IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

LOW POWER N*N MULTIPLIER USING VARIOUS ALGORITHMS

Ankam Tharun Kumar Reddy, Lovely Professional University,

Siddavatam Naveen Sai, Lovely Professional University,

Boyalla Sudhakara Rao, Lovely Professional University,

Kare Rakesh, Lovely Professional University

Abstract:-

As a result of the growing number of computer and signal processing applications, there has been an increase in demand for high-speed processing. In multiplier design, low power consumption is also a major consideration. A good multiplier should be able to provide a physically compact, high-speed device with low power consumption. The multipliers used in digital systems decide the power consumption and speed of such systems. In order to increase the efficiency of any processor, it must utilize an efficient multiplication algorithm. The main objective here is to present a model of a n*n bit optimized multiplier architecture with low power and high speed using different algorithms and optimized full adders with low power dissipation and minimum propagation delay. The algorithms in this proposed architecture multiplier are Vedic multiplier, Wallace, and Dadda. The algorithms work by reducing the number of steps in the process that include partial products. The partial products are added to get the algorithm's output. The number of partial products determines the speed.

1. INTRODUCTION

The most important thing in processors and many digital applications such as with signal processing and digital filters is multiplication. Multiplication is a mathematical procedure that is essentially an abbreviated version of adding an integer to itself a certain number of times. A binary multiplier is a digital electronics circuit that multiplies two binary numbers. Since some Digital Signal Processing algorithms spent more time multiplying than others, designers sacrificed a substantial portion of chip area to make multiplication as fast as possible. Because of large-scale integration, it is now possible to fit more transistors per chip on a single chip, allowing for more address. The most widely known method entails computing a series of partial products and adding them together. Bit multiplication is accomplished with logic gates, and partial product summation is accomplished with a half adder or a full adder. Various algorithms have existed since the decimal system's inception, depending on the size of the numbers. Some algorithms are proposed here among existing algorithms. The main focus of this paper is on the Vedic multiplier, Dadda and Wallace algorithms which have the shortest delay, the smallest area, the highest generating speed, and the lowest power consumption. The Urdhva Triyagbhyam Sutra, which was generally used in ancient India for the multiplication of two decimal numbers in a short amount of time, inspired the Vedic multiplier architecture. The architecture of the Dadda and Wallace multiplier using compression technique consists of

- Half adders, full adders, and compressors are used to minimise partial product.

- For the summation of reduced partial products, use a propagation adder.

2. LITERATURE SURVEY

The significance of parallel multipliers is discussed in paper[1] in comparison to serial adders. It is evident from a literature review that the slowest path causes the most delay.

To minimise latency, the carry save addition algorithm is used in paper[2] performed in Xilinx simulation tool.

In paper[3] primarily focused multiplier is Vedic multiplier, which has advantages such as high speed and less area than array multipliers.

8-bit Vedic multiplier is implemented using DIF-FFT application of booth , array and Wallace multipliers[10]

3. EXISTING ALGORITHMS

3.1 *Karatsuba algorithm*

The Karatsuba algorithm is a fast multiplication algorithm that multiplies two numbers using a divide-and-conquer strategy. Multiplication of two n-digit numbers is reduced from O(N^2) to O(N^(log 3 base 2), which equals O(N^1.585), using this algorithm. The algorithm generally requires O(n) additions and subtractions. As a result, it is asymptotically faster than the standard algorithm, which usually requires the use of the n^2 single-digit product.

3.2 *Array Multiplier*

Because of its normal form, the array multiplier is well-known. The add and shift algorithm is used in the multiplier circuit. The multiplicand is multiplied by one multiplier bit to produce each partial product. The partial products are then added and shifted then according their bit orders. The addition is carried out both serially and in parallel. A standard carry propagate adder can be used to perform the addition. There are N-1 adders required, where N is the multiplier length.

3.3 *Booth Algorithm*

By considering two bits of the multiplier at a time, the booth multiplier reduces the number of partial products, giving it a speed advantage over other multiplier architectures. This method works for both signed and unsigned numbers. This approach has the advantage of making the partial product circuit simpler and easier to execute.

3.4 *Parallel Hierarchical Overlay Multiplier*

For high-speed DSP operations, a parallel Hierarchical Overlay Multiplier architecture is used. The architecture is based on the ancient Indian Vedic Mathematics vertical and crosswise algorithm. Both the multiplicand and the multiplier are grouped four bits at a time in the proposed architecture. As a result, the entire multiplication operation is broken down into four 4x4 bit multiplication modules. Any multiplier, such as array, booth, Wallace, or a potential proposed effective multiplier, can be used to implement the 4x4 multiplication modules.

4. PROPOSED ALGORITHMS

An 8*8 bit multiplier is used as an example for all proposed algorithms.

4.1 *Vedic Multiplier*

In the sections below, the hardware architecture of the 2X2, 4x4, and 8x8 bit Vedic multiplier modules is shown. The Urdhva-Triyagbhyam sutra is used here to propose such an architecture for multiplying two binary numbers. The uniqueness of the Vedic multiplier is that it produces and adds partial products at the same time. As a result, it's well-suited to parallel processing. It's more appealing for binary multiplications because of this function.

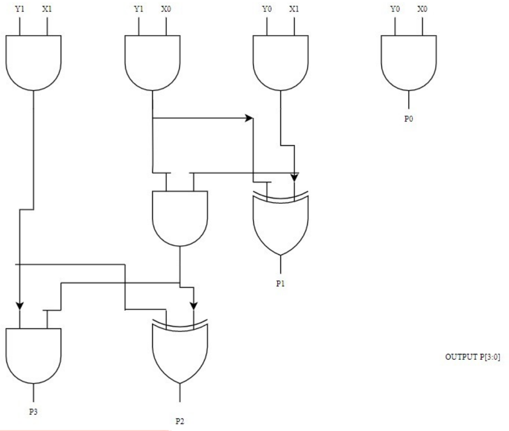

fig. 1. 2*2 bit Vedic Multiplier

Logic gates are used to create a 4-bit output (P[3:0]) from a 2*2 bit vedic multiplier.

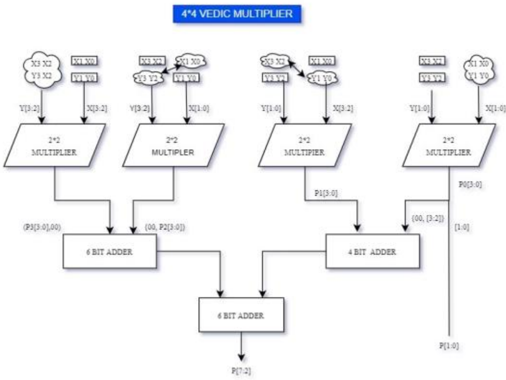

fig. 2. 4*4 bit Vedic Multiplier

We can create 4x4 bit multipliers using four 2x2 multipliers and three adders, as shown in the figure(2). Thus, 8-bit output is formed.

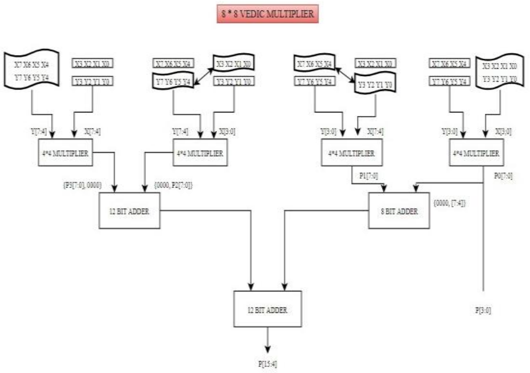

fig. 3. 8*8 bit Vedic Multiplier

We can create 8x8 bit multipliers using four 4x4 multipliers and three adders, as shown in the figure(3). Thus, 16-bit output is formed.

In the same way, for n*n bit vedic multiplier, (n/2)*(n/2) bit vedic multiplier and three adders are used. N-bit adder modules will be useful to add any number of bits.

4.2 *Wallace Multiplier*

The Wallace algorithm performs the addition operations in parallel, which reduces the amount of time it takes. Wallace tree is a hardware implementation of a digital circuit that multiplies two integers that is fast and efficient. At the end, partial products are reduced to a two-row matrix, which is added with the propagate adder, and then the final output is generated.

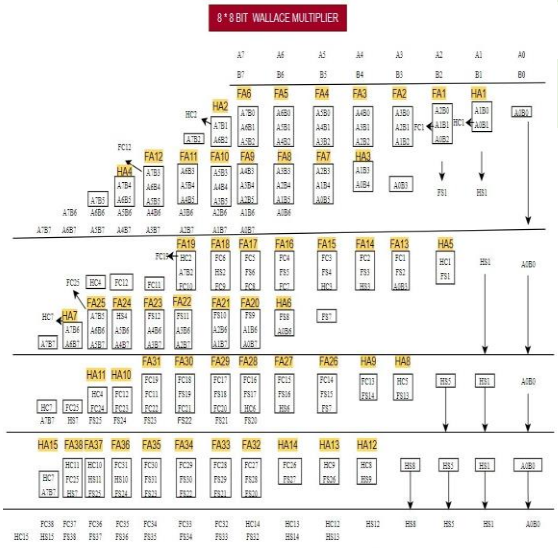

fig. 4. 8*8 bit Wallace Multiplier

In the figure(4), eight rows are reduced to six rows in 1st stage, six rows to four rows in 2nd stage, four rows to three rows in 3rd stage, three rows to two rows in 4th stage of all partial products. At the final stage, two rows are added with the ripple carry adder and final output is formed.

4.3 *Dadda Algorithm*

The Dadda multiplier is similar to the Wallace tree multiplier in that it takes up less area than the Wallace multiplier.

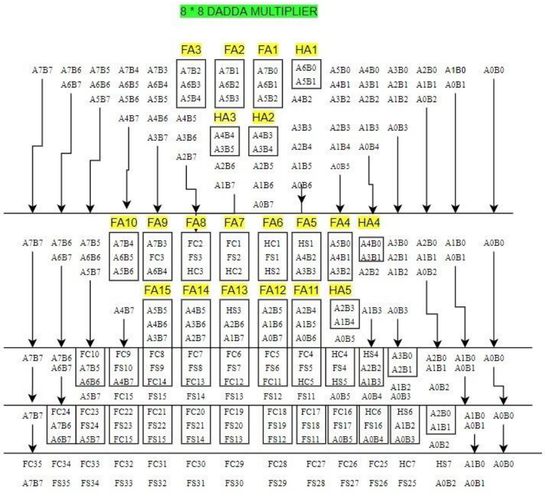

fig. 5. 8*8 bit Dadda Multiplier

Dadda multiplier reduces the number of rows in the same way as wallace multiplier reduces the number of stages in the wallace tree. The dadda multiplier outperforms the Wallace multiplier.

5. RESULTS AND ANALYSIS

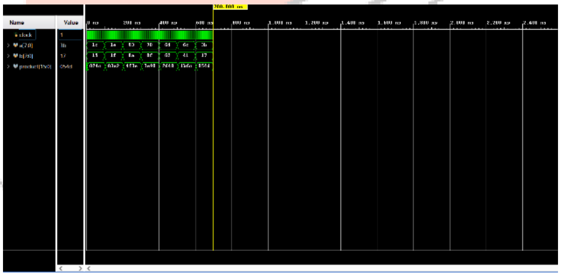

fig. 6. Simulation of multiplication result

The simulation results of the multiplier that multiplies two binary numbers and gives the exact result of multiplication by using all the proposed algorithm multipliers where ‘a’ and ‘b’ represents two input operands and ‘product’ represents the output. The simulation can be performed using Vivado 2019.1 or Xilinx ISE 9.2i.

6. CONCLUSION

As compared to the wallace tree multiplier and even the vedic multiplier, the computational delay for 8*8 bit binary multiplication for the dadda algorithm is less. Power consumption for vedic multiplier is high when compared to Wallace tree multiplier and very high when compared to dadda algorithm.

7. FUTURE SCOPE

The techniques described in this paper are an attempt to establish arithmetic algorithm and architecture level optimization techniques for low power high-speed multiplier design. However, there are some drawbacks, and there are some potential research paths that could be pursued. To improve efficiency, higher order compressors such as 7:2, 9:2 compressors may be used to accumulate partial products before obtaining the final product from the propagate adder.

8. REFERENCES

[1] Kaushik, Vikas, and Himanshi Saini. "Comparative Analysis of Proposed Parallel Digital Multiplier with Dadda and Other Popular Multipliers." *International Journal* 4 (2017): 237-240.

[2] Sinthura, Siva S., et al. "Implemenation and Analysis of Different 32-Bit Multipliers on Aspects of Power, Speed and Area." *2018 2nd international conference on trends in electronics and informatics (ICOEI)*. IEEE, 2018.

[3] Bandi, Vijaya Lakshmi, Prathima Gamini, and Balla Sai Harshith. "Performance Analysis of Dadda Multiplier Using Modified Full Adder." *International journal of innovative research in computer and communication engineering* 6.2.

[4] Sangeetha, P., and Aijaz Ali Khan. "Comparison of Braun multiplier and Wallace multiplier techniques in VLSI." *2018 4th International Conference on Devices, Circuits and Systems (ICDCS)*. IEEE, 2018.

[5] Ram, G. Challa, et al. "Design of delay efficient modified 16 bit Wallace multiplier." *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, 2016.

[6] Hussain, Inamul, Chandan Kumar Pandey, and Saurabh Chaudhury. "Design and Analysis of High Performance Multiplier Circuit." *2019 Devices for Integrated Circuit (DevIC)*. IEEE, 2019.

[7] Kaushik, Vikas, and Himanshi Saini. "A Review on Comparative Performance Analysis of Different Digital Multipliers." *Advances in Computational Sciences and Technology* 10.5 (2017): 1257-1272.

[8] K. A. K. Maurya, Y. R. Lakshmanna, K. B. Sindhuri and N. U. Kumar, "Design and implementation of 32-bit adders using various full adders," *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Vellore, India, 2017, pp. 1-6, doi: 10.1109/IPACT.2017.8245176.

[9] G.ChallaRam,, D.Sudha Rani, Y.RamaLakshmanna, K.Bala Sindhuri ―Area Efficient Modified Vedic Multiplier‖ 2016 International Conference On Circuit, PowerAnd Computing Technologies ,pp. 1-5,March 2016

[10] Rohith, S., Kasetty Ram Babu, and M. N. Chandrashekar. "FPGA Implementation of 8-Bit Vedic Multiplier for DIT-FFT Application Using Urdhva Tiryagbhyam Sutra."