Kogge Stone Adder Using Transmission Gate Logic

[1]S Pranava Koundinya, [2]Dr. Kiran V

[1]PG Student, [2]Associate Professor
[1]Department of Electronics and Communication Engineering
[1]R V College of Engineering

***Abstract*: Today's electronics, including DSP and VLSI applications, depend on adders, which are essential parts of every device. Numerous electrical devices employ high-speed adders called Parallel Prefix Adders (PPA). PP Adders have lower latency because adds happen more frequently with less waiting time in between. The addition function must be executed by any control system, whether it be digital, analogue, or both. For the digital system to run swiftly and precisely, adders must be in good working order. In numerous logical and mathematical procedures, adders are employed. Given that improving the performance of digital systems is the goal of VLSI system design research, there is a strong incentive to examine and assess adder architectures. The region typically degrades the adder block's performance, though. In accordance with the parameters of power, latency, and transistor count independently, this paper constructs and analyses a 4-bit Kogge Stone adder using pass transistor logic (PTL) and transmission gate logic (TGL). Finally, an 8-bit Kogge Stone adder is created and simulated using the best performing architecture using Cadence Virtuoso in 45nm Technology.**

***Keywords*: Pass Transistor Logic, Transmission Gate Logic, Kogge Stone Adder, Very Large Scale Integrated design, Parallel Prefix Adder, Cadence Virtuoso, and 45nm Technology.**

I. INTRODUCTION

It is advised to use high-speed parallel design techniques to generate a well-organized collection of mathematical fundamental operations, such as addition, subtraction, division, and multiplication, which are used in VLSI and digital signal processing applications. Various binary carry adders, including as the Carry Skip (CS), Ripple Carry (RC), Carry Select (CSL), Carry Save (CSA), and Carry Look Ahead (CLA) Adders, were previously used to compute the fundamental addition operations. The full and half adders are the two basic binary adders. Each and every binary cascaded adder is built by the entire adder. There are 4 full adders required to do the addition for a 4-bit RCA design. Each full adder has to wait till the previous full adder's process is complete.

Because these carry binary adders are utilised, the design has a high power consumption and a slow rate of operation. Considering that each output carry value is required by the next Full adder in the series of binary adders. Since every result value from the entire adder is saved before further processing, the carry save adder utilises more energy and storage space. The quick Parallel Prefix Adder was utilised by DSP and VLSI-CMOS applications to address the high latency problem. The Parallel Prefix Adder has a similar layout to the Carry Look Ahead Adder. Parallel Prefix Adders have often provided the quick and effective CMOS design applications with low power consumption. Many PPA, such the Brent Kung Adder (BKA), Ladner Fischer Adder (LFA), and Kogge Stone Adder (KSA), were designed in the past.

Kogge and Stone invented the Kogge-Stone Adder, or KSA, a parallel prefix version of CLA in 1973. A carry is generated in 0 (logn) time. It is also known as the faster adder and is utilised in high-performance arithmetic circuits in business. KSA creates carry faster by computing the carry and the area at the same time. KSA provides a standard architecture that enables integrating and utilizing contemporary technologies straightforward. Another advantage of KSA is that it reduces fan-out.

The expression log2n is used to calculate KSA's latency. The area of the KSA is (n*log2n) - n + 1 where n is the total number of input bits. This adder architecture has been shown to perform better in speedier applications. The three steps that comprise the basic operation of KSA are listed below:

Pre-Processing: The signals for generating G(i) and propagating P(i) that correspond to the bit pair in inputs A and B are computed in this stage.

Carry Generation (Using CLA network): This stage, sometimes known as the "carry look ahead network," involves calculating the carries for each bit (i). It uses group propagate and generate as intermediate signals to calculate the signals needed for the final Sum and Cout calculations. The carries are separated into individual components after being concurrently computed. Two AND gates and a single OR gate are present in the carry operator. This stage, sometimes known as the "carry look ahead network," involves calculating the carries for each bit (i). It uses group propagate and generate as intermediate signals to calculate the signals needed for the final Sum and Cout calculations. The carries are separated into individual components after being concurrently computed. Two AND gates and a single OR gate are present in the carry operator.

Post-Processing: All adders in this family share the process of computing the input total (CLA). Both sum and carry call for bit calculations.

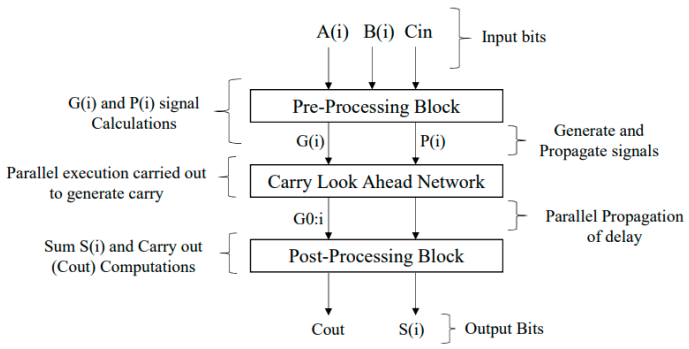

Figure 1: Stages of Operation in Kogge Stone Adder

A Kogge-Stone adder for 8 bits is shown in block diagram form in Figure 2. The propagate-generate (PG) block, the fundamental carry operator (FCO) block, the PPC block, and the XOR logic gate are examples of sub blocks with links to assemble the required logic.

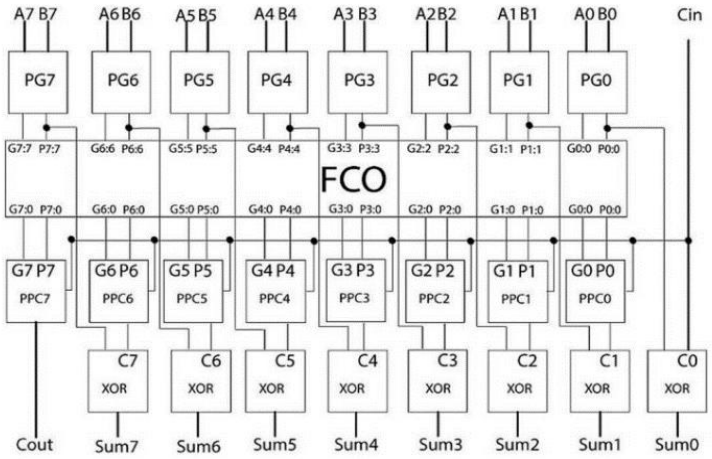

Figure 2: 8-bit KSA block diagram

The performance of the design can be assessed using the delay, power, and performance measures from the previous research work [1]. Based on comparison analysis results, the 4 bit Kogge stone adder definitely surpasses the current design in terms of latency, power consumption, and performance.

II. METHODOLOGY

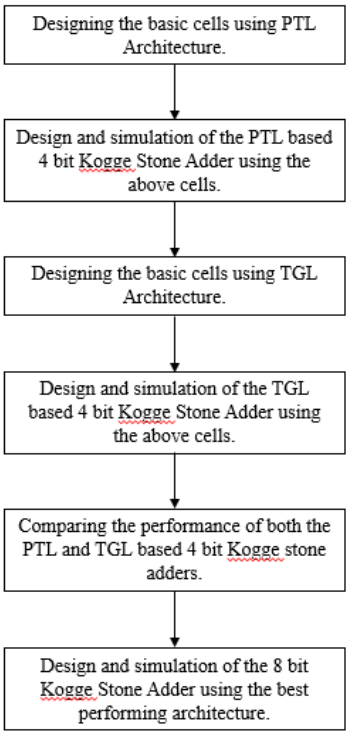

Figure 3: Design Methodology Flow

The fundamental building blocks of the ASIC (Application Specific Integrated Circuit) Design flow are standard cells, which are pre-defined, pre-characterized cells. These equally sized cells can easily be arranged in a row with cells of different widths and heights. They can be applied multiple times. A Boolean logic function is made up of Standard Cells, which are a group of coupled transistors (in both PTL and TGL logic types). In this project, sub-blocks are built using the Standard Cells approach.

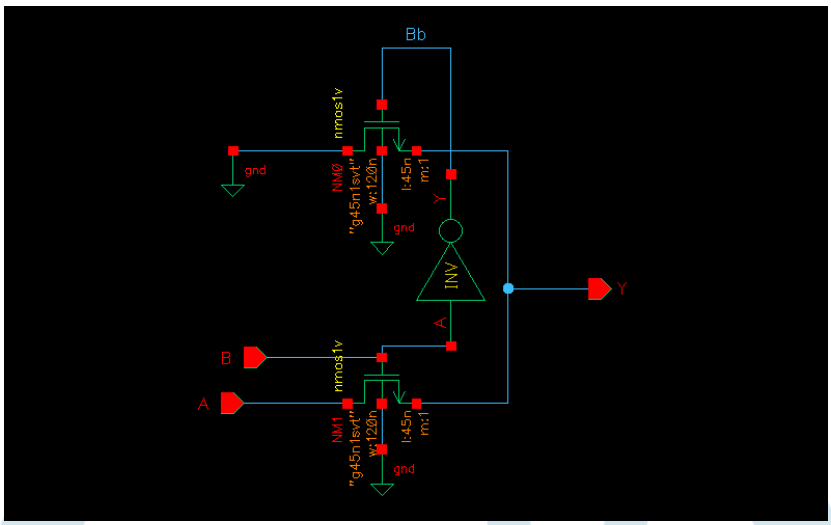

Figure 4: AND gate schematic (PTL)

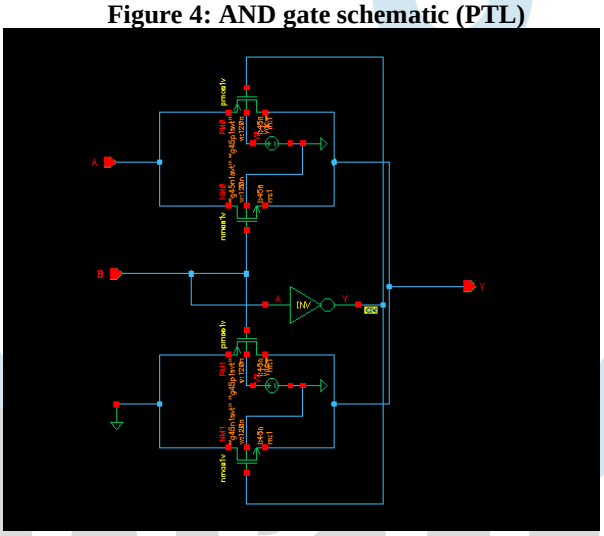

Figure 5: AND gate schematic (TGL)

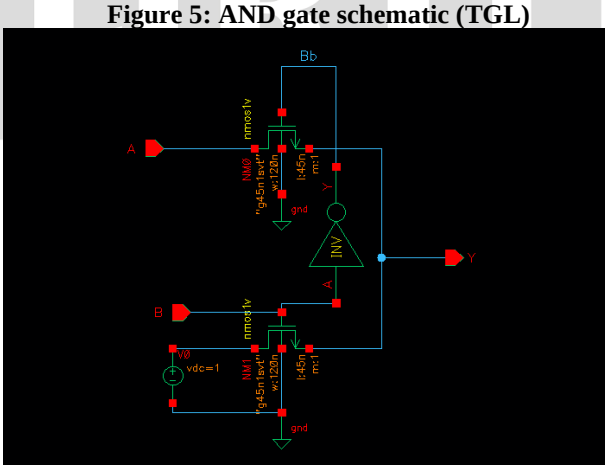

Figure 6: OR gate schematic (PTL)

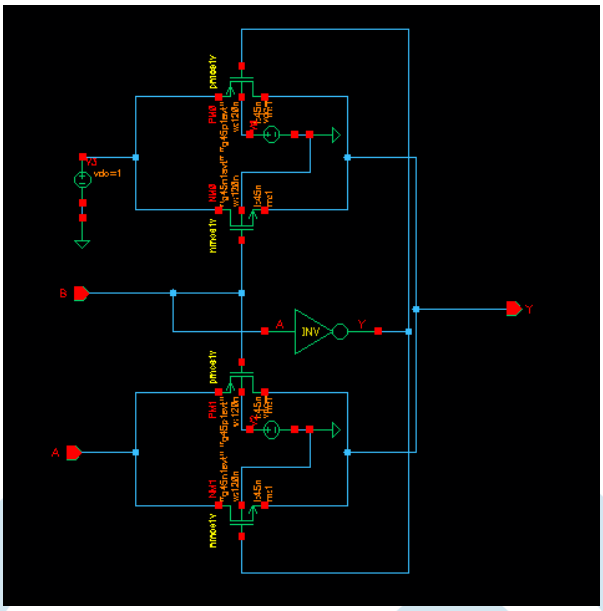

Figure 7: OR gate schematic (TGL)

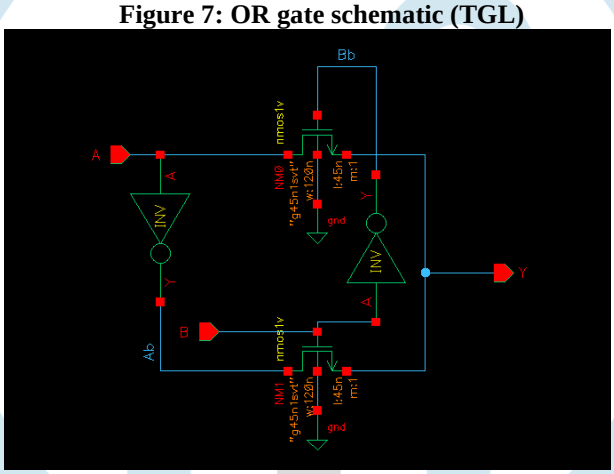

Figure 8: XOR gate schematic (PTL)

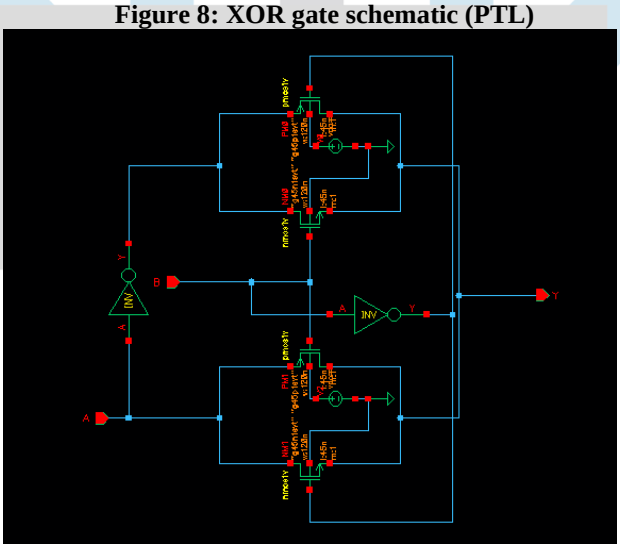

Figure 9: XOR gate schematic (TGL)

The standard cells that were previously described are instantiated in the sub blocks, and linkages are built to obtain the required simulation results. These are the sunblock schematics and symbols, along with their descriptions.

Gen Prop Block: Both AND and an XOR gates are present in the pre-processing unit Gen Prop. This block computes the creation, propagation, and amplitude of the signals. The bits used as inputs for the gates are A and B. Following logical equations generate these signals:

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

$$G(i) = A(i) * B(i) \quad (2)$$

Table 1: Truth table of Gen Prop block

A(i)	B(i)	P(i)	G(i)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

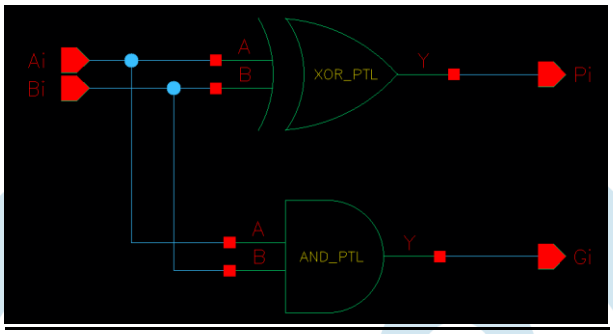

Figure 10: Gen Prop block Schematic using both PTL and TGL Architectures

Gi Pi Block: Two AND gates and one OR gate make up the Gi Pi block of the Carry Look Ahead network. Every bit's carry is generated by this block. The bits that are used as input are Gi, Gj, Pi, and Pj. As intermediary signals, group propagation, and generators, the following logical formulations are employed:

$$Pi:j = Pi:k+1 * Pk:j \quad (3)$$

$$Gi:j = Gi:k+1 + (Pi:k+1 * Gk:j) \quad (4)$$

Table 2: Truth table of Gi Pi block

Gi	Pi	Gj	Gij
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Pi	Pj	Pij
0	0	0
0	1	0
1	0	0
1	1	1

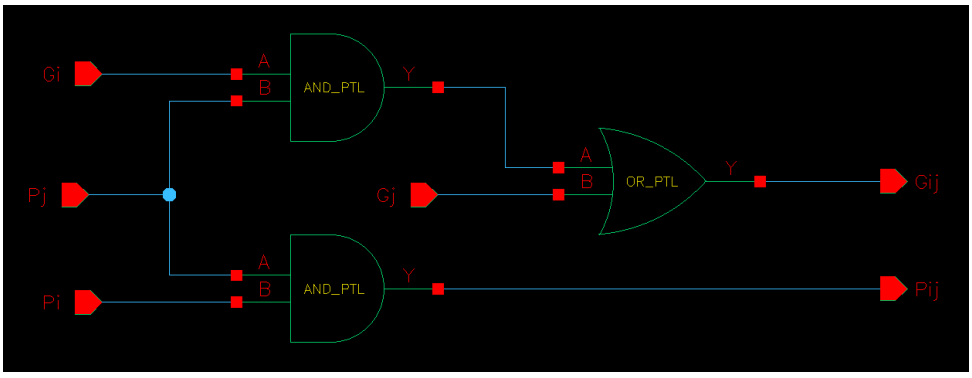

Figure 11: Gi Pi block Schematic using both PTL and TGL Architectures

Gi Block: Both an AND gate and an OR gate are part of the Gi post-processing block. The penultimate stage of every adder in this family shares a block (carry look ahead). Gi, Pj, and Gj make up the input bits of the system. Bit sum procedures come in different varieties. The bit sum is determined by the indicated logic equations:

$$S(i) = Pi \oplus Ci-1 \quad (5)$$

$$Cout = Gi + (Pi * Gi-1) \quad (6)$$

Table 3: Truth table of Gi block

Gi	Pi	Gj	Gij
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

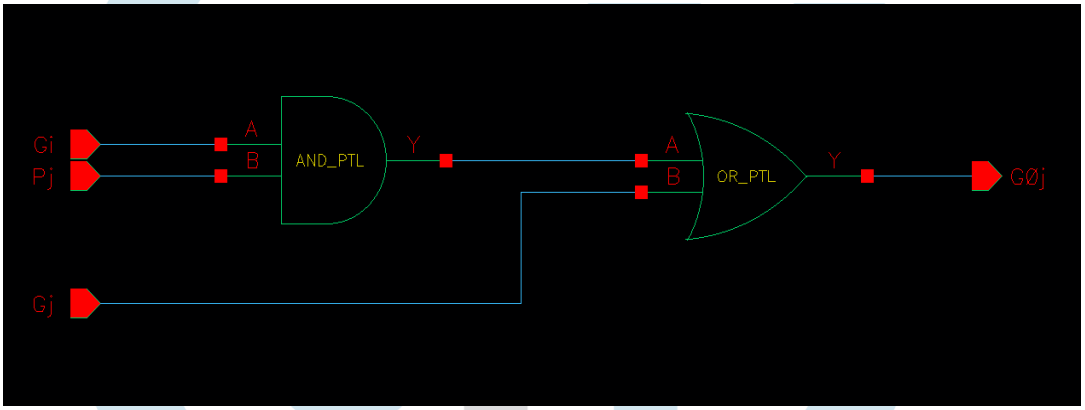

Figure 12: Gi block Schematic using both PTL and TGL Architectures

The top-level design for the 4-bit Kogge Stone Adder is implemented using both PTL and TGL architectures, as shown in Figure 13.

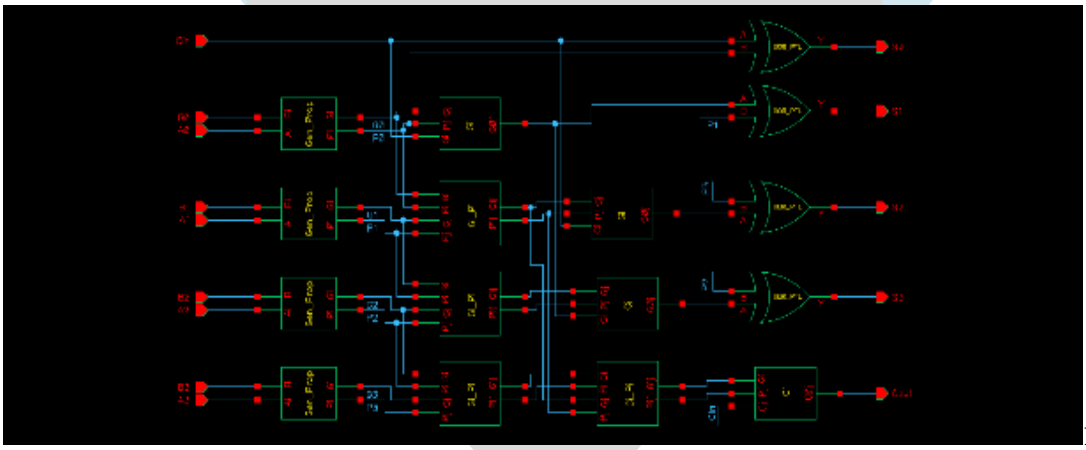

Figure 13: 4-bit Kogge Stone Adder Schematic using both PTL and TGL Architectures

The top-level design for the 8-bit Kogge Stone Adder implemented using TGL architecture is shown in Figure 14.

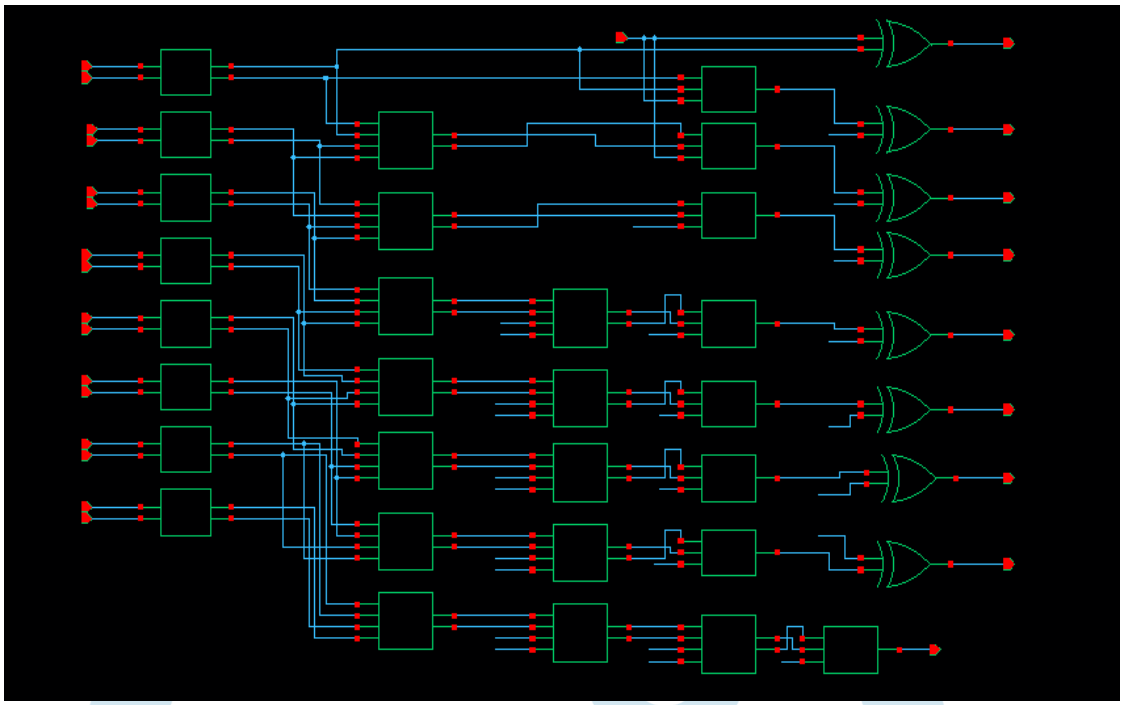

Figure 14: 8-bit Kogge Stone Adder Schematic using TGL Architecture

After simulations on all of the blocks are completed to obtain the parameters, the data is collated and graphically depicted to allow for the parameters to be suitably compared and analysed.
Figure 15 shows the 4-bit Kogge Stone Adder in operation utilising PTL architecture.

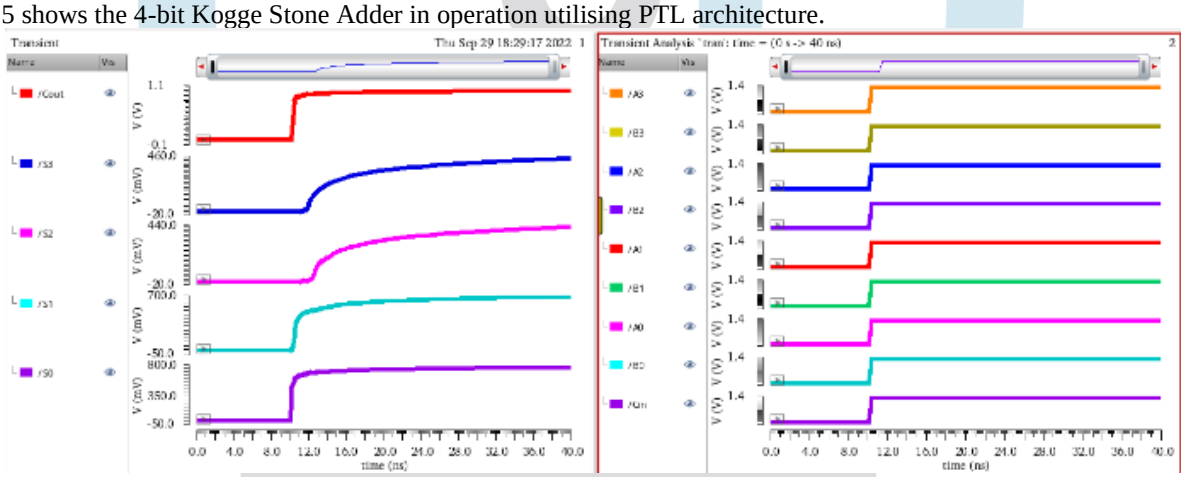

Figure 15: 4-bit Kogge Stone Adder using PTL architecture waveforms

The reduced output voltage swing is evident when we examine the final Sum and Cout's of the adder block in Figure 15. The main drawback of the Pass Transistor Logic design of the Kogge Stone Adder is this.
By enabling us to get the full output voltage swing for the same set of inputs at the expense of a larger surface area, the suggested Transmission Gate Logic architecture could fix this problem. The 4-bit Kogge Stone Adder's waveforms, which use TGL architecture, are shown in Figure 16.

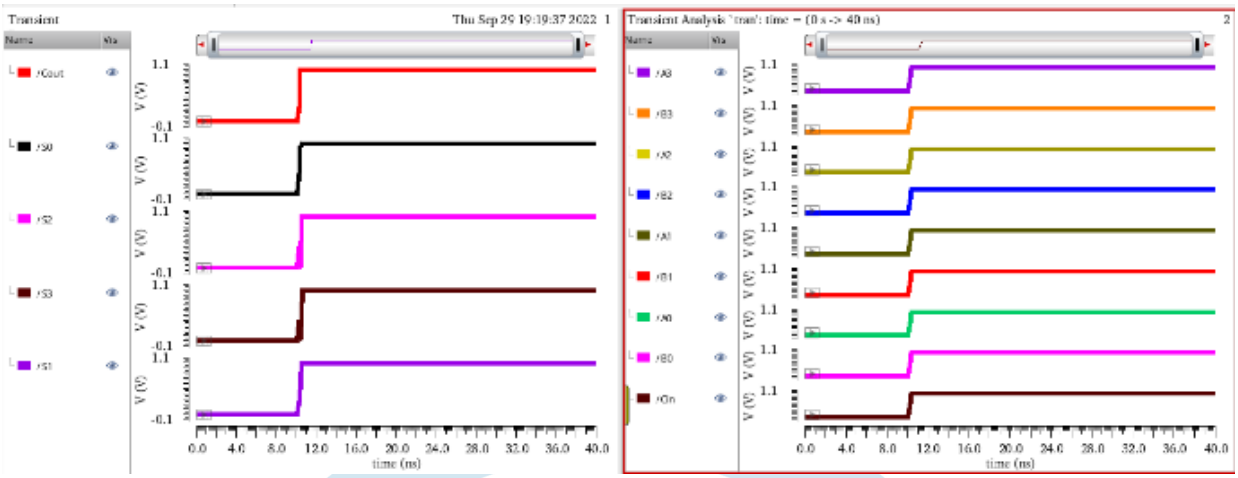

Figure 16: 4-bit Kogge Stone Adder using TGL architecture waveforms

As a result, the proposed TGL architecture is used to design an 8-bit Kogge Stone Adder (as shown in Figure 14) with the waveforms illustrated in Figure 17.

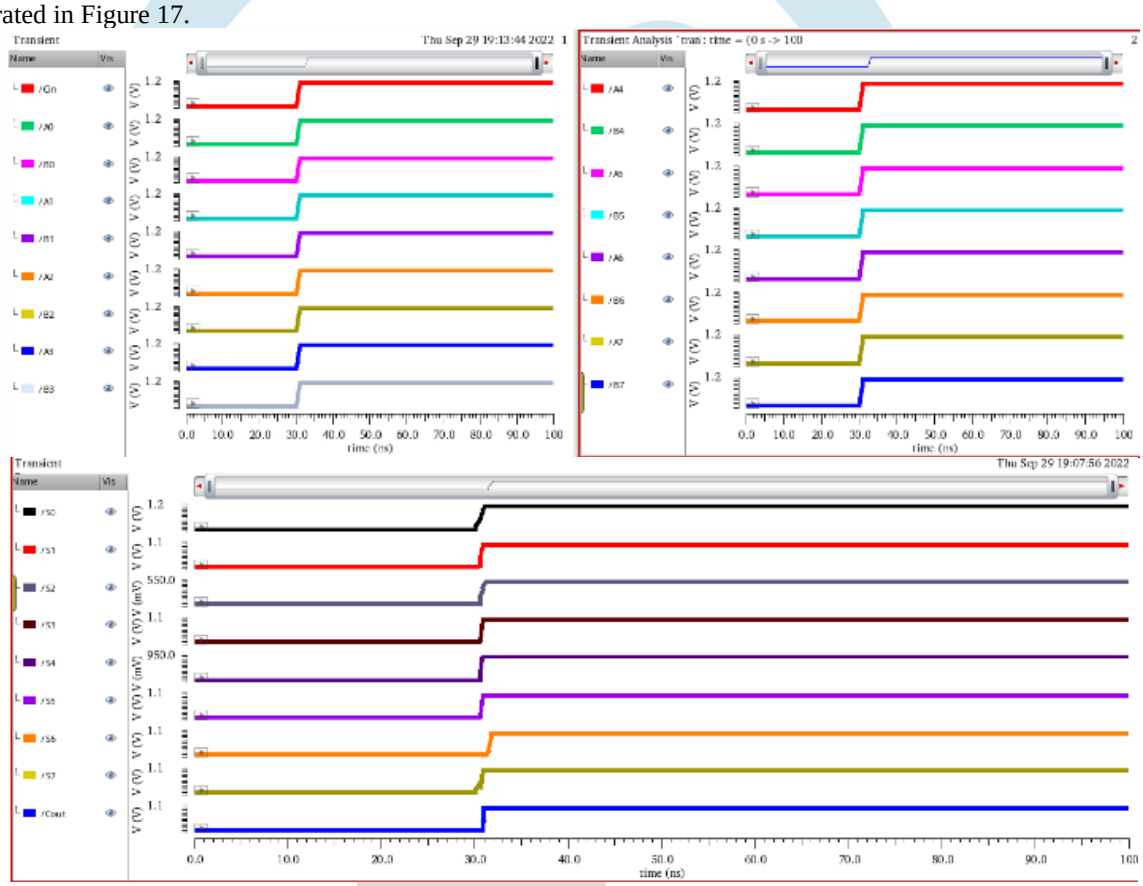

Figure 17: 8-bit Kogge Stone Adder using TGL architecture waveforms

III. RESULTS

The functionality of all primitive structures, sub blocks, and top-level implementations are validated, and simulations utilizing 4-bit and 8-bit models are performed. Delay, Average Power, and transistor/gate count of 4-bit Kogge Stone Adder circuits as shown in the figures 18, 19, and 20 illustrate comparisons of Kogge Stone Adders developed with both PTL and TGL architectures.

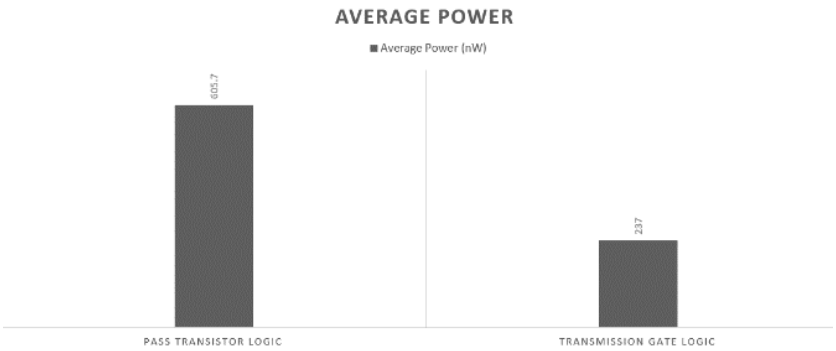

Figure 18: 4-bit Kogge Stone Adder Average Power Comparison

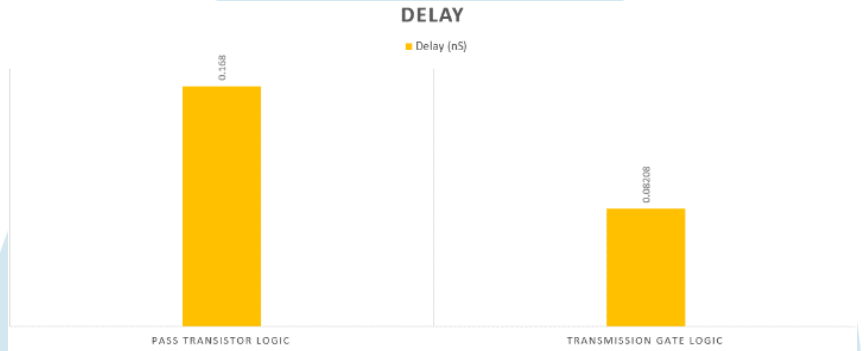

Figure 19: 4-bit Kogge Stone Adder Delay Comparison

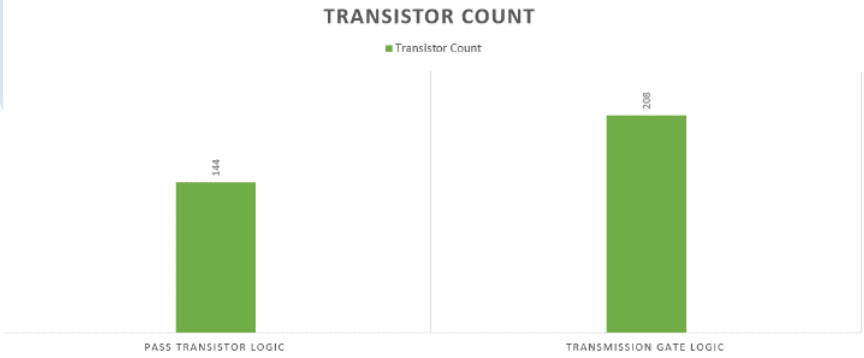

Figure 20: 4-bit Kogge Stone Adder Transistor Count Comparison

Table 4 shows the circuit specifications of the 8-bit Kogge Stone Adder built with the TGL design.

Table 4: 8 bit KSA Parameters

Parameter	Value
Average Power	29.1215uW
Delay	140.723ns
Transistor Count	306

IV. CONCLUSION

The Kogge-Stone Adder, one of the fastest and most effective designs in the class, was to be designed. AND gates, OR gates, EX-OR gates, and Sub-Blocks like the Gen prop Block, Gi Pi Block, and Gi Block are all designed using both the pass transistor and transmission gate based architecture. These cells are built into a 4-bit Kogge Stone adder using both designs, and the results of the performance comparison are presented. Last but not least, a Transmission Gate Based Architecture is used in the 45nm technology node to develop an 8-bit Kogge Stone Adder that is faster and more energy efficient than its predecessor Pass Transistor Logic. Using this adder's simulations, schematics, and symbols, the PPA parameters are thoroughly investigated.

REFERENCES

[1]S. Daphni and K. S. Vijula Grace, "Design an Area Efficient Kogge Stone Adder using Pass Transistor Logic," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021, pp. 614-618, doi: 10.1109/ICICV50876.2021.9388489.

[2] B. Penumutchi, S. Vella and H. Satti, "Kogge Stone Adder with GDI technique in 130nm technology for high performance DSP applications," 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), 2017, pp. 5-10, doi: 10.1109/SmartTechCon.2017.8358334.
[3] H. Harish, C. Kruthika, M. S. Lakshmi Kanth, C. R. Patel and M. S. Ohileshwari, "Design and Implementation of 4-bit and 8-bit KSA in 18nm FinFET Technology," 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), 2022, pp. 1-6, doi: 10.1109/SCEECS54111.2022.9741058.
[4] A. Raju and S. K. Sa, "Design and performance analysis of multipliers using Kogge Stone Adder," 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2017, pp. 94-99, doi: 10.1109/ICATCCT.2017.8389113
[5] A. Prasath A.M., R. V. Arjun, K. Deepaknath and K. Gayathree, "Implementation of optimized digital filter using sklansky adder and kogge stone adder," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 661-664, doi: 10.1109/ICACCS48705.2020.9074440.
[6] N. Poornima, V. S. Kancahana Bhsaskaran, "Area Efficient Hybrid Parallel Prefix Adders," in Procedia Materials Science, Vol.10, PP: 371-380 (2015)
[7] M. Hatem, El - Boghdadi., "A class of almost optimise size independent parallel prefix circuits", in Journal of Parallel and Distributed computing, Vol.73, No.6, PP:888-894 (2013).
[8] Shilpa C. N, Kunjan D. Shinde, Nithin H. V," Design, Implementation and Comparative Analysis of Kogge Stone Adder using CMOS and GDI design: A VLSI Based Approach", in 8th International Conference on Computational Intelligence and Communication Networks IEEE conference, PP: 570-574 (2016).
[9] S. Daphni, K. S. Vijula Grace, "A review analysis of Parallel Prefix Adders for better performnce in VLSI applications", in Proceedings of 2017 IEEE International Conference on Circuits and Systems, PP:103-106, (ICCS 2017).
[10] Amy Mariam George, G. Jyothish Chandran, "Design and analysis of 8 bit parallel prefix comparators using constant delay logic", in Procedia Technology, Vol.24, PP:1178-1185 (2016).
[11] S. Rakesh, K. S. Vijula Grace, "A comprehensive review on the VLSI design performance of different Parallel Prefix Adders", in Materials today proceedings, Vol.11, Part 3, PP:1001-1009, (2019).
[12] Mahesha-Y, Priya-rSeema Miranda,-Jayalakshmi K_P, Keerthana Bhandarkar, Aniceta Priya Dsouza," A ComparativezStudy on Low_Power Adders forzWearable-Devices", International-journal of scientific and technology research volume 9, issue 10, October 2020, pp.1-4,2020.
[13] LeezMei Xiang, Muhammad-MunimzAhmadzZabidi, AinyzHaziyah Awab, Ab-Al-Hadi Ab Rahman, "VLSI-Implementation of a Fast KoggeStone Parallel-Prefix Adder", InternationalzPostgraduate Con- ference on Applied Science and Physics 2017, DOI:10.1088/1742- 6596/1049/1/012077, pp.2-11,2017.
[14] U Penchalaiah, SivazKumar VG, "Design-of High-Speed and-EnergyEfficient Parallel Prefix Kogge Stone Adder", 2018-IEEEzInternational Conferencezon System, Computation, Automation and Networking (ICSCAN), pp.1-7, 2018.
[15] Athira.T.S, Divya R,-Karthik M, ManikandanzA, "Design-of-Kogge for Fast_Addition", InternationalzJournal of2IndustrialzElectronicszand ElectricalzEngineering, ISSN: 2347-6982, Volume-5, Issue-4, pp.1-3, 2017.