Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

# High Performance and Area Efficient Approximate LUT Based Adder for Realtime Applications

**K. Jhansi Rani[1], K. Satya Sheela[2],**

[1]Asst. professor

*Department of ECE, UCEK, JNTU kakinada.*

*Kakinada, Andhra Pradesh, India.*

[2]M.Tech.

*Department of ECE, UCEK, JNTU kakinada.*

*Kakinada, Andhra Pradesh, India.*

***Abstract***— In this project, high speed, power efficient adder was designed at the cost of accuracy. The proposed design fixes the LSP of the implementation which will truncate half of the adder area and there by reduces the power consumption along with that the propagation delay can be minimized to half of it and further can be optimized by inner stage input-output pipelining to the adder to increase its speed. The MSP utilizes the accurate computation utilizes LUT's as resources and reduces the parameters by slightly increasing the error of approximate adders. Compared to the existing LEADx and APEx now our suggested adder is efficient in parameters and the design and simulation and effectiveness of the proposed method is synthesized using Xilinx Vivado.

***Keywords***— *Approximate computing, approximate adder, FPGA, low speed, low power, LUT*

## 1. Introduction

"In applications where errors can be accommodated, such as video encoding, approximate computing serves as a design approach that exchanges accuracy for enhancements in speed, compactness, and energy efficiency." FPGAs, despite their flexibility, consume more power than ASICs, making them a target for energy-efficient computing. Due to the demise of Moore's law, approximate computing is gaining popularity and has the potential to improve FPGA performance. This approach involves trading accuracy for gains in critical path delay, area, power, and energy consumption. Many applications, including image and video processing, are naturally error-tolerant. Approximation techniques can be applied at various levels, from hardware (voltage over-scaling, functional approximation) to software (loop perforation, function approximation). However, direct application of ASIC-based approximation principles to FPGAs doesn't work due to architectural differences.

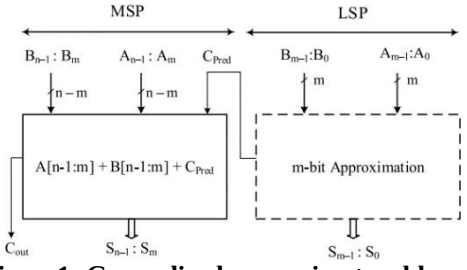


**Figure1: Generalized approximate adder**

FPGAs are commonly used for digital signal processing (DSP) tasks, where slight numerical imprecision is acceptable. Approximation may be used throughout the board, from the logic to the architecture to the algorithm, to maximise power efficiency. For instance, using approximate hardware for computationally intensive blocks in image and video compression can substantially reduce power consumption without significant quality loss. There are various approximate circuits, including adders, which are fundamental in digital hardware. Approximate adders are categorized into segmented, speculative, and full-adder based types. Most existing designs are tailored for ASICs and may not perform as well on FPGAs due to different logic implementation methods. FPGAs rely on lookup tables (LUTs) for logic functions, unlike ASICs that use logic gates. FPGA-specific approximate adder designs are relatively scarce, but they hold promise for improving efficiency and accuracy. In this paper, a methodology is proposed to enhance FPGA-based approximate adders by leveraging unused LUT inputs. LEADx and APEx are two new approximation adders designed for use with FPGAs. When compared to other approximation adders, LEADx provides superior video encoding quality by reducing mean square error (MSE). APEx, while having a slightly higher MSE than LEADx, outperforms existing designs in terms of area, power consumption, and quality.

## 2. Proposed Method

In the proposed method, approximate full adders based on FPGA LUT's has been proposed. Two designs are proposed one establishes a tradeoff between power and accuracy and the other on area and accuracy. The n-digit viper design found in Fig. is used as the basis for the suggested design process. The generalized approximation adder separates n-bit addition into LSP and MSP implementations, each of which is n-bits accurate or about so.

If the carry chain is broken at location m, the final total will often be off by a factor of 2m. Anticipating the carry-in to the MSP (CMSP)all the more unequivocally and changing the rationale capability of LSP to make up for the misstep are the two methods for bringing down the mistake rate and blunder size. Any k-bit input pairs from the approximation portion, where k m, may be used to estimate the carry to the correct section. k = 1 is used by the vast majority of current approximation adders.

Each 6-input LUT in an FPGA version of a precise adder contributes just two inputs and one output. We suggest using the first MSP LUT's extra four inputs for CMSP prediction, which are currently underutilized. As a result, we suggest combining the LSP and MSP for carry prediction by sharing the two most important bits of each input. The error rate in forecasting CMSP may be decreased by sharing more of the LSP with the MSP. However, this will cause the estimated adder's area and delay to grow.

Objective: investigate how changing the value of k affects the performance and accuracy of an approximation adder implemented on a field-programmable gate array (FPGA). For k > 2, there is a little reduction in the error rate, but it comes at the expense of more space and more time. For k 2, on the other hand, there is a little reduction in latency but a substantial rise in mistake rate. Because of the trade-off between accuracy and performance in approximation adders for FPGAs, we recommend setting k = 2.

While utilizing the recommended estimate adders, on the off chance that a convey happens at bit position m 1, or on the other hand assuming a convey happens at bit position m 2 and is engendered at bit position m 1, the convey is shipped off the MSP. Gi and Pi are the signs made and spread at the ith bit position, individually, and (4) portrays the CMSP. In Fig. below, you can see the general layout of the approximation adders we suggest.

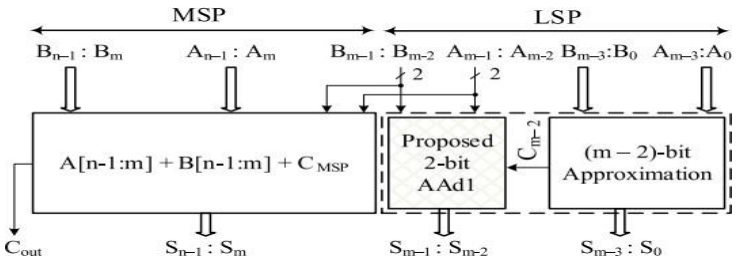


**Figure4: Proposed approximate adder**

The CMSP is predicted using the two most significant bits (MSBs) of the LSP, with the total bits for each being calculated using AAd1. AAd1 is appropriate only when a reliable prediction can be made for the Cout of 2-bit

inputs. Predicting Cout with high accuracy requires either more manpower or the utilization of spare LUT inputs. Accordingly, the most un-critical m 2 pieces of the LSP are avoided with regard to AAd1 while planning region proficient guess adders for FPGAs. Utilizing the design displayed in Fig, we propose two approximation adders with n bits of precision.

The principal m 2 pieces of the LSP are approximated involving two unmistakable capabilities in the two proposed n-bit guess adders. Modern FPGAs make use of LUTs with six inputs. Two 5-input tasks might be executed utilizing these LUTs. The presentation of a LUT-based arrangement is unaffected by the intricacy of the executed rationale capability. Two-bit adders take in five inputs and spit out two results. This means that an approximate 2-bit adder may be built using a LUT.

We propose the partitioning of the initial m/2 bits of LSP into d(m/2)e groups of 2-bit inputs, and each group is associated with a single LUT to achieve a more space-efficient FPGA implementation. Within each subset, we employ an approximate 2-bit adder (AAd2) to combine two 2-bit inputs along with a carry-in. Our approach to eliminate the carry chain within LSP involves setting the Cout of the ith group equal to one of the ith group's inputs (Ai+1). In 8 out of 32 scenarios, this results in an error, which exhibits an absolute error magnitude of 4. We suggest the following method for computing the Si and Si+1 output bits, which should result in a less inaccuracy:

- Standard 2-bit addition guarantees accuracy if the Cout prediction is right, and so do the summed outputs.
- If the predicted value of Cout is 0, meaning an inaccurate prediction was made, both sum outputs will beset to 1..

In the case when an inaccurate prediction of Cout results in a value of 1 for that variable, both sum outputs are zeroed out.

The absolute size of the mistake has been reduced from 4 to 2 in two situations and from 6 to 1 in the remaining six. The resultant AAd2 truth table is shown in Table. Red indicates an erroneous condition. The mistake probability of AAd2 is 0.25 as it gives a wrong answer in 8 out of 32 tests.

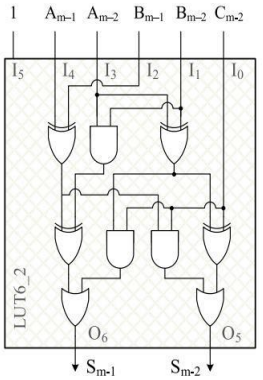


**Figure2:** AAd1

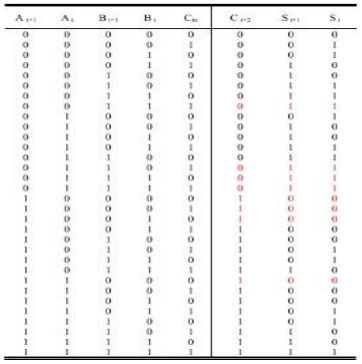

**Table1:** AAd2

## 3. LEADX

LEADx is an innovative FPGA-specific approximate adder designed to optimize computational efficiency while maintaining reasonable accuracy. It stands out by minimizing the mean square error (MSE), a measure of the discrepancy between approximate and accurate computation results. LEADx has been engineered to excel in video encoding applications, where maintaining a certain level of quality is crucial for a good user experience. Through a carefully crafted methodology that takes advantage of FPGA resources, including unused LUT (lookup table) inputs, LEADx achieves a balance between computational speed and output quality, making it a valuable

contribution to the field of approximate computing for FPGA-based systems. Its ability to reduce error while optimizing FPGA resources positions LEADx as a promising solution for energy-efficient and high-performance digital hardware implementations in applications where approximation is acceptable.

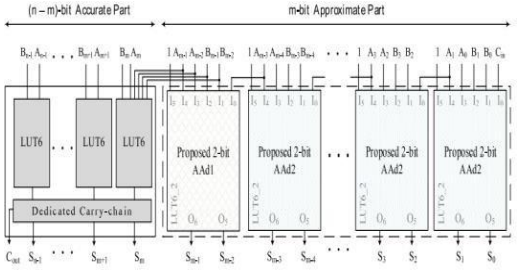


**Figure2:** LEADx

## 4. APEX

APEx, another FPGA-specific approximate adder, is engineered with a focus on achieving efficient area and power consumption while maintaining acceptable levels of accuracy. Although it may exhibit a slightly higher mean square error (MSE) compared to LEADx, APEx has demonstrated superior performance when compared to existing approximate adder designs. It manages to strike a balance between computational efficiency and output quality, making it a compelling choice for FPGA-based systems, especially in scenarios where minimizing area and power consumption are critical concerns. APEx's ability to outperform other approximate adders in terms of area, power, and error rates positions it as a valuable solution for enhancing energy efficiency and computational performance in FPGA-based applications, such as digital signal processing and multimedia processing.

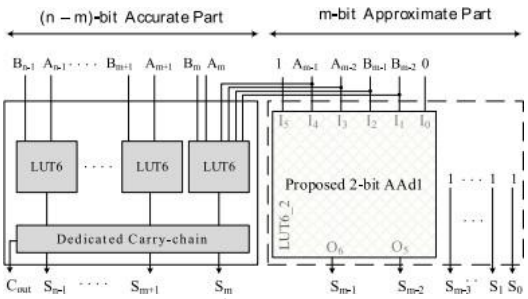


**Figure3:** APEx

To "pipeline" is to accumulate processor instructions in a pipeline. The system provides a methodical place for storing and running programmers. You may recognize this as "pipeline processing" as well. In systems with no feedback loops, increasing concurrency may be as simple as using pipelining. The term "pipelining" refers to a method of processing where numerous instructions run simultaneously. The stages of a pipeline are joined to one another to produce the pipe's overall shape. The directions enter at one end and leave at the other. As a result of pipelining, more lessons may be taught per unit of time.

## 5. Proposed power efficient adder

When an instruction stream from a processor is accumulated in a pipeline, it may be executed in one continuous operation. Instructions may be stored and run sequentially. In other words, it's a kind of pipeline processing. Concurrency in feedback-free systems may be easily increased via the use of pipelining. With the use of pipelining, many instructions may be executed simultaneously. Pipelines are constructed from a series of interconnected stages, thus the name. From one end, instructions are given and then emitted. Pipelining increases the rate at which instructions may be processed.

Each leg of the pipeline system is made up of an input register and a combinational circuit. Combinatorial circuits operate on data stored in the register. The combinational circuit's output is connected to the next segment's input

register. When latches are strategically placed at feed-forward points in a pipeline, the critical path length may be decreased. In order to maximize the system's efficiency, this decrease should be used to its full potential. Pipelining, in other terms, is a technique whereby a topology (with no feedback loops) is converted into a form more suited to a high-speed application when the original topology is inadequate for that purpose. The input-output delay, also known as system latency, is increased by latches, while the critical path is shortened thanks to pipelining.

After applying pipelining to the proposed power efficient adder, the circuit can be a shown as below figure

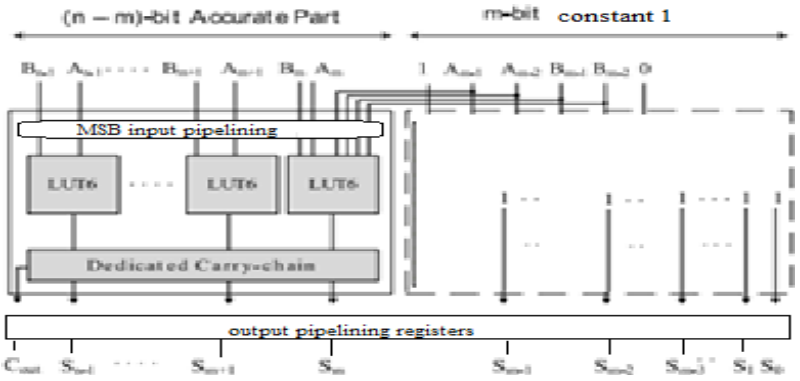


**Fig4: Proposed power efficient adder**

- In this approach, ignored to reduce area, delay and power compared to leadx and apex by considering constant “1” at the LSB side and computing accurate addition at the MSB part. This leads to high error than LEADx and Apex but results in reduction of power
- "In each section of a pipeline system, a combinational circuit comes after an input register. “The combinational circuit stores data in the register and processes it later. A combinational circuit's output issent into the first register of the subsequent sub-section.
- Putting latches in the right feed-forward positions is one way in which pipelining might shorten the critical path. This decrease may be used to run the system more quickly.

## 6. Results

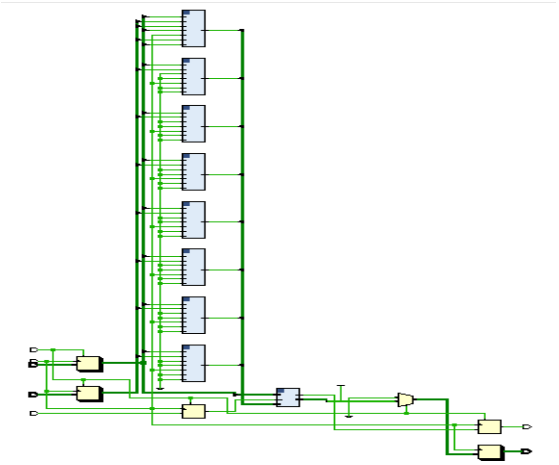

**Figure5:RTL schematic**

egister transfer level is a level of abstraction in digital circuit design. RTL schematic describes the logical

operation of a digital circuit at the level of individual registers and the data transfers between them. It depicts the movement of data between registers and the logical operations that are carried out on that data. RTL schematics are crucial in designing and understanding the functionality of digital circuits. They are used to specify the behavior of digital system before they are synthesized into hardware components.

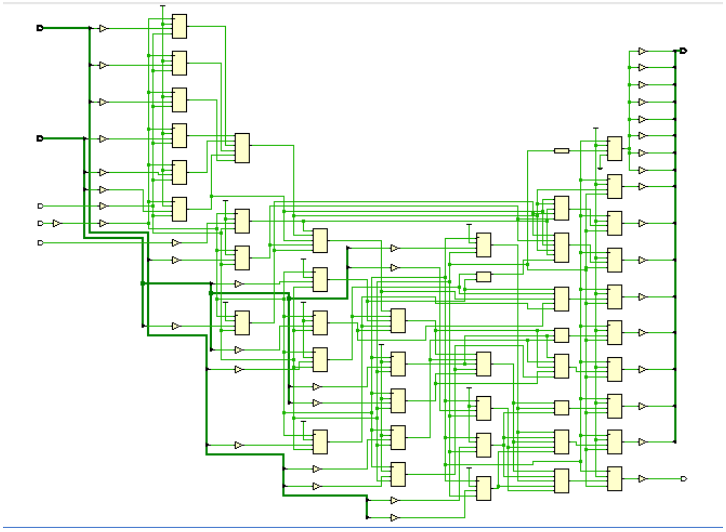

**Figure6:Technology schematic**

A technology schematic typically refers to a diagram representation that outlines the physical layout and connections of components with in a specific technology or a system.

These technology schematics are essential tools for engineers, designers and technicians

To understand , plan and troubleshooting complex technological systems. Here there is brief overview key elements and uses of technology schematic. They are components, connections, functionality, power supply, , documentation ,signal flow, troubleshooting, design and development, education, integration.

## 7. SIMULATION RESULTS

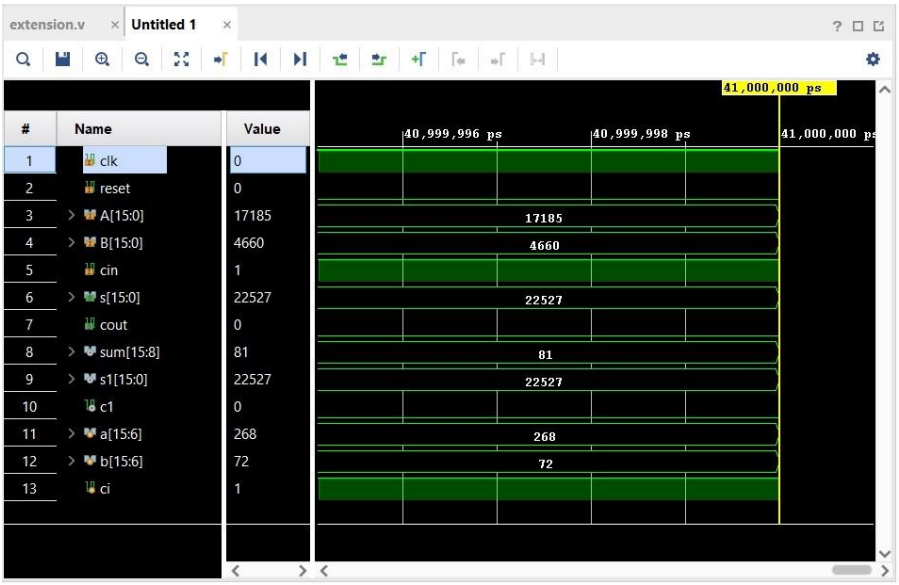

**Table1: I/O Table**

| **Name** | **value** |
|---|---|
| clock | **1** |
| Reset | **0** |
| A | **17985** |
| B | **4660** |
| Cin | **1** |
| Sum[15:0] | **22527** |
| Cout | **0** |

**Table2:** Area

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 10 | 20800 | 0.05 |
| FF | 31 | 41600 | 0.07 |
| IO | 40 | 106 | 37.74 |

```
Max Delay Paths
--------------------------------------------------------------------------------
Slack:                  inf
  Source:               b_reg[6]/C
                          (rising edge-triggered cell FDRE)
  Destination:          cout_reg/D
  Path Group:           (none)
  Path Type:            Max at Slow Process Corner
  Data Path Delay:      4.067ns  (logic 1.235ns (30.366%)  route 2.832ns (69.634%))
  Logic Levels:         6  (FDRE=1 LUT5=4 LUT6=1)
```

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

| | |
|---|---|
| **Total On-Chip Power:** | **6.262 W** |
| **Design Power Budget:** | **Not Specified** |
| **Power Budget Margin:** | **N/A** |
| **Junction Temperature:** | **56.3°C** |

**Table2: Evaluation table**

| | Area | Power | Delay |
|---|---|---|---|
| LEADx | 14 | 10.272 | 7.562 |
| APEx | 11 | 7.543 | 7.562 |
| Proposed | 10 | 6.262 | 4.067 |

## 8. Conclusion

This work proposes low-delay, power-efficient approximate adders for FPGAs, where approximation is performed only in the LSP using constant value assignment to bits while the MSP maintains accuracy. When compared to LEADx adder solutions, the size and power efficiency of the constant bit approximation adder are much higher. The second technique to approximate adder, in addition to addition assigning constant bit to LSPs, pipelining has reduced delay than the leadx and apex adders. It uses less power and has less of a delay than other approximation adders do now. Because of their great efficiency and low latency, the suggested approximation adders are well-suited for usage in FPGA implementations of error-tolerant applications.

## References

[1] G. A. Gillani, M. A. Hanif, B. Verstoep, S. H. Gerez, M. Shafique, and A. B. J. Kokkeler, ‘‘MACISH: Designing approximate MAC accelerators with internal-self-healing,’’ IEEE Access, vol. 7, pp. 77142–77160, 2019.

[2] E. Kalali and I. Hamzaoglu, ‘‘An approximate HEVC intra angular prediction hardware,’’ IEEE Access, vol.8, pp. 2599–2607, 2020.

[3] T. Ayhan and M. Altun, ‘‘Circuit aware approximate system design with case studies in image processing andneural networks,’’ IEEE Access, vol. 7, pp. 4726–4734, 2019.

[4] W. Ahmad and I. Hamzaoglu, ‘‘An efficient approximate sum of absolute differences hardware for FPGAs,’’in Proc. IEEE Int. Conf. Consum. Electron. (ICCE), Las Vegas, NV, USA, Jan. 2021, pp. 1–5.

[5] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, ‘‘A review, classification, and comparative evaluation of approximate arithmetic circuits,’’ ACM J. Emerg. Technol. Comput. Syst., vol. 13, no. 4, pp. 1–34, Aug. 2017.

[6] A. C. Mert, H. Azgin, E. Kalali, and I. Hamzaoglu, ‘‘Novel approximate absolute difference hardware,’’ in Proc. 22nd Euromicro Conf. Digit. Syst. Design (DSD), Kallithea, Greece, Aug. 2019, pp. 190–193.

[7] N. Van Toan and J.-G. Lee, ‘‘FPGA-based multi-level approximate multipliers for high-performance error-

resilient applications,'' IEEE Access, vol. 8, pp. 25481–25497, 2020.

[8] L. Chen, J. Han, W. Liu, P. Montuschi, and F. Lombardi, ''Design, evaluation and application of approximatehigh-radix dividers,'' IEEE Trans. Multi-Scale Comput. Syst., vol. 4, no. 3, pp. 299–312, Jul. 2018.

[9] A. K. Verma, P. Brisk, and P. Ienne, ''Variable latency speculative addition: A new paradigm for arithmetic circuit design,'' in Proc. Design, Automat. Test Eur. (DATE), Munich, Germany, Mar. 2008, pp. 1250–1255.

[10] N. Zhu, W. L. Goh, G. Wang, and K. S. Yeo, ''Enhanced low-power highspeed adder for error-tolerant application,'' in Proc. Int. SoC Design Conf., Incheon, South Korea, Nov. 2010, pp. 323–327.

[11] A. B. Kahng and S. Kang, ''Accuracy-configurable adder for approximate arithmetic designs,'' in Proc. 49thAnnu. Design Automat. Conf. (DAC), New York, NY, USA, 2012, pp. 820–825.

[12] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, ''Low-power digital signal processing using approximate adders,'' IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan.2013.

[13] W. Ahmad, B. Ayrancioglu, and I. Hamzaoglu, ''Comparison of approximate circuits for H.264 and HEVC motion estimation,'' in Proc. 23rd Euromicro Conf. Digit. Syst. Design (DSD), Kranj, Slovenia, Aug. 2020, pp. 167–173.

[14] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, ''Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft computing applications,'' IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[15] A. Dalloo, A. Najafi, and A. Garcia-Ortiz, ''Systematic design of an approximate adder: The optimized lowerpart constant-OR adder,'' IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 8, pp. 1595–1599, Aug.2018.

[16] P. Balasubramanian, R. Nayar, D. L. Maskell, and N. E. Mastorakis, ''An approximate adder with a near-normal error distribution: Design, error analysis and practical application,'' IEEE Access, vol. 9, pp. 4518–4530,2021.

[17] S. Dutt, S. Nandi, and G. Trivedi, ''Analysis and design of adders for approximate computing,'' ACM Trans.Embedded Comput. Syst., vol. 17, p. 40, Dec. 2017.

[18] D. Celia, V. Vasudevan, and N. Chandrachoodan, ''Optimizing poweraccuracy trade-off in approximate adders,'' in Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE), Dresden, Germany, Mar. 2018, pp. 1488– 1491.

[19] A. Becher, J. Echavarria, D. Ziener, S. Wildermann, and J. Teich, ''A LUT based approximate adder,'' in Proc. IEEE 24th Annu. Int. Symp. Field Program. Custom Comput. Mach. (FCCM), Washington, DC, USA, May2016, p. 27.

[20] B. S. Prabakaran, S. Rehman, M. A. Hanif, S. Ullah, G. Mazaheri, A. Kumar, and M. Shafique, ''DeMAS: An efficient design methodology for building approximate adders for FPGA-based systems,'' in Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE), Dresden, Germany, Mar. 2018, pp. 917–920.

[21] S. Boroumand, H. P. Afshar, and P. Brisk, ''Approximate quaternary addition with the fast carry chains of FPGAs,'' in Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE), Dresden, Germany, Mar. 2018, pp. 577– 580.