VLSI IMPLEMENTATION OF KOGGE-STONE ADDER FOR LOW-POWERAPPLICATIONS

By

EPPILI JAYA* **B. SAI SRI**** **P. AKSHAY KUMAR**** **O.HEM KUMAR****

jaya.baratam@gmail.com saisriboina@gmail.com potnuruakshay@gmail.com hemkumarodisi@gmail.com

D.SUNIL** **R. RAJESH****

dasarisunil9359@gmail.com rajeshrokkam567@gmail.com

*ASSISTANT PROFESSOR, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, ADITYAINSTITUTE OF TECHNOLOGY AND MANAGEMENT, TEKKALI.

**STUDENT, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, TEKKALI.

ABSTRACT

The adder is a vital part of the central processing unit (CPU), the main processing unit of any device that can perform computational operations. These are used in the digital components that are mostly used in the design of integrated circuits. Recent decades have seen a sharp rise in demand for mobile electronics, which has increased the need for highly efficient VLSI structures. All operations must be computed using low-power, space-efficient designs that run faster. The Kogge-Stone adder (KSA) is an extension of the carry look-ahead adder used for performing fast addition in high-performance computing systems. The latency, space, and energy used by the Kogge-stone adder after development and implementation in Xilinx Vivado using Verilog are compared in this study to those of the RCA and CLA. The Kogge Stone adders (KSA) results show a decrease in power consumption as well as improvements in high speed and area compaction when compared to the RCA and CLA.

Keywords:*Adder, Carry look-ahead adder (CLA), Kogge-Stone Adder (KSA), Ripple Carry Adder (RCA).*

INTRODUCTION

The ALU serves as the main building block for digital processors (DSP), microprocessors, microcontrollers, and other data-processing devices. Adder is an essential hardware unit for the following application in many arithmetic operations. The optimization of speed and reduction of power consumption has a significant impact on the latency andoverall energy used by the microprocessors because adders are the most crucial component in ALU. The Kogge-Stone adder uses a tree-like structure that extends the carry-lookahead adder by parallelizing the computation of the carry signals for several bit positions. The carry signals for adjacent bit positions are computed for each stage of the tree using the carry signals from the previous stage. The final sum and carry signals are calculated using the output carry signals from the tree's final stage. The Kogge-Stone adder's main benefit is its capacity for quick addition with a short critical path delay. The adder can compute the carry signals for several bit positions at once thanks to the parallel processing employed in the tree structure, which shortens the overall processing time. Because of this, high-performance computing systems that need quick arithmetic operations frequently use the KSA. The primary goal of this paper is to implement the KSA and compare it with other adders like RCA and CLA. The above adders have been simulated and synthesized on the Xilinx Vivado platform and their parameters are captured. The specifications of the Xilinx Vivado software are Artix–7 families, csg324 Package with Speed Grade of –1. Finally captured parameters like latency, area, and energy used by the above-mentioned adders are compared.

1. Literature survey:

Penchalaiah and Kurnar investigated the Kogge-stone adder, a new PPA architecture. The results of the application of the proposed method are verified by comparing the Kogge-stone adder with Carry skip adder in terms of size, latency, and speed [1]. A general procedure used in digital circuits to simplify the circuit and its operation is the addition of a certain number of bits. Selecting an adder with the appropriate characteristics is even more important for the circuit to function properly [2]. Daphni and Vijula Grace explained the design and analysis of many parallel prefix adders and compared their performance in terms of area, latency, and power usage. [3]. High-speed designs frequently employ the carry-lookahead adder (CLA) and its variants, such as the parallel prefix (PPF) adder [4]. Although adders are necessary, the kind used depends on the program in terms of speed, power consumption, and area usage[5]. Circuit designers benefit greatly from the ability to reach faster rates with less power dissipation. Minimizing the supply voltage is a simple method to reduce the energy consumption of the circuits because there is a quadratic relationship between the switching energy and the voltage [7]. To increase energy and speed, the designer can use several adder structural modifications. There are many

adder families, and they all have various delays, energy needs, and spatial requirements. There are several different types of adders, including parallel prefix adders, ripple carry adders, carry increment adders, carry skip adders, carry select adders, and carry look-ahead adders (PPA) [8].

2. Background information:

There have been recent events that have a considerable increase in the need for high-performance computing, which has prompted the creation of cutting-edge processors and memory architectures. The portability and battery life of mobile devices is, however, frequently constrained by the higher power consumption that results from this improved performance. The creation of low-power computing systems that can deliver excellent performance while using the least amount of power is becoming more and more popular as a solution to this problem. The performance and power consumption of adders, which are essential components of digital circuits, significantly affects those of the entire system. In this research, we want to construct KSA for low-power applications using very large-scale integration (VLSI) approaches. To reduce power consumption while retaining excellent performance, we will optimize the KSA design. To show the benefits of KSA in terms of speed and power consumption, we will also compare the performance and power consumption of KSA with that of other frequently used adders, such as the carry-lookahead adder (CLA) and ripple carry adder (RCA). The overall goal of this project is to show how well KSA performs high-speed arithmetic operations, which will help in the development of low-power computing systems. Mobile devices, embedded systems, and high-performance computer systems can all benefit from the project's findings in terms of increased performance and power economy.

METHODOLOGY

Adders:

(I) Ripple carry adder (RCA):

A ripple carry adder is a simple digital circuit that adds two binary values. The carry signal ripples over each stage of the circuit while the addition is done, giving the circuit its name [6]. Each full adder in the circuit inputs two bits from the input numbers and the carry signal from the preceding step, and the outputs are a sum and a carry. The sum bit is output as a component of the final sum, while the carry bit from each full adder is sent on as the carry signal to the following stage.

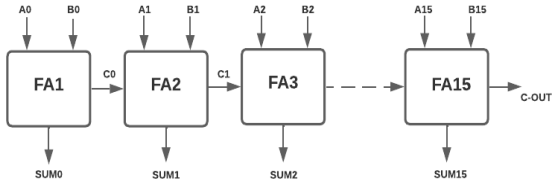


Fig-2.1.1: 16-Bit Ripple carry adder

In the above diagram, A0 to A15 and B0 to B15 represent the 16-Bit binary numbers, while the output sum is denoted by SUM. The circuit uses sixteen full adders (FA) to compute the sum and carry bits for each bit position. The carry bits are propagated from each stage to the next, resulting in a ripple effect as the addition is performed.

(II) Carry look-ahead adder (CLA):

A carry-lookahead adder (CLA) is a parallel adder circuit adder that can reduce the propagation delay of carry signals. Instead of waiting for the carry to propagate through the entire adder circuit, it uses a lookahead carry generator to generate the carry signals for each bit in parallel. The CLA divides the adder into bit groups, with each group having its lookahead carry generator. Each group's carry generator takes the input bits and generates the carry signals for that group. The carry generator determines the carry signals using a set of Boolean functions based on the input bits. Each carry generator produces a set of carry signals for that group, which are then combined with the previous group's carry signals to produce the carry signals for the next group. This process is repeated until the final carry signal is produced.

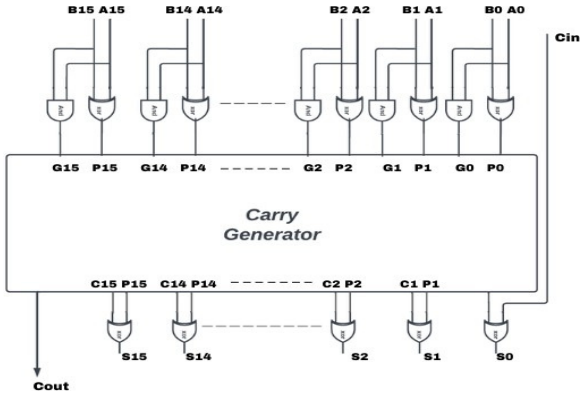


Fig-2.2.1: 16-Bit Carry look-ahead adder

In the above diagram, A (A0 to A15) and B (B0 to B15) represent the input numbers, while the output sum is denoted by S (S0 to S15. The adder unit then takes in the input numbers and the carry signals and performs the addition to generate the final sum.

The carry generator unit consists of a series of carry lookahead logic gates, which generate the carry signals for each stage of the adder in parallel. By calculating the carry signals in parallel, the CLA can significantly reduce the propagation delay compared to a ripple carry adder, resulting in faster operation.

(III) Kogge-stone adder (KSA):

The Kogge-Stone adder is a parallel adder that computes the sum of two binary numbers at high speed. It is similar to the carry-lookahead adder, but it generates the carry signals in parallel using a tree-based structure, resulting in faster operation. The Kogge-Stone adder is made up of a series of full adders that are arranged in a tree-like structure. Each full adder accepts two input bits and a carry bit and outputs a sum bit and a carry bit. Each full adder's carry bit output is then propagated up the tree to the next level, where it is combined with the carry bits from the other full adders in that

level to produce the next set of carry bits. The complete functioning of KSA can be easily comprehended by analyzing it in terms of the following three distinct parts:

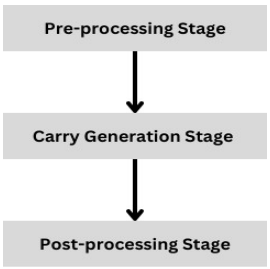


Fig-2.3.1: Stages of Kogge-stone adder

(i)Pre-processing Stage:
This process includes calculating the propagate and generate carry that corresponds to each pair of bits in A and B. The logic equations below provide these signals:
Propagate carry (Pi) = Ai XOR Bi
Generate carry(Gi) = Ai AND Bi

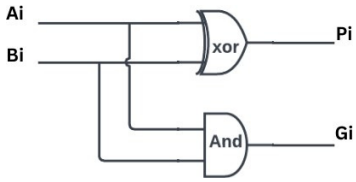


Fig-2.3.2: Logic diagram of the pre-processing stage

(ii) Carry Generation Stage:
This block distinguishes KSA from other adders and is the force behind its superior performance. This step includes calculating the carries associated with each bit. It employs group propagation and generates intermediate signals, which are given by the following logic equations**:**

G= (Pi AND Gi*) +Gi P = (Pi AND Pi*)

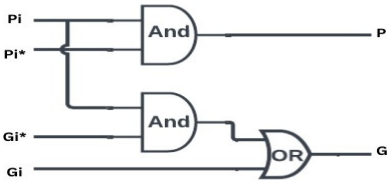


Fig-2.3.3: Logic diagram of carry generator stage

(iii) Post-processing Stage:
This is the final step, which is shared by all adders in this family (carry look ahead). It involves the calculation of sum bits. The logic shown below is used to compute sum bits:

Ci=Gi
Si=Pi XOR Ci-1

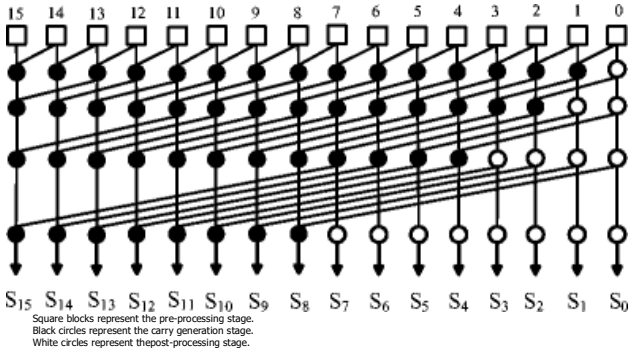


Fig-2.3.4: 16-Bit Kogge-stone adder [9]

RESULTS**:**

(I) Simulation outputs:

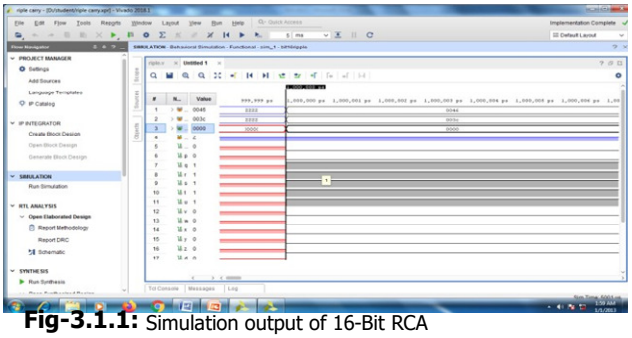


Fig-3.1.1: Simulation output of 16-Bit RCA

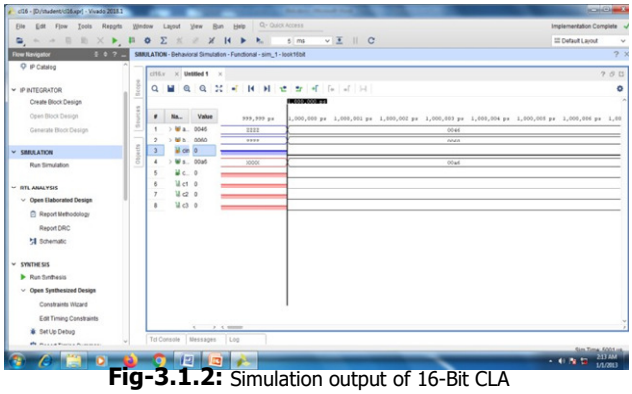


Fig-3.1.2: Simulation output of 16-Bit CLA

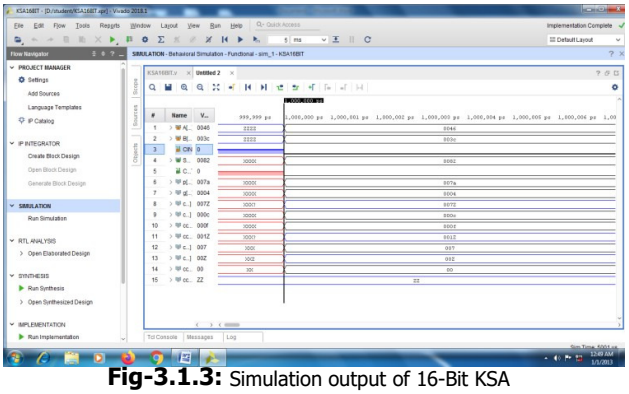


Fig-3.1.3: Simulation output of 16-Bit KSA

(II) Power outputs:

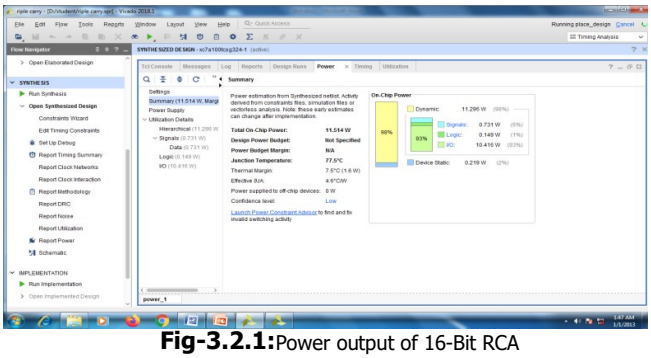


Fig-3.2.1: Power output of 16-Bit RCA

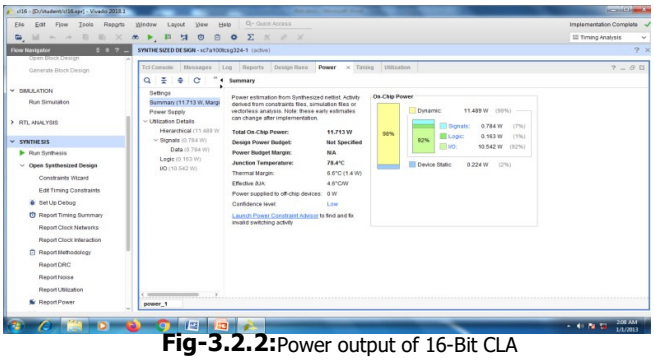


Fig-3.2.2: Power output of 16-Bit CLA

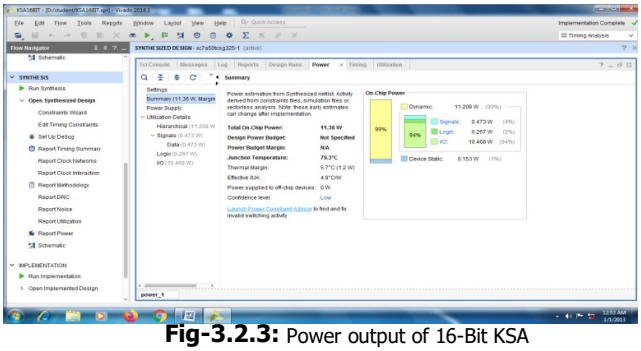
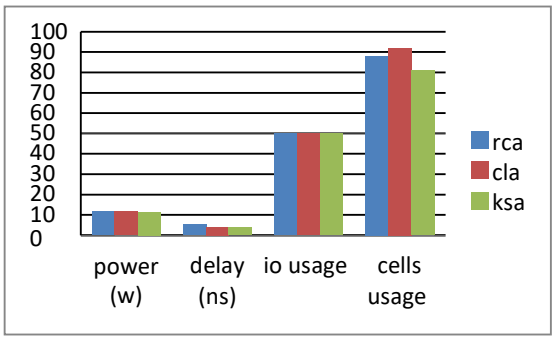


Fig-3.2.3: Power output of 16-Bit KSA

COMPARATIVE ANALYSIS OF ADDERS:

PARAMETERS	**RCA**	**CLA**	**KSA**
POWER	11.514w	11.713w	11.36w
Delay	5.26ns	4.12ns	3.77ns
IO (utilization out of 210)	50	50	50
Cells (rtl schematic)	88	92	81

Table-1: Comparison Table



Graph-1: Comparison Graph

CONCLUSION:

In conclusion, the implementation of the Kogge Stone adder for low-power applications is an effective approach for reducing power consumption in digital circuits. Kogge Stone adder is a parallel prefix adder that has a regular and scalable structure, which allows for efficient implementation and optimization. It is particularly well-suited for applications that require high-speed addition of large numbers, such as in digital signal processing, graphics processing, and cryptography.

In summary, the Kogge Stone adder is a promising adder architecture for low-power applications, with several advantages over other adder architectures. With continued research and development, it is likely to remain a key component in the design of low-power digital circuits.

REFERENCES:

[1] U. Penchalaiah and S. K. VG, "Design of High-Speed and Energy-Efficient Parallel Prefix Kogge Stone Adder," 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA), Pondicherry, India, 2018, pp. 1-7, Doi: 10.1109/ICSCAN.2018.8541143.

[2] B. Koyada, N. Meghana, M. O. Jaleel and P. R. Jeripotula, "A comparative study on adders," 2017 International Conference on Wireless Communications, Signal Processing and Networking (Wisp NET), Chennai, India, 2017, pp. 2226-2230, Doi: 10.1109/WiSPNET.2017.8300155.

[3] S. Daphni and K. S. V. Grace, "A review analysis of parallel prefix adders for better performance in VLSI applications," 2017 IEEE International Conference on Circuits and Systems (ICCS), Thiruvananthapuram, India, 2017, pp. 103-106, Doi: 10.1109/ICCS1.2017.8325971.

[4] Ling, H. (1981). High-speed binary adder. IBM Journal of Research and Development, 25(3), 155–166.

[5] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance

VLSI adders in the energy-delay space," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 6, pp. 754-758, June 2005, doi: 10.1109/TVLSI.2005.848819.

[6] Jasbir Kaur and Lalit Sood, "Comparison between various types of adder topologies", in IJCST, vol. 6, no. 1, Jan.–Mar. 2015.
Source:https://www.researchgate.net/publication/323349524_A_comparative_study_on_adders.

[7] D. Markovic, C. C. Wang, L. P. Alarcon, T.-T. Liu, andJ.M.Rabaey, "Ultralow-power design in the near-threshold region," Proc. IEEE, vol. 98, no. 2, pp. 237–252, Feb. 2010.

[8] R. Zimmermann, "Binary adder architectures for cell-based VLSI and their synthesis," Ph.D. dissertation, Dept. Inf. Technol. Elect. Eng., Swiss Federal Inst. Technol. (ETH), Zürich, Switzerland, 1998
Source:https://www.researchgate.net/publication/3450947_Corrections_to_A_Universal_Architecture_for_Designing_Efficient_Modulo_Multipliers.

[9] Kogge, P., & Stone, H. (1980). A parallel algorithm for the efficient solution of a general class of recurrence equations. IEEE Transactions on Computers, C-22(8), 831–838.
Source:https://www.researchgate.net/publication/220526336_A_Novel_Hybrid_Parallel-Prefix_Adder_Architecture_With_Efficient_Timing-Area_Characteristic.

ABOUT THE AUTHORS

Smt. Eppili Jayacompleted her B. Tech and M. Tech from Aditya Institute of Technology and Management, Tekkali in 2009 & 2011 respectively. Now she is Pursuing Ph. D at JNTU Kakinada, Andhra Pradesh, INDIA, and working as an Assistant professor at Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, INDIA. She has twelve years of working experience in teaching and actively published seven research papers in various international journals. She also attended various Faculty Development Programs to enhance her knowledge in research areas. Her research interests are Digital Signal processing, VLSI, and the Internet of Things
.