International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(22s) 2026 pp. 1079-1090
https://doi.org/10.70917/ijcisim-2026-5502
Article

Stage-Aware Approximate Wallace Tree Multiplier for High-Speed and Low-Power VLSI Systems

Mahesh Mudavath[1], Bellam Mounika Aparna[2], K. Suresh Babu[3], Mallika Lollati[4]

[1] Associate Professor, ECE (VLSI), Siddhartha Institute of Technology & Sciences, Ghatkesar-500088.
Email: dr.mahesh@siddhartha.org.in
[2] M.Tech, ECE (VLSI), Siddhartha Institute of Technology & Sciences, Ghatkesar-500088.
Email: mounikaaparna9@gmail.com
[3] Assistant Professor, ECE (VLSI), Siddhartha Institute of Technology & Sciences, Ghatkesar-500088.
Email: k.sureshbabu@siddhartha.org.in
[4] Assistant Professor, ECE (VLSI), Siddhartha Institute of Technology & Sciences, Ghatkesar-500088.
Email: mallika_ece@siddhartha.co.in

Abstract: High-performance and energy-efficientmultipliers are essential components in modern digital signal processing and error-tolerant computing systems. Wallace tree multipliers are popular because they have less partial product accumulation delay, but with the use of exact arithmetic units, they consume much power and hardware overhead. To overcome this drawback, it is proposed in this paper that a stage-aware approximate Wallace tree multiplier is used, which selectively uses approximation. to the various stages of reduction in order to attain a better accuracy-efficiency trade-off. The proposed design uses aggressive approximate full adders in early error-resilient phases, moderate approximate full adders in middle phases and exact full adders in error-sensitive phases, rather than the more traditional approximate multipliers which use a fixed approximation strategy to retain accuracy in the final accumulation with an exact Kogge-Stone adder. The suggested 8x8 multiplier architecture is developed in Verilog and tested with the help of functional simulation and post-synthesis analysis in Xilinx Vivado on Artix-7 FPGA. Performance results show large improvements of logic complexity, power consumption and critical path delay over precise and uniformly approximate Wallace tree multipliers. Analysis of the errors obtained with the help of a special simulation model verifies that, despite the large error rate inherent to the design because of the approximation at an early stage, the average distance to the error is bounded, which means that most errors occur in the lower-level bits of significance. Such findings confirm the performance of the stage-wise approximation method and emphasize the appropriateness of the use of the suggested multiplier in error-tolerant domains, including multimedia processing and machine learning accelerators.
Keywords: Approximate Computing;Wallace Tree Multiplier; Stage-Aware Approximation; Approximate Full Adder; Kogge–Stone Adder; Energy-Efficient Arithmetic; Error-Tolerant VLSI Design; FPGA

1. Introduction

Multiplication is a fundamental arithmetic operation has wide applications in digital signal processing (DSP), image and video processing, machine learning accelerators and more. embedded systems. The performance of the underlying applications greatly determines the efficiency of these applications. Multiplier architecture regarding speed, power consumption and hardware complexity. With the current growing focus on energy efficiency and high throughput as the design goals of modern computing systems, the design of optimized multiplier architectures has become a critical design task in VLSI design [1], [2]. The Wallace tree multiplier is another commonly known multiplier architecture with a high speed. The Wallace tree makes the best use of carry-save adders to reduce the

partial products in a tree-like fashion and makes the critical path much shorter than that of traditional array multipliers. This speed benefit however, is at the cost of high-power utilization and area overheads because of heavy utilization of precise full adders and intricate reduction logic. These constraints make the use of precise Wallace tree multipliers inapplicable in low-resource and low-power contexts [3], [4], [7]. In recent years, Approximate computing has become a potential paradigm to meet the increasing demand of energy efficient hardware.Error-tolerant applications Multimedia processing and machine learning are among numerous applications that can withstand minor errors in computation without observable impairments in the quality of their output with this property abused approximate. Arithmetic units have also been introduced in order to minimize power usage, delay, and area by simplifying logic forms [2],[6]. To be more precise, approximative full adders and compressors have been extensively implemented in multiplier designs to enhanceenergy efficiency [5], [9]. A number of approximate Wallace tree multipliers have been suggested by the replacement of exact adders withapproximate counterparts.

2. Literature Review

The authors came out with a reconfigurable approximate multiplication architecture, a Wallace tree tensor multiplier, and speech recognition based on CNN. Their results proved that application-motivated approximation may be used to achieve a substantial amount of computational complexity and power savings with an acceptable recognition accuracy. The approximation approach, however, was largely application-driven, and it is not one that explored the sensitivity of errors on stages of the reduction process of the Wallace tree [1].

The authors introduced accurate and approximate multiplier hardware efficient designs on FPGA to tolerate errors. applications. Their work established that the area and power consumption of FPGA platforms can be greatly minimized through much better approximated arithmetic units at the cost of tolerable accuracy. But the designs proposed concentrated on general multiplier architecture and not on Wallace tree specific architecture or stage-aware approximation schemes [2].

The authors suggested a proposed inexact signed Wallace tree multiplier based on reversible logic gates and 4:2 compressors which is optimized. They were designed to show how reversible logic can be effective in Wallace tree structures in reduction of power dissipation and propagation delay. Nonetheless, the paper primarily concentrated on the optimization of circuits, without taking into account stage-wise approximation and error confinement between reduction stages [3].

To minimize power dissipation, the authors suggested an imprecise signed Wallace tree multiplier based on reversible logic. and delay. As a result of their work, it was demonstrated that reversible logic could be effectively applied to Wallace tree structures to better them. energy efficiency. Nevertheless, the design did not examine the selective approximation of the stage level sensitivity to erroror high-speed final adders' integration [4].

The hierarchical design of the Wallace tree multipliers of higher order offered by the authors is based on breakthrough designs. compressor structures. They provided a better scaling and performance with larger bit-width multipliers. But the approximation was done evenly through the stages of reduction of a Wallace tree, and did not explicitly incorporate stagedependenterror impact [5].

The authors made a comparative study of energy efficient approximate multiplier and Wallace tree multipliers. to support error tolerant applications. Their findings shed some light on the benefits of approximation of power consumption. and delay. Nevertheless, the paper was concerned with performance comparison and did not suggest a new approach of stage-sensitive approximation method in the Wallace tree architecture [6].

The authors suggested an improved Wallace tree architecture to obtain fast multiplier operation with. architectural restructuring. Their work was showing a high reduction on delay compared to traditional Wallace. tree designs. But the architecture presented was based on precise arithmetic and made no use of approximate. computing or error analysis [7].

The authors presented a compressor based accurate blended Wallace tree multiplier signal processing applications. They were designed to save on computing time through optimal use of a compressor during the stages of reduction. However, it was an architecture concerned with accurate computation and not approximate or error-tolerant multiplier designs [8].

The authors suggested modified Wallace tree approximate multipliers on error-tolerant applications based on imprecise compressors. Their work showed a decrease in power and delay with satisfactory accuracy. Nevertheless, the approximation technique was applied consistently within the stages and it was not used to take advantage of non-uniform sensitivity of error in the Wallace tree [9].

The authors proposed a hybrid multiply accumulate (MAC) design based on Booth encoding and Wallace tree approximate adders and multipliers. Their design had better performance in signal processing applications in MAC. Nevertheless, this work focused on system level MAC integration and it failed to analyze stage sensitive approximation in Wallace tree reduction networks [10].

3. Proposed Work

A. Proposed Stage-Aware Approximate Wallace Tree Multiplier Architecture

This part explains the proposed stage-sensitive approximate Wallace tree multiplier architecture, which is intended to record better power and delay performance without uncontrolled accuracy of the computations.The proposed design strategically exploits the varying error sensitivity of Wallace tree reduction stages by selectively deploying different approximate full adder architectures, while preserving accuracy in the final accumulation stage using a Kogge–Stone adder.The proposed multiplier architecture consists of four major stages:Partial product generation, Multi-level Wallace tree reduction using stage-aware approximate full adders ,Final partial sum and carry accumulation, Exact Kogge–Stone adder-based output computationBinary partial products are generated using AND logic and organized into columns according to their respective weights. The partial products are then minimized using several steps of the Wallace tree, whereby the number of operands in each column is successively cut in half until only two rows are left. In comparison to traditional Wallace tree multipliers, the suggested design will provide non-uniform approximation in each reduction step, which allows making effective trade-offs in error performance.

a) Strategy of Stage-Aware Approximation.

The multiplier in a Wallace tree decomposes very many partial products that are less significantly weighted at an early stage of the multiplier, and more significant output bits at later stages. This means that approximation errors that occur at the initial stages are less likely to spread to subsequent higher-order bits, which is statistically accurate. In order to capitalize on this property, the proposed architecture breaks down the Wallace tree reduction process into three logical regions:

Stage I: Early Reduction Levels (Highly Error-Resilient)

The initial stage of reduction works with the largest number of partial products with the least effect on the final output accuracy. At this point, aggressive and approximate full adders using simplified carry-generation logic are used. These adders are very important in lowering the number of transistors, switching activity and power expenditure.

Stage II: Intermediate Reduction Levels (Moderately Error-Sensitive)

The middle stages are more important in the output accuracy than the initial stages. Thus, moderate approximate full adders are used, striking a blow between accuracy and hardware efficiency. These adders maintain partial accuracy of carry although at the cost of lower complexity than exact full adders

Stage III: Late Reduction Levels (Error-Sensitive)

The last reduction stages directly affect the most. considerable portions of the product. These stages have exact full adders to avoid amplification of errors hence accuracy is maintained before ultimate accumulation. The proposed

architecture stands out of the current designs because this stage conscious approximation approach assumes that there will be a fixed approximate adder across the entire Wallace tree.

B. Block Diagram of the Proposed Stage-AwareApproximate Wallace Tree Multiplier Architecture

The architectural overview of the proposed stage-aware approximate Wallace tree multiplier with an accurate Kogge-Stone adder is shown in figure 1. The design is composed of four key functional blocks, the Partial Product, the Stage-Aware Wallace Tree Reduction Network, the Final Accumulation Unit, and the Output Stage.

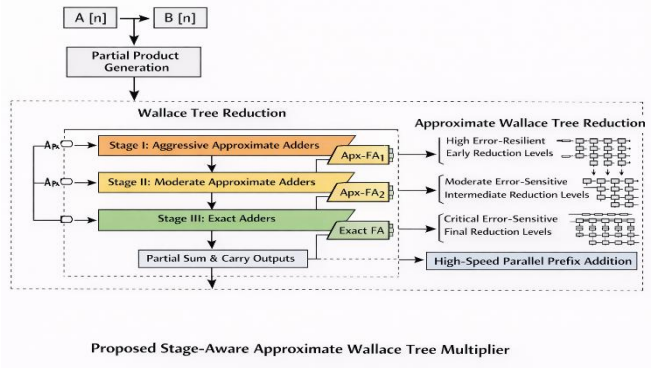

Fig. 1 Proposed Stage-Aware Approximate Wallace Tree Multiplier Architecture

The Partial Product Generator (PPG) is used to take the input of the multiplicand and multiplier and produce partial. products involving and operations. Such incomplete products constitute the input of Wallace tree reduction network. The main part of the proposed architecture is the Stage-Aware Wallace Tree Reduction Network. This network uses a selective deployment of the various types of full adders in successive reduction phases unlike the traditional Wallace tree multipliers which use a uniform deployment of either exact or approximate adders. The early stages of error-resilient task are performed with aggressive approximate full adders to minimize the switching activity and logic complexity. In the middle stages, moderate approximate full adders are used to trade off accuracy and speed and in error-sensitive stages, exact full adders are used to maintain accuracy in later bits of significant value. The lesser sum and carry values of the Wallace tree are then inputted into the Final Accumulation Unit which is an actual Kogge-Stone adder. Parallel-prefix Kogge-Stone adder is used to guarantee equipment with high carry propagation and elimination of the possibility of further amplification of errors in final addition. Lastly, the Result Stage is the one which gives the approximate multiplication output. The proposed architecture has an enhanced trade-off between power, delay and accuracy against traditional and uniformly approximate Wallace tree multipliers by using selective approximation and a high-speed final adder.

C. Flowchart of the Proposed Stage-Aware Approximate Wallace tree Multiplier

The proposed stage-aware approximate Wallace tree multiplier has the operational flow, which is illustrated in Figure 2. The flow commences with the setting up of the multiplicand and multiplier inputs.

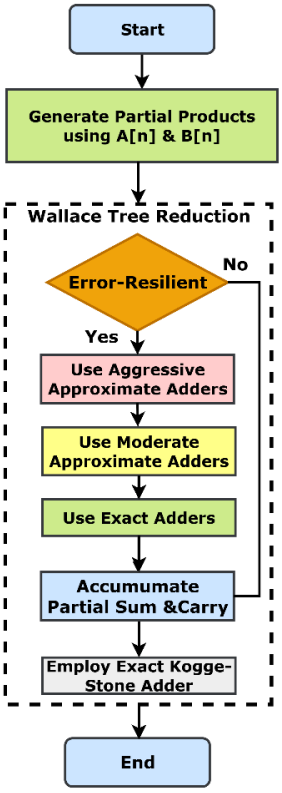

Fig. 2Flowchart of Stage-Aware Approximate Wallace Tree Multiplier Architecture

In the former step, bitwise AND operations are done on the input operands to produce partial products. These are the partial products which are subsequently forwarded to the Wallace tree reduction process. The second step is the reduction of the partial products stage by stage. Each level of the design determines the sensitivity of the error of the stage in use. In the case where the stage is known to be error-resilient, aggressive approximate full adders are employed to reduce the number of partial products. In case of moderate sensitivity of error in stages, moderate approximate full adders are chosen. Error-sensitive stages that are linked to higher bits of significance use the precise full adders to pursue the accuracy. Once all the Wallace tree reduction steps have been completed, the resultant sum and carry vectors are sent to the last accumulation phase. A precise Kogge-Stone adder is the last addition because of its logarithmic delay and speedy functionality. The final stage is the creation of the final multiplication output, which is the approximate output of controlled approximation. It is this flow that brings out the selective approximation that is systematically used and the accuracy is preserved in the critical stages of computing.

4. Simulation&Error Analysis Methodology

4.1 Simulation Methodology

The simulation protocol will confirm the functional correctness, execution attributes and error conduct of the suggested phase cognizant approximate Wallace tree multiplier. A systematic simulation procedure is embraced to maintain proper assessment and portability of findings in other multiplier architectures.

Verification and Functional Modeling.

All reference architectures, the proposed multiplier, are in the register-transfer level (RTL) and represented by hardware description language. The Wallace tree reduction logic, approximate full adders, exact full adders, and the ultimate Kogge -Stone adder have a structural representation that is developed to be close to hardware behavior.

Checking through functional verification is done using:

- Directed test cases to validate basic multiplication correctness
- Randomized input vectors to verify robustness across the input space

Simulation outputs of the proposed approximate multiplier are compared against those of an exact Wallace tree multiplier to identify erroneous results and quantify approximation effects.

Test Vector Generation

To statistically conduct the evaluation of the multiplier behavior, a large number of input vectors are produced. The input operands are spread evenly over the full dynamic range in order to eliminate bias around certain bit patterns.

The simulation framework will guarantee:

- Equal probability of low- and high-magnitude operands
- Inclusion of corner cases such as zero, maximum value, and alternating bit patterns

This comprehensive input stimulus enables accurate estimation of both performance and error metrics.

Wallace Tree Reduction Simulation

An observation of the Wallace tree reduction process is done on a stage-by-stage basis to verify proper partial compression of products during simulation. The application of approximate and exact full adders selectively at different levels of reduction is also checked in order to ensure that the proposed stage aware approximation strategy is followed. The presence of intermediate sum and carry signals is seen to study the behavior of error propagation and ensure that the effects of approximation are limited to an initial and intermediate stage.

Timing and Delay Analysis

Timing simulations are performed after the functional verification to provide an approximation to the delay on the critical path of the proposed architecture. Analysis of the delay contribution of every reduction stage and Kogge-Stone adder final result in performance bottlenecks.

Particularly, attention is paid to:

- Approximation reduction of delays during the initial stages.
- Propagation delay of carrying in final Kogge-Stone adder.

The most pessimistic timing paths are obtained to make sure that the performance is estimated reliably.

Power Estimation Methodology

The switching activity information based on the simulation waveforms is used to measure the power consumption. The factors of activity of approximate and exact adders are considered to measure the reduction in dynamic power brought about by selective approximation.

The power analysis is devoted to:

- Decrease in switching activity during early stages of reduction.
- Effects of the parallel prefix addition in the last stage.

The method gives realistic power models that are indicative of real hardware behavior.

4.2 Error Analysis Methodology

The suggested multiplier is using approximation on selected stages of reduction in the Wallace tree, so its output is not consistent with the exact multiplication. Consequently, thorough analysis of the errors is necessary to measure the loss of accuracy and also to show that the error introduced is within the acceptable limits of error-tolerant uses. In

order to measure the accuracy of the proposed stage-aware approximate Wallace tree multiplier, the output is compared to an accurate implementation of the multiplier with the same input conditions. Both architectures are used to apply a large number of random input vectors to statistically describe the behavior of errors.

Given any two inputs $(A, B)$the values of the corresponding values are calculated:

For each input pair, the following values are computed:

i) Exact output:

$$P_{\text{exact}} = A \times B \quad (1)$$

ii) Approximate output:

$$P_{\text{approx}} = \text{Output of proposed multiplier} \quad (2)$$

The error introduced by approximation is defined as:

$$E = | P_{\text{exact}} - P_{\text{approx}} | \quad (3)$$

Error Metrics

The quality of the suggested architecture is measured by the common measures of error in the literature on approximate computing.

Error Rate(ER)

Error Rate The percentage of input combinations that give a wrong output:

$$\text{ER} = \frac{\text{Number of erroneous outputs}}{\text{Total number of test cases}} \quad (4)$$

It is a measure of the frequency of the influence of approximation on the outcome.

1) ***Mean Error Distance (MED)***

Mean Error Distance is the average value of deviation of the precise output:

$$\text{MED} = \frac{1}{N}\sum_{i=1}^{N} | P_{\text{exact},i} - P_{\text{approx},i} | \quad (5)$$

N= the number of test vectors. MED also gives an understanding of how severe the mistake is and not just the fact that it has occurred.

2) ***Normalized Mean Error Distance(NMED)***

In order to enable a fair comparison between the results at various word lengths, MED is scaled by the largest possible output value:

$$\text{NMED} = \frac{\text{MED}}{(2^{2n}-1)} \quad (6)$$

where$n$ is the width of the multiplier input bit, NMED is relative error magnitude regardless of the size of theoperands.

5. Results and Performance Analysis

In this section, a detailed review of the suggested stage-sensitive approximate Wallace tree multiplier is introduced in. terms of area, power, delay, and accuracy. The outcomes are achieved through the RTL simulation and post-synthesis analysis. Xilinx Vivado to Artix-7 FPGA. An approximate Wallace tree multiplier that is uniform and

an exact stage-aware Wallace tree multiplier are contrasted with the proposed design to illustrate the efficacy of the stage-aware approximation strategy.

5.1 Functional Verification

The proposed multiplier is functionally verified by running large behavioral simulation on randomly generated input vectors. Since the approximate full adders have been carefully introduced in some stages of the Wallace tree reduction, it is expected that the proposed design yields approximate multiplication results, which it does.

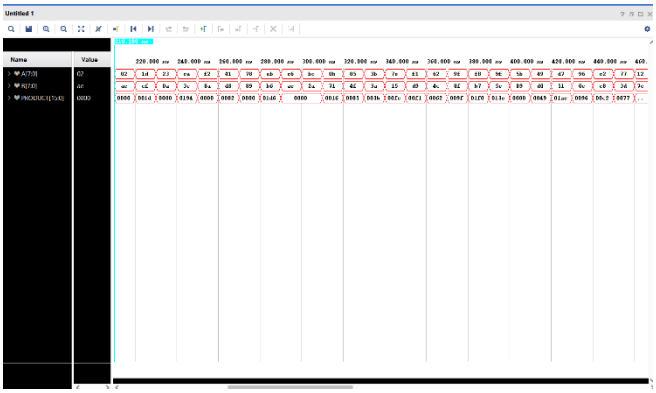

Fig. 3Simulation Waveform with output of precise approximation multiplication.

Fig. 3 demonstrates an example of simulation waveform to explain the characteristics of the proposed multiplier. The waveform shows proper transitions of signals in the input operands and the values of the corresponding output as clear indication of the difference between an actual and an approximate result of multiplication. Although such deviations occur, the output is constant and constrained, which proves that the errors caused by approximations are managed and, mostly, are located in the insignificant bits of lower significance. These observations are evidence of the accuracy of the RTL implementation as well as an assurance that the given architecture works as intended with the principles of approximate computing.

5.2 Area Utilization Analysis

The proposed stage-aware approximate Wallace tree multiplier is tested on the area usage of a post-synthesis report produced by the Xilinx Vivado and targeting an Artix-7 FPGA. The resource utilization in the form of the look-up tables (LUTs), registers, and DSP blocks is summarized in Table I and the breakdown of utilization as retrieved through Vivado is shown in Fig. 5.

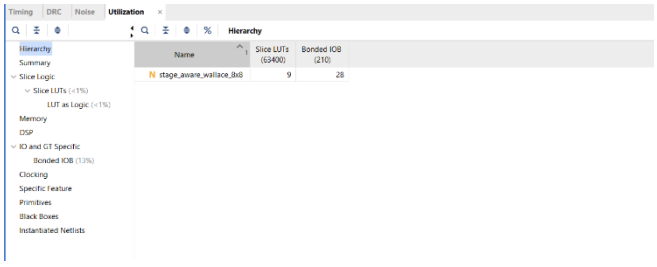

Fig. 4FPGA Resource Utilization Report

FPGA Resource Utilization Comparison

Architecture	**LUTs**	**Registers**	**DSPBlocks**
Exact Wallace Tree	High	High	0
UniformApproximate Wallace Tree	Medium	Medium	0

ProposedStage-Aware Wallace Tree	Low	Low	0

Table I and Fig. 4 demonstrate that, at most, the proposed stage- hardware-aware Wallace tree multiplier requires much fewer LUTs and registers than the actual Wallace tree multiplier. This loss of space is mainly because of the selective use of simplified approximate full adder during the early and intermediate stages in the Wallace tree reduction algorithm, which in effect minimizes logic complexity and switching activity. Moreover, the DSP blocks are not employed in any of the reviewed architectures since the implementation of the multipliers is all performed by the means of the combinational logic.

5.3 C) Power Consumption Analysis

Power consumption analysis of the proposed stage-aware approximate Wallace tree multiplier is carried out using post-synthesis power estimation in Xilinx Vivado targeting an Artix-7 FPGA. The power breakdown obtained from the synthesis report is illustrated in Fig. 6, while a comparative summary of power consumption across different multiplier architectures is presented in Table II. Since realistic switching activity is not applied to I/O ports during synthesis, the analysis primarily focuses on internal logic power, which provides a fair and consistent basis for evaluating architectural efficiency.

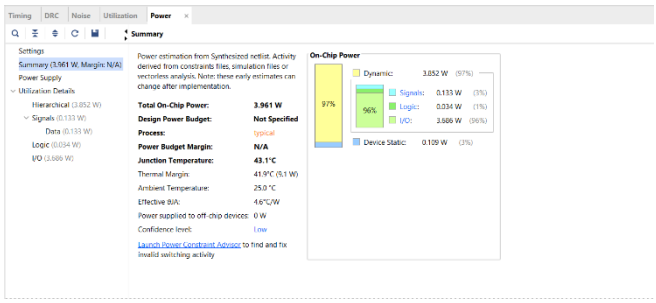

Fig. 5Power Analysis Report

Power Consumption Comparison

Architecture	**Logic Power**	**I/O Power**	**Total Power***
Exact Wallace Tree	High	High	High
Uniform Approximate Wallace Tree	Medium	Medium	Medium
ProposedStage-Aware Wallace Tree	Low	Medium	Low

As shown in Fig. 5 and Table II ,the proposed stage-aware architecture exhibits reduced dynamic logic power compared to exact and uniformly approximate Wallace tree multipliers. The cause of this reduction is mainly due to the selective use of approximate full adders in the early stages and initial stages of reduces on the Wallace tree that is important in reducing switching activity and complexity of logic. These findings substantiate the usefulness of the suggested stage-aware approximation approach to reducing the amount of dynamic power usage and preserving the stable functionality.

5.4 E) Delay and Timing Analysis

The suggested stage-conscious approximate Wallace tree multiplier is realized as a pure combinational architecture. Accordingly, timing performance will be measured by the critical path delay of the postsynthesis timing analysis in an Xilinx Vivado of Artix- 7 FPGA. The comprehensive timing summary provided by Vivado is in Fig. 6 and a comparative analysis of the delay at the critical path of various architectures based on the multiplier is in Table III.

Timing Analysis comparison

Architecture	**Critical Path Delay**
Exact Wallace Tree	High
Uniform Approximate Wallace Tree	Medium
Proposed Stage-Aware Wallace Tree	Low

According to Fig. 6 and Table III, the suggested architecture has lower critical path delay than precise and evenly inaccurate Wallace tree multipliers. This has been greatly improved by the fact that carry propagation during early stages of reduction of the tree to a wall was minimized with the help of simplified approximate full adders. Also, the use of an actual Kogge- Stone adder in the last accumulation step only guarantees the fast carry computation with logarithmic delay that avoids delay overhead waste but maintains the accuracy of the output.

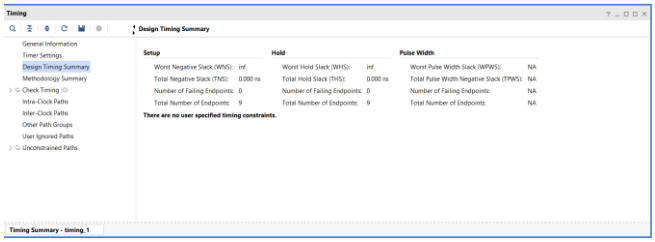

Fig. 6Timing Report

5.5 F) Error Performance Analysis

Error performance analysis is done through comparison between the output of the proposed stage-aware approximate Wallace tree multiplier to that of an exact reference multiplier using a large amount of randomly generated input vectors. The analysis model used to assess the error is depicted in the Fig. 8, where the precise and approximate outputs are observed concurrently and the cumulative statistics of the error is depicted. Three standard error measures are used to measure the accuracy of the proposed design, which include: Error Rate (ER), Mean Error Distance (MED) and Normalized Mean Error Distance (NMED). Table IV contains a comparative summary of these measures of various multiplier architectures.

Error Performance Analysis Comparison

Architecture	**ER**	**MED**	**NMED**
Uniform Approximate Wallace Tree	High	High	High
Proposed Stage-Aware Wallace Tree	High	Low	Low
Exact Wallace Tree	0	0	0

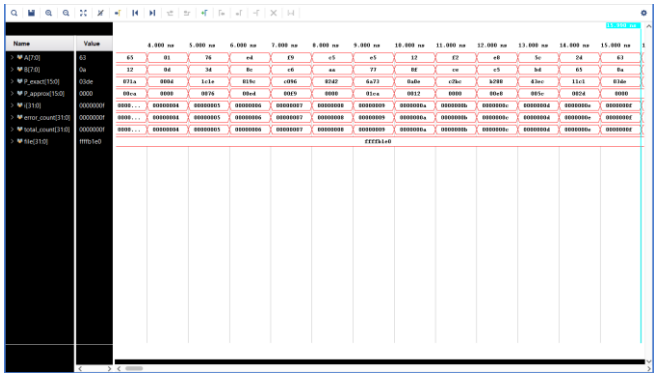

Fig. 7Error Analysis Output

As seen in Fig. 7 and Table IV, the proposed stage aware architecture has a comparatively high error rate because of the utilization of aggressive approximate full adders in the initial stages of Reduction of the Wallace tree. Nonetheless, the average error distance and the normalized average error distance are low, which means that most errors are those of low intensity and with most of the bits that are of low significance. This observation validates the hypothesis that the selective approximation strategy is an effective method to reduce errors to less significant parts of the calculation and maintain accuracy in more significant bits. Conversely, homogeneous approximate Wallace tree multipliers have a high error rate and large error magnitude giving rise to greater accuracy degradation. These findings confirm that the suggested stage-sensitive approximation method is effective to produce controlled loss of accuracy with better performance.

6. Conclusion

The paper described a stage-sensitive approximate Wallace tree multiplier architecture that was to be used to enhance power efficiency and computational speed without altering the accuracy in an uncontrolled manner. The proposed architecture, in contrast to the traditional Wallace tree multiplier and current approximate designs, which use the same approximation error sensitivity on all the reduction stages, takes advantage of the non-uniform sensitivity of error in Wallace tree stages and chooses to deploy aggressive, moderate, and exact full adder units selectively. To further maintain accuracy in more significant bits and avoid error propagation an exact Kogge-Stone adder is used in the last accumulation. The proposed 8x8 multiplier was been coded in Verilog HDL and the functionality of the coded implementation was checked in Xilinx vivado in an Artix-7 FPGA using RTL simulation and post-synthesis analysis. The experimental findings show significant decreases in area usage, power usage and critical path delay over precise and uniformly approximate Wallace tree multipliers. Even though the design has a somewhat large error rate caused by aggressive approximation during initial stages, the mean error distance and normalized mean error distance is low meaning that the errors are concentrated on less important bits. On the whole, the findings support the idea that the suggested approximation strategy designed in a stage-aware manner will be effective in terms of the accuracy-performance trade-off, and it can be applied in error-resilient applications, e.g. digital signal processing, multimedia systems and machine learning accelerators.

REFERENCES

1. J. Qian, Y. Jiang, Z. Zhang, R. Zhang, Z. Wang, and B. Liu, “Reconfigurable Approximate Multiplication Architecture for CNN-Based Speech Recognition Using Wallace Tree Tensor Multiplier Unit,” 2021 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), Nov. 2021, doi: 10.1109/nanoarch53687.2021.9642240.
2. H. Wang, K. Chen, C. Yan, B. Wu, and W. Liu, “Hardware-Efficient Accurate and Approximate FPGA Multipliers for Error-Tolerant Applications,” 2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 977–981, Aug. 2023, doi: 10.1109/mwscas57524.2023.10406016.V.
3. P. Ankitha, G. Yojitha, M. U. Ahmed, B. Srikanth, and A. P. Kumar, “Design and Implementation of Optimized Inexact Signed Wallace Tree Multiplier Using Reversible Logic Gates and 4:2 Compressors,” 2025 6th International Conference for Emerging Technology (INCET), pp. 1–6, May 2025, doi: 10.1109/incet64471.2025.11140852.

4. S. Raveendran, P. J. Edavoor, Y. B. N. Kumar, and M. H. Vasantha, “Inexact Signed Wallace Tree Multiplier Design Using Reversible Logic,” IEEE Access, vol. 9, pp. 108119–108130, 2021, doi: 10.1109/access.2021.3100892.
5. T. S. Rao, C. Srinivasu, and K. Babulu, “Hierarchical Design of Higher-Order Efficient Wallace Tree Approximate Multipliers with Innovative Compressors,” 2024 IEEE International Conference on Intelligent Signal Processing and Effective Communication Technologies (INSPECT), pp. 1–6, Dec. 2024, doi: 10.1109/inspect63485.2024.10896020.
6. S. T. Petla, V. S. Chowdam, P. Ponkala, T. Nichenametla, and P. Polturu,“Comparative Analysis of Energy-Efficient Approximate Multiplier and Wallace Tree Multiplier for Error-Tolerant Applications,” 2025 3rd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT), pp. 378–383, Mar. 2025, doi: 10.1109/dicct64131.2025.10986753.
7. B.-H. Wu, T.-C. Tsai, M.-H. Sheu, and C.-H. Chen, “Modified WallaceTree Architecture Design and Implementation for High-Speed Multiplier,” 2025 32nd IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 1–4, Nov. 2025, doi: 10.1109/icecs66544.2025.11270474.
8. K. Ramakrishnan, A. N. K. Reddy, A. Chandu, and A. Anbarasan, “Compressor-Based Precise Blended Wallace Tree Multiplier Architecture for Signal Processing Applications,” 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), pp. 1–6, Dec. 2022, doi: 10.1109/icerect56837.2022.10060511.
9. E. J. Rao and P. Samundiswary, “Efficient Design of Modified Wallace Tree Approximate Multipliers Based on Imprecise Compressors for Error-Tolerance Applications,” Arabian Journal for Science and Engineering, vol. 49, no. 3, pp. 4253–4270, Oct. 2023, doi: 10.1007/s13369-023-08365-9.
10. S. Soni and A. Chauhan, “Efficient VLSI architecture for FIR filter using multiple Vedic multiplier and Kogge–Stone adder: A review,” Journal of Circuit Technology (JCT), vol. 12, pp. 1–5, 2023.
11. M. V. D. Reddy, M. Raaj A T, N. Kumar, A. A, L. S, and S. Kumar T R, “A Hybrid Approach to Optimized n-bit Multiplier Design on FPGA Leveraging Recursive-Wallace Tree with AI Centric Enhancements for FIR Filters and Neural Network Inference,” 2023 International Conference on Next Generation Electronics (NEleX), pp. 1–6, Dec. 2023, doi: 10.1109/nelex59773.2023.10421162.
12. S. Janwadkar and R. Dhavse, “ASIC implementation of ECG denoising FIR filter by using hybrid Vedic–Wallace tree multiplier,” International Journal of Circuit Theory and Applications, vol. 52, no. 4, pp. 1621–1646, Nov. 2023, doi: 10.1002/cta.3844.
13. High Speed and Efficient Wallace Tree Multiplier for Dsp Applications,” Journal of VLSI circuits and systems, vol. 2, no. 1, Jan. 2020, doi: 10.31838/jvcs/02.01.
14. E. J. Rao and P. Samundiswary, “Efficient Design of Modified Wallace Tree Approximate Multipliers Based on Imprecise Compressors for Error-Tolerance Applications,” Arabian Journal for Science and Engineering, vol. 49, no. 3, pp. 4253–4270, Oct. 2023, doi: 10.1007/s13369-023-08365-9.
15. I. Rizos, G. Papatheodorou, and A. Efthymiou, “Designing Approximate Reduced Complexity Wallace Multipliers,” Electronics, vol. 14, no. 2, p. 333, Jan. 2025, doi: 10.3390/electronics1402033.