International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 15, No. 1, March 2026, pp. 248~258
ISSN: 2089-4864, DOI: 10.11591/ijres.v15.i1.pp248-258

Design and implementation of a novel approximate carry look-ahead adder for low-power FIR filter applications

Badiganchela Shiva Kumar[1], Galiveeti Umamaheswara Reddy[2]
[1]SVU College of Engineering, Sri Venkateswara University, Tirupati, India
[2]Department of Electronics and Communication Engineering, Sri Venkateswara University, Tirupati, India

Article Info

Article history:

Received Aug 10, 2025
Revised Jan 24, 2026
Accepted Feb 17, 2026

Keywords:

Approximate adder
Carry look-ahead adder
Finite impulse response filter
Multi-bit adder
Xilinx

ABSTRACT

Approximate computing is a low-power circuit design strategy that trades off computational accuracy for gains in speed, power efficiency, and area reduction. This approach achieves considerable power and area efficiency by introducing acceptable errors. The acceptable error in computation systems refers to a loss in accuracy that does not affect overall system performance. Approximate computing is mainly suitable for multimedia and signal-processing applications. In this work, a novel approximate carry look-ahead adder (CLA) based on logical level modification is proposed. The new carry prediction term is derived to reduce the overall propagation delay of the addition operation. The proposed multi-bit adder design uses a square root-based division method to partition the adder stages. Moreover, the proposed adder is applied in finite impulse response (FIR) filter implementation to evaluate the performance in real-time applications. The proposed adder and FIR filter are coded in Verilog and verified using the Xilinx simulator. The result shows that the proposed FIR filter achieves better results in terms of all parameters.

Corresponding Author:

Badiganchela Shiva Kumar
SVU College of Engineering, Sri Venkateswara University
Tirupati, Andhra Pradesh, India
Email: bshivakumaries@gmail.com

1. INTRODUCTION

Low-power VLSI devices are electronics that consume power at low rates [1], [2]. The major objective of low-power VLSI is to reduce power consumption without affecting the original function and speed of the digital circuits. There are two strategies followed in low-power VLSI design: logical-level alterations and circuit-level changes. In circuit-level modifications, the various circuit styles are used. The well-known circuit styles are adiabatic logic, transmission gate logic, and gate diffusion logic styles. In logical level modifications, techniques like K-map simplification, resource sharing, pipelining, and approximate computing are used.

Approximate computing is identified as a hopeful solution for modern computing systems [3]. This approach achieves considerable power and area efficiency by introducing acceptable errors. The acceptable error in computation systems refers to a loss in accuracy that does not affect overall system performance. Approximate computing is mainly suitable for multimedia and signal-processing applications [4].

Higher accuracy and precision are difficult to achieve in conventional systems because of higher power consumption and larger area requirements. Data path circuits are mainly affected by increased power consumption due to their higher bit. Any processor or controller's data path circuit is a central component where logical and arithmetic operations are carried out. Addition is one of these processes that is essential to optimizations. An adder is a digital circuit that performs the addition of numbers. In any system, the adder

unit is not only used for addition operations but also as a part of address generation and incrementing program counters [5]. The performance of the adder directly influences the overall speed and power consumption of the system. Various types of adders have been developed to optimize speed, power consumption, and area, including ripple carry adders (RCA), carry look-ahead adders (CLA), and carry save adders (CSA). Among these, the CLA is the fastest adder due to its ability to quickly determine carry bits by using parallel processing. However, the high speed is achieved with the penalty of increased power consumption and complexity. To address these challenges, we propose an innovative approach by introducing an approximate CLA based on gate-level modifications. The proposed design achieves the balance between speed, power efficiency, and computational accuracy.

2. RELATED WORK

Fan *et al.* [6] presented an adder using multi level circuit integration techniques. The proposed adder uses two transistors as a switch to vary the approximation level. The energy efficiency of the adder is analysed by applying a neural network in image processing applications.

Stefanidis *et al.* [7] proposed a synthesizer-based approximate parallel-prefix adders for signal processing applications. When compared to the existing adders, the synthesized adders achieve an average improvement in error frequency of 27% to 36% for random inputs and enhance image quality metrics by 8% to 42% in image filtering tasks. Kavand *et al.* [8] introduce a new type of 4:2 compressor using reconfigurable field-effect transistors. Unlike traditional CMOS transistors, reconfigurable field-effect transistors can be configured to operate as either an n-type or p-type transistor based on the control of a gate terminal. This reconfigurability provides significant advantages in terms of circuit design flexibility, area efficiency, and power consumption. In their paper, Seo and Kim [9] introduce a novel dual sub-adder-based approximate adder to reduce overall latency. The dual sub-adder divides accurate adder portions into two sub-adders with error compensation blocks. This compensation block reduces the approximation error from the approximate adder part.

The approximate adder with the features of the ability to switch between accurate and inaccurate operation modes is proposed by Jha *et al.* [10]. Also, the dual approximate strategy is established to fine-tune the approximate modes. Experimental results from the cadence tools show that the developed adder uses 39% and 52% less energy than the exact mirror adder. Zhang *et al.* [11] presented a design of an approximate compressor-based multiplier for noisy functions. The number of sums of product terms in the compressor is reduced using logic level modifications. Implementation results of image processing benchmarks show that the compressor-based multiplier reduces area requirements by 54% with an error rate of 1.8% in comparison with the accurate multiplier. Chen *et al.* [12] proposed a new design technique for square construction. The number of partial products needed in the squarer module is reduced by applying the Booth folding technique. The proposed squarer is applied in signal processing and clustering algorithms for validation. The result shows that the proposed squarer achieves a signal-to-noise ratio close to 30 dB with the power-delay product reduced by up to 51%.

The modified parallel prefix adder is introduced by Rosa *et al.* [13]. The proposed adder consists of approximate prefix operators to reduce area and delay in computations. The proposed adder shows better results with increased bit widths. Chu *et al.* [14] presented an approximate circuit design using majority logic gate functions. Further, the new type of error compensation block is introduced to avoid the propagation of inaccurate carry-out signals to higher-order blocks to increase the precision level. The concept of “Logic-In-Memory" is a design approach that integrates logic and memory functions within the same architecture. However, most of the energy is wasted during the writing process. Barla *et al.* [15] integrated an approximate self-write-termination logic into the adder design. Implementation results on a 16-bit adder show that writing and reading energy consumption is reduced by 90.56%. Zhang *et al.* [16] proposed a radix-4 Booth multiplier based on inaccurate majority logic. The number of partial product terms in multiplications is reduced by compressed full adders with majority logic. Likewise, Aizaz and Khare [17] introduce a truncation-based Booth multiplier with an error compensation block. The error compensation element is produced through selective alterations in the K-map to address the carry issue arising from the approximated portion. Execution results on neural network application show that the Booth multiplier accomplishes a 0.02% mean relative error deviation and a 12% reduction in the area-power product compared to the accurate one.

Alan and Henkel [18] constructed an approximate adder using probability theory. This strategy reduces overall carry chain delay. Tsai *et al.* [19] proposed an accuracy configurable multi-bit adder with a power gating mechanism. The power gating technique allows the elimination of leakage current by forming a virtual ground. The proposed adder achieves 3.7 times lower delay than the existing one with minimum error rates. In the same way, Amirany *et al.* [20] proposed an accuracy adaptive adder specifically for filter applications.

Roy *et al.* [21] proposed an adder named an approximate least significant bit (LSB) adder. This adder consists of two components: an accurate most significant bit (MSB) adder component and an inexact LSB adder component. The proposed adder was analysed and compared with other adders in terms of mean error distance, mean square error distance, and worst-case error.

In their study by Ebrahimi-Azandaryani *et al.* [22], a low-energy consumption block-based carry speculative approximate adder is introduced. This adder operates by dividing the adder into separate summation parts which can be selected from either carry propagate or parallel-prefix adders. The carry of each portion is speculated based on the block's input operands and those of the subsequent element. Furthermore, an error detection and recovery technique is implemented to enhance accuracy and reduce the output error rate.

A new bit truncation strategy is Frustaci *et al*. [23] for inaccurate circuit designs. The execution outcomes on signal and signal processing applications show that the developed truncation scheme reduces energy by 20% with insignificant delay and area overhead. Pashaeifar *et al.* [24] establish a reverse carry propagate adder in their paper. Unlike traditional adders, the proposed adder propagates the carry signal from the MSB to the LSB which gives the carry input signal higher priority than the output carry. This reverse propagation enhances stability under delay variations. The results show that using the proposed adder can achieve average improvements of 27% in delay, 6% in energy, and 31% in energy-delay-product with higher accuracy levels. Xu *et al.* [25] proposed a delay-adaptive self-configurable adder for low-power applications. The maximum level of accuracy is achieved without redundancy or correction blocks. Akbari *et al.* [26] presented a switchable CLA adder used for both accurate and error-tolerant applications. The carry precomputation terms are modified using a multiplexer to vary the computation accuracy. The results show that the energy reduces by up to 49% and the delay is 19% lower, respectively.

Schlachter *et al.* [27] introduce a method called gate-level pruning to systematically balance accuracy with reductions in area, power, and delay in digital circuits. This method selectively removes or cuts the elements to trade off accuracy. Experimental results on applying the discrete cosine transform in a video processing application show that the proposed approach reaches a 21% reduction in energy with a tolerable image quality degradation of 24 dB. Liu *et al.* [28] proposed an inexact Booth multiplier based on modified Booth encoding techniques. In the modified Booth, the modified Booth algorithm uses a specific encoding scheme to reduce the number of partial products. This scheme examines three consecutive bits of the multiplier at a time to determine the partial products.

3. PROPOSED MODIFIED APPROXIMATE CLA

In this work, an approximate CLA design is proposed for low-power applications. The proposed carry prediction logic considers only the last two adder stages to reduce the complexity. This method neglects the remaining terms which allows more efficient computation with a controlled loss in accuracy. The proposed approximate CLA is shown in Figure 1. Where CPB denotes the carry prediction block to predict the carry from past stages. To explain this, let's consider an adder segment where we focus on the last two stages of the segment for carry prediction. The carry prediction can be approximated using (1):

$$C_{approx} = a_{i+1}b_{i+1} + a_i b_i (a_{i+1} + b_{i+1}) \quad (1)$$

Where a_i and b_i are the ith bits of the two numbers being added. a_{i+1} and b_{i+1} are the (i+1) th bits of the two numbers being added. C_{approx} represents the approximate carry generated. In this equation -$a_{i+1}b_{i+1}$ generates a carry when both a_{i+1} and b_{i+1} are 1. $a_i b_i (a_{i+1} + b_{i+1})$ accounts for the propagation of carry from the ith stage to the (i+1) th stage.

To extend this concept to an n-bit adder, we divide the adder into square root segments and apply the above approximation within each segment. Let n be the total number of bits and s be the segment size. Each segment consists of s bits, and we focus on the last two stages of each segment for carry prediction.

For a given segment k (where k ranges from 0 to $\frac{n}{s}$- 1, the approximate carry for the segment can be expressed as (2):

$$C_{approx,k} = a_{(k+1)s-1}b_{(k+1)s-1} + a_{ks}b_{ks}(a_{(k+1)s-1} + b_{(k+1)s-1}) \quad (2)$$

The carry of every segment is calculated based on this approximation to reduce propagation delay. This prediction allows a method to ensure that the critical path delay of the adder is reduced to speed up the overall computation time. The truth table for two-stage prediction with the corresponding proposed approximation outputs given in Table 1. From the total combination, the proposed carry predictions show

error results for only six combinations out of thirty-two combinations. So, the error probability of the proposed prediction is very low.

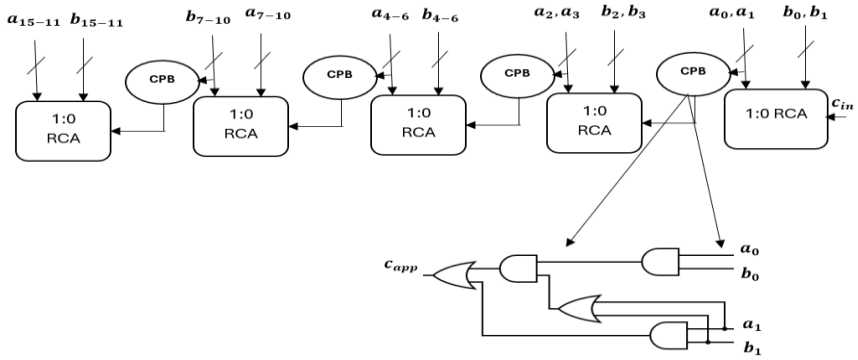

Figure 1. Proposed modified approximate carry look-ahead adder (MA-CLA)

Table 1. Truth table for approximate prediction

a_{i+1}	a_i	b_{i+1}	b_i	c_{i-1}	Original output	Approximated output
0	0	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	0
0	0	1	0	1	0	0
0	0	1	1	0	0	0
0	0	1	1	1	1	×
0	1	0	0	0	0	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	0	1	1	0	0
0	1	1	0	0	0	0
0	1	1	0	1	1	×
0	1	1	1	0	1	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	0	0	1	0	0
1	0	0	1	0	0	0
1	0	0	1	1	1	0
1	0	1	0	0	0	×
1	0	1	0	1	1	1
1	0	1	1	0	1	1
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	0	0	1	1	×
1	1	0	1	0	0	×
1	1	0	1	1	1	1
1	1	1	0	0	0	×
1	1	1	0	1	1	1
1	1	1	1	0	1	1
1	1	1	1	1	1	1

The overall design flow of the proposed adder is given in Figure 2. The proposed CLA divides the n-bit adder into √n segments and predicts carry using only the last two bits of each segment. These limits carry propagation within segments which results in O(√n) delay. The hardware grows linearly with bit width. and yields O(n) space complexity. The proposed CLA scales efficiently for large bit widths such as 256-bit and 512-bit adders due to its segment-based design. As the adder size increases, the number of segments increases proportionally and carry predictions remain localized within each segment. Consequently, the critical path delay grows as O(√n) rather than O(n) which maintains controlled delay increase. The maximum error is limited to one segment-level carry, and errors do not propagate across all bits due to segmentation. Hence, the relative error remains bounded and does not grow linearly with adder width.

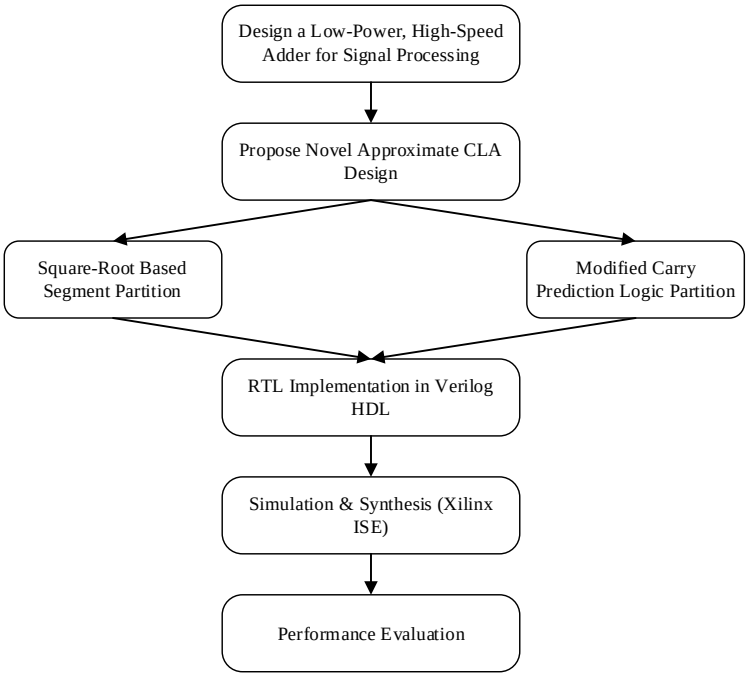

Figure 2. Proposed design flow

3.1. Finite impulse response implementation

One fundamental component of digital signal processing is the finite impulse response (FIR) filter. The impulse response of this filter has a finite duration. The majority of real-time applications for FIR filters include biomedical signal processing, image processing, speech processing, and signal processing in signals. It can also be used to create a robust and effective filter because of its unconditional stability and linear phase response. The ability to create filters with coefficients smaller than one, improved computing efficiency, ease of implementation, and lack of overflow oscillations are further advantages.

FIR filters can work in either continuous or discrete time. They are primarily implemented with windowing to remove unwanted noise and distortion. The general structure of an FIR filter can be represented by (3):

$$Y(n) = \sum_{i=0}^{N-1} h(i).X(n-i) \quad (3)$$

where, $Y(n)$ is the output of the filter, X is the input data, $h(i)$ is the filter coefficients, and N is the number of filter coefficients.

An FIR filter typically requires N-1 adders and N multipliers for its implementation. Figure 1 shows the structure of the FIR filter. It consists of N-1 multipliers, N adders and delay elements. The basic operation of the FIR filter involves calculating the transfer functions of the filter coefficients.

The implementation of FIR filters on field-programmable gate arrays (FPGAs) can consume significant area and resources. This complexity often results in high implementation costs. Multiplication and addition are a critical operation in FIR filters which directly influences the overall performance and speed of the filter. Efficient adder design is essential for achieving high-speed FIR filters, particularly in microprocessors and digital signal processing algorithms. Numerous adder designs have been developed to optimize performance. In this work, we focus on implementing a FIR filter using proposed approximate adder.

4. RESULT AND DISCUSSION

For evaluation, the proposed adder is implemented using Xilinx software. The design is coded in Verilog HDL. To analyze slice and look-up table (LUT), the design is implemented on Spartan3E FPGA XC3S100E device. The power analysis is carried out using the Xilinx XPower analyzer. The generated register transfer logic (RTL) view and gate-level netlist are given in Figure 3. The output waveform of the adder is shown in Figure 4. For fair comparison, the proposed adder is compared with previously proposed

adders like reconfigurable approximate carry look-ahead adder (RAP-CLA) [26], a low latency approximate adder design based on dual sub-adders with error recovery (LLAP-DSAWER) [29], and GDI-based accuracy configurable adder design (GDI-ACA) [30].

The measured values of the adder are given in Table 2. The slice and LUT results are increases linearly with increasing bit sizes. In RAP-CLA, the slice and LUT values are increased very little due to the insertion of a multiplexer for mode selection. But the delay is lower than RCA. In GDI-ACA, the delay performance is better than RAP-CLA. The proposed adder achieves excellent delay reduction when compared to all other designs. The slice and LUT count are nearly close to the hybrid adder. In summary, the proposed adder shows a reasonable and elevated results in terms of all metrics when compared to existing RCA, CLA, and hybrid adders. The results are graphically shown in Figure 5. The improvements are particularly achieved in terms of delay which makes the proposed adder a promising solution for high-speed arithmetic operations in FPGA implementations.

The RTL and Netlist of FIR are given in Figure 6. The implemented FIR filter uses a filter order of N=16 and uses a hamming window for coefficient generation. The input signal is sampled at 360 Hz with a 16-bit fixed-point representation. The results of FIR filter are shown in Figure 7. The measured values are given in Table 3.

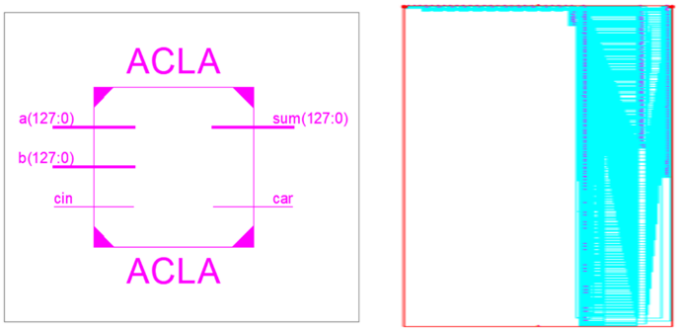

Figure 3. Register transfer logic and generated Netlist

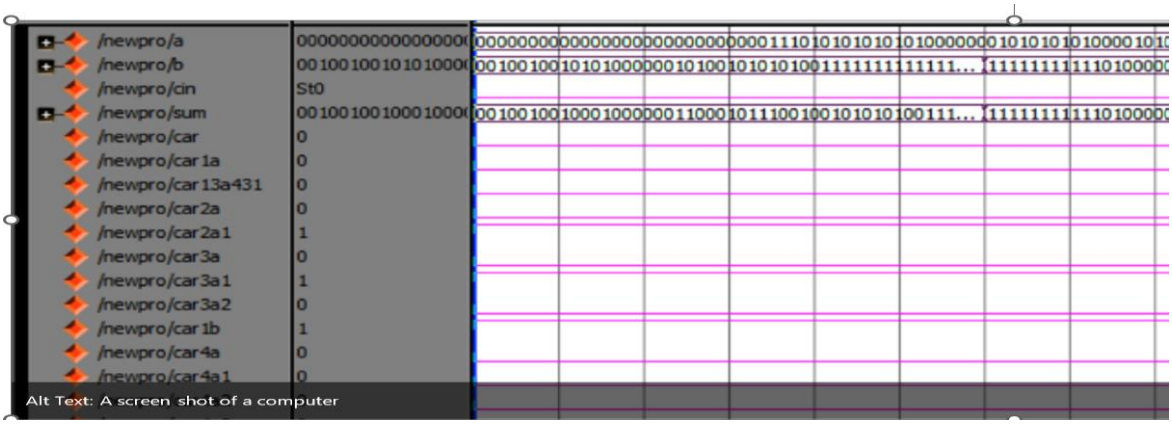

Figure 4. Output verification

Table 2. Performance analysis

	RCA				RAP-CLA				GDI-ACA				LLAP-DSAWER				MA-CLA			
	16	32	64	128	16	32	64	128	16	32	64	128	16	32	64	128	16	32	64	128
SLICE	18	37	73	148	19	36	78	152	18	36	74	140	12	33	67	133	18	36	72	138
LUT	32	64	127	285	34	63	144	292	25	60	121	255	21	57	117	231	32	64	128	244
DELAY	21.22	37.65	69.49	120.6	19.52	36.67	55.082	118.45	16.3	33.45	61.85	119.7	15.777	31.32	58.65	111.087	11.0	12.10	14.15	37.614

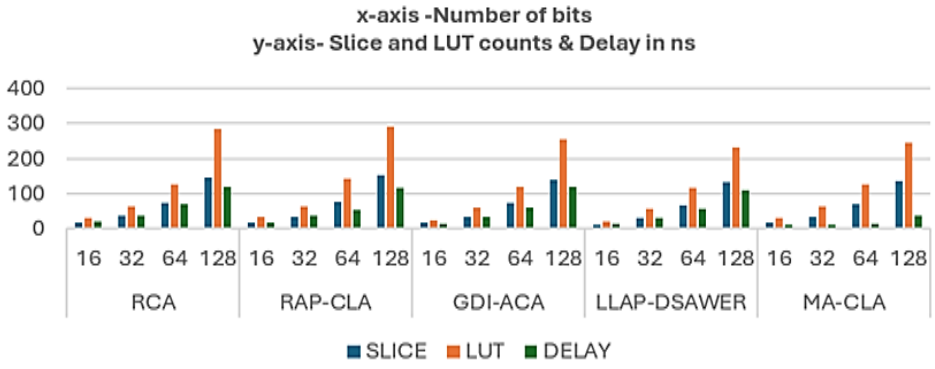

Figure 5. Performance analysis

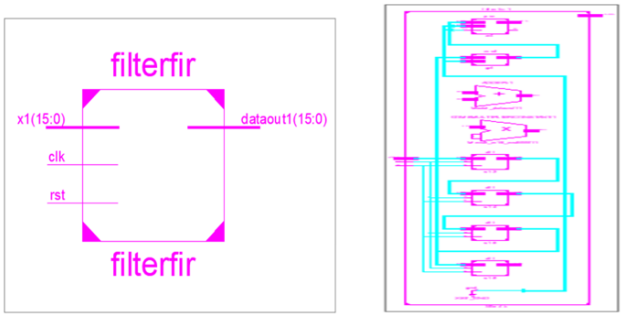

Figure 6. RTL and Netlist of FIR filter

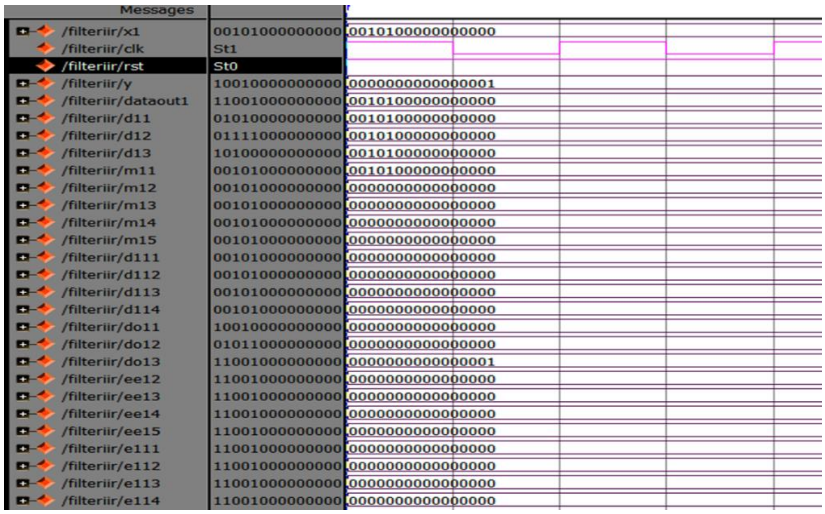

Figure 7. Simulated waveform

Table 3. FIR execution outputs

Parameter	FIR with a hybrid adder	FIR with proposed adders
Slice	288	265
LUT	497	433
Delay	31.266	18.453

The FIR filter using the hybrid adder consumes 288 slices. The FIR filter with the proposed adder achieves improved efficiency. It requires only 265 slices, which results in an 8% reduction in slice usage.

Also, the hybrid adder-based FIR filter uses 497 LUTs, whereas the proposed adder reduces this to 433 LUTs. It achieves a 13% reduction in LUT consumption. In terms of delay, the hybrid adder-based FIR filter exhibits a delay of 31.266 units, but the proposed adder-based FIR filter significantly improves upon this. It takes the delay of just 18.453 units.

For error analysis, N pairs of 128 random bits are generated. These pairs are denoted as (A_i, B_i) where i ranges from 1 to N. The exact and approximated 'sum' outputs are calculated, which denoted as $ExactSum_i$ and $ApproxSum_i$ respectively. The absolute error for a pair of inputs is calculated as (4):

$$AbsError_i = |ExactSum_i - ApproxSum_i| \quad (4)$$

The mean absolute error (MAE) of the adder can be computed as (5):

$$MAE = \frac{1}{N}\sum_{i=1}^{N} AbsError_i \quad (5)$$

The relative error is as (6):

$$RelError_i = \frac{|ExactSum_i - ApproxSum_i|}{ExactSum_i} \quad (6)$$

The mean relative error (MRE) is as (7) and (8):

$$MRE_i = \frac{1}{N}\sum_{i=1}^{N} RelError_i \quad (7)$$

$$MaxAbsError_i = \frac{1}{N}\sum_{i=1}^{N} AbsError_i \quad (8)$$

The standard deviation of the absolute error is computed as (9):

$$StdDevError_i = \sqrt{\frac{1}{N}}\sum_{i=1}^{N}(AbsError_i - MAE_i)^2 \quad (9)$$

The comparison between the hybrid adder and the proposed adder for 128-bit inputs is given in Table 4. The hybrid adder reaches the MAE of 15642. The proposed adder shows the lower value of 12347. The hybrid adder reaches the MRE of 0.012. The proposed adder shows the lower MRE value of 0.009. Additionally, the proposed adder exhibits a lower maximum absolute error of 76,543 and a smaller standard deviation of 4,321 which indicates more consistent performance. Overall, the proposed adder offers enhanced precision and reliability. The statistical confidence analysis of delay and error metrics is given in Table 5. The narrow confidence intervals indicate low variability across test cases. The p-values below 0.05 confirm that improvements in delay and error metrics are statistically significant and not due to random variation.

Table 4. Error analysis

Metric	Hybrid adder	Proposed adder
Number of test cases	10000	10000
MAE	15642	12347
MRE	0.012	0.009
Maximum absolute error	98456	76543
StdDevError	5432	4321

Table 5. Confidence intervals and significance analysis

Metric	Proposed adder (mean)	95% confidence interval	p-value	Interpretation
Delay (ns)	18.453	[17.92, 18.99]	<0.01	Statistically significant delay reduction
MAE	12347	[12110, 12584]	<0.05	Significant error improvement
MRE	0.009	[0.0086, 0.0094]	<0.05	Stable relative error
Max Abs error	76543	[74200, 78950]	<0.05	Bounded worst-case error

To evaluate the impact of approximation on signal quality, standard signal fidelity metrics are used. The results are given in Table 6. The signal-to-noise ratio (SNR), peak signal-to-noise ratio (PSNR), and mean squared error (MSE) between the exact and approximate FIR filter outputs are computed.

Table 6. Signal fidelity metrics

Metric	FIR with hybrid adder	FIR with proposed adder
SNR (dB)	38.6	37.9
PSNR (dB)	42.1	41.4
MSE	0.0028	0.0031

The results indicate only a marginal degradation in signal fidelity and maintain acceptable quality for DSP applications. The proposed CLA outperforms existing adders by reducing logic levels in the carry path and enabling parallel carry prediction across segmented blocks. This strategy significantly shortens the critical path delay. Unlike conventional CLA designs, the proposed CLA avoids complex global carry-generate networks which leads to faster computation. The primary trade-off is a slight increase in LUT usage due to carry prediction blocks; however, this overhead is marginal compared to the substantial delay reduction achieved, making the design highly efficient for high-speed and low-power DSP applications.

5. CONCLUSION

In this work, we presented a novel approximate CLA that integrates approximate computing logic to increase computational efficiency. The square root-based division method is applied for splitting adder stages. Then, the last two stages of each adder segment for carry prediction which simplifies the carry generation logic. To evaluate in real real-time scenario, the proposed adder is used in the FIR filter. The proposed adder not only reduces the critical path delay but also balances the trade-off between accuracy and performance.

FUNDING INFORMATION

The authors declare that no funding was received for the research described in this article.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Badiganchela Shiva Kumar	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Galiveeti Umamaheswara Reddy		✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," in *Proceedings - Design Automation Conference*, 2013, doi: 10.1145/2463209.2488873.
[2] S. Mittal, "A survey of techniques for approximate computing," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–33, May 2016, doi: 10.1145/2893356.

[3] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, "A Review, Classification, and Comparative Evaluation of Approximate Arithmetic Circuits," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 13, no. 4, pp. 1–34, Oct. 2017, doi: 10.1145/3094124.
[4] B. Sakthivel, K. Jayaram, N. M. Devarajan, S. M. Basha, and S. Rajapriya, "Machine Learning-Based Pruning Technique for Low Power Approximate Computing," *Computer Systems Science and Engineering*, vol. 42, no. 1, pp. 397–406, 2022, doi: 10.32604/CSSE.2022.021637.
[5] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1352–1361, Apr. 2017, doi: 10.1109/TVLSI.2016.2643003.
[6] X. Fan, T. Zhang, H. Li, H. Liu, S. Lu, and J. Han, "DACA: Dynamic Accuracy-Configurable Adders for Energy-Efficient Multi-Precision Computing," *IEEE Transactions on Nanotechnology*, vol. 22, pp. 400–408, 2023, doi: 10.1109/TNANO.2023.3297325.
[7] A. Stefanidis, I. Zoumpoulidou, D. Filippas, G. Dimitrakopoulos, and G. C. Sirakoulis, "Synthesis of Approximate Parallel-Prefix Adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 11, pp. 1686–1699, Nov. 2023, doi: 10.1109/TVLSI.2023.3287631.
[8] N. Kavand, A. Darjani, S. Rai, and A. Kumar, "Design of Energy-Efficient RFET-Based Exact and Approximate 4:2 Compressors and Multipliers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 9, pp. 3644–3648, Sep. 2023, doi: 10.1109/TCSII.2023.3275983.
[9] H. Seo and Y. Kim, "A Low Latency Approximate Adder Design Based on Dual Sub-Adders with Error Recovery," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 3, pp. 811–816, Jul. 2023, doi: 10.1109/TETC.2023.3270963.
[10] C. K. Jha, A. Nandi, and J. Mekie, "Single Exact Single Approximate Adders and Single Exact Dual Approximate Adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 7, pp. 907–916, Jul. 2023, doi: 10.1109/TVLSI.2023.3268275.
[11] M. Zhang, S. Nishizawa, and S. Kimura, "Area Efficient Approximate 4-2 Compressor and Probability-Based Error Adjustment for Approximate Multiplier," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 5, pp. 1714–1718, May 2023, doi: 10.1109/TCSII.2023.3257852.
[12] K. Chen, C. Xu, H. Waris, W. Liu, P. Montuschi, and F. Lombardi, "Exact and Approximate Squarers for Error-Tolerant Applications," *IEEE Transactions on Computers*, vol. 72, no. 7, pp. 2120–2126, 2023, doi: 10.1109/TC.2022.3228592.
[13] M. M. E. A. Da Rosa, G. Paim, P. U. L. Da Costa, E. A. C. Da Costa, R. I. Soares, and S. Bampi, "AxPPA: Approximate Parallel Prefix Adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 1, pp. 17–28, Jan. 2023, doi: 10.1109/TVLSI.2022.3218021.
[14] Z. Chu, C. Shang, T. Zhang, Y. Xia, L. Wang, and W. Liu, "Efficient Design of Majority-Logic-Based Approximate Arithmetic Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 12, pp. 1827–1839, Dec. 2022, doi: 10.1109/TVLSI.2022.3210252.
[15] P. Barla, V. K. Joshi, and S. Bhat, "Fully Nonvolatile Hybrid Full Adder Based on SHE+STT-MTJ/CMOS LIM Architecture," *IEEE Transactions on Magnetics*, vol. 58, no. 9, pp. 1–11, Sep. 2022, doi: 10.1109/TMAG.2022.3187605.
[16] T. Zhang *et al.*, "Design of Majority Logic-Based Approximate Booth Multipliers for Error-Tolerant Applications," *IEEE Transactions on Nanotechnology*, vol. 21, pp. 81–89, 2022, doi: 10.1109/TNANO.2022.3145362.
[17] Z. Aizaz and K. Khare, "Area and Power Efficient Truncated Booth Multipliers Using Approximate Carry-Based Error Compensation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 2, pp. 579–583, Feb. 2022, doi: 10.1109/TCSII.2021.3094910.
[18] T. Alan and J. Henkel, "Probability-Driven Evaluation of Lower-Part Approximation Adders," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 1, pp. 204–208, Jan. 2022, doi: 10.1109/TCSII.2021.3093984.
[19] K. L. Tsai, Y. J. Chang, C. H. Wang, and C. T. Chiang, "Accuracy-Configurable Radix-4 Adder with a Dynamic Output Modification Scheme," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 8, pp. 3328–3336, Aug. 2021, doi: 10.1109/TCSI.2021.3085572.
[20] A. Amirany, G. Epperson, A. Patooghy, and R. Rajaei, "Accuracy-Adaptive Spintronic Adder for Image Processing Applications," *IEEE Transactions on Magnetics*, vol. 57, no. 6, pp. 1–10, Jun. 2021, doi: 10.1109/TMAG.2021.3069161.
[21] A. S. Roy, R. Biswas, and A. S. Dhar, "On Fast and Exact Computation of Error Metrics in Approximate LSB Adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 876–889, Apr. 2020, doi: 10.1109/TVLSI.2020.2967149.
[22] F. Ebrahimi-Azandaryani, O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Block-Based Carry Speculative Approximate Adder for Energy-Efficient Applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 137–141, Jan. 2020, doi: 10.1109/TCSII.2019.2901060.
[23] F. Frustaci, S. Perri, P. Corsonello, and M. Alioto, "Energy-Quality Scalable Adders Based on Nonzeroing Bit Truncation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 964–968, Apr. 2019, doi: 10.1109/TVLSI.2018.2881326.
[24] M. Pashaeifar, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Approximate Reverse Carry Propagate Adder for Energy-Efficient DSP Applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 11, pp. 2530–2541, Nov. 2018, doi: 10.1109/TVLSI.2018.2859939.
[25] W. Xu, S. S. Sapatnekar, and J. Hu, "A simple yet efficient accuracy-configurable adder design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 6, pp. 1112–1125, Jun. 2018, doi: 10.1109/TVLSI.2018.2803081.
[26] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: A reconfigurable approximate carry look-ahead adder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 8, pp. 1089–1093, Aug. 2018, doi: 10.1109/TCSII.2016.2633307.
[27] J. Schlachter, V. Camus, K. V. Palem, and C. Enz, "Design and Applications of Approximate Circuits by Gate-Level Pruning," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1694–1702, May 2017, doi: 10.1109/TVLSI.2017.2657799.
[28] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing," *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1435–1441, Aug. 2017, doi: 10.1109/TC.2017.2672976.
[29] G. Anusha and P. Deepa, "Design of approximate adders and multipliers for error tolerant image processing," *Microprocessors and Microsystems*, vol. 72, no. 1, Feb. 2020,102940, doi: 10.1016/j.micpro.2019.102940.
[30] S. B. and P. A., "Area and delay efficient GDI based accuracy configurable adder design," *Microprocessors and Microsystems*, vol. 73, p. 102958, Mar. 2020, doi: 10.1016/j.micpro.2019.102958.

BIOGRAPHIES OF AUTHORS

Badiganchela Shiva Kumar received the B.Tech. degree in Electronics and Communication Engineering from V.I.T.S. Proddatur, Jawaharlal Nehru Technological University, Anantapur, India, in 2009, M.Tech. degree in Digital Electronics and Communication Systems from J.N.T.U.A.C.E. Pulivendula, Jawaharlal Nehru Technological University, Anantapur, India, in 2018. He is currently pursing research scholar at SVU College of Engineering, Sri Venkateswara University, Tirupati, India. His current research interests include through VLSI circuit design, device modeling, VLSI system design, and VLSI signal processing. He can be contacted at email: bshivakumaries@gmail.com.

Galiveeti Umamaheswara Reddy received B.Tech. degree in Electronics and Communication Engineering, M.Tech. in Instrumentation and Control Systems, and obtained Ph.D. from Sri Venkateswara University, Tirupati. He is a member of the ISTE, IEI, and BMESI. Currently, he is Professor at Department of Electronics and Communication Engineering, Sri Venkateswara University, Tirupati, Andhra Pradesh, India. He has a teaching experience of more than 28 years and has 69 technical publications in national/international journals and conferences. His areas of interest include VLSI design and signal processing. He can be contacted at email: umaskit@gmail.com.