DOI : https://doi.org/10.56452/6-1-26

ISSN: 0974-5823 Vol. 6 No. 1 June, 2021

International Journal of Mechanical Engineering

DESIGN AND ANALYSIS OF HIGH SPEED WALLACE TREE MULTIPLIER USING PARALLEL PREFIX ADDERS FOR VLSI CIRCUIT DESIGNS

[1]K.A.Rahaman,[2]Dr.S.Govinda rajulu,[3]GUMMA VAJJULA SINDHUJA

[13]Assistant Professor,[2]Professor, Department of ECE
Dr. K V Subba Reddy Institute Of Technology

ABSTRACT

Arithmetic and logic unit has been the most significant unit in any electronic devices. Multiplication is one of the most widely used arithmetic operations. In the recent advancement, for an arithmetic and logic unit to be significant it needs to have an efficient algorithmic operation. There are many multiplication algorithms are proposed, by using which multiplier structure can be designed. Among various multiplication algorithms, Wallace tree multiplication algorithm is beneficial in terms of speed of operation. With the advancement of technology, demand for circuits with high speed and low area is increasing. In order to improve the speed of Wallace tree multiplier without degrading its area parameter, a new structure of Wallace tree multiplier is proposed in this paper. In the proposed structure, the final addition stage of partial products is performed by parallel prefix adders (PPAs). In this paper, two Wallace tree multiplier structures are proposed using Kogge stone adder and Sklansky adder. The proposed Wallace tree multiplier is synthesized and simulated using Xilinx ISE 14.7 software. The proposed structures are analyzed with respect to traditional multiplier design in terms of area (No. of LUTs) and delay(ns).

I. INTRODUCTION

In the process of evolution of technology, many Information Technology (IT) applications are requiring low-power integrated circuits (ICs), since handling huge amount of data to process. Due to integrated applications the size also increases. The energy efficiency is measured as the product of power consumption and computational time. To provide an energy efficient solution, the designer has to reduce the power consumption and the computing time as well. By reducing logic blocks, the static power can be reduced. However, reduction in power consumption has been achieved by different methods that demonstrates a trade-off relationship with circuit performance [1]. Low power consumption has become the decisive design goal in wide range of electronic systems and sub-systems. Inbuilt-powered sensing modules stand first in this row for the designers [2].
At present, the technology is advancing very rapidly in very short duration of time. The circuits being design have some billions of components with low area, high speed and low power consumption. Hence area, speed and power play crucial role in the design of any circuit [1], [2]. In order to satisfy the current trends, demand a circuit must be designed with low area and

less delay constraints. Arithmetic units are major blocks in any processing units which perform various arithmetic operations. Multiplication operation is important among all arithmetic operations. Several multiplication algorithms are studied in literature survey of multiplier designs like Binary multiplier, array multiplier, Booth's multiplier, Dadda multiplier, Wallace tree multiplier [4]. Wallace tree multiplier is advantageous in different types of multipliers [5].

The operation of Wallace tree multiplier is same in the first stage of multiplication which is generating partial products. In the second stage, Wallace tree multiplier adds first three rows partial products. Then the generated sum and carry are added with the next row of partial products. This addition process continues until the generation of final products. For this row-wise addition process half and full adders are employed. Thus, adders are playing a very important role in generation of final product terms. The speed of addition is going to affect the operation of the multiplication.

In order to improve the performance of multiplication operation, the adder structure used in design of Wallace tree multiplier has a major role. In this paper, a new structure of Wallace tree multiplier is proposed in which PPAs are used to add final row of partial products with the previous stage generated sum and carry out terms to generate final product terms. The PPAs are designs which are originally derived from carry look ahead adder concept of generating and propagating of carry bits.

In PPAs, a carry generation tree is present which generates carry for all preceding stages which improves the speed of operation. The carry generation tree mainly consists of two components-black cell and grey cell. The black cell and grey cell are interconnected to form carry tree network. Carry generation tree block is also called as parallel carry generation block as it generates carry bits for all stages at a time parallel [6]. There are different types of PPAs [7] whose classification mainly depends on two factors

I. Number of black and grey cells in carry generation tree

II. Interconnection of black and grey cells in carry generation tree.

II. LITERATURE SURVEY

Wallace Tree Multiplier Designs: A Performance Comparison Review by Himanshu Bansal, K. G. Sharma, Tripati Sharma Multiplication process is often used in digital signal processing systems, microprocessors designs, communication systems, and other application specific integrated circuits. Multipliers are complex units and play an important role in deciding the overall area, speed and power consumption of digital designs. This paper presents a comparison review of various Wallace tree multiplier designs in terms of parameters like latency, complexity and power consumption.

Design of 64-bit low power parallel prefix VLSI adder for high speed arithmetic circuits by Nehru, K., A. Shanmugam, and S. Vadivel.

The addition of two binary numbers is the basic and most often used arithmetic operation on microprocessors, digital signal processors and data processing application specific integrated circuits. Parallel prefix adder is a general technique for speeding up binary addition. This method implements logic functions which determine whether groups of bits will generate or propagate a carry. The proposed 64-bit adder is designed using four different types prefix cell operators, even-dot cells, odd-dot cells, even-semi-dot cells and odd-semi-dot cells; it offers robust adder solutions typically used for low power and high-performance design application needs. The comparison can be made with various input ranges of Parallel Prefix adders in terms power, number of transistors, number of nodes. Tanner EDA tool was used for simulating the parallel

DOI : https://doi.org/10.56452/6-1-26

prefix adder designs in the 250nm technologies.

III. WALLACE TREE MULTIPLIER

Wallace tree multiplier is the most extensively used multiplier design in many processors and memory units. In Wallace tree multipliers, multiplication process takes place in 3 steps.

Fig. 1 shows the block diagram of a Wallace multiplier. In this multiplier architecture, after generating the partial product, accumulation of partial product and final addition are done. The final addition is done, only when the final stage obtains only two rows.

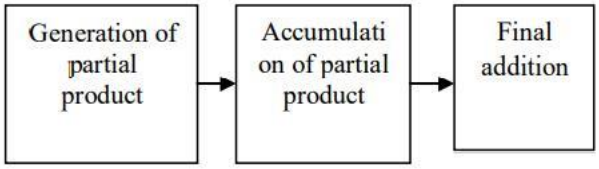

Fig. 1 Block diagram of a Wallace multiplier

3.1 Steps involved in Wallace tree multiplier Algorithm

I. Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding N results. Depending on position of the multiplied bits, the wires carry different weights. Product terms are generated by a collection of AND gates as shown in Fig. 2.

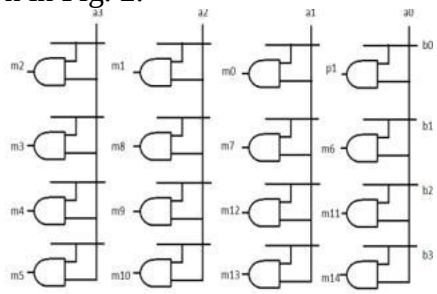

Fig. 2 Product terms generated by a collection of AND gates

II. Reduce the number of partial products to two rows using half adders and full adders. Full adders are applied to columns containing three bits and half adders to column containing two bits.

III. Last step is to add remaining two rows using a fast adder.

Fig 3 shows the algorithm of Wallace multiplier in matrix form. In this process of multiplication, addition process holds major role. In the existing methodology, final addition stage is performed using ripple carry adder in which carry propagation delay is more.

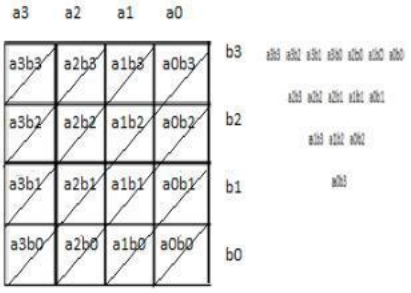

Fig. 3 4x4 WALLACE Algorithm

To perform fast addition, carry must be propagated quickly. This is the major drawback. To overcome this, Parallel prefix adders are used in final stage of addition.

IV. PROPOSED SYSTEM

In this section, we proposed a modification in the Wallace tree multiplier to reduce its carry propagation delay by using parallel prefix adders. In this paper the high speed Wallace tree multipliers are proposed using Kogge stone adder and Sklansky adder. Fig. 4 shows the block diagram of our proposed system.

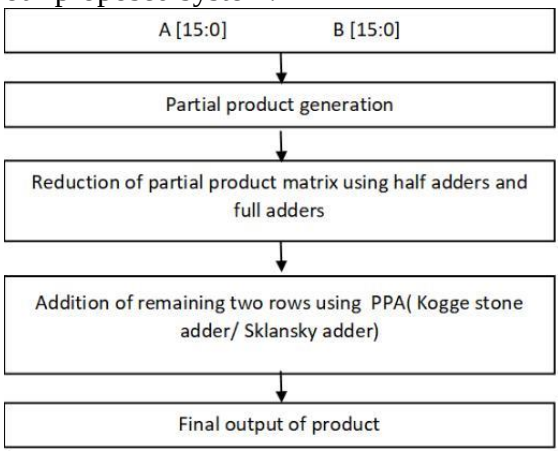

Fig. 4 Wallace tree multiplier using PPA

DOI : https://doi.org/10.56452/6-1-26

The multiplier operation in this approach is same which is performed in two phases as explained in section 3.1. In phase 1, partial products are generated with the help of AND gates. In Phase 2 partial products generated in phase 1 are added in step by step approach using half and full adders. But in the proposed design the final phase of addition of partial products in phase 2 is performed using PPA.

4.1 Parallel prefix adder

In PPA, the carry input for the next bits is generated at a time with the help of parallel prefix carry tree which consists of black cells and grey cells. There are many types of PPAs are present like Kogge stone adder, Sklansky adder, Brent kung adder, Ladner Fischer adder, Han Carlson adder. The PPA consists of three main blocks as shown in Fig. 5.

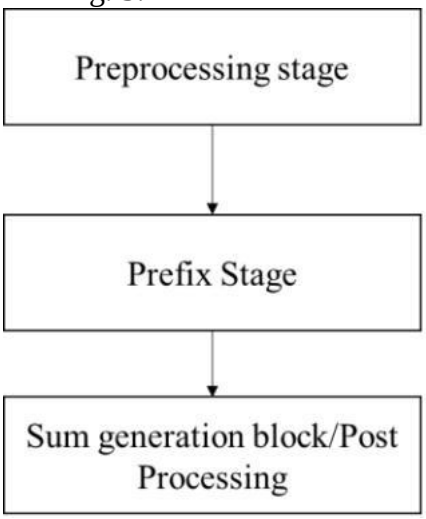

Fig. 5 Block diagram of PPA

I. The first stage is pre-processing stage, which generates Propagate(Pi) and Generate signals (Gi) using

Pi = Ai XOR Bi

Gi = Ai AND Bi

II. The second stage is the prefix stage. In prefix stage, generates next stage carry using propagate and generate signals from previous stage. Two operators namely black and grey cells are used in parallel prefix trees are shown in Fig. 6. The next stage carry generate and propagate signals are generated by black cells and grey cells. The black cell is used to calculate carry generate and carry propagate signals whereas grey cell calculates carry generate signals.

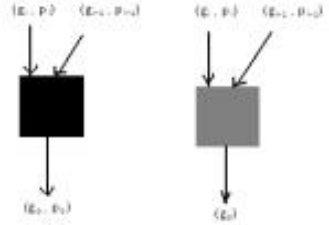

Fig. 6 Black cell and Grey cell notation

The black operator receives two sets of generate and propagate signals (Gi , Pi),(Gi-1 ,Pi-1), computes one set of generate and propagate signals (Gi+1 , Pi+1) by the following equations

P i+1 = Pi AND P i-1

G i+1 = Gi OR (G i-1 AND Pi)

The grey operator receives two sets of generate and propagate signals (Gi, Pi),(Gi-1 ,Pi-1), computes only one generate signal by following equation

G i+1 = Gi OR (G i-1AND Pi)

III. The third stage is sum generation block which gives the final summation or the output.

Si = Pi XOR G i-1

Cout = Gi

It is readily apparent that a key advantage of the tree-structured adder is that the critical path due to the carry delay is on the order of log 2N for an N-bit wide adder. The arrangement of the prefix network gives rise to various families of adders, for a discussion of the various carry-tree structures.

4.2 Kogge stone adder

In the proposed multiplier, the last two rows are added with Kogge stone adder (KSA) [7] which is also known as a fast adder. The Parallel Prefix expansion in Kogge stone adder is done in three stages, which is appeared in Fig. 5.

DOI : https://doi.org/10.56452/6-1-26

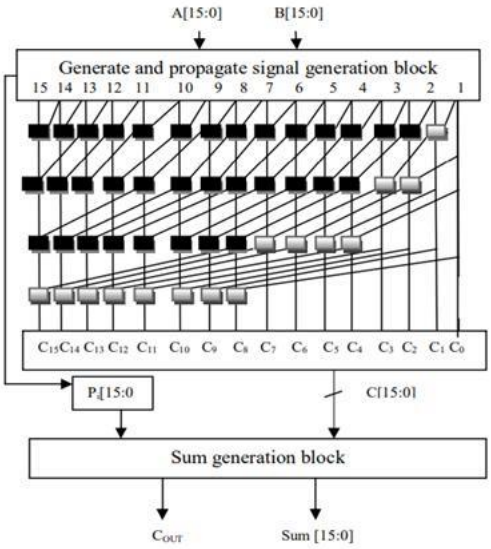

Fig. 7 16-bit Kogge stone adder structure

Fig. 7 shows 16 bit Kogge stone adder structure consists of 15 grey cells and 34 black cells. Thus, implementation of Kogge stone adder requires large area and circuit is complex with large number of interconnects. The output at each stage is two, thus Kogge stone adder has lower fanout of two.

4.3 Sklansky adder

The Sklansky adder builds recursively, considering 2-bit adder in first stage of prefix network then 4-bit adder, then 8-bit adder and so on, by abutting each time two smaller adders. Thus it has simple and regular architecture but suffers from fanout problem. This adder fanout increases hugely from input to output along the critical path leading for large amount of latency. Due to this adder performance degrades. Fig. 8 shows 16-bit Sklansky adder structure which requires 17 black cells and 15 grey cells. Thus, Sklansky structure occupies less area than the Kogge stone adder but fanout increases which increases latency.

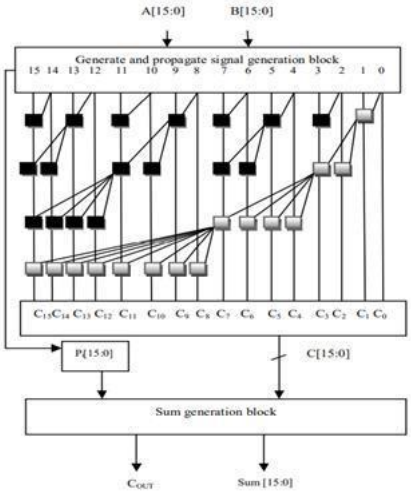

Fig. 8 16-bit Sklansky adder structure

V. RESULTS

Two wallace tree multiplier structures using Kogge stone adder and Sklansky adder are designed for input size of 8-bits in this paper using Verilog HDL. The designed mutipliers are simulated and synthesised using Xilinx ISE 14.7. The simulation waveforms of proposed multipliers ae shown in Fig. 9-Fig. 10.The proposed multiplier's synthesis result is compared with the traditional wallace tree multiplier in terms of area(No. of LUT's) and delay(ns) and tabulated in Table 1.

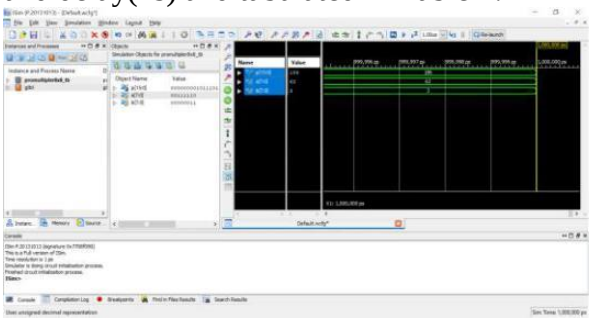

Fig. 9 Simulation waveform for 8-bit Wallace tree multiplier using Kogge stone adder

Total three multipliers i.e., proposed multipliers using two parallel prefix adders and one traditional Wallace tree multiplier are synthesized using XST synthesizer.

DOI : https://doi.org/10.56452/6-1-26

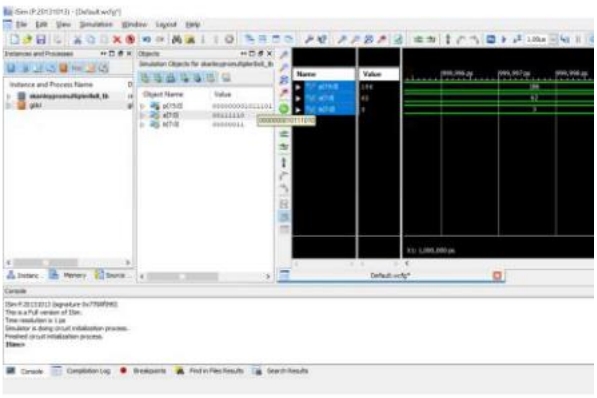

Fig. 10 Simulation waveform for 8-bit Wallace tree multiplier using Sklansky adder

Table 1 Area and delay of traditional and proposed Wallace tree multipliers

Input size	Multiplier structure	Area (No. of LUT's)	Delay(ns)
8 bit	Traditional Wallace tree multiplier	145	20.287
	Proposed multiplier using Kogge stone adder	195	19.485
	Proposed multiplier using Sklansky adder	173	17.903

The synthesis results are tabulated in Table 1. From the table it is clear that proposed multiplier using Kogge stone adder have less delay but occupies more area than traditional Wallace tree multiplier. The proposed multiplier using Sklansky has least delay and comparatively occupies less area than that of Kogge stone adder.

VI. CONCLUSION

Total three multiplier structures are designed in this paper in which two are proposed Wallace tree multipliers using two different PPA's and one is traditional Wallace tree multiplier. Wallce tree multiplier using Kogge stone adder and sklansky adder provides less delay than the traditional one. Thus, the delay parameter is achieved. In terms of area the difference of LUTs occupied by traditional and proposed structures is not more which intend that area parameter is also achieved up to some extent. So, it can be concluded that the proposed multipliers are better in delay and area when compared with the traditional multiplier structure.

Ladner Fischer adder is having least area, thus Wallace tree multipliers designed using Ladner fisher adder would occupy less area. Wallace tree multiplier using Ladner Fischer adder can be used for application circuits which require less area and speed of circuits will also be in medium, whereas for high speed application circuits, Wallace tree multiplier using Kogge stone adder can be used but costing little bit more area when compared to other proposed structures. The proposed multiplier structures can be used in high speed and low area application circuits basing on requirement. As future scope of this work, the multiplier structures can be designed by increasing input size (i.e. 32-bit,64- bit……). By designing the multiplier circuit with higher input size may lead to multiplier structure with less number of LUTs and also better delay value can be achieved.

BIBLIOGRAPHY

[1] N.Weste and D. Harris, CMOS VLSI Design. Reading, MA: Addison Wesley, 2004.

[2] N. Weste and K. Eshragian, Principles of CMOS VLSI Designs: A System Perspective, 2nd ed., Addison-Wesley, 1985-1993.

[3] Milos D. Ercegovac and Thomas Lang, "Digital arthimetic," Morgan Kaufmann, Elsevier INC, 2004

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

[5] Himanshu Bansal, K. G. Sharma, Tripti Sharma," Wallace Tree Multiplier Designs: A Performance

Comparison Review“,Innovative Systems Design and Engineering, Vol.5, No.5, 2014.

[6] Nehru, K., A. Shanmugam, and S. Vadivel. "Design of 64-bit low power parallel prefix VLSI adder for high speed arithmetic circuits." In 2012 International Conference on Computing, Communication and Applications, pp. 1-4. IEEE, 2012.

[7] Rakesh.S,K.S.Vijula Grace, “A comprehensive review on the VLSI design performance of different Parallel Prefix Adders” ScienceDirect, Materials Today: Proceedings 11 (2019) 1001–1009

[8] Kesava, R. Bala Sai, B. Lingeswara Rao, K. Bala Sindhuri, and N. Udaya Kumar. "Low power and area efficient Wallace tree multiplier using carry select adder with binary to excess-1 converter." In 2016 Conference on Advances in Signal Processing (CASP), pp. 248-253. IEEE, 2016.

[9] Sunil M, Ankith R D, Manjunatha G D and Premananda B S, “Design and Implementation of Faster Parallel Prefix Kogge-Stoneadder”,Int.Journal of Electrical and Electronic Engineering & Telecommunications, Vol.4, No.1, pp 116-121, 2015.

[10] Raghumanohar Adusumilli and Vinod Kumar K, “Design and Implementation of a high speed 64 bit Kogge-Stone adder using Verilog HDL”, Int. Journal of Electrical and Electronic Engineering & Telecommunications, Vol.3, No.1, pp 13 – 18, 2014.

[11] Nurdiani Zamhari, Peter Voon, Kuryati Kipli, Kho Lee Chin and Maimun Huja Husin, “Comparison of Parallel Prefix Adder (PPA)”, Proc.World Congress on Engineering 2012 (WCE 2012) Vol II, London, July 2012.

[12] David H. K. Hoe, Chris Martinez and Sri Jyothsna Vundavalli, “Design and Characterization of Parallel Prefix Adders using FPGAs”,IEEE 43rd Southeastern Symposium on System Theory (SSST), USA, pp 168 – 172, March 2011.

13] Jasmine Saini, Somya Agarwal and Aditi Kansal, “Performance, Analysis and Comparison of Digital Adders”, Proc. IEEE Int. Conference on Advances in Computer Engineering and Applications (ICACEA), Ghaziabad, India, pp 80 – 84, 2015.

[14] Sudheer Kumar Yezerla and B Rajendra Naik, “Design and Estimation of delay, power and area for Parallel prefix adders”, Proc. IEEE 2014 RAECS UIET Panjab University, Chandigarh, March 2014.

[15] R.Brent and H.Kung, “A regular layout for parallel adders”, IEEE Transaction on Computers, vol. C-31, no.3, pp 260 – 264, March,1982.

[16] K.Babulu,Y.Gowthami,”Implementation and Performance Evaluation of Prefix Adders using FPGAs”, IOSRJournal of VLSI and Signal Processing, Vol. 1, Iss.1, pp 51 - 57, 2012.

[17] S. Sri Katyayani, Dr.M.Chandra mohan Reddy and Murali.K, “Design of Efficient Han-Carlson-Adder”, International Journal of Innovations in Engineering and Technology, Special issue on ETiCE, pp 69– 75,2016.