



A NOVEL ARCHITECTURE OF AREA AND POWER EFFICIENT KOGGE STONE ADDER USING GDI LOGIC

S Mrinal Kashyap, S Vijayakumar, M Saravanan

VLSI System Design, Electronics and communication Engineering,
Sreenivasa Institute of Technology and Management Studies (Autonomous), Chittoor, Andhra Pradesh, India-517127

Abstract : The paper describes about the novel architecture of Kogges tone adder using GDI Logic. The Kogge Stone adder is a widely used high-speed parallel adder, and the proposed design aims to reduce its area overhead while maintaining its performance characteristics. The paper presents a detailed analysis of the GDI logic-based Kogge Stone adder, including its circuit design, area estimation, and delay analysis. The proposed design is compared with other Kogge Stone adders, including the conventional Kogge Stone adder designed using CMOS and PTL Logics. Therefore, the proposed design can be a good alternative for Low power and less area for arithmetic operations in digital signal processing and other related applications.

IndexTerms - Parallel Prefix Adder, Pass Transistor Logic, Kogge Stone Adder, Gate Diffusion Inputs, Very Large-Scale Integrated design, Area consumption.

I. INTRODUCTION

Parallel prefix adders (PPAs) are specialized digital circuits designed to perform rapid addition of two n-bit binary numbers in high-performance computing applications. Unlike traditional ripple-carry adders, which process bits sequentially and can be slow due to carry propagation delays, PPAs leverage a carry-lookahead technique. This allows multiple full adders to operate simultaneously, significantly reducing latency. PPAs are available in several architectures such as Kogge-Stone, Brent-Kung, and Han-Carlson adders, each offering different trade-offs in speed, area efficiency, and power consumption. These designs are critical in applications where computational speed and efficiency are paramount. This paper focuses on optimizing the Kogge-Stone PPA, known for its lower delay and power consumption but with concerns over area efficiency. We propose an area-efficient design using Pass Transistor Logic in DSP and VLSI-CMOS technologies. This approach aims to maintain the fast operation of the Kogge-Stone PPA while reducing its area footprint. The structure of this paper includes a review of related work on adder designs, details of our proposed Pass Transistor Logic-based Kogge-Stone PPA design, comparative performance analysis with the conventional design, and conclusions with implications for future research in efficient arithmetic circuits. This introduction captures the essence of PPAs, highlights the motivation for optimizing their design, and sets the stage for the specific contributions and structure of your research paper.

II. EXISTING METHOD

The Kogge-Stone adder is a powerful type of parallel prefix adder designed for rapid binary number addition. Introduced by Kogge and Stone in 1973, it employs a hierarchical tree structure where each node represents a partial sum of the binary numbers being added. This tree-like arrangement allows for simultaneous processing of all bits, leading to faster addition compared to sequential methods. Implemented using CMOS (Complementary Metal-Oxide-Semiconductor) technology, the Kogge-Stone adder utilizes both p-type and n-type transistors to create logic gates. These gates are interconnected to handle input signals and generate output sums efficiently. The adder's parallel processing capability leverages the inherent speed advantages of CMOS circuits, making it suitable for high-performance applications. Despite its efficiency, the Kogge-Stone adder can become complex and require a large number of interconnects due to its recursive doubling algorithm. It achieves low fanout and optimal depth characteristics, contributing to its suitability for high-speed operations. The adder's regular layout further enhances its utility in digital systems requiring fast arithmetic operations. In essence, the Kogge-Stone adder exemplifies a balance between complexity and performance efficiency in parallel prefix adder designs, making it a valuable component in modern digital circuitry.

III. PROPOSED METHOD

In this section, we explore the implementation of the Kogge-Stone adder using GDI (Gate Diffusion Input) logic. GDI logic is a methodology that utilizes transistor-level designs to create logic gates with reduced complexity and power consumption. It employs a combination of gate diffusion and multiple-input transistors to achieve logic functions efficiently. The chapter covers an introduction to GDI logic and details the implementation of various logic gates using this technique. Specifically, it focuses on how GDI logic can be applied to design the components required for the Kogge-Stone adder, leveraging its advantages in terms of simplicity and reduced power consumption compared to traditional CMOS implementations.

By utilizing GDI logic, the goal is to demonstrate an effective approach to implementing the Kogge-Stone adder while optimizing for performance metrics such as speed, area efficiency, and power consumption. This chapter provides insights into the practical application of GDI logic in digital circuit design, highlighting its potential benefits in enhancing circuit efficiency and reducing implementation complexity.

GDI:

Gate Diffusion Input or GDI technique is a new method for designing circuits which reduces the power requirements considerably. Also, GDI results in a reduced number of transistors, which in turn makes for a decrease in chip-area giving an edge to the designs over conventional methods.

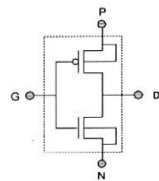


Table 1: Various Logic Functions Implemented with of Basic GDI Cell

N	P	G	D	Functions
Low	B	A	A'B	F1
B	High	A	A' + B	F2
High	B	A	A + B	OR
B	Low	A	AB	AND
C	B	A	A'B + AC	MUX
Low	High	A	A'	NOT

Fig No: 1 GDI Logic

In the realm of GDI (Gate Diffusion Input) logic, a fundamental cell comprises just two transistors: one NMOS and one PMOS. This basic structure features four terminals: G (gate), N (NMOS source), P (PMOS source), and D (drain), where G, N, and P serve as input terminals. One challenge with traditional GDI is its swing problem, arising because NMOS transistors produce a weak logic 1 and PMOS transistors produce a weak logic 0. However, modifications to the conventional GDI technique have effectively addressed this issue, resulting in enhanced efficiency compared to PTL (Pass Transistor Logic) and CMOS.

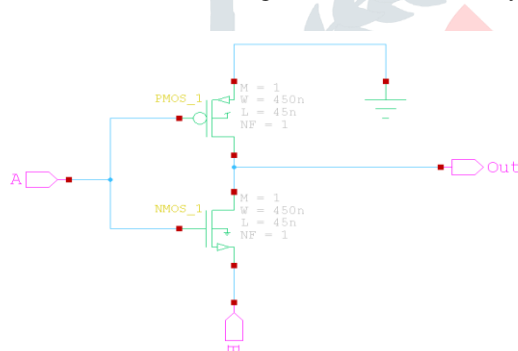


Fig No: 2 Schematic of AND Gate using GDI

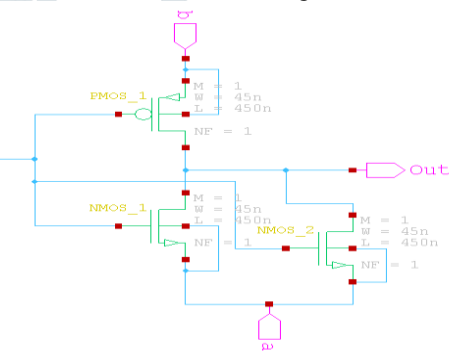


Fig No: 3 Schematic of OR gate using GDI

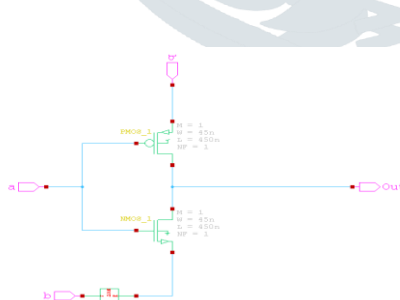


Fig No: 4 Schematic of XOR using GDI

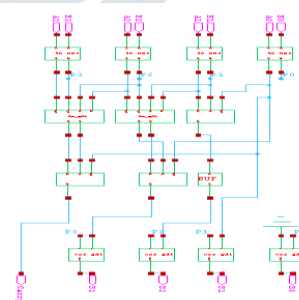


Fig No: 5 Schematic of KSA using GDI

3.1 TANNER TOOL:

To launch S-Edit, double-click on the S-Edit icon.

The user interface consists of the elements shown below. Unless you explicitly retrieve a setup file, the position, docking status and other display characteristics are saved with a design and will be restored when the design is loaded.



3.2 PARTS OF THE USER INTERFACE

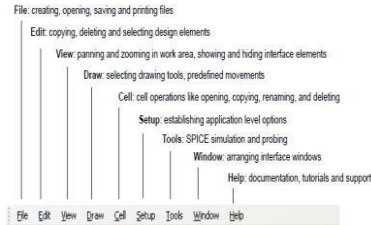
Title Bar:

The title bar shows the name of the current cell and the view type (symbol, schematic, etc.).



Menu Bar:

The menu bar contains the S-Edit menu titles. The menu displayed may vary depending on the view type that is active. See “Shortcuts for Cell and View Commands” on page 70 for the various methods S-Edit provides for executing commands.



Menu List Filtering:

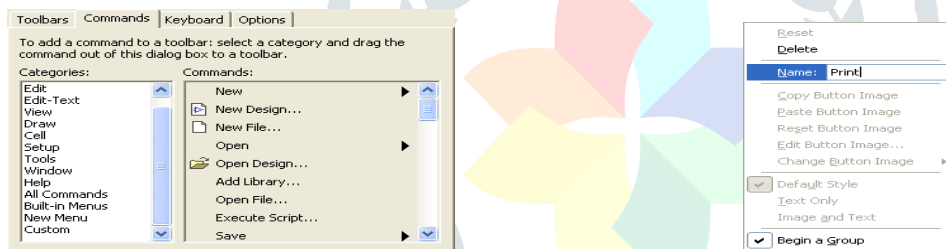
Most S-Edit menus and dialogs allow for filtering to speed the process of selecting from a drop-down list. So, when you enter a character, S-Edit will jump to the first list item that begins with that character. For example, typing g highlights the first list item beginning with that letter and filters the display to show only items that begin with g. Typing a u after the g highlights the first list item beginning with gu, and filters the display to show only items that begin with gu, and so on. The search procedure is case-insensitive.

Toolbars:

You can display or hide individual toolbars using the View > Toolbars command, or by right-clicking in the toolbar region. Toolbars can be relocated and docked as you like. For added convenience, S-Edit displays a tool tip when the cursor hovers over an icon.

ADDING A COMMAND TO A TOOLBAR & ADDING A NEW MENU

Use the Commands tab to add a button for an existing command to any toolbar.



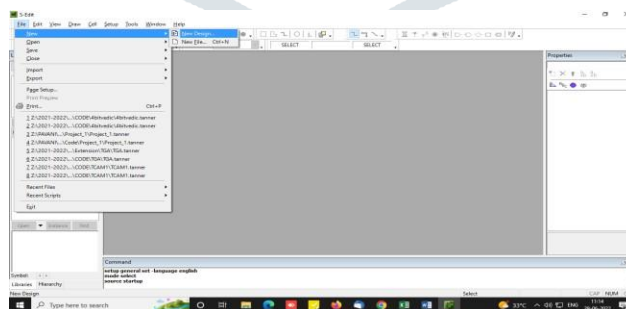
OPENING S-EDIT PLATFORM:

First of all double click on the icon of s-edit on the desktop

A new window will open

Go to >>file >> New >> New Design

Select New Design

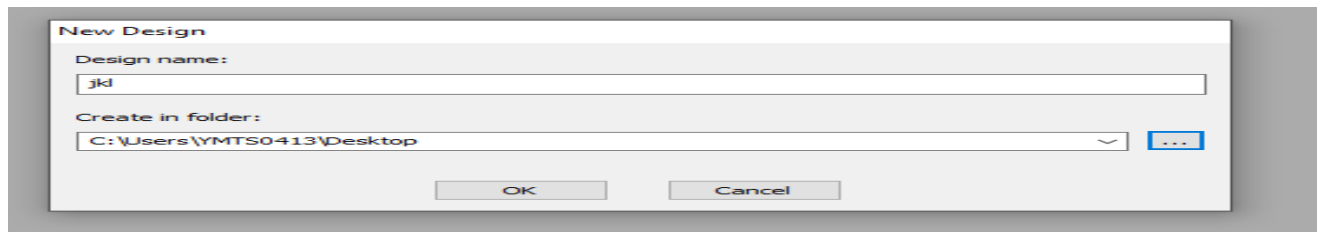


One dialog box will appear

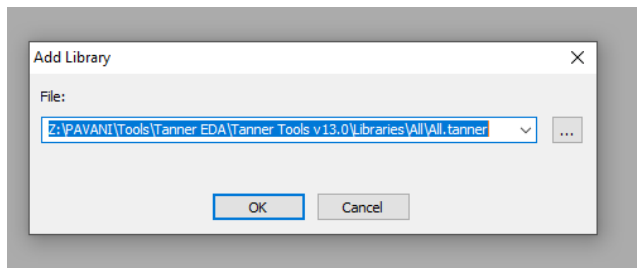
Design Name : Give the name your design as you wish

Create a Folder : Give the path where you want to save the S-Edit Files.

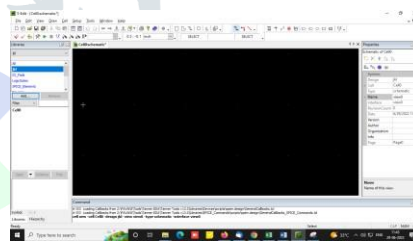
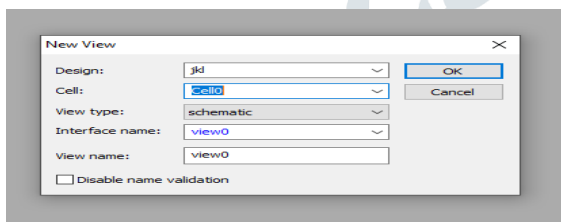
Then Click on 'OK'



Now to add libraries in your work click on Add , left on the library window.
Give the path where Libraries are stored .



Now to create new cell
Go to cell menu >> New view --
Select 'New view'
The new cell will appear like below:
Design = your design name
Cell = cell no. (cell no you can change but your design name inv will be same for different cell.
Design name should be changed only when you are going to design another circuit)
View type = schematic
Interface name = "by default"
View name = "by default"
Then press "OK".

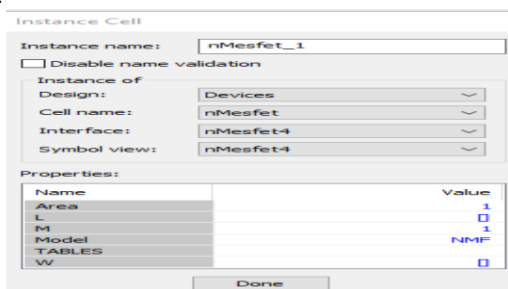


Then a cell will be appeared where we can draw the schematic of any circuit. In the black window you have seen some white bubble arranged in specific order. This is called grid. You can change grid distance by clicking on black screen and then scroll the mouse.

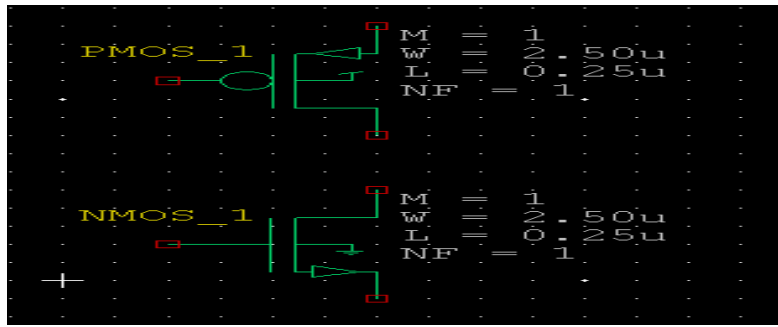
If you want your screen big enough for design space , then you can close the Find & command window. You can again bring these window from view menu bar.

IN INSTANCE CELL

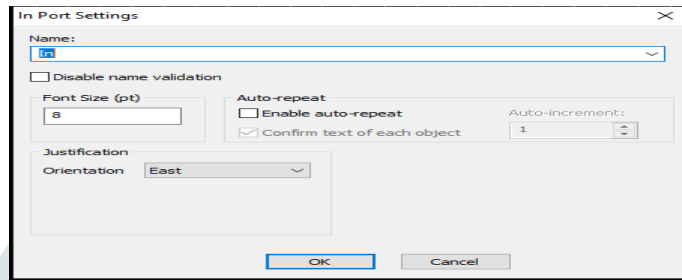
- You can change the values of various device parameters according to your requirements.
- Go to properties >> change the parameter values as your requirement.
- Now before clicking DONE you have to DRAG the selected device into the cell and drop it where you want it to FIX .
- Then click DONE or press ESC.



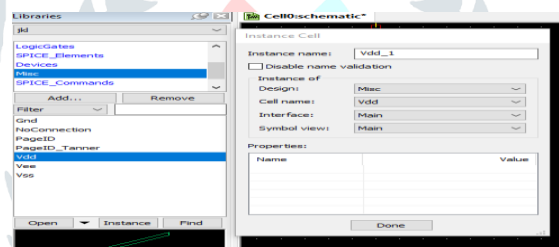
Similarly you can DRAG & DROP any device into the cell for draw your schematic circuit.
For inverter we need another Pmos.



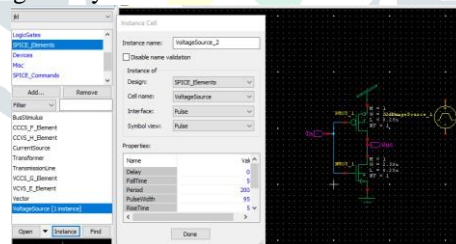
Now connect two device with wire.
 Go to tool bar and select wire.
 Similarly to give input & output port in the circuit , select input port that shown by red ellipse.
 Now you can give Port name as you wish in the dialog box.
 Then click OK



Similarly give Output Port name.
 NOTE : you can rotate the port (short cut key "R").
 Now, after completed these steps, you should give the supply (VDD) & ground (GND).
 For that Go to libraries >> MISC >>Select VDD or GND



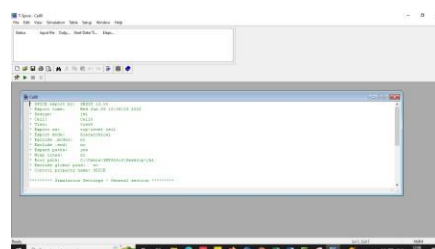
Now you have to create a source of VDD. For that go to libraries >>spice_element >> and then select voltage source of type DC . you can give any value in vdd .lets take vdd =5v.



By doing all the above steps you have completed schematic of Inverter
 Pre layout simulation
 After schematic design you have to check whether your design match with the specification required or not . That's why you need to simulate the design which is called Pre layout simulation.



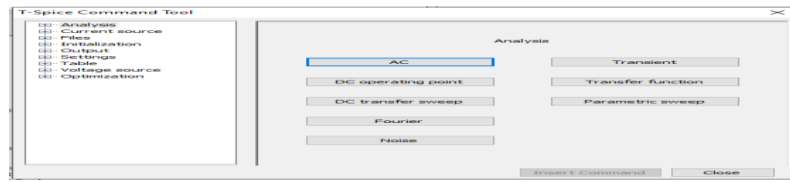
A T-spice window will open.



Then click on the bar shown by red ellipse



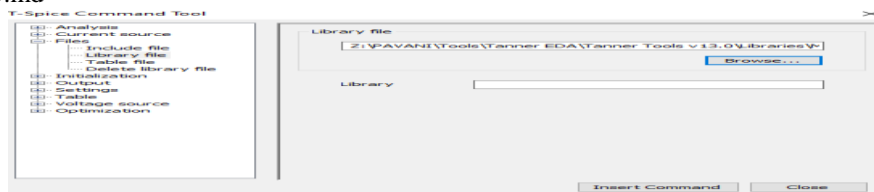
A “T-spice command Tool “ dialog box will open as shown beow.



For that

Go to >> T-spice command tool >> Files >> Include >> browse TSMC .25µm files >> Insert command.

C:\Documents and Settings\Bhowmik.IIIT-3AC288AD0A\Desktop\TSMC 0.25um\MODEL_0.25.md



File is included shown by highlight.

```
***** Simulation Settings - Additional SPICE commands *****
.lib "Z:\PAVANI\Tools\Tanner EDA\Tanner Tools v13.0\Libraries\Models\Generic_025.lib"
.end
```

Step2 : Then to give Input

T-spice command tool >> Voltage source >> select type of input you want to give(lets take bit) >> Insert command

Step 3: Analysis

T-spice command tool >> Analysis >> select type of analysis you want to give(lets take transient) >> Insert command

step 4: Output

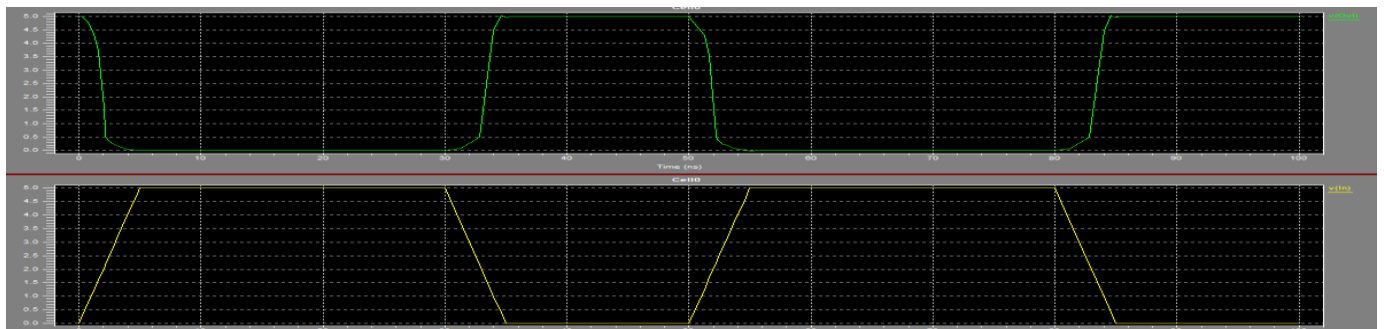
T-spice command tool >> Output >> which output you want to see >> Insert Command

The total spice netlist will come like this.

```
***** Simulation Settings - Analysis section *****
***** Simulation Settings - Additional SPICE commands *****
.lib "Z:\PAVANI\Tools\Tanner EDA\Tanner Tools v13.0\Libraries\Models\Generic_025.lib"
.tran 5n 100n
v1 vdd GND 5
.print v(In) v(Out)
.end
```

Now save it . And run.

Output of Pre layout simulation of Inverter



IV. SIMULATION TOOL

The design cycle for the development of electronic circuits includes an important pre-fabrication verification phase. Because of the expense and time pressures associated with the fabrication step, accurate verification is crucial to efficient design. The role

of EDA tool is to help design and verify a circuit's operation by numerically solving the differential equations describing the circuit. These simulation results allow circuit designers to verify and fine-tune designs before submitting them for fabrication. Tanner EDA tool is a complete circuit design and analysis system that includes:

Schematic Editor (S-Edit): Schematic editor is a powerful design capture and analysis package that can generate netlist directly usable in T-Spice simulations. **T-Spice Circuit Simulator:** T-Spice performs fast and accurate simulation of analog and mixed analog/digital circuits. The simulator includes the latest and best device models available, as well as coupled line models and support for userdefined device models via tables or C functions. T-Spice uses an extended version of the SPICE input language that is compatible with all industry standard SPICE simulation programs. All of SPICE's device models are incorporated, as well as resistors, capacitors, inductors, mutual inductors, single and coupled transmission lines, current sources, voltage sources, controlled sources, and a full complement of the latest advanced semiconductor device models from Berkeley and Philips Labs. **Waveform Editor (W-Edit):** W-Edit displays T-Spice simulation output waveforms as they are being generated during simulation. Visualizing the complex numerical data resulting from VLSI circuit simulation is critical to testing, understanding, and improving those circuits. W-Edit is a waveform viewer that provides ease of use, power, and speed in a flexible environment designed for graphical data presentation. **Layout Editor (L-Edit):** Tanner EDA tool includes L-Edit for layout editing, • Interactive DRC for real-time design rule checking during editing, Standard DRC for hierarchical DRC, Standard Extract for netlist extraction, Standard LVS for layout versus schematic, Node Highlighting for highlighting all geometry associated with a node and SPR for standard cell place & route.

4.1 RESULT

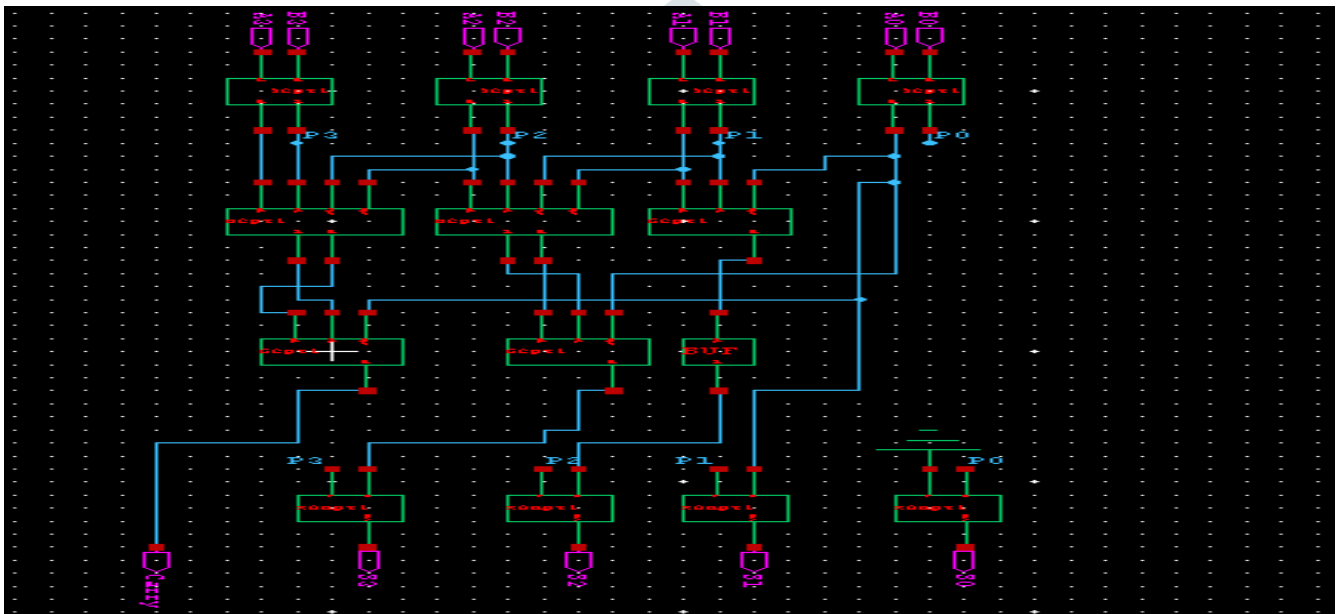


Fig No: 6 CMOS schematic diagram

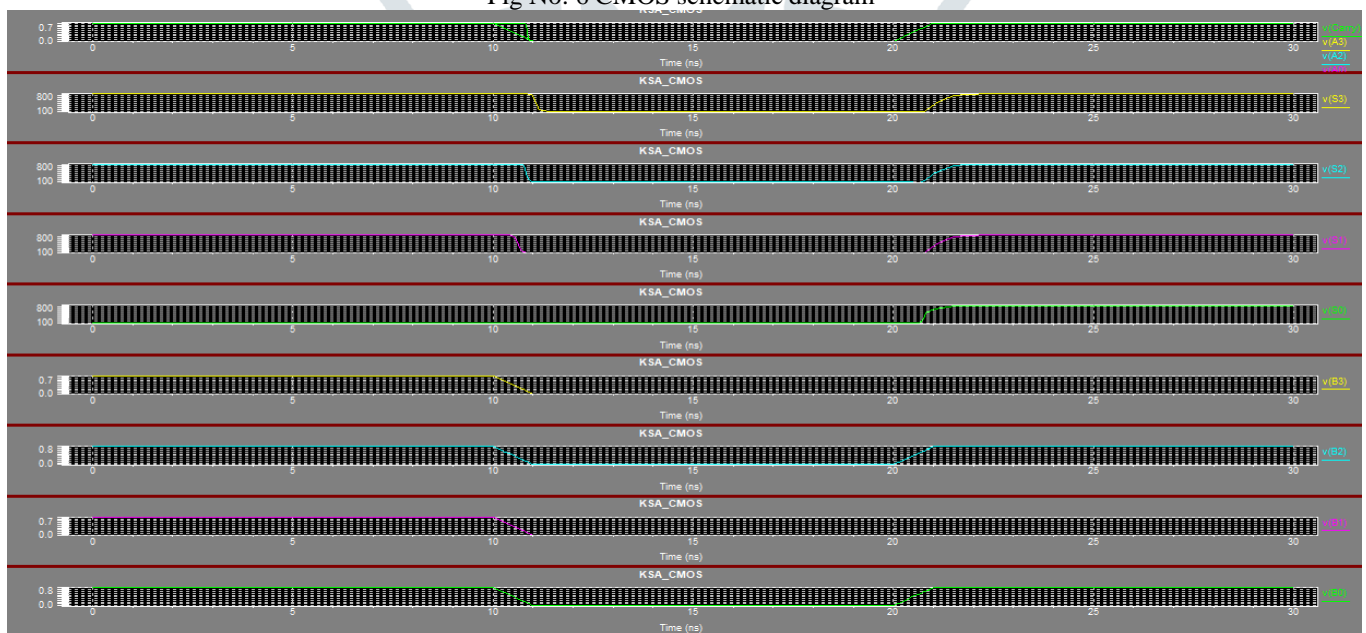


Fig No: 7 Schematic waveform of a CMOS



Fig No: 8 Schematic waveform of a PP-KSA

	Power result	Delay	Area
CMOS	5.721033e-004 watts	-6.1436e-010	196
PTL	4.320054e-004 watts	9.5534e-009	116
GDI	9.675e-005 watts	6.6645e-010	79

It is clear that by using GDI logic, the power, delay and area is reduced. If we look at the power consumption in CMOS logic and PTL logic, the consumption is around 5.721033e-004 watts and . 4.320054e-004 watts, But if we look at GDI logic the consumption is 9.675e-005 watts. Hence, the area will be reduced and if delay is reduced the efficiency is increased. Therefore, by using GDI logic the area, delay and power consumption is decreased.

V. CONCLUSION

In conclusion, the proposed architecture of an area and power-efficient Kogge-Stone adder using GDI logic presents several compelling advantages over traditional PTL and CMOS logics. By leveraging GDI logic, the design achieves a streamlined approach with a reduced transistor count, thereby leading to decreased power consumption and minimized area requirements. Moreover, GDI logic enables straightforward scalability and enhances noise immunity, further bolstering the reliability of the adder design. The Kogge-Stone adder itself is renowned for its high-performance parallel processing capabilities, particularly beneficial in digital signal processing and arithmetic-intensive applications. The integration of GDI logic into its architecture offers an efficient and effective solution, delivering improved performance metrics alongside reduced power demands. This combination positions the adder architecture favorably for deployment across diverse applications such as digital signal processing, high-speed communication systems, and microprocessor designs. Overall, the proposed architecture utilizing GDI logic represents a promising advancement in the field of adder design. It not only enhances efficiency and reliability but also aligns well with contemporary demands for high-performance computing solutions that prioritize both functionality and energy efficiency.

VI. REFERENCES

1. N. Poornima, V. S. Kanchana Bhsakaran, "Area Efficient Hybrid Parallel Prefix Adders," in *Procedia Materials Science*, Vol.10, PP: 371-380 (2015).
2. S. Daphni, K. S. Vijula Grace, "Design and Analysis of 32-bit Parallel Prefix Adders for Low Power VLSI Applications" in *Advances in Science, Technology and Engineering Systems Journal* Vol. 4, No. 2, PP: 102-106 (2019).
3. 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).
4. 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).
5. 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).
6. 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).
7. 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).
8. Li - Wen Chang, Juan Gómez - Luna, "Chapter 8 - Parallel patterns: prefix sum: An introduction to work efficiency in parallel algorithms", in *Programming Massively Parallel Processors (Third Edition)*, PP: 175-197 (2017).
9. Marc Snir, "Depth-size trade-offs for parallel prefix computation", in *Journal of Algorithms*, Vol. 7, no. 2, (1986).
10. Reshma Mary James, Ajith Ravindran, "Review of Full Adder Performance Analysis Using Kogge Stone Adder And Magnetic Tunnel Junction", in *Fourth International Conference on Devices, Circuits and Systems (ICDCS'18)*, PP: 84-90 (2018).