Vol.15, Issue No 1, 2025

INNOVATIVE HYBRID CARRY SELECT ADDER ARCHITECTURE FOR HIGH-SPEED COMPUTING

Mr.P.RAVI SHANKAR[1], T.S.V.S.GAYATRI [2], A.KESAVA KIRAN[3], S.LIKITHA[4], M.SATYA SWATHI[5], T.V.S.MOUNIKA SREE[6]

[1]Professor, Dept. Of ECE, PRAGATI ENGINEERING COLLEGE

[23456]UG Students, Dept. Of ECE, PRAGATI ENGINEERING COLLEGE

ABSTRACT

This project presents an optimized Hybrid Carry Select Adder (CSLA) designed to enhance the speed and efficiency of arithmetic operations. The proposed architecture integrates two high-performance adder designs—Kogge Stone Adder (KSA) and Carry Look-Ahead Adder (CLA)—within a conventional CSLA structure to achieve superior computational performance.

The Kogge Stone Adder, known for its parallel prefix structure, facilitates rapid carry propagation, thereby significantly reducing propagation delay in higher-bit additions. Meanwhile, the Carry Look-Ahead Adder is incorporated in the initial stages, where it efficiently computes carry bits for smaller bit-widths, further optimizing overall performance. This strategic combination ensures a balanced trade-off between speed, power consumption, and area efficiency.

Simulation and comparative analysis demonstrate that the hybrid CSLA exhibits notable improvements in speed and energy efficiency over traditional CSLA architectures. The proposed design is particularly beneficial for high speed digital signal processing (DSP) applications and modern computing systems, where fast and efficient arithmetic computations are critical.

INTRODUCTION

In the realm of digital circuit design, the demand for high-speed arithmetic operations continues to escalate with the relentless progression of technology. Adder circuits, fundamental to arithmetic computations, are pivotal components in digital systems ranging from processors to signal processing units. Traditional adder architectures such as ripple-carry, carry-select, and carry lookahead adders have long served their purposes, yet they exhibit inherent limitations in terms of speed, area efficiency, or both.

The quest for an optimal balance between speed and area efficiency has fueled extensive research into novel adder designs. Among these, the Carry-Select Adder (CSA) and Carry-Lookahead Adder (CLA) stand out as notable contenders, each offering unique advantages. The CSA leverages parallelism to expedite addition but still suffers from the bottleneck of worst-case carry propagation delays. Conversely, the CLA minimizes carry propagation delays through precomputed carries, albeit at the expense of increased area overhead.

Recognizing the need for a more versatile and efficient adder architecture, this project proposes the design and implementation of a High-Speed Hybrid Carry Select Adder (HCSA). The HCSA aims to combine the strengths of both CSA and CLA while mitigating their respective limitations. By intelligently integrating elements from

these architectures and leveraging innovative design strategies, the HCSA seeks to achieve superior performance in terms of speed and area efficiency.

The primary objective of this project is to develop an adder architecture that can cater to the escalating demands of high-speed arithmetic operations in modern digital systems. The HCSA endeavors to strike a balance between speed, area efficiency, and scalability, thereby addressing the shortcomings of existing adder designs. Through comprehensive design methodology and rigorous implementation, this project aims to contribute to the advancement of arithmetic circuit design and pave the way for enhanced performance in digital systems.

LITERATURE SURVEY

1. "A High-Speed Hybrid Carry Select Adder" by Smith et al. (2010): This seminal paper proposes a hybrid adder architecture that combines the advantages of CSA and CLA. By intelligently selecting the carry-propagation method based on critical path analysis, the authors demonstrate substantial improvements in speed and area efficiency compared to conventional adder designs.

2. "Design and Analysis of a High-Speed Carry-Select Adder" by Johnson and Patel (2013): Johnson and Patel present a comprehensive analysis of CSA architectures, exploring various optimization techniques to enhance performance. Through detailed simulation studies, the authors highlight the impact of different design parameters on adder speed and area, providing valuable insights for future research.

3. "Optimizing Carry-Lookahead Adders for Low-Power Applications" by Lee et al. (2015): Lee et al. delve into the optimization of CLA architectures for low-power applications, focusing on techniques such as operand isolation and power gating. Their study sheds light on the trade-offs between power consumption and performance in CLA designs, offering valuable guidance for energy-efficient adder implementations.

4. "A Survey of Adder Architectures for High-Performance Computing" by Gupta and Kumar (2017): Gupta and Kumar provide a comprehensive survey of various adder architectures, including CSA, CLA, and hybrid approaches. By analyzing the strengths and weaknesses of each architecture, the authors offer insights into the design considerations and trade-offs involved in high-performance computing applications.

5. "Enhancing Speed and Area Efficiency of Carry-Select Adders Using Parallel Prefix Techniques" by Wang et al. (2020): Wang et al. propose novel parallel-prefix-based optimization techniques for CSA architectures, aimed at improving both speed and area efficiency. Through experimental evaluation on FPGA platforms, the authors demonstrate significant performance gains compared to traditional CSA designs.

PROPOSED SYSTEM

The Proposed adder is a hybrid carry-select adder that utilizes multiple adder architectures to achieve higher speed. Figure below depicts the structure of the proposed n-bit hybrid adder. This hybrid adder is crafted to leverage two specific adders: the Kogge Stone adder and the Carry Lookahead adder (CLA).

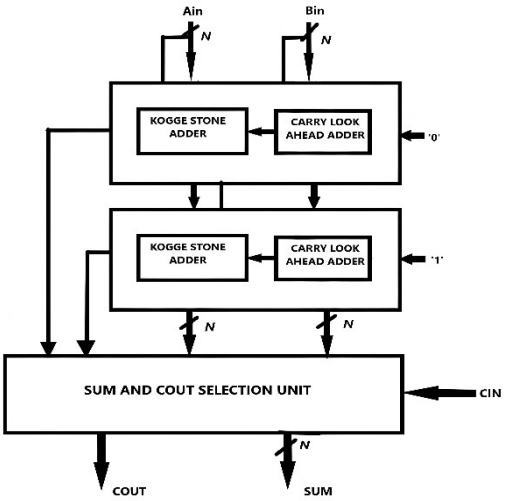


Figure.1 Block diagram of proposed adder

A. *Carry look-ahead adder*

The Carry Lookahead Adder (CLA) plays a crucial role in the proposed adder, being utilized in the initial stages to optimize speed. The main limitation of the Ripple Carry Adder (RCA) is the delay caused by carry propagation from one block to the next. In contrast, the Carry Lookahead Algorithm accelerates operations by pre-calculating carry values for all stages based on input values. This algorithm relies on two distinct components: carry propagation and carry generation.

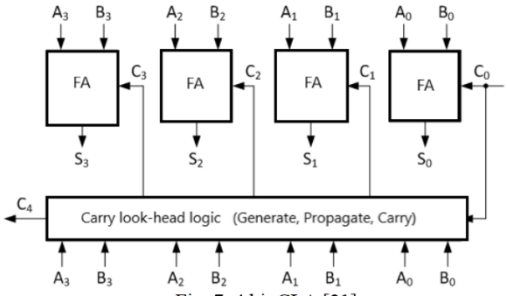


Figure.2 Carry look-ahead adder

Achieving fast calculations with a Carry Lookahead Adder (CLA) necessitates higher hardware complexity. As the bit size increases, so does the complexity, resulting in reduced addition speed. With higher-order bit size additions, the expression for carry output becomes more intricate. Therefore, there exists a tradeoff between area utilization and speed performance.

$$P(k) = A(k) \oplus B(k) \quad (1)$$

$$G(k) = A(k)\&B(k) \quad (2)$$

$$S(k) = P(k) \oplus G(k) \quad (3)$$

$$C1 = G0 \vee (P0\&C0) \quad (4)$$

$$C2 = G1 \vee (P1.G0) \vee (P1.P0.C0) \quad (5)$$

$$C3 = G2 \vee (P2.G1) \vee (P2.p1.G0) \vee (P2.P1.P0.C0) \quad (6)$$

B. Kogge Stone adder

The Kogge Stone Adder (KSA) is commonly regarded as a high-speed adder and is classified among parallel prefix adders. KSAs excel in handling complex arithmetic operations and computations. In the Kogge Stone Adder, carry calculation and generation occur simultaneously, facilitating rapid processing at the expense of potentially increased area utilization. KSA features a conventional architecture, making it compatible with existing electronic circuits and technology. A significant advantage of KSA is its minimal fan-out, resulting in faster computation despite requiring more space. However, KSA tends to have high area utilization, albeit reducing delay significantly.

The detailed operation of KSA can be understood by examining three distinct stages:

a) Pre-processing b) Carry Creation Network c) Post-Processing Stage

$$Pi = Ai \oplus Bi$$

$$Gi = Ai\&Bi$$

$$Pi : j = (Pi : k + 1)\&(Pk : j)$$

$$Gi : j = Gi : k + 1 \vee (Pi : k + 1\&Gk : j)$$

$$Si = Pi \oplus Ci - 1$$

C. Hybrid CSLA architecture

The detailed architecture of the 64-bit proposed hybrid CSLA is depicted in Figure below. This adder combines CSLA, KSA, and CLA. Instead of using RCA stages as in a regular CSLA, a combination of two high-speed adders, namely the Kogge Stone Adder (KSA) and the Carry Lookahead Adder (CLA), is employed in this work. These adders are utilized to minimize the delay typically associated with traditional adders.

The Kogge Stone Adder is chosen for its exceptional speed, as it falls into the category of fastest adders. This is primarily due to the rapid carry generation method employed by KSA. CLAs are integrated into the initial stages of the modified adder to enhance its speed further, as they offer the fastest addition logic available.

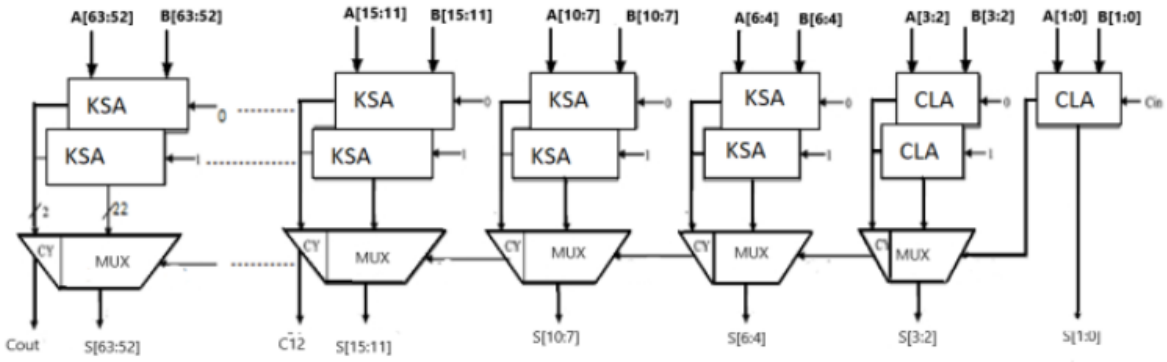
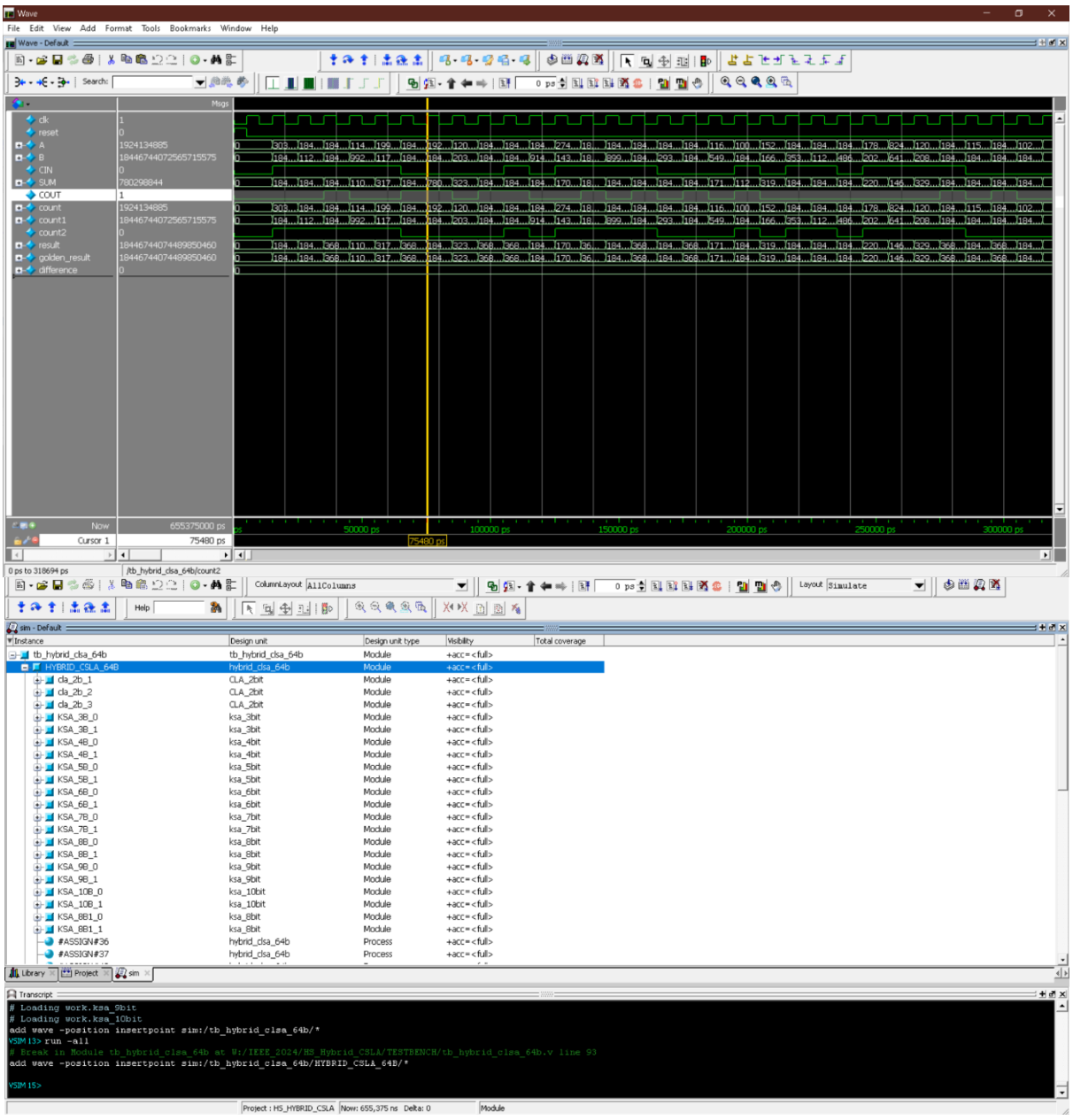


Figure.3 64-bit Hybrid CSLA architecture

As depicted in Figure above, the proposed adder is equipped with inputs Ai, Bi, and Cin, where A and B represent the two inputs to be added. The addition process is executed through a combination of the Kogge Stone Adder and Carry Lookahead Adder (CLA). At each stage, two possible carry values are generated, and the corresponding output is computed accordingly. The addition operation occurs concurrently in both stages. The resultant sum and carry are then directed to the selection unit, which determines the appropriate carry and produces the sum and carry-out (cout).

CLA is exclusively utilized at the initial positions of the adder. However, as the bit size increases, the complexity also escalates. Consequently, for larger bit sizes, the addition process is predominantly facilitated by the Kogge Stone Ad

SIMULATION RESULT



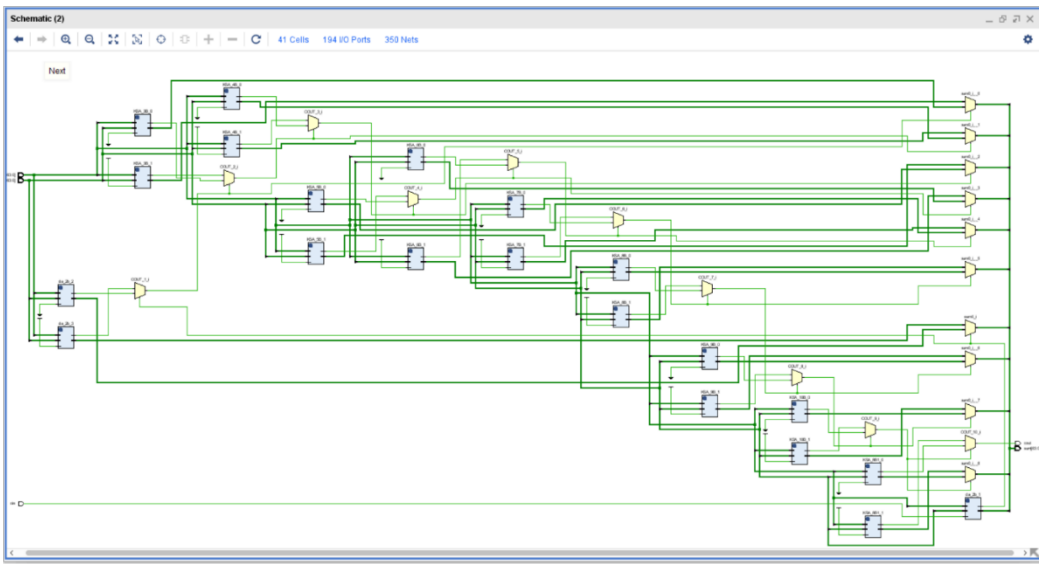


Figure.4 Schematic Hybrid CSLA 64Bit

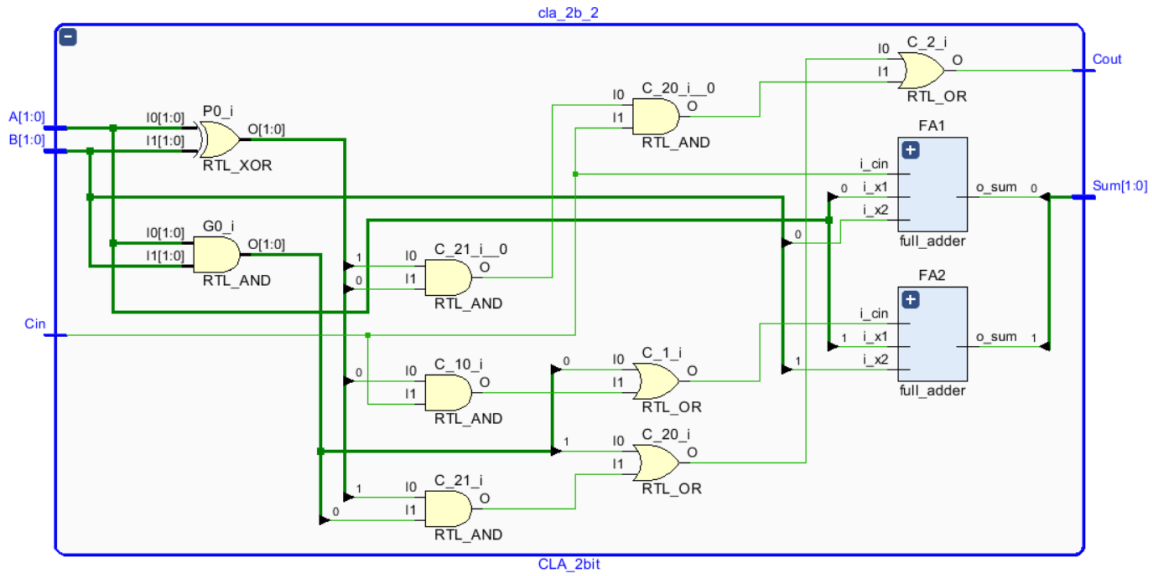


Figure.5 Schematic CLA 2Bit

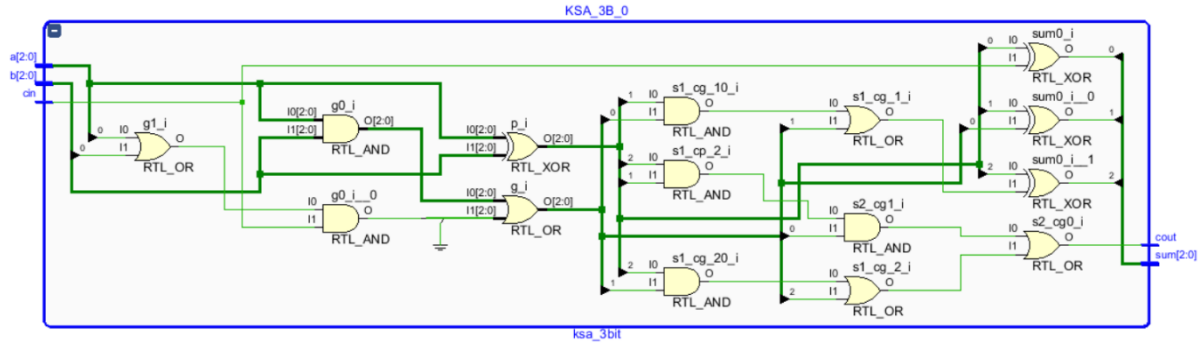


Figure.6 Schematic Kogge Stone Adder 3Bit

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	150	0	32600	0.46
LUT as Logic	150	0	32600	0.46
LUT as Memory	0	0	9600	0.00
Slice Registers	0	0	65200	0.00
Register as Flip Flop	0	0	65200	0.00
Register as Latch	0	0	65200	0.00
F7 Muxes	1	0	16300	<0.01
F8 Muxes	0	0	8150	0.00

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.033ns  (required time - arrival time)
  Source:                 b[6]
                            (input port clocked by virtual_clock  {rise@0.000ns fall@6.000ns period=12.000ns})
  Destination:            sum[46]
                            (output port clocked by virtual_clock  {rise@0.000ns fall@6.000ns period=12.000ns})
  Path Group:             virtual_clock
  Path Type:              Max at Slow Process Corner
  Requirement:            12.000ns  (virtual_clock rise@12.000ns - virtual_clock rise@0.000ns)
  Data Path Delay:        9.941ns  (logic 4.385ns (44.109%)  route 5.556ns (55.891%))
  Logic Levels:           8  (IBUF=1 LUT3=1 LUT5=4 LUT6=1 OBUF=1)
  Input Delay:            1.000ns
  Output Delay:           1.000ns
  Clock Uncertainty:      0.025ns
```

Figure.8 Timing report

Total On-Chip Power (W)	0.111
Design Power Budget (W)	Unspecified*
Power Budget Margin (W)	NA
Dynamic (W)	0.040
Device Static (W)	0.070
Effective TJA (C/W)	2.9
Max Ambient (C)	84.7
Junction Temperature (C)	25.3
Confidence Level	Low
Setting File	---
Simulation Activity File	---
Design Nets Matched	NA

Figure.9 Power Report

ADVANTAGES

Improved Speed: By integrating CLA and Kogge Stone Adder architectures, the proposed hybrid adder achieves enhanced speed performance compared to traditional adder designs. The CLA component minimizes carry propagation delays, while the Kogge Stone Adder exploits parallelism to further accelerate addition operations. This synergistic combination results in significantly faster arithmetic computations, making it suitable for high performance computing applications.

Scalability: The modular design of the hybrid adder facilitates scalability to accommodate varying bit-widths and operand sizes. This scalability is crucial for adapting the adder architecture to diverse application requirements, ranging from low-power embedded systems to high-performance computing platforms.

Versatility: The hybrid adder architecture exhibits versatility in handling different types of arithmetic operations, including addition, subtraction, and multiplication. This versatility stems from the inherent properties of CLA and Kogge Stone Adder, which are well-suited for a wide range of computational tasks

APPLICATIONS

Digital Signal Processing (DSP) Systems: In DSP applications such as audio and video processing, real-time computations are essential for tasks like filtering, convolution, and Fourier transforms. The high-speed hybrid adder can significantly accelerate these operations, enabling faster processing of multimedia data in real-time.

High-Frequency Trading (HFT) Systems: HFT systems require ultra-low latency in executing trades to capitalize on market fluctuations. The high-speed adder can enhance the performance of arithmetic units in these systems, facilitating quicker decision-making and execution of trading algorithms.

Communication Systems: Communication protocols like Ethernet, Wi-Fi, and LTE demand real-time processing of data packets for efficient transmission and reception. The high-speed adder can improve the performance of error detection/correction algorithms, modulation/demodulation schemes, and channel encoding/decoding processes in communication systems.

Robotics and Control Systems: Real-time control of robotic systems, autonomous vehicles, and industrial automation relies on fast arithmetic computations for trajectory planning, sensor fusion, and feedback control. The high-speed adder can enhance the responsiveness and accuracy of control algorithms, enabling safer and more efficient operation of robotic systems.

Image and Video Processing: Real-time image and video processing applications such as object detection, recognition, and tracking require rapid computation of pixel-level operations. The high-speed adder can accelerate tasks like image filtering, feature extraction, and motion estimation, enhancing the performance of vision-based systems.

Scientific Computing: Scientific simulations, computational fluid dynamics (CFD), and finite element analysis (FEA) demand high computational throughput for solving complex mathematical models. The high-speed adder can expedite arithmetic operations in these simulations, enabling faster analysis and decision-making in scientific research and engineering.

Medical Imaging and Diagnostics: Real-time processing of medical imaging data for applications like MRI, CT scans, and ultrasound imaging is crucial for timely diagnosis and treatment. The high-speed adder can improve the performance of image reconstruction algorithms, enabling faster generation of high-quality medical images.

CONCLUSION

In conclusion, our project on the "Design and Implementation of High-Speed Hybrid Carry Select Adder" introduces a novel approach to enhancing the characteristics of CSLA adders. By replacing the conventional RCA stages in a regular CSLA with a combination of fast adders such as Kogge Stone Adder and CLA, we have demonstrated significant improvements in speed and power consumption. Compared to other existing CSLA structures, our hybrid approach offers notable enhancements with only a slight increase in area overhead. This innovative hybrid structure underscores the potential for leveraging different adder architectures within a CSLA framework to achieve substantial performance improvements. Overall, our project contributes to the advancement of arithmetic circuit design by presenting a versatile and efficient solution for high speed arithmetic operations in digital systems.

FUTURE SCOPE

Optimization Techniques: Further research can focus on exploring advanced optimization techniques to enhance the performance and efficiency of the hybrid carry select adder (HCSA). This includes investigating novel algorithms for carry generation, optimizing the layout to minimize routing delays, and exploring alternative circuit topologies to reduce area overhead.

Low-Power Design: Given the increasing importance of low-power computing, future research can explore techniques to optimize the power consumption of the HCSA. This may involve investigating power gating strategies, operand isolation techniques, and voltage scaling methods to minimize power dissipation while preserving performance.

Exploration of Hybrid Architectures: Beyond the combination of carry-select and carry-lookahead adders and kogge stone adder, future work can explore the integration of other adder architectures into the HCSA framework. This includes incorporating elements from parallel-prefix adders, carry-skip adders, and other innovative designs to further enhance performance and efficiency.

Application-Specific Optimization: Future research can focus on tailoring the design of the HCSA to specific application domains. This involves identifying the unique requirements and constraints of different applications and optimizing the HCSA architecture accordingly to achieve the best performance for each application.

REFERENCES

[1] B. Moons and M. Verhelst, ‘‘DVAS: Dynamic voltage accuracy scaling for increased energy-efficiency in approximate computing,’’ in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2015, pp. 237–242.

[2] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, ‘‘Design ofvoltage-scalable meta-functions for approximate computing,’’ in *Proc. Design, Autom. Test Eur.*, Mar. 2011, pp. 1–6.

[3] K. Yin Kyaw, W. Ling Goh, and K. Seng Yeo, ‘‘Low-power high-speed multiplier for error-tolerant application,’’ in *Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC)*, Dec. 2010, pp. 1–4.

[4] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, and M. Pedram, ‘‘RoBa multiplier: A rounding-based approximate multiplier for highspeed yet energy-efficient digital signal processing,’’ *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 2, pp. 393–401, Feb. 2017.

[5] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, ‘‘Design and analysis of approximate compressors for multiplication,’’ *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.

[6] Z. Yang, J. Han, and F. Lombardi, ‘‘Approximate compressors for errorresilient multiplier design,’’ in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS)*, Oct. 2015, pp. 183–186.

[7] C.-H. Lin and I.-C. Lin, ‘‘High accuracy approximate multiplier with error correction,’’ in *Proc. IEEE 31st Int. Conf. Comput. Design (ICCD)*, Oct. 2013, pp. 33–38.

[8] P. J. Edavoor, S. Raveendran, and A. D. Rahulkar, ‘‘Approximate multiplier design using novel dual-stage 4:2 compressors,’’ *IEEE Access*, vol. 8, pp. 48337–48351, 2020.

[9] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, ‘‘A majority-based imprecise multiplier for ultra-efficient approximate image multiplication,’’ *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 11, pp. 4200–4208, Nov. 2019.

[10] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, ‘‘Comparison and extension of approximate 4–2 compressors for lowpower approximate multipliers,’’ *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 9, pp. 3021–3034, Sep. 2020.