ISSN 2277-2685
IJESR/Jan-Mar. 2025/ Vol-15/Issue-1s/347-356
Mynampati Aarthi *et. al.*, / International Journal of Engineering & Science Research

Design Of Power And Area Efficient Approximate Multipliers

[1]Ms. S Surekha, [2]Mynampati Aarthi, [3]Gangapurapu Abhinaya , [4]Karra Gouthami

[1]Assistant professor, Electronics and Communication Engineering, Bhoj Reddy Engineering College For Women

[2,3,4]B.Tech Students, Department of Electronics and Communication Engineering, Bhoj Reddy Engineering College For Women

Abstract

Approximate computing can decrease the design complexity with an increase in performance of area, delay and power efficiency for error resilient applications. This brief deals with a new design approach for approximation of multipliers. The partial products of the multiplier are altered to introduce varying probability terms. Logic complexity of approximation is varied for the accumulation of altered partial products based on their probability. The proposed approximation is utilized in two variants of 8-bit multipliers and that proposed approximated dada multiplier achieve power saving, area and delay respectively, compared to an exact dada multiplier. Performance of the proposed multipliers is evaluated with an image processing application.

1- INTRODUCTION

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.

The common multiplication method is "add and shift" algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, Modified Booth algorithm is one of the most popular algorithms. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages. Further by combining both Modified Booth algorithm and Wallace Tree technique we can see advantage of both algorithms in one multiplier. However with increasing parallelism, the amount of shifts between the partial products and intermediate sums to be added will increase which may result in reduced speed, increase in silicon area due to irregularity of structure and also increased power consumption due to increase in interconnect resulting from complex routing. On the other hand "serial-parallel" multipliers compromise speed to achieve better performance for area and power consumption. The selection of a parallel or serial multiplier actually depends on the nature of application. In this lecture we introduce the multiplication algorithms and architecture and compare them in terms of speed, area, power and combination of these metrics. The basic method of multiplier is explains below

```
  123
x 456
=====
  738  (this is 123 x 6)
 615   (this is 123 x 5, shifted one position to the left)
+ 492   (this is 123 x 4, shifted two positions to the left)
=====
56088
```

The binary multiplication also happens in same way of digit multiplication as shown in below example here by getting partial products and gates are used and we are using adder (half adder ,full adder)adding the columns .

2- LITERATURE SURVEY

1. Introduction to Approximate Computing:

Approximate computing is an emerging paradigm that exploits the inherent error tolerance of applications such as image processing, machine learning, and multimedia processing to improve efficiency. Several studies demonstrate how error-tolerant applications can trade off accuracy for improvements in power, area, and performance. Approximate arithmetic units, particularly multipliers, have been identified as key candidates for this optimization due to their high power consumption and complexity.

2. Truncated Multipliers:

In "Design of Low-Power Truncated Multipliers" by S. K. Gupta et al., truncated multiplication is proposed as a power-efficient method. By discarding less significant bits during multiplication, both the power and area are reduced, with minor accuracy loss. This approach has proven effective in many real-time applications that can tolerate small errors. A common technique involves reducing the number of partial products, cutting the power required for their summation.

3. Approximate Partial Product Generation:

H. Jiang et al. in "Low-Power Approximate Multiplication with Inaccurate Partial Product Reduction" proposed a novel way to reduce the complexity of partial product generation by skipping or simplifying some of the least significant bit products. The study shows significant reductions in power consumption and area overhead compared to traditional multipliers, with minimal impact on output quality. This work suggests that approximate multipliers are particularly effective in applications like video and audio processing, where human perception is more tolerant of errors.

4. Logarithmic Approximate Multipliers:

A logarithmic-based approach is described in "Logarithmic Approximation for High-Speed Low-Power Multipliers" by V. Gupta et al., where multiplication is approximated as addition in the logarithmic domain. This reduces both the number of computations and circuit complexity. The logarithmic approximation achieves significant savings in area and power consumption, though at the cost of increased approximation errors.

5. Error-Tolerant Multiplier Designs:

In "Design of Error-Tolerant Low-Power Multipliers for Digital Signal Processing" by J. Liang et al., a new class of error-tolerant multipliers (ETMs) is

presented, which focuses on reducing computational complexity while retaining high precision for significant bits. The study proposes partitioning the multiplication process into critical and non-critical regions, allowing designers to optimize the power and area of non-critical regions, where errors are more tolerable.

6. Hybrid Approximate Multiplier Architectures:

In "Hybrid Approximate Multiplier Design for Power-Efficient Systems" by S. Narayana Moorthy et al., a hybrid approach is introduced, combining various approximation techniques such as truncated multiplication and approximate adders. The study demonstrates that hybrid approaches offer more flexibility in balancing power, area, and accuracy, making them suitable for diverse applications. The paper shows how adaptive precision control can further reduce power consumption based on dynamic workload requirements.

7. Approximate Multipliers Using Approximate Adders:

The paper "Design of Energy-Efficient Approximate Multipliers Using Approximate Adders" by M. Shafique et al. investigates the impact of using approximate adders in the summation of partial products. The authors show that using approximate adders, especially for the lower significant bits, can significantly reduce the overall energy and area requirements. This approach offers a good balance between power savings and acceptable accuracy for error-resilient applications.

8. Design Methodologies for Approximate Multipliers:

In "Design Methodologies for Approximate Arithmetic Circuits: A Survey" by Z. Wang et al., the authors provide a comprehensive review of various methodologies for designing approximate arithmetic circuits, including multipliers. The paper discusses key design principles such as precision scaling, dynamic accuracy adjustment, and circuit simplifications. The survey emphasizes the importance of selecting appropriate approximation techniques based on the target application's error tolerance and performance requirements.

9. Approximate Wallace Tree Multipliers:

A study by R. Maheshwari et al. in "Approximate Wallace Tree Multiplier with Low Power and Area Overhead" explores the use of approximate compressors in Wallace tree structures. By simplifying the compression stage of the Wallace tree, the power and area savings are significant. This approach is effective for applications requiring high-speed multipliers with moderate accuracy constraints.

10. Practical Applications of Approximate Multipliers:

For example "Application of Approximate Multipliers in Convolutional Neural Networks" by Q. Zhang et al., the authors demonstrate how approximate multipliers can reduce power and area in neural network hardware accelerators without significantly degrading inference accuracy. Similar applications in image processing and video encoding demonstrate the practical relevance of approximate multipliers for energy-efficient computation.

3- SOFTWARE REQUIREMENTS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

Xilinx Vivado

software is used by the VHDL/VERILOG designers for performing Synthesis operation. Any simulated code can be synthesized and configured on FPGA. Synthesis is the transformation of HDL code into gate level net list. It is an integral part of current design flows.

- Verilog synthesis tools can create logic-circuit structures directly from Verilog behavioral description and target them to a selected technology for realization (I.e,translate Verilog to actual hardware).
- Using Verilog , we can design ,simulate and synthesis anything from a simple combinational circuit to a complete microprocessor on chip.
- Verilog HDL has evolved as a standard hardware description language. Verilog HDL offers many useful features for hardware design.

Program Structure

The basic unit and programming in Verilog is ”MODULE”.(a text file containing statements and declarations)

- A Verilog module has declarations that describes the names and types of the module inputs and outputs as well as local signals, variables, constants and functions that are used internally to the module, are not visible outside.
- The rest of the module contains statements that specify the operation of the module output and internal signals.
- Verilog is a case-sensitive language like C. Thus sense, Sense, SENSE,…etc., are all treated as different entities / quantities in Verilog.

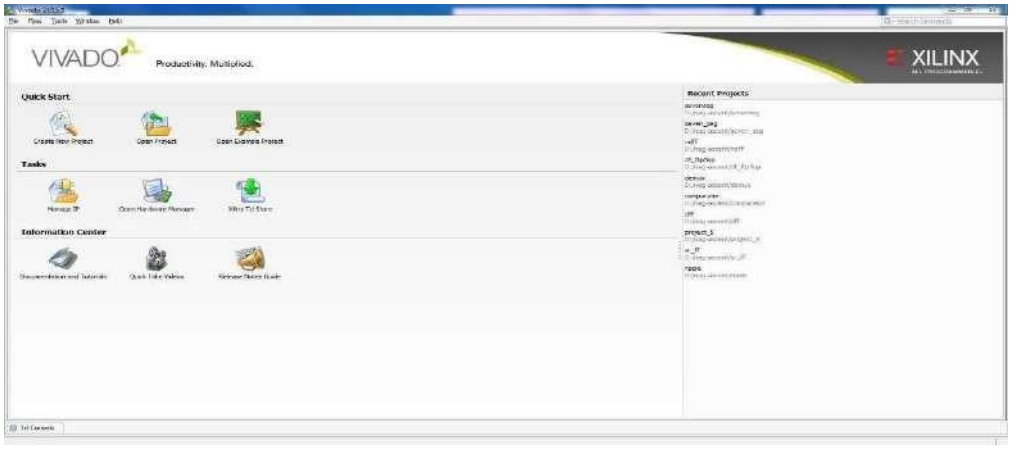


Fig 2.1 Vivado desktop

4- DADDA MULTIPLIER

In modern computing systems, multipliers are a critical component of digital signal processing (DSP), image processing, machine learning, and other data-intensive applications. However, traditional multipliers are power-hungry and require significant chip area, especially in systems where high precision is paramount. With the increasing demand for energy-efficient hardware, especially in portable and embedded devices, there is a growing interest in approximate computing, which seeks to trade-off accuracy for improvements in power consumption, area, and performance.

Existing System

Multipliers are critical in the present advanced flag handling and for different applications. Numerous scientists have attempted and many are endeavoring to plan the multipliers which will enhance the

outline parameters like – speed, low power utilization less range or mix of these in one multiplier by making them appropriate for various fast, low power VLSI usage. The basic idea of DADDA multiplier depends on the underneath framework shape appeared in Fig 1 The partial product is framed in the principal organize by AND stages which is delineated in Fig 2.

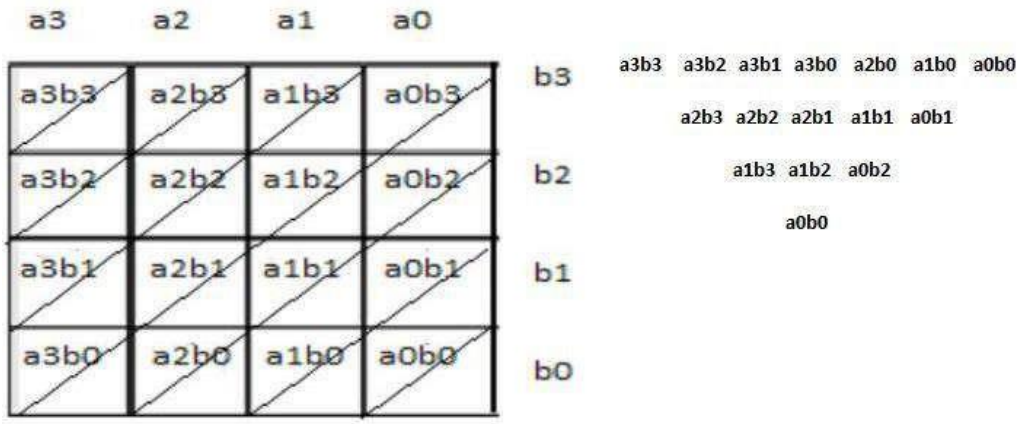


Fig 3.1 Algorithm of Dadda multiplier

Algorithm:

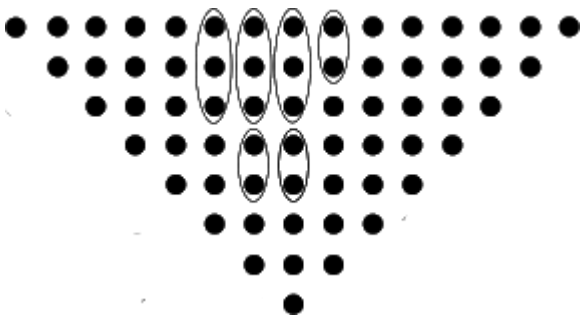
Let, us assume the final two-rowed matrix height d1 = 2, based on d1 the successive matrix heights are obtained from dj+1 = 1.5 * dj , where j = 1,2,3,4, ..., Rounding of fraction in this matrix height should be done down to least. i.e., 13.5 = 13(rounded). The matrix heights will be in this fashion 2,3,4,6,9,13,19,28..... Finally the largest dj should be obtained such that derived matrix height shouldn't exceed the Matrix overall height.

1. In the first reduction stage, the column compression is to carried with the [3,2] and [2,2] counters such that the obtained reduced matrix height should not exceed dj.
2. During the compression, the sum is to be passed to same column in the next reduction stage and the carry is to be passed to the next column.

The above two steps are to be repeated until a final two-rowed reduced matrix is obtained.

Illustrating the Algorithm with 8 by 8 Multiplication:

1. The Partial products obtained are to be arranged in the Tree form. The Matrix heights possible are 2,3,4,6(where 6 < 8). The largest dj = 6.



In the above,1st reduction stage. The column's 0 to 5 have height not more than '6'. The column 6 height is '7' and it is reduced to 6 by a [2,2] counter. By considering the previous carry from column 6 the

column height of column 7 is '9'. To reduce column 7 height to '6', a [3,2] and a [2,2] counters are used. By considering the two carry's from column 7 the column 8 requires a [3,2] and a [2,2] counter. Similarly, the column 9 requires a [3,2] counter.

Proposed System

Implementation of multiplier comprises three steps: generation of partial products, partial products reduction tree, and finally, a vector merge addition to produce final product from the sum and carry rows generated from the reduction tree. Second step consumes more power. In this brief, approximation is applied in reduction tree stage.

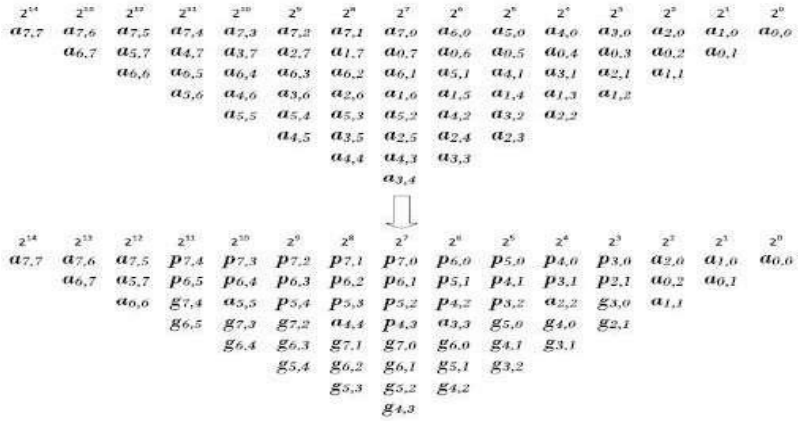


Fig. 3.6 Transformation of generated partial products into altered partial products.

A 8-bit unsigned1 multiplier is used for illustration to describe the proposed method in approximation of multipliers. Consider two 8-bit unsigned input operands $\alpha = \sum_{m=0}^{7} \alpha_m 2^m$ and $\beta = \sum_{n=0}^{7} \beta_n 2^n$ The partial product am, n = αm ⋅ βn in Fig. 1 is the result of AND operation between the bits of αm and βn. From statistical point of view, the partial product am, n has a probability of 1/4 of being 1. In the columns containing more than three partial products, the partial products am, n and an, m are combined to form propagate and generate signals as given in. The resulting propagate and generate signals form altered partial products pm, n and gm, n. From column 3 with weight 23 to column 11 with weight 211, the partial products am, n and an, m are replaced by altered partial products pm, n and gm, n. The original and transformed partial product matrices are shown in Fig. 1

5- ADVANTAGES, DISADVANTAGES AND APPLICATIONS

Advantages

Designing approximate multipliers offers several advantages, especially in applications where perfect accuracy is not essential, such as image processing, machine learning, and neural networks. Here are some key benefits:

1. **Power Efficiency:** Approximate multipliers consume significantly less power compared to accurate multipliers. This is because they often simplify operations, reduce the number of logic gates, or use low-power components.
2. **Reduced Area:** By simplifying the logic, approximate multipliers require fewer transistors,

leading to a smaller silicon footprint. This is crucial for applications in embedded systems and mobile devices, where area constraints are important.

3. **Higher Speed:** Approximate multipliers often have faster computation times since they perform fewer or simplified operations. This can improve the overall speed of a system, especially in real-time applications.

Disadvantages

While approximate multipliers offer several advantages, they also come with some notable disadvantages. These trade-offs must be carefully considered depending on the application. Here are the key drawbacks:

1. **Reduced Accuracy:** The primary disadvantage of approximate multipliers is their intentional inaccuracy. The simplifications used in their design can introduce errors in the output, which might not be acceptable for certain applications requiring high precision, such as scientific computing or financial calculations.
2. **Error Propagation:** In systems that rely on multiple stages of calculations, errors introduced by approximate multipliers can propagate and accumulate, leading to more significant inaccuracies in the final output. This may impact the reliability of the system over time, especially in complex computations.
3. **Limited Applicability:** Approximate multipliers are not suitable for all applications. For example, in critical systems where precision is paramount (e.g., medical devices, aerospace systems, cryptography), even small inaccuracies can lead to failure or undesirable results.

Applications

1. Approximate Multipliers in Image Processing:

In image processing, a small loss in accuracy can often go unnoticed because the human eye is less sensitive to slight changes in image quality. Approximate multipliers are used to speed up and reduce the power consumption of common image processing operations such as:

Filtering: Image filters (e.g., blur, sharpen, edge detection) involve repetitive multiplications that can be approximated to speed up the processing.

Compression: Algorithms like JPEG and MPEG use discrete cosine transforms (DCTs), which involve multiplications that can be approximated without significantly degrading image or video quality.

Enhancement: Operations like contrast adjustment, brightness control, and color correction can also use approximate multipliers to enhance efficiency, as small errors are typically not perceptible.

2. Approximate Multipliers in FIR/IIR Filters:

Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters are common in signal processing for applications like noise reduction and audio processing. These filters involve frequent multiplications of coefficients and input data, making them good candidates for approximation:

FIR Filters: In FIR filters, each output sample is a weighted sum of current and past input samples. Approximate multipliers reduce power and area usage in these operations by allowing small errors in the multiplication of the filter coefficients.

6- RESULTS

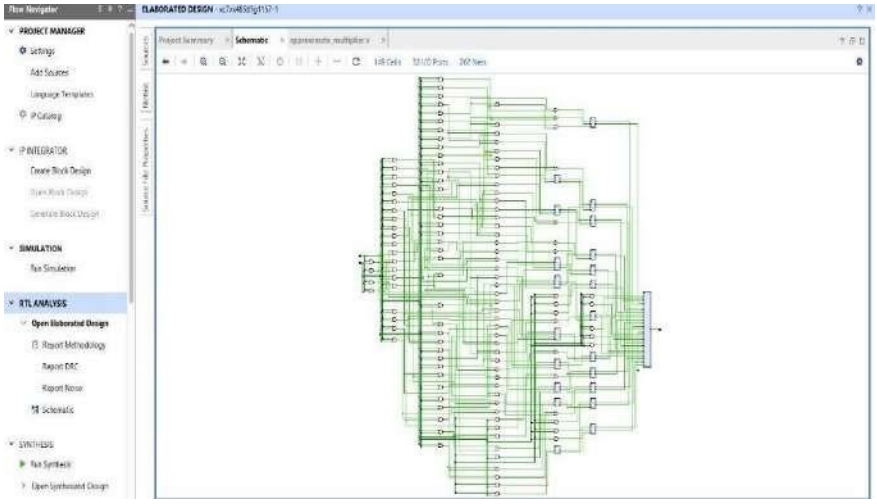


Fig 6.1 RTL internal block diagram

The RTL design shown in the image is for an approximate multiplier created using Vivado, a tool commonly used for FPGA and ASIC design. The schematic view provides a visual representation of the logic, displaying 149 cells, 32 input/output ports, and 202 connections (nets). Approximate multipliers are designed to trade off some accuracy in Favor of improvements in speed, power efficiency, or reduced hardware complexity. These types of multipliers are particularly useful in applications like image processing or machine learning, where minor inaccuracies are acceptable. The schematic is part of the "Elaborated Design" stage in Vivado, meaning the design has been compiled and optimized for analysis but has not yet been synthesized. This stage allows designers to examine the structure of the circuit and make any necessary adjustments before proceeding to synthesis.

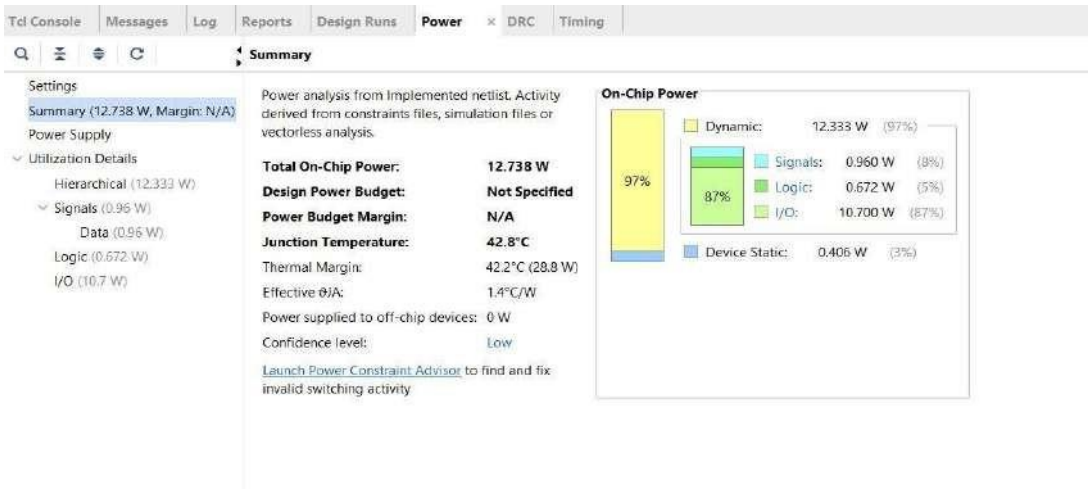


Fig 5.2 Efficiency of Power

Power efficiency refers to the ratio of useful output power to the total input power consumed by a system, device, or process. Improving power efficiency reduces energy waste, lowers energy costs, and minimizes environmental impact.

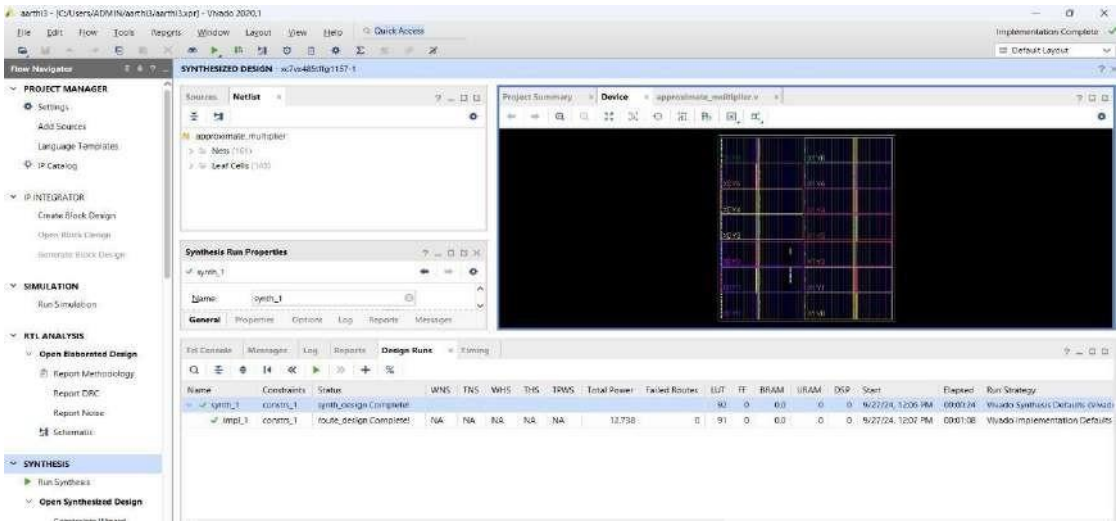


Fig 5.3 Area efficiency

Area efficiency refers to the optimization of digital circuit design to minimize the physical area occupied on a semiconductor chip, while maintaining or improving performance. It's crucial in modern electronics, as it directly impacts.

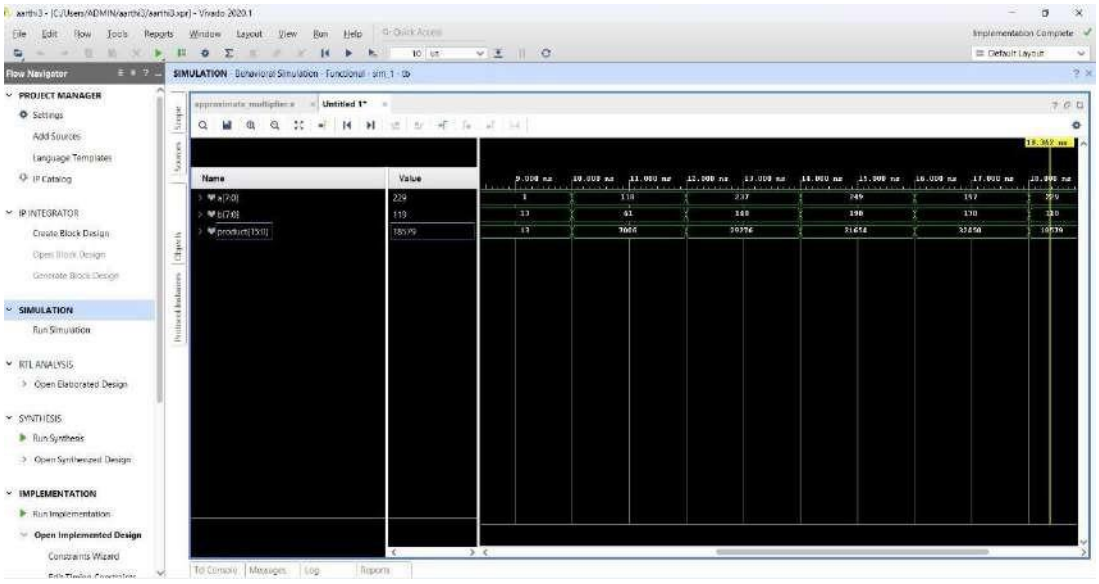


Fig 5.4 simulation results

7- CONCLUSION

In this project, The proposed configurable approximate dadda multiplier is better than the existing traditional dadda multiplier. In this brief, to propose efficient approximate multipliers, partial products of the multiplier are modified using generate and propagate signals. Approximation is applied using simple OR gate for altered generate partial products. Approximate half-adder, full adder, and 4-2 compressor are proposed to reduce remaining partial products. The proposed multiplier designs can be used in applications with high-speed minimal loss in output quality while saving significant power and area.

REFERENCES

[1] S. Narayana Moorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications," IEEE Trans. Very Large Scale Integration. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[2] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Trans Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.

[3] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-efficient approximate multiplication circuits through partial product perforation," IEEE Trans. Very Large Scale Integration. (VLSI) Syst., vol. 24, no. 10, pp. 3105–3117, Oct. 2016.

[4] C.-H. Lin and C. Lin, "High accuracy approximate multiplier with error correction," in Proc. IEEE 31st Int. Conf. Comput. Design, Sep. 2016,pp. 33–38.

[5] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modelling and analysis of circuits for approximate computing," in Proc.IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Oct. 2017.

[6] S. Suman et al., "Image enhancement using geometric mean filter and gamma correction for WCE images," in Proc. 21st Int. Conf., Neural Inf. Process. (ICONIP), 2018.

[7] T. Yang, T. Ukezono, and T. Sato "A Low-Power High-Speed Accuracy-Controllable Approximate Multiplier Design", Proc. Of IEEE Asia and South Pacific Design Automation Conference (ASPD 1,.AC), 2018.

[8] T. Yang, T. Ukezono, and T. Sato "A Low-Power High-Speed Accuracy-Controllable Approximate Multiplier Design", Proc. Of IEEE Asia and South Pacific Design Automation Conference (ASPD 1,.AC), 2018.

[9] T. Yang, T. Sato and T. Ukezono, "An Approximate Multiply- Accumulate Unit with Low Power and Reduced Area," IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Miami, FL, USA, pp. 385-390, 2019.

[10] W. Liu, T. Zhang, E. McLarnon, M. O. Neill, P. Montuschi and F. Lombardi, "Design and Analysis of Majority Logic Based Approximate Adders and Multipliers," IEEE Transactions on Emerging Topics in Computing, 2019.