International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 19 Issue 09, SEP, 2019

# IMPLEMENTING THE DADDA MULTIPLIER ON AN FPGA: A COMPREHENSIVE ANALYSIS

[#1]**MUTTNOORI NAVEEN,** ***Asst. Professor,***

[#2]**KUKKA RAJ KUMAR,** ***Asst. Professor,***

*Department of Electronics Communication Engineering,*

*SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES, KARIMNAGAR, TS.*

**Abstract:** The final result of the mathematical operation known as multiplication is formed by performing a number of alternative mathematical operations on the operands provided, leading to the decision that multiplication is the correct mathematical operation to apply. Natural multiplication involves the formation of partial products in a sequential manner, which are subsequently combined to give the final result. Increasing the bit-width, on the other hand, will necessitate the deployment of additional resources, including both time and energy. These additional resources will be required. Based on the information supplied above, we have decided to use a Dadda Multiplier, which will boost the system's processing speed as well as its power efficiency. This positive conclusion is observed even when the Dadda Multiplier is implemented on a Field-Programmable Gate Array (FPGA). The arithmetic method as described can be successfully employed in a wide range of error-tolerant applications, including but not limited to image processing, digital processing, and FFT, among others.

## 1. INTRODUCTION

Despite the fact that mathematical operations are simple in their construction, they can be found in a wide range of circumstances. One of the methods contained in this set is the operation of multiplication. When the goal is to attain high data rates based on a system's operational speed, it is definitely required to consider the function of multiplication as a critical feature.

Because of the rapid advancement of technology in recent years, it has become important to upgrade electronic components in terms of the amount of power they offer, the amount of space they take up, the rate at which they process data, and any other relevant qualities. When multipliers are compared to more traditional techniques of conducting multiplication, it is evident that a more time and labor efficient strategy is necessary. A number of more complex techniques to executing standard multiplication have been invented in order to improve its effectiveness. Examples include the Array Multiplier, Wallace Multiplier, Dadda Multiplier, and Vedic Multiplier.

In conventional multiplication procedures, the act of generating partial products and then adding them together can be completed in two stages. However, in order to employ the Dadda Multiplier, three different processes must be completed. The process can be divided into three major steps: i) the production of partial products, ii) the lowering of the tree's height (which also results in the production of partial products), and iii) the summing of the obtained partial products. The first stage, partial product development, is the most crucial of the three. The goal of this inquiry is to look into the capabilities of the Dadda Multiplier as well as the techniques for verifying its operation on a Field-Programmable Gate Array (FPGA).

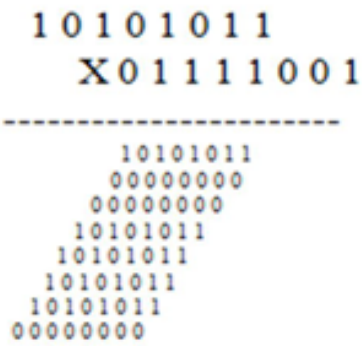
## 2. DADDA MULTIPLIER

The Dadda Multiplier is a type of binary multiplier that performs the process of multiplying two binary operands and then returns the product of that multiplication. The use of Half and Full Adders will be required to complete the process of lowering the tree's height, as indicated by its hierarchical structure. The proportion deducted from the total cost of the purchase will vary based on the height of the tree. It is advised that the minimum height be set to 2, and the maximum height be defined using the bit-width parameter. The difference in size between each succeeding step should not be higher than a factor of 1.5. In the case of an 88-bit multiplication operation, for example, the greatest degree of importance that

will be considered is 6, and it will then steadily decline until it reaches the level of 2.

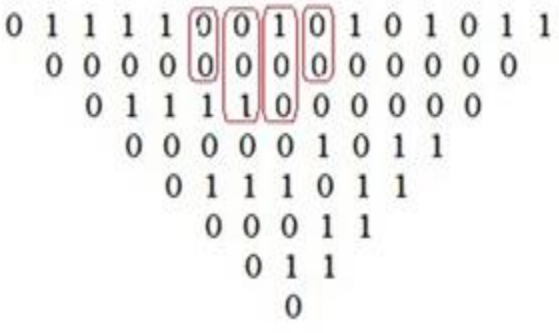
This study will look at the findings of Dadda trees with dimensions of eight bits by eight bits, sixteen bits by sixteen bits, and thirty-two bits by thirty-two bits. Please see the example of an 8 bit by 8 bit Dadda Multiplier that has been provided here. Consider the operands A and B, which are represented by the notations (10101011)2 and (01111001)2, respectively. With an 8-by-8 bit, you can obtain a maximum reduction height of six..

## Stage-1

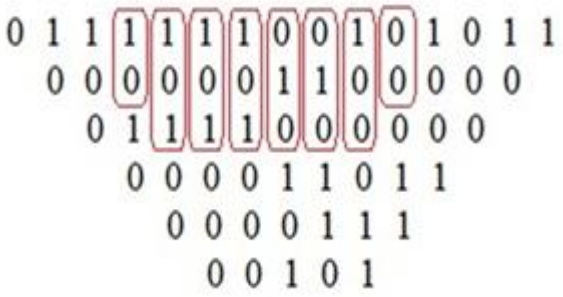



## Stage-2

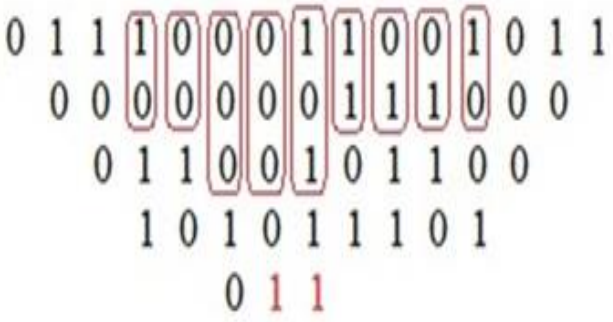
Step-1: At the moment, the height of a column can only be expanded to a maximum of six units, and any height that surpasses this limit is immediately reduced to six units. As a result, the following will occur:



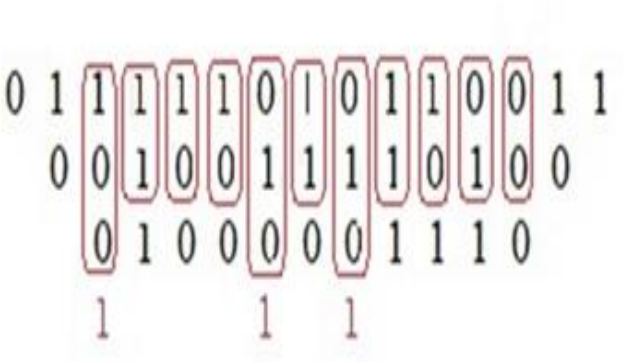
Step-2: At the moment, the column can only reach a maximum height of four units. Furthermore, the column's height will be reduced only if it reaches or passes the criteria for this threshold. As a result, the following will occur:



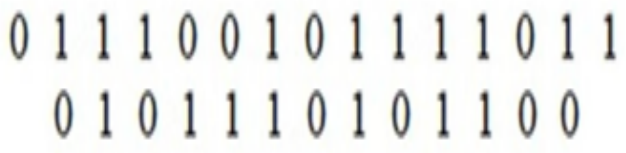
Step-3: At the moment, the column can reach a maximum height of 3 units. Furthermore, the column's height will only be reduced if it is greater than the value of 3. As a result, the following will occur:



Step-4: At the moment, the maximum height of a column is limited to merely two units. Only if a column's size increases by more than two units will its height be reduced. As a result, the following will occur:



Step-5: As a result of this aspect, the end result will be fragmented, resulting in the development of a fragmented final result.



Stage-3

At this stage, the fragmentation process outputs will be aggregated, yielding the binary

representation (0101000011010011)2.

## 3. RESULTS

In this study, the Dadda Multiplier was compared to various types of multipliers, with a special emphasis on the Dadda Multiplier's efficiency in terms of power usage and space use. The evaluation was carried out in comparison to other multipliers, as well as more precisely on Field-Programmable Gate Arrays (FPGAs). Furthermore, we conducted a comparative research to see how effective it was in comparison to the performance of other multipliers. The figures below show the results of simulations with bit widths of 8*8, 16*16, and 32*32. The simulation results are depicted in these figures:

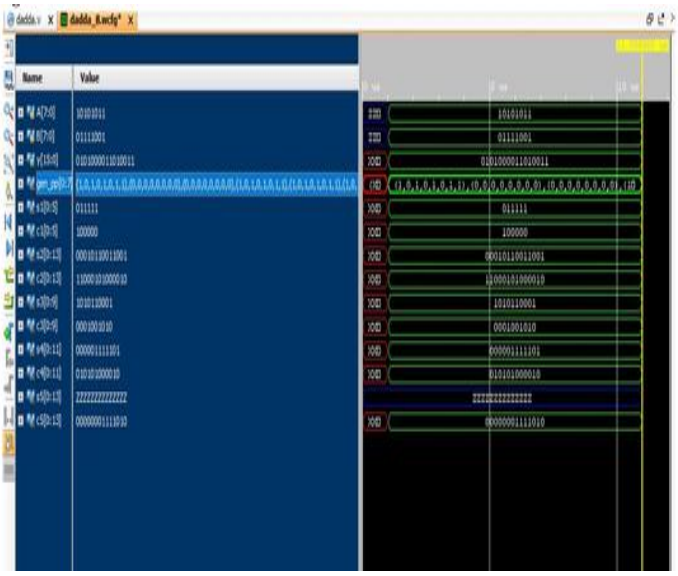


Fig-1: The results in this article represent the outcome of running the simulation for the 8*8 Dadda Multiplier

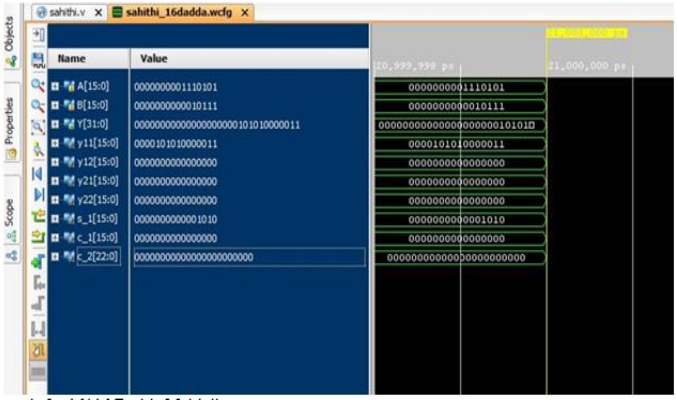


Fig-2: The product achieved by conducting the simulation for the 16*16 Dadda Multiplier is the result stated in this article.

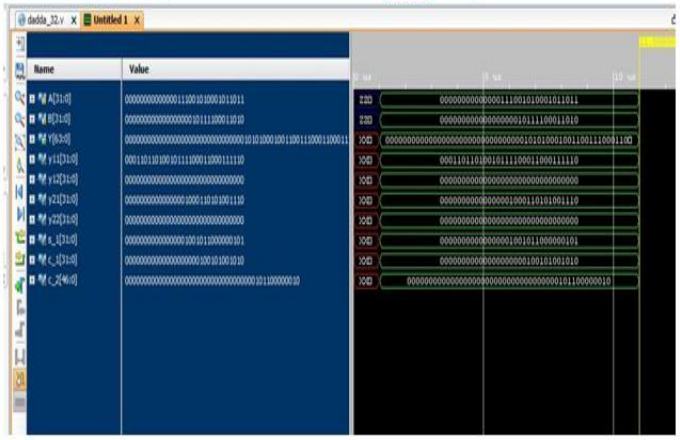


Fig-1: The findings of a simulation of the 32*32 Dadda Multiplier indicate to the conclusion that can be reached.

The RTL design of the 8x8 Dadda Multiplier is described in the following format:

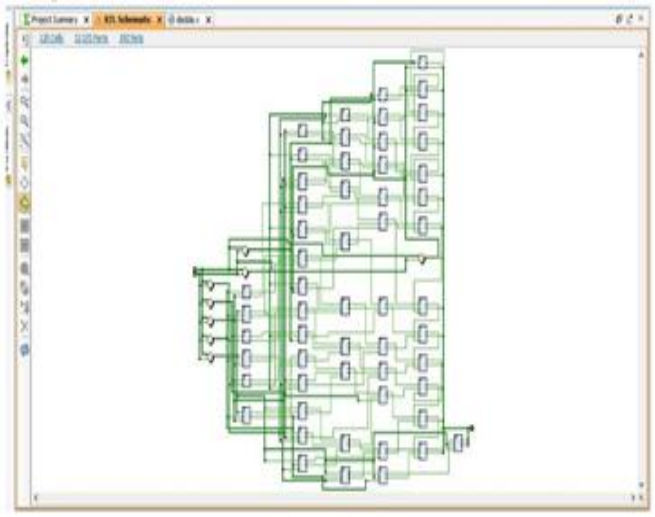


Fig-4: The RTL (Register Transfer Level) design of the 8x8 Dadda Multiplier is being debated.

Following that is a table with a graphical representation of how effective the Dadda multiplier is at various bit settings.

| Parameters | 8*8 bit | 16*16 bit | 32*32 bit |
|---|---|---|---|
| Number of LUT's used | 675 | 865 | 942 |
| Number of occupied slices | 1034 | 1124 | 1236 |
| Power Consumption | 0.24 w | 1.23w | 1.867w |
| Delay | 5.4 ns | 6.96 ns | 7.87 ns |

## 4. CONCLUSION

The goal of this research is to look into the efficiency of the Dadda Multiplier with different bit sizes. After examining the similarities and differences between the results obtained by the Dadda Multiplier, the Array Multiplier [3,] and other multipliers, it was revealed that the Dadda

Multiplier is more effective than the previously mentioned multipliers. This was determined by comparing the Dadda Multiplier findings to those of the other multipliers. The dadda multiplier has numerous potential applications, including but not limited to image processing, digital processing, and a variety of other industries.

## REFERENCES

[1] Muteen Munawar and Talha Khan "Low power and High Speed Dadda Multiplier using Carry Select Adder with Binary to Excess-1 Converter" ISBN: 978-1- 7281-7113-5, March 2020.

[2] B. Jeevan and S. Narender "A High-Speed Binary Floating-Point Multiplier Using Dadda Algorithm" ISBN- 978-1-4673-5090-7, March 2013.

[3] G. Ramachandran and Dr Chandra Kumar Dixit "Performance Analysis of Mantissa Multiplier and Dadda Tree Multiplier and Implementing with DSP Architecture" ISBN: 978-1-7281-9537-7, November 2022.

[4] Hassan Afzali-Kusha and Marzieh Vaeztourshizi "Design Exploration of Energy-Efficient Accuracy-Configurable Dadda Multipliers with Improved Lifetime Based on Voltage Overscaling" ISSN: 1557-9999, March 2020.

[5] Ayush Kumar and Ankit Kumar "Design and Study of dadda Multiplier by using 4:2 Compressors and Parallel Prefix Adders for VLSI Circuit Designs" ISBN: 978-1-7281-7029-9, May 2021