Non-Linear Carry Select Adder Based Enhanced Wallace Tree Multiplier Structure

Damarla Paradhasaradhi, M. Prashanthi

Abstract— A multiplier is one of the key hardware blocks in most digital and high performance systems such as FIR filters, micro processors and digital signal processors etc. Deign of a high performance and high-density multiplier is presented. This multiplier is designed by using the Wallace tree structure with pipelining the Carry Select Adder (CSLA) provides a good compromise between cost and performance in carry propagation adder design. A Non-Linear Carry Select Adder (Square root carry select adder) using RCA is introduced but it offers some speed penalty. However, conventional CSLA is still area-consuming due to the dual ripple carry adder structure. In the proposed work, generally in Wallace multiplier the partial products are reduced as soon as possible and the final carry propagation path carry select adder is used. In this paper, modification is done at gate level to reduce area and power consumption. The Modified Non-Linear Carry Select-Adder (MCSLA) is designed using Common Boolean Logic and then compared with regular CSLA respective architectures, and this MCSLA is implemented in Wallace Tree Multiplier. This work gives the reduced area compared to normal Wallace tree multiplier. Finally an area efficient Wallace tree multiplier is designed using common Boolean logic based Non-Linear carry select adder. This work gives the performance of the Wallace tree multiplier in terms of area, delay and their products may be implementing in Xilinx FPGA.

Index Terms— Wallace Tree Multiplier, Non-Linear Carry Select Adder (NL CSLA), Carry Select Adder (CSLA), Common Boolean Logic (CBL), Field Programmable Gate Array (FPGA)

1 Introduction

MULPILIERS are among the essential components of many digital systems consequently; their power dissipation and speed are of primary anxiety. For portable applications where the power consumption is the most important parameter, one should be reduce the power dissipation as much as possible. One of the best background to reduce the dynamic power dissipation, hereafter referred to as power dissipation in this project, is to minimize the total switching activity means the total number of signal transitions of the system. The task that took most of the processor’s time is multiplication thus enhancing the performance of multiplier leads to better performance of processor especially in field of digital signal processing and data processing ASIC. Many application systems based on DSP require extremely fast processing of a huge amount of digital data. The multiplier is an essential element of the digital signal processing such as filtering and convolution. The demand of fast processors is increasing for high-speed data processing. Since the multiplier requires the longest delay among the basic operational blocks in digital system [3]. Any multiplier can be divided into three stages: Partial products generation stage these are generated by AND operation, partial products addition stage can be carried by different adders, and the final addition stage. Many high-performance algorithms and architectures have been proposed to accelerate multiplication. The speed of multiplication can be increased by reducing the number of partial products. Various multiplication algorithms such as Modified Booth, Booth, Braun, and Baugh-Wooley have been proposed.

This paper work presents two different form of Wallace tree multiplier using two different adder circuits namely Ripple carry adder and carry select adder. Ripple Carry Adder (RCAs) have the most compact design among all types of adders. After developing these two different forms of Wallace tree multiplier a comparative study is being carried out on the basis of area and power consumption by the two designs. The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independent generation multiple carries and then select a carry to generate the sum.

whereas, the CSLA is not an area efficient because it uses multiple pairs of Ripple Carry Adders(RCA) to generate partial sum and carry by considering carry input as C_{in}=0 and C_{in}=1. And the final summation and carry are selected by the multiplexers [1]. The proposed architecture generates a duplicate sum and carry-out signal by using NOT and OR gate and select value with the help of multiplexer. BY using the multiplexer select the correct output according to it’s previously carry out signal. Thus it can be interpreted from this fact that addition is a sub-process in multiplication criterion that has to be satisfied. In the process the Wallace tree arrangement aligned the partial products in form of a tree and then with the help of fast adders final product is obtained.

This paper is organized as follows; Section II describes the conventional and modified Wallace tree architectures and section III explains the Modified Wallace tree Structure with RCA and normal Non-Linear CSLA respectively. A section IV deals with proposed Wallace tree architecture using Common Boolean Logic (CBL) based Non-Linear CSLA. Results are analyzed in the section V. and finally Section VI with the conclusion.

- *Damarla Paradhasaradhi is currently pursuing masters degree program in electronics engineering in Pondicherry University, India, PH-+919786919697. E-mail: dpardhasaradhi@hotmail.com*
- *M. Prashanthi is currently pursuing masters degree program in electronics engineering in Pondicherry University, India, PH-+919489382096. E-mail: veenam2000@gmail.com*

2 Wallace Tree Architectures

The Wallace tree multiplier is significantly faster than a simple array multiplier because its height is logarithmic in word size. However, in addition to the large number of adders required, the Wallace tree understanding is much less regular and more complicated. The Wallace tree multiplier is a high speed multiplier [3]. As a result, these are often avoided by designers, because the design complexity is a concern to them.

The summing of the partial product bits in parallel using a tree of carry-save adders became generally known as the “Wallace Tree Multiplier”. The three main ladders are used to multiply two numbers.

- Configuration of partial products.
- Diminution of the partial products matrix into a two row matrix by means of a carry save adder.
- Addition of remaining two rows using a faster Carry Look Ahead Adder(CLA).

2.1 Conventional Wallace Tree Multiplier

In the conventional Wallace Tree multiplier the partial products are formed by N^2 AND gates in the same manner as that of Dadda multiplier. The produced partial products are collected to group of three or two. Since the Wallace multiplier performs the reduction as soon as possible the number of half adders and full adders required is high [4]. The Basic conventional Wallace multiplier for N=8 is shown in Fig. 1.

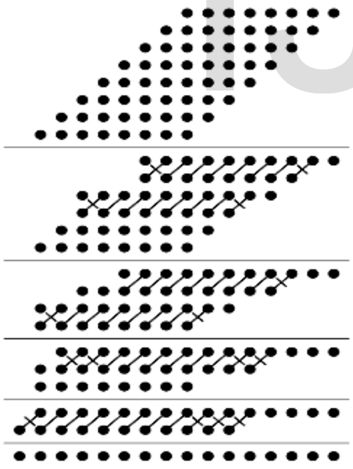

Fig. 1. Conventional Wallace Tree Multiplier

2.2 Modified Wallace Tree Multiplier

The Modified Wallace multiplier is similar to that of Conventional Wallace multiplier in that it uses as so many full adders as possible, only different in that it only use half adders when necessary to ensure that the number of reduction stage is same as for Conventional Wallace Tree multiplier. The Modified Wallace Tree at first make the partial product formed into the pyramidal structure and divide the structure into tree rows of group and uses full adders for each group of three bits in a column. The modified Wallace tree multiplier is shown in Fig. 2. A group of three bits in a column is not processed, it is passed on to the next stage [5]. Single bits are passed on to the next stage as in the conventional Wallace reduction. The Fig. 3 explains the algorithm for the modified Wallace tree multiplier structure algorithm. The three main steps in the algorithm of Wallace tree multiplier design are:

1. The multiplication of the multiplier bits with the multiplicand generates a bit product stream.
2. The bit product matrix thus formed has been reduced into less number of rows with the help of half and full adders, this step persist till the final addition is done.
3. Last step is the final addition using adders and the final result can be obtained after this step.

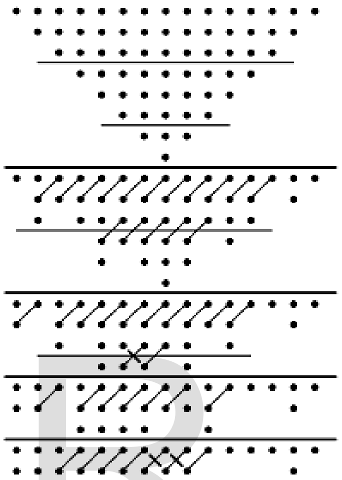

Fig. 2. Modified Wallace Tree Multiplier

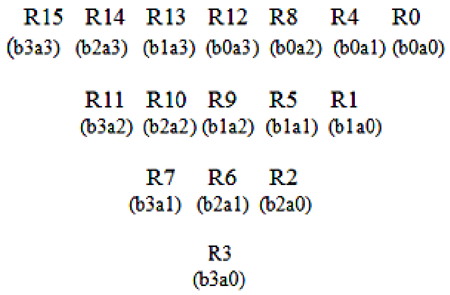

Fig. 3. Modified Wallace Tree Multiplier with Pyramid Structure

The Fig. 3. explains the Wallace tree formation using partial products for 4-bits. Here R0, R1, R2,…R15 are the partial products of the multiplicand and multiplier.

3 Modified Wallace Tree Multiplier using RCA and Non-Linear CSLA

3.1 Modified Wallace Tree Multiplier using RCA

The basic architecture for Modified Wallace Tree Multiplier using ripple carry adder is shown in Fig. 4. In this architecture the both multiplier and multiplicand are AND ed together and generates partial products and then an Wallace Tree is implemented.

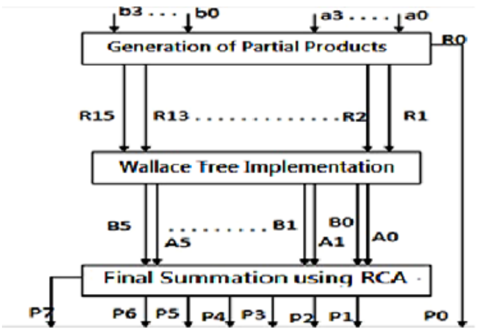
b3 . . . b0
a3 . . . . a0
Generation of Partial Products
R0
R15
R13 . . . . R2
R1
Wallace Tree Implementation
B5
. . . . B1
A5
B0
A1
A0
Final Summation using RCA
P7
P6
P5
P4
P3
P2
P1
P0

Fig. 4. Modified Wallace Tree Multiplier Architecture using RCA

In this arrangement the partial products with same weight are grouped together and written in a column. Here total six columns in the arrangement in accordance to the max weight carried by partial product term i.e. a3b3 has top weight 6. Now the addition is applied in each column using half and full adder according to the require, if there are two numbers to be added then half adder is employed and if there are three numbers for addition then full adder has to be used. The main intention of Wallace tree implementation in this architecture is to generate the input terms for adders whether RCA. As we know the first partial product a0b0 doesn't need any computation, it is directly in use as the LSB of the product. Therefore R0 is equals to P0 [4]. Now the task we have to do is to extract two final bits from each remaining column after excluding the first column from right as it contains R0. Now suspiciously examining the columns noticed that the second column from right has two partial products P4 and P1 that means it has only two bits in all so they can be taken directly and termed as A0 and B0. Moving additional to the next column we are having three terms with us P8, P5 and P2, so here apply adders to get final two bits for RCA. A half adder is applied with p8 and P5 as inputs which gives two outputs sum and carry, now from this column the sum obtain is in use as B1, the partial product P2 as A1 and the carry so obtained as A2. Going further on in this manner, six pairs of bits as A0B0, A1B1, A2B2, A3B3, A4B4, and A5B5 are obtained. These numbers so obtain can be treated as two input numbers for the adder [5], [6]. The Wallace tree implementation with the RCA can be shown in Fig. 5.

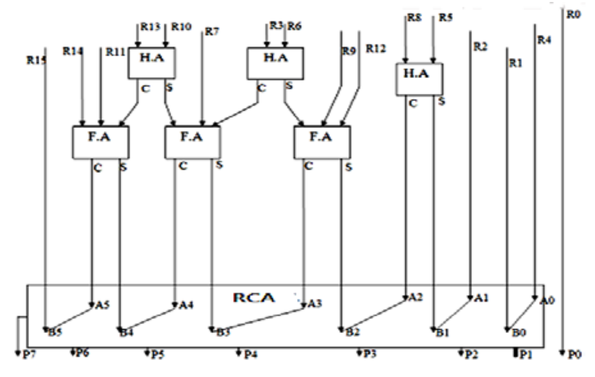

Fig. 5. Wallace Tree Multiplier using RCA

Similarly the RCA is replaced with Non-Linear Carry Select adder and the architecture and the tree formation is shown in the following Fig. 6, 7.

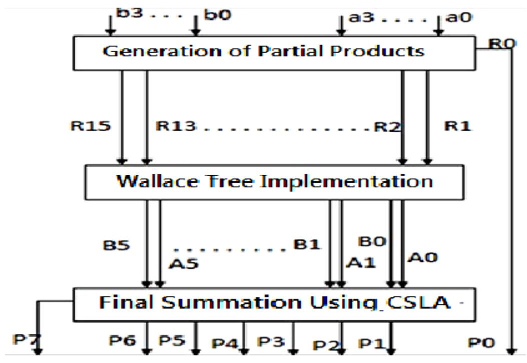
b3 . . . b0
a3 . . . . a0
Generation of Partial Products
R0
R15
R13 . . . . R2
R1
Wallace Tree Implementation
B5
. . . . B1
A5
B0
A1
A0
Final Summation Using CSLA
P7
P6
P5
P4
P3
P2
P1
P0

Fig. 6. Modified Wallace Tree Multiplier Architecture using Non-Linear CSLA

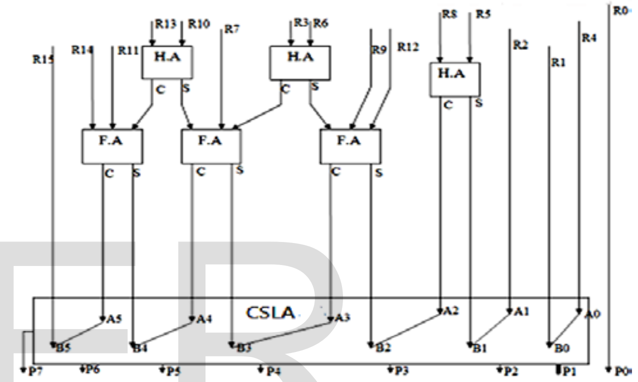

Fig. 7. Wallace Tree Multiplier using Non-Linear CSLA

4 Modified Wallace Tree Multiplier using Common Boolean Logic based Non-Linear CSLA

In proposed work the normal Non-Linear CSLA is replaced with the Common Boolean logic based Non-Linear CSLA. This architecture of 16-bit CBL based Non-Linear CSLA is shown in Fig. 8. And the area evaluations of internal structures are calculated by using the gate level simulation [2]. This architecture is implemented on the Wallace tree Multiplier and that final structure is shown in below Fig. 9.

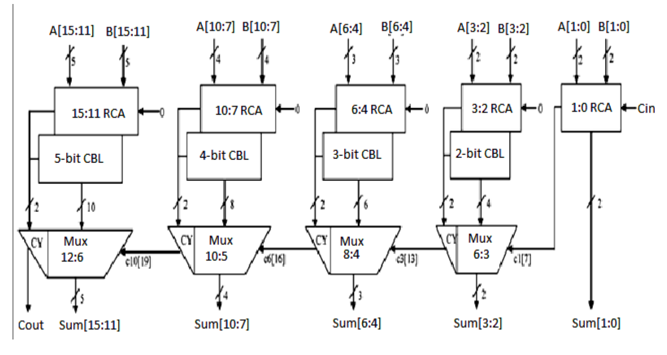

Fig. 8. 16-bit CBL based Non Linear CSLA

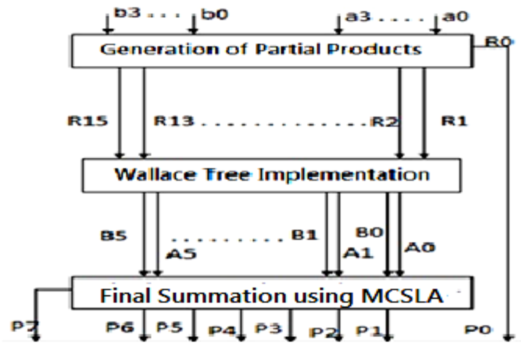
b3 . . . b0
a3 . . . . a0
B0
Generation of Partial Products
R15
R13 . . . . R2
R1
Wallace Tree Implementation
B5
A5
. . . . B1
B0
A1
A0
Final Summation using MCSLA
P7
P6
P5
P4
P3
P2
P1
P0

Fig. 9. Wallace Tree Multiplier Architecture using CBL bsed Non-Linear CSLA

5 Result Summary

The comparison table for the 8-bit and 16-bit Wallace tree Multiplier using RCA, normal Non-Linear CSLA and Common Boolean Logic based Non-Linear CSLA are shown in Table 1. And Table 2. The total simulation done in Xilinx ISE 14.2 targeted Spartan 3E family. From the both tables the number of slices are reducing comparing with RCA to CBL based Non-Linear CSLA Wallace tree Multiplier. And slight increase in delay. The no.of logic levels are decreasing from RCA to Nonlinar CSLA. So this gives an area efficient Wallace tree multiplier using CBL based Non-Linear CSLA.

TABLE 1
Delay and Area Comparision of 8-bit Wallace Tree Multiplier

Wallace Multiplier	Delay (ns)	No. of Slices	Logic Levels
Using RCA	15.36	32	14
Using NL CSLA	17.05	28	12
Using CBL based NL CSLA	16.13	22	10

TABLE 2
Delay and Area Comparision of 16-bit Wallace Tree Multiplier

Wallace Multiplier	Delay (ns)	No. of Slices	Logic Levels
Using RCA	35.62	57	28
Using NL CSLA	38.47	50	25
Using CBL based NL CSLA	36.73	46	23

6 Conclusion

In this paper, an area efficient Wallace tree multiplier using Common Boolean logic (CBL) based Non-Linear carry select adder is proposed. By sharing the common Boolean logic (CBL) term, the duplicated adder cells in the regular carry select adder is removed. The reduce number of gates of this work offers the great advantage in the reduction of delay and area. It would be interesting to test the design of the 32 and 64 bits. This work may helpful for designing of MAC unit.

Acknowledgment

The authors wish to thank to K. Anusudha, R. Nakkeeran and P. Samundiswary of the Department of Electronics Engineering, Pondicherry University, Pondicherry, India, for providing softwares and contributions to this work.

References

[1] Damarla Paradhasaradhi and K. Anusudha, “An Area Efficient Enhanced SQRT Carry Select Adder”, International Journal of Engineering Research and Applications, vol. 3, Issue 6, Nov-Dec 2013.

[2] Jasbir Kaur and Kavita, “ Structural VHDL Implementation of Wallace Multiplier”, International Journal of Scientific & Engineering Research, vol. 4, Issue. 4, pp. 1829-1833, April 2013.

[3] B. Ramkumar and Harish M Kittur, “Low power and area efficient carry select adder”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 2, pp. 371-375, Feb 2012.

[4] N. Sureka, R. Porselvi and K. Kumuthapriya, “An efficient high speed Wallace tree Multiplier”, Proceedings of IEEE International Conference on Information Communication and Embedded Systems, pp. 1023-1026, Feb 2013.

[5] Naveen K Gahlan, Prabhat, Jasbir Kaur “ Implementation of Wallace Tree Multiplier Using Compressor” International Journal of Computer & Technology.

[6] W.J. Townsend, E.E. Swartzlander Jr., and J.A. Abraham, “A Comparison of Dadda and Wallace Multiplier Delays,” Proc. SPIE, Advanced Signal Processing Algorithms, Architectures, and Implementations XIII, pp. 552-560, 2003.

[7] B. Ramkumar , Harish 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, Jun 2010.

[8] P. Sreenivasulu, K. Srinivasa rao, Malla Reddy and A. Vinay Babu, “Energy and area efficient carry select adder on a reconfigurable hardwware”, International Journal of Engineering Research and Applicaions, vol. 2, Issue. 2, pp. 436-440, Mar 2012.

[9] R. Priya and J. Senthilkumar, “Implementation and comparision of effective area efficient architecuture for CSLA”, Proceedings of IEEE International Conference on Emerging trends in Computing, Communicaton and Nano Technology, pp. 287-292, 2013.

[10] I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin and Chien-Chang Peng, “An area efficient carry select adder design by sharing the common boolean logic term”, Proceedings on the International Multiconference of Engineering and computer scientist, IMECS 2012.

[11] 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.

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

[13] 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.

[14] Edison A. J and C. S. Manikanda babu, “An efficient CSLA architecture for VLSI hardware implementation”, Interanational Journal for Mechanical and Industrial Engineering, vol. 2, Issue 5, 2012.