

## **DESIGN AND SIMULATION OF HIGH-SPEED BINARY MULTIPLICATION: GROUPING AND DECOMPOSITION MULTIPLIER**

**M Sagar**, Assistant Professor Department Of ECE Sree Chaitanya College of Engineering,  
Karimnagar

**Dr V Krishna Naik** Assistant Professor Department Of ECE Sree Chaitanya College of  
Engineering, Karimnagar

**ABSTRACT:** For applications involving the Fast Fourier transform (FFT) and digital signal processing (DSP), binary multipliers are crucial parts of computing systems. Multipliers are significant mathematical operations that need additional processing power and physical resources. Consequently, a great deal of research has been done to shorten processing times and hardware requirements. In this study report, a high-speed binary multiplier called the Grouping and Decomposition (GD) multiplier is suggested in order to reduce processing time. Enhancing algorithm processing efficiency over current multiplier architectures is the main objective of the suggested multiplier. Two strategies are used to achieve this goal: first, partial products of the same size are grouped in parallel, and second, each partial-product bit in the grouped sets is broken down. For summing, a 5:2 logic adder, sometimes referred to as a 5LA, is utilized. Multiplication operations become more efficient when parallel processing and decomposition logic are used to cut down on the number of computational steps. Using Cadence® Virtuoso and Cadence® Virtuoso Assura tools, the front-end and physical design implementation of the proposed GD multiplier was completed in the 180 nm technology library. The front-end design of the 8 8 proposed GD multiplier showed a significant 56% and 53% reduction in computation time and power-delay product, respectively, in comparison to existing multiplier architectures. The physical design implementation further reduces the power-delay product of the recommended multiplier by employing the shortest-path technique for internal subsystem routing. The suggested multiplier's efficacy rises with increasing complexity of multiplication tasks, which makes it perfect for sophisticated applications.

### **Keywords:**

5:2 logic adder, grouping and decomposition multiplier, quick Fourier transform, digital signal processing.

### **1. INTRODUCTION**

Incorporating multimedia, image processing, and the Internet of Things necessitates enormous computational resources, as well as the vital need for quick responsiveness and energy efficiency. Digital logic circuits serve an important part in a variety of computer arithmetic applications, exhibiting a high level of reliability and accuracy. The multiplier is a fundamental mathematical element that is essential in many applications, particularly signal processing. There are numerous rapid multipliers available, each with its own set of perks and downsides. The current focus in academia is on improving performance measures, including power efficiency, area utilization, and processing speed. For multipliers, there are two basic design options: sequential and parallel architectures. Sequential designs use less electricity, but with a longer delay. The Wallace tree and Dadda architectures, on the other hand, are parallel systems with substantial power usage.

The optimization of power consumption and processing speed is critical in the design of digital circuits, especially when multipliers are involved. A frequent method is to optimize a single parameter while taking into account a limitation placed on another parameter. The current project provides a tough challenge due to the limited power capacity of portable devices, necessitating careful planning to

achieve optimal efficiency. The presence of a specific amount of reliability may impede the system's intended performance level. There are multiple methods available at various levels of design abstraction that can be used to meet the power and speed requirements. The array multiplier is a typical type of multiplier that performs multiplication by shifting and adding. However, because of its increased processing demands, it consumes more energy, takes up more physical space, and takes longer to complete computing tasks. The main focus of current research is on the Vedic multiplier, which has significant advantages over array multipliers. Some of the advantages include increased operating efficiency and reduced geographical needs. The topic of inexact computing seeks to improve computer operations by prioritizing objective aspects over computational precision. The fundamental concept underlying the process of inexact arithmetic calculation is the reduction of circuit complexity inside arithmetic units. The aforementioned methodology can be used in situations when a definitive solution is not possible and/or a collection of approximate outcomes is acceptable.

## 2. REVIEW OF LITERATURE

Sebastian, Alen, et al. present a 16-bit Dadda multiplier using novel compressor designs. The new compressors have two 4–2 and an enhanced higher-order compressor. Three potential multiplier designs were compared to existing ones. The proposed design matches the ideal state better than earlier designs based on latency and area analyses. It can also be utilized for accurate multipliers. The method above provides 84 slices and 20.612 nanoseconds of latency. The power loss was disregarded.

To reduce multiplier delay time and maintain area efficiency, Devi Ykuntam et al. proposed a Wallace tree multiplier architecture. In the configured setup, parallel prefix adders efficiently add partial products (PPAs). The work introduced Kogge–Stone, Sklansky, Brent–Kung, Ladner–Fischer, and Han–Carlson Wallace tree multiplier architectures. The proposed multiplier designs are compared to the typical one for nanosecond latency and Look-Up Tables. We studied a 16-bit Wallace tree multiplier utilizing the Kogge–Stone adder. The design had 634 Look-Up Tables and 29.44 nanoseconds.

In high-computation applications, approximation methods reduce device use, battery consumption, and delay. The superfluous proportion and data storage will shrink. Approximation approaches lower arithmetic circuit computational complexity while maintaining coding efficiency. The approximate, practically complete adder-based Dadda multiplier enhances speed, energy usage, and device count. This study evaluates many multipliers using a nearly full adder approximation. An almost-full adder 8-bit Dadda multiplier requires 11.409 W and 0.20 LUTs.

Jaiswal, Kokila Bharti, et al. created a multiplexer-based complete adder to save multiplier power. The Wallace tree multiplier design evaluated the structure's efficiency. The proposed multiplier reduced power consumption by 37.45%, area utilization by 45.75%, and latency by 17.65% compared to a typical topology using ASIC synthesis. The recommended complete adder builds a 16-bit Wallace multiplier in 8.81 nanoseconds and consumes 6.5534 milliwatts.

Wallace and Dadda multiplied using Devnath's hybrid 3–2 counter. The recommended method generates partial products utilizing numerous AND gates with two transistors. The 65 nm PTM transistor architecture made multipliers. A comprehensive assessment and analysis compared both multipliers' system outcomes to other models. The hybrid full adder in the 4-bit Dadda multiplier had a latency of 220.9 ps and used 20.34  $\mu$ W of power.

Ram et al. compared Wallace multiplier delay to array and Dadda multipliers. The suggested 16-bit Wallace multiplier employed CSLA and BEC adders. The Wallace multiplier computes faster with a CSLA than a BEC. Wallace multiplier with 16-bit input converts binary to excess code in 24.948 nanoseconds and 86.48 mW.

Two multiplier 4–2 compressor estimates were established by Momeni et al. Different compression algorithms accommodate computational faults in circuit-based performance measurements in these systems. Researchers examined four Dadda multiplier approximation compressor integration approaches. A 4–2 compressor approximation 8-bit Dadda multiplier has 44.35 picoseconds latency and 1.14 micro-Watts power consumption.

Four 4:2 compressors with accurate and approximation modes were introduced by Akbari, Omid, et al. The approximation mode uses 2D compressors to boost speed and power but reduce accuracy. The compressors' latency, power consumption, and accuracy vary between exact and approximate modes.

16-bit Dadda Multiplier with dual-quality 4:2 mixed compressors has 1.19 nanosecond delay and 2339 microWatts power consumption.

Wallace/Dadda/parallel binary multipliers' performance specifications are in Table 1. All architectures increase power-delay product by slowing processing or consuming more power. This study proposed a new binary multiplication design to lower power-delay product and other crucial factors. Decomposing partial goods by size and parallel decomposing each category partial product helped complete this homework. Parallel processing and decomposition logic decrease computing stages and speed up calculation in the revolutionary architecture. The GD multiplier decomposes using a 4x4 Wallace and Dadda binary multiplier.

Table 1. The parameters of numerous literature works will be compared in this study.

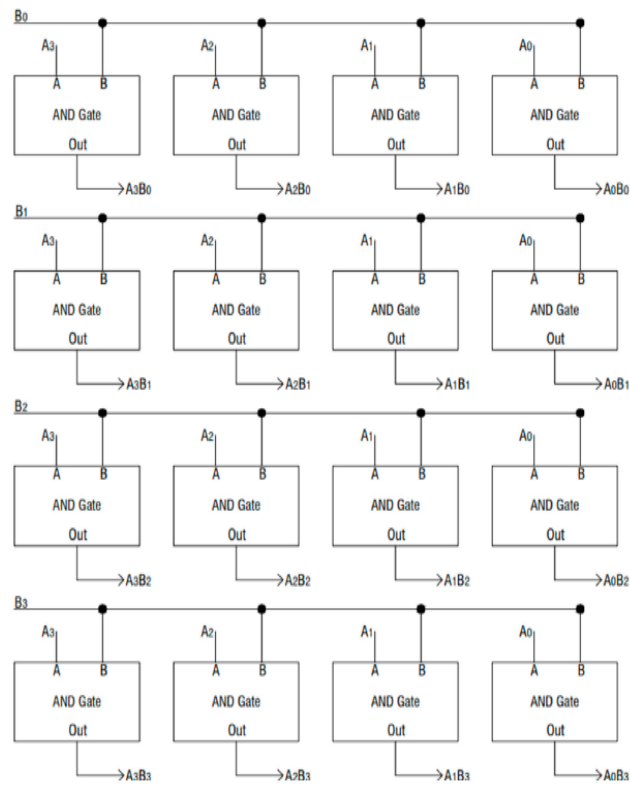
	Title	Multiplier	Computational Time	Power Consumption/ Area Consumption	Implementation Tool/ Technological Node
1.	Design and Implementation of an Efficient Dadda Multiplier Using Novel Compressors and Fast Adder	16-bit Dadda multiplier with 4-2	20.612 ns	84 slices	Xilinx ISE
2.	Design and Analysis of High speed Wallace Tree Multiplier Using Parallel Prefix Adders for VLSI Circuit Designs	16-bit Wallace tree multiplier using Kogge-Stone adder compressors	29.44 ns	634 LUT	Xilinx ISE
3.	An Efficient Dadda Multiplier using Approximate Adder	8-bit Dadda multiplier using almost full adder	NA	11.409 Watt/ 0.20 LUT	Xilinx ISE
4.	Low-Power Wallace Tree Multiplier Using Modified Full Adder	16-bit Wallace tree multiplier using full adder	8.81 ns	6.5534 mW/ 12,627.71 $\mu\text{m}^2$	Synopsys design compiler using SAED 90 nm CMOS technology
5.	4-bit Wallace and Dadda Multiplier Design Using Novel Hybrid 3-2 Counter	4-bit Dadda multiplier with hybrid full adder	220.9 ps	20.34 $\mu\text{W}$	65 nm technology
6.	Design of Delay Efficient Modified 16-bit Wallace Multiplier	16 x 16-bit Wallace multiplier binary to excess code converter	24.948 ns	86.48 mW / 1019 LUT	Xilinx
7.	Design and Analysis of Approximate Compressors for Multiplication	8-bit Dadda multiplier with approximate 4-2 compressor	44.35 ps	1.14 $\mu\text{W}$	32 nm HSPICE simulation
8.	Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers	8-bit Dadda multiplier with dual-quality 4:2 compressors mixed	0.25 ns	424 $\mu\text{W}$ / 423 $\mu\text{m}^2$	45 nm technology node

### 3. 4\*4 MULTIPLIER

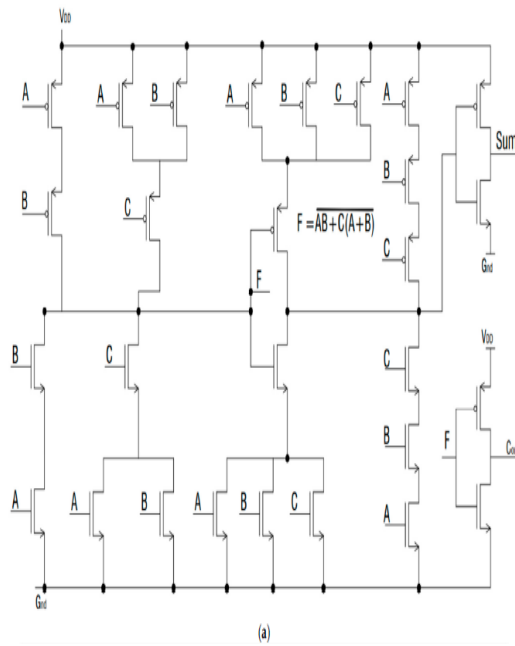
#### Conventional Wallace Multiplier (4\*4)

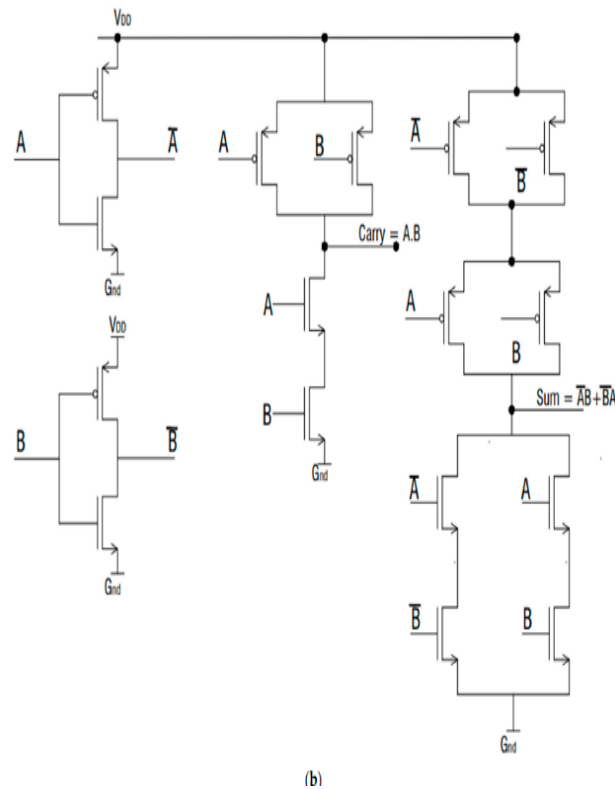
##### Wallace's method binary multiplier.

The attainment of the ultimate outcome is accomplished by diminishing the generator of partial products [22,23] (as depicted in Figure 1) via the utilization of a single-bit full adder and a half adder (as illustrated in Figure 2a,b).



**Figure 1.** Schematic design of 4\*4 partial product generator.

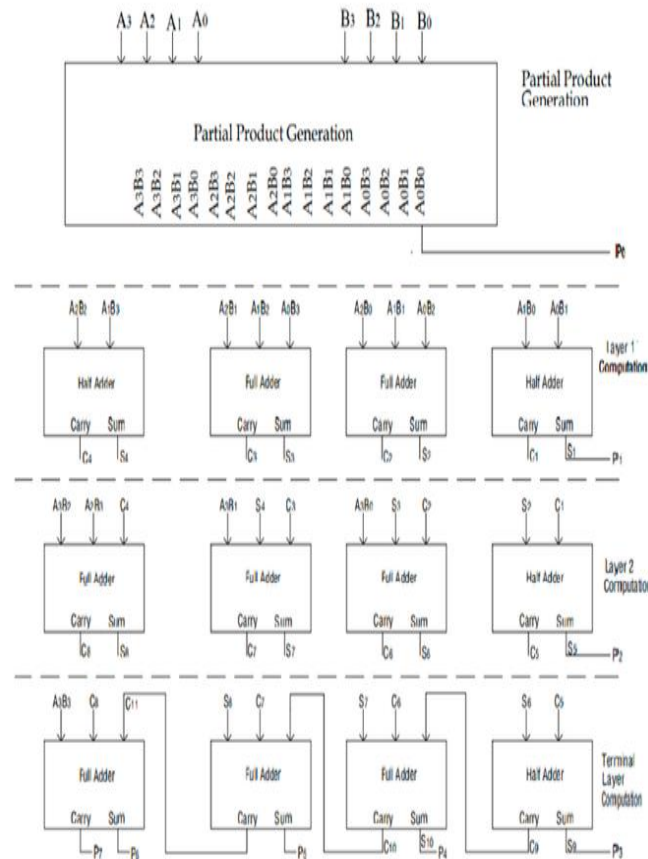




**Figure 2. (a)** This paper presents the schematic design of a static CMOS single-bit full adder.

**(b)** This paper presents the schematic design of a static CMOS single-bit half adder.

After the partial products have been generated, an analysis is conducted on each column, which has a maximum height of  $h = 3$ . The subsequent step involves the reduction of layers. The schematic construction of a conventional 4x4 Wallace multiplier is illustrated in Figure 3. The processing of layers in the provided system is contingent upon the number of items present in the column. Single-bit half adders are utilized in situations when there exists a singular bit that needs to be carried over to the subsequent layer. In contrast, when there are two or three partial products listed in the column, the computational task is carried out using full adders. The sum of all previous values and the remainder are passed on to the next set of entries. The technique described above is repeated until it reaches the final layer, which is defined by only one pair of entries in each column.

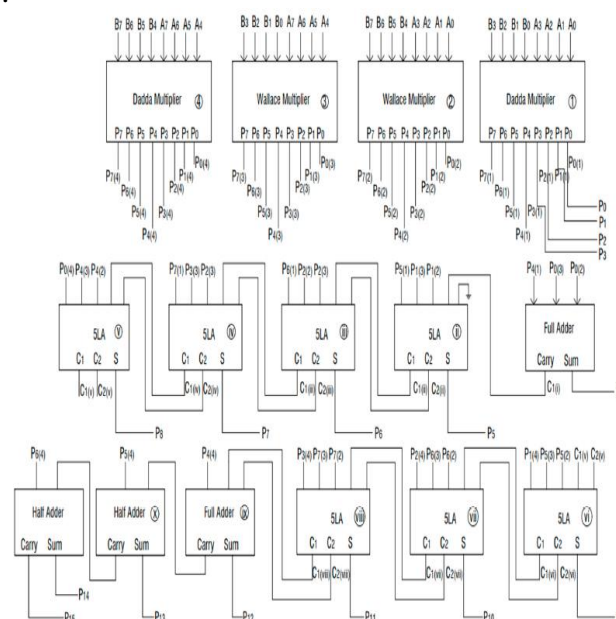


**Figure 3.** The current investigation is on the schematic design of a 4x4 Wallace multiplier.

The computing time is reduced by the Wallace multiplier by the implementation of parallel bit reduction in layers, which involves the utilization of single-bit half adders and full adders. Nevertheless, it is conceivable to further diminish the quantity of logical levels required for executing the summing, hence decreasing the intricacy of the process. The complicated nature of the physical construction of the Wallace multiplier becomes particularly apparent when contemplating higher order multiplication.

#### 4. PROPOSED GD MULTIPLIER (8\*8)

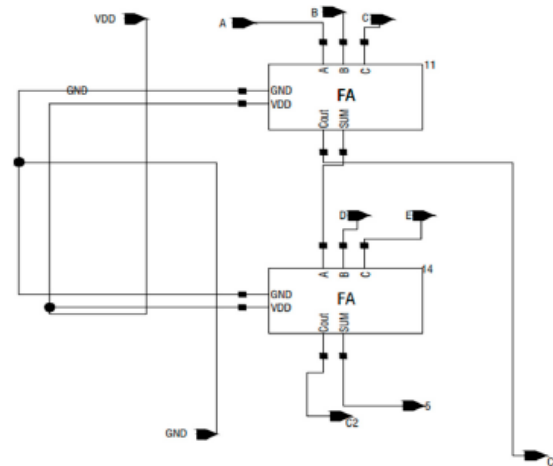
The GD multiplier, as seen in Figure 4, is a high-speed multiplier that employs a parallel multiplication and grouping process. The GD multiplier technique is implemented as follows for binary multiplications of size 8\*8.



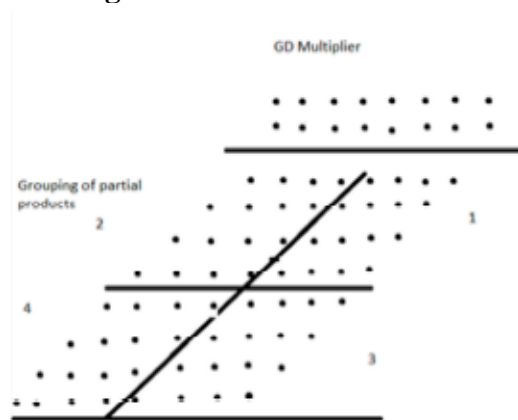
**Figure 4.** This is a schematic diagram of an 8x8 gate diffusion (GD) multiplier.

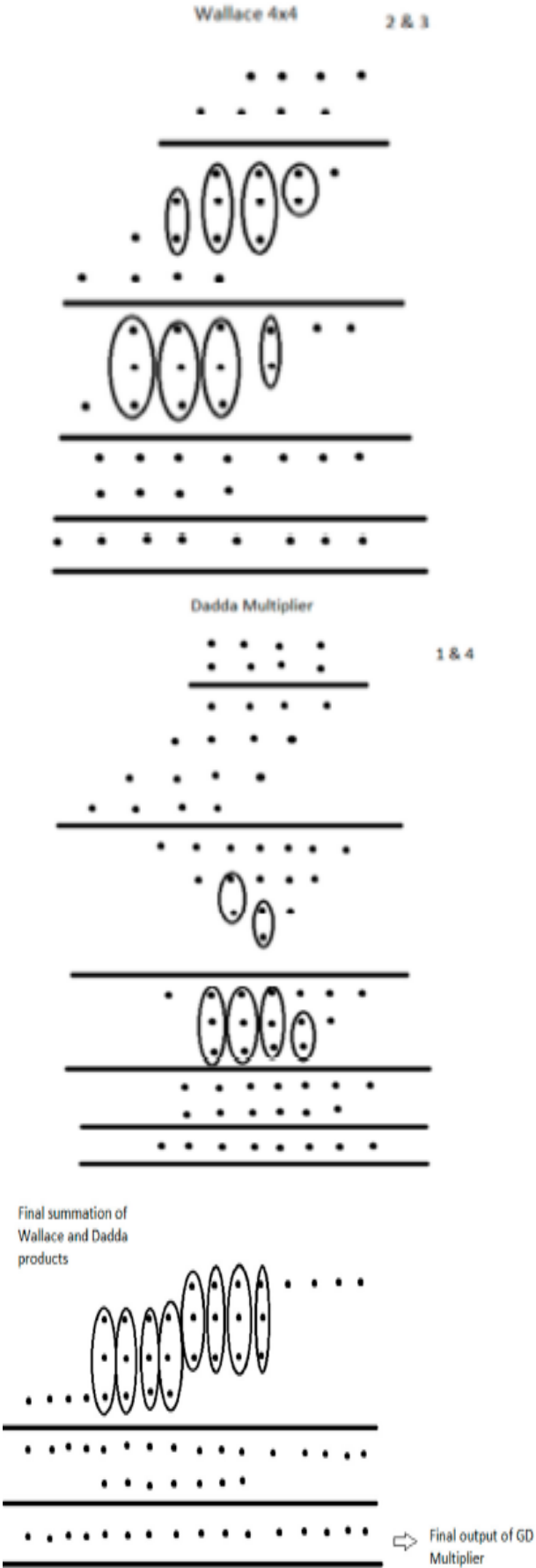
- Decomposing grouped bits and obtaining partial products via parallel grouping is the study's main goal. This study will use a 4x4 Wallace multiplier and a 4x4 Dadda multiplier.
- This analysis concludes the Five-Level Alphabet (5LA) bit examination.
- After generation, imperfect goods are separated into equal-sized groups.

The decomposition method sequentially processes each group through the Dadda and Wallace multipliers. The concurrent implementation of Wallace and Dadda multipliers reduces partial product accumulation computing time. The half adders and byproducts from each group are transferred to the 5LA, as shown in Figure 5, to yield the final findings. The 5LA is a composite architecture with two fully working adders designed to improve carry propagation reduction. An unusual hybrid multiplier, the GD multiplier combines Wallace and Dadda features. Its goal is to synergistically use each multiplier architecture's advantages. Figure 6 illustrates the operating concept of the GD multiplier using an 8\*8 input test case.



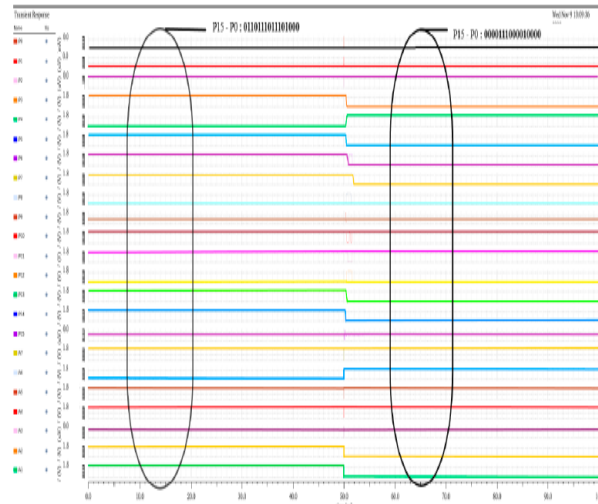
**Figure 5.** Schematic of 5LA.



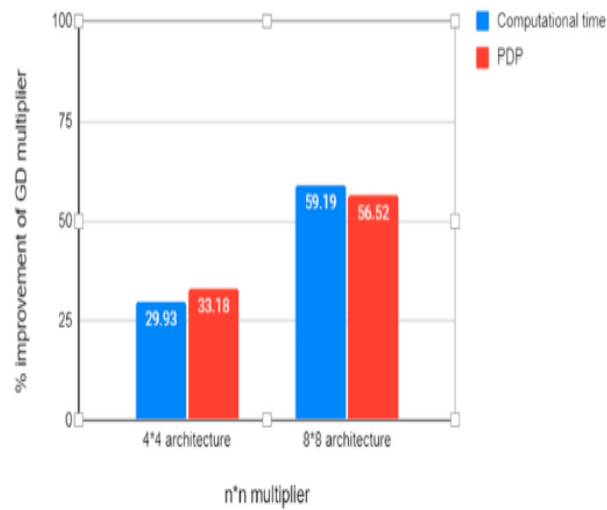


(c)  
Figure:6

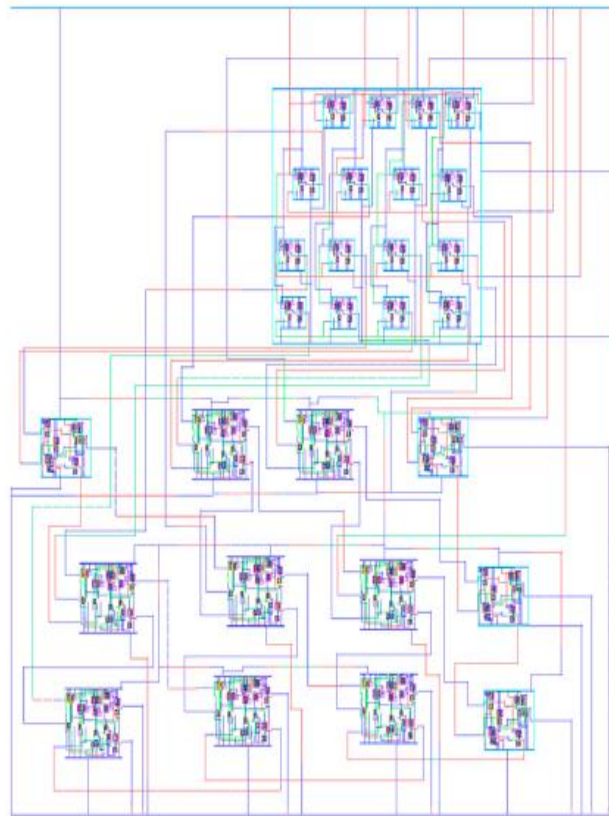
## 5. RESULTS



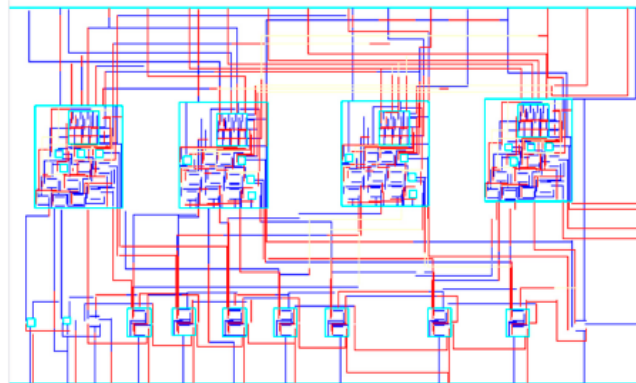
**Figure 7.** The objective of this study is to show a simulation waveform of an 8x8 gate diffusion (GD) multiplier.



**Figure: 8** Comparison of the Proposed Multiplier with the Existing Multiplier



**Figure 9:** The physical design execution of a 4x4 Wallace multiplier was carried out at the DRC and LVS stages.



**Figure 10. :** The DRC and LVS were cleaned extensively during the physical design implementation of an 8x8 Grouping and Decomposition (GD) multiplier.

## 5. CONCLUSION

In this paper, a novel high-speed binary multiplier based on grouping and deconstructing is presented. Applications of 180nm CMOS technology were used to implement the multiplier. Compared to parallel designs, the front-end architecture of the proposed multiplier shows a noteworthy reduction of 56.29% in processing time and 53.49% in power-delay product. Regardless of the number of partial products involved, the suggested multiplier efficiently reduces the bit count simultaneously by using the grouping and decomposition technique. When compared to the Wallace and Dadda multipliers, respectively, the measured reductions in computing time of the GD multiplier are 47.52% and 43.85%, respectively. This work compares the suggested architecture with current multipliers, highlighting the advantages of the parallel grouping and decomposition approach over other approaches such as Wallace, Dadda, and related methods. Because of this particular feature, it is very suitable for applications related to high-speed Very Large Scale Integration (VLSI).

## REFERENCES

1. Wallace, C.S. A suggestion for a fast multiplier. IEEE Trans. Electron. Comput. **1964**, 1, 14–17.

2. Kulkarni, P.; Gupta, P.; Ercegovic, M. Trading accuracy for power with an underdesigned multiplier architecture. In Proceedings of the 2011 24th International Conference on VLSI Design, Chennai, India, 2–7 January 2011; pp. 346–351.
3. Habibi, A.; Wintz, P.A. Fast multipliers. *IEEE Trans. Comput.* **1970**, 100, 153–157.
4. Bandi, V.L.; Gamini, P.; Harshith, B.S. Performance analysis of dadda multiplier using modified full adder. *Int. J. Innov. Res. Comput. Commun. Eng.* **2018**, 6, 126–130.
5. Baran, D.; Aktan, M.; Oklobdzija, V.G. Multiplier structures for low power applications in deep-CMOS. In Proceedings of the 2011 IEEE International Symposium of Circuits and Systems (ISCAS), Rio de Janeiro, Brazil, 15–18 May 2011; pp. 1061–1064.
6. Townsend, W.J.; Swartzlander, E.E., Jr.; Abraham, J.A. A comparison of Dadda and Wallace multiplier delays. In *Advanced Signal Processing Algorithms, Architectures, and Implementations XIII*; SPIE: San Diego, CA, USA, 2003; Volume 5205.
7. Weste, N.H.; Harris, D.M. *Harris, Integrated Circuit Design*; Pearson: Boston, MA, USA, 2010.
8. Maurya, K.A.; Lakshmana, Y.R.; Sindhuri, K.B.; Kumar, N.U. Design and implementation of 32-bit adders using various full adders. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017.
9. Bandi, V. Performance Analysis Modified Full for Vedic Multiplier using Adders. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017.
10. Ram, G.C.; Lakshmana, Y.R.; Rani, D.S.; Sindhuri, K.B. Area Efficient Modified Vedic Multiplier. In Proceedings of the 2016 International Conference On Circuit, Power and Computing Technologies, Nagercoil, India, 18–19 March 2016; pp. 1–5.
11. Akbari, O.; Kamal, M.; Afzali-Kusha, A.; Pedram, M. RAP-CLA: A reconfigurable approximate carry look-ahead adder. *IEEE Trans. Circuits Syst. II Express Briefs* **2016**, 65, 1089–1093.
12. Raha, A.; Jayakumar, H.; Raghunathan, V. Input-based dynamic reconfiguration of approximate arithmetic units for video encoding. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2015**, 24, 846–857.
13. Sampson, A.; Dietl, W.; Fortuna, E.; Gnanapragasam, D.; Ceze, L.; Grossman, D. EnerJ: Approximate data types for safe and general low-power computation. In Proceedings of the 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 11), New York, NY, USA, 4–8 June 2011; pp. 164–174.
14. Sebastian, A.; Jose, F.; Gopakumar, K.; Thiagarajan, P. Design and Implementation of an Efficient Dadda Multiplier Using Novel Compressors and Fast Adder. In Proceedings of the 2020 International Symposium on Devices, Circuits and Systems (ISDCS), Howrah, India, 4–6 March 2020; IEEE: Piscataway, NJ, USA; pp. 1–4.
15. Srinivas, L., & Umapathi, N. (2022, May). New realization of low area and high-performance Wallace tree multipliers using booth recoding unit. In *AIP Conference Proceedings* (Vol. 2393, No. 1, p. 020221). AIP Publishing LLC.
16. N. Umapathi, G. M. Krishna and L. Srinivas, "A Comprehensive Survey on Distinctive Implementations of Carry Select Adder," *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, 2021, pp. 1-5, doi: 10.1109/ICNTE51185.2021.9487718.
17. Murali Krishna G., Karthick G., Umapathi N. (2021) Design of Dynamic Comparator for Low-Power and High-Speed Applications. In: Kumar A., Mozar S. (eds) ICCCE 2020. Lecture Notes in Electrical Engineering, vol 698. Springer, Singapore. [https://doi.org/10.1007/978-981-15-7961-5\\_110](https://doi.org/10.1007/978-981-15-7961-5_110)
18. N. Umapathi, G.L. 2020. Design and Implementation of Low Power 16x16 Multiplier using Dadda Algorithm and Optimized Full Adder. *International Journal of Advanced Science and Technology*. 29, 3 (Feb. 2020), 918 - 926.
19. Swarnalatha, B., & Umapathi, N. (2022). Voltage over Scaling-Based Dadda Multipliers for Energy-Efficient Accuracy Design Exploration. *Specialusis Ugdymas*, 2(43), 2942-2956.
20. Prasad, R., Umapathi, N., & Karthick, G. (2022). Error-Tolerant Computing Using Booth Squarer Design and Analysis. *Specialusis Ugdymas*, 2(43), 2970-2985.