Full Custom Design

3 Bit Ripple Carry Adder

Project report submitted for

Vth Semester Course Project

in

Department of Electronics and Communication Engineering

By,

Bhavesh Balendra (211010221)

Soumya Rayast (211010248)

Guided By: Dr. Rohit Chaurasiya

Department of Electronics and Communication Engineering

Dr. Shyama Prasad Mukherjee

International Institute of Information Technology, Naya Raipur

(A Joint Initiative of Govt. of Chhattisgarh and NTPC)

Email: iiitnr@iiitnr.ac.in, Tel: (0771) 2474040, Web: www.iiitnr.ac.in

3 Bit Ripple carry Adder

Introduction

A Ripple Carry Adder (RCA) is a fundamental digital circuit used for binary addition. It operates by adding two binary numbers bit by bit, starting from the least significant bit (LSB) to the most significant bit (MSB). The key characteristic of a ripple carry adder is that the carry-out from each bit addition ripples or propagates to the carry-in of the next higher-order bit. This sequential carry propagation introduces a delay in the overall addition process, making the ripple carry adder relatively slower compared to more advanced adder architectures

A 3-bit Ripple Carry Adder (RCA) is a digital circuit designed for the addition of two 3-bit binary numbers. This circuit is composed of a series of full-adders, each responsible for adding the corresponding bits of the operands and propagating the carry to the next stage. The basic building blocks used in each full-adder are AND gates, OR gates, and XOR gates.

For each bit position, the XOR gate calculates the sum of the bits, while the AND gate computes the carry generated by that specific bit. The OR gate combines the carries from the current bit addition with the carry-in from the previous stage, producing the final carry-out. The sum and carry-out from each full-adder contribute to the subsequent stage, creating a ripple effect from the least significant bit (LSB) to the most significant bit (MSB).

While the 3-bit ripple carry adder is simple and conceptually easy to grasp, its operational speed is limited by the sequential carry propagation. This arises because each stage has to wait for the carry signal from the previous stage before it can complete its addition. Nonetheless, the use of basic logic gates like AND, OR, and XOR makes the circuit implementation straightforward, providing a solid foundation for understanding more complex digital arithmetic circuits.

Diagram

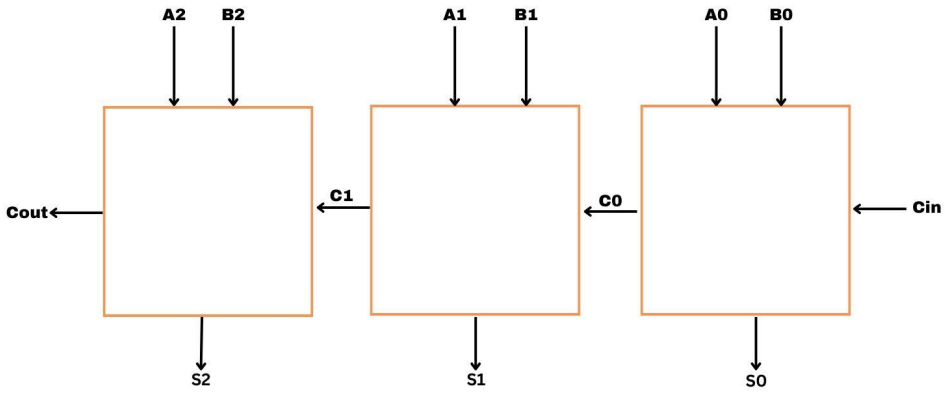


Fig: 3 Bit Ripple Carry Adder

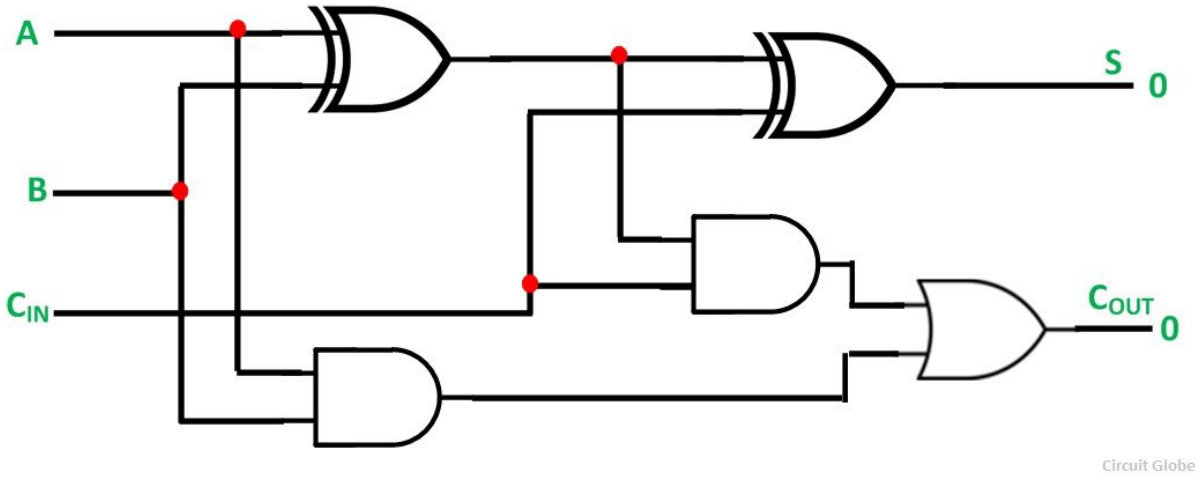


Fig : Full Adder

Input			Output	
A	B	Cin	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

Table : Truth table for Full Adder

Working of 3-bit Ripple Carry Adder

A 3-bit Ripple Carry Adder (RCA) is composed of three full-adder circuits connected in series. Each full-adder takes two input bits (A and B) and a carry-in (Cin), and produces a sum (S) and a carry-out (Cout). The carry-out from one stage serves as the carry-in for the next, creating a ripple effect.

Here's a basic description of the operation of a 3-bit Ripple Carry Adder:

Input: The 3-bit binary numbers to be added are labeled as A2A1A0 and B2B1B0, where A2, A1, A0, B2, B1, and B0 are the individual bits.

Stage 1 (Least Significant Bit - LSB): The least significant bits (A0 and B0) are input into the first full-adder along with an initial carry-in (usually set to 0). The full-adder generates a sum bit (S0) and a carry-out (Cout0).

- $S_0 = A_0 \oplus B_0 \oplus C_{\text{in}}$
- $C_{\text{out}_0} = (A_0 \cdot B_0) + (C_{\text{in}} \cdot (A_0 \oplus B_0))$

Stage 2 (Next Bit): The sum bit (S0) from the first stage is fed into the second full-adder along with the next set of bits (A1, B1) and the carry-out (Cout0) from

- $S_1 = A_1 \oplus B_1 \oplus C_{\text{out}_0}$
- $C_{\text{out}_1} = (A_1 \cdot B_1) + (C_{\text{out}_0} \cdot (A_1 \oplus B_1))$

Stage 3 (Most Significant Bit - MSB): The process is repeated for the most significant bits (A2, B2) with the carry-out (Cout1) from the previous stage.

- $S_2 = A_2 \oplus B_2 \oplus C_{\text{out}_1}$
- $C_{\text{out}_2} = (A_2 \cdot B_2) + (C_{\text{out}_1} \cdot (A_2 \oplus B_2))$

Output: The final sum (S2S1S0) is formed by concatenating the sum bits from each stage. This sequence of calculations illustrates the step-by-step process of a 3-bit Ripple Carry Adder. The carry-out from the MSB stage represents the overall carry-out of the addition operation

Cadence Virtuoso

Cadence Virtuoso is a widely used electronic design automation (EDA) tool that encompasses a suite of tools for designing and simulating integrated circuits. Within the Cadence Virtuoso environment, the GPDK (Generic Process Design Kit) provides a standardized set of parameters, models, and design rules for a specific semiconductor process technology.

The GPDK 180 nm technology in Cadence Virtuoso is tailored to support the design and simulation of integrated circuits at the 180 nm process node. This technology node represents a milestone in semiconductor manufacturing, offering a balance between performance and power consumption. Designers using Cadence Virtuoso with the GPDK for 180 nm can leverage a comprehensive set of tools for schematic entry, layout, and simulation to create and validate their designs. The technology-specific models and parameters provided in the GPDK ensure accurate simulation results, aiding designers in optimizing their circuits for performance, power efficiency, and manufacturability. Overall, the Cadence Virtuoso platform, coupled with the GPDK for 180 nm technology, facilitates the design process and enables engineers to create sophisticated integrated circuits within the constraints of this specific semiconductor process.

DRC (Design Rule Check) and LVS (Layout vs. Schematic) are crucial processes in the domain of semiconductor and integrated circuit (IC) design, ensuring the integrity and functionality of the designed circuits.

Design Rule Check (DRC):

DRC is a validation process that ensures the physical layout of an IC adheres to the specified manufacturing rules and constraints. These rules encompass guidelines related to minimum feature size, spacing, width, and other geometrical parameters defined by the semiconductor fabrication process. DRC is essential for identifying and rectifying potential issues that may arise during the manufacturing process. It helps prevent short circuits, manufacturing defects, and other layout-related problems, ensuring that the final physical layout is compliant with the semiconductor foundry's capabilities and requirements.

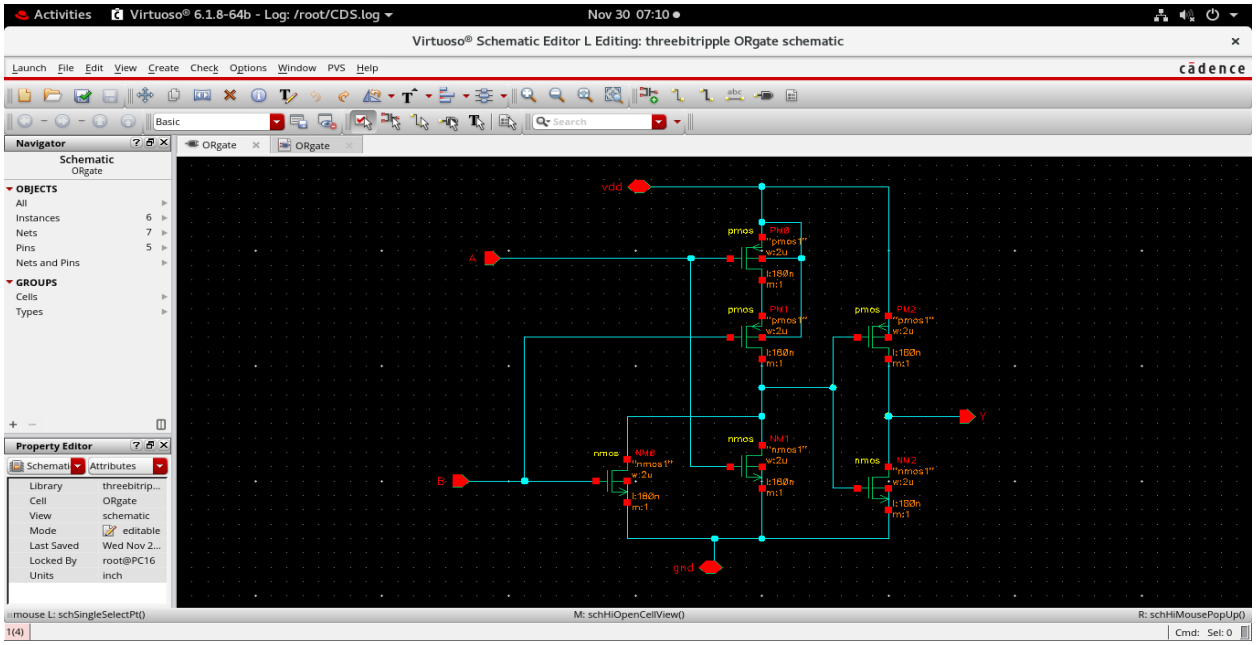
Layout vs. Schematic (LVS):

LVS is a verification process that compares the schematic representation of a circuit with its physical layout to confirm their consistency. It checks that the connections, devices, and components on the layout match the intended schematic design. LVS ensures that the fabricated chip will operate as intended, avoiding issues such as missing or incorrectly connected components. It is a critical step to verify that the design's electrical functionality matches its physical representation, reducing the risk of errors that could lead to faulty or non-functional ICs.

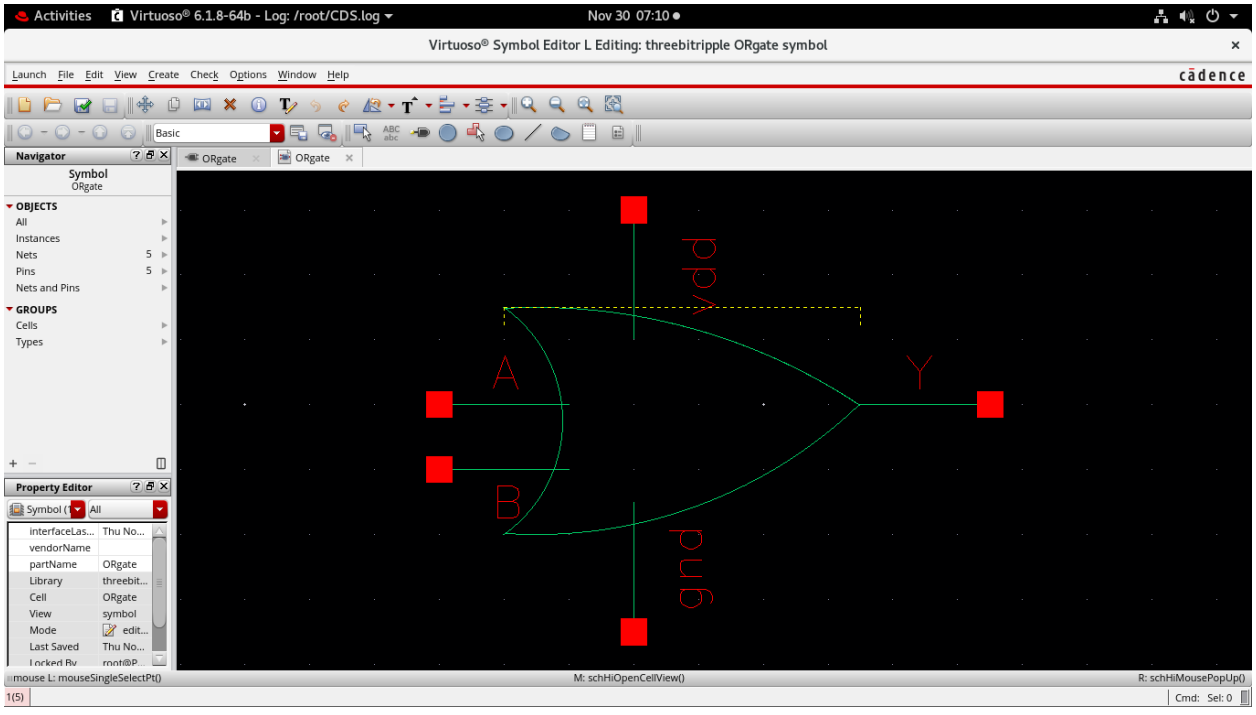
Process of making 3-Bit Ripple Carry Adder

Or gate

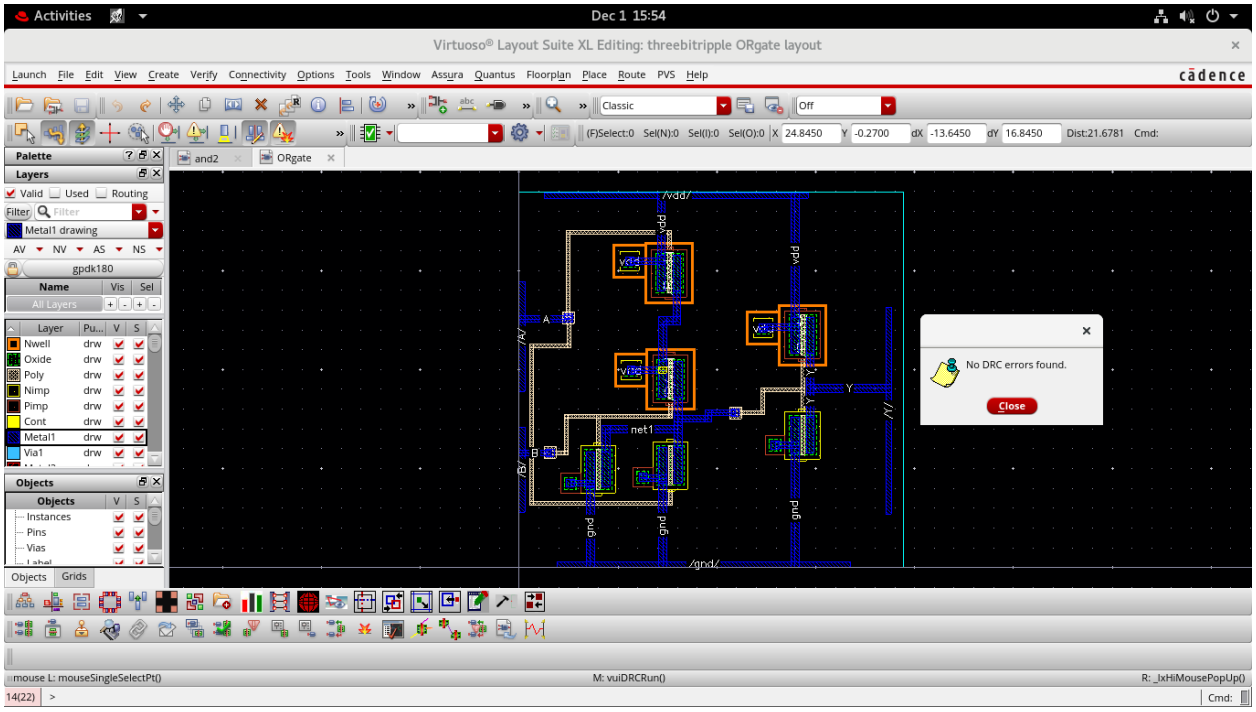
:Schematic



:Symbol

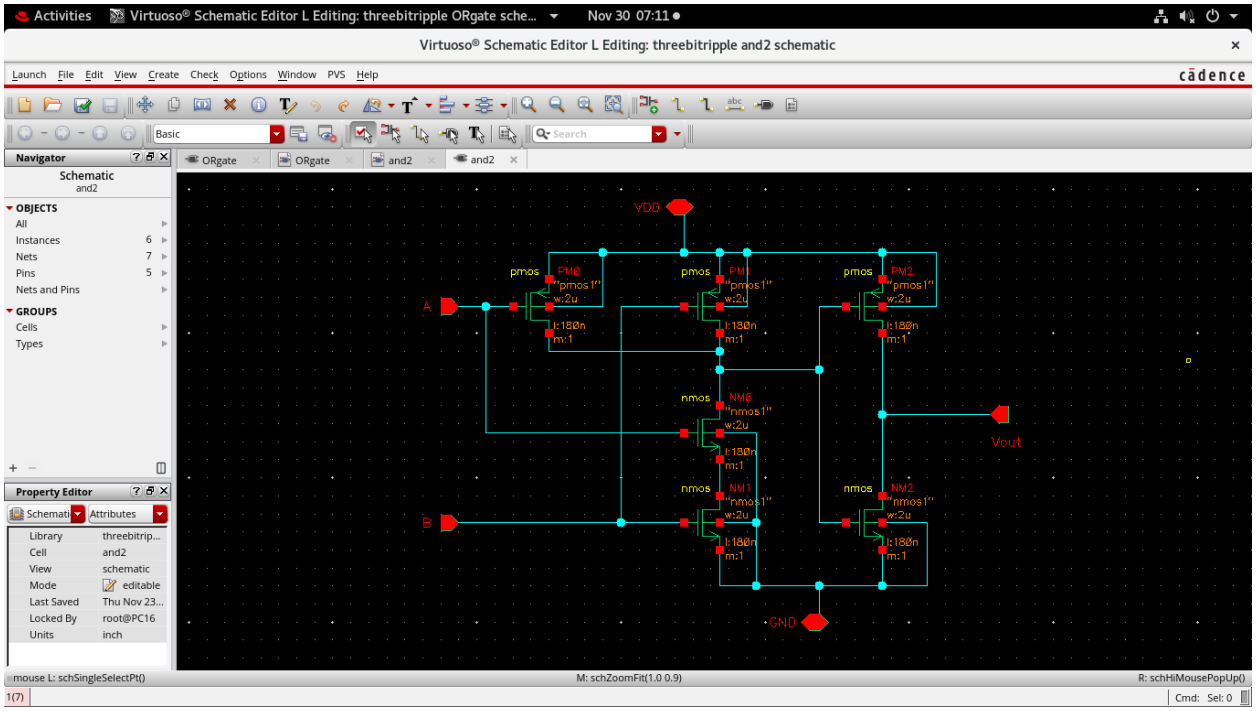


OR gate layout with DRC check

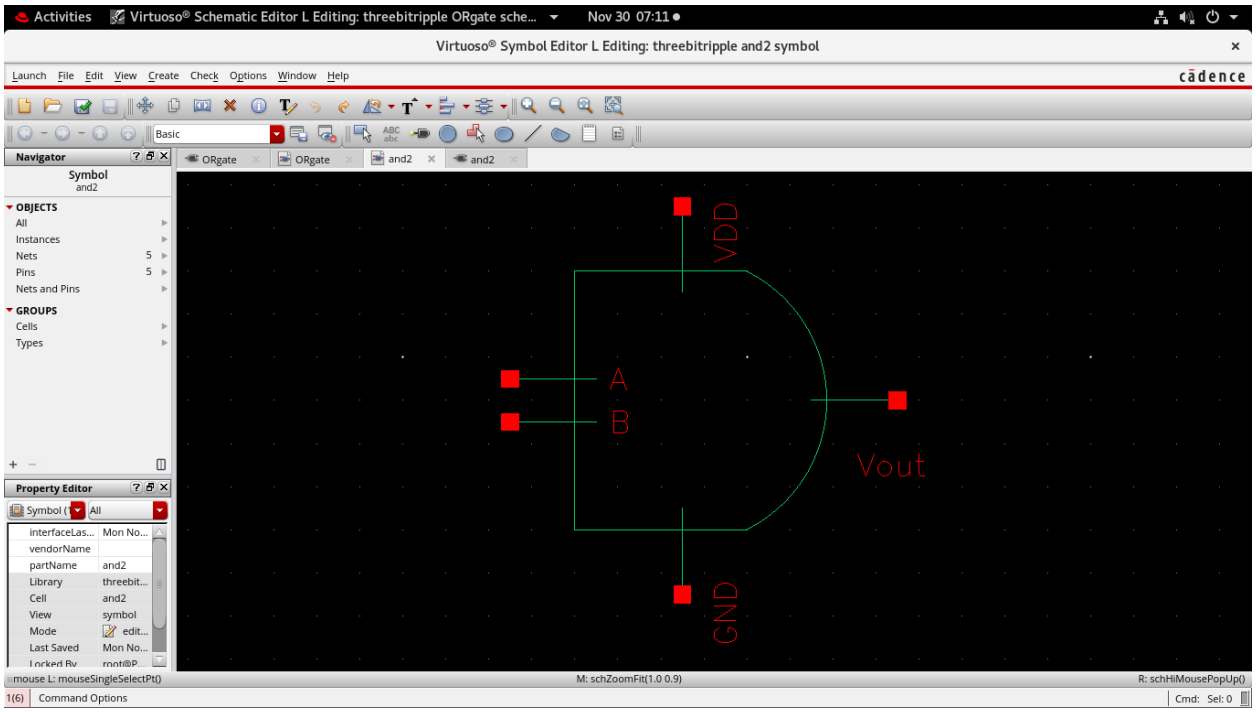


And

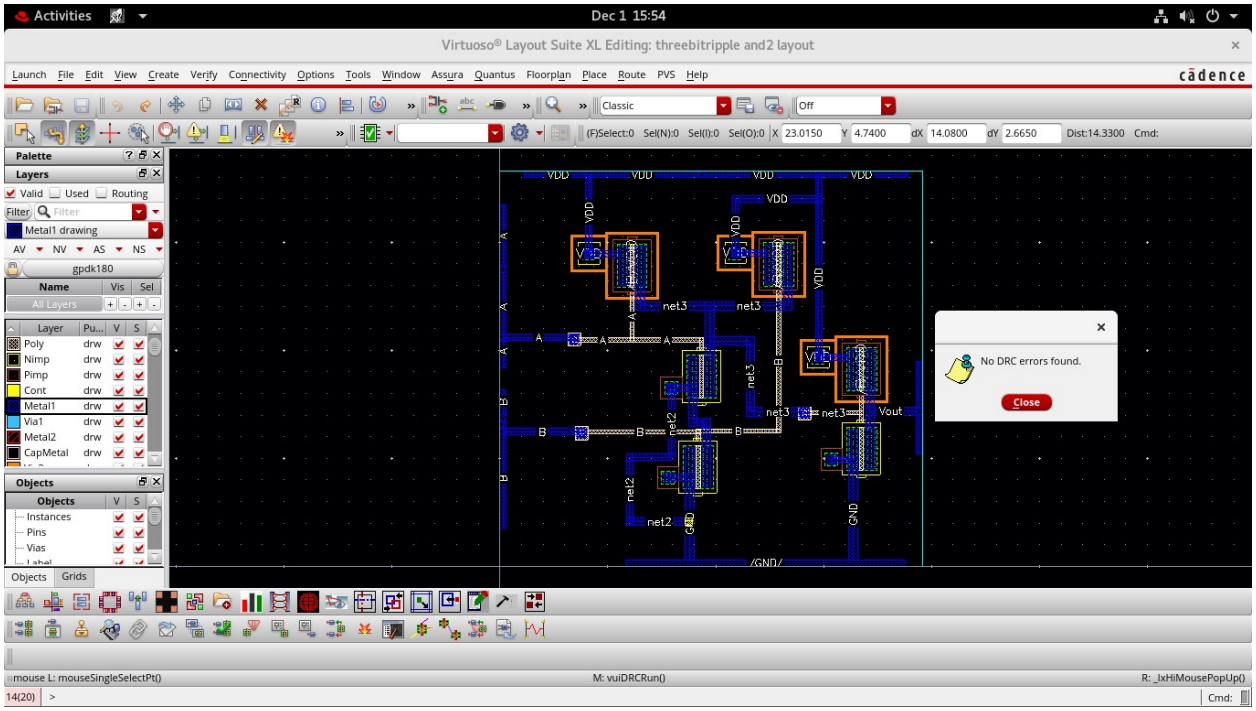
:Schematic



:Symbol

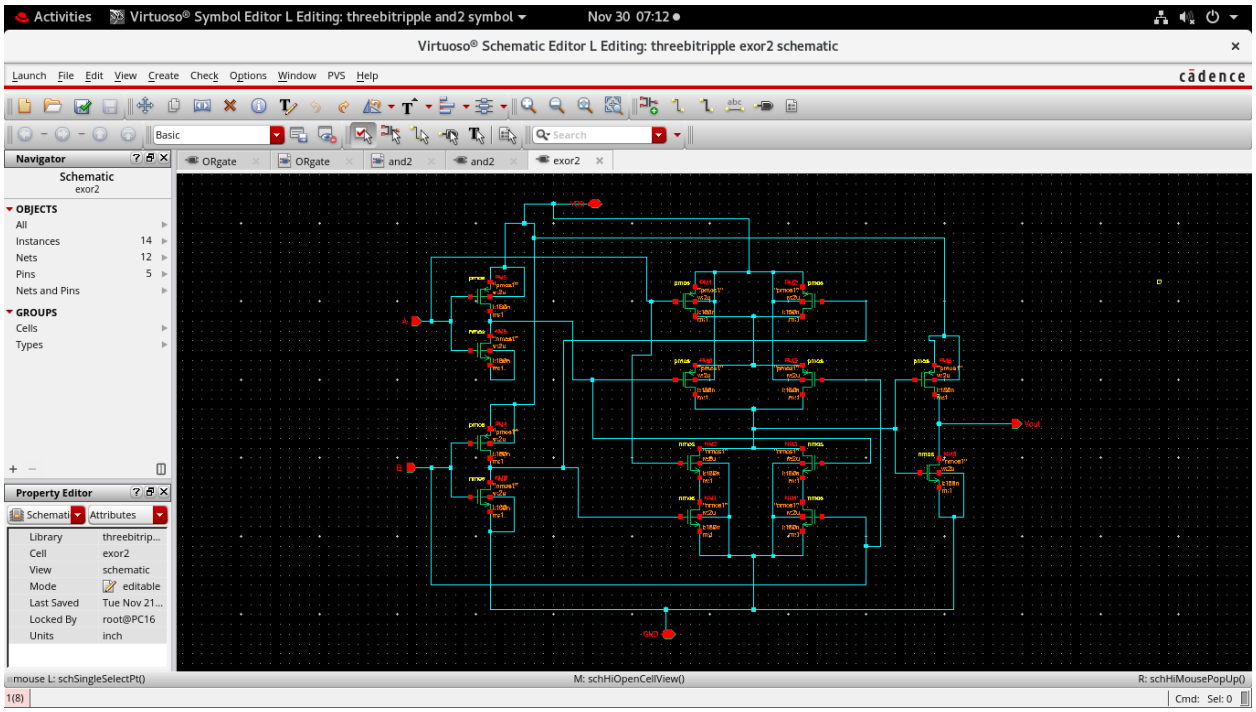


And Gate layout with DRC check

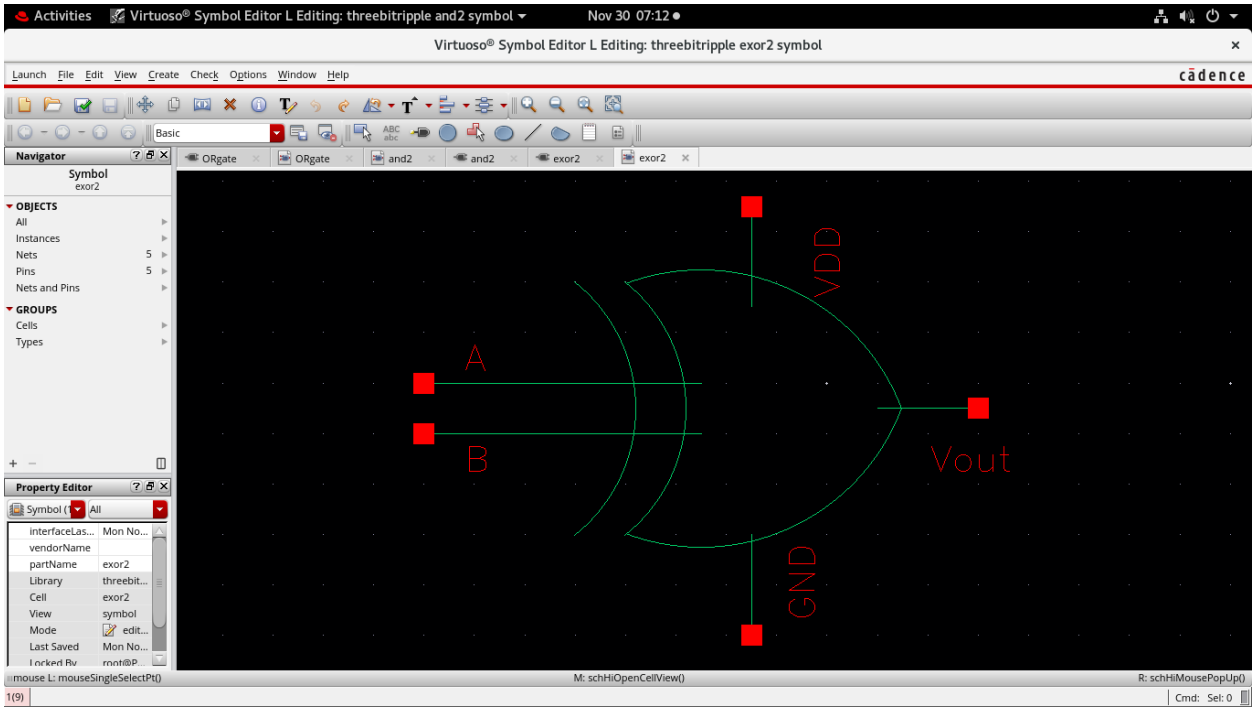


Xor

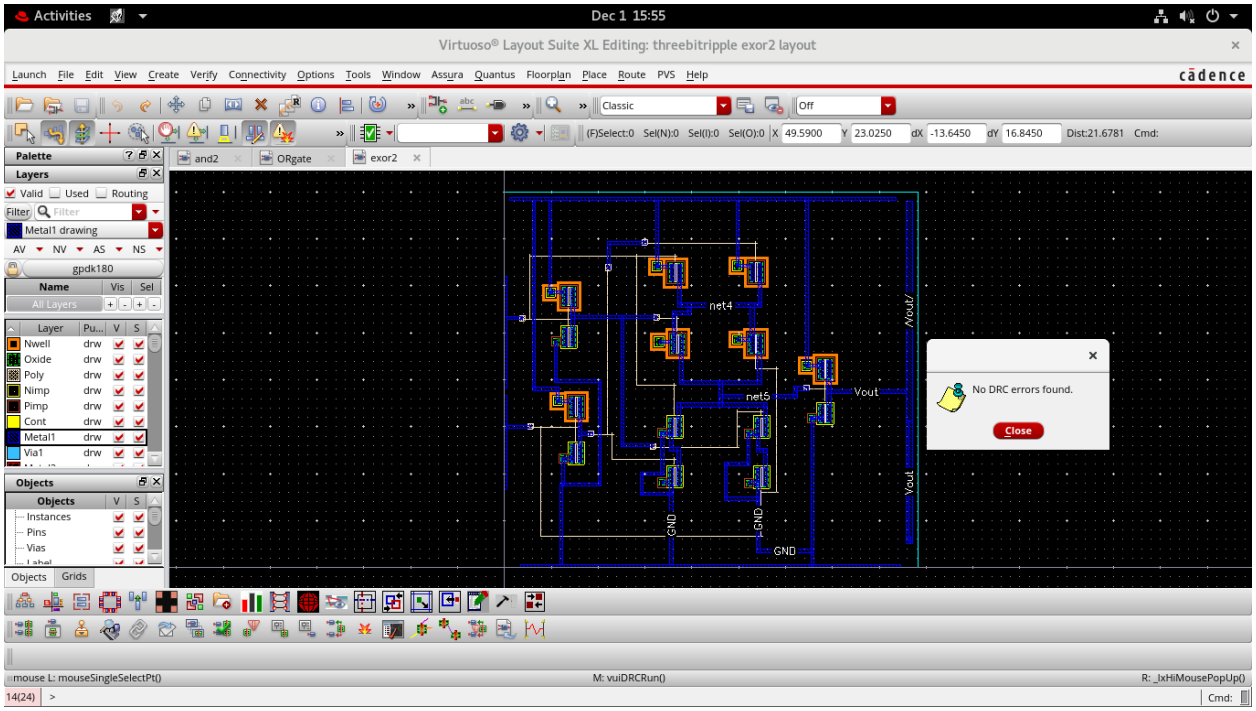
:Schematic



:Symbol

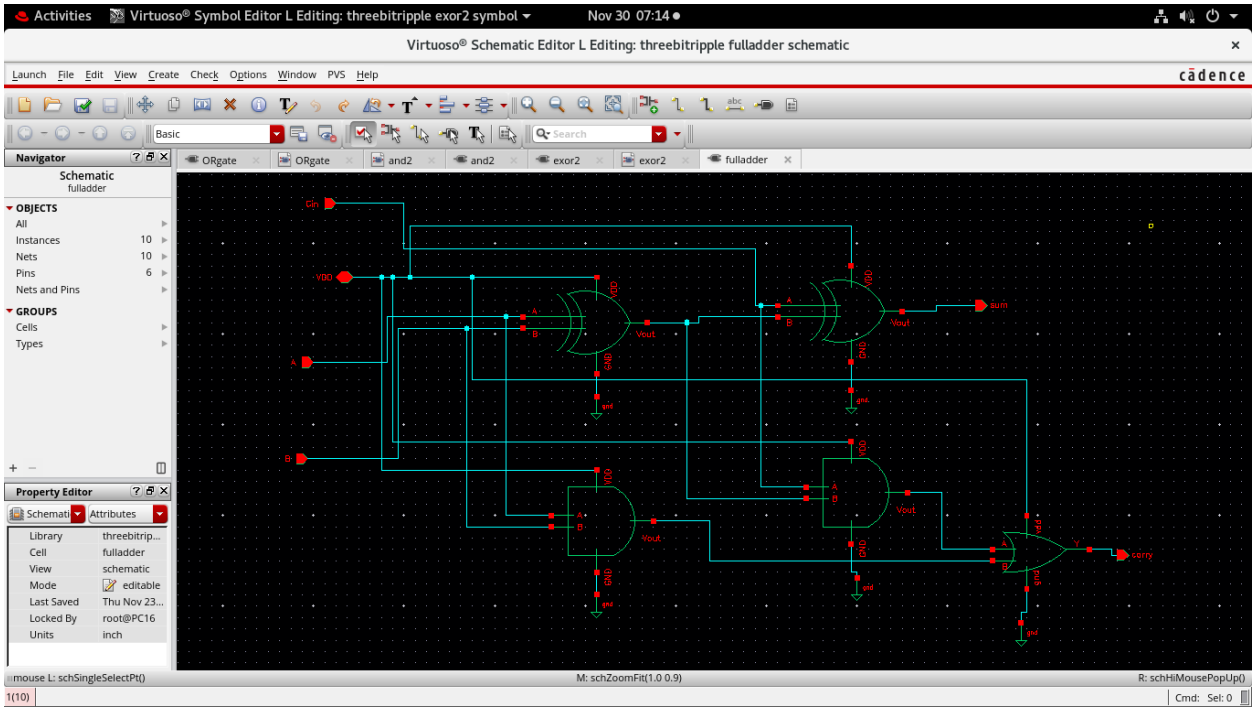


XOR gate layout and DRC check

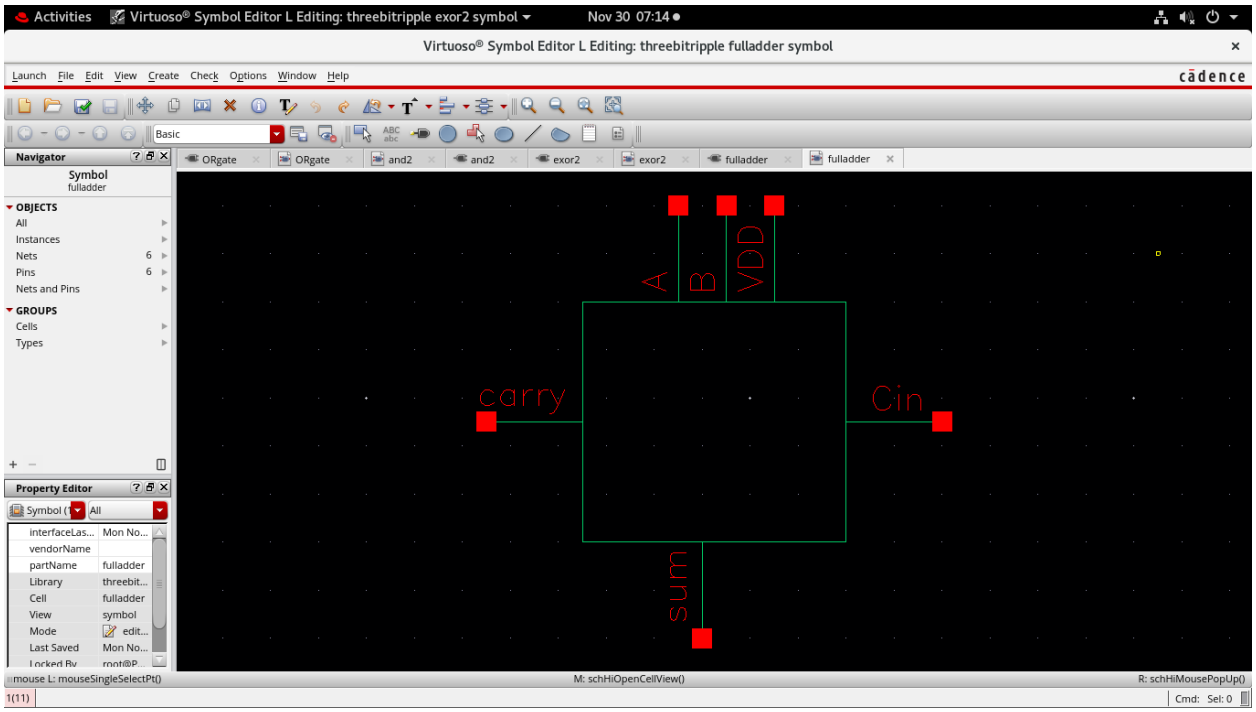


Full adder

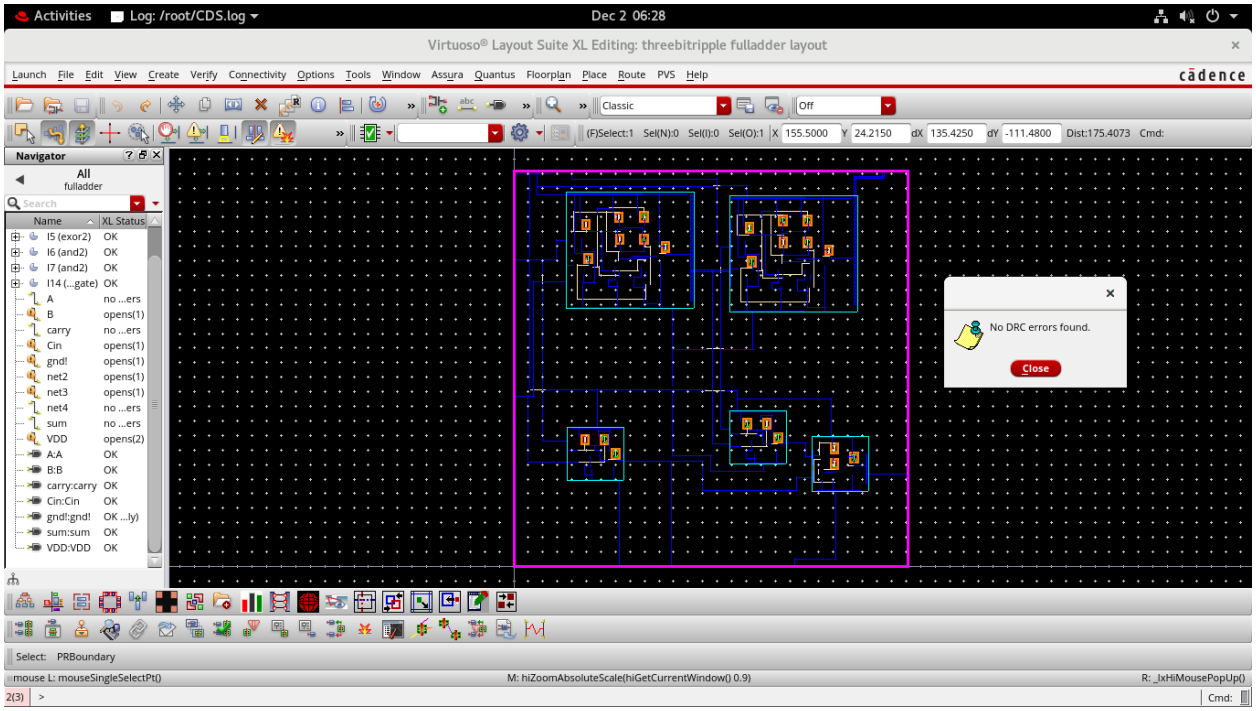
:Schematic



:Symbol

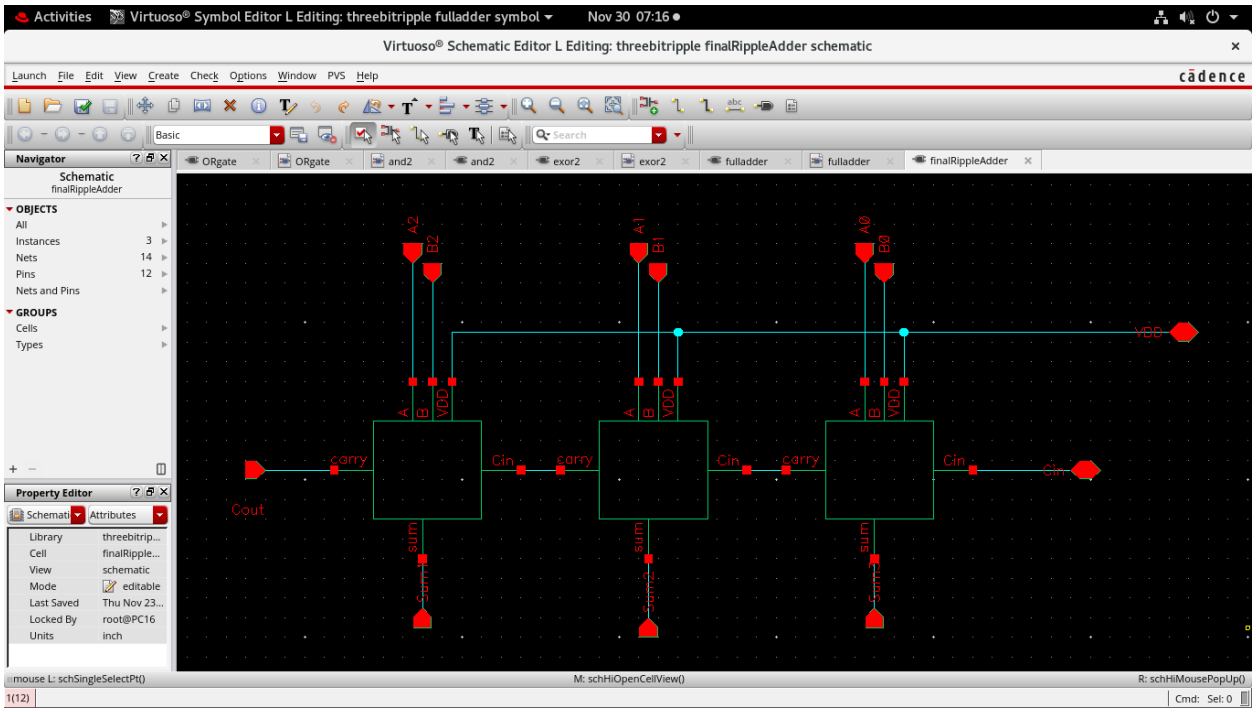


Layout with DRC check

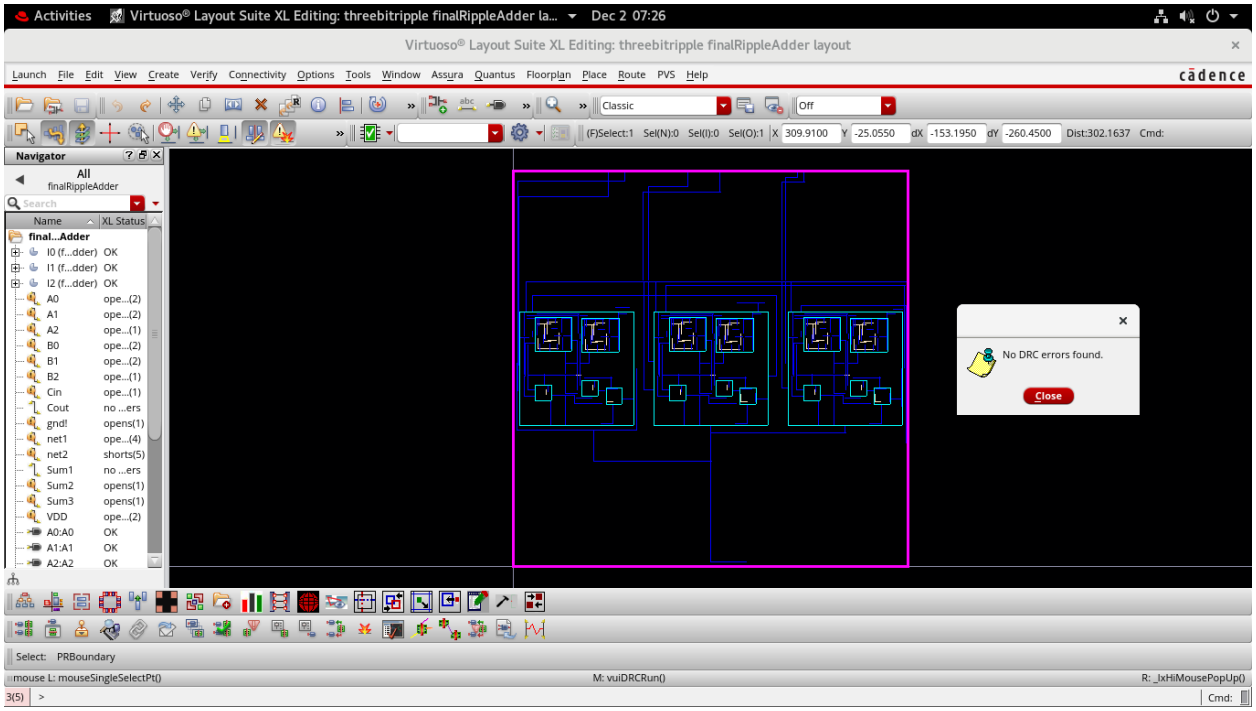


3 Bit Ripple carry adder

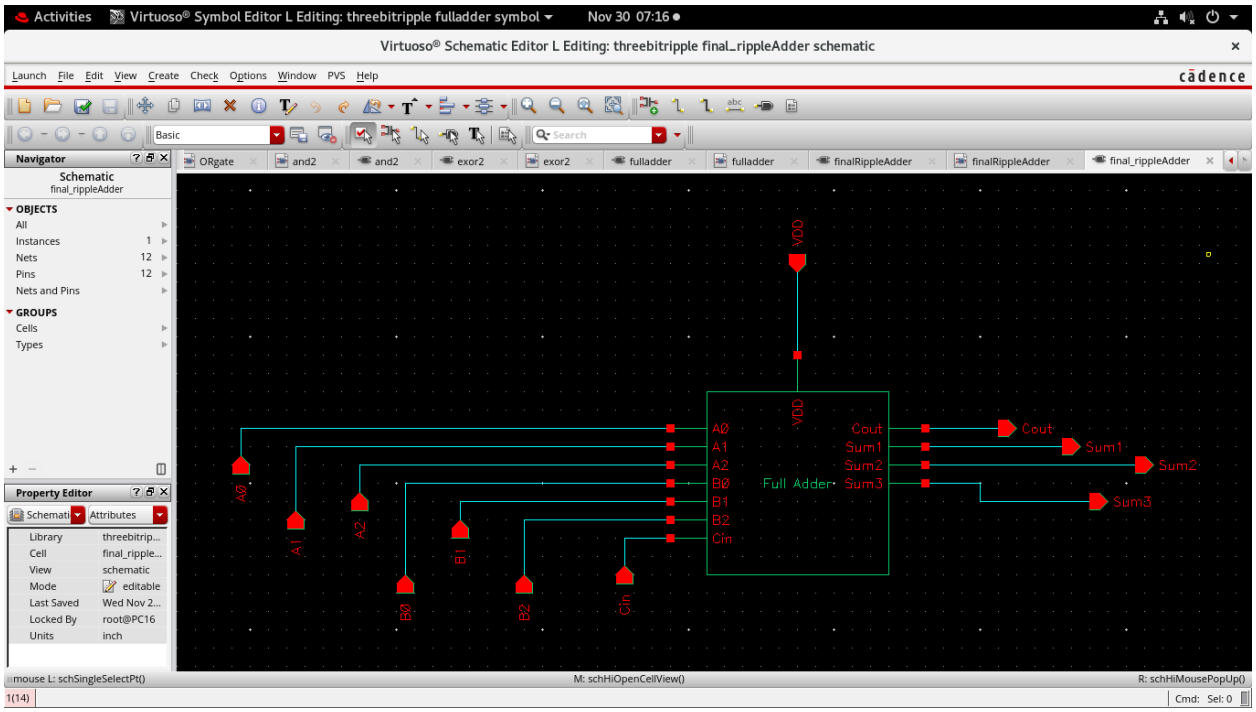
:Schematic



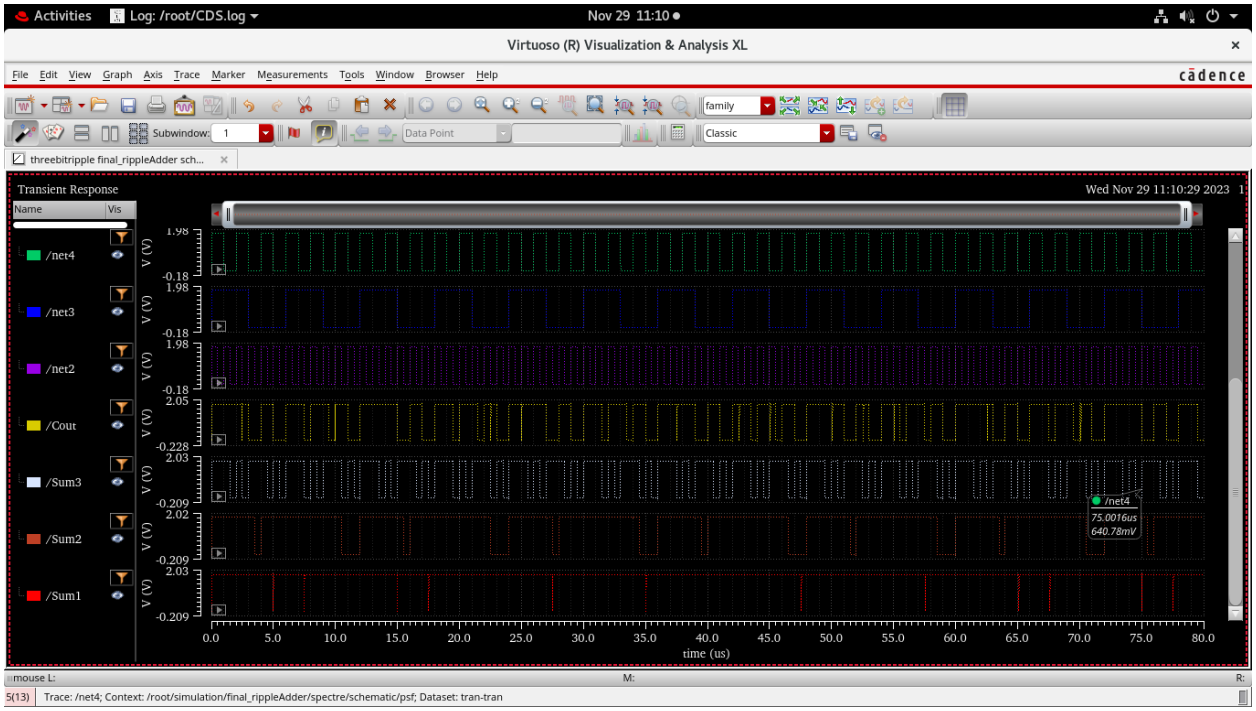
Layout with DRC check

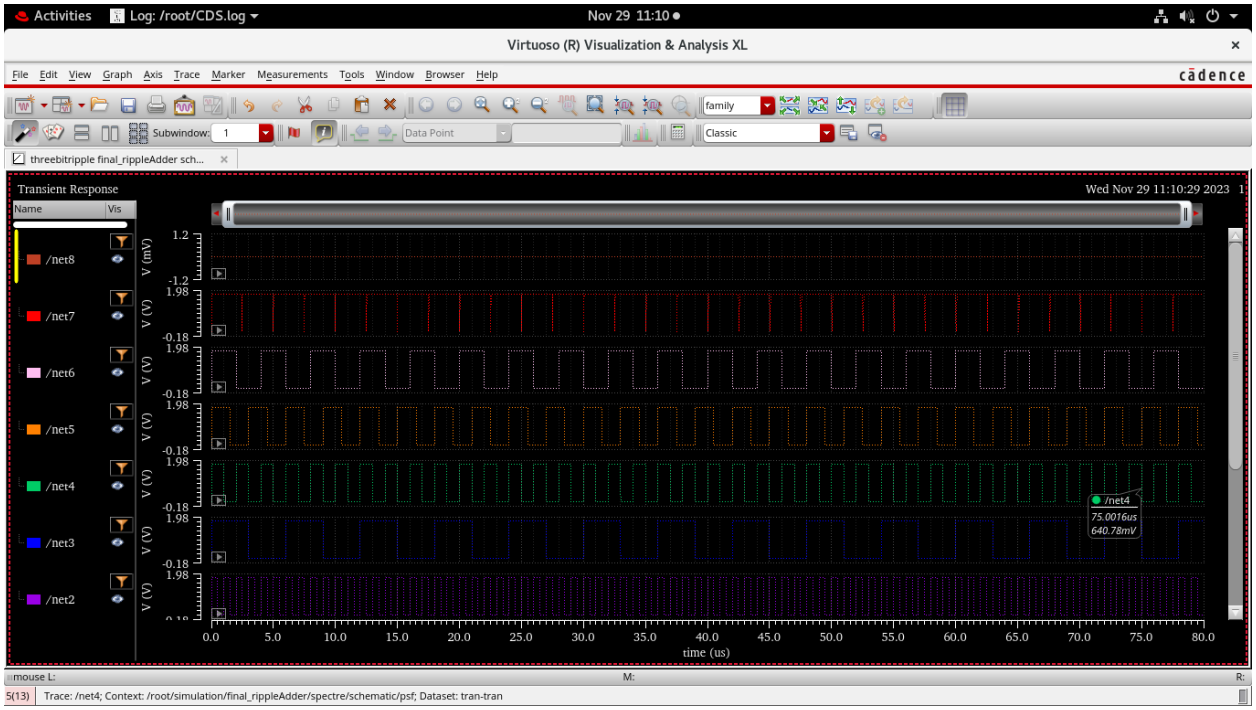


:Symbol

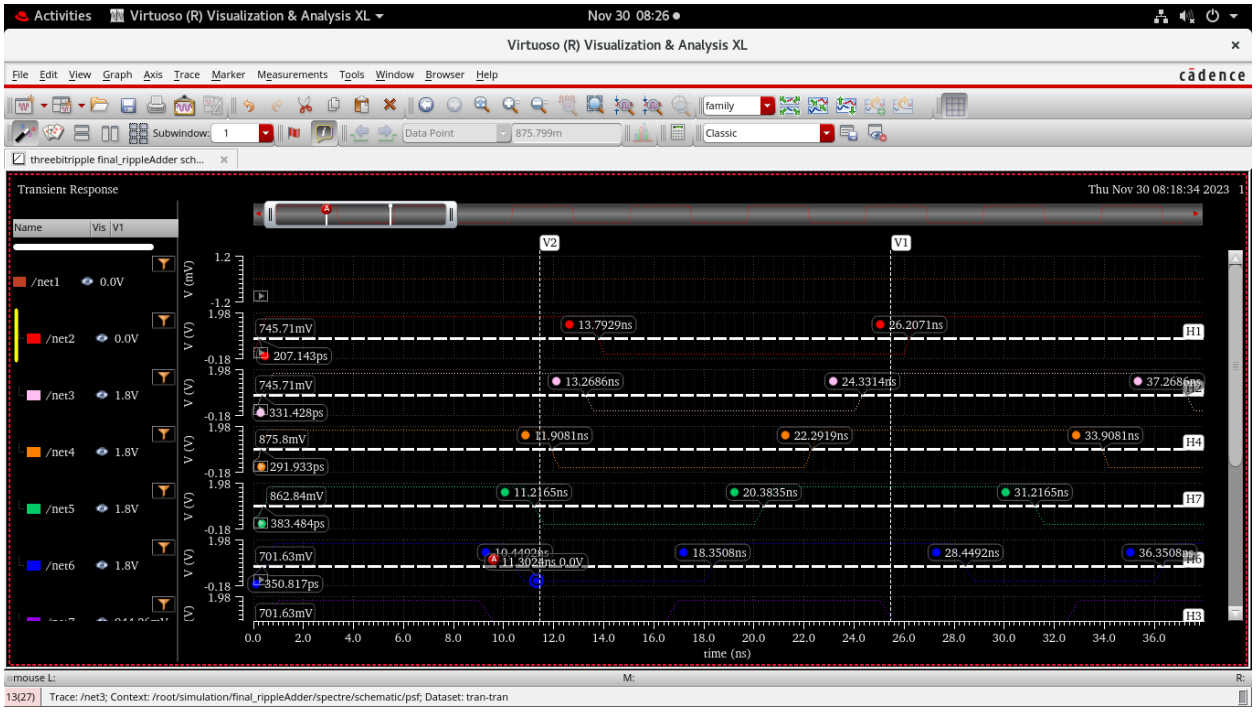


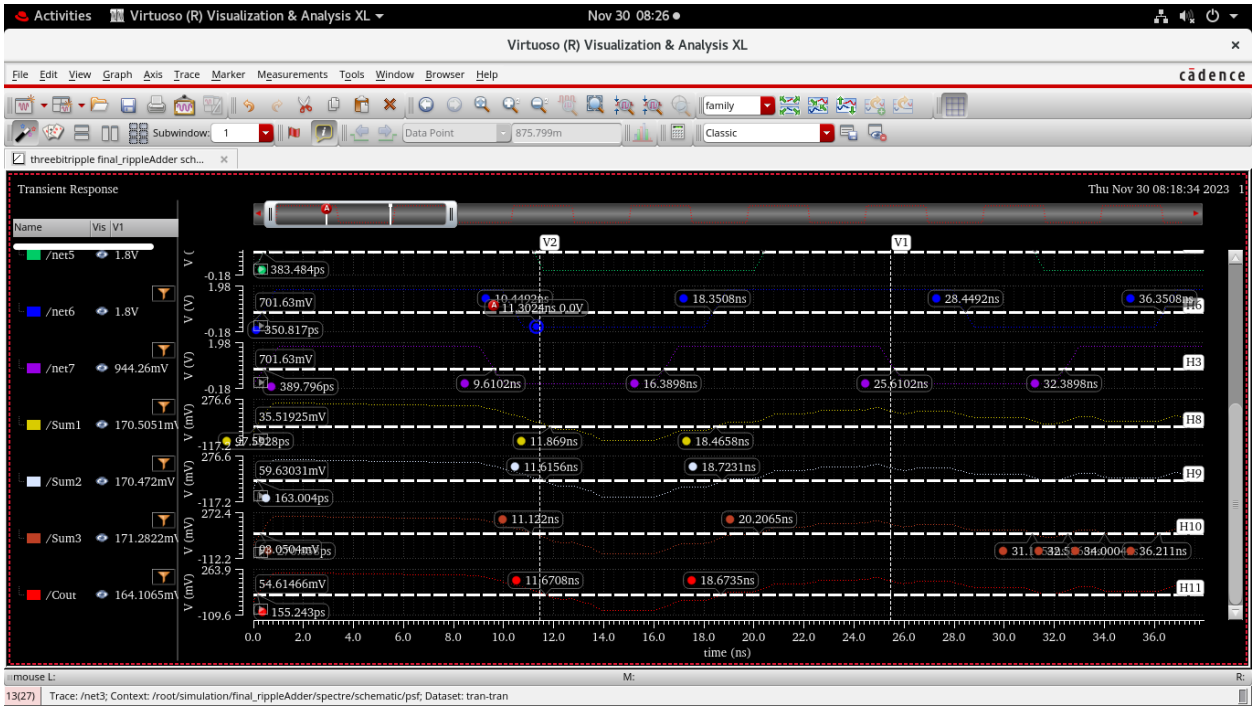
Transient response



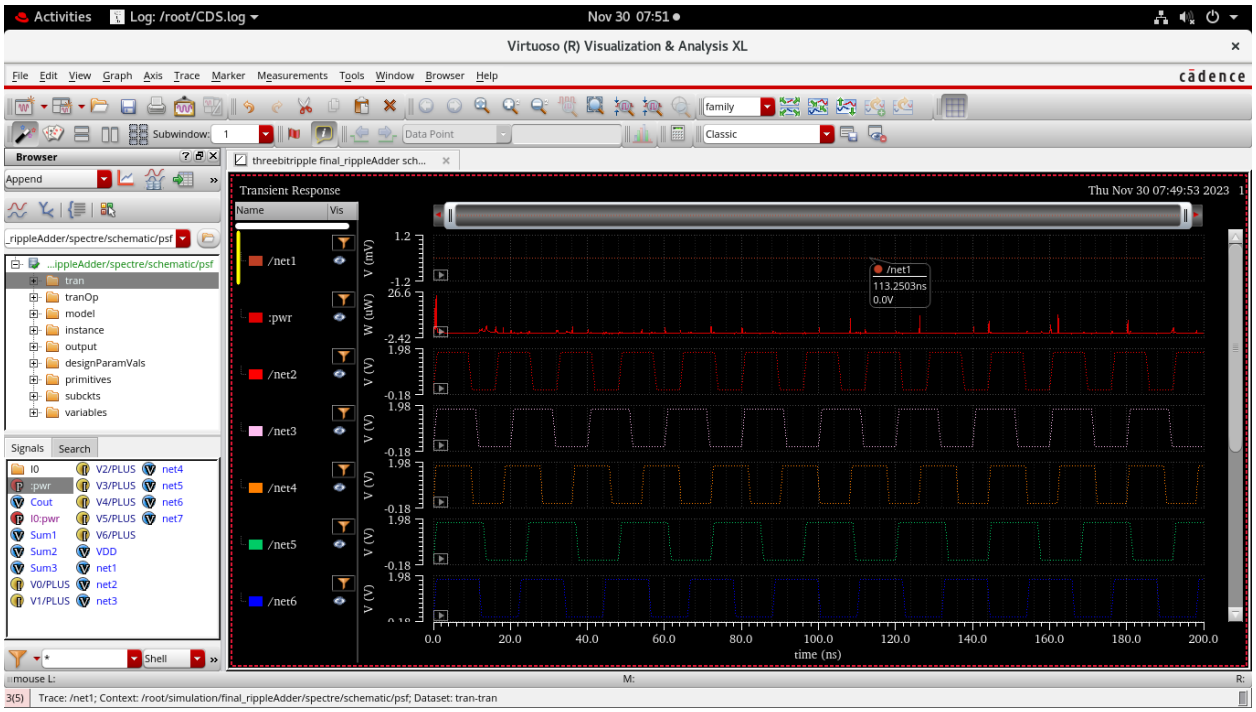


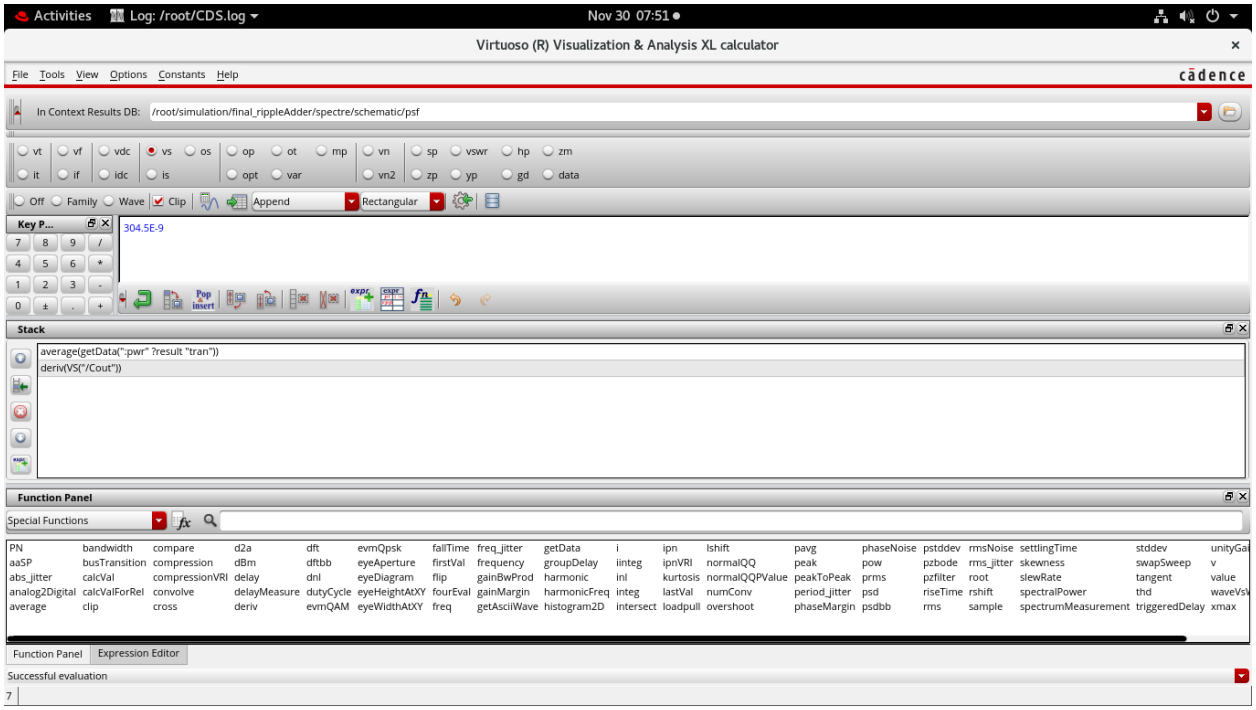
Propagation delay



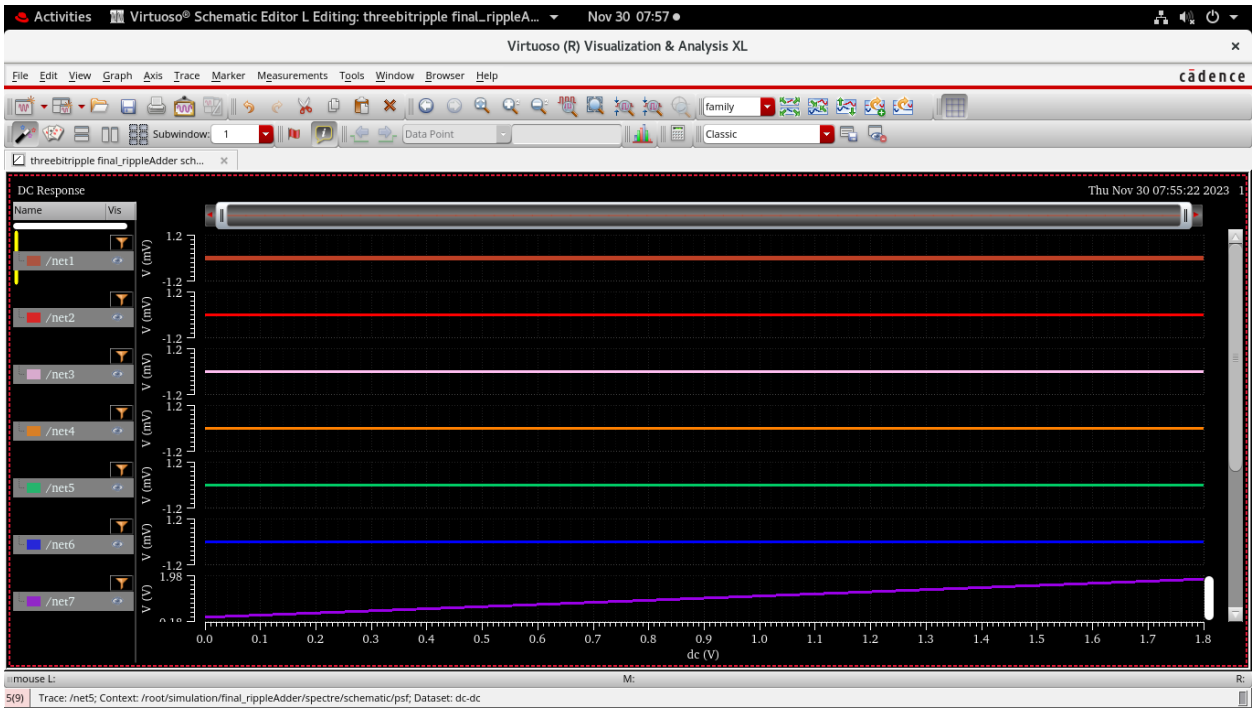


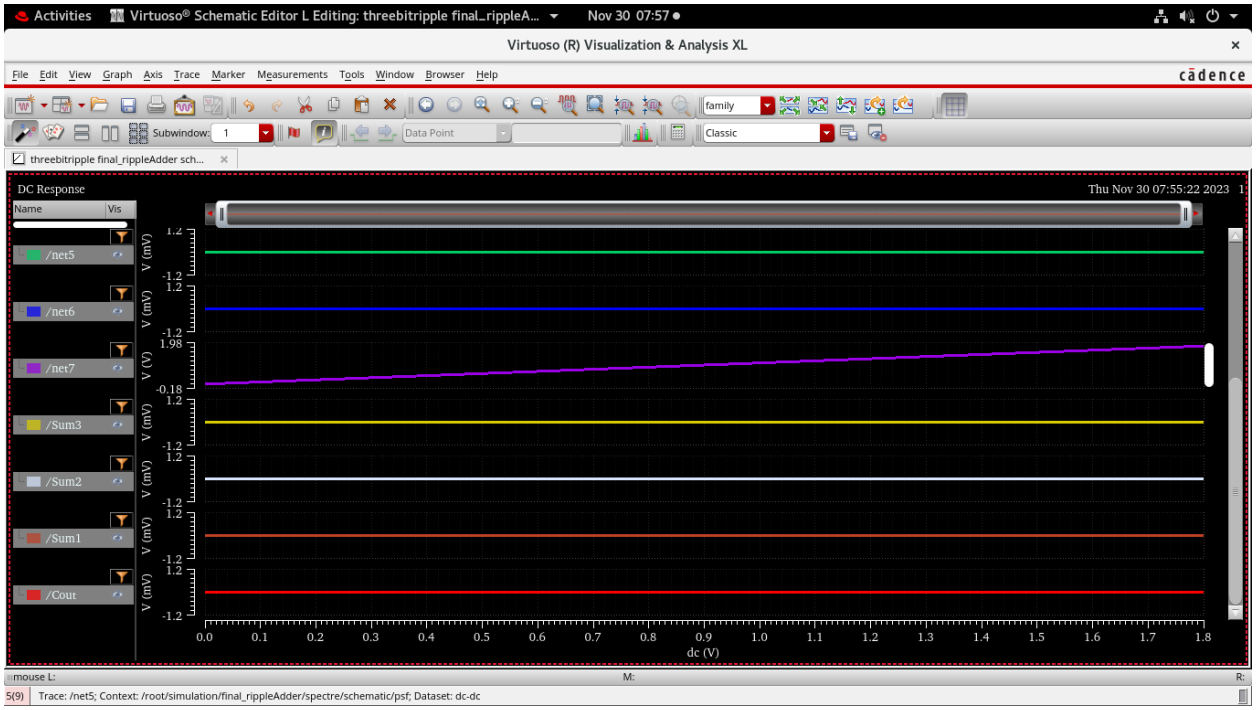
Power dissipation

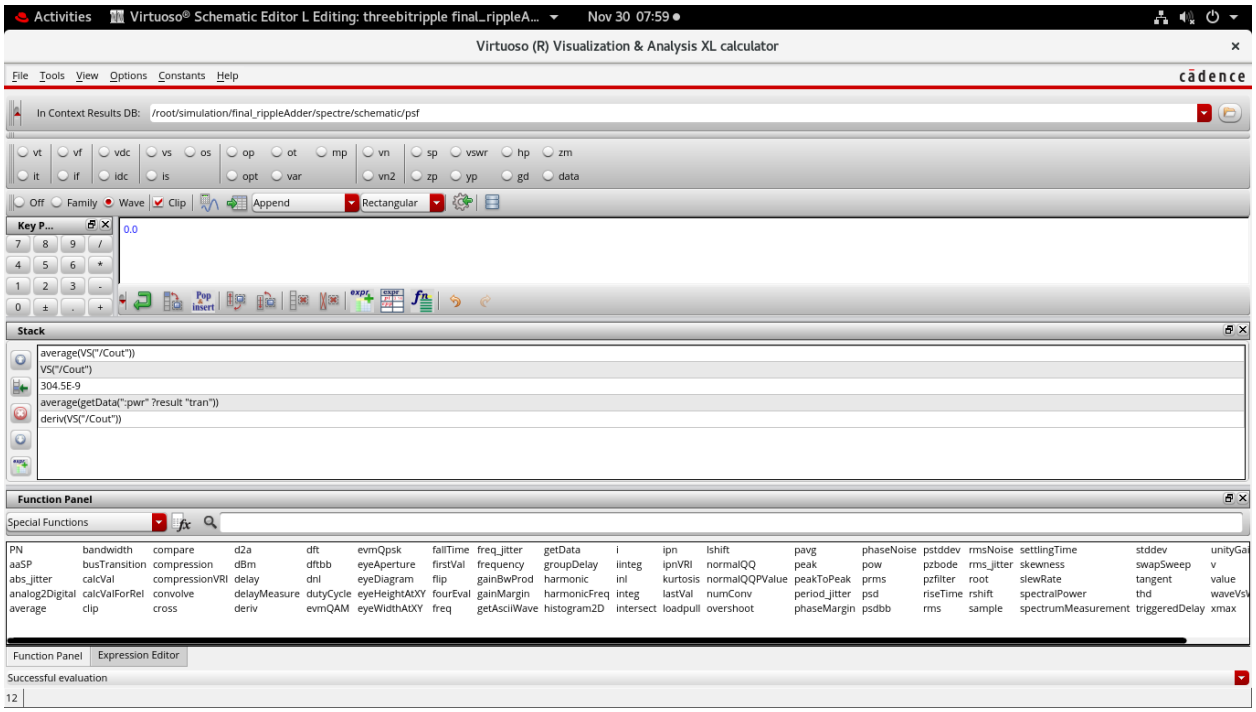



Activities
Log: /root/CDS.log
Nov 30 07:51
Virtuoso (R) Visualization & Analysis XL calculator
File Tools View Options Constants Help
cadence
In Context Results DB: /root/simulation/final_rippleAdder/spectre/schematic/psf
304.5E-9
Stack
average(getData(":pwr" ?result "tran"))
deriv(VS("/Cout"))
Function Panel
Special Functions
Function Panel
Expression Editor
Successful evaluation

Dc analysis


Activities
Virtuoso® Schematic Editor L Editing: threebitripple final_rippleA...
Nov 30 07:57
Virtuoso (R) Visualization & Analysis XL
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
cadence
threebitripple final_rippleAdder sch...
DC Response
Thu Nov 30 07:55:22 2023
Name
Vis
/net1
/net2
/net3
/net4
/net5
/net6
/net7
V (mV)
V (V)
dc (V)
mouse L:
M:
R:
Trace: /net5; Context: /root/simulation/final_rippleAdder/spectre/schematic/psf; Dataset: dc-dc


Activities
Virtuoso® Schematic Editor L Editing: threebitripple final_rippleA...
Nov 30 07:57
Virtuoso (R) Visualization & Analysis XL
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
cadence
DC Response
Thu Nov 30 07:55:22 2023
/net5
/net6
/net7
/Sum3
/Sum2
/Sum1
/Cout
V (mV)
V (V)
dc (V)
Trace: /net5; Context: /root/simulation/final_rippleAdder/spectre/schematic/psf; Dataset: dc-dc


Activities
Virtuoso® Schematic Editor L Editing: threebitripple final_rippleA...
Nov 30 07:59
Virtuoso (R) Visualization & Analysis XL calculator
File Tools View Options Constants Help
cadence
In Context Results DB: /root/simulation/final_rippleAdder/spectre/schematic/psf
Stack
average(VS("/Cout"))
VS("/Cout")
304.5E-9
average(getData(":pwr" ?result "tran"))
deriv(VS("/Cout"))
Function Panel
Special Functions
Function Panel
Expression Editor
Successful evaluation