International Journal of Engineering & Technology, 7 (4) (2018) 2386-2390

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.12261
Research paper

Design of high speed Wallace tree multiplier using 8-2 and 4-2 adder compressors

E. Jagadeeswara Rao [1] *, K. Jayaram Kumar [2], Dr. T. V. Prasad [3]

[1] Assistant professor, Department of Electronics and communication Engineering, Aditya College of Engineering and Technology, Surampalem, Andhra Pradesh, India
[2] Assistant professor, Department of Electronics and communication Engineering, Godavari institute of Engineering and Technology, Rajahmundry, Andhra Pradesh, India
[3] Principal, Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh, India
**Corresponding author E-mail: emandi.jagadeesh@gmail.com*

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 improv-ing 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 present an approach towards the reduction of delay in Wallace tree multipliers by using 8:2 and 4:2 adder com-pressors, in the partial product reduction stage. The proposed design is also compared to the Wallace Tree multiplier which uses 4:2 and 8:2 adder compressors in terms of propagation delay. The proposed design enhances speed of the system by 74.1% compared to the conven-tional Wallace Tree multiplier, while 24.1 % reduction was achieved in the delay of the system relative to Wallace tree multiplier with 16-bit adder with one of the 8-2 adder compressors.

Keywords: *Wallace Tree Multiplier; Compressors and Adders.*

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 day's 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 to 4-bit and 8-bit multiplication algorithms are given in Section 4 and high speed 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 arithmetic 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. Gupta et al. [10] made a reduction approximate multiplier at transistor level. For simplifying logic area Shin et 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 analysing 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 (CLA) 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.

3. Compressors

Compressors 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 XORs mainly) [18]. The compressor design implemented in this paper prefers both MUXs and XORs.

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 expression (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.

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

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 three XOR gates give the maximum delay. The 4-2 compressor has five inputs (A, B, C, D, C_{in}), where C_{in} is the input carry, and three outputs (Sum, Carry and C_{out}). In this adder compressor, the carry output C_{out} is independent of the input carry (C_{in}), 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).

$$S = Sum + 2 * (C_{out} + Carry) \quad (2)$$

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).

$$S = Sum + 2* (C_{out1} + C_{out2} + Carry) \quad (3)$$

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).

$$S = Sum + 2 * (C_{out1} + C_{out2} + Carry) \quad (4)$$

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).

$$S = Sum + 2 * (C_{out0} + C_{out1} + C_{out2} + C_{out3} + C_{out4} + Carry) \quad (5)$$

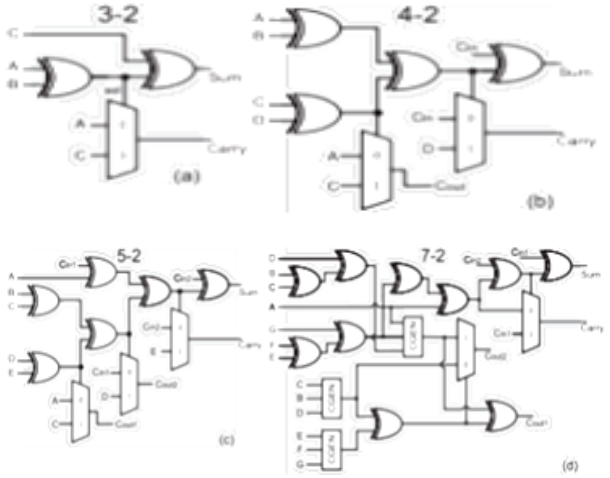

Fig. 1: Adder Compressors Internal Structures: (A) 3-2; (B) 4-2; (C) 5-2; (D) 7-2.

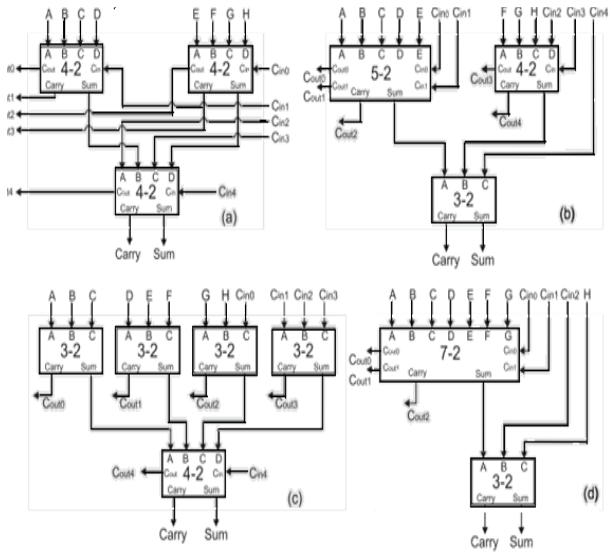

Fig. 2: The Structure of 8-2 Adder Compressor Using: (A) Only 4-2 Adder Compressors; (B) Combination of 5-2, 4-2 and 3-2 Adder Compressors; (C) Combination of 3-2 and 4-2 Adder Compressors; (D) Combination of 7-2 and 3-2 Adder Compressors [20].

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). Design of 4-2 adder compressor using MUX in the place of XOR gates for increasing the speed was discussed in [22].

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

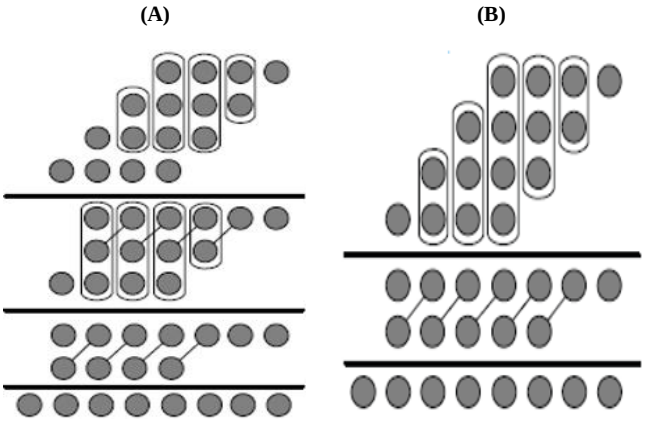

Fig. 3: A 4×4 Wallace Multiplier Dot Rotation with (A) Only Half-Adders and Full-Adder (B) Only Half-Adder, Full-Adder and 4-2 Adder Compressor [22].

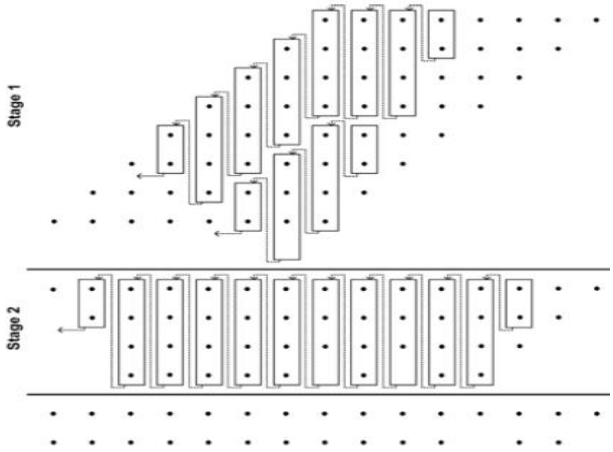

Fig. 4: Reduction Circuitry of an 8×8 Dadda Multiplier Using 4-2 Adder Compressor [23].

5. High speed adders

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 CLA, 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. CLAs have a minimum delay but areas 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 does not 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 done in carry save adder for sacrificing its speed for area [27].

To implement a W-bit 3-2, 4-2, 5-2 and 7-2 adder compressors it is needed a recombination of partial Carry and Sum terms. To make the recombination of Carry and Sum it is used a cascade of half-adder and full-adders circuits in a Ripple Carry form, as presented in the example of the Fig 3, for an 8-bit 4-2 compressor. In this work we use a more efficient CLA adder to perform the recombination of the partial results. We have used a pipeline stage between the line of the compressors and the adder line used to recombine the partial results (highlighted in Fig 3).

Fig. 6 presents an addition of eight 8-bit values as an example. It is noted in Fig. 6 that adder circuits are required to recombine the partial sums of previous values (i.e. recombination line), since a Carry signal from the compressor n must be added with the Sum signal of the compressor n + 1 to generate the final sum (S) of bit n + 1.

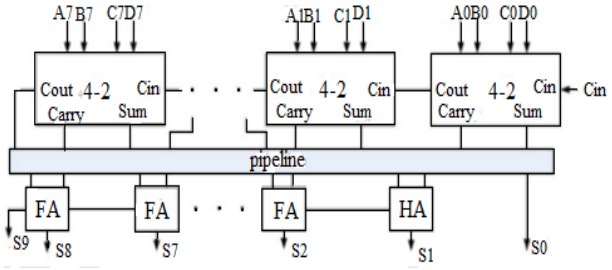

Fig. 5: 8-Bit Addition Using 4-2 Adder Compressors [28].

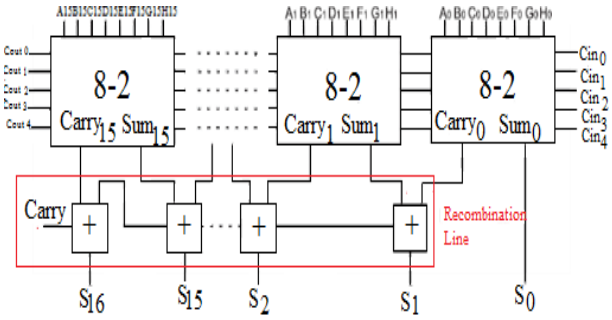

Fig. 6: 16-Bit Adder Using 8-2 Compressors [29].

6. Proposed high speed multipliers

If we increase the speed of the any multiplier either reduce the area of the partial products generation or reduce final sum. In this paper, proposed 4-bit Wallace tree multiplier design using 8-bit adder using 4-2 adder compressor and 8-bit Wallace tree multiplier using 16-bit adder using different 8-2 adder compressors and these compressors discuss in section 3. The proposed 4-bit Wallace multiplier is shown in Fig. 7(a) and proposed 8-bit multiplier is shown in Fig. 7(b). The two multipliers develop a code in Verilog and simulate using Xilinx 14.7 and these delays results were compared with different Wallace tree multipliers developed by various researchers and these results and relevant explanations given, see section 7.

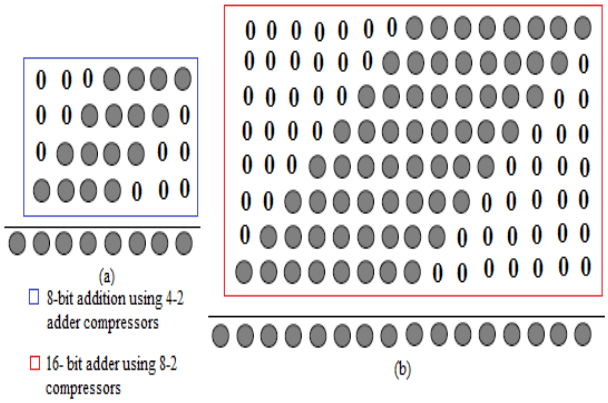

Fig. 7: Wallace Tree Multipliers (A) Four-Bit Wallace Tree Multiplier with 4-2 Adder Compressor, (B) 8-Bit Wallace Tree Multiplier with Different 8-2 Adder Compressors.

7. Results and discussion

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 results of the delay obtained from the proposed design of the novel Wallace tree multiplier and the results published by contemporary researchers.

Table 1: Delays in Wallace Tree Multipliers of 8x8 Configurations

Design	Multipliers Types	Delay(ns)
Method-1	Wallace multiplier with Sklansky Adder[30]	28.323
Method-2	Wallace multiplier with kogge-stone adder[30]	26.090
Method-3	Reduced complexity Wallace multiplier with Sklansky Adder[30]	28.849
Method-4	Reduced complexity Wallace multiplier with kogge-stone adder[30]	27.457
Method-5	Conventional Wallace tree multiplier with carry save adder and carry propagation adder [30]	33.13
Method-6	Design for Wallace tree using adder compressors and Landner-Fsicher adder[31]	19.281
Method-7	Wallace tree multiplier with 4-2 adder compressor[23]	14.666
Method-8	Wallace tree multiplier with half adders and full adders(conventional method)	16.829
Method-9	Proposed Wallace tree multiplier with 16-bit adder compressors with Fig. 2(a) 8-2 adder compressor	22.143
Method-10	Proposed Wallace tree multiplier with 16-bit adder compressors with Fig. 2(b) 8-2 adder compressor	11.807
Method-11	Proposed Wallace tree multiplier with 16-bit adder compressors with Fig. 2(c) 8-2 adder compressor	12.601
Method-12	Proposed Wallace tree multiplier with 16-bit adder compressors with Fig. 2(d) 8-2 adder compressor	9.033

Table 2: Delays in Wallace Tree Multipliers of 4x4 Configurations

Design	Multipliers Types	Delay(ns)
Method-1	Wallace multiplier with half adder and full adder	13.57
Method-2	Wallace multiplier with 4-2 adder compressor	13.97
Method-3	Proposed Wallace tree multiplier with 8-bit adder with 4-2 adder compressor	8.797

The delays of different 8-bit Wallace tree multiplier is shown in Table 1 and this table we can consider different researchers developed Wallace tree multiplier delays also in Table 2 give the different delays of 4-bit multipliers. Relevant graphs of above tables shown in Fig. 8(a) and Fig. 8(b). In Fig. 8(a) Method-12 (9.033 ns) gives very less delay compare to other methods and indicates in dark blue colour. Also large delay appears in method-5 (33.13 ns) indicates in light red colour and dark red. Method-1 to Method-4 approximately give the same delay indicates in graph dark red colour.

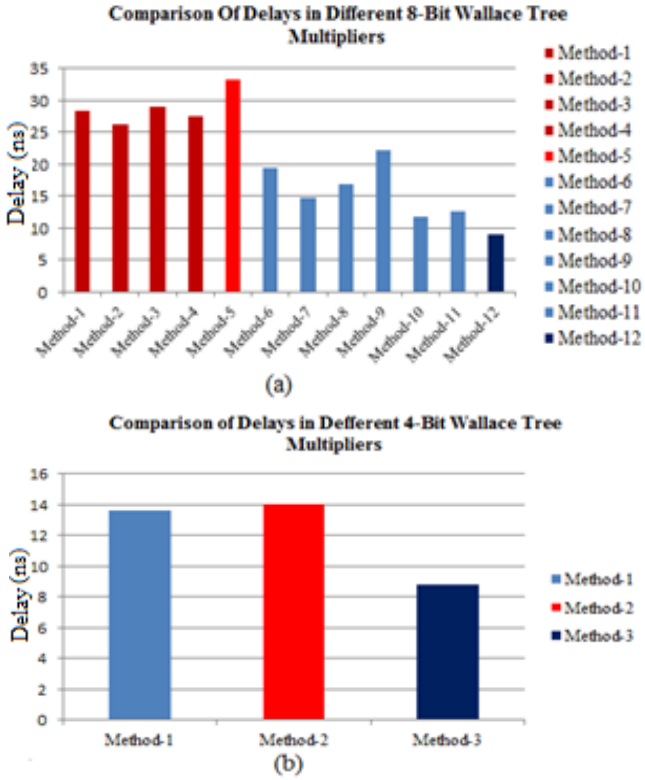

Fig. 8: Comparison of Different Wallace Tree Multipliers (a) Only 8-Bit Multipliers, (b) Only 4-Bit Multipliers.

The relevant graph of Table-2 is shown in Fig. 8(b). This graph mainly tells about how to vary the delay of 4-Bit Wallace tree multiplier with normal adders, 4-2 adder compressors and 8-Bit adder using 4-2 adder compressors. After executing of program we can less delay in method-3(8.797 ns) indicates in graph dark blue colour and large delay get method-2(13.97 ns) indicates in dark red colour and conventional method give little bit less delay(13.57 ns) compare to method-2 indicates sky colour.

7.1. Delay

Normally delay of Wallace Tree Multiplier depends on the number of reduction stages and the delay of each stage. In method 9 to method 12 multipliers with four 16-bit adders and each 16-bit adder implemented with one of the four 8-2 adder compressors. Each proposed multiplier compared to an exact multiplier. The delay improvement is tabulated in Table 3. Also delay improvement of proposed 4-bit multiplier compared to exact multiplier and multiplier using 4-2 adder compressors are shown in Table 4. When comparing the multiplier with 4-2 adder compressors delay increases by 1 % due to conventional multiplier and is indicated by '-' sign in Table 4.

Table 3: Delay Improvement of Proposed 8-Bit Wallace Tree Multiplier

Design	Improvement (%)
Method 9	11
Method 10	21.4
Method 11	20.5
Method 12	24.1

Table 4: Delay Improvement of Proposed 4-Bit Wallace Tree Multiplier

Design	Improvement (%)
Method 2	-1
Method 3	0.8

8. Conclusions

From the obtained results for the proposed design, it can be seen that use of 4-2 and 8-2 adder compressors can enhance performance of the system significantly. The results and the comparisons were presented clearly illustrating the advantages of the proposed design. Enhanced speed of the system by 74.1% as compared to the conventional Wallace Tree multiplier was achieved. Reduction in the delay of the system by 24.1% relative to Wallace tree multiplier with 16-bit adder with one of the 8-2 adder compressor was also obtained.

Results of proposed design of 4-bit Wallace Tree Multiplier using 8-bit adder using adder compressor were obtained and a delay improvement result was discussed. Finally, experiments achieved very high speed multiplier, and the proposed 4-bit Wallace Tree Multiplier improved delay of 53.1% compared to an exact multiplier and 54 % compared to multiplier with 4-2 adder compressor.

References

[1] S. Akhter and S. Chaturvedi, "HDL based implementation of n x n bit serial multiplier", Proc. of 2014 IEEE Int. Conf. on Sig. Processing and Integrated Networks (SPIN), pp. 470-474, 2014.

[2] S. Shah, A. AI-Khalili, and D. AI-Khalili, "Comparison of 32-bit Multipliers for Various Performance Measures", Proc. of the 12th Int. Conf. on Microelectronics, ICM 2000, 2000.

[3] V. Sindhu and R. Kumar, "Analysis Simulation and Comparison of Different Multiplier Algorithms", Int. J. of Management, IT and Engg. Vol. I, No. 3, pp. 146-156, 2011.

[4] Rajaram, Srinath, and K. Vanithamani, "Improvement of Wallace Multipliers using Parallel Prefix Adders", Proc. of 2011 Int. Conf. on Sig. Processing, Comm., Computing and Networking Technologies (ICSCCN), 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", Proc. of 3rd Int. Conf. on Electronics Computer Tech. (ICECT 2011), Vol. 1, pp. 330-334, IEEE, 2011.

[6] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for Low-Power Approximate Computing", Proc. of Int. Symp. on Low Power Electronics and Design (ISLPED). 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, 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 Trans. 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", Int. ASIC/SOC Conf., 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 computing", Proc. of Int. Symp. on Low Power Electronics and Design (ISLED), pp. 409-414, 2011.

[11] A.B. Kahng and S. Kang, "Accuracy-configurable adder for Approximate Arithmetic Design", Proc. of Design Automatic Conf. (DAC), pp. 820-825, 2012.

[12] D. Shin and S. Kunita, "Approximate logic synthesis for error tolerance applications", Proc. of Design, Automatic and Test in Europe Conf. and Exhibition (DATE), pp. 957-960, 2010.

[13] Anju, S., and Saravanan, M., "High Performance Dadda Multiplier Implementation Using High Speed Carry Select Adder", Int. J. of Adv. Res. in Computer and Comm. Engg., 2(3), 2013.

[14] Wang, Z., Jullien, G. A., and Miller, W. C., "A New Design Technique for Column Compression Multipliers", IEEE Trans. on Computers, 44(8), pp. 962-970, 1995.

[15] Anitha, P., and Ramanathan, P., "A New Hybrid Multiplier using Dadda and Wallace Method", Proc. of Int. Conf. on Electronics and Comm. Systems (ICECS), pp. 1-4, IEEE, 2014.

[16] Chu, W., Unwala, A. I., Wu, P., and Swartzlander, E. E., "Implementation of a High Speed Multiplier using Carry Look-Ahead Adders", Proc. of Asilomar Conf. on Signals, Systems and Computers (pp. 400-404). IEEE, 2013.

[17] Samundiswary, P., Anitha, K., "Design and Analysis of CMOS Based DADDA Multiplier", Int. J. of Comp. Engg. and Management IJCEM, 1(16), 12-17, 2013.

[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, 1982.

[19] M. Rouholamini, O. Kavehie, A.-P. Mirbaha, S. J. Jasbi, and K. Navi, "A new design for 7:2 compressors," 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, pp. 310–315, Sep. 2010.

[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 Trans. 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 Trans. on Computers, pp 1-11, 2015

[24] Guyot, B. Hochet, and J. Muller, "A Way to Build Efficient Carry Skip Adders, " IEEE Trans. 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", Proc. of 20th IEEE Int. Conf. on App. Specific Systems, Architectures and Processors (ASAP 2009), IEEE, pp. 207-210, 2009

[26] R. Uma, V. Vijayan, M. Mohanapriya, and S. Paul, "Area, Delay and Power Comparison of Adder Topologies," Int. J. 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 Trans. 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 Trans. on Circuits and Systems–I, pp. 1-12, 2017.

[30] S. Rajaram and Mrs. K. Vanithamani, "Improvement of Wallace Tree Multipliers using Parallel Prefix Adders", Proc. of 2011 Int. Conf. on Sig. Processing, Comm., Computing and Networking Technologies (ICSCCN 2011), 2011.

[31] Raiyyan E Masumdar, "Design of high performance Wallace Tree Multiplier using Compressors and Parallel Prefix Adders," Int. J. of Electrical, Electronics and Data Comm., Vol. 4, 2016.