IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 15, Issue 4, April 2025, ||Series -1|| PP 68-75

Implementation of High Speed Area Efficient Multiplier using 7:3 Compressors

P. NIHARIKA[1], G. JAGADISH[2], P. SAI DEVI[3], K. HIMA BINDU[4]
[1,2,3]B.Tech Scholars, Department of ECE,
[4]Assistant Professor, Department of ECE, Aditya College of Engineering and Technology, Surampalem, Kakinada, AP, India.

ABSTRACT
Our project, "Optimized Wallace Tree Multiplier Using High-Order Compressors," focuses on enhancing the performance of multipliers in digital systems and signal processing applications. Adders and multipliers play a major role in designing complex circuits, but traditional multipliers consume significant area, delay, and power. To improve multiplier efficiency, multiple architectures have been designed. In this project, a Wallace tree multiplier is proposed to enhance speed using compressors.

The key focus is on partial product reduction, which significantly impacts multiplier performance. Initially, 4:2 compressors are used for reduction. However, as the input bit length increases, complexity grows, making the multiplication process more time- consuming. To address this, multi-input compressors are used to further shorten partial products.

In the proposed design, a higher-order 7:3 compressor is incorporated to minimize delay, achieving better performance compared to previous multiplier architectures.

Keywords: Approximate computing, Compressors, Multiplier, Partial product reduction.

I. INTRODUCTION

Modern computing applications, such as Digital Signal Processing (DSP), machine learning, and image processing, extensively use multipliers. These applications require a large number of arithmetic operations, making multiplication a key component in computationally intensive tasks like Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), and Convolutional Neural Networks (CNNs). However, exact multipliers consume significant power and area while increasing computational delay, which affects system performance and efficiency. To address this challenge, researchers have proposed Approximate Multipliers that trade accuracy for reduced power consumption, area, and delay, making them suitable for error- tolerant applications.

Recent advancements in arithmetic optimization have demonstrated that approximate computing techniques can significantly improve system performance. By relaxing the requirement for exact results, Approximate Multipliers achieve lower power dissipation and smaller circuit complexity while maintaining acceptable accuracy. Several architectures have been designed to optimize multiplication operations by reducing the number of logic gates and critical path delay. The primary techniques for designing approximate multipliers include truncation, bit-width reduction, and error-resilient arithmetic modifications.

Although Approximate Multipliers introduce small computational errors, they are widely used in DSP accelerators, neuromorphic computing, and real-time image processing applications where perfect accuracy is unnecessary. Researchers have developed various approximation strategies to balance accuracy and efficiency, including approximate Booth encoding and fused add-multiply (FAM) units. Recent studies have explored different approximation techniques to optimize performance for specific applications, focusing on minimizing error while maximizing energy efficiency.

High-speed computing applications by optimizing the multiplication process through reduced logic complexity and efficient arithmetic encoding schemes. We analyze different Approximate Multiplier designs and compare them with conventional multipliers regarding performance metrics such as power consumption, area occupation, and delay. The goal is to provide an efficient multiplier design suitable for error-resilient applications while achieving significant gains in computational efficiency.

II. Related Work

Approximate multipliers have gained significant attention in modern computing due to their ability to trade accuracy for efficiency. Multiplication is a fundamental operation in digital systems, but conventional multipliers require substantial hardware resources, leading to high power consumption, increased circuit area, and longer computational delays. In applications like image processing, machine learning, and signal processing, where minor errors are tolerable, approximate computing provides a promising alternative. By allowing small errors in calculations, approximate multipliers enhance energy efficiency and improve processing speed, making them suitable for low- power devices such as smartphones, embedded systems, and Internet-of-Things (IoT) applications.

The need for energy-efficient computing has grown due to the increasing use of battery- operated devices. Traditional multipliers consume significant power due to their complex arithmetic operations. Approximate computing provides an effective solution by relaxing accuracy constraints while improving performance. Many real-world applications, such as deep learning and multimedia processing, naturally tolerate minor computational errors. Approximate multipliers help reduce power consumption, minimize delay, and lower hardware complexity, making them an attractive choice for energy- efficient systems. Researchers have explored various approximation techniques to achieve a balance between accuracy and efficiency.

Several techniques have been proposed in literature to design approximate multipliers. Truncated multiplication is one of the most common approaches, where the least significant bits of the operands or intermediate results are discarded. While this introduces minor errors, it significantly reduces power consumption and hardware overhead. Another approach involves reducing bit-width, which lowers computational complexity and speeds up processing while sacrificing some accuracy. Approximate partial product generation is another widely used technique, where complex additions are simplified using logical operations such as AND gates instead of full adders. Some critical computations. These methods ensure that approximate multipliers remain highly efficient while maintaining acceptable accuracy levels.

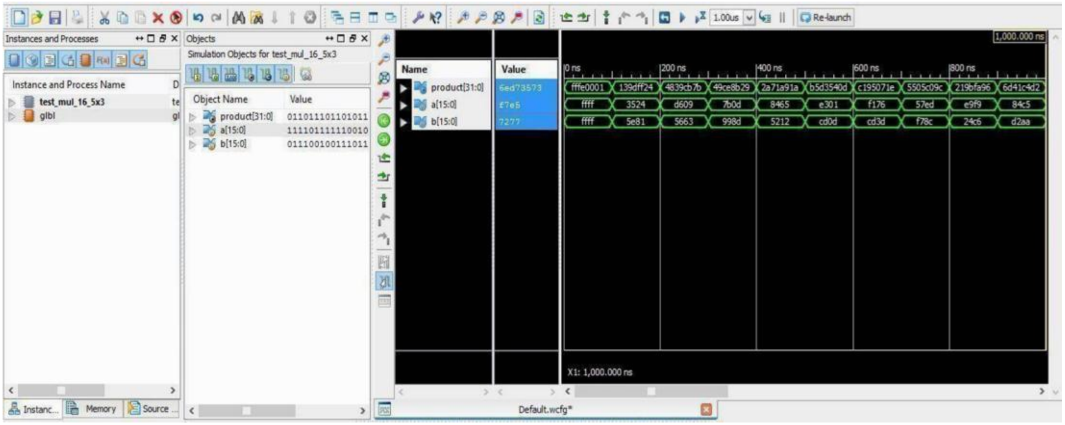
III. PROJECT DESCRIPTION

Multiplication is one of the most fundamental arithmetic operations in digital computing and plays a crucial role in various applications such as digital signal processing (DSP), machine learning, image processing, and cryptographic systems. Many computationally intensive tasks, including Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), and neural network computations, rely heavily on multipliers. However, traditional multiplier architectures suffer from several challenges, including high power consumption, increased delay due to long carry propagation, and large area occupation in hardware implementations such as FPGA and ASIC designs.

One of the major limitations of conventional multipliers, such as the array multiplier or the Wallace tree multiplier using 4:2 compressors, is the long carry chain, which increases the critical path delay (CPD). The ripple carry effect in addition stages further slows down the multiplication process, making it inefficient for high-speed applications. To address these issues, researchers have explored approximate computing and optimized compressor-based architectures that trade off minimal accuracy for significant gains in speed and power efficiency.

In this work, a high-speed, area-efficient multiplier is proposed using 7:3 compressors to improve the partial product reduction process. Unlike conventional designs that use 4:2 compressors, the 7:3 compressor reduces the number of addition stages, thereby minimizing power consumption and computational delay. Additionally, a Wallace tree structure is used for efficient partial product reduction, along with Carry- Save Adders (CSA) to further enhance performance. By optimizing the design with 6- input LUT configurations in modern FPGAs such as Xilinx Virtex-7 and Zynq series, the proposed multiplier achieves significant improvements in speed, power, and area utilization.

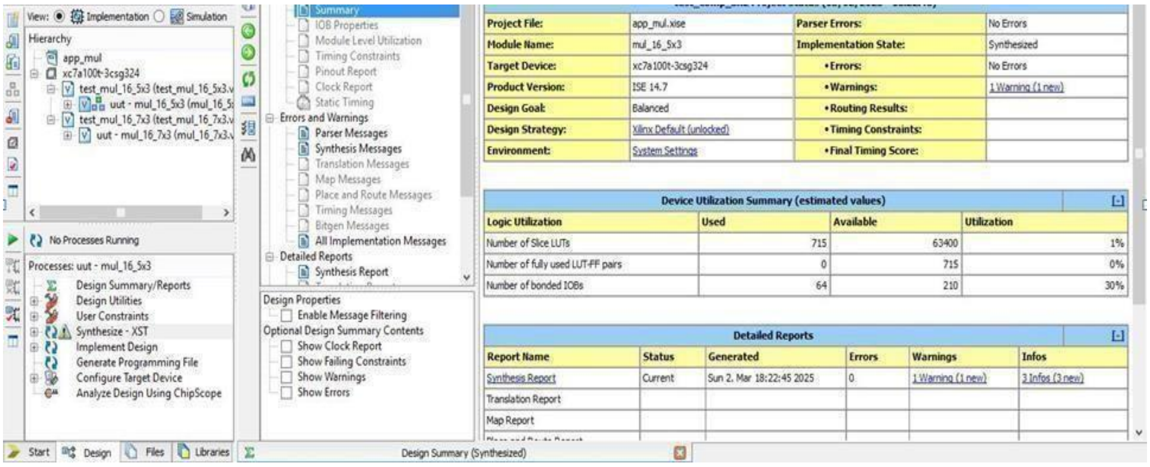
IV. SIMULATION RESULTS



Object Name	Value
product[31:0]	011011101101011
a[15:0]	111101111110010
b[15:0]	011100100111011

```
Name           Value
product[31:0]  6ed73573  fffe0001  139dff24  4839cb7b  49ce8b29  2a71a91a  b5d3540d  c195071e  5505c09c  219bfa96  6d41c4d2
a[15:0]        f7e5      ffff      3524      d609      7b0d      8465      e301      f176      57ed      e9f9      84c5
b[15:0]        7277      ffff      5e81      5663      998d      5212      cd0d      cd3d      f78c      24c6      d2aa
```

Figure. Simulation result of the Proposed16-bit multiplier using 5-3 compressor



Project File:	app_mul.xise	Parser Errors:	No Errors
Module Name:	mul_16_5x3	Implementation State:	Synthesized
Target Device:	xc7a100t-3csg324	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	1 Warning (1 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

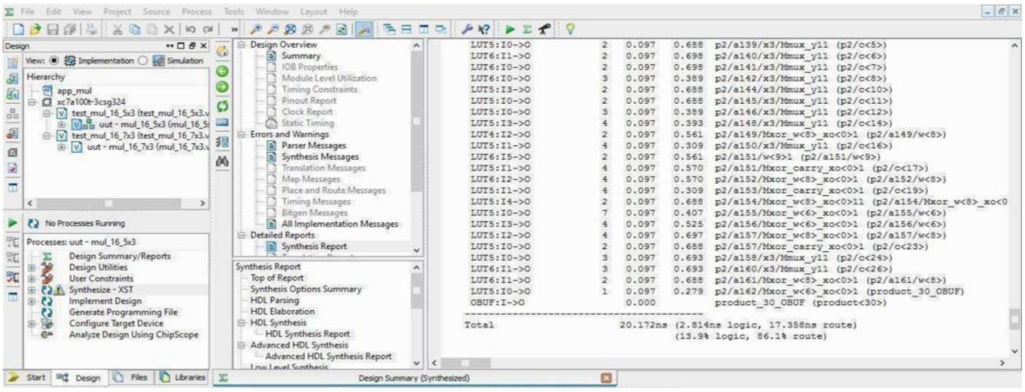
Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	715	63400	1%
Number of fully used LUT-FF pairs	0	715	0%
Number of bonded IOBs	64	210	30%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Sun 2. Mar 18:22:45 2025	0	1 Warning (1 new)	3 Infos (3 new)
Translation Report					
Map Report					

Figure: Summary report of the Proposed16-bit multiplier using 5-3 compressor



```
    LUT5:I0->O            2   0.097   0.688  p2/a139/x3/Mmux_y11 (p2/c<5>)
    LUT6:I1->O            2   0.097   0.698  p2/a140/x3/Mmux_y11 (p2/c<6>)
    LUT6:I0->O            2   0.097   0.698  p2/a141/x3/Mmux_y11 (p2/c<7>)
    LUT6:I0->O            3   0.097   0.389  p2/a142/x3/Mmux_y11 (p2/c<8>)
    LUT5:I3->O            2   0.097   0.688  p2/a144/x3/Mmux_y11 (p2/c<10>)
    LUT5:I0->O            2   0.097   0.688  p2/a145/x3/Mmux_y11 (p2/c<11>)
    LUT5:I0->O            3   0.097   0.389  p2/a146/x3/Mmux_y11 (p2/c<12>)
    LUT5:I3->O            4   0.097   0.393  p2/a148/x3/Mmux_y11 (p2/c<14>)
    LUT4:I2->O            2   0.097   0.561  p2/a149/Mxor_w<8>_xo<0>1 (p2/a149/w<8>)
    LUT5:I1->O            4   0.097   0.309  p2/a150/x3/Mmux_y11 (p2/c<16>)
    LUT6:I5->O            2   0.097   0.561  p2/a151/w<9>1 (p2/a151/w<9>)
    LUT5:I1->O            4   0.097   0.570  p2/a151/Mxor_carry_xo<0>1 (p2/c<17>)
    LUT6:I2->O            4   0.097   0.570  p2/a152/Mxor_w<8>_xo<0>1 (p2/a152/w<8>)
    LUT5:I1->O            4   0.097   0.309  p2/a153/Mxor_carry_xo<0>1 (p2/c<19>)
    LUT5:I4->O            2   0.097   0.688  p2/a154/Mxor_w<8>_xo<0>11 (p2/a154/Mxor_w<8>_xo<0
    LUT5:I0->O            7   0.097   0.407  p2/a155/Mxor_w<6>_xo<0>1 (p2/a155/w<6>)
    LUT5:I3->O            4   0.097   0.525  p2/a156/Mxor_w<6>_xo<0>1 (p2/a156/w<6>)
    LUT5:I2->O            4   0.097   0.697  p2/a157/Mxor_w<8>_xo<0>1 (p2/a157/w<8>)
    LUT5:I0->O            2   0.097   0.688  p2/a157/Mxor_carry_xo<0>1 (p2/c<23>)
    LUT5:I0->O            3   0.097   0.693  p2/a158/x3/Mmux_y11 (p2/c<24>)
    LUT6:I1->O            3   0.097   0.693  p2/a160/x3/Mmux_y11 (p2/c<26>)
    LUT6:I1->O            2   0.097   0.688  p2/a161/Mxor_w<8>_xo<0>1 (p2/a161/w<8>)
    LUT5:I0->O            1   0.097   0.279  p2/a162/Mxor_w<6>_xo<0>1 (product_30_OBUF)
    OBUF:I->O                 0.000          product_30_OBUF (product<30>)
   ----------------------------------------
   Total                     20.172ns (2.814ns logic, 17.358ns route)
                                      (13.9% logic, 86.1% route)
```

Figure: Delay report of the 16-bit multiplier using 5-3 compressor

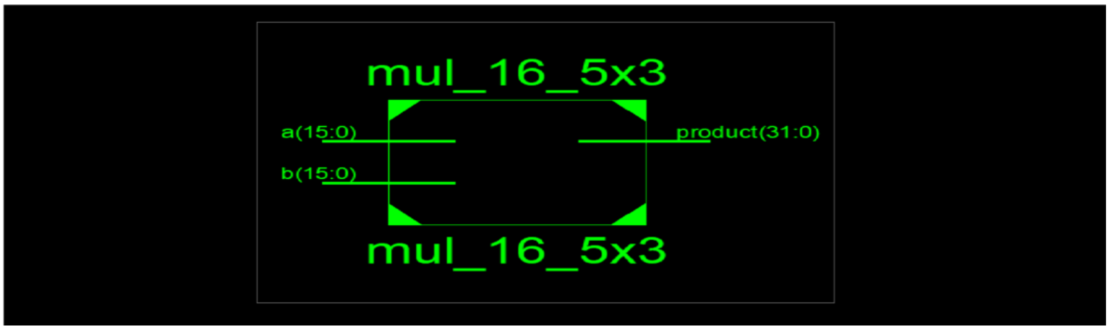


Figure: Block Diagram of 16-bit multiplier using 5-3 compressor

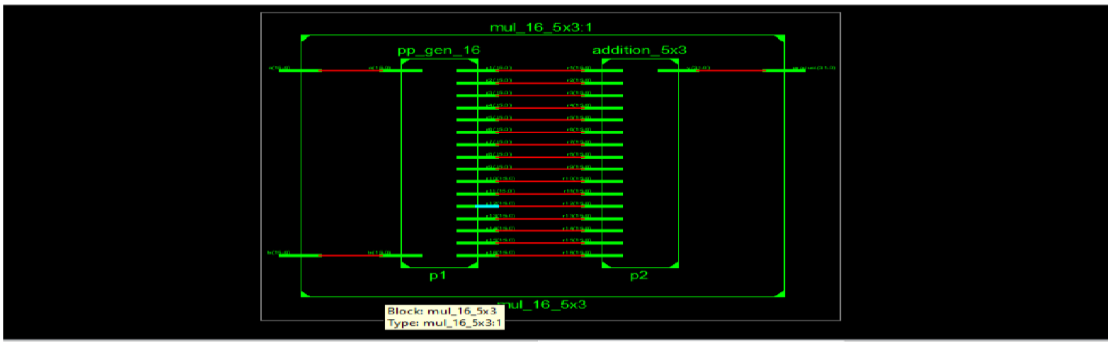


Figure: Internal Structure of Block Diagram of 16-bit multiplier using 5-3 compressor

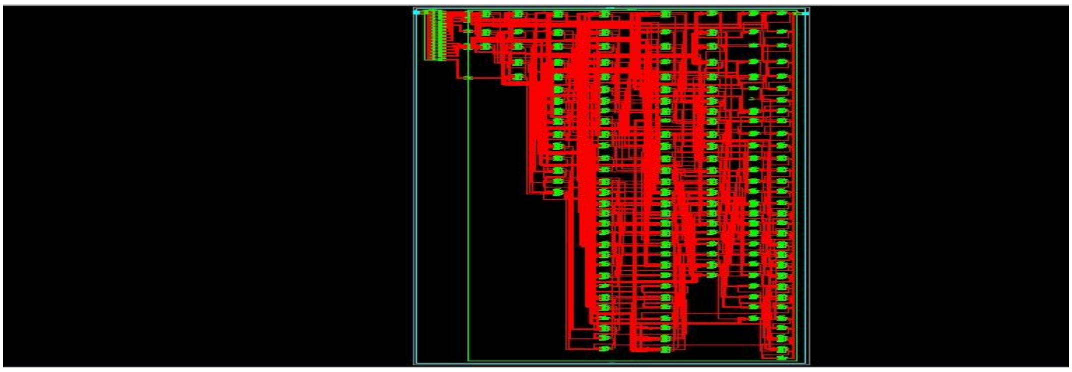


Figure: RTL schematic of the 16-bit multiplier using 5-3 compressor

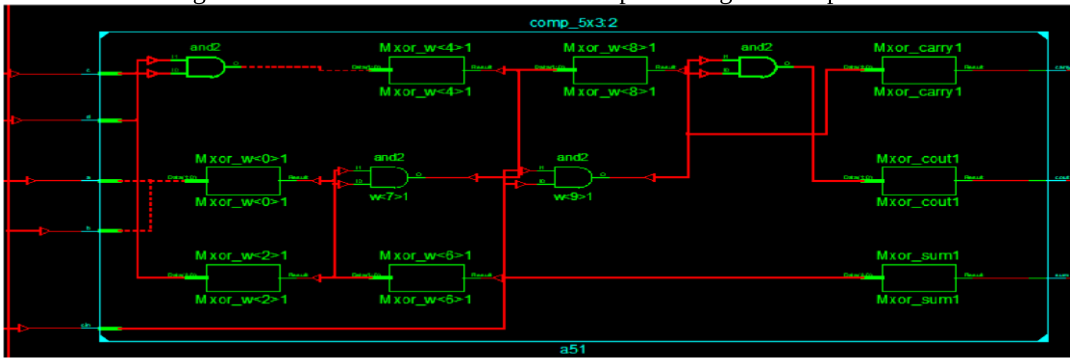


Figure: RTL schematic of the 16-bit multiplier using 5-3 compressor

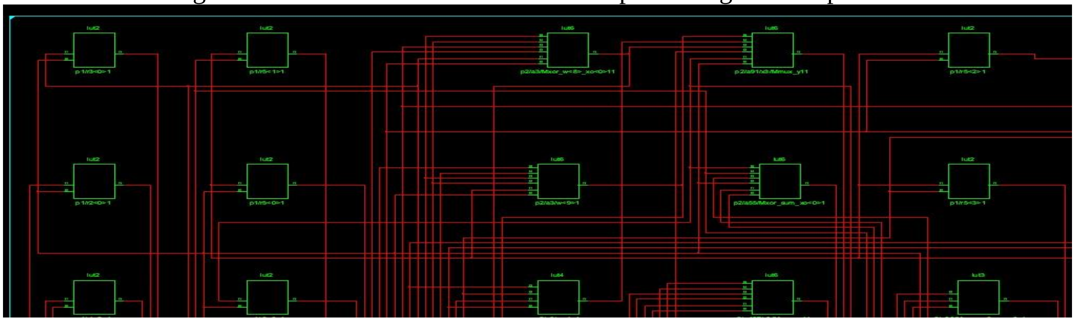


Figure: Technology schematic of the 16-bit multiplier using 5-3 compressor

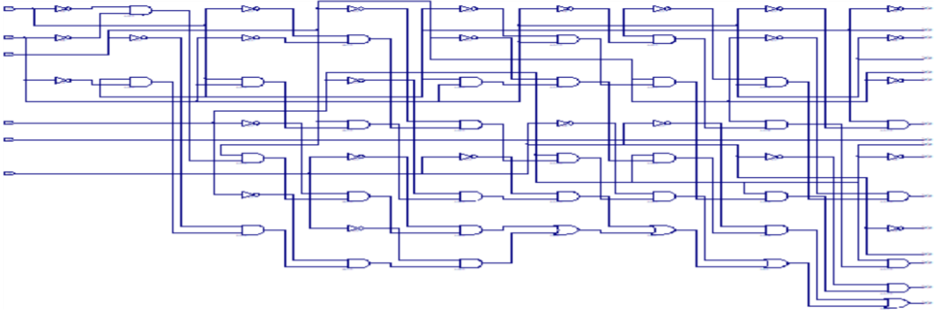


Figure: Internal Block Diagram of LUT6 of 16-bit multiplier using 5-3 compressor

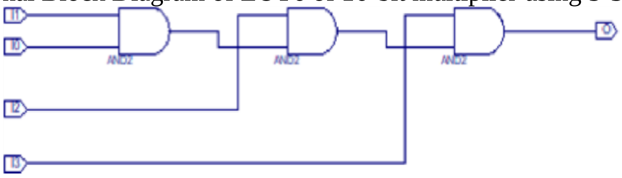


Figure: Internal Block Diagram of LUT4 of 16-bit multiplier using 5-3 compressor

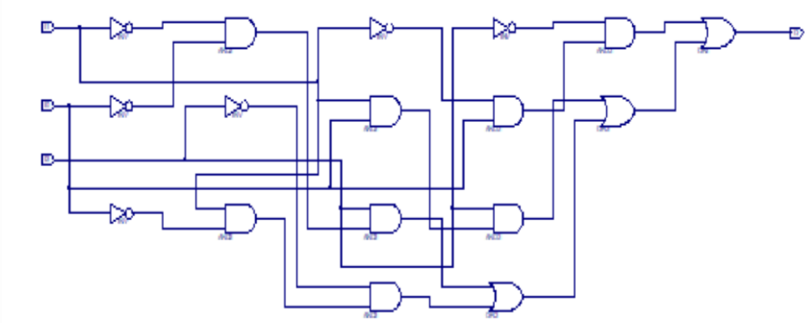


Figure: Internal Block Diagram of LUT3 of 16-bit multiplier using 5-3 compressor

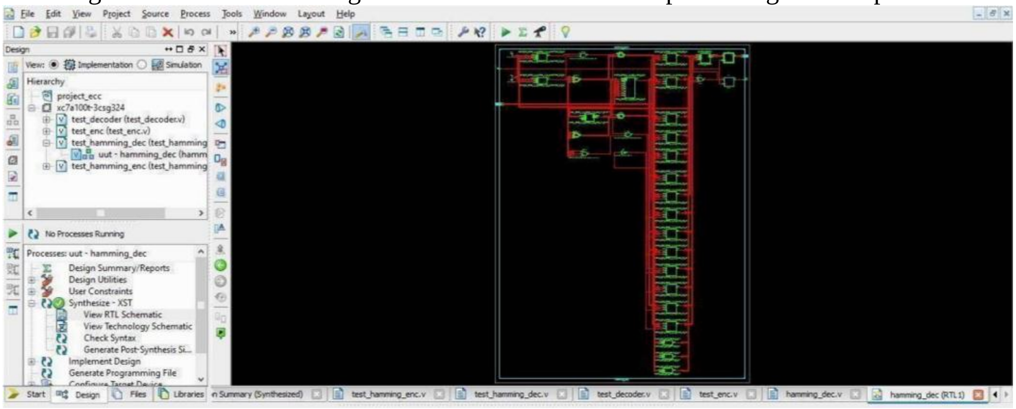


Figure: RTL schematic of the 16 bit multiplier 5-3 compressor

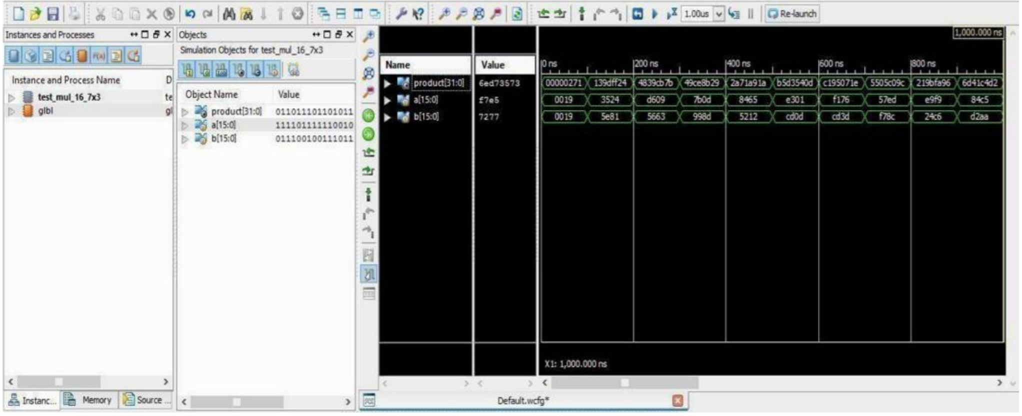
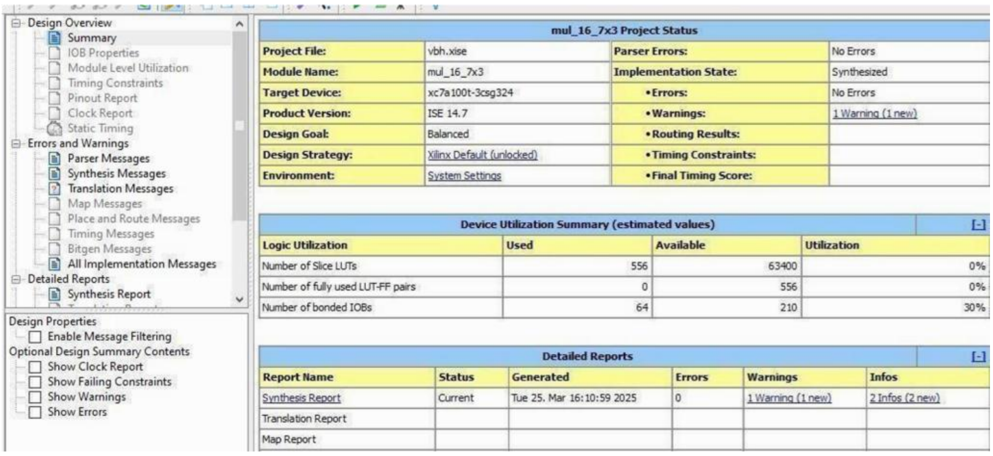


Figure: Simulation result of the proposed 16-bit multiplier using 7-3 compressor



mul_16_7x3 Project Status			
Project File:	vbh.xise	**Parser Errors:**	No Errors
Module Name:	mul_16_7x3	**Implementation State:**	Synthesized
Target Device:	xc7a100t-3csg324	**•Errors:**	No Errors
Product Version:	ISE 14.7	**•Warnings:**	1 Warning (1 new)
Design Goal:	Balanced	**•Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	
Environment:	System Settings	**•Final Timing Score:**	

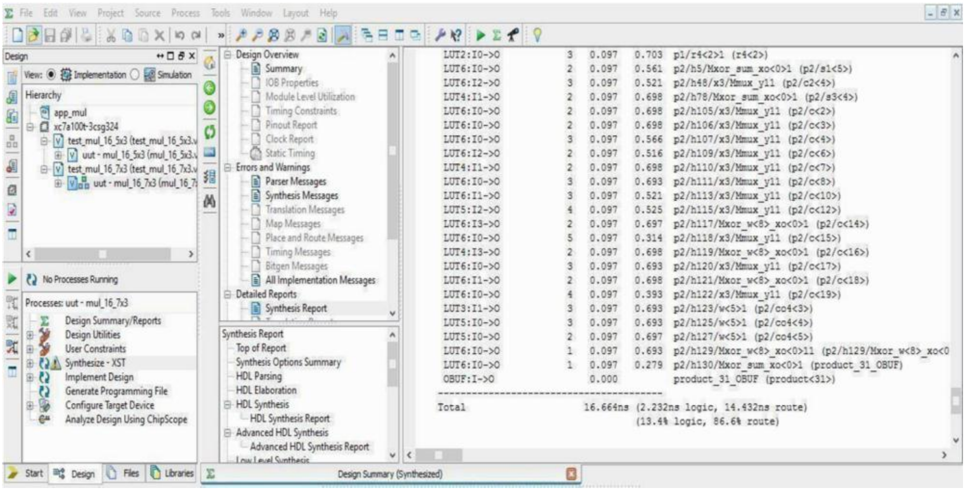
Device Utilization Summary (estimated values) [-]

Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	556	63400	0%
Number of fully used LUT-FF pairs	0	556	0%
Number of bonded IOBs	64	210	30%

Detailed Reports [-]

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Tue 25. Mar 16:10:59 2025	0	1 Warning (1 new)	2 Infos (2 new)
Translation Report					
Map Report					

Figure: Summary report of the proposed16-bit multiplier using 7-3 compressor



```
LUT2:I0->O            3   0.097   0.703  p1/r4<2>1 (r4<2>)
LUT6:I0->O            2   0.097   0.561  p2/h5/Mxor_sum_xo<0>1 (p2/s1<5>)
LUT6:I2->O            3   0.097   0.521  p2/h48/x3/Mmux_y11 (p2/c2<4>)
LUT4:I1->O            2   0.097   0.698  p2/h78/Mxor_sum_xo<0>1 (p2/s3<4>)
LUT6:I0->O            2   0.097   0.698  p2/h105/x3/Mmux_y11 (p2/c<2>)
LUT6:I0->O            2   0.097   0.698  p2/h106/x3/Mmux_y11 (p2/c<3>)
LUT6:I0->O            3   0.097   0.566  p2/h107/x3/Mmux_y11 (p2/c<4>)
LUT6:I2->O            2   0.097   0.516  p2/h109/x3/Mmux_y11 (p2/c<6>)
LUT4:I1->O            2   0.097   0.698  p2/h110/x3/Mmux_y11 (p2/c<7>)
LUT6:I0->O            3   0.097   0.693  p2/h111/x3/Mmux_y11 (p2/c<8>)
LUT6:I1->O            3   0.097   0.521  p2/h113/x3/Mmux_y11 (p2/c<10>)
LUT5:I2->O            4   0.097   0.525  p2/h115/x3/Mmux_y11 (p2/c<12>)
LUT6:I3->O            2   0.097   0.697  p2/h117/Mxor_w<8>_xo<0>1 (p2/c<14>)
LUT6:I0->O            5   0.097   0.314  p2/h118/x3/Mmux_y11 (p2/c<15>)
LUT4:I3->O            2   0.097   0.698  p2/h119/Mxor_w<8>_xo<0>1 (p2/c<16>)
LUT6:I0->O            3   0.097   0.693  p2/h120/x3/Mmux_y11 (p2/c<17>)
LUT6:I1->O            2   0.097   0.698  p2/h121/Mxor_w<8>_xo<0>1 (p2/c<18>)
LUT6:I0->O            4   0.097   0.393  p2/h122/x3/Mmux_y11 (p2/c<19>)
LUT3:I1->O            3   0.097   0.693  p2/h123/w<5>1 (p2/co4<3>)
LUT5:I0->O            3   0.097   0.693  p2/h125/w<5>1 (p2/co4<4>)
LUT5:I0->O            2   0.097   0.697  p2/h127/w<5>1 (p2/co4<5>)
LUT6:I0->O            1   0.097   0.693  p2/h129/Mxor_w<8>_xo<0>11 (p2/h129/Mxor_w<8>_xo<0
LUT6:I0->O            1   0.097   0.279  p2/h130/Mxor_sum_xo<0>1 (product_31_OBUF)
OBUF:I->O                 0.000          product_31_OBUF (product<31>)
----------------------------------------
Total                    16.664ns (2.232ns logic, 14.432ns route)
                                  (13.4% logic, 86.6% route)
```

Figure. Delay report of the proposed16-bit multiplier using 7-3 compressor

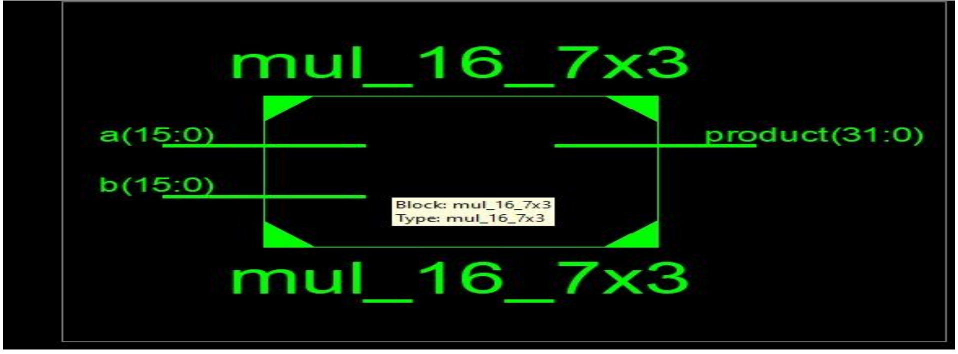


Figure. Block Diagram of 16-bit multiplier using 7-3 compressor

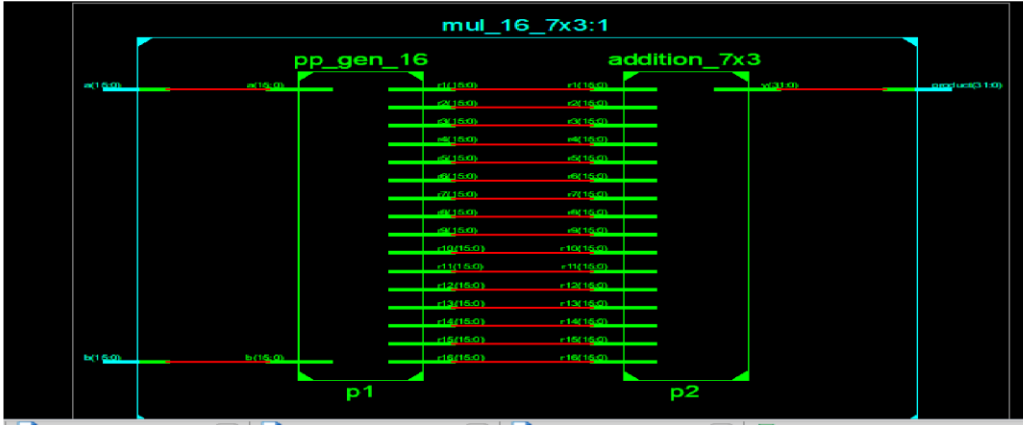


Figure: Internal Block Diagram of 16-bit multiplier using 7-3 compressor

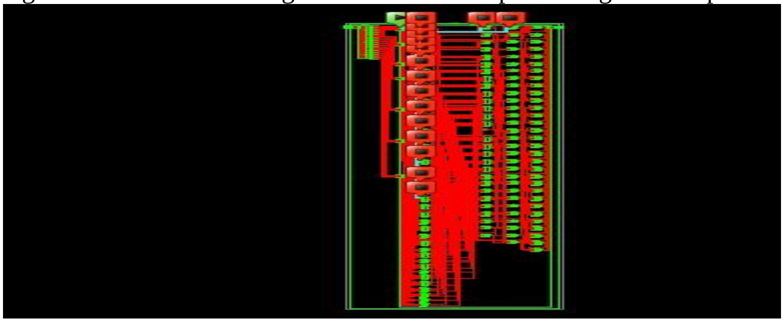


Figure: RTL schematic of the 16-bit multiplier using 7-3 compressor

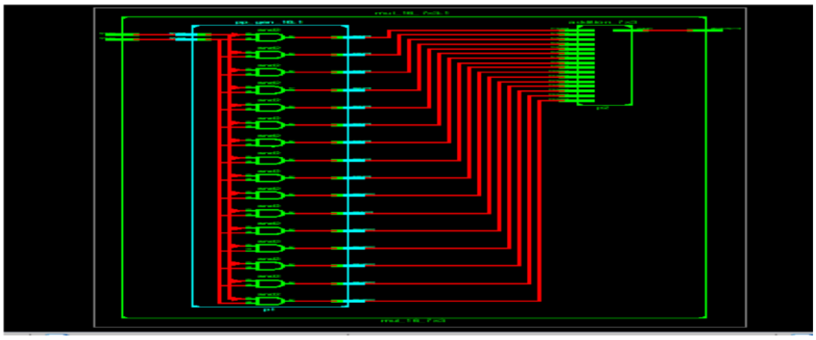
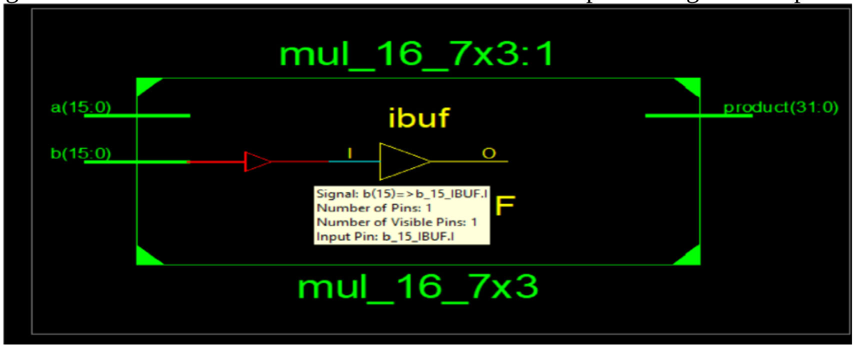
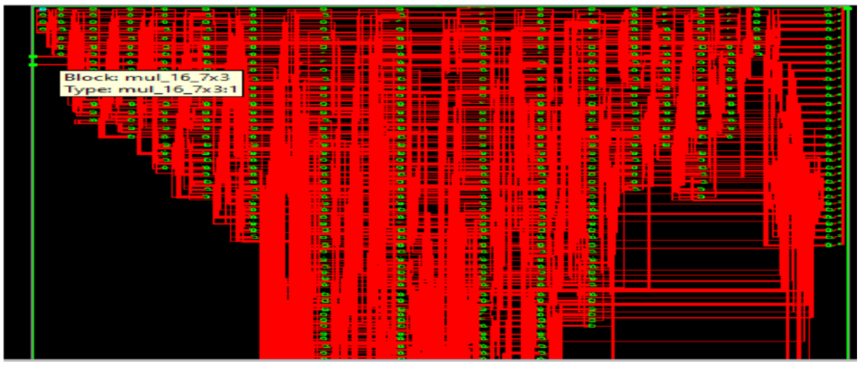


Figure: Internal View RTL schematic of the 16-bit multiplier using 7-3 compressor



Signal: b(15)=>b_15_IBUF.I
Number of Pins: 1
Number of Visible Pins: 1
Input Pin: b_15_IBUF.I

Figure: Technology schematic of 16 bit multiplier 7-3 compressor



Block: mul_16_7x3
Type: mul_16_7x3:1

V. CONCLUSION

In this paper, The proposed energy-efficient approximate multiplier effectively balances accuracy, area, and power-delay-product (PDP) requirements, making it suitable for error- resilient applications. By using 7:3 compressors, along with Wallace Tree Multiplier,the design demonstrates significant improvements in efficiency. The -higher order compressors reduce the circuit area and power consumption while maintaining a low error rate. The elimination of a resource-intensive error recovery module further enhances system performance. This innovative approach provides a viable alternative to conventional multipliers, especially in applications where slight errors are acceptable in exchange for improved resource efficiency.

REFERRENCES

[1] W . Liu, C. Gu, M. O'Neill, G. Qu, P. Montuschi, and F. Lombardi. Security in Approximate Computing and Approximate Computing for Security: Challenges and Opportunities. Proceedings of the IEEE, 2020.

[2] W. Liu, F. Lombardi, and M. Schulte. Approximate Computing: From Circuits to Applications [Scanning the Issue]. Proceedings of the IEEE, 2020.

[3] A. Momeni, J. Han, P. Montuschi, and F. Lombardi. Design and Analysis of Approximate Compressors for Multiplication. IEEE Trans. on Computers (TC), 2015.

[4] S. Venkatachalam and S. Ko. Design of Power and Area Efficient Approximate Multipliers. IEEE Trans. on Very Large-Scale Integration Systems (TVLSI), 2017.

[5] M. Ha and S. Lee. Multipliers With Approximate 4–2 Compressors and Error Recovery Modules. IEEE Embedded Systems Letters (ESL), 2018.

[6] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram. Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers. IEEE Trans. on Very Large-Scale Integration Systems (TVLSI), 2017.

[7] F. Sabetzadeh, M.H. Moaiyeri, and M. Ahmadinejad. A Majority-Based Imprecise Multiplier for Ultra-Efficient Approximate Image Multiplication. IEEE Trans. on Circuits and Systems I (TCAS I), 2019.

[8] M. Ahmadinejad, M.H. Moaiyeri, and F. Sabetzadeh. Energy and Area Efficient Imprecise Compressors for Approximate Multiplication at Nanoscale. AEU - International Journal of Electronics and Communications, 2019.

[9] A. Strollo, E. Napoli, D. Caro, N. Petra, and G. Meo. Comparison and Extension of Approximate 4-2 Compressors for Low-Power Approximate Multipliers. IEEE Trans. on Circuits and Systems I (TCAS I), 2020.

[10] N. Shiri, A. Sadeghi, and M. Rafiee. High-efficient and error-resilient gate diffusion input