iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72209

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Design and Implementation for Compressor

Sai Irikedandla[1], Madhuri Vemuluri[2]
[1]*M-Tech Scholar department of Electronics and Communication Engineering, Newtons Institute of Engineering, Macherla-Andrapradesh, India*
[2] *Associate professor department of Electronics and Communication Engineering, Newtons Institute of Engineering, Macherla-Andrapradesh, India*

Abstract: ***Approximate multipliers attract a large interest in the scientific literature that proposes several circuits built with approximate 4-2 compressors. Due to the large number of proposed solutions, the designer who wishes to use an approximate 4-2 compressor is faced with the problem of selecting the right topology. In this project, we present a comprehensive survey and comparison of approximate 4-2 compressors previously proposed in literature. We present also a novel approximate compressor, so that a total of twelve different approximate 4-2 compressors are analyzed. The investigated circuits are employed to design 8 × 8 and 16× 16 multipliers, implemented in CMOS technology. For each operand size we analyze two multiplier configurations, with different levels of approximations, both signed and unsigned. Our study highlights that there is no unique winning approximate compressor topology since the best solution depends on the required precision, on the signedness of the multiplier and on the considered error metric.***

Keywords: 4-2 Compressors, Design 8 × 8 and 16 × 16 multipliers, CMOS technology, Error metric.

I. INTRODUCTIONS

Approximate computing has emerged as an energy-efficient design paradigm for error- resilient applications, where strict accuracy is not mandatory. By trading off computational precision for reduced power consumption and improved performance, approximate circuits enable significant energy savings in multimedia, machine learning, and digital signal processing systems. Among these circuits, approximate multipliers play a crucial role due to their frequent use in arithmetic operations, and their optimization can lead to substantial power reductions without severely compromising output quality.

A key component in multiplier design is the compressor, which reduces partial product bits efficiently. The 4-2 compressor, widely used in high-speed multipliers, adds four bits of the same significance along with a carry-in, producing a sum and carry-out. Approximate versions of these compressors simplify logic operations, reducing transistor count and power dissipation while introducing controlled errors. Recent research has focused on designing energy-efficient 4-2 compressors that maintain acceptable accuracy levels, making them suitable for low-power approximate multipliers.

This work explores various approximate 4-2 compressor designs and their integration into multipliers to achieve optimal power-accuracy trade-offs. By analyzing different approximation strategies, we identify architectures that minimize energy consumption while keeping error metrics within application-specific limits. The findings contribute to the development of energy-efficient multipliers for error-tolerant computing, enabling sustainable and high-performance digital systems. A widely used compressor in multiplier architectures is the 4-2 compressor, which takes four input bits of the same significance along with a carry-in and produces a sum and carry-out. By efficiently compressing partial products, these structures minimize the critical path delay in multiplier designs, particularly in tree-based multipliers like Wallace and Dadda trees. The optimized logic of 4-2 compressors ensures that intermediate additions are performed with minimal overhead, enhancing both speed and energy efficiency. Their ability to balance carry propagation makes them a preferred choice for high-speed multipliers in digital signal processors and AI accelerators. Beyond traditional exact multipliers, compressors also play a vital role in approximate multipliers, where controlled errors are introduced to save power and area. Approximate compressors simplify logic gates, reducing transistor count and dynamic power consumption while maintaining acceptable accuracy for error-resilient applications. By leveraging these optimized compressors, designers can build energy-efficient multipliers for use in image processing, neural networks, and other domains where minor computational inaccuracies are tolerable. Thus, compressors serve as a fundamental building block in both precise and approximate multipliers, enabling a balance between performance, power, and precision.

To validate the efficiency of low-power compressors, ASIC implementations are typically tested using standard cell libraries in tools like Cadence Innovus or Synopsys Design Compiler. Power analysis, performed through post-layout simulations, helps compare different compressor designs in terms of energy per operation, delay, and area.

Recent advancements in FinFET and FD-SOI technologies further enhance power efficiency by reducing leakage currents and improving switching characteristics. As a result, modern ASIC implementations of compressors enable the development of ultra-low-power multipliers, making them ideal for battery-operated devices and energy-constrained computing systems.

A. Objectives of the Work

This research aims to design and analyze energy-efficient compressor architectures for use in low-power multipliers, targeting optimal trade-offs between power consumption, computational speed, and acceptable error margins. The study will explore innovative circuit- level optimizations and approximation techniques to reduce dynamic and static power dissipation while maintaining sufficient accuracy for error-resilient applications. Key objectives include developing standardized evaluation metrics for comparing compressor designs, validating performance through ASIC implementations, and identifying ideal use cases in AI accelerators, DSP systems, and mobile computing platforms where power efficiency is critical. The findings will contribute to advancing energy-conscious arithmetic unit designs for next-generation embedded and portable devices.

- To identify the low power logical circuits for the construction of low power compressor
- To develop the hardware description language for functional verification of the design.
- To synthesize the complete architecture using Genus tool and generate the optimized reports such as power, area, timing.
- To compare the performance of the design with existing designs.

II. METHODOLOGY OF THE WORK

The methodology adopted in this work employs a structured design exploration to develop and evaluate low-power compressor architectures for energy-efficient multipliers. Initial phase involves a critical analysis of existing compressor designs, identifying their limitations in power consumption, delay, and area efficiency. Circuit-level optimizations are then applied, including logic simplification, transistor sizing, and alternative logic styles, to minimize dynamic and leakage power without compromising critical path performance. The proposed designs are rigorously modeled at the gate and transistor levels to assess their theoretical power-delay product and error characteristics.

Validation is carried out through extensive simulations using industry-standard EDA tools, with performance benchmarks compared against conventional compressors. A novel evaluation framework is introduced, incorporating standardized metrics for power efficiency, computational accuracy, and hardware overhead. The methodology further investigates the integration of optimized compressors into multiplier architectures, analyzing their impact on overall system performance in practical applications like AI acceleration and digital signal processing. This systematic approach ensures comprehensive characterization of the proposed low-power solutions across multiple design parameters.

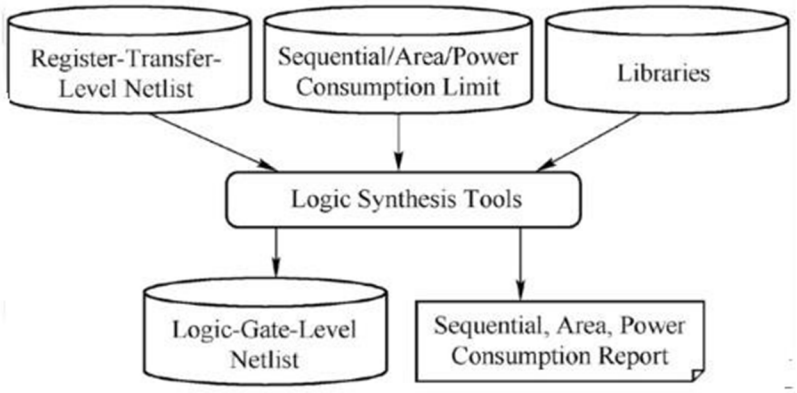


Figure 1: Digital design flow

This study employs a systematic approach to design and evaluate low-power compressors, beginning with a critical review of existing architectures to identify optimization opportunities. Proposed designs are modeled at the gate and transistor levels, incorporating logic simplification, voltage scaling, and alternative logic styles to minimize power while preserving performance. Rigorous verification is conducted using industry-standard EDA tools, with power, delay, and area metrics benchmarked against conventional designs. The methodology further validates practical applicability by integrating optimized compressors into multiplier circuits, assessing their impact on real-world applications like AI acceleration and signal processing through standardized evaluation frameworks.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

III. FUNCTIONAL VERIFICATION OF LOW POWER COMPRESSOR

Functional verification of low-power compressors involves rigorous testing to ensure correct arithmetic operations while meeting power efficiency targets. Testbenches are developed to apply exhaustive input combinations, including edge cases, to validate the compressor's behavior under typical and worst-case scenarios. Power-aware simulations track dynamic and leakage power consumption during operation, ensuring the design adheres to predefined energy budgets. Metrics such as error rate, propagation delay, and power-delay product are analyzed to verify compliance with both functional and low-power objectives. This step is critical to identify logic flaws or unintended power overheads before physical implementation.

To enhance verification coverage, advanced techniques like assertion-based checking and constrained random testing are employed. These methods systematically explore the compressor's response to diverse input patterns, including those specific to approximate computing applications where minor errors are permissible. Tools such as ModelSim or VCS correlate simulation results with RTL and gate-level models, while power analysis tools (e.g., PrimeTime) quantify energy savings. By cross-validating results against golden reference models, the verification process ensures the compressor maintains reliability in real-world deployments, such as AI accelerators or DSP units, where power efficiency and functional accuracy are equally critical.

1) *Code Development:* The 4:2 compressor module shown in the image is part of a hardware design implemented in Verilog HDL using the Xilinx ISE tool. The module, named COMPRESSOR_APPROX_MULTIPLIER, aims to optimize the multiplication process by reducing the number of partial product rows generated during binary multiplication. Instead of using a standard multiplication approach, this design compresses the intermediate results using a 4:2 compressor technique. In this architecture, multiple partial products are fed into a compressor circuit that reduces four input bits into two output bits—sum and carry—along with a carry-in and carry-out, improving both speed and resource efficiency. This approach is particularly beneficial in high-performance applications where latency and hardware utilization are critical, such as in digital signal processing and VLSI design for communication systems. The code defines various inputs and outputs representing partial products and the resulting compressed values. Each wire in the module corresponds to a bit in the binary multiplication process, and through logical operations, the module compresses them efficiently. This results in a reduced critical path and fewer logic levels, enabling faster arithmetic operations while consuming less power and chip area compared to conventional methods.

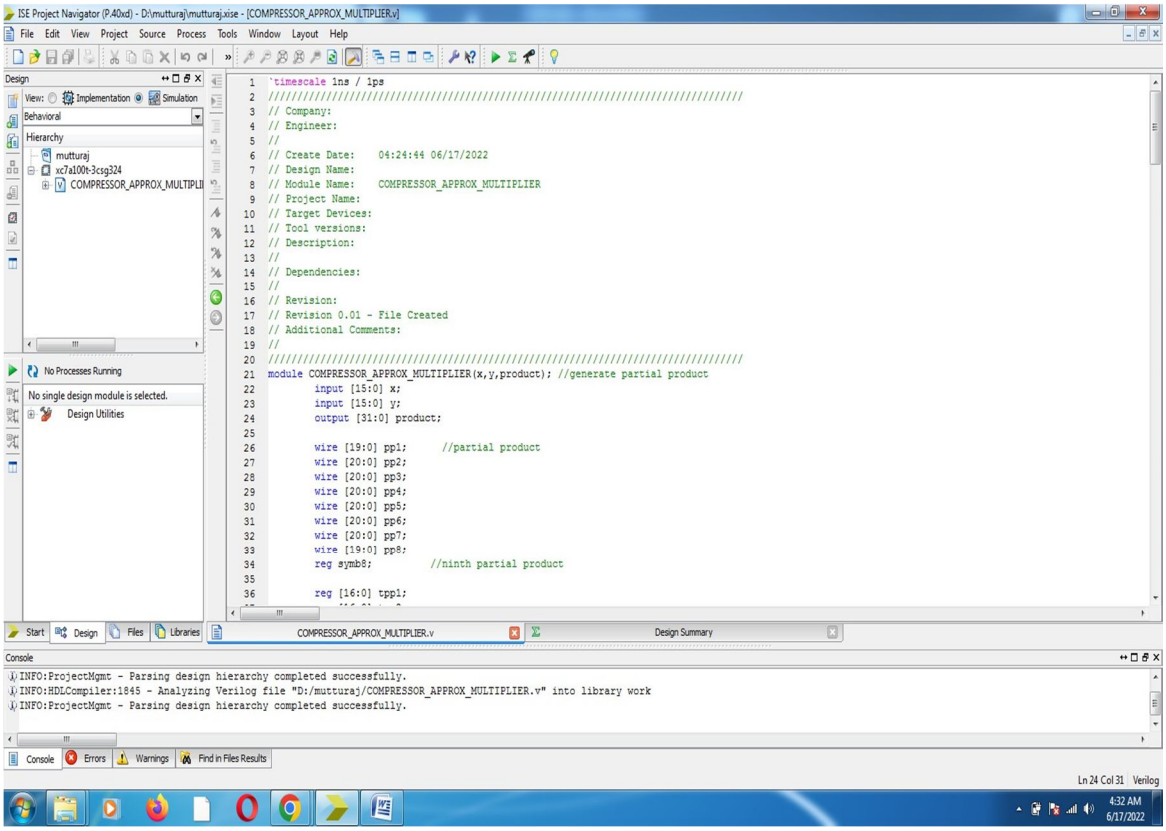


Figure 2: Verilog code Execution in Xilinx

2) *RTL Schematic:* The RTL schematic displayed in the image represents the internal structure of the COMPRESSOR_APPROX_MULTIPLIER module. This module is designed to take two 16- bit inputs, labeled x[15:0] and y[15:0], and generate a 32-bit output labeled product[31:0]. The schematic clearly shows the inputs entering from the left and the output exiting on the right, indicating the flow of data from input to output through the multiplier logic. The naming convention and organization reflect a hierarchical design approach, where the module is neatly encapsulated for integration into a larger digital system. This design uses a 4:2 compressor-based approach to approximate multiplication, which helps reduce hardware complexity while maintaining acceptable accuracy. By compressing the partial products during the multiplication process, this module achieves better performance in terms of speed and power efficiency. The RTL view is crucial for verifying signal connectivity and logic structure before synthesis, ensuring that the logic implemented in Verilog matches the intended functional design. The visual clarity of the schematic helps developers trace signals and debug the design at the register-transfer level.

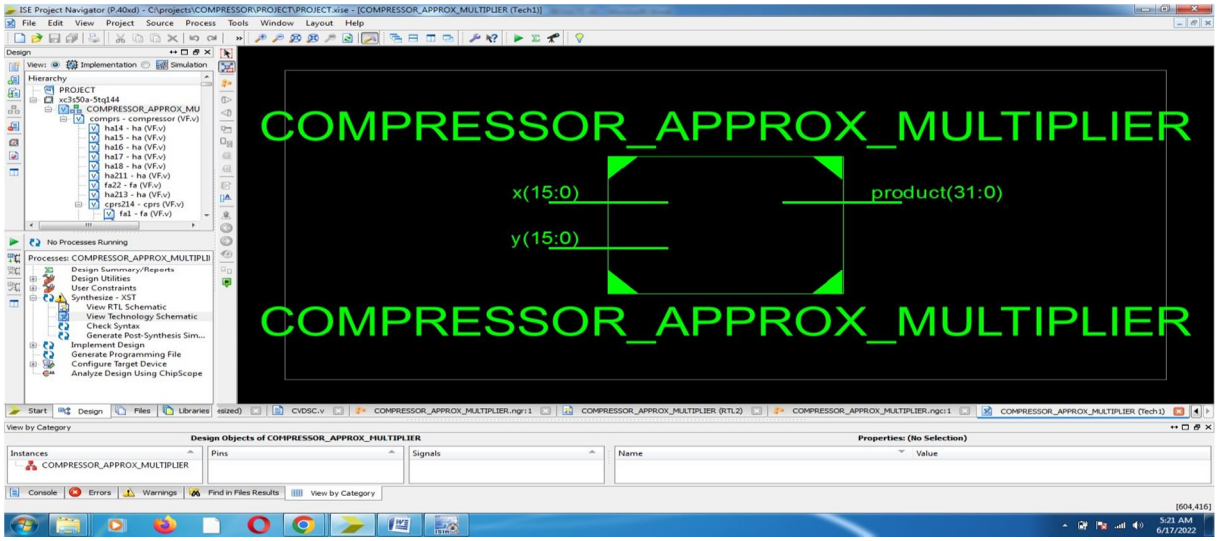


Figure 3: RTL Schematic

COMPRESSOR_APPROX_MULTIPLIER module, which represents the synthesized view of the hardware logic. In this schematic, basic logic gates such as AND, OR, XOR, and multiplexers are arranged in a structured layout generated by the synthesis tool. This view reveals how the high-level Verilog code is transformed into actual gate-level components using the target FPGA's or ASIC's standard cell library. Each small block and interconnecting line corresponds to a hardware element and its associated signal path, reflecting the real hardware implementation.

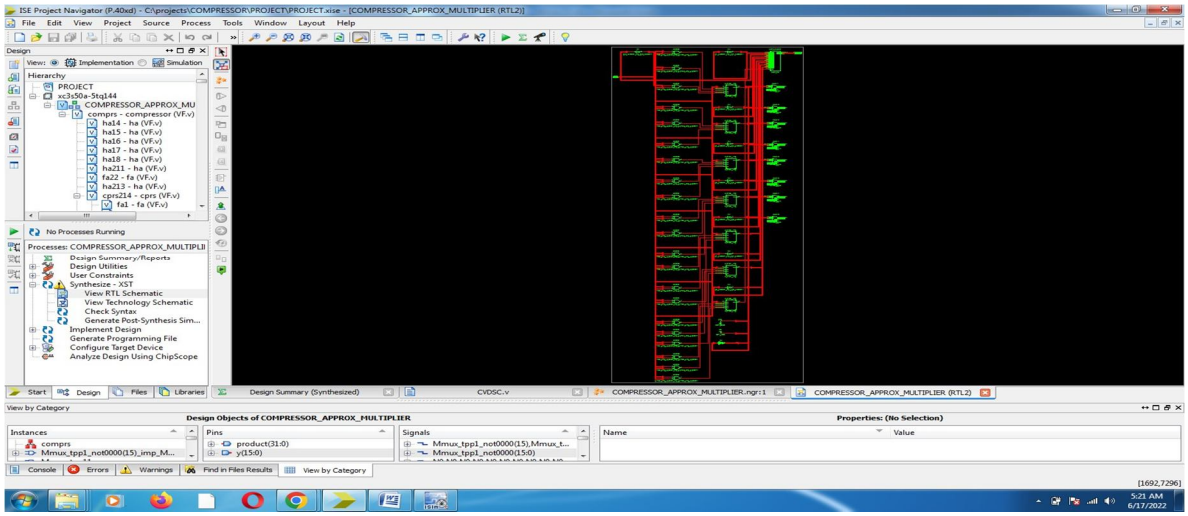


Figure 4: Technology Schematic

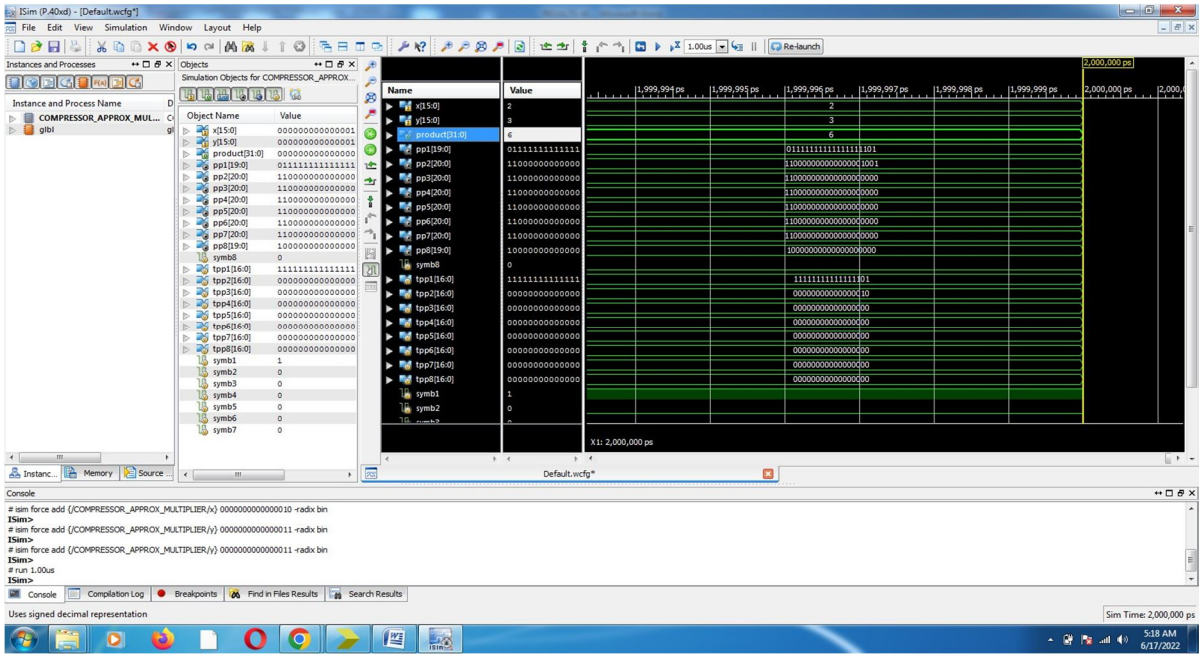


Figure 5: Functional verification of the results

The simulation confirms that the COMPRESSOR_APPROX_MULTIPLIER module operates correctly by producing valid output results for given 16-bit input combinations. Throughout the waveform, each change in inputs x and y leads to a corresponding update in the 32-bit product output, demonstrating consistent and expected behavior. The absence of glitches or undefined values indicates stable signal transitions and proper functionality of the 4:2 compressor logic, validating the design's correctness and readiness for further hardware implementation.

IV. SYNTHESIS OF LOW POWER COMPRESSOR

The synthesis of a low-power 4:2 compressor using the Cadence Genus tool involves optimizing the RTL design for reduced power consumption while maintaining functional accuracy and timing performance. Genus performs logic mapping, gate-level optimization, and technology-specific cell selection to minimize dynamic and static power. During synthesis, techniques such as clock gating, logic restructuring, and low-power cell insertion are employed to achieve energy-efficient operation. The tool also generates detailed power analysis reports, enabling designers to identify and eliminate power-hungry paths within the compressor structure. Simulation following synthesis is essential to verify that the low-power optimizations do not affect the intended behavior of the design. Post-synthesis simulation uses the gate-level netlist generated by Genus and includes switching activity to reflect real power usage scenarios. The waveform outputs confirm that the compressed partial product generation logic remains intact, and the timing requirements are satisfied. This step ensures the compressor is not only functionally correct but also optimized for deployment in power- sensitive applications such as portable communication devices or battery-operated systems.

The synthesis procedure involves converting the RTL (Register Transfer Level) Verilog code into a gate-level netlist using a synthesis tool like Cadence Genus. The process begins by importing the RTL design and setting up the design constraints, including timing, area, and power requirements. The tool then analyzes the design, performs optimization, and maps the logic to technology-specific standard cells. During this step, techniques such as logic minimization, retiming, and resource sharing are applied to improve performance and reduce power consumption. Finally, the tool generates reports detailing timing, area, and power estimates, and outputs a gate-level netlist ready for further verification or physical implementation.

Simulation after synthesis validates that the gate-level implementation preserves the intended functionality defined in the RTL. The synthesized compressor design is tested with multiple input vectors, and its outputs are observed in comparison to expected results. Accurate signal routing and cell placement, as seen in the image, help confirm that the design meets the defined power and timing specifications. The post-synthesis simulation confirms both structural correctness and low-power behavior, ensuring the circuit can be reliably integrated into larger arithmetic units for low-energy digital systems.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

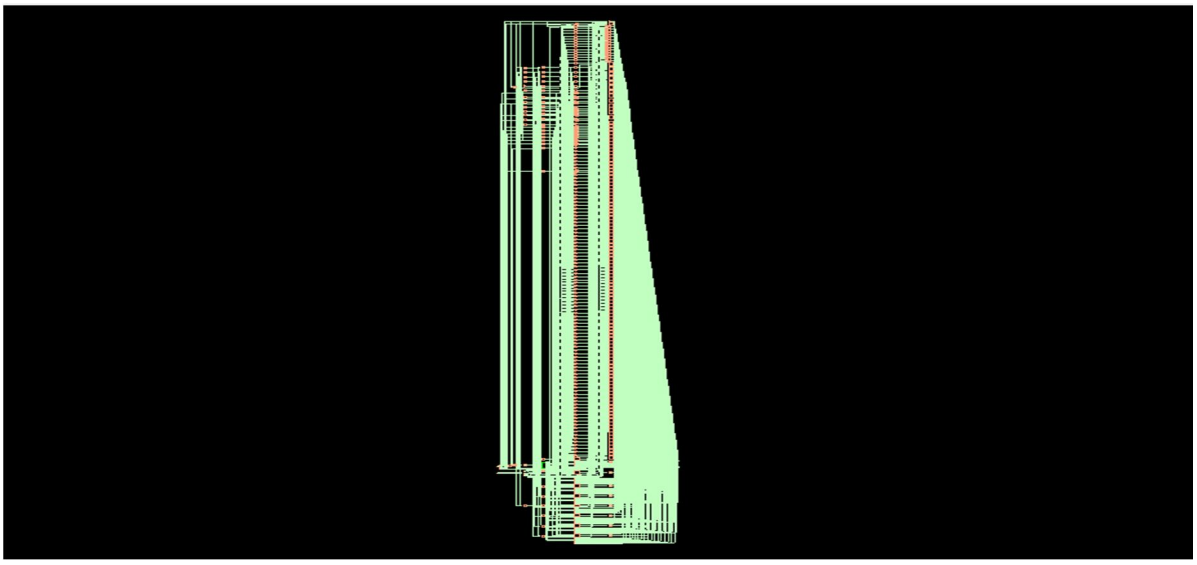


Figure 6: Synthesis

1) *Power Report:* The power report from the Cadence Genus synthesis tool provides detailed information on the power consumption of the COMPRESSOR_APPROX_MULTIPLIER module. It shows that the design utilizes 341 standard cells, with a total power consumption of approximately 65,379.739 nanowatts. This total power is broken down into two main components: leakage power, which amounts to 4,608.253 nanowatts, and dynamic power, which contributes the majority at 60,771.486 nanowatts. The data confirms that the design maintains a balanced trade-off between performance and power efficiency, which is crucial for low-power applications. The enclosed wireload and timing library modes indicate that synthesis was done under realistic physical constraints, ensuring accurate power estimation.

```
legacy_genus:/> report power
================================================================

  Module:                   COMPRESSOR_APPROX_MULTIPLIER
  Technology library:       typical
  Operating conditions:     typical (balanced_tree)
  Wireload mode:            enclosed
  Area mode:                timing library
================================================================

                                    Leakage    Dynamic     Total
          Instance          Cells Power(nW) Power(nW) Power(nW)
----------------------------------------------------------------
COMPRESSOR..OX_MULTIPLIER    341  4608.253 60771.486 65379.739
  comprs                       0     0.000   278.906   278.906


legacy_genus:/> report area
============================================================


  Module:                 COMPRESSOR_APPROX_MULTIPLIER
  Technology library:     typical
  Operating conditions:   typical (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

         Instance           Module  Cells  Cell Area  Net Area   Total Area  Wireload
------------------------------------------------------------------------------------
COMPRESSOR_APPROX_MULTIPLIER          341       1991         0         1991    <none> (D)
```

The synthesis summary of the COMPRESSOR_APPROX_MULTIPLIER reveals an optimized design with a total of 341 logic cells used, occupying a combined area of 1991 units as reported by the Genus synthesis tool. The design operates under typical conditions with a balanced tree wireload model. Power analysis shows that the total power consumption stands at approximately 65.38 µW, with dynamic power contributing the most at 60.77 µW and leakage power being relatively low at 4.60 µW. The timing report indicates that the maximum delay from input to output is 460 ps, which confirms that the design meets timing requirements efficiently. This synthesis result reflects a low-power and area-efficient implementation, suitable for high-performance applications where resource constraints are critical.

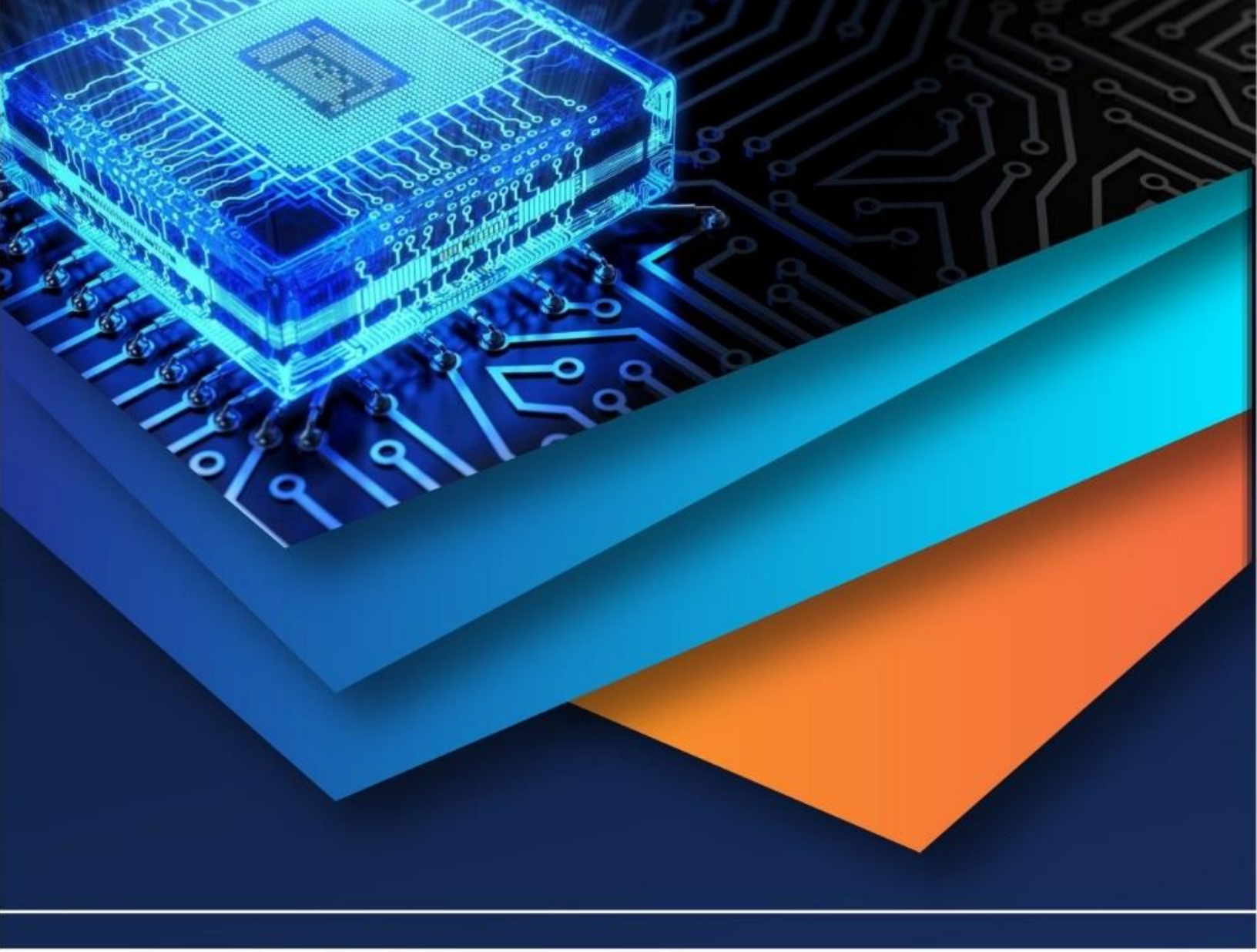
IV. CONCLUSIONS

The thesis presented in this thesis successfully develops and analyses low-power 4-2 compressor architectures tailored for energy-efficient multipliers in error-resilient applications. By exploring various innovative circuit-level optimizations and approximation techniques, the study achieves a significant reduction in both dynamic and static power consumption while maintaining acceptable levels of computational accuracy. The introduction of standardized evaluation metrics allows for a more comprehensive comparison among different compressor designs, providing a clearer framework for assessing trade-offs between power efficiency and performance reliability. This work contributes crucial insights into the development of sustainable and high-performance digital systems, particularly in domains like AI accelerators and digital signal processing.

Furthermore, the findings indicate that employing approximate compressors not only enhances energy savings but also enables the design of multipliers that can tolerate minor inaccuracies, aligning with the growing demand for energy-efficient computing solutions. The systematic methodology applied throughout the research ensures that the proposed low- power compressors can be effectively integrated across various applications, paving the way for future advancements in arithmetic circuit designs. As the need for power-efficient solutions continues to rise, this research lays a solid foundation for ongoing exploration and innovation in the field, with practical implications for next-generation embedded and portable devices.

REFERENCES

[1] Mr. Yogesh G. Desale, Dr. Mrs. V.V. Ingale, ―Design of Power Efficient Bit Serial Finite Field GF(2m) Multiplier‖, 978-1-5386-8075-9/19/$31.00 ©2017 IEEE.
[2] A.V. Aravind and Senthil Murugan, ―VLSI Design of 128 Point Finite Field FFT Multiplier‖, 978-1-5090-3800-8/17/$31.00 ©2017 IEEE.
[3] Jianing Su, Zhenghao Lu, ―Parallel Structure of GF (214) and GF (216) Multipliers Based on Composite Finite Fields‖, 978-1-61284-193-9/11/$26.00 ©2016 IEEE.
[4] Karl Leboeuf, Ashkan Hosseinzadeh Namin, Huapeng Wu, Roberto Muscedere, and Majid Ahmadi, ―Efficient VLSI Implementation of a Finite Field Multiplier Using Reordered Normal Basis‖, 978-1-4244-7773-9/10/$26.00 ©2015 IEEE.
[5] Ashkan Hosseinzadeh Namin, Huapeng Wu, ―A New Finite-Field Multiplier Using Redundant Representation‖, 0018-9340/08/$25.00 2015 IEEE.
[6] P. Meher, ―On efficient implementation of accumulation in finite field over GF(2m) and its applications,‖ IEEE Trans. VLSI Syst., vol. 17, pp. 541–550, Apr. 2014.
[7] W. Tang, H. Wu, and M. Ahmadi, ―VLSI implementation of bit-parallel word-serial- multiplier in GF(2233),‖ in Proc. The 3rd International IEEE-NEWCAS Conference, Xi'an, China, Jun. 2013, pp. 198–201.
[8] A. Reyhani-Masoleh and M. A. Hasan, ―Efficient digit-serial normal basis multipliers over GF(2m),‖ in Proc. 2012 IEEE International Symposium on Circuits and System.
[9] J.Großschadl ,―A Low-Power Bit-Serial Multiplier For Finite Fields GF(2m)‖.34th IEEE International Symposium on Circuits and Systems (ISCAS) 2011.
[10] S. Gao and S. A. Vanstone, ―On orders of optimal normal basis generators,‖ Math. Comp, vol. 64, pp. 1227–1233, Jul. 2011.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY