, D. Villeger,"Improving Multiplier Design by using Improved Column Compression Tree optimized Final Adder in CMOS technology", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 3. Issue-2, pp 292-301, 1892.

[19] M. Rouholamini, O. Kavehie, A.-P. Mirbaha, S. J. Jasbi, and K. Navi, “A new design for 7:2 compressors,” in Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. (AICCSA), Amman, Jordan, May 2007, pp. 474–478.

[20] J. S. Altermann, E. A. C. da Costa, and S. Bampi, “Fast forward and inverse transforms for the H.264/AVC standard using hierarchical adder compressors,” in Proc. IEEE/IFIP Int. Conf. VLSI Syst. Chip (VLSI-SoC), Madrid, Spain, Sep. 2010, pp. 310–315.

[21] C.S. Wallace,”A suggestion for a fast multiplier”, IEEE Transactions on Electronics, pp 14-17, 1964.

[22] Chia Hao Cin Lin, Ing-Chao lin,”High accuracy approximate multiplier with error correction”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp 33-38, 2013.

[23] A Momeni and F. Lombardi,” Design and Analysis of Approximate Compressors for Multiplication”, IEEE Transactions on Computers, pp 1-11, 2015.

[24] A. Guyot, B. Hochet, and J. Muller, "A way to build efficient carryskip adders, " IEEE Transactions on Computers, vol. 36, no. 10, pp. 1144--1152, 1987.

[25] M. Ortiz, F. Quiles, J. Hormigo, F. J. Jaime, J. Villalba, and E. L. Zapata, "Efficient implementation of carry-save adders in fpgas, " in Applicationspecific Systems, Architectures and Processors, 2009. ASAP 2009. 20th IEEE International Conference on. IEEE, 2009, pp. 207-210.

[26] R. Uma, V. Vijayan, M. Mohanapriya, and S. Paul, "Area, delay and power comparison of adder topologies," International Journal of VLSI and Communication Systems, vol. 254, 2012.

[27] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder, " Very Large Scale Integration (VLSl) Systems, IEEE Transactions on, vol. 20, no. 2, pp. 371-375, 2012.

[28] Joao S. Altermann, Eduardo A. C. da Costa and Sergio Bampi, "Fast Forward and Inverse Transforms for the H.264/AVC Standard Using Hierarchical Adder Compressors, " IEEE Transactions on Very Large Scale Integration (VLSl) Systems, pp. 310-315, 2010.

[29] Bianca Silveira, Guilherme Paim, Cláudio Machado Dinizand and Sergio Bampi, "Power-Efficient Sum of Absolute Differences Hardware Architecture Using Adder Compressors for Integer Motion Estimation Design," IEEE Transactions on circuits and systems–I, pp. 1-12, 2017.