

## TABLE OF CONTENTS

<b>SECTION:</b>	<b>PAGE</b>
1. ABSTRACT.....	3
2. INTRODUCTION .....	4
3. OBJECTIVE .....	6
4. BACKGROUND .....	6
5. METHODOLOGY .....	9
6. RESULT AND VERIFICATION .....	11
a) Functional Verification in ModelSim .....	11
b) Schematic Design and Stimulation in Cadence Virtuoso .....	12
c) Synthesis and Performance Analysis Using Synopsys .....	30
d) Layout Design in Cadence Virtuoso .....	36
7. Results and Analysis .....	40
8. CONCLUSION.....	44
9. REFERENCES .....	45

# **Performance Comparison of CMOS and TGL-Based Low-Power Full Adders for High-Efficiency Digital Circuits**

## **1. ABSTRACT**

The exponential growth of mobile computing, wearable devices, Internet of Things (IoT) systems, and edge AI accelerators has placed an increasing demand on the semiconductor industry to develop energy-efficient computing hardware. Among the critical components of digital systems, arithmetic circuits such as full adders play a pivotal role in determining the overall computational performance, power efficiency, and area utilization. Full adders are the building blocks of various complex digital operations including addition, subtraction, multiplication, and data path processing in microprocessors, digital signal processors (DSPs), and AI acceleration hardware [2].

However, conventional full adder designs based on Complementary Metal-Oxide-Semiconductor (CMOS) technology suffer from significant limitations including increased transistor count, high dynamic and static power dissipation, and larger silicon area requirements. These challenges make CMOS-based adders suboptimal for power-constrained applications, especially in portable and embedded systems where battery life and thermal management are critical design considerations. Furthermore, the excessive switching activity in CMOS designs leads to increased energy loss, limiting their suitability for high-performance low-power computing environments [1][2].

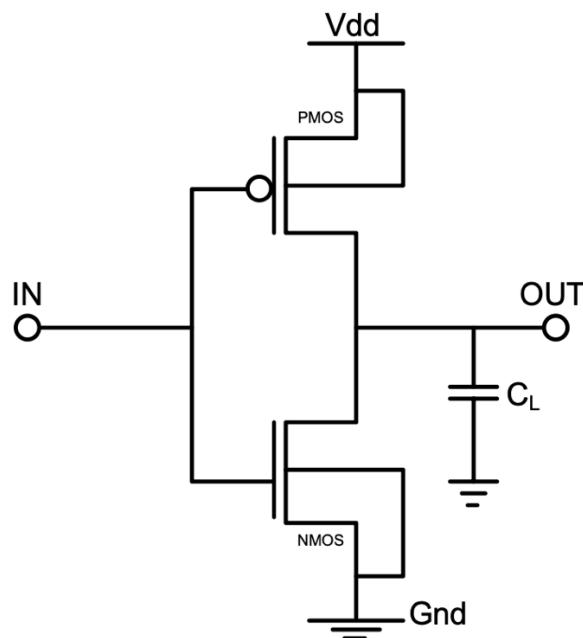
This project aims to address these challenges by presenting a Transmission Gate Logic (TGL)-based 1-bit full adder, designed to reduce power consumption, minimize transistor count, and maintain high computational accuracy. TGL employs complementary NMOS and PMOS transistors working in parallel to achieve full voltage swing, superior signal integrity, and reduced leakage current, making it ideal for low-power VLSI circuits [3]. The TGL-based full adder is designed at the transistor level using Cadence Virtuoso, with thorough functional validation carried out using ModelSim to ensure correct operation across all possible input combinations. In addition, comprehensive power, area, and timing analyses are conducted using Synopsys tools to benchmark the proposed design against traditional CMOS-based full adders [1][3].

The expected outcomes of this research will demonstrate that the proposed TGL-based full adder not only significantly reduces power consumption and transistor count but also achieves competitive performance in terms of speed and area efficiency [4]. These advantages position the TGL full adder as a promising solution

for energy-efficient VLSI systems, including battery-operated devices, AI accelerators, and next-generation portable computing platforms where power and performance are equally critical [2].

## 2. INTRODUCTION

In modern digital system design, arithmetic circuits form the computational backbone that enables essential data processing operations in applications ranging from consumer electronics to high-performance computing platforms. Among these, full adders are key components that perform binary addition, a fundamental operation in arithmetic logic units (ALUs), microprocessors, digital signal processors (DSPs), and artificial intelligence (AI) accelerators. The efficiency of these circuits directly influences the overall system performance, making the design of energy-efficient and high-speed full adders a critical research objective in the field of Very Large-Scale Integration (VLSI) design [5].

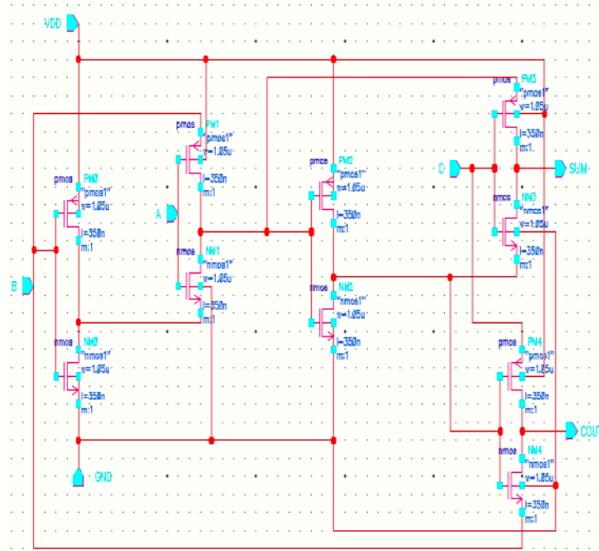


**Fig 1: CMOS Inverter [1]**

The continued miniaturization of semiconductor devices and the proliferation of portable and embedded systems have intensified the need for low-power circuit solutions. Traditional full adders, typically implemented using Complementary Metal-Oxide-Semiconductor (CMOS) technology, are known for their robust operation and compatibility with standard digital design flows. However, they also present significant drawbacks including a high transistor count, increased silicon area utilization, and substantial dynamic and static power dissipation [2][3]. These limitations make CMOS-based adders less suitable for applications

where energy efficiency, reduced footprint, and prolonged battery life are essential, such as in wearable devices, IoT sensors, and edge AI platforms [1].

To overcome these limitations, various alternative design approaches have been explored, including hybrid logic styles that combine CMOS with Pass Transistor Logic (PTL) or other low-power techniques. While these designs can achieve moderate power savings, they often introduce issues such as signal degradation, voltage threshold drops, and increased design complexity. These challenges can negatively impact circuit reliability and scalability, especially when integrated into larger systems with stringent performance requirements [2].



**Fig 2: Schematic of Full Adder [2]**

Transmission Gate Logic (TGL) has emerged as a highly promising alternative due to its ability to reduce power consumption and transistor count while maintaining full voltage swing and high signal integrity. By leveraging complementary NMOS and PMOS transistors operating in parallel, TGL eliminates the voltage degradation commonly associated with pass transistor-based designs and reduces unnecessary switching activities that contribute to dynamic power loss. This makes TGL particularly effective for implementing energy-efficient arithmetic circuits without compromising computational speed or accuracy.

In this project, a 1-bit TGL-based full adder is designed and functionally verified to assess its suitability for low-power VLSI applications. The design is implemented at the transistor level using Cadence Virtuoso, followed by simulation-based functional verification in ModelSim using structured testbenches to validate correctness across all input conditions. Further, Synopsys tools are utilized to perform detailed power, area,

and timing analyses, providing a comprehensive evaluation of the proposed design's performance compared to conventional CMOS-based full adders.

By addressing the limitations of traditional adder designs, this research aims to contribute a viable low-power solution that meets the growing demands of modern computing systems, particularly in applications where energy efficiency, area optimization, and computational reliability are paramount.

### **3. OBJECTIVE**

The primary objective of this project is to design, implement, and analyze a 1-bit full adder based on Transmission Gate Logic (TGL) with the goal of minimizing power consumption, reducing transistor count, and improving area efficiency without compromising computational accuracy or speed. This objective is driven by the increasing demand for energy-efficient arithmetic circuits in battery-powered and performance-constrained applications such as portable electronics, embedded systems, AI accelerators, and edge computing devices [4]. By leveraging the complementary switching characteristics of NMOS and PMOS transistors configured as transmission gates, the proposed design seeks to achieve full voltage swing, superior signal integrity, and lower switching activity compared to conventional CMOS implementations [1]. The project involves creating a transistor-level schematic of the TGL-based full adder in Cadence Virtuoso, followed by functional verification using ModelSim to validate correct sum and carry outputs across all possible input conditions. To quantify the practical benefits of the design, power consumption, area utilization, and timing performance will be evaluated using Synopsys tools, enabling a comprehensive comparison with traditional CMOS-based full adders. Through this process, the project aims to demonstrate that TGL-based full adders offer significant advantages in low-power VLSI applications, positioning them as a suitable choice for modern digital systems where both energy efficiency and computational performance are critical [2][3].

### **4. BACKGROUND**

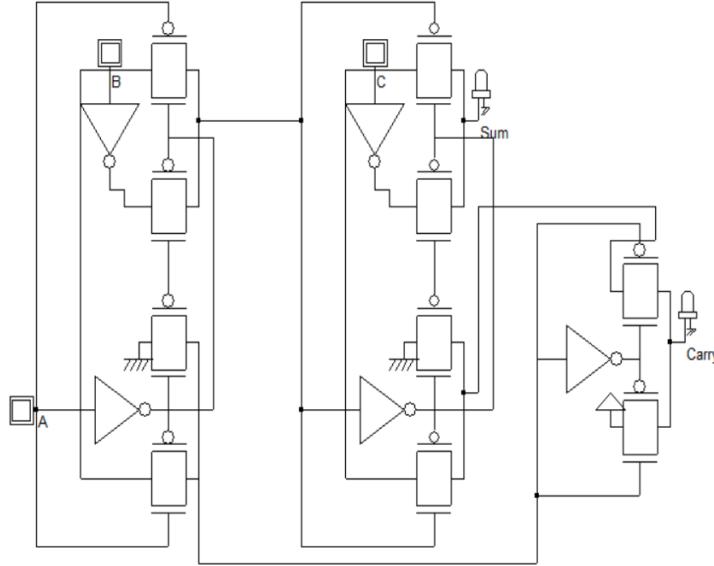
The demand for high-performance yet energy-efficient computing has intensified with the growing adoption of portable electronic devices, edge computing platforms, and AI-enabled systems. These systems require not only high computational power but also strict power management to extend battery life and reduce heat dissipation. Among the most critical building blocks in digital hardware are arithmetic circuits, particularly full adders, which perform essential binary addition operations in processors, digital signal processors (DSPs), and AI accelerators. Because full adders are used in large quantities within arithmetic logic units

(ALUs) and other computation-heavy units, their design significantly impacts the overall power, area, and speed of the system [5].

Traditional full adders are predominantly implemented using Complementary Metal-Oxide-Semiconductor (CMOS) technology, which provides robust and scalable solutions for various digital logic functions. CMOS circuits operate with low static power dissipation when idle, making them favorable for many digital applications [6]. However, the dynamic power consumed during switching activities, combined with leakage currents in deep submicron technologies, makes CMOS-based full adders less suitable for ultra-low-power applications. Furthermore, conventional CMOS full adder designs require a higher number of transistors—typically 28 or more—leading to increased silicon area and higher fabrication costs [4][5].

To address these challenges, researchers have explored alternative logic design styles such as Pass Transistor Logic (PTL) and hybrid CMOS-PTL implementations [3]. These designs aim to reduce transistor count and power consumption. However, they often suffer from reliability issues, including threshold voltage degradation, charge sharing, and signal integrity loss, particularly in deep submicron and high-frequency operations. These limitations hinder their widespread adoption in commercial low-power systems, where consistency and predictability of performance are crucial [6].

Transmission Gate Logic (TGL) has emerged as a promising alternative that addresses the drawbacks of both CMOS and PTL designs [5]. TGL leverages the parallel operation of NMOS and PMOS transistors to achieve full voltage swing, thereby eliminating the voltage drop issues commonly observed in pass transistor circuits. This ensures that the logic levels are maintained robustly across the entire circuit, improving signal integrity and switching reliability. Additionally, TGL circuits typically require fewer transistors than traditional CMOS implementations, leading to reduced area utilization and fabrication costs [1].



**Fig 3: Transmission gate base full adder logic [1]**

One of the primary advantages of TGL is its ability to minimize dynamic and static power consumption. By reducing unnecessary switching activities and optimizing the transistor-level structure, TGL-based circuits exhibit significantly lower energy dissipation [1]. This makes them particularly suitable for battery-operated devices, wearable electronics, and IoT applications, where energy efficiency is a primary concern. Moreover, TGL's ability to maintain high operational speed while reducing power makes it ideal for AI accelerators and embedded computing systems, which require both performance and efficiency [3].

In the context of VLSI design, the integration of TGL-based full adders can contribute to overall system-level energy savings, especially when scaled to larger bit-width operations such as 8-bit, 16-bit, or 32-bit arithmetic units. The reduction in power and area not only improves device longevity but also enables more compact and thermally stable system designs. This is critical for emerging technologies such as edge AI processors, where computational tasks must be performed locally on resource-constrained devices without relying on cloud computing infrastructure [2].

Truth table				
A	B	C	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

**Fig 4: Truth table TGL based Full Adder [2]**

This project focuses on the design, functional verification, and performance analysis of a 1-bit TGL-based full adder. The design process involves the creation of a transistor-level schematic in Cadence Virtuoso, followed by functional verification in ModelSim using structured testbenches to ensure logical correctness. Power, area, and timing characteristics are evaluated using Synopsys tools, enabling a comprehensive comparison with conventional CMOS-based full adders. The outcomes of this project aim to validate the suitability of TGL-based designs for modern low-power VLSI systems, positioning them as a key enabler for next-generation energy-efficient computing platforms [1][3].

## 5. METHODOLOGY

The design and analysis of the CMOS-based and Transmission Gate Logic (TGL)-based 1-bit full adders were carried out through a systematic methodology consisting of four key phases: functional verification, schematic design and simulation, synthesis and performance analysis, and layout design. Each phase was aimed at evaluating and comparing the efficiency of CMOS and TGL implementations in terms of functional correctness, power consumption, area utilization, and timing performance.

### 5.1. Functional Verification in ModelSim

The first phase involved verifying the logical correctness of both CMOS and TGL full adder designs using ModelSim simulation environment. Verilog-based behavioral models were created for both designs, and dedicated testbenches were developed to apply all possible input combinations. This ensured that the Sum and Carry outputs matched the expected truth table for a 1-bit full adder. Functional verification helped establish a reliable baseline by confirming that both implementations produced correct results under all operational scenarios.

## **5.2. Schematic Design and Simulation in Cadence Virtuoso**

Following functional verification, transistor-level schematic designs of both CMOS and TGL full adders were implemented in Cadence Virtuoso. The designs were simulated using transient analysis to validate their switching behavior and performance under realistic input conditions. During this phase, power consumption and propagation delay were calculated for both designs to evaluate their efficiency. This analysis provided initial insights into the dynamic performance of each design, highlighting the expected advantages of TGL in reducing power and improving speed.

## **5.3. Synthesis and Performance Analysis Using Synopsys Design Compiler**

The third phase involved synthesizing both CMOS and TGL full adder designs using Synopsys Design Compiler. RTL-level Verilog descriptions of both designs were synthesized using an appropriate technology library. After synthesis, detailed reports were generated to extract key performance metrics including area utilization (gate count), timing (propagation delay), and power consumption. This step provided a technology-mapped evaluation, allowing for a fair comparison of both designs in terms of silicon resource usage and timing performance at the synthesis level.

## **5.4. Layout Design in Cadence Virtuoso**

In the final phase, the layout design for the TGL-based full adder was carried out in Cadence Virtuoso. The layout was designed to meet Design Rule Check (DRC) and Layout Versus Schematic (LVS) requirements to ensure physical correctness and electrical consistency with the schematic. This phase provided a realistic visualization of the silicon area required by the TGL design and validated its manufacturability. Although the CMOS layout was not developed in this project, the TGL layout results further strengthened the assessment of the design's area efficiency and practical implementation feasibility.

## 6. RESULT AND VERIFICATION

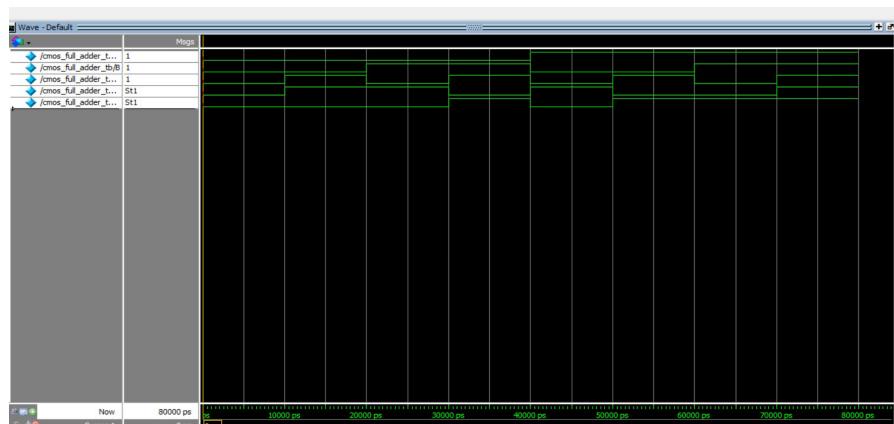
### a) Functional Verification in ModelSim

#### Step 1: Preparing Verilog Source Files

- The CMOS full adder functionality was modeled in Verilog HDL with defined input ports A, B, Cin and output ports Sum, Cout.
- A testbench module was created to apply all possible input combinations and observe the corresponding outputs.
- The testbench generated eight stimulus sets, covering all truth table entries for a 1-bit full adder.
- The console output displayed the truth table verification with simulation time points, showing all combinations of A, B, Cin and their corresponding Sum and Cout outputs.

```
VSIM 3> run -all
# Time  A B Cin | Sum Cout
# -----
# 0ns  0 0 0 | 0 0
# 10ns 0 0 1 | 1 0
# 20ns 0 1 0 | 1 0
# 30ns 0 1 1 | 0 1
# 40ns 1 0 0 | 1 0
# 50ns 1 0 1 | 0 1
# 60ns 1 1 0 | 0 1
# 70ns 1 1 1 | 1 1
# ** Note: $finish  : C:/intelFPGA/18.1/cmos_testbench.v(38)
#   Time: 80 ns  Iteration: 0  Instance: /cmos_full_adder_tb
# 1
# Break in Module cmos_full_adder_tb at C:/intelFPGA/18.1/cmos_testbench.v line 38
VSIM 4>]
```

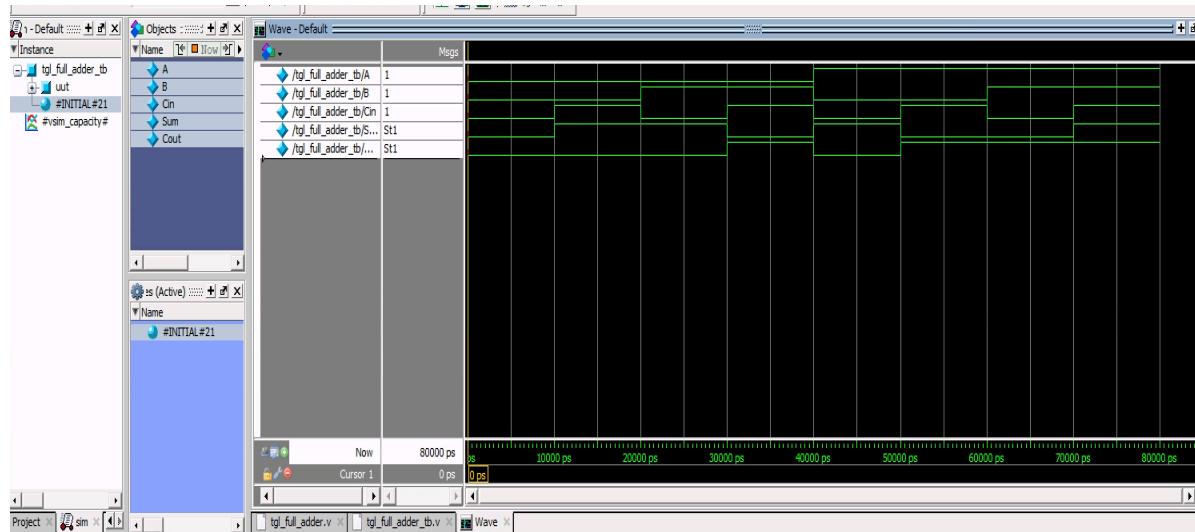
- The waveform viewer was used to cross-verify the output transitions, confirming that Sum and Cout matched expected values for all input states.



CMOS Full Adder Waveform

Similarly, for **TGL based Full Adder** the compilation and stimulation are done and generated

```
ModelSim> vsim -gui TGLADDER.tgl_full_adder_tb
# vsim -gui TGLADDER.tgl_full_adder_tb
# Start time: 01:19:57 on May 08,2025
# Loading TGLADDER.tgl_full_adder_tb
# Loading TGLADDER.tgl_full_adder
add wave -position insertpoint sim:/tgl_full_adder_tb/*
VSIM 3> run -all
# Time  A B Cin | Sum Cout
# 0      0 0 0   | 0   0
# 10     0 0 1   | 1   0
# 20     0 1 0   | 1   0
# 30     0 1 1   | 0   1
# 40     1 0 0   | 1   0
# 50     1 0 1   | 0   1
# 60     1 1 0   | 0   1
# 70     1 1 1   | 1   1
# ** Note: $finish    : //apporto.com/dfs/CSUFRESNO/Users/c21872_csufresno/Desktop/TGLADDER/tgl_full_adder_tb.v(35)
#   Time: 80 ns  Iteration: 0  Instance: /tgl_full_adder_tb
# 1
```



TGL full adder waveform

## b) Schematic Design and Simulation in Cadence Virtuoso

### Step 1: Launching Cadence Virtuoso

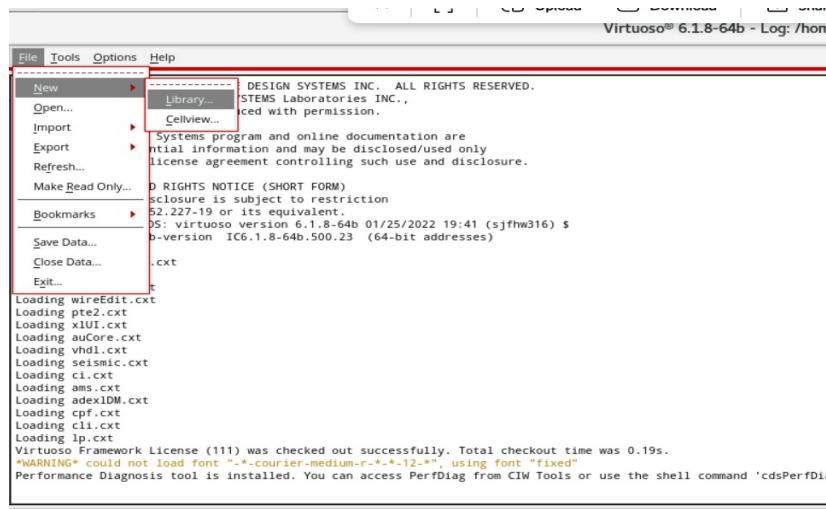
The Virtuoso environment was launched from the terminal using the command: `virtuoso &`

- This command opened the Cadence Command Interpreter Window (CIW), confirming that all necessary licenses and technology libraries were successfully loaded

## Step 2 : Creating a New Library

From the CIW, the following steps were taken to create a new library:

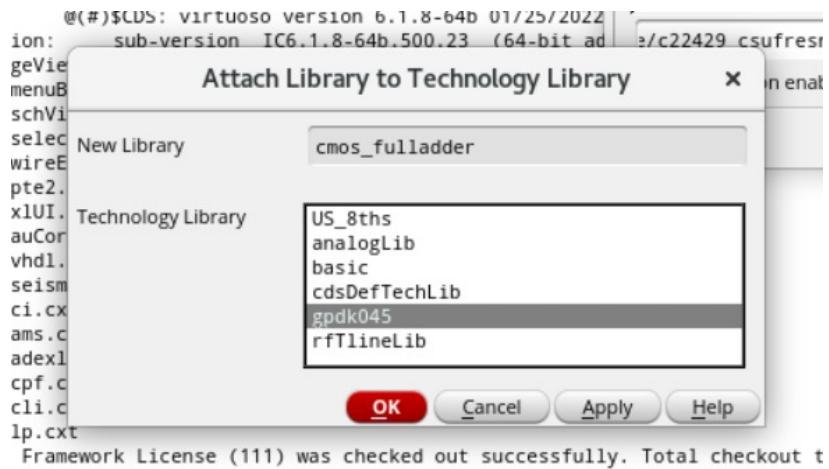
- **File → New → Library** was selected to open the "New Library" window.



- The library was named according to the design, for example, `cmos_fulladder`.



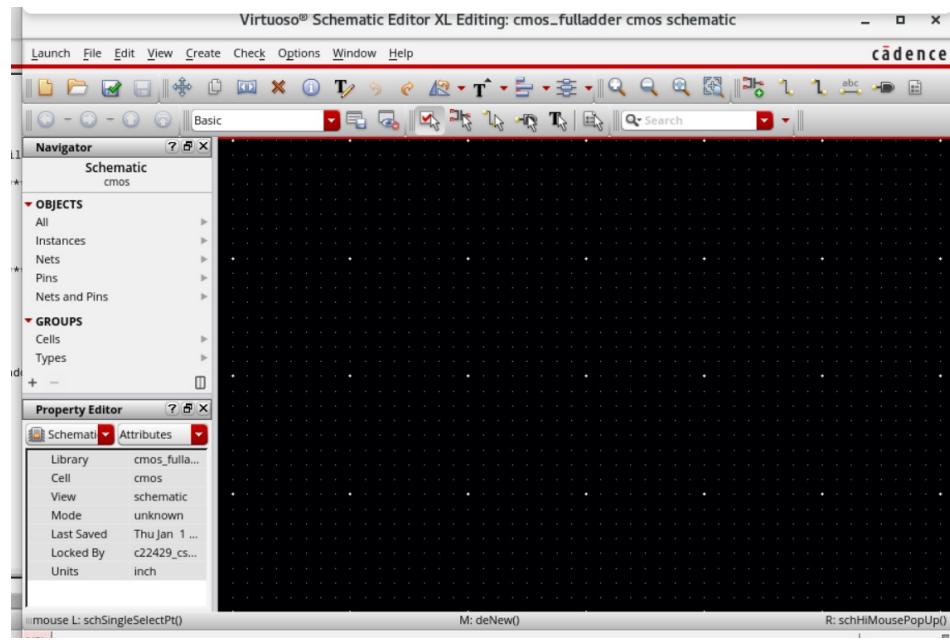
- The option Attach to an existing technology library was chosen.
- The **gdk045** technology library (representing 45nm process technology) was selected for this project.



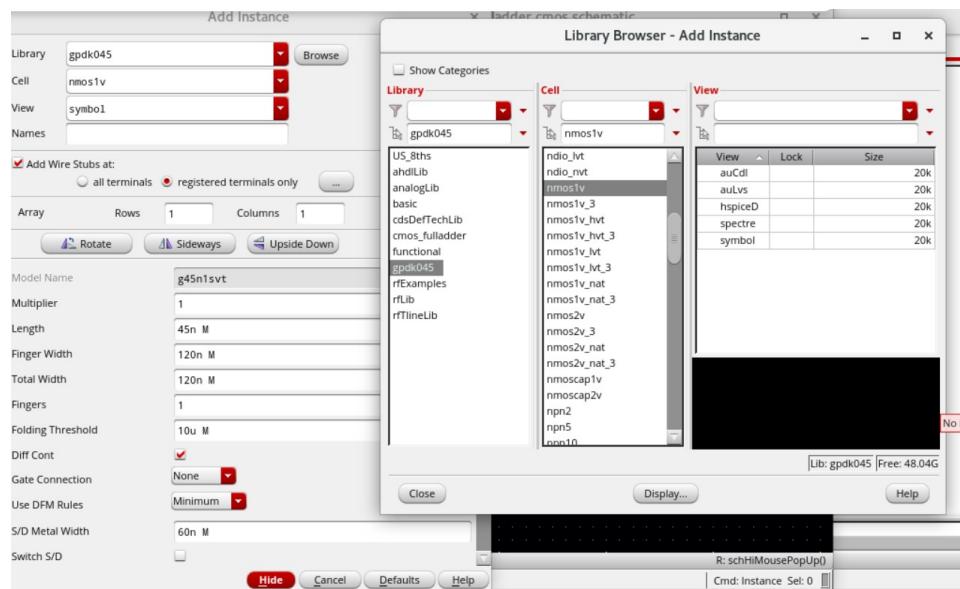
### Step 3: Placing Transistor Instances

With the schematic editor open:

- The **Library Browser** was used to select the **gdk045** library.



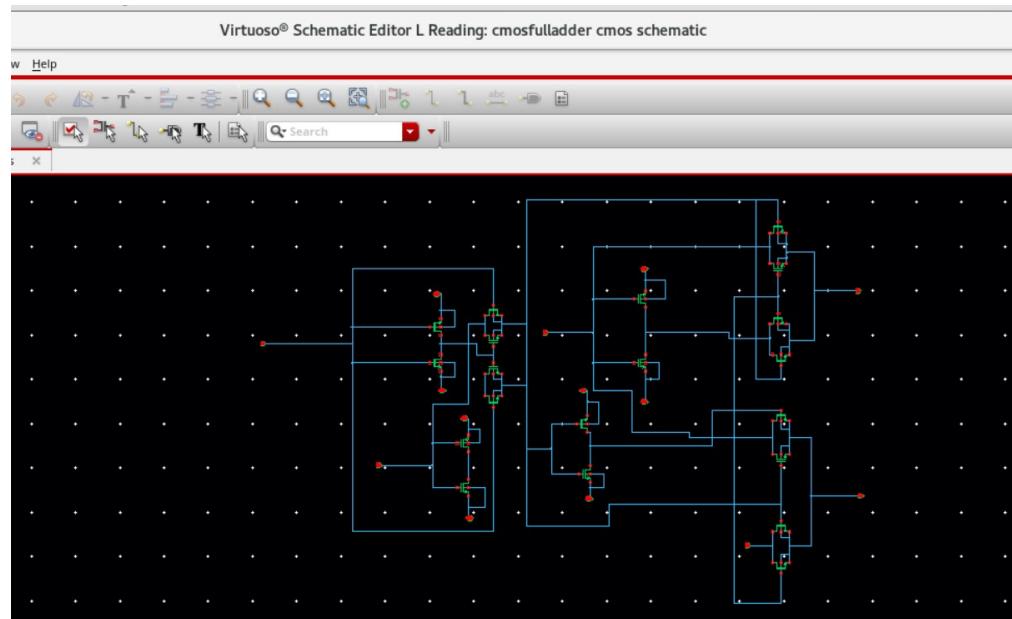
- NMOS and PMOS transistors were placed by choosing `nmos1v` and `pmos1v` cells respectively from **gdk045**.
- The device parameters like **Length**, **Width**, and **Fingers** were configured as per the project specifications.



## Step 4: Completing Schematic Design

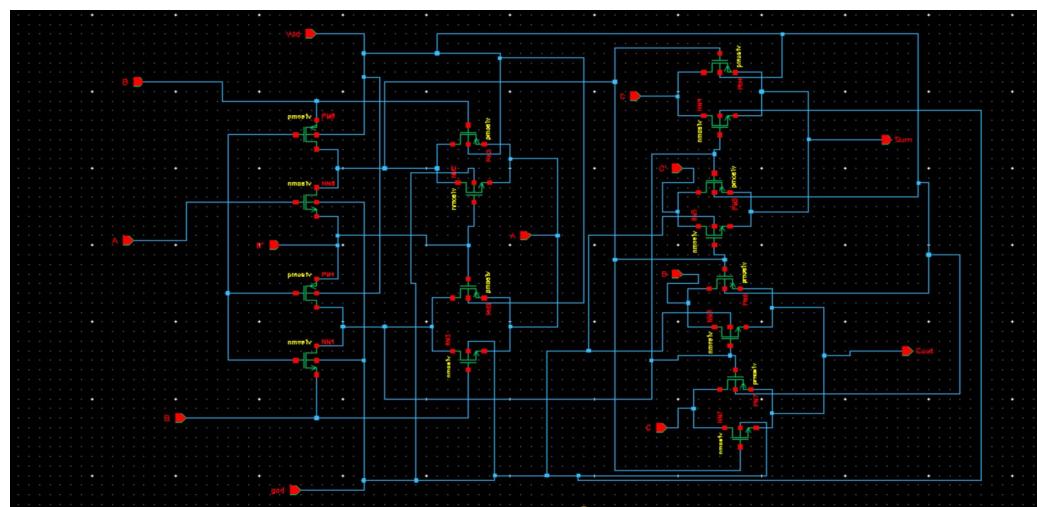
The CMOS-based full adder circuit was constructed by:

- Connecting the transistors based on the required logic equations for Sum and Carry.
- Properly naming all the **input** and **output** pins to define the circuit's interface.
- Checking the **Design Rules** using the **Check and Save** tool to ensure the schematic was error-free.



CMOS FULL ADDER SCHEMATIC DESIGN

Similar steps were followed for Designing TGL Full Adder

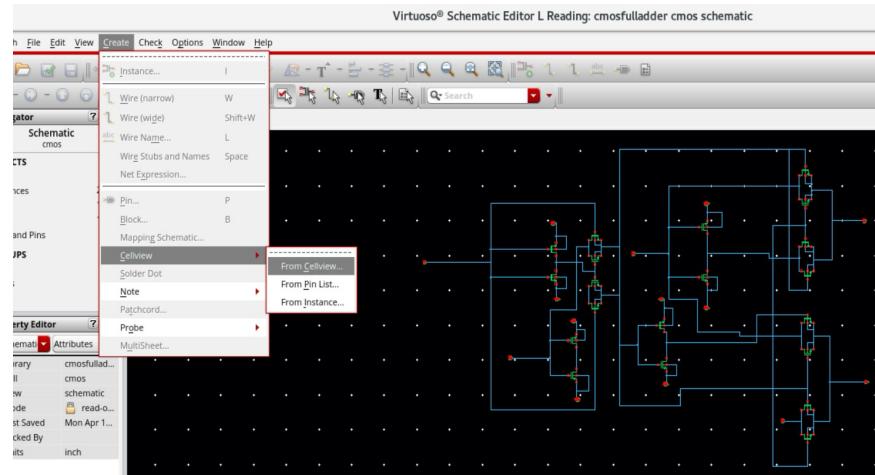


TGL BASED FULL ADDER SCHEMATIC DESIGN

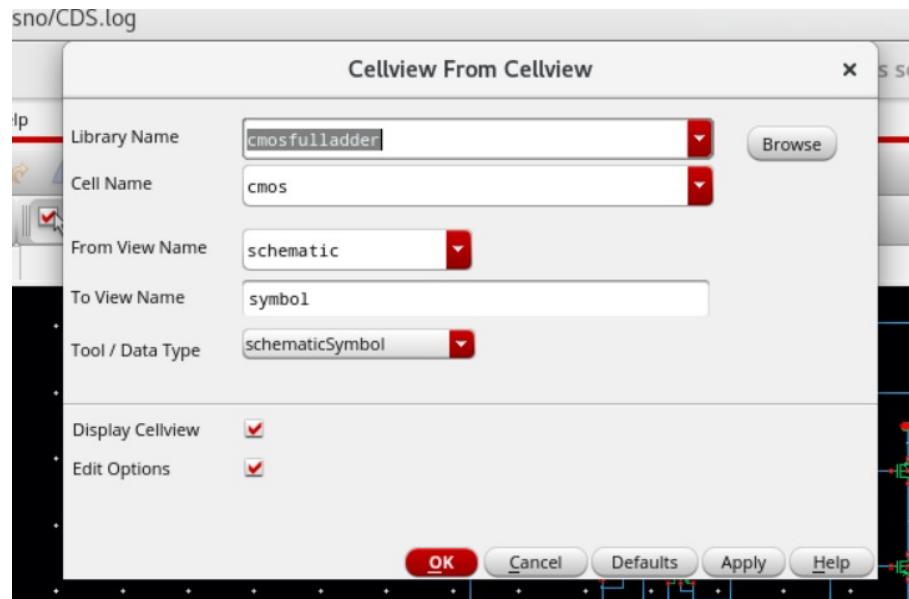
## Step 5: Creating a New Cell view

After creating the library, the next step was to define a **new cell view**:

- **File → New → Cell view** was selected.
- The library was set as `cmos_fulladder`, the cell name as `cmos`, and the view name as `schematic`.
- **Schematics XL** was selected as the application for schematic design.

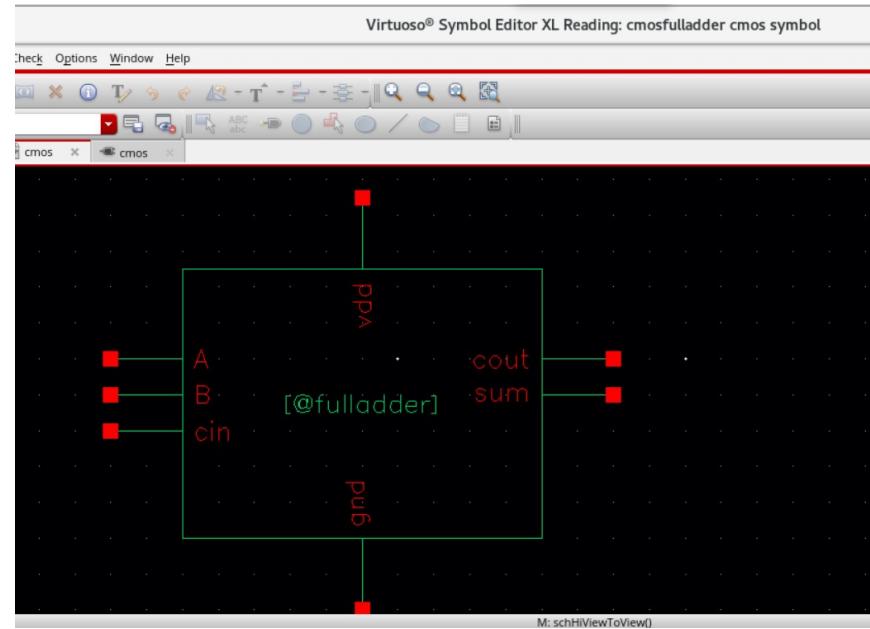


Browse the Library name and confirm ok

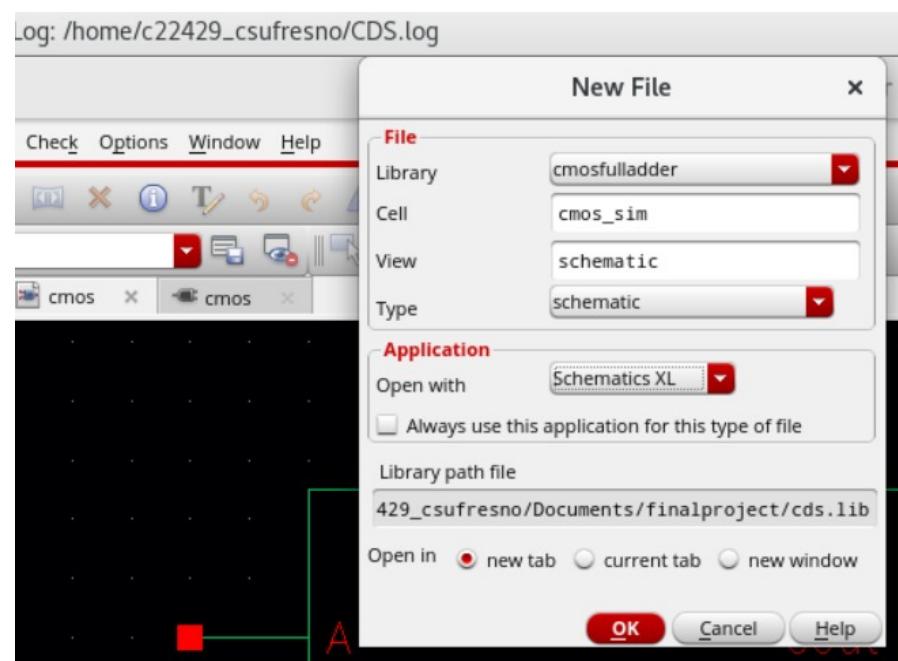


After designing the schematic of the CMOS full adder, a **symbol view** was created to facilitate hierarchical simulation. The symbol consisted of labeled pins for inputs **A**, **B**, and **Cin**, outputs **Sum** and **Cout**, and

power connections **VDD** and **GND**. This symbol representation allowed the full adder to be instantiated easily in larger testbench schematics.

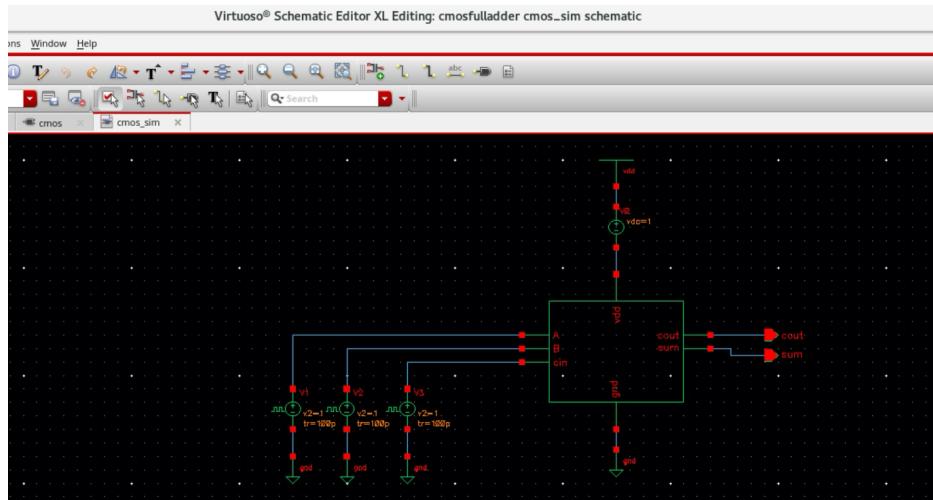


Create new file in new tab and given cell name as cmos\_sim and Open with Schematic XL

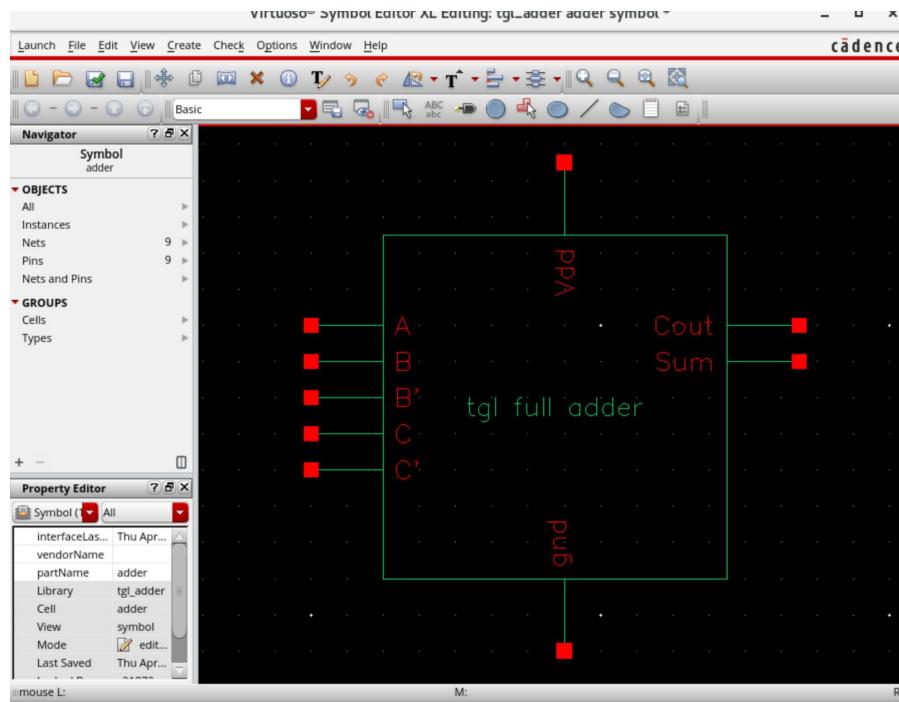


The **vpulse sources** were configured to generate square wave pulses with appropriate rise time, fall time, and pulse width to simulate all input combinations. These pulse sources were connected to **A**, **B**, and **Cin** inputs of the full adder to systematically apply different logic states.

**Output pins for Sum** and **Cout** were connected to simulation markers or output probes to monitor their behavior.

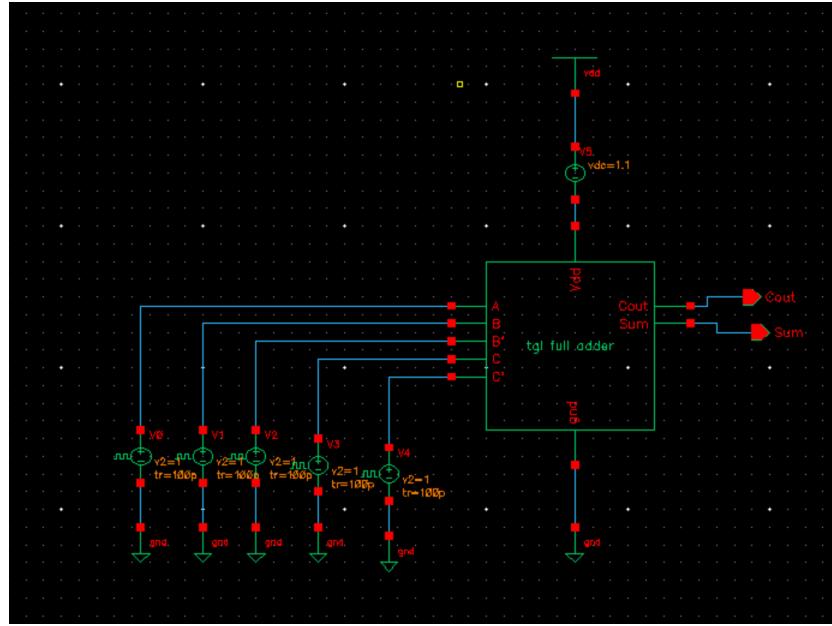


Similarly, steps followed for TGL FULL adder



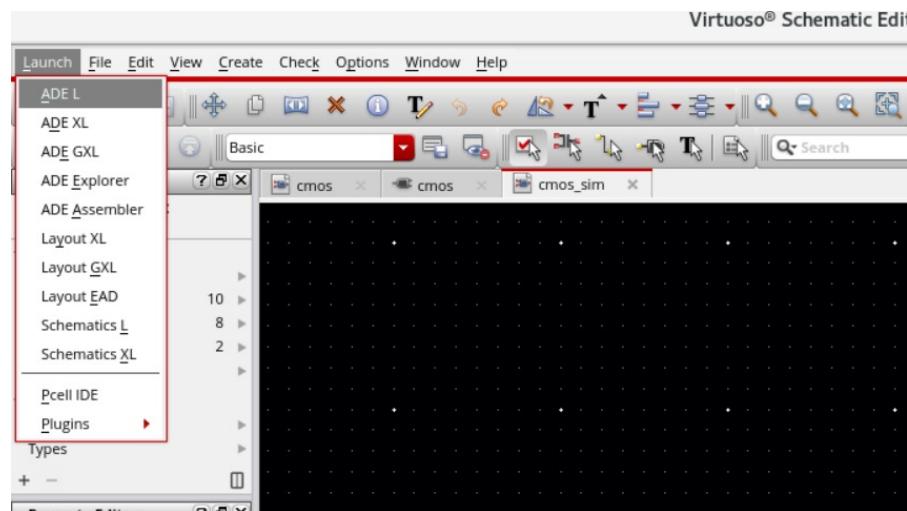
Cell view of tgl full adder

## Symbol view of TGL based full adder in Cadence Virtuoso

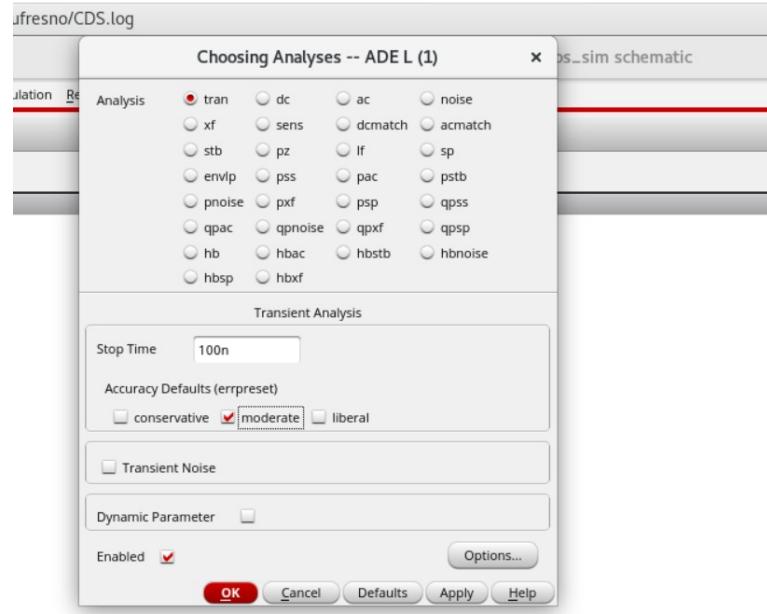


### Step 6: Launching ADE L Environment

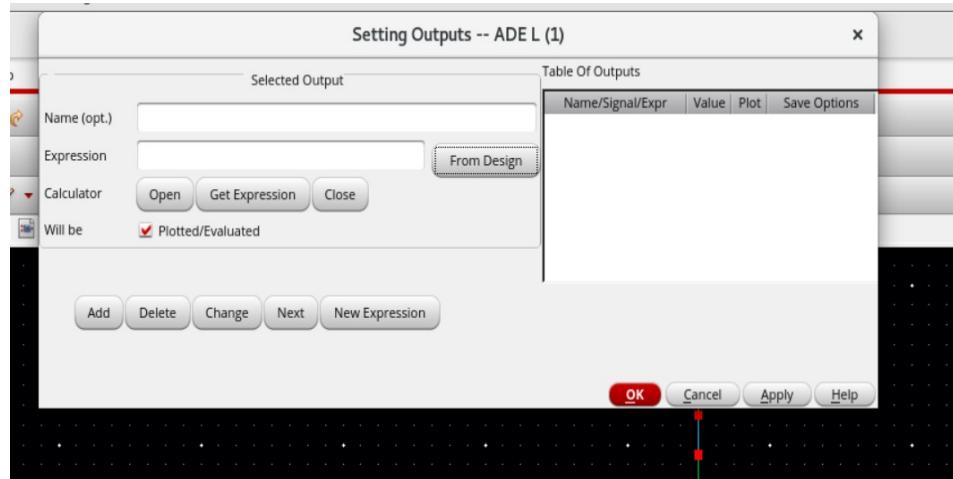
- From the schematic window, **Launch → ADE L** was selected to open the **Analog Design Environment**.
- This environment provided the interface to configure simulation parameters, analyses, and outputs.



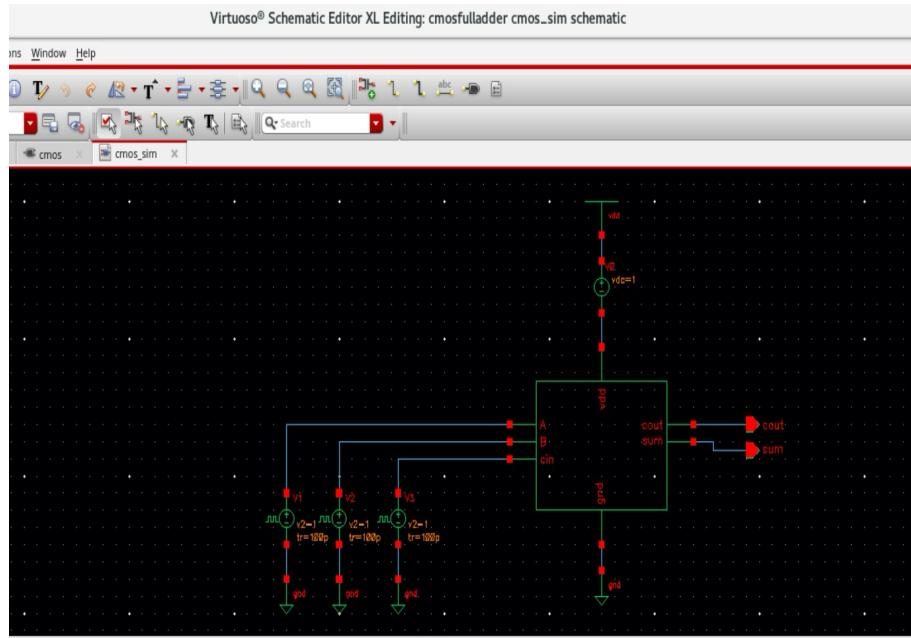
- Transient Analysis (tran)** was selected as the simulation type.
- The **stop time** was specified (e.g., **100 ns**) to ensure sufficient simulation coverage for multiple input switching events.



- The **accuracy** was set to **moderate** to balance between simulation speed and result precision.
- Outputs → **To Be Plotted** → **Select on Schematic** was used to select the output nodes **Sum** and **Cout** by clicking on them in the schematic.



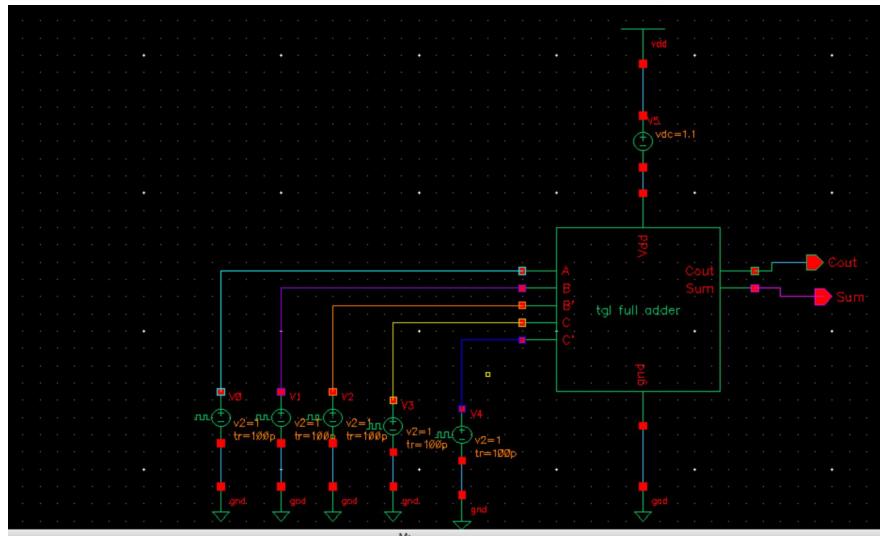
This action added these nodes to the **Table of Outputs** in ADE L, enabling waveform observation after simulation.



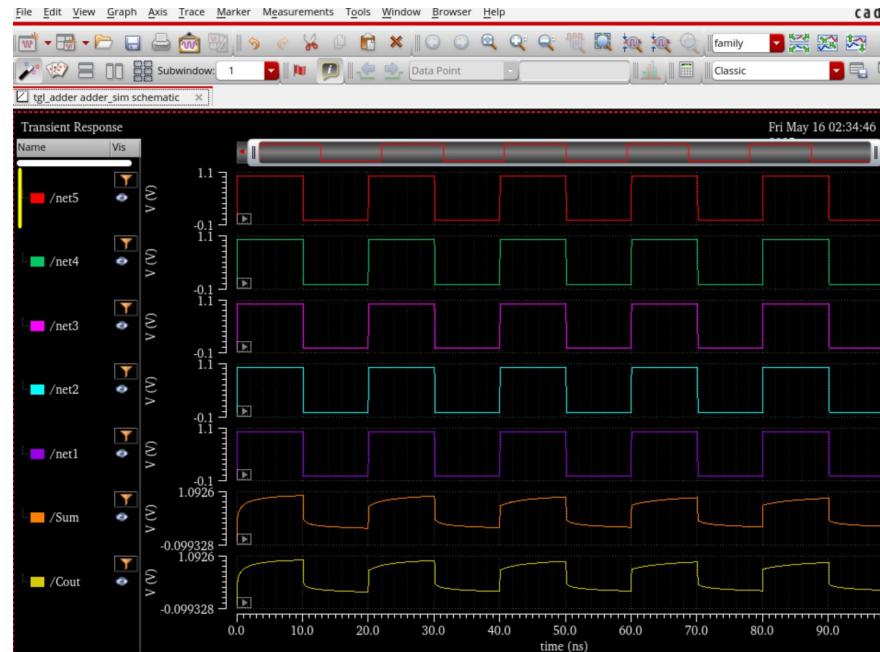
- The waveforms for **input signals (A, B, Cin)**, **Sum**, and **Cout** were plotted against simulation time.



Similar steps followed for TGL Full Adder,



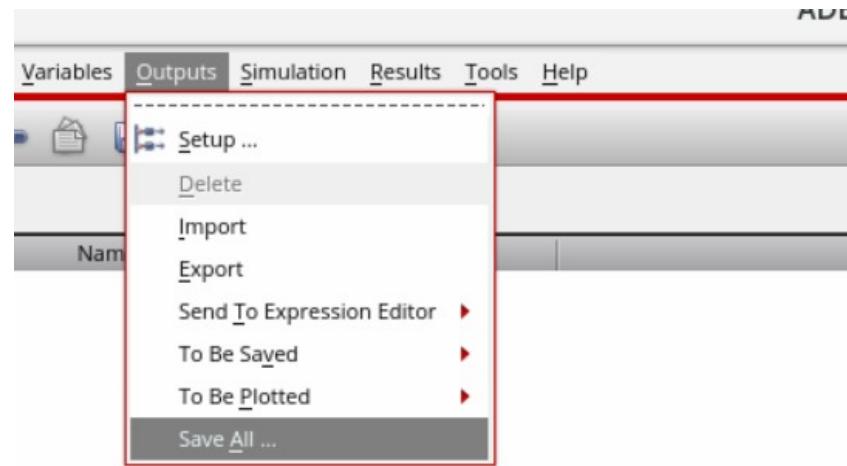
After Selecting outputs from design save them and click run in ADE L, it automatically generates waveforms if there are no errors



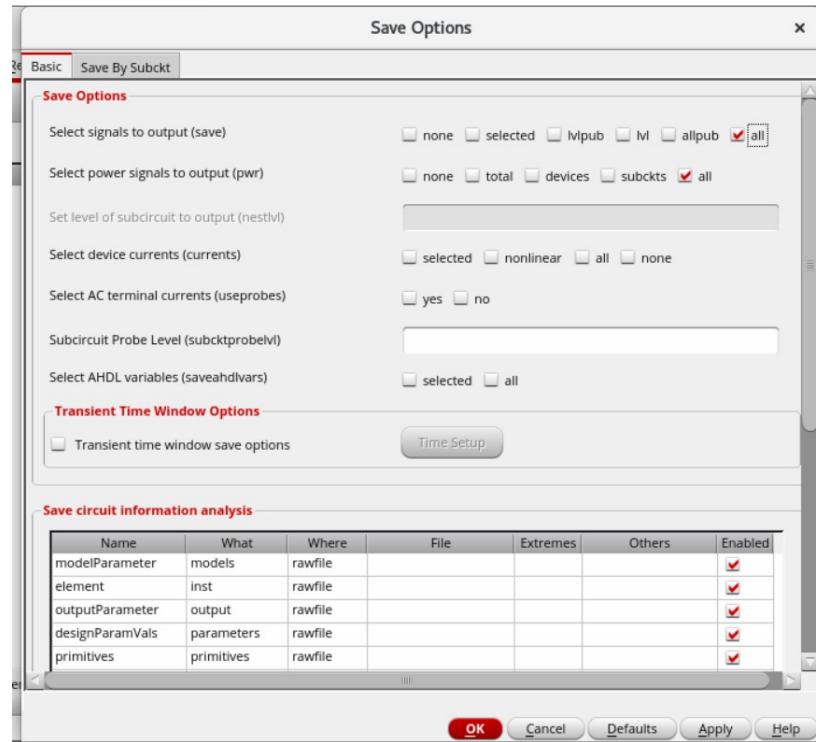
TGL based full adder Waveform

### Step 7: Configuring Transient Analysis

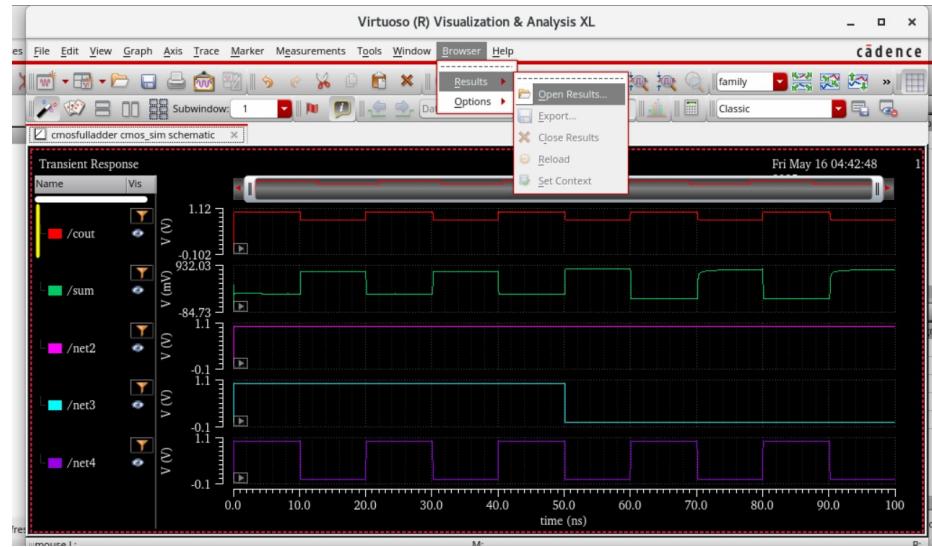
- In ADE L, **Setup → Analyses** was selected to open the **Choosing Analyses** window.
- **Enabled** was checked to activate the analysis.
- The configuration was confirmed by clicking **OK**.



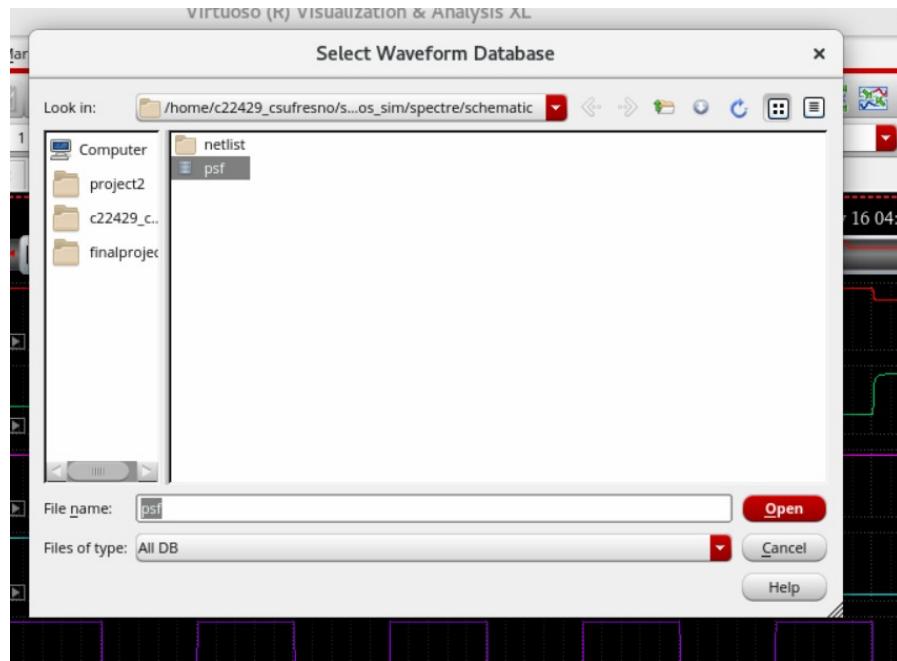
- After selecting save all select all to view the power and click ok



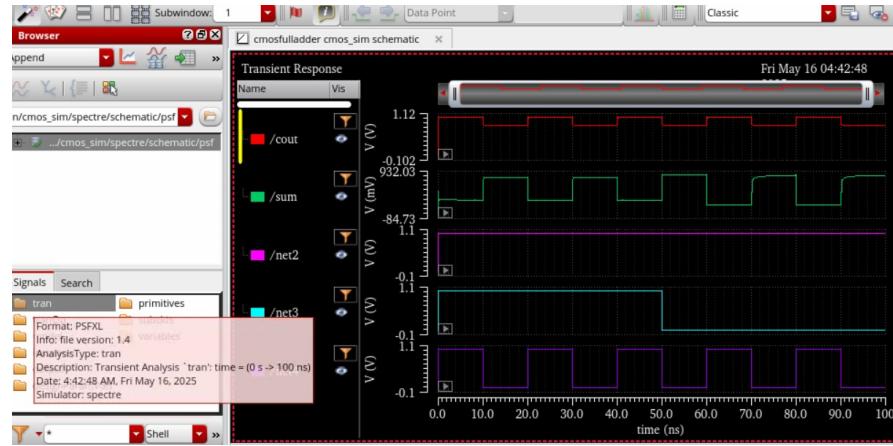
Now open simulation window and select Browser -> Open results



After opening the results select psf and open it



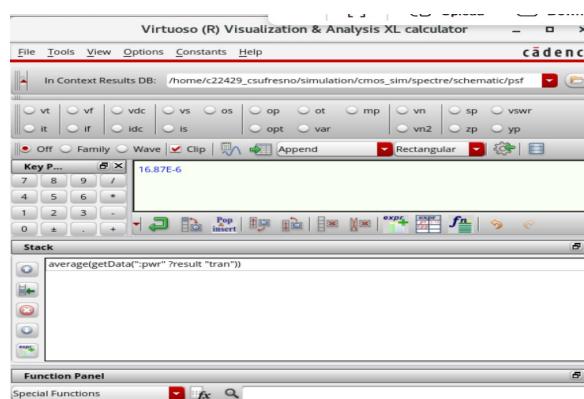
- Virtuoso Visualization & Analysis XL **automatically opened, displaying the Transient Response Waveforms.**



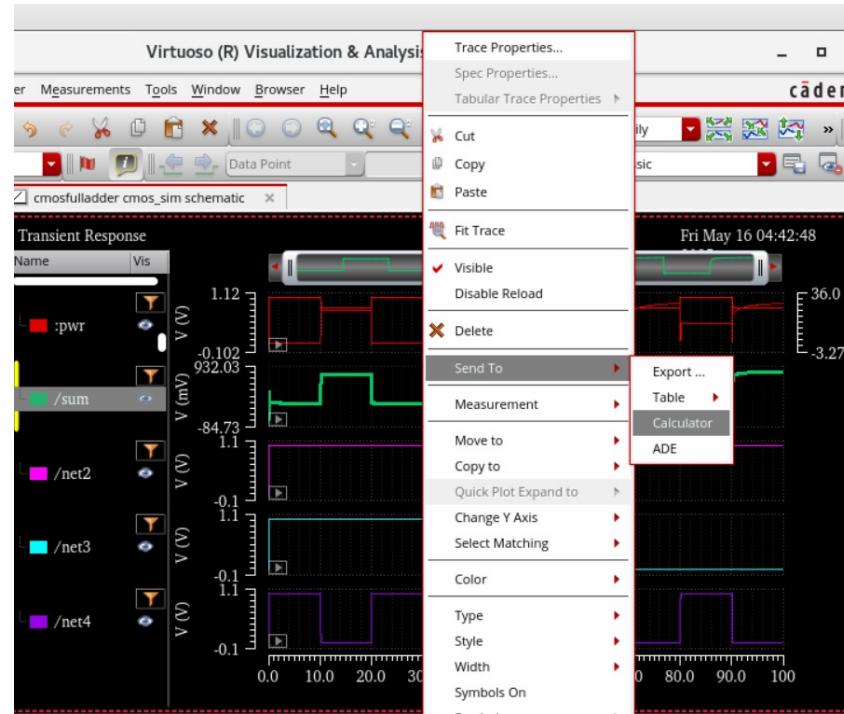
Click on the pwr to view the power waveform and select on the line of power and deviate to the calculator



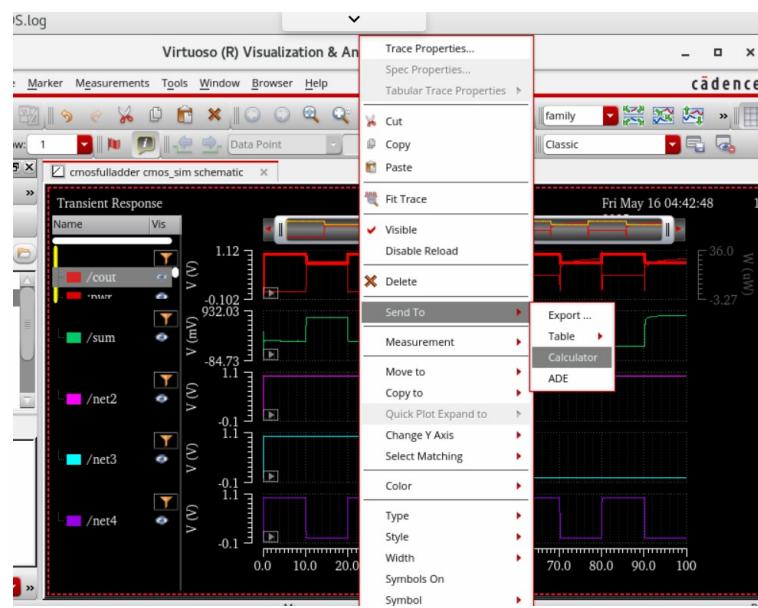
- The **power signal** (labeled as: pwr) was selected in the waveform browser.
- Right-click → Send To → Calculator was used to send the signal to the **Virtuoso Calculator**.



For calculating the propagation delay, we have taken delay of sum from there we have selected random point on the wave form on sum and deviated to calculator



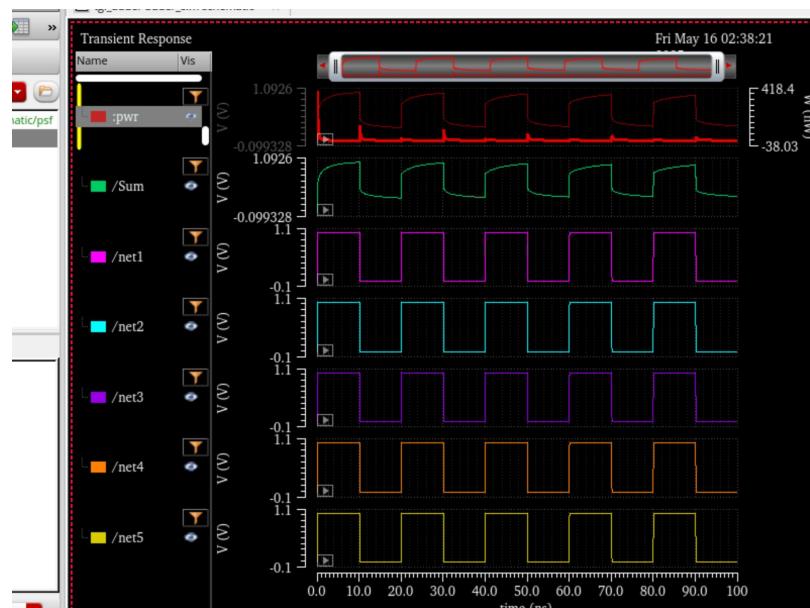
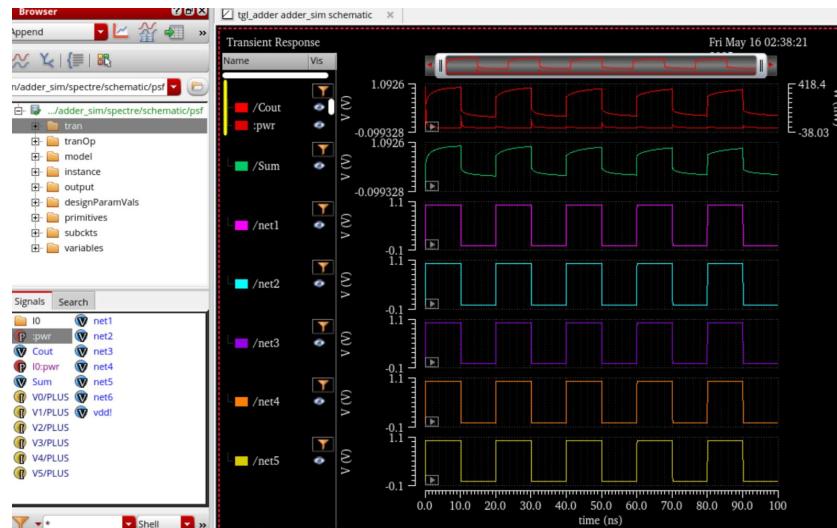
In the Simulation window, Click on Cout waveform line Right click -> Send to -> Calculator

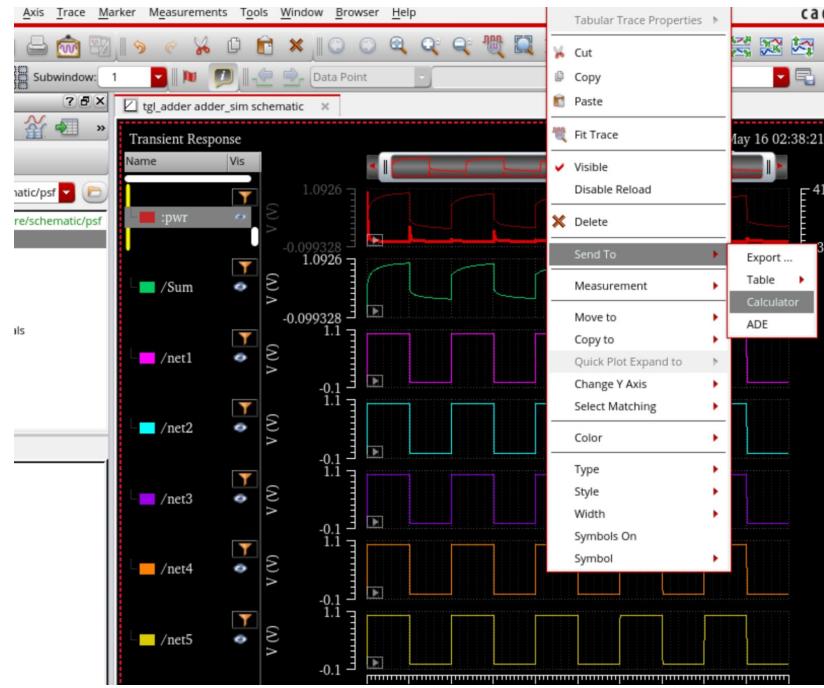


A measurement function such as **delay()** was used in the Calculator to calculate the propagation delay.

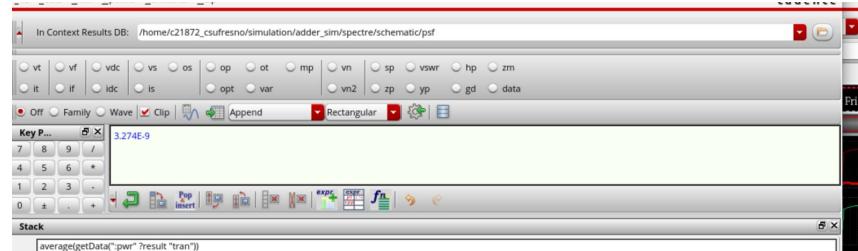
```
delay(?wf1 VT("/net2"),?value1 0.5,?edge1 "rising",?nth1 1,?td1 0.0,?tol1 nil,?wf2 VT("/Cout"),?value2 0.5,?edge2 "falling",?nth2 1,?tol2 nil,?td2 nil,?stop nil,?multiple nil)
VT("/Cout")
vt"/Cout" ?result "tran"
10.14E-9
delay(?wf1 VT("/net1"),?value1 0.5,?edge1 "rising",?nth1 1,?td1 0.0,?tol1 nil,?wf2 VT("/Sum"),?value2 0.5,?edge2 "falling",?nth2 1,?tol2 nil,?td2 nil,?stop nil,?multiple nil)
VT("/Sum")
```

Similarly, for TGL based full adder





Calculation of Power of TGL Full Adder:



Propagation Delay of TGL full adder

```
9.923E-9
delay(?wf1 VT("/sum"), ?value1 0.5, ?edge1 "rising", ?nth1 1, ?td1 0.0, ?tol1 nil, ?wf2 v("/sum" ?result "tran"), ?value2 0.5, ?edge2 "falling", ?nth2 1, ?tol2 nil, ?td2 nil, ?stop nil, ?multiple nil)
```

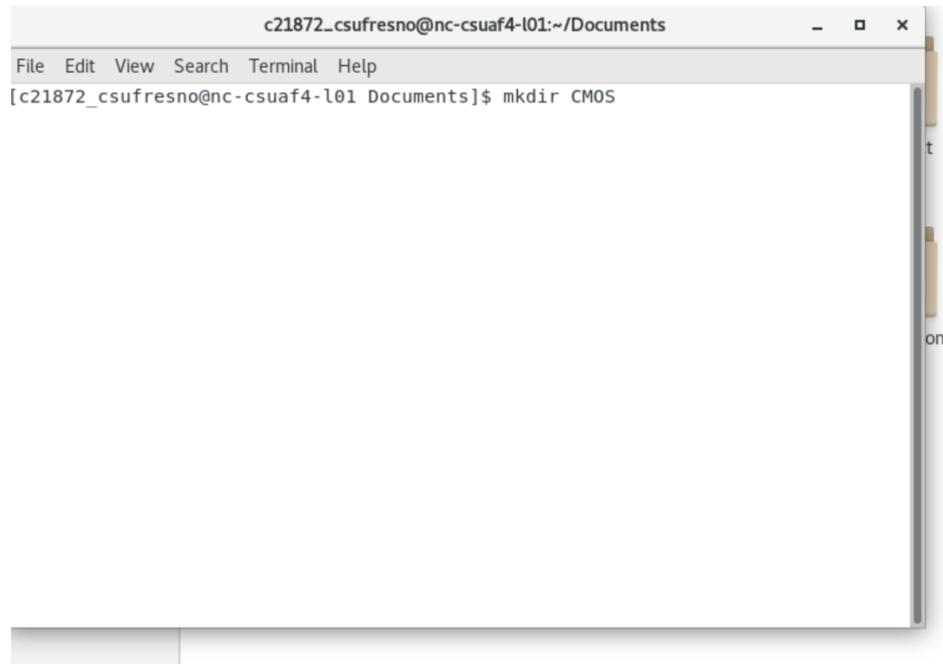
## C) Synthesis and Performance Analysis Using Synopsys Design Compiler

After completing the schematic-level simulations and functional verification in ModelSim, the CMOS and TGL full adder designs were synthesized using Synopsys Design Compiler (DC) to evaluate their area, timing, and power characteristics under realistic fabrication constraints.

### Step 1: RTL Source Preparation

- The transistor level Verilog descriptions of both the CMOS-based and TGL-based full adders were prepared for synthesis.

- Both designs were validated to ensure there were no syntax or semantic errors.



c21872\_csufresno@nc-csuaf4-l01:~/Documents

```
File Edit View Search Terminal Help
[c21872_csufresno@nc-csuaf4-l01 Documents]$ mkdir CMOS
```

The terminal window shows the command `mkdir CMOS` being run in the `Documents` directory. The output shows the directory `CMOS` has been created.

The link library and target library were defined in the Design Compiler setup scripts to enable mapping to standard cells.



c21872\_csufresno@nc-csuaf4-l01:~/Documents/CMOS

```
File Edit View Search Terminal Help
[c21872_csufresno@nc-csuaf4-l01 CMOS]$ cp -r DCFilesAndExamples/* CMOS/
```

The terminal window shows the command `cp -r DCFilesAndExamples/* CMOS/` being run in the `CMOS` directory. The output shows the files from the `DCFilesAndExamples` directory have been copied into the `CMOS` directory.

For synthesis give command: make NAME= top module name synth

After synthesis, **area**, **timing**, and **power reports** were generated

```
*****
Report : area
Design : cmos_full_adder
Version: 5-2021.06-SP5-1
Date   : Thu May 15 18:21:42 2025
*****  

Library(s) Used:  

| NangateOpenCellLibrary (File: /synopsys/Nangate_FreePDK45/NangateOpenCellLibrary_PDKv1_3_v2010_12/Front_End/Liberty/CCS/NangateOpenCellLibrary.db)  

| Number of ports:          5  

| Number of nets:           7  

| Number of cells:          4  

| Number of combinational cells: 4  

| Number of sequential cells: 0  

| Number of macros/black boxes: 0  

| Number of buf/inv:         1  

| Number of references:      3  

|  

| Combinational area:       5.054000  

| Buf/Inv area:              0.532000  

| Noncombinational area:     0.000000  

| Macro/Black Box area:      0.000000  

| Net Interconnect area:     undefined (Wire load has zero net area)  

|  

| Total cell area:          5.054000  

| Total area:                undefined  

| 1
```

### Area Report of CMOS full Adder

```
*****
Report : timing
  -path full
  -delay max
  -nworst 10
  -max_paths 10
Design : cmos_full_adder
Version: S-2021.06-SP5-1
Date   : Thu May 15 18:21:43 2025
*****  

Operating Conditions: fast Library: NangateOpenCellLibrary
Wire Load Model Mode: top  

Startpoint: B (input port)
Endpoint: Cout (output port)
Path Group: (none)
Path Type: max  

Des/Clust/Port      Wire Load Model      Library
-----  

cmos_full_adder    5K_hvratio_1_1      NangateOpenCellLibrary  

Point              Incr      Path
-----  

input external delay      0.00      0.00 r
B (in)                  0.00      0.00 r
U8/Z (XOR2_X1)          0.05      0.05 r
U7/ZN (AOI22_X1)        0.03      0.03 f
U6/ZN (INV_X1)          0.02      0.12 r
Cout (out)               0.01      0.13 r
data arrival time        0.13  

-----  

(Path is unconstrained)
```

## Timing Report of CMOS Full Adder

```
| Loading db file: '/synopsys/Nangate_FreePDK45/NangateOpenCellLibrary_PDKv1_3_v2010_12/Front_End/Liberty/CCS/NangateOpenCellLibrary.db'
Information: Propagating switching activity (low effort zero delay simulation). (PMR-6)
Warning: There is no defined clock in the design. (PMR-80)
Warning: Design has unannotated primary inputs. (PMR-414)
*****  

Report : power
  -analysis_effort low
Design : cmos_full_adder
Version: S-2021.06-SP5-1
Date   : Thu May 15 18:21:43 2025
*****  

Library(s) Used:
  NangateOpenCellLibrary (File: /synopsys/Nangate_FreePDK45/NangateOpenCellLibrary_PDKv1_3_v2010_12/Front_End/Liberty/CCS/NangateOpenCellLibrary.db)  

Operating Conditions: fast Library: NangateOpenCellLibrary
Wire Load Model Mode: top  

Design      Wire Load Model      Library
-----  

cmos_full_adder  5K_hvratio_1_1  NangateOpenCellLibrary  

Global Operating Voltage = 1.25
Power-specific unit information :
  Voltage Units = 1V
  Current Units = 1.000000fF
  Time Units = 1ns
  Dynamic Power Units = 1uW (derived from V,C,T units)
  Leakage Power Units = 1nW  

Cell Internal Power = 2.6172 uW (60%)
Net Switching Power = 1.7448 uW (48%)
Total Dynamic Power = 4.3621 uW (100%)
Cell Leakage Power = 396.1610 nW  

Information: report_power power group summary does not include estimated clock tree power. (PMR-789)
Power Group      Internal Power      Switching Power      Leakage Power      Total Power ( % ) Attrs
-----  

io_pad          0.0000             0.0000             0.0000             0.0000 ( 0.00% )
memory          0.0000             0.0000             0.0000             0.0000 ( 0.00% )
block_box        0.0000             0.0000             0.0000             0.0000 ( 0.00% )
clock_network   0.0000             0.0000             0.0000             0.0000 ( 0.00% )
register         0.0000             0.0000             0.0000             0.0000 ( 0.00% )
sequential       0.0000             0.0000             0.0000             0.0000 ( 0.00% )
combinational   2.6172            1.7448            396.1610           4.7582 ( 100.00% )
-----  

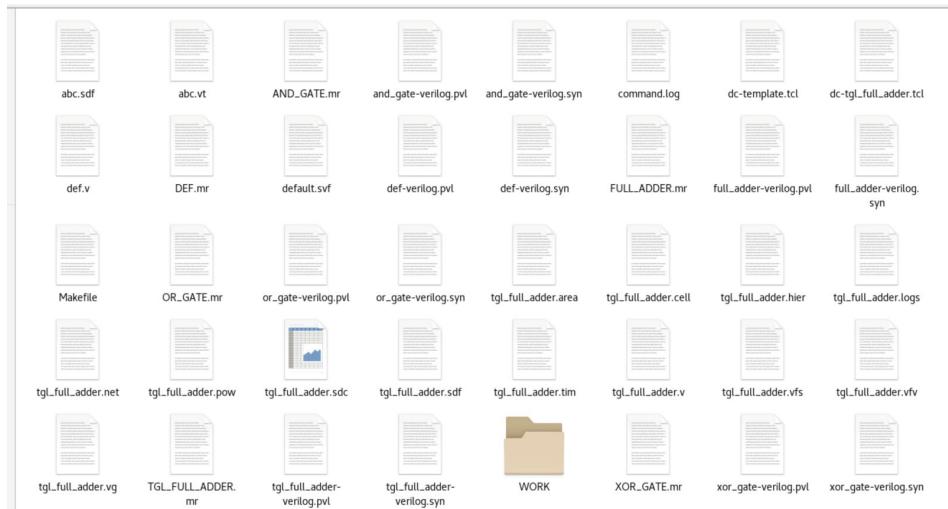
Total           2.6172 uW          1.7448 uW          396.1610 nW          4.7582 uW  

1
```

## Power Report of CMOS Full Adder

Similarly, for TGL full adder

After synthesis Of TGL Full Adder, all files are generated



```
Open Save x
tgl_full_adder.area
~/Documents/240Project
*****
Report : area
Design : tgl_full_adder
Version: S-2021.06-SP5-1
Date   : Thu May 15 19:51:18 2025
*****
Library(s) Used:
    NangateOpenCellLibrary (File: /synopsys/Nangate_FreePDK45/
    NangateOpenCellLibrary_PDKv1_3_v2010_12/Front_End/Liberty/CCS/NangateOpenCellLibrary.db)

Number of ports:          5
Number of nets:           7
Number of cells:          4
Number of combinational cells: 4
Number of sequential cells: 0
Number of macros/black boxes: 0
Number of buf/inv:         1
Number of references:     3

Combinational area:      5.054000
Buf/Inv area:            0.532000
Noncombinational area:   0.000000
Macro/Black Box area:    0.000000
Net Interconnect area:   undefined (Wire load has zero net area)

Total cell area:          5.054000
Total area:                undefined
1

Plain Text Tab Width: 8 Ln 1, Col 1 INS
```

Area report of TGL full adder

Open Save 
  
**tgl\_full\_adder.pow**  
 ~/Documents/240Project

```

Global Operating Voltage = 1.25
Power-specific unit information :
  Voltage Units = 1V
  Capacitance Units = 1.000000ff
  Time Units = 1ns
  Dynamic Power Units = 1uW      (derived from V,C,T units)
  Leakage Power Units = 1nW

  Cell Internal Power = 2.6172 uW (60%)
  Net Switching Power = 1.7448 uW (40%)
  -----
  Total Dynamic Power = 4.3621 uW (100%)
  Cell Leakage Power = 396.1610 nW

Information: report_power power group summary does not include estimated clock tree power.
(PWR-789)

      Internal          Switching          Leakage          Total
Power Group     Power            Power           Power        Power   ( % )  Attrs
-----
io_pad          0.0000          0.0000          0.0000      0.0000 ( 0.00%)
memory         0.0000          0.0000          0.0000      0.0000 ( 0.00%)
black_box       0.0000          0.0000          0.0000      0.0000 ( 0.00%)
clock_network  0.0000          0.0000          0.0000      0.0000 ( 0.00%)
register        0.0000          0.0000          0.0000      0.0000 ( 0.00%)
sequential      0.0000          0.0000          0.0000      0.0000 ( 0.00%)
combinational  2.6172          1.7448          396.1610    4.7582 ( 100.00%)
-----
Total          2.6172 uW      1.7448 uW      396.1610 nW    4.7582 uW
1
  
```

Plain Text ▾ Tab Width: 8 ▾ Ln 56, Col 64 ▾ INS

### Power report of TGL Full Adder

Open Save 
  
**tgl\_full\_adder.tim**  
 ~/Documents/240Project

```

input external delay          0.00      0.00 f
A (in)                      0.00      0.00 f
U8/Z (XOR2_X1)              0.05      0.05 f
U7/ZN (AOI22_X1)             0.04      0.09 r
U6/ZN (INV_X1)               0.02      0.11 f
Cout (out)                   0.01      0.12 f
data arrival time             0.12

(Path is unconstrained)

Startpoint: B (input port)
Endpoint: Sum (output port)
Path Group: (none)
Path Type: max

Des/Clust/Port    Wire Load Model    Library
-----
tgl_full_adder    5K_hvratio_1_1    NangateOpenCellLibrary

Point           Incr      Path
-----
input external delay      0.00      0.00 r
B (in)                  0.00      0.00 r
U8/Z (XOR2_X1)           0.06      0.06 r
U5/Z (XOR2_X1)           0.05      0.11 r
Sum (out)                 0.01      0.12 r
data arrival time         0.12

(Path is unconstrained)
1
  
```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

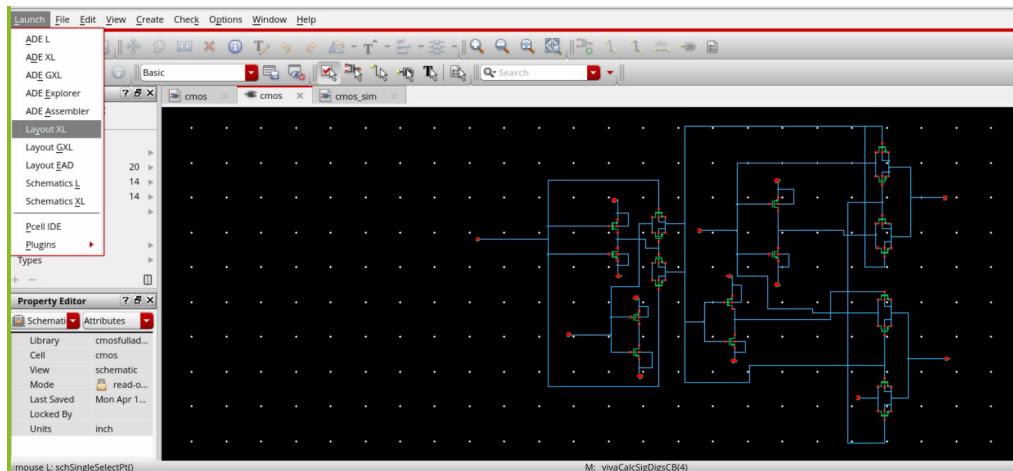
### Timing Report of TGL Full Adder

#### d) Layout Design in Cadence Virtuoso

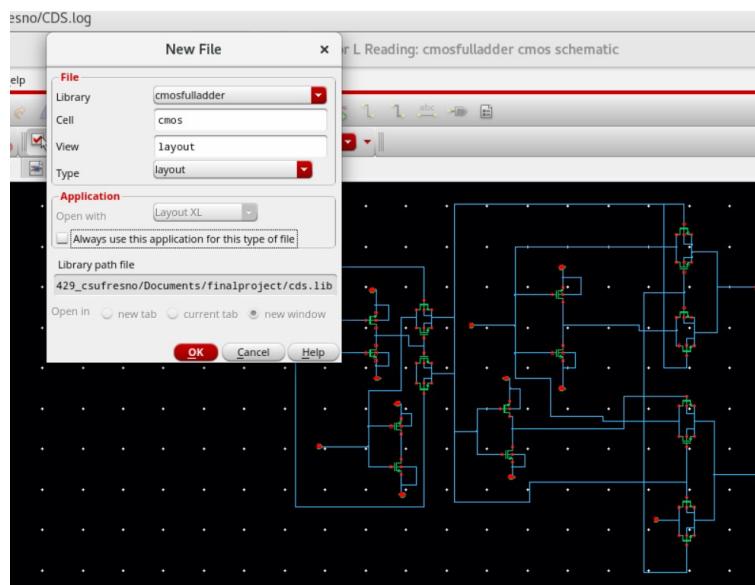
Following schematic design, simulation, synthesis, and verification, the physical layout of the TGL-based full adder & CMOS full adder was developed in Cadence Virtuoso Layout Suite XL. This process transforms the schematic into a physical representation ready for fabrication.

#### Step 1: Launching Layout XL Environment

Opened Virtuoso Layout Suite XL by selecting Launch > Layout XL from the schematic editor.

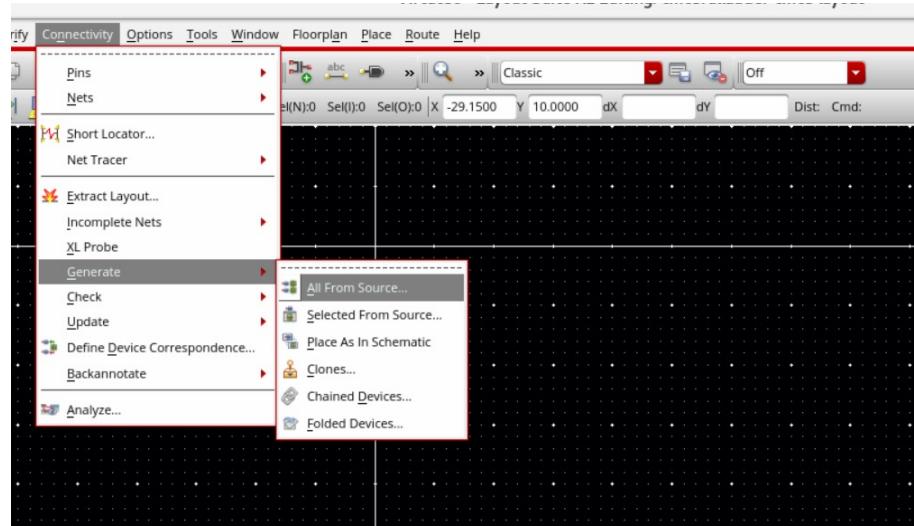


Created a new layout view in the same cmos\_fulladder library as the schematic.



## Step 2: Generating Layout from Schematic

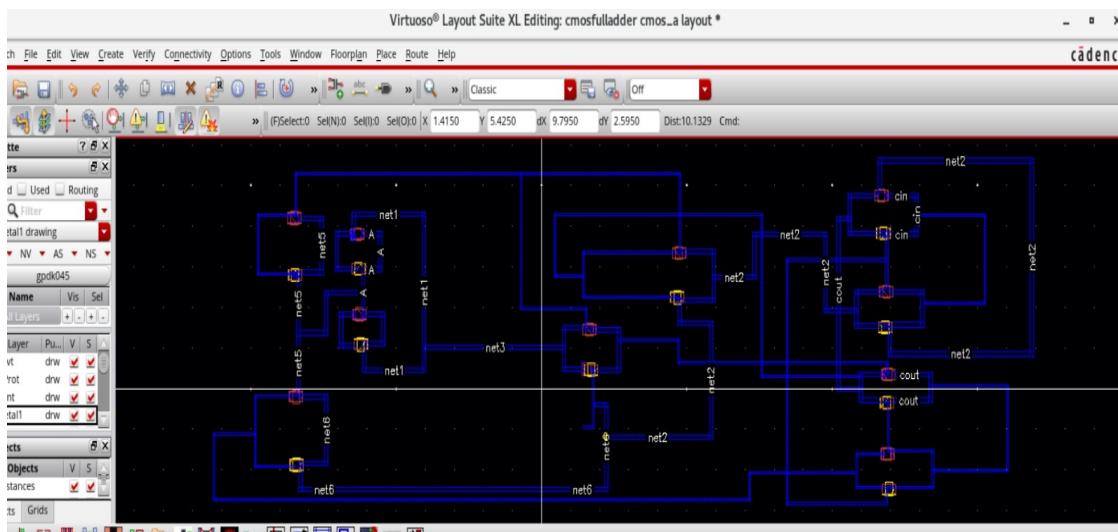
Used the Connectivity > Generate > All from Source option to import all devices and nets from the schematic.



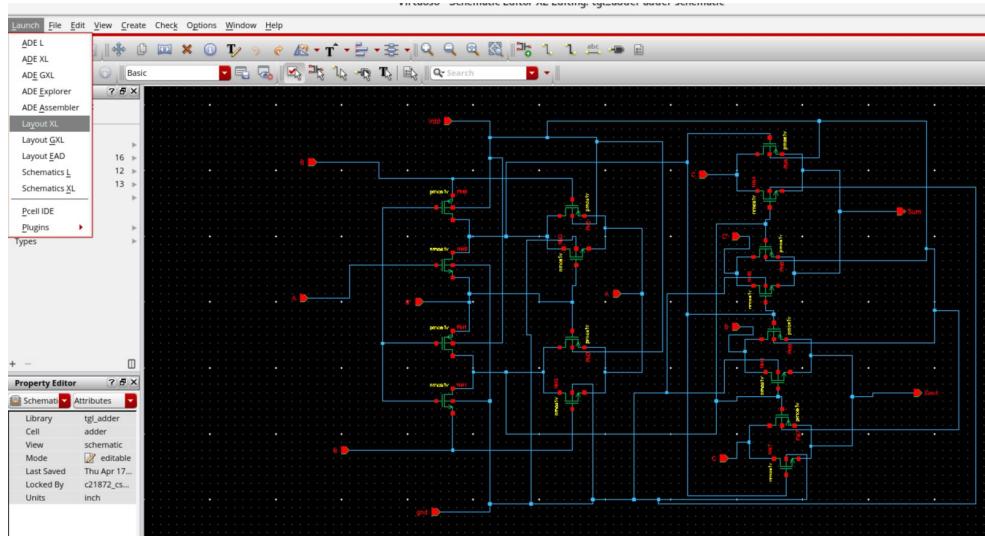
## Step 3: Routing and Connectivity

Used the Create Wire (narrow) or Create Wire (wide) tool to manually route interconnections between transistors.

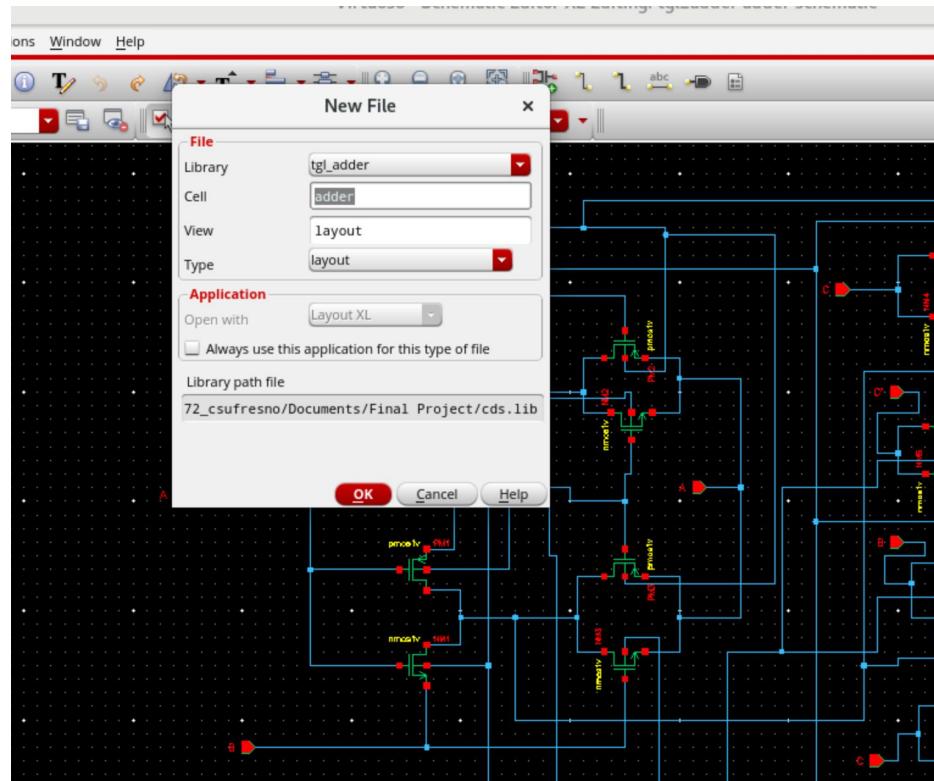
Ensured correct metal layer usage (e.g., Metal1, Metal2) for power, ground, and signal connections to minimize crosstalk and maximize reliability.



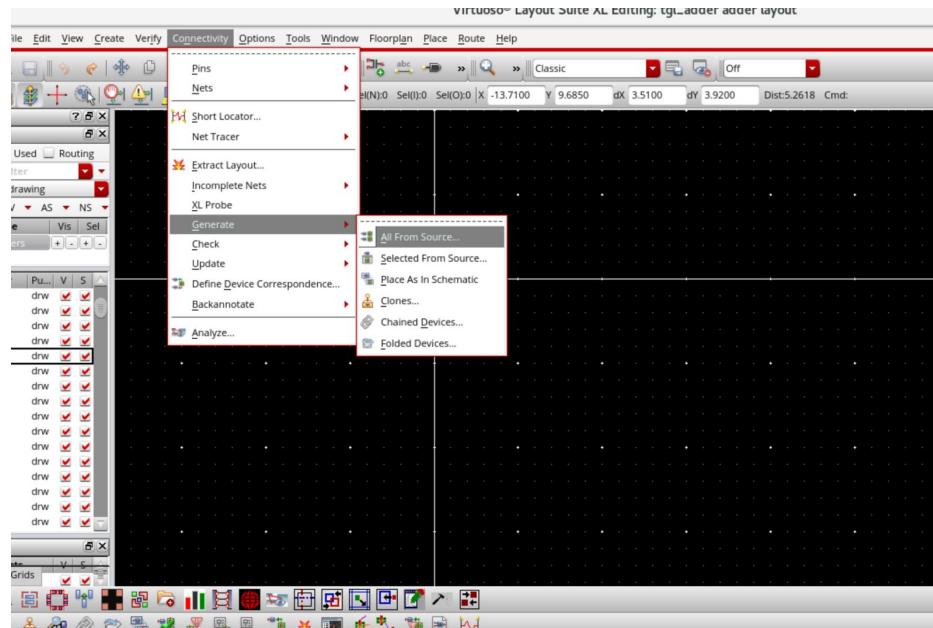
Similarly, for TGL full adder



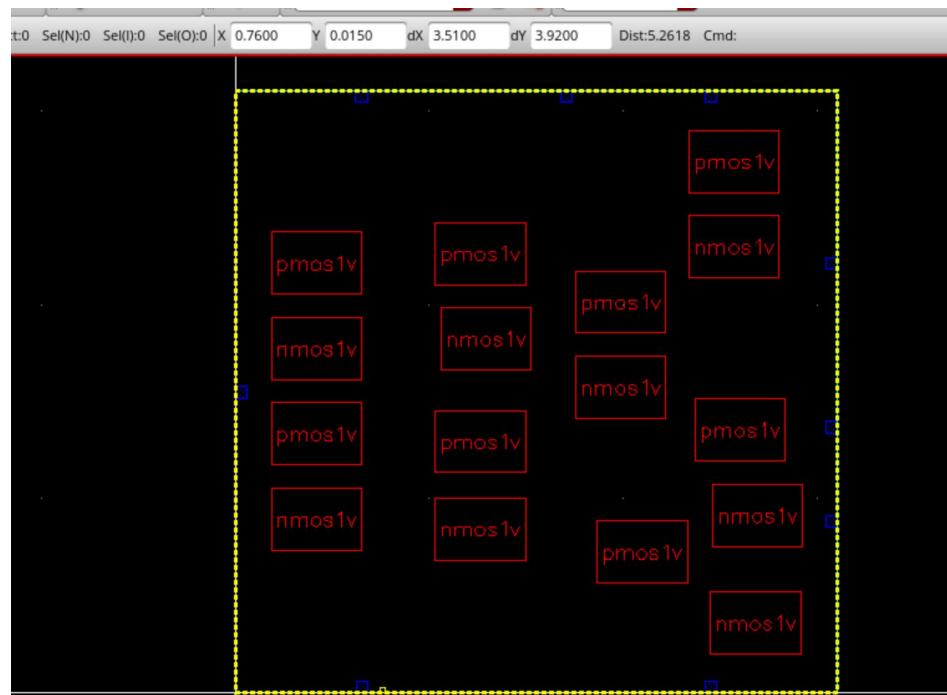
Created a new file for layout



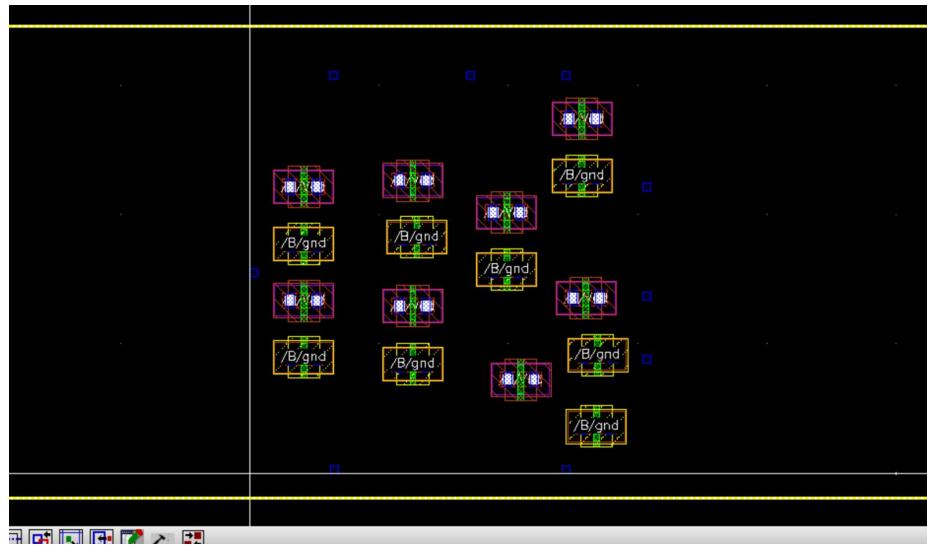
In the Layout XL window Connectivity -> generate -> all from source



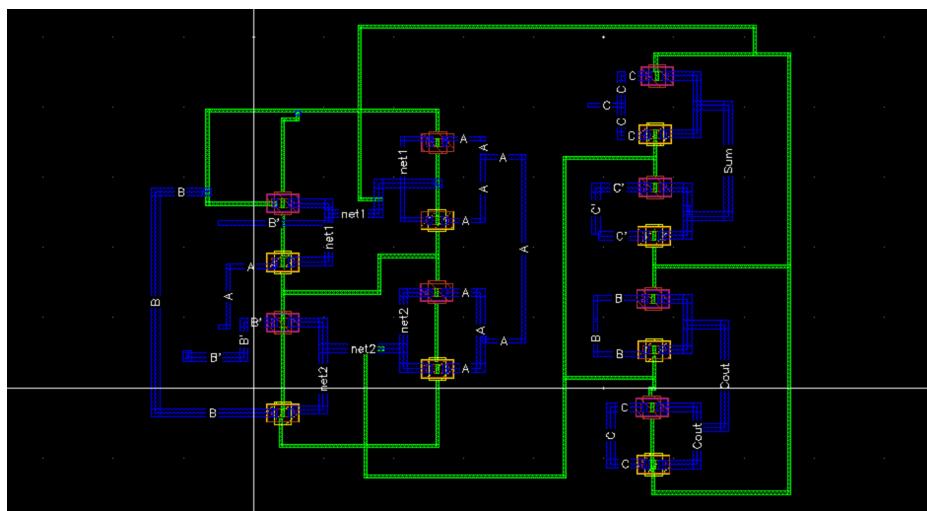
After connectivity we got this pmos and nmos transistors generated



From here we have clicked shift F and arranged all transistors according to the schematic design



Given Connections according to the Schematic design and completed layout of tgl based full adder



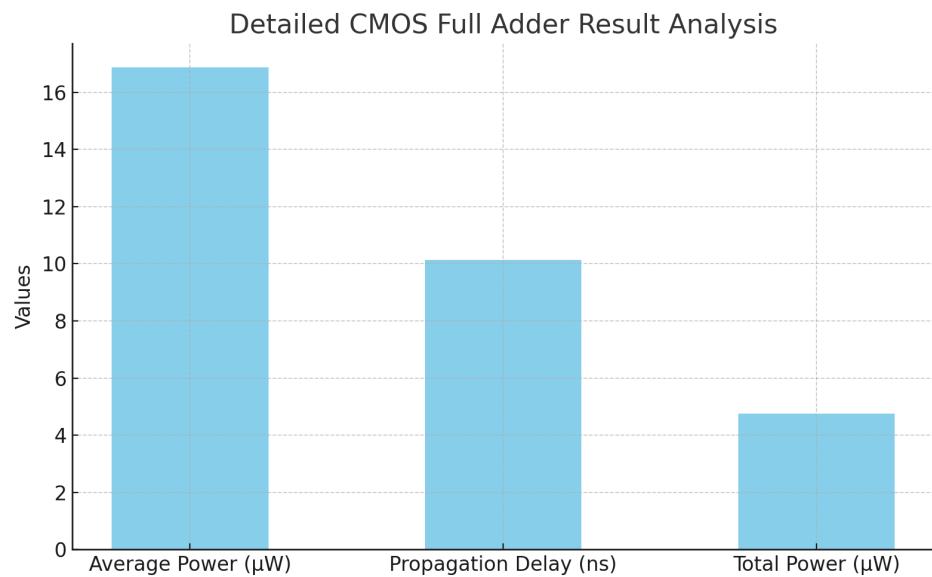
Layout of TGL Full adder

## 6. RESULTS AND ANALYSIS

### 6.1. Analysis of CMOS Full Adder

The CMOS Full Adder design was simulated and evaluated using both Cadence Virtuoso and Synopsys Design Compiler. In the transistor-level transient analysis performed in Virtuoso, the average power consumption was measured to be  $16.87 \mu\text{W}$ . This value was obtained by running a transient simulation with defined input patterns and capturing the power waveform for computation using the Cadence calculator tool. The propagation delay measured from the sum output during the rising and falling transitions was found to be  $10.14 \text{ ns}$ , indicating the time it takes for the output to stabilize after a change in input.

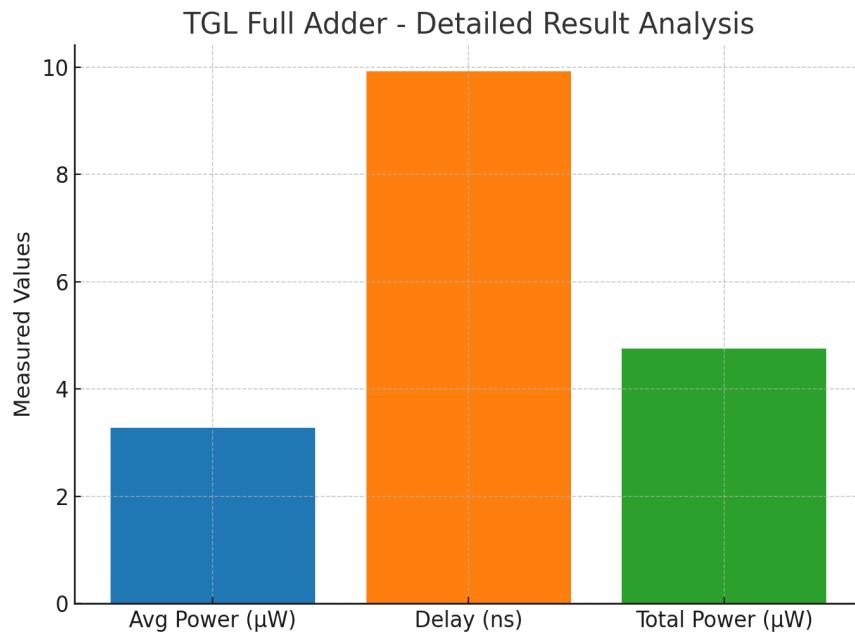
Moving to the synthesis results from Synopsys Design Compiler, the CMOS Full Adder showed a total cell area of  $5.054 \mu\text{m}^2$ . The synthesis report confirmed that the design used 4 combinational cells, 7 nets, and 5 ports. The power analysis indicated a total dynamic power of  $4.3621 \mu\text{W}$ , with  $2.6172 \mu\text{W}$  attributed to internal cell power and  $1.7448 \mu\text{W}$  to switching power. The leakage power was reported as  $396.1610 \text{ nW}$ , resulting in a total power of  $4.7582 \mu\text{W}$ . The timing analysis showed a data arrival time of  $0.13 \text{ ns}$ , following the longest path from the input B to the output Cout. These results establish a baseline for power, area, and timing performance using conventional CMOS logic.



## 6.2. Analysis of TGL Full Adder

The Transmission Gate Logic (TGL) Full Adder was also evaluated through the same set of analyses to ensure consistency in the comparison. From the transient simulation in Cadence Virtuoso, the average power consumption was significantly lower at  $3.274 \mu\text{W}$ . This lower power value demonstrates the efficiency of TGL in reducing unnecessary switching activity and leakage, making it a highly power-efficient design. The propagation delay for the sum output was measured to be  $9.923 \text{ ns}$ , slightly faster than the CMOS counterpart, indicating improved response time under the same simulation conditions.

In the synthesis results, the TGL Full Adder exhibited the same total cell area of  $5.054 \mu\text{m}^2$  as the CMOS Full Adder, using 4 combinational cells, 7 nets, and 5 ports. The power report from Synopsys Design Compiler showed a total dynamic power of  $4.3621 \mu\text{W}$ , with  $2.6172 \mu\text{W}$  coming from internal power and  $1.7448 \mu\text{W}$  from switching power. The leakage power was again  $396.1610 \text{ nW}$ , resulting in a total power of  $4.7582 \mu\text{W}$ . The timing analysis reported a data arrival time of  $0.12 \text{ ns}$  from the input B to the output Sum, indicating a marginally faster performance compared to the CMOS implementation.



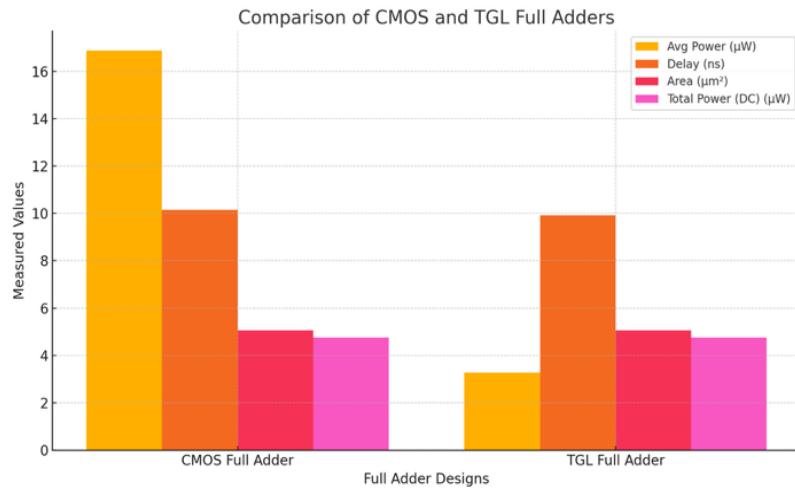
## 6.3 Comparison of CMOS and TGL Full Adder

The designed CMOS and Transmission Gate Logic (TGL) based Full Adders were thoroughly analyzed to compare their performance in terms of power consumption, propagation delay, and area utilization. The analysis was carried out using Cadence Virtuoso for transient simulations to capture dynamic behavior and

Synopsys Design Compiler to estimate synthesized area, timing, and power. This multi-tool evaluation ensured that the designs were assessed both at the transistor level and the gate-level for a more comprehensive understanding.

In terms of power consumption, the results from Virtuoso simulations indicated that the TGL Full Adder is significantly more power-efficient than the CMOS Full Adder. The CMOS Full Adder consumed an average power of approximately  $16.87 \mu\text{W}$ , whereas the TGL Full Adder exhibited a much lower average power consumption of about  $3.274 \mu\text{W}$ . Despite this difference at the simulation level, the synthesized reports from Synopsys Design Compiler showed identical total power for both designs, reported as  $4.7582 \mu\text{W}$ . This is because both designs use the same cell library for synthesis, which standardizes power figures at the gate level. Nonetheless, the transistor-level analysis highlights the superior switching efficiency of the TGL design.

Propagation delay, another critical performance metric, was also evaluated using transient simulation in Virtuoso. The CMOS Full Adder demonstrated a propagation delay of approximately  $10.14 \text{ ns}$ , while the TGL Full Adder achieved a slightly faster response with a delay of  $9.923 \text{ ns}$ . This marginal improvement suggests that the TGL architecture not only reduces power consumption but also provides a minor advantage in speed, making it suitable for timing-critical applications.



The area analysis conducted using Synopsys Design Compiler revealed that both the CMOS and TGL Full Adders occupy the same total cell area of  $5.054 \mu\text{m}^2$ . This consistency is expected since both designs were synthesized using the same PDK and standard cell library. Therefore, while the TGL design shows clear

advantages in power and timing, it does not incur any additional area overhead, making it a space-efficient choice.

Further timing analysis using DC Compiler provided the data arrival times for both designs. The CMOS Full Adder showed a data arrival time of 0.13 ns, whereas the TGL Full Adder achieved a slightly better value of 0.12 ns. This reinforces the observation that the TGL Full Adder provides minor but measurable improvements in performance while maintaining the same area and comparable synthesized power.

## 7. CONCLUSION

In this project, two different full adder architectures—Conventional CMOS Full Adder and Transmission Gate Logic (TGL) Full Adder—were designed, functionally verified, simulated, synthesized, and analyzed for key performance metrics including power consumption, propagation delay, and area utilization. The CMOS Full Adder, while functionally accurate, exhibited higher average power consumption and slightly greater propagation delay when compared to the TGL Full Adder. Specifically, transient simulations in Cadence Virtuoso showed that the CMOS Full Adder consumed approximately  $16.87 \mu\text{W}$  of average power with a propagation delay of 10.14 ns, whereas the TGL Full Adder demonstrated a lower average power consumption of  $3.274 \mu\text{W}$  and a slightly improved delay of 9.923 ns.

Further, both designs were synthesized using Synopsys Design Compiler, revealing identical total cell areas of  $5.054 \mu\text{m}^2$  and matching total power estimations of  $4.7582 \mu\text{W}$ . While the synthesized area and total power results appeared similar due to the use of the same standard cell library, the transistor-level simulations highlighted the superior switching efficiency and lower dynamic power characteristics of the TGL Full Adder.

Overall, the TGL Full Adder emerged as the better-performing architecture in this study, achieving significant power savings and slightly better timing performance without incurring additional area overhead. These results demonstrate that the TGL-based full adder is more suitable for low-power, high-performance VLSI applications, such as battery-powered devices, AI accelerators, and portable embedded systems, where minimizing power consumption and maximizing speed are critical design priorities. The success of this design also reinforces the relevance of Transmission Gate Logic as a viable alternative to traditional CMOS logic in the pursuit of energy-efficient digital circuit design.

## 7. REFERENCES

- [1] N. Jain, P. Gour, and B. Shrman, "A High-Speed Low Power Adder in Dynamic Logic Based on Transmission Gate," in *Proc. Int. Conf. Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1–6, 2015, doi: 10.1109/ICCPCT.2015.7159393.
- [2] C. K. Ray and K. Srinivasarao, "Design and Implementation of Low Power Carry Select Adder Using Transmission Gate Logic," *IOSR J. VLSI Signal Process. (IOSR-JVSP)*, vol. 5, no. 3, pp. 28–32, May-Jun. 2015, doi: 10.9790/4200-05312832.
- [3] A. Baliga and D. Yagain, "Design of High-Speed Adders Using CMOS and Transmission Gates in Submicron Technology: A Comparative Study," in *Proc. 4th Int. Conf. Emerging Trends in Engineering & Technology*, pp. 284–289, 2011, doi: 10.1109/ICETET.2011.25.
- [4] K. J. Singh, T. Sharan, and H. Tarunkumar, "High Speed and Low Power Basic Digital Logic Gates, Half-Adder, and Full-Adder Using Modified Gate Diffusion Input Technology," *J. VLSI Design Tools Technol.*, vol. 8, no. 1, pp. 34–42, 2018, doi: 10.37628/JOVDTT.v8i1.34-42.
- [5] K. Navi, V. Foroutan, M. Rahimi Azghadi, M. Maeen, M. Ebrahimpour, and M. Kaveh, "A Novel Low-Power Full-Adder Cell with New Technique in Designing Logical Gates Based on Static CMOS Inverter," *Microelectron. J.*, vol. 40, no. 10, pp. 1441–1448, 2009, doi: 10.1016/j.mejo.2009.06.005.
- [6] M. Mewada, M. Zaveri, and R. Thakker, "Improving the Performance of Transmission Gate and Hybrid CMOS Full Adders in Chain and Tree Structure Architectures," *Integr. VLSI J.*, vol. 69, pp. 381–392, 2019, doi: 10.1016/j.vlsi.2019.09.002.