Design and Implementation of High-Speed Modified Carry Select Adder Using Sklansky-Based Conditional Sum Adder

P. Deekshith[1], Rakesh Reddy G[2], T. Sivasankar[3], D. Aneesh[4], J. Adinarayana[5], M. Surekha[6]

[123456]*Department of Electronics and Communication Engineering,*
PBR Visvodaya Institute of Technology & Science, Kavali (Autonomous),
SPSR Nellore (Dt.), Andhra Pradesh – 524201, India

Department of Electronics and Communication Engineering,
PBR Visvodaya Institute of Technology & Science, Kavali (Autonomous),
SPSR Nellore (Dt.), Andhra Pradesh – 524201, India

Abstract - In this paper, a Sklansky-based Conditional Sum Adder (S-CSA) is introduced to enhance the performance of the Carry Select Adder (CSLA) architecture. The Sklansky adder utilizes a parallel prefix structure with minimal logic depth, enabling faster carry computation compared to conventional prefix adders such as Kogge-Stone and Brent-Kung. The conditional sum technique precomputes sum and carry values for possible input carry conditions, significantly reducing carry propagation delay during final selection. By combining the advantages of the Sklansky prefix network and the conditional sum approach, the proposed CSLA architecture minimizes redundant logic while maintaining high computational speed. The proposed S-CSA-based CSLA is designed and evaluated using standard VLSI design metrics including delay, power consumption, area, and power-delay product (PDP). Simulation results obtained using Xilinx Vivado on the Spartan-7 FPGA demonstrate that the proposed architecture achieves reduced area (127 LUTs vs. 225 LUTs), lower power (49.689 W vs. 51.065 W), and improved delay (9.835 ns vs. 12.229 ns) compared to the Brent-Kung based CSLA. The results confirm that the Sklansky-Conditional Sum Adder provides a superior trade-off among speed, area, and power, making it well-suited for high-performance and low-power VLSI applications.

Key Words: Carry Select Adder, Sklansky Adder, Conditional Sum Adder, Parallel Prefix Adder, VLSI Design, FPGA, Power-Delay Product.

---

1. INTRODUCTION

Binary adders form the critical arithmetic core of every processor and ASIC. In 64-bit architectures, Carry Propagation Delay (CPD) is the primary bottleneck. Ripple Carry Adders (RCA) suffer O(n) delay; Carry Select Adders (CSLA) reduce this but introduce hardware redundancy by dual-computing results for both carry-in conditions. Parallel prefix adders — Kogge-Stone and Brent-Kung — achieve O(log n) delay but trade off area, wiring complexity, and power.

This work proposes integrating a Sklansky-based Conditional Sum Adder (S-CSA) within the CSLA framework. The Sklansky prefix tree minimizes logic depth for fast carry generation; the conditional sum technique precomputes both sum results and selects the correct one via a multiplexer, converting carry propagation delay into the smaller multiplexer delay. The design is implemented on a Xilinx Spartan-7 FPGA using Verilog HDL and Vivado 2018, and benchmarked against a Brent-Kung CSLA baseline.

2. LITERATURE SURVEY

Soundarya and Rankuwa [1] showed that BEC-based modified CSLA achieves significant area savings over conventional designs while maintaining competitive speed across 8–64 bit configurations. Bidar Putt Raju [2] confirmed superior area-power efficiency of BEC-CSLA via FPGA synthesis. Annapurna Bai [3] demonstrated high-speed 128-bit Kogge-Stone adder operation, though with considerable wiring overhead. Simson and Deepak [4] combined CLA for lower bits and parallel prefix for upper bits in a hybrid CSLA, achieving speed improvements with moderate area cost. Gaur et al. [5] reduced 16-bit CSLA power using power gating and BEC logic. Obol Reddy [6] applied Common Boolean Logic (CBL) to eliminate redundant RCA blocks, achieving 25–35% area reduction. Kishore Kumar [7] introduced a reconfigurable CSLA with adaptive block sizing for runtime delay optimization. Chakali and Patnala [8] integrated Kogge-Stone within CSLA for minimal delay at the cost of dense interconnect. Kokilavani et al. [9] provided FPGA-specific optimization guidelines for hybrid CSLA. Saini et al. [10] and Koyada et al. [12]

confirmed through comprehensive comparisons that no single architecture is universally optimal — the selection must be application-driven based on speed, area, and power priorities.

3. EXISTING METHODOLOGY

The existing design employs a 64-bit Brent-Kung Adder (BKA) within a CSLA framework. The BKA organizes carry computation in a hierarchical tree with $2\log_2(N) - 1$ logic levels. The 64-bit operand is split into variable-sized blocks; each block computes partial sums and carry signals independently through the prefix network, and a multiplexer selects the correct output once the actual carry-in is resolved. This sparse tree structure reduces prefix node count and wiring compared to Kogge-Stone, lowering routing congestion and improving VLSI layout feasibility.

Limitations: redundant adder blocks increase circuit area; higher switching activity raises power; complex routing causes wiring congestion; additional logic levels increase delay; and scalability degrades at wider bit-widths.

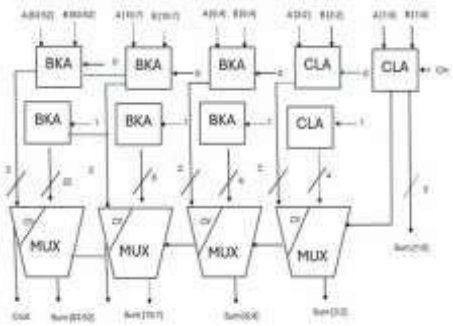

Fig. 1: 64-bit Brent-Kung Based CSLA

4. PROPOSED METHODOLOGY

The proposed S-CSA based CSLA replaces Brent-Kung prefix blocks with Sklansky adder units inside a conditional sum framework. Within each CSLA block, two Sklansky adders run in parallel — one assuming carry-in as logic '0' and the other assuming carry-in as logic '1' — precomputing both sum and carry outputs simultaneously. When the true carry-in from the preceding block arrives, a 2:1 multiplexer selects the correct result. This converts the slow inter-block carry chain into a single fast MUX selection, drastically reducing the critical path. Variable-sized CSLA blocks — smaller for low-order bits and larger for high-order bits — further balance delay and area across the full 64-bit width.

The Sklansky adder is a parallel prefix adder that organizes carry generate and propagate signals in a tree structure, allowing all carry bits to be resolved simultaneously rather than rippling sequentially. The prefix tree uses three types of internal cells: Black cells compute both the group generate and group propagate signals from two sub-groups; Gray cells compute only the group generate signal and are used at the final prefix stage where propagation is no longer needed; and Buffer cells pass signals unchanged to maintain fanout balance across the tree. The tree completes carry computation in logarithmic logic stages — fewer than Brent-Kung — with moderately higher fan-out at intermediate nodes. The final sum for each bit is obtained by combining each bit's propagate signal with the carry resolved from the prefix tree.

The design targets the Xilinx Spartan-7 FPGA (xc7s15ftgb196-1) on the PINE S7 development board, with a 7-segment display used for real-time output verification. The hardware description is written in Verilog HDL (IEEE 1364-2005) and the complete design flow — RTL entry, behavioral simulation, synthesis, place-and-route, timing and power analysis, and bitstream generation — is carried out using Xilinx Vivado 2018.x.

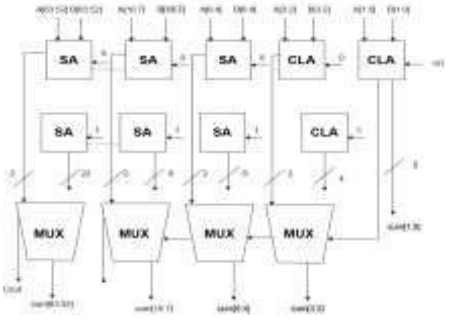

Fig. 2: 64-bit Sklansky-Based Conditional Sum Adder

5. RESULTS

Comparison between Existing and Proposed Adder

Table 1: Performance Comparison — Brent-Kung vs. Sklansky CSLA

Parameter	BKA Adder	Sklansky Adder
Area (Slice LUTs)	225	127
Critical Path Delay	12.229 ns	9.835 ns
Total Power	51.065 W	49.689 W
Power-Delay Product	624.3 W·ns	488.7 W·ns

For hardware verification, a 4-bit Sklansky Adder was deployed on the PINE S7 board. Input operands were applied via DIP switches and the computed sum was

displayed on the 7-segment display, confirming functionally correct operation across all tested input combinations.

6. DISCUSSION

The proposed S-CSA CSLA achieves simultaneous improvements across all three VLSI metrics — area, delay, and power — an outcome rarely seen in adder design, where optimizing one metric typically degrades another. The 43.6% area reduction from 225 to 127 LUTs is attributed to the Sklansky tree requiring fewer prefix cells than Brent-Kung at 64-bit width, combined with the conditional sum approach eliminating redundant inter-block carry ripple logic entirely. The 19.6% delay improvement from 12.229 ns to 9.835 ns results from reducing logic levels from 15 to just 8 — the Sklansky tree resolves all carries in logarithmic prefix stages and the MUX selection adds only two more levels. The 2.7% power reduction follows from fewer active nodes and shorter signal paths, which reduce gate switching activity per clock cycle; the static power of 0.171 W is identical in both designs as it is substrate-determined. The power-delay product improves by approximately 21.7%, from 624.3 W·ns to 488.7 W·ns, confirming that the proposed design delivers more computation per unit of energy.

7. CONCLUSION

A high-speed Modified Carry Select Adder using a Sklansky-based Conditional Sum Adder has been designed, simulated, and implemented on a Xilinx Spartan-7 FPGA. The Sklansky prefix network achieves $O(\log_2 n)$ carry computation depth; the conditional sum technique eliminates inter-block carry propagation by precomputing both outcomes and selecting via MUX. Together, these deliver a 43.6% area reduction, 19.6% delay improvement, 2.7% power reduction, and 21.7% PDP improvement over the Brent-Kung CSLA baseline — with hardware correctness confirmed on-board.

ACKNOWLEDGEMENT

The authors sincerely thank Mrs. M. Surekha (Associate Professor, ECE, PBR VITS Kavali) for her guidance, Dr. R. Sravanthi (Professor & HoD, ECE) for providing facilities, and Dr. V. Anil Kumar (Principal, PBR VITS Kavali) for the academic environment that enabled this work.

REFERENCES

[1] J. Soundarya and Rankuwa, "Comparative analysis for different models of carry select adder," International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), vol. 3, Jan. 2016.

[2] N. Bidar Putt Raju and Adithya, "Performance Comparison of Carry Select Adders," International Research Journal of Advanced Engineering and Science, vol. 1, no. 2, pp. 73–75, 2016.

[3] Annapurna Bai and Vijaya Laxmi, "Design of 128-bit Kogge-Stone Low Power Parallel Prefix VLSI Adder for High-Speed Arithmetic Circuits," International Journal of Engineering and Advanced Technology (IJEAT), vol. 2, no. 6, Aug. 2013.

[4] A. Simson and D. S., "Design and Implementation of High-Speed Hybrid Carry Select Adder," 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 2021.

[5] Nidhi Gaur, Anu Mehra, and Pradeep Kumar, "16 Bit Power Efficient Carry Select Adder," International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, 2016.

[6] D. Obol Reddy and P. Yadav, "Carry Select Adder with Low Power and Area Efficiency," International Journal of Engineering Research and Development, vol. 3, no. 3, pp. 29–35, Aug. 2012.

[7] G. Kishore Kumar and N. Balaji, "Reconfigurable Delay Optimized Carry Select Adder," Proceedings of IEEE International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICIEEIMT), 2017.

[8] P. Chakali and M. K. Patnala, "Design of High-Speed Kogge-Stone Based Carry Select Adder," International Journal of Emerging Science and Engineering (IJESE), vol. 1, no. 4, Feb. 2013.

[9] V. Kokilavani, K. Preethi, and Balasubramanian, "FPGA Based Synthesis of High-Speed Hybrid Carry Select Adders," Hindawi Publishing Corporation Advances in Electronics, 2015.

[10] J. Saini, S. Agarwal, and A. Kansal, "Performance Analysis and Comparison of Digital Adders," International Conference on Advances in Computer Engineering and Applications (ICACEA), IEEE, 2015.

[11] S. Manjui and V. Sornagopal, "An Efficient SQRT Architecture of Carry Select Adder Design by Common Boolean Logic," IEEE, 2013.

[12] B. Koyada, N. Meghana, Md. Omair Jaleel, and P. R. Jeripotula, "A Comparative Study on Adders," IEEE WiSPNET Conference, 2017002E