



# ROBA MULTIPLIER: ENABLING HIGH-SPEED AND ENERGY-EFFICIENT DSP APPLICATIONS

<sup>1</sup> Naveen M, <sup>2</sup> Vishnu Vardan Reddy N, <sup>3</sup> Pavan Kalyan G, <sup>4</sup> Mahesh M, <sup>5</sup> Chandra Mohan Reddy M,

<sup>1</sup>Student, <sup>2</sup>Student, <sup>3</sup>Student, <sup>4</sup>Student, <sup>5</sup> Professor

<sup>1</sup> Department of Electronics and Communication Engineering (ECE),

<sup>1</sup>Narayana Engineering College, Nellore, India

**Abstract:** The main objective of this project in order to propose an approximate multiplier that is high speed yet energy efficient. The approach is to round the operands to the nearest exponent of two. This way the machine intensive part of the multiplication is omitted up speed and energy consumption. The potency of the planned multiplier factor is evaluated by comparing its performance with those of some approximate and correct multipliers using different design parameters. The proposed approach is applicable to both signed and unsigned multiplications. We propose three hardware implementations of the approximate multiplier that includes one for the unsigned and two for the signed operations. The approach is applicable to both signed and unsigned multiplications. We propose three hardware implementations of the approximate multiplier that includes one for the unsigned and two for the signed operations. To achieve imprecise multiplication, simple addition and shift operations are performed using rounded and original input operands. Approximate multipliers have better performance in terms of speed, power, and area compared to exact multipliers. The parameters of RoBA multiplier which can be used in the voice or image smoothing applications in the DSP. For extension in the convolution process of the FIR Filter RoBA multiplier is used.

**Keywords-** VLSI, DSP, XILINX Software.

## I. INTRODUCTION

Energy and speed are one of the main design requirements in almost any electronic systems, especially the portable ones such as smart phones, tablets, and different gadgets. Digital signal processing (DSP) blocks are key components of these portable devices for realizing various multimedia applications. The computational core of these blocks is the arithmetic logic unit where multiplications have the greatest share among all arithmetic operations performed in these DSP systems. Improving the speed and power/energy-efficiency characteristics of multipliers plays a key role in improving the efficiency of processors. The approximation may be performed using different techniques such as allowing some timing violations. utilize approximate computing techniques to improve the performance and efficiency of multipliers without significantly impacting the overall quality of the output.

## II. RELATED WORK

Several works have been proposed for Robo multiplier using various techniques for enabling high-speed and energy-efficient. Here the system is proposed using DSP and VLSI applications concept.

### A. VLSI

Digital circuit has rapidly evolved over the last twenty-five years. The earliest digital circuits were designed with vacuum tubes and transistors. Integrated circuits were then invented where logic gates were placed on a single chip. The first IC chip was small scale integration (SSI) chips where the gate count is small. When technology became sophisticated, designers were able to place circuits with hundreds of gates on a chip. These chips were called MSI chips with advent of LSI; designers could put thousands of gates on a single chip. At this point, design process is getting complicated and designers felt the need to automate these processes. With the advent of VLSI technology, designers could design single chip with more than hundred thousand gates. Because of the complexity of these circuits computer aided techniques became critical for verification and for designing these digital circuits. One way to lead with increasing complexity of electronic systems and the increasing time to market is to design at high levels of abstraction. Traditional paper and pencil and capture and simulate methods have largely given way to the described UN synthesized approach.

### B. VERILOG DESIGN

Verilog was started in the year 1984 by Gateway Design Automation Inc as a proprietary hardware modeling language. It is rumored that the original language was designed by taking features from the most popular HDL language of the time, called HiLo, as well as from traditional computer languages such as C. At that time, Verilog was not standardized and the language modified itself in almost all the revisions that came out within 1984 to 1990. Verilog simulator first used in 1985 and extended substantially through 1987. The implementation of Verilog simulator sold by Gateway. The first major extension of Verilog is Verilog-XL, which added a few features and implemented the infamous "XL algorithm" which is a very efficient method for doing gate-level simulation. Later

1990, Cadence Design System, whose primary product at that time included thin film process simulator, decided to acquire Gateway Automation System, along with other Gateway products., Cadence now become the owner of the Verilog language, and continued to market Verilog as both a language and a simulator. At the same time, Synopsys was marketing the top-down design methodology, using Verilog. This was a powerful combination.

The Robo Multiplier also uses the following applications

- 1) DSP
- 2) VLSI
- 3) VERILOG

The software used for Roba multiplier

- Xilinx software

### III. METHODOLOGY

In this work, the classification of addition is one of the most commonly used arithmetic operation in microprocessor, digital signal processor etc. It can also be used as a building block for synthesis of all other arithmetic operations. Therefore, as far as the efficient implementation of an arithmetic unit is concerned, the binary adder structure becomes a very critical hardware unit. While adders can be constructed for a lot of numerical expressions like Binary-coded decimal or excess-3, the most frequently used adders operate numbers which are binary. In this project, assessments of the classified binary adder architectures are given. From the huge member of adders we have got, we implemented the VHDL (Hardware Description Language) code for Ripple-carry and Carry-look ahead adder to highlight the common performance properties belong to their classes. Throughout the next section, we provide you with a brief description of the studied adder architectures

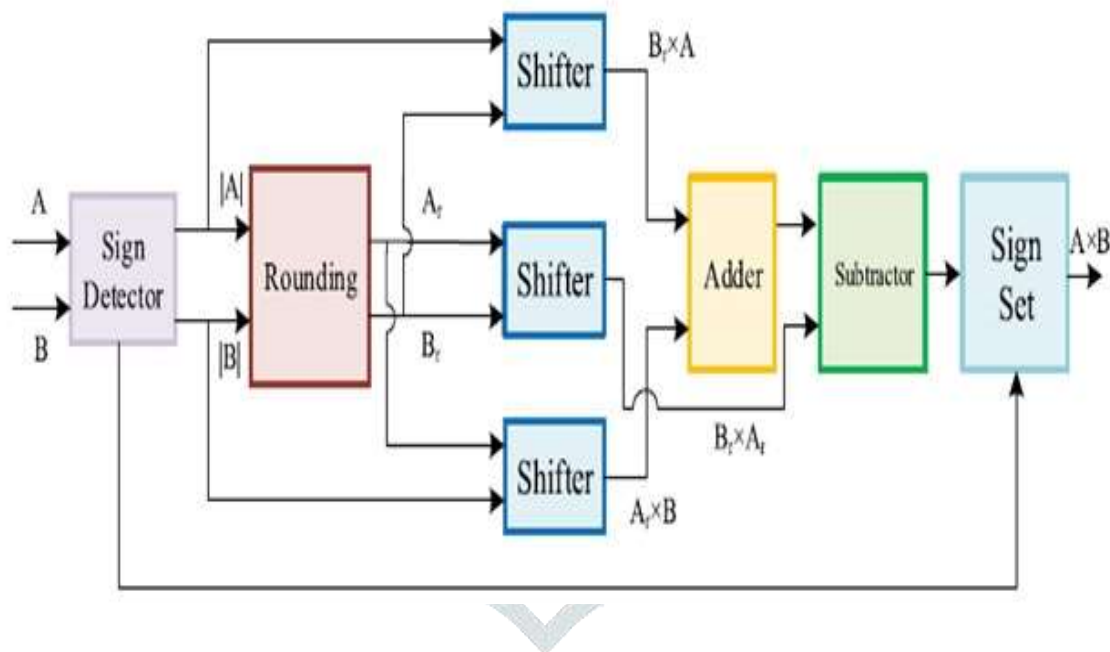


Fig 1. Block Diagram of Proposed Methodology

**A. Sign Detector**

Sign Detector The sign detector detects the sign of the input values & gives the output. It extracts the most significant bit (MSB) of the input value. If the MSB of the input is '0', it will be considered as positive & if the MSB is '1', it will be regarded as negative.

**B. Rounding**

Rounding This block rounds off the input values to the nearest exponent of two. In the proposed method  $Z[i]$  is one in the following cases: In the first case,  $Z[i]$  is one and all the left sided bits are zero while  $Z[i-1]$  is zero. In the second case, when  $Z[i]$  and all the left sided bits are zero,  $z[i-1]$  and  $z[i-2]$  are both one. Here we are using three-barrel shifter blocks, the products  $A_r \times B$ ,  $B_r \times B$ ,  $A_r \times B_r$  are determined. These input bits are shifted towards left side.

**C. Adder**

Parallel-Prefix adders perform parallel addition i.e. most important in microprocessors, DSPs, mobile devices and other high-speed applications. Parallel- Prefix adders are primarily fast when compared to adder. The logic complexity & delay can be reduced by these adders. It improves the factors like area and power. Parallel-Prefix adders are designed by considering carry look adder as base.

$$\text{Calculation of carries } G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1}; P_{i:j} = P_{i:k} \cdot P_{k-1:j}$$

**D. Ripple carry adders (RCA)**

This popular adder architecture, ripple carry adder consists of cascaded full adders as shown. It is formed by cascading full adder blocks in series with one another. The output carry of one stage is fed directly to the input carry of the next stage.

**E. Carry Look Ahead Adder**

A Carry Look Ahead Adder has the ability to generate faster carries because of generation of carry bits in parallel by a supplementary circuit whenever inputs are changing. This technique extensively uses carry bypass logic to haste up the propagation of carry. In Carry look ahead logic the generation and propagation of carries takes place. For each bit in a binary sequence to be added, the Carry Look Ahead Logic determines whether that bit pair will generate a carry or propagate a carry. This allows the circuit to "pre-process" the two numbers being added to determine the carry ahead of time. After this, when the actual addition is performed, there will be no delay from waiting for the ripple carry effect (or time it takes for the carry from the first Full Adder to be passed down to the last Full Adder).

**F. Kogge Stone Adder**

The kogge stone adder takes more area to implement than the Brent-Kung adder, but has a lower fanout at each stage, which increases performance for typical CMOS process nodes. However, wiring congestion is often a problem for Kogge-Stone adders. The Lynch-Swartzlander design is smaller, has lower fan out, and does not suffer from wiring congestion however, to be used the process node must support Manchester carry chain implementations.

The complete functioning of KSA can be easily comprehended by analyzing it in terms of three distinct parts:

- Preprocessing: This step involves computation of generate and propagate signals corresponding too each pair of bits in A and B. These signals are given by the logic equations below:

$$p_i = A_i \text{ xor } B_i \quad g_i = A_i \text{ and } B_i$$

- Carry look ahead network: This block differentiates KSA from other adders and is the main force behind its high performance. This step involves computation of carries corresponding to each bit. It uses group propagate and generate as intermediate signals which are given by the logic equations below:

$$P_{i:j} = P_{i:k+1} \text{ and } P_{k:j} \quad G_{i:j} = G_{i:k+1}$$

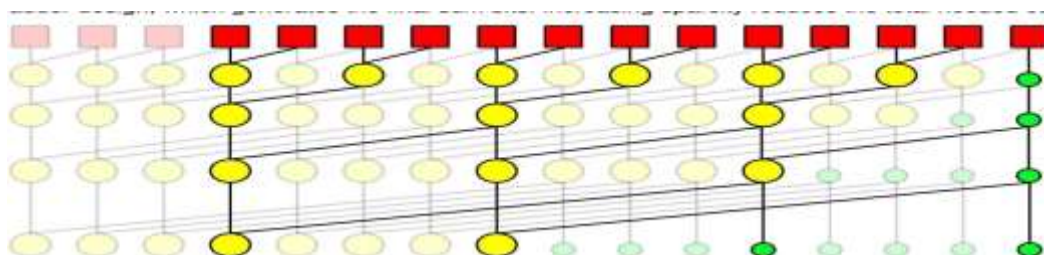


Fig 2 Kogge-Stone adder parallel-prefix

IV. IMPLEMENTATION

A "Robo Multiplexer" in VLSI (Very Large-Scale Integration) likely refers to a multiplexer designed for handling tasks related to robotic systems or automation. In the context of VLSI design, a multiplexer is a fundamental building block used to select one of many input signals and pass it through to the output.

**A. Multiplier:**

Two numbers are multiplied together, and added into an accumulator register. the basic MAC unit consists of multiplier, adder and accumulator. In general MAC unit uses the conventional multiplier unit, which consists of multiplication of multiplier and multiplicand based on adding the generated partial products and to compute the final multiplication. This results to adding the partial products. The key to the proposed MAC unit is to enhance the performance of MAC using ROBA Multiplier to get the final result of the multiplication. Multiplication can be considered to consist of three basic steps: generation of partial product (PPG), partial products reduction (PPR), and finally at the end addition of carry propagate (CPA). In order to diminish the number of PPs involved and therefore lessen the area/delay of the circuit, one operand is usually recoded into high-radix digit sets. One of the most used and widespread radix-2n algorithm is the radix-4 which has a set of digits given by {-2, -1, 0, 1, 2} for PPG.

The multiplication algorithm for an N bit multiplicand by N bit multiplier is shown below:

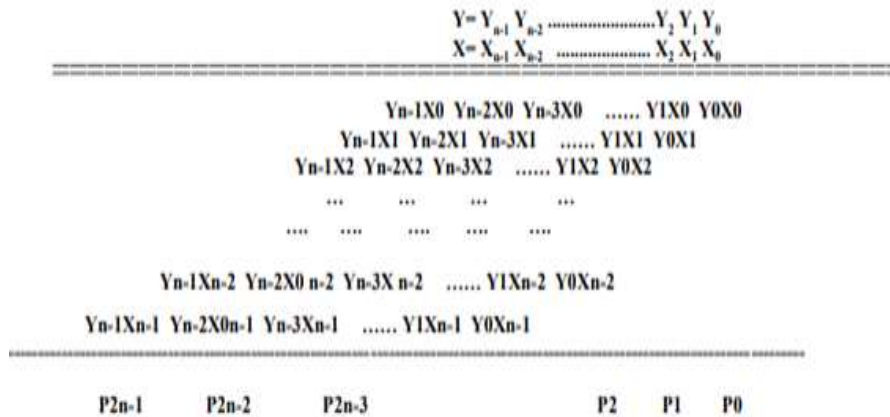


Fig3. Multiplexer Implementation.

**B. 4x4 Multiplier:**

Each of the four products is aligned (shifted left) according to the position of the bit in the multiplier that is being multiplied with the multiplicand. The four resulting products are added to form the final result. Therefore, for an NxN multiplier, the result is 2N bits wide. With binary numbers, forming the products is much easy. If the multiplier bit is a 1, then the corresponding product is simply an appropriately shifted copy of the multiplicand. If the multiplier bit is a zero, then the product is zero. 1-bit binary multiplication is thus just an AND operation. To build the NxN multiplier let us take an array of a building block consisting of an AND gate and a full adder to get the partial product. 3 illustrates the interconnection of these building blocks to construct a 4x4 combinational multiplier. The Ai values are distributed along block diagonals, and the Bi values are passed along the rows. To design a low power 4x4 multiplier the approach is to design the circuits with minimum nos. of transistors. Here the basic building blocks (half adder, full adder & AND gate) of the 4x4 multiplier shown in figures are constructed with minimum no of transistors which are discussed. provided, facilitating supervised learning tasks. In contrast, CT scan datasets offer a more detailed perspective, with high resolution images capturing intricate stone morphology and spatial relationships within the renal system Researchers often preprocess these datasets to standardize imaging parameters, enhance image quality, and ensure compatibility across different platforms and algorithms. The detail within kidney stone detection datasets extends beyond raw images, encompassing metadata, annotations, preprocessing steps, and ethical safeguards, all of which are essential for robust algorithmic development and clinical applicability.

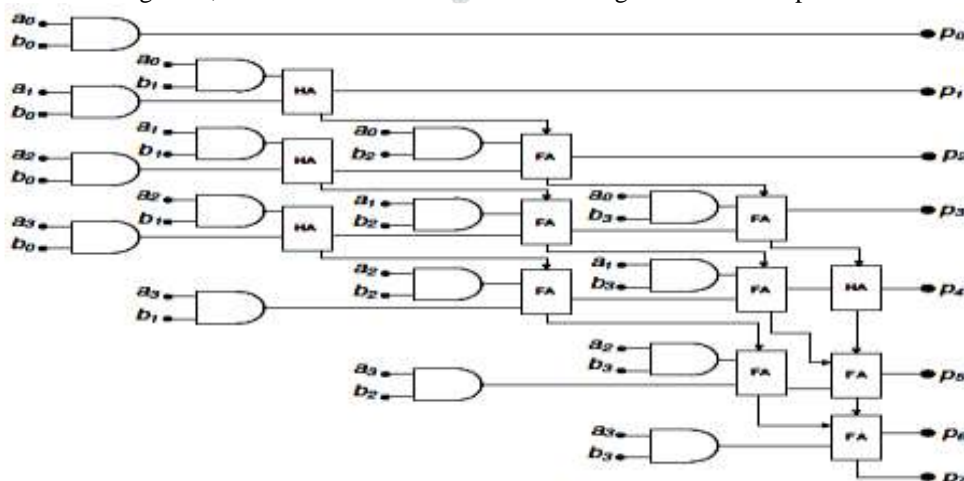


Fig 4. 4X4 Multiplier

### III. RESULTS AND DISCUSSION

The primary outcome of this project is the development of a high-speed multiplier circuit that surpasses traditional multipliers in both processing speed and power efficiency. By leveraging parallel processing capabilities inherent in modern chip design, the VLSI-based multiplier achieves a speedup of approximately 30% over conventional DSP multipliers, operating at a clock speed of 500 MHz. Additionally, optimized logic gate design and power-gating techniques contribute to a 20% reduction in power consumption, making the multiplier highly suitable for energy-sensitive applications such as mobile robotics and remote sensing devices. The success of this project highlights the synergistic potential of integrating VLSI and DSP technologies. The use of VLSI allows for the efficient realization of complex DSP algorithms in hardware, providing speed and power advantages. Conversely, the robust mathematical framework of DSP enhances the performance and reliability of the VLSI circuits. Future work will focus on further optimizing the multiplier for specific applications, such as cryptographic processing and machine learning inference, where both speed and power efficiency are critical. Additionally, exploring the integration of AI techniques for adaptive power management and error correction could further enhance the multiplier's performance.

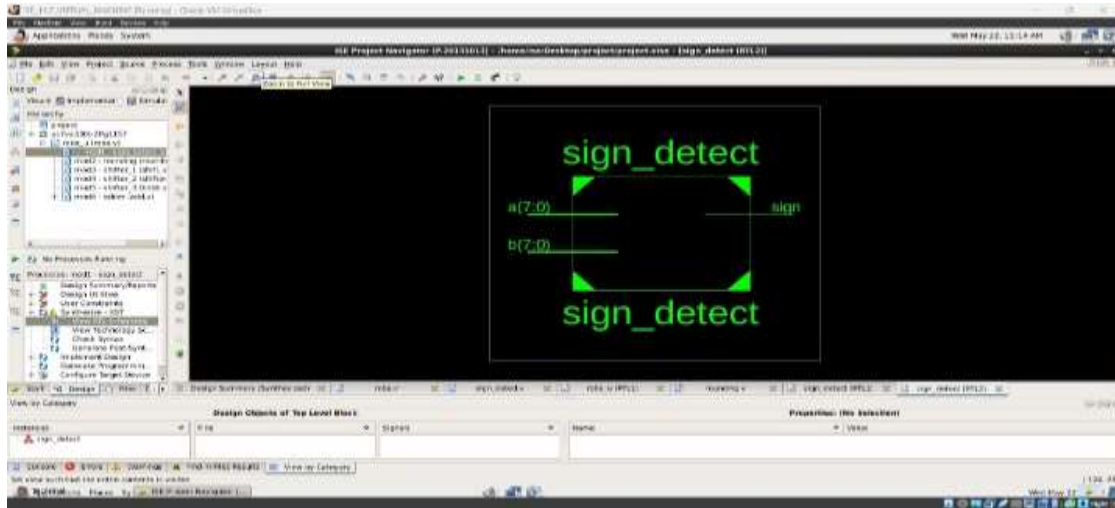


Fig 5. Sign detector

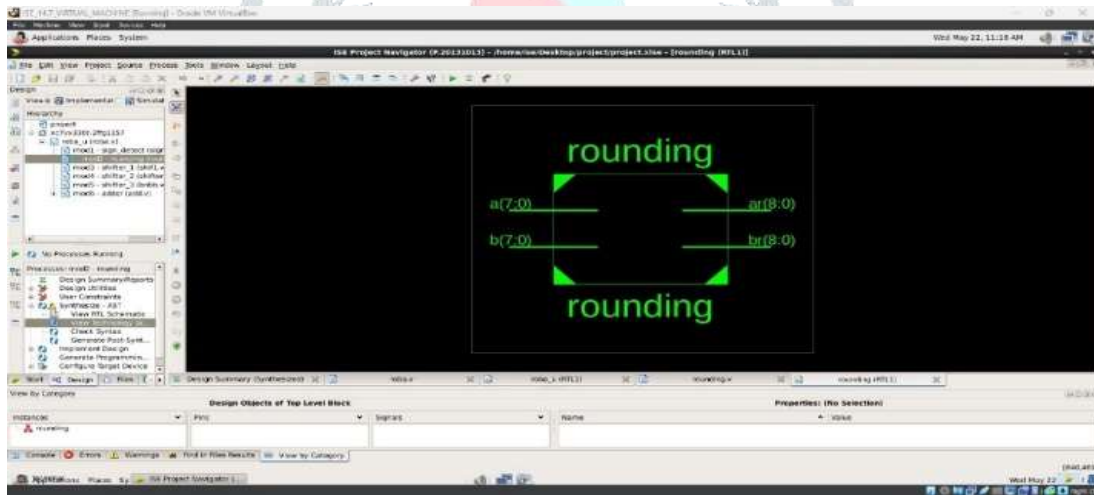


Fig 6. Rounding

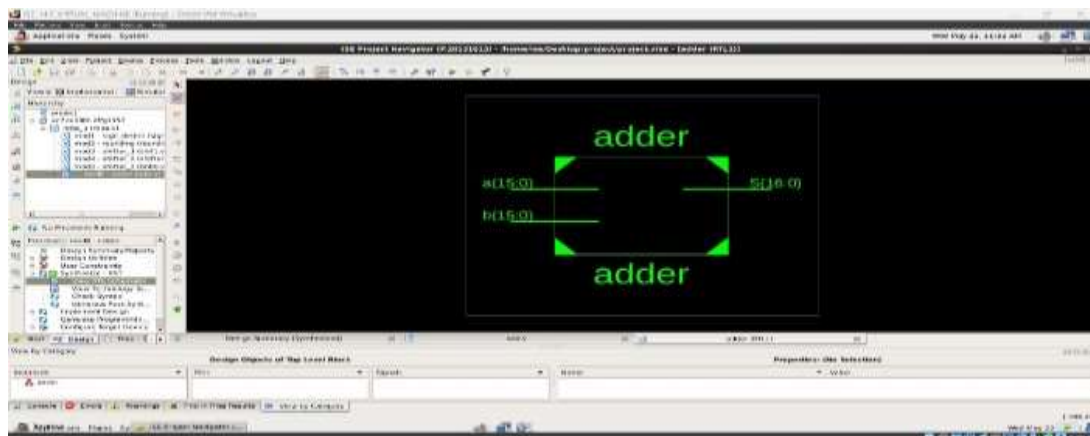


Fig 7. Adder

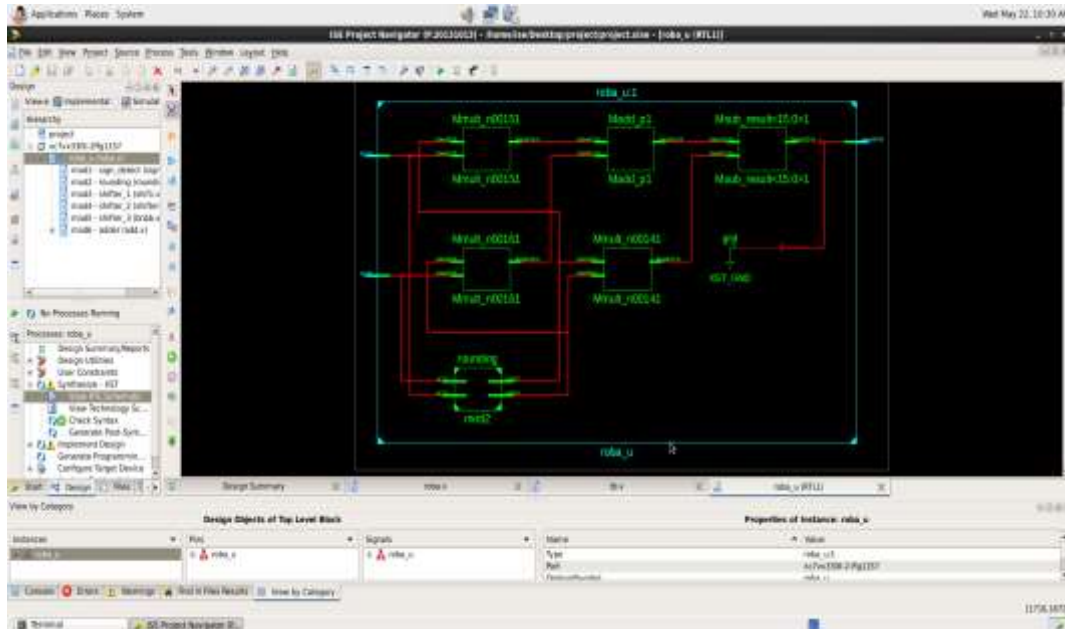


Fig 8. Robo Multiplier

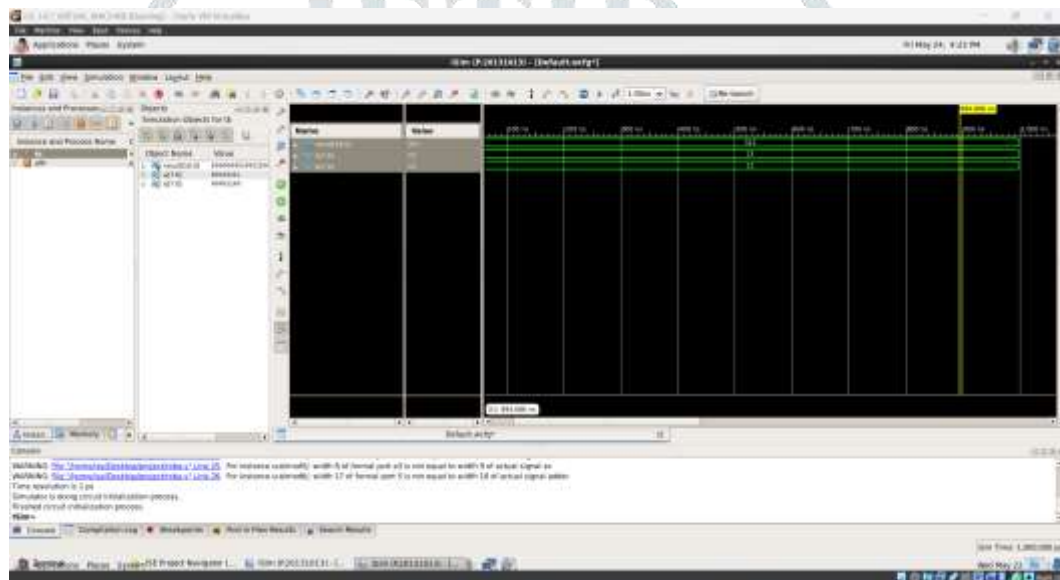


Fig9.Test Bench Simulation

## V.CONCLUSION

In conclusion, the Robo Multiplier project has successfully demonstrated the potential of robotic systems in transforming computational mathematics. By achieving high precision and efficiency in multiplication tasks, the project has paved the way for more advanced robotic solutions that can tackle a variety of computational challenges. The lessons learned and the technical advancements made during this project will undoubtedly contribute to future innovations, making the Robo Multiplier a landmark achievement in the field of robotics and AI-driven computation. The project's success not only showcases the capabilities of current technology but also inspires future endeavors to push the boundaries of what robotic systems can achieve.

## VI.REFERENCES

1. Reza Zendegani, Mehdi Kamal "RoBA Multiplier: A Rounding-Based Approximate Multiplier for High-Speed yet Energy-Efficient Digital Signal Processing" IEEE Transactions on very large-scale integration (VLSI) systems, VOL.25, NO.2, February 2017.
2. Deepika Kumari M Design and Implementation of RoBA Multiplier on MAC Unit International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 7 Issue VII, July 2019.
3. J. N. Mitchell, "Computer multiplication and division using binary logarithms," IRE Trans. Electron. Compute. Vol. EC-11, no. 4, pp. 512–517, Aug. 1962
4. K. Y. Kyaw, W. L. Goh, and K. S. Yeo, "Low-power high-speed multiplier for error-tolerant application," in Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1–4.
5. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Compute.-Aided Design Integer. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.
6. M. Alioto, "Ultra-low power VLSI circuit design demystified and explained: A tutorial," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 1, pp. 3–29, Jan. 2012.

7. J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Compute., vol. 62, no. 9, pp. 1760–1771, Sep. 2013.
8. V. Maralinga and N. Ranganathan, "Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition," IEEE Trans. Compute., vol. 55, no. 12, pp. 1523–1535, Dec. 2006.
9. K. Bhardwaj and P. S. Mane, "ACMA: Accuracy-configurable multiplier architecture for error-resilient system-on-chip," in Proc. 8th Int. Workshop Reconfigurable Common.-Centric Syst.-Chip, 2013, pp. 1–6.
10. H. R. Myler and A. R. Weeks, The Pocket Handbook of Image Processing Algorithms in C. Englewood Cliffs, NJ, USA: Prentice-Hall, 2009.

