

DESIGN AND ANALYSIS OF NOVEL PARALLEL PREFIX ADDERS FOR VLSI CIRCUITS

Prabakaran Govindaraj*, Shanmugasundaram Nallasamy, Mohankumar Mysamy and Sathiyapriya Krishnamoorthy

Received: February 14, 2023; Revised: November 24, 2023; Accepted: January 19, 2024

Abstract

Arithmetic Logic Unit is the brain of all processors, is composed of an Adder circuit. The main component of multiplier circuits is the adder, which performs subtraction (by 2's complement arithmetic). Since the most fundamental operation in mathematics is addition, and the adder is the most essential part of the processor, the study of VLSI arithmetic has required many digital VLSI researchers. When designing digital systems employing the VLSI approach, the digital adder plays a major role. Low power VLSI based adder designs perform poorly because of the propagation delay issue. With the aid of the PPA design, an assessment of adders' availability for low power VLSI designs with minimal propagation delay is conducted. Using these performance measures, this study compares and evaluates the performance of several parallel prefix adders, allowing readers to select the optimum adder architecture for their application Specific Integrated Circuits implementations. The three adder topologies taken into considered in this Article include the Han carlson adder, Brent kung, Kogge stone adders. Xilinx ZYNQ XC7Z020 (7000 series) SoC is used to implement the adders using the Vivado Design Suite 2014 and Verilog.

Keywords: ALU, Han Carlson Adder, Brent Kung, Kogge Stone Adders, PPA

Introduction

In majority of digital circuits, the Binary adder serves as the critical path component. As a result, the speed of the adders has a significant impact on how well the overall system works. The design of digital circuits nowadays must consider for smaller size, lower power consumption, and quick speed in the age of portable electronics. For calculations, the VLSI chips heavily rely on high-performance, low-power adder circuits. Up to a certain length of input bits,

conventional adders are successful in meeting the criteria. Some of these adders slowdown in computations as the input bit width grows while using less power and taking up less space, whereas other variations are quick despite using substantially more power. Parallel prefix adders are provided a fair trade-off between consumed power and also speed. This article introduces the connection between graph energy and electrical energy or power

Vivekanandha College of Technology For women, Namakkal, Tamilnadu, India. E-mail: prabaece345@gmail.com; hodece@sece.ac.in; mohankumar.m@sece.ac.in; ksathiyapriya1999@gmail.com

* Corresponding author

DOI: <https://doi.org/10.55766/sujst-2024-01-e0379>

Suranaree J. Sci. Technol. 31(1):010276(1-8)

consumption for PPA designs. The number of recursive steps is limited, and it mainly focuses on the Sklansky Parallel Prefix adder.

As a result, Parallel Prefix internal nodes are reduced. The suggest adder was created as a result of the suggested adder I's employment of certain graph nodes of dot and semi-dot operators. (Moghaddam and Ghaznavi, 2011) proposed adder they trimmed the nodes and edges and set a maximum of 8 steps for the recursive stages. Used transistors count, overall consumed power, and major time delay route in the sklansky adder are all reduced by about 31%, 29%, and 7%, respectively, when compared to our proposed adder. (Daphni and Grace, 2017) explained how different Parallel prefix adders (PPA) were designed and analyzed, as well as how well they performed in terms of latency, area, and power. (Thakur *et al.*, 2020) The KSA is preferable for the delay process according to the investigation's findings, automatically increases the addition's pace. However, it requires more space and electricity; Parallel prefix adder is offered as a non-speculative and speculative parallel prefix adder, which increases its dependability for usage in situations where high-speed circuits are needed.

(Yezerla, 2014) article shows the area and power, concluded that the efficiency is improved by 5.77 % in ours delay for RCA. (Kaneko, 2019) The framework provide us a novel, compact and expandable representation of individual parallel prefix structures, which might be used in search-based design optimizations. For FPGA implementation, (Rahila and Kumar, 2019) compared parallel prefix adders. They compared several adders under various software tool optimization circumstances and demonstrated that, in some cases, the algorithmic advantage of particular adders is not maintained. Additionally, they demonstrated that for FPGA implementations, the integrated FPGA adder outperforms parallel adders in terms of delay performance. (Zamhari *et al.*, 2012). Maimun Huja Husin produced the vector waveform which then shows the computational delay for the adders. Hence, this project is significant in showing which of the two adders being tested perform better in terms of computational delay and design area based on different sizes of bits. (Sudhakar *et al.*, 2012) explores a variation of the Han-Carlson adder for large word sizes and compares the performance

of the new design with the traditional design. This work introduces a second type of design with two Brent-Kung stages each at the beginning and at the end and with Kogge-Stone stages in the middle, henceforth referred to as the "Hybrid Han-Carlson design." With the new design, the Hybrid Han-Carlson adder, the delay increases slightly, but the complexity, silicon area and power are reduced significantly. (Vitoroulis and Al-Khalili, 2007) look at the effectiveness of parallel prefix adders that use FPGA technology. They discuss the essential route latency and space requirements for a number of conventional parallel prefix adder topologies. (Bais and Ali, 2016) uses a comparison of three parallel adder designs created with the Xilinx ISE14.4 Design Suite System and Spartan 3 FPGA is used to synthesis. The CLA, RCA, and KSA After analyzing the impact of parallelism on the speed and area of adder designs, it was found that the two parameters could not be maximized simultaneously. (Hoe, 2011) The Kogge-stone (KSA), Sparse Kogge-stone (SKS), and Spanning tree adders are examined in this research along with current adder designs. Their performances are compared in light of various design constraints to discover which adder performs the best among them. (Xiang, 2017) If parallelism is enhanced to speed up operation, it will lead to more area occupancy in Overall area of a chip; if area is to be optimized, we must make accommodations for the system's poor speed. (Cho *et al.*, 2021) Su Min Cho, Pramod Kumar Meher proposed FIR filters are synthesized to measure the maximum throughput and the balance between complexity and speed. (Macedo *et al.*, 2017) evaluates the use of Brent-Kung, Kogge-Stone, Han Carlson, Ladner-Fisher, and Sklansky PPAs in the precise part of two well-known approximate adders. (Venkata, 2019) The Different 4-bit, 8-bit, and 16-bit operands have been used to design the each adders, and a comparison of delay performance and according to the information gleaned from the synthesis findings, space has been used. In this paper explains the characteristics of a few of the several standard parallel prefix adder techniques in section 2. The adder configures rations utilized for simulation and comparative analysis are covered in section 3. The results of simulations utilizing the Xilinx vivado ISE design suite 2014 design environment are analyzed and compared in Section 4, which is concluded by Section 5.

Parellal prefix adders

A PPA performs the process in parallel manner, which is accomplished by segmenting the operation into smaller units. In VLSI designs, parallel-prefix adders perform the best. These adders employ the idea of spreading and creating carries. Pre-computed propagate and produce signals are joined in a variety of ways utilizing the basic carry operator. The objective of adding two numbers is to get the sum (S) of operands A and B, both of which are binary words with length n. Each bit of a binary word is represented by a tiny subscript letter, and binary words are written in capital letters. Beginning with 0 for the LSB, the subscript shows where in the word the bit is located. For instance, c_i stands for the carry in first bit, whereas a (7:0) denotes the least significant 8 bits in the word A. The input carry is C_0 . Sum S can only be as long as $n+1$ because the length of the phrases and B combined equals n.

The generated output (G) and propagated output (P) terms for all bit of the operands that is computed at the first stage of the adder for n-bit addition using the following Equations:

$$P(r:t) = P(r:s) \cdot P(s-1:t) \quad (1)$$

$$G(r:t) = G(r:s) + (G(s-1:t) \cdot P(r:s)) \quad (2)$$

The last stage, known as post processing, is used to add up all the input bits and output carries, which are standard across all the adders. The various types of PPA included by the next section.

KSA (Kogge stone adder)

KSA is one of the most popular parallel prefix adders that used in circuits. Its computing speed is high due to a quicker design, but the circuit area is

larger. Preprocessing, Carry graph, and Post-processing are three steps that make up any parallel prefix adder's addition generation process (sum block). See the carry graph network graphic in Figure 1.

At stage 0, generate (G_i) and propagate (P_i) are created, respectively, for each bit from inputs A and B. With the aid of group propagate and generate, carry is derived in the subsequent step, known as carry generate. Here, the number of steps in this stage changes depending on the quantity of input bits. The formation of the sum occurs in the last stage, when the propagate (P_i) and carry, which are produced by the carry network, are XORed. The adder has the advantages of having a low fan-out, quick addition, and a minimal circuit delay, which is determined by $\log_2(n)$, here n, is the total no of input bits. Enormous area that the adder occupies is a drawback. Using Vivado Design Suite, Figure 2 displays the schematic diagram of a designed KSA. (Explained in 3) and also Figure 3. Shows the test bench waveform of KSA.

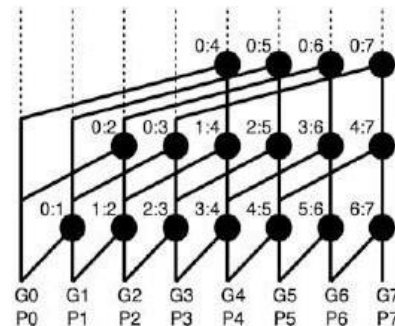


Figure 1. Prefix tree for 8-bit KSA

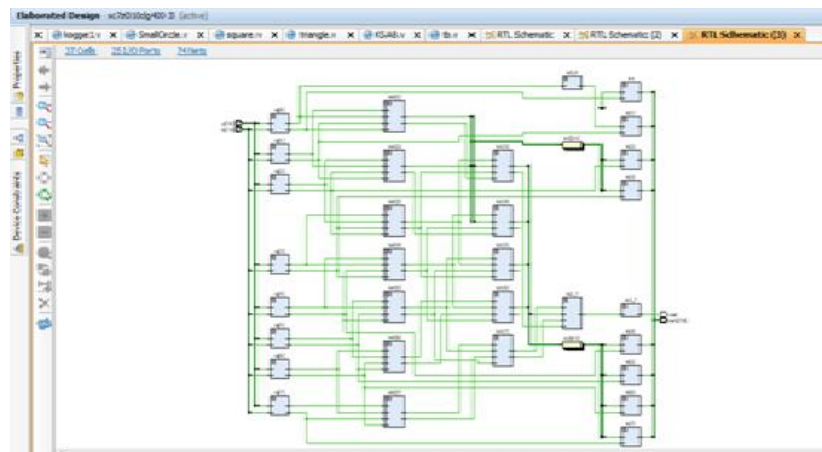


Figure 2. Shows RTL Schematic Diagram of Proposed KSA

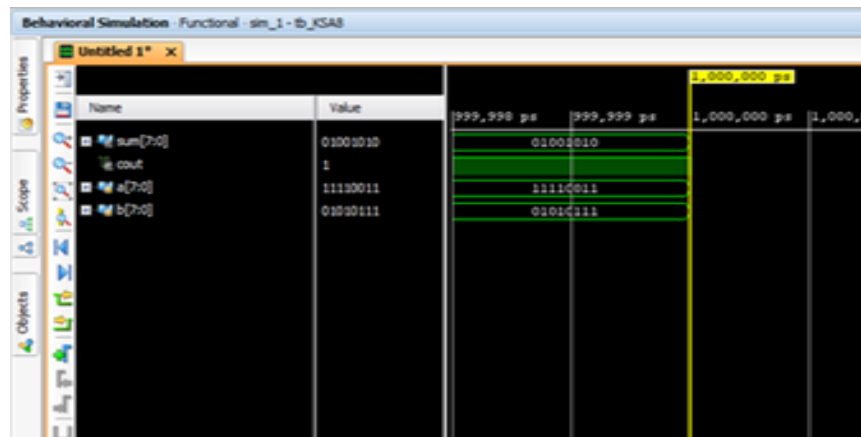


Figure 3. Shows Test bench waveform of KSA

Brentkung adder (BKA)

Carry look ahead adder (CLA) has a PPA variant called the Brent-Kung adder. When BKA is compared to ripple-carry adder substantially faster than RCA. The bit length is being added closely correlated with the time required to adding the bits Figure 4.

The carry is calculated concurrently in BKA, greatly reducing the adding time. The BKA is significantly easier to construct than the KSA since it requires some modules to implement. In addition of BKA has a lot less links to other modules, which furthers its simplicity. The carry graph network has shown the amount of wires to be one of the KS adder's drawbacks. This amount is high and increases quickly as the size of the inputs increases. The wires must also cross numerous other lines, which makes routing more difficult. Let's look at the 8-bit KS adder under the premise that only the last carry matters. The majority of the dot operations may be eliminated in this situation, leaving the tree diagram is shown in below:

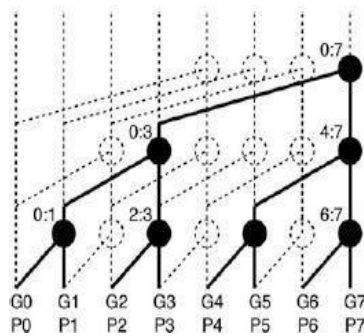


Figure 4. Shows Pruned tree diagram for 8-bit KSA

Although the routing congestion is lessened in this fashion, the internal carries are also required to compute S0 through S6, therefore this is not a true 8-bit adder. Therefore, we modify this trimmed tree to create a new adder known as the BKA. At the top of the last-bit carry, An BKA (8bit) requires four more dot operators. That is employed to determine the not presented carries with least amount of fresh components in hardware. An 8-bit BKA prefix tree diagram is showed in Figure 5.

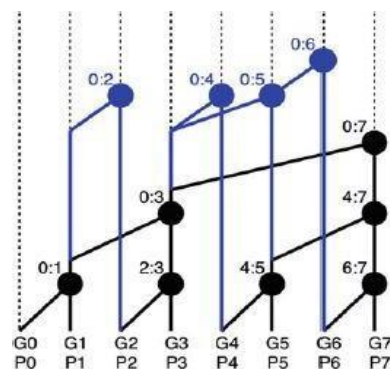


Figure 5. Shows Prefix tree for 8-bit BKA

As you can see, BKA are considerably smaller in size than KSA. The 8-bit BKA uses 11 dot operations as opposed to 17 in KSA. 16 Bit KSA has the 49 dot operators while the 16-bit BKA has only 26 operators. As a result, the area is the BK adder's advantage, which is accentuated more and more as input size increases. In addition, the BK adder's wiring is

more routable than the KS adder's. The BK adder, however, has two significant drawbacks. First, there are now more phases required to compute all internal bit carries. Second, a lot of dot operations fan out into more than two units. Figure 6 displays the Vivado generated design of an 8-bit BKA and Figure 7. Shows Test bench waveform of 8 bit BKA.

HCA (Han-Carlson adder or Hybrid Prefix adder):

A hybrid strategy called the HCA makes use of the KSA (hence hybrid) and BKA. In the initial and end phases of the carry generation, it

employs the BKA technique (slow but little), while in the middle stages, it employs the KSA approach (rapid but huge). The Kogge-prefix Stone's trees and the Han-Carlson prefix tree concept are both prefix trees with a maximum fan-out of two. The distinction is that Kogge stone adder employs substantially more hardware components and wiring than Han carlson prefix tree. One more logic level is the price. This adder therefore provides a suitable trade-off between the quantity of logic units and also fan-out. As a result, it uses less electricity. The following Figure 8 shows the prefix tree for han carlson adder (16 bit).

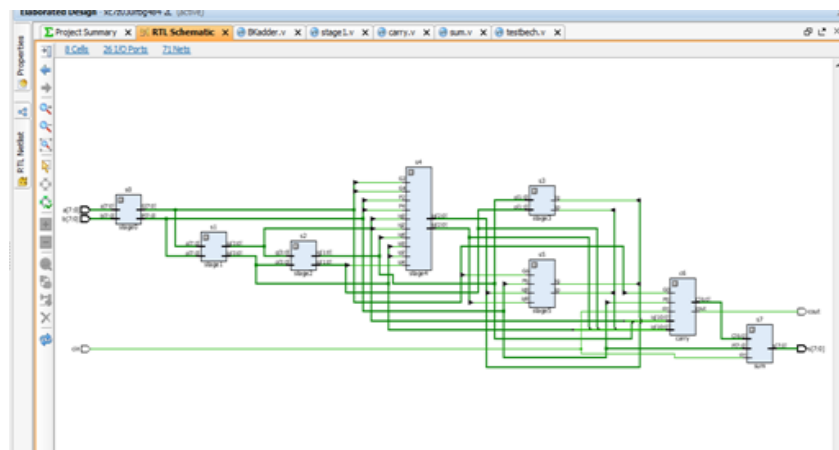


Figure 6. Shows RTL Schematic diagram of 8 Bit BKA

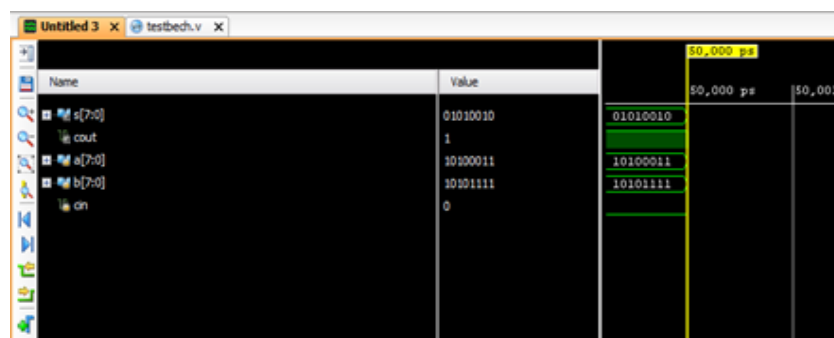


Figure 7. Shows Test bench waveform of 8 bit BKA

In above mentioned diagram, the Han-Carlson adder uses the Kogge Stone approach in the middle and a single Brent Kung level at the beginning and end of the graph. Simply explained, the Han-Carlson adder operates at

greater speeds while using less space and electricity. The 8-bit Han-Carlson adder's synthetic schematic was created with Vivado design suite is described in Figure 9 (Table 1 and Table 2).

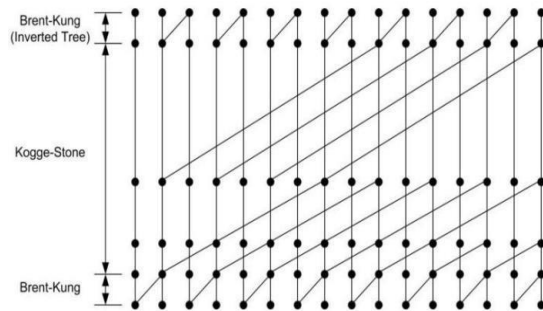


Figure 8. Shows Prefix tree for 16-bit HCA

Table 1. Shows Analysis of the PPAs described in this study

Prefix structure	Logic levels	Fan out	Prefix count
Kogge stone adder	$\log n$	2	$n \log n - 1n + 1$
Brent kung adder	$2 \log n - 1$	2	$2(n-1) - \log n$
Han carlson adder	$\log n + 1$	2	$(n/2) \log n$

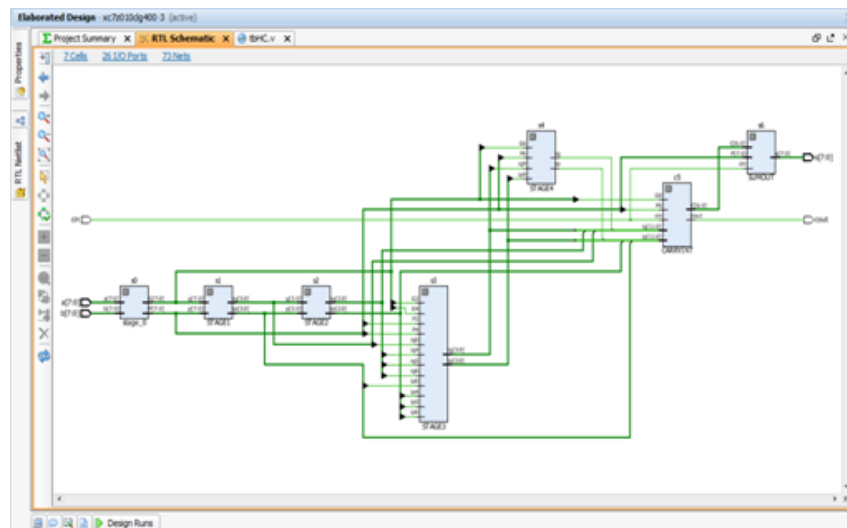


Figure 9. Shows RTL Schematic Diagram of 8 bit HCA

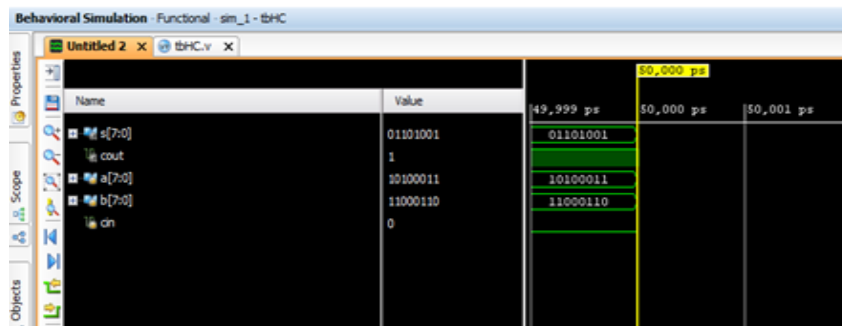
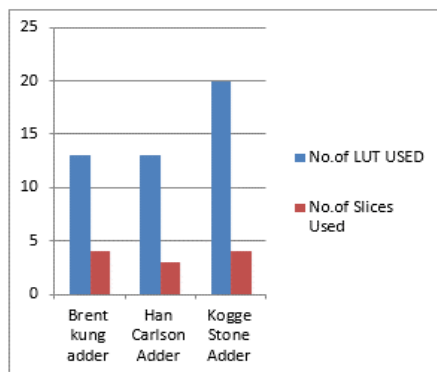
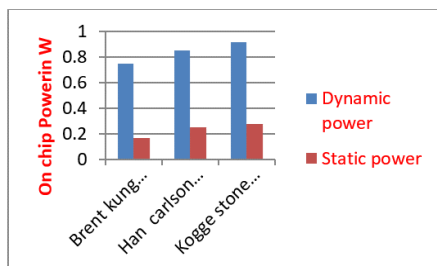


Figure10. Shows Test bench waveform of 8 bit HCA

Table 2. Shows Comparing the performance of the prefix adders presented in the article

	PPA	AREA		On chip power in W	
		LUT	Slices	Dynamic power	Static power
1	Brent kung adder	13	4	0.747	0.167
2	Han carlson adder	13	3	0.848	0.251
3	Kogge stone adder	20	4	0.911	0.276

**Figure 11. Shows Comparison of Area In terms of LUT and Slices****Figure 12. Shows Comparison of Static and Dynamic Power****Area and Power Analysis:**

It's time to evaluate several parallel prefix adders to determine which one is the best for a given application now that we have a better understanding of them and the related prefix tree diagrams. The performance characteristics of all the PPAs covered in this study are briefly summarized in the table below as a function of input bit width.

Performance Evaluation Using Xilinx's Vivado Design Suite:

The hardware development tools like the Vivado Design Suite are used in conjunction

with Xilinx's FPGAs and development boards. The designs are implemented in simulation use the hardware specification feature on the APSoC known as the XC-7Z020 CLG484 in the Zynq Evaluation and Development Kit (Zedboard). (Zynq-7000 series). Eighty-five thousand cells of programmable logic are combined with a dual Cortex A9 Processing System (PS) (PL or FPGA). Figure 11 and Figure 12 shows the Graphical representation of Comparison mentioned adders including parameters like area and power.

Conclusions and Future Work

"Novel design of parallel prefix adders for VLSI (Very Large Scale Integration) circuits" refers to a creative and unique way of creating adder circuits that employ parallel prefix computation techniques. Because they add multiple bits at once, parallel prefix adders are crucial components of digital circuits because they boost efficiency and speed. When comparing a novel design to other developed parallel prefix adders, several considerations are made: Performance Metrics, area efficiency, dynamic power.

Comparing three different adders using the metrics area and power, the Brent-kung adder is the best adder out of the ones chosen, yields the strongest results. Our investigation shows that, among PPA, the BKA consumes the least electricity. The area of the Brent-kung adder is significantly reduced when the number of slices used is to take into the account. When comparing the three PPAs in terms of area, cost, and electricity, the BKA emerges as the superior choice. The BKA's area does expand as the bit size increased, but not as dramatically as KSA. The adder is larger in terms of area the more bits that the PPAs can accommodate. In our further investigations, we want to contrast adder circuit fan-outs, which are not included in this paper. The utilization of reconFigureurable FPGAs in conjunction with a processing system (like the Zynq-7000 SoC's) that can be altered in the field with the proper adder as needed by the application is the most exciting aspect.

List Of Abbreviation

VLSI: Very Large Scale Integration
 ALU: Arithmetic and logic unit
 PPA: Parallel prefix adder
 FPGA: Field programmable gate array
 KSA: Kogge stone adder

BKA: Brent kung adder
HCA: Han carlson adder
LUT: Look up table
CLA: Carry look ahead adder

References

- Bais, K. and Ali, Z. (2016). Comparison of Various Adder Designs in terms of Delay and Area. *International Journal of Science and Research*, 2013:1,292-1,296.
- Cho S.M., Meher, P.K., Trung, L.T.N., Cho, H.J., and Park, S.Y. (2021), Design of Very High-Speed Pipeline FIR Filter Through Precise Critical Path Analysis. *IEEE Access*, 9: 34,722-34,735. <https://doi.org/10.1109/ACCESS.2021.3061759>
- Daphni, S. and Grace, K.S.V. (2017). A review analysis of parallel prefix adders for better performance in VLSI applications *IEEE International Conference on Circuits and Systems (ICCS)*, 103-106.
- Hoe, D.H.K. Martinez, C. and Vundavalli, S.J. (2011). Design and Characterization of Efficient Parallel Prefix Adders using FPGAs. 2011 *IEEE 43rd Southeastern Symposium on System Theory*. <https://doi.org/10.1109/SSST.2011.5753800>
- Kaneko M. (2019). A Novel Framework for Procedural Construction of Parallel Prefix Adders. *IEEE International Symposium on Circuits and Systems (ISCAS)*. <https://doi.org/10.1109/ISCAS.2019.8702117>
- Macedo, M., Soares, L., Silveira, B., Diniz, C.M., Costa E.A.C. (2017). Exploring the use of parallel prefix adder topologies into approximate adder circuits. *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 298-301. <https://doi.org/10.1109/ICECS.2017.8292078>
- Moghaddam, M. and Ghaznavi-Ghouschi M.B. (2011). A New Low-Power, Low-area, Parallel Prefix Sklansky Adder with Reduced Inter-Stage Connections Complexity. 2011 *IEEE EUROCON - International Conference on Computer as a Tool*. <https://doi.org/10.1109/EUROCON.2011.5929280>
- Rahila K.C, Kumar U.S. (2019). A Comprehensive Comparative Analysis of Parallel Prefix Adders for ASIC Implementation. *International Conference on System Energy And Environment*, 1-5.
- Sudhakar, S.M., Chidambaram K.P., and Swartzlander E.E. (2012). Hybrid Han-Carlson Adder. *Midwest Symposium on Circuits and Systems (MWSCAS)*, 818-821. <https://ieeexplore.ieee.org/document/6292146>
- Thakur, G., Sohal, H., and Jain, S., (2020). FPGA-Based Parallel Prefix Speculative Adder for Fast Computation Application. 2020 *Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 206-210. <https://doi.org/10.1109/PDGC50313.2020.9315783>
- Venkata, V., Visweswara, S., Ramachandram, S., and Amulya, P. (2019). Performance Comparison Among Various Parallel Prefix Adder Topologies, *Journal of Engineering Sciences*, 10(1):342-346.
- Vitoroulis, K., and Al-Khalili, A.J. (2007). Performance of Parallel Prefix Adders implemented with FPGA technology. *Annual IEEE Northeast Workshop on Circuits and Systems (NEWCAS)*, 498-501. <https://doi.org/10.1109/NEWCAS.2007.4487969>
- Xiang, L.M., Zabidi, M.M.A., Awab, A.H., Rahman, A.A.H.A. (2018). VLSI Implmentation of a Fast Kogge-Stone Parallel-Prefix Adder. *Journal of Physics: Conference Series*, 1049 (2018):012077. <https://doi.org/10.1088/1742-6596/1049/1/012077>
- Yezerla, S.K. and Naik, B.R. (2014). Design and estimation of delay, power and area for Parallel prefix adders. *Recent Advances in Engineering and Computational Sciences (RAECS)*. <https://doi.org/10.1109/RAECS.2014.6799654>
- Zamhari, N., Voon, P., Kipli, K., Chin, K.L. and Husin M.H. (2012). Comparison of Parallel Prefix Adder (PPA). *Proceedings of the World Congress on Engineering 2012 Vol II*.