Vietnam National University Ho Chi Minh City

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

LAB 3: DESIGN OF COMBINATIONAL AND SEQUENTIAL CIRCUITS

Subject: Digital IC Design

Class L03 --- Semester 242

Group: 12

Student	Student ID	Mark
Ho Xuan Tuan Tai	2212970	
Chau Ngoc Thach	2014526	
Mai Gia Huy	2113493	

Lecturer: PhD. Tran Hoang Linh

Ho Chi Minh City, 03-2025

Contents

1. EXPERIMENT 1 3

1.1 Truth table, schematic and symbol for transient simulation: 3

1.2 Layout: 6

1.3 Transient Analysis Simulation: 9

2. EXPERIMENT 2 12

2.1. The schematic, symbol and waveform of PRBS: 12

2.2. PRBS's Clock Frequency: 18

1. EXPERIMENT 1

Objective: Design a combinational circuit – a 1-bit Full Adder

Requirements:

➢ Complete the truth table, schematic, and symbol for each component then run transient simulation.

➢ Define the speed of your design.

➢ Create layouts for each logic gate, then show DRC confirmation and corresponding schematic with proof of LVS.

1.1 Truth table, schematic and symbol for transient simulation:

This lab gives information about how to clarify the complexities of 1-bit full adder by examining many topologies. Exploring the option that is simplicity, balances and efficiency. Moreover, we consider the vital aspects like power consumption and delay limitation.

A	B	CIN	COUT	SUM
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

Table 3.1.1: Truth table of a full adder

The first circuit that we will explore about is the 1-bit full adder built from logic gates and the testbench for it:

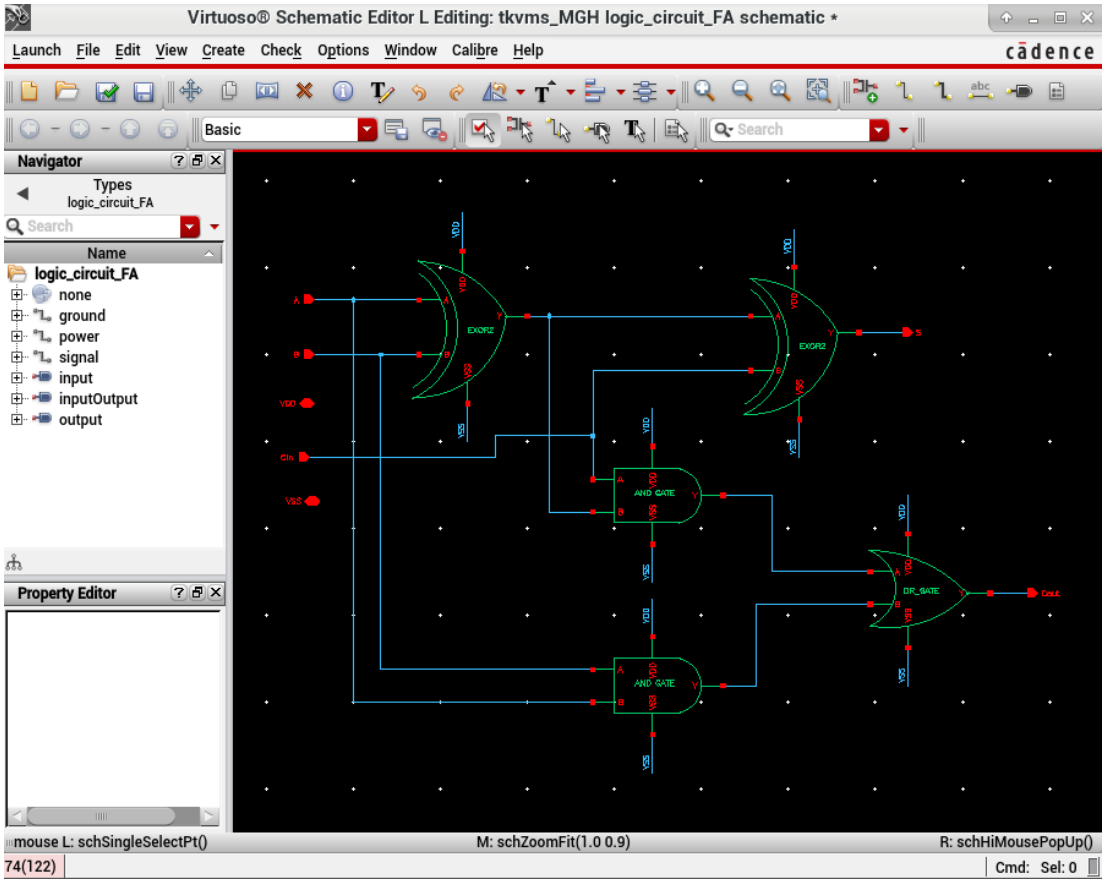


Figure 3.1.1: Schematic of logic circuit 1-bit full adder

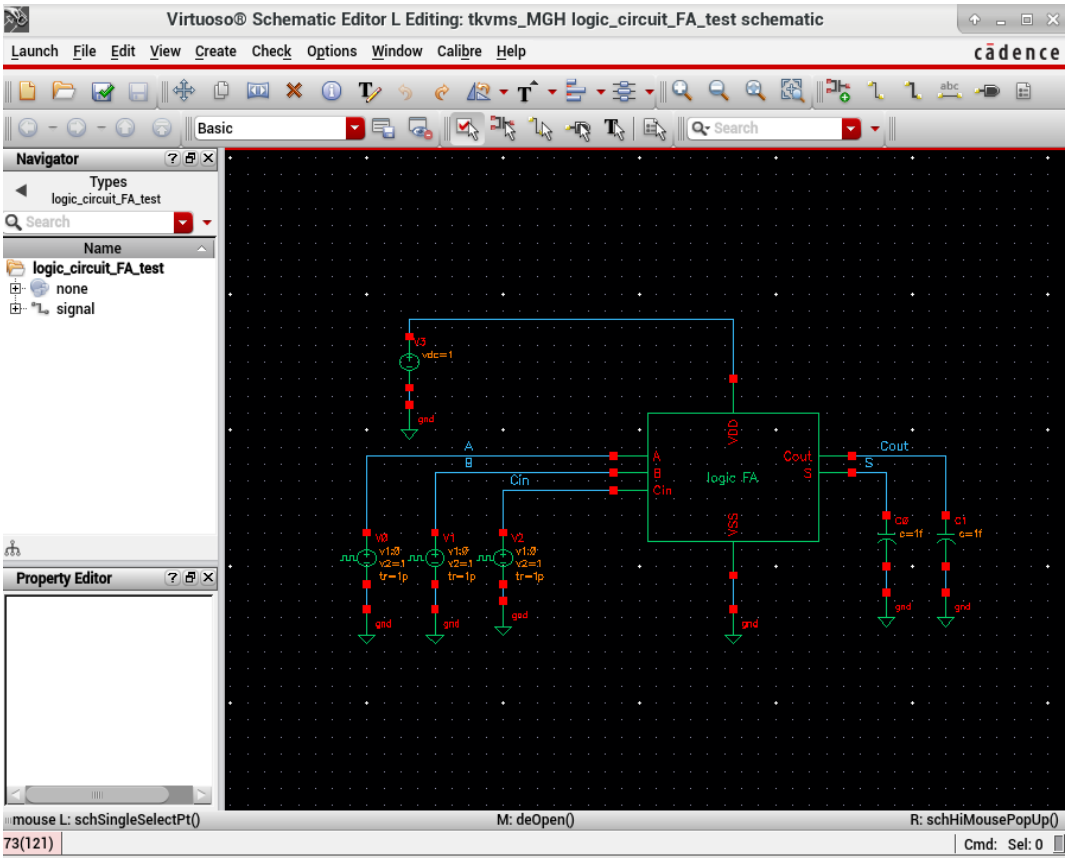


Figure 3.1.2: Testbench of logic circuit 1-bit full adder

As we can see from the schematic of logic circuit, this topology needs 46 transistors to implement. Therefore, the area of it is large, so the solution that can be used to minimize the number of transistors is using 28T Full Adder circuit.

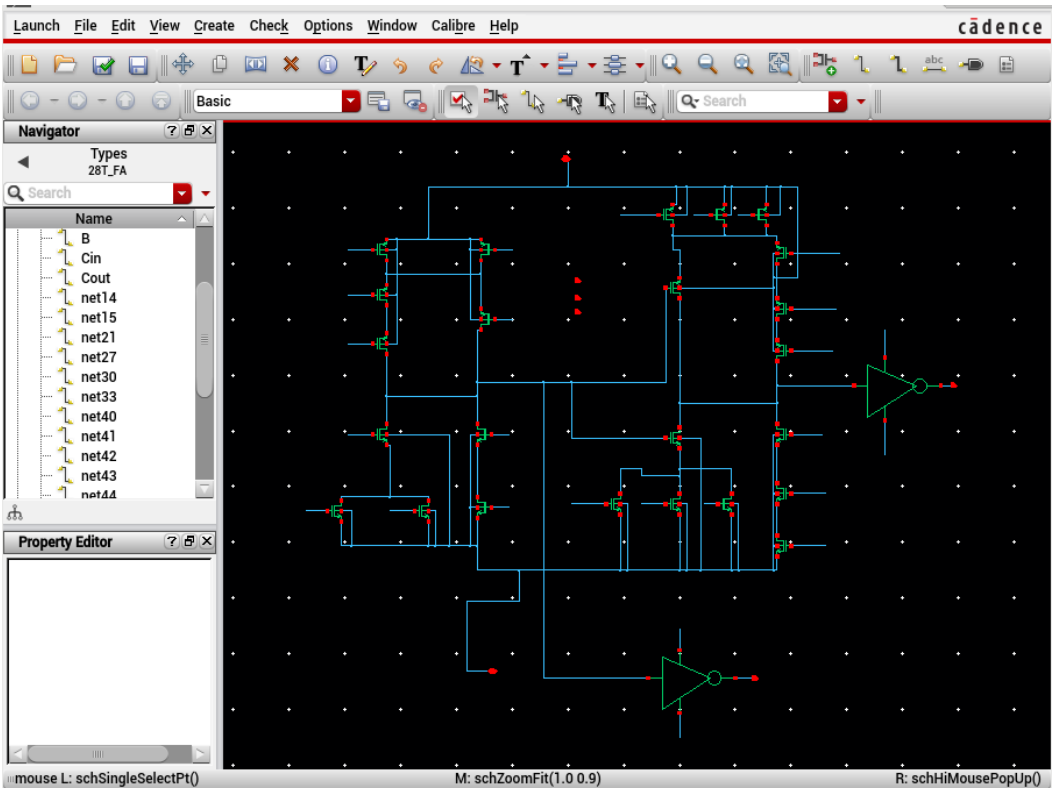


Figure 3.1.3: Schematic of 28T 1-bit full adder

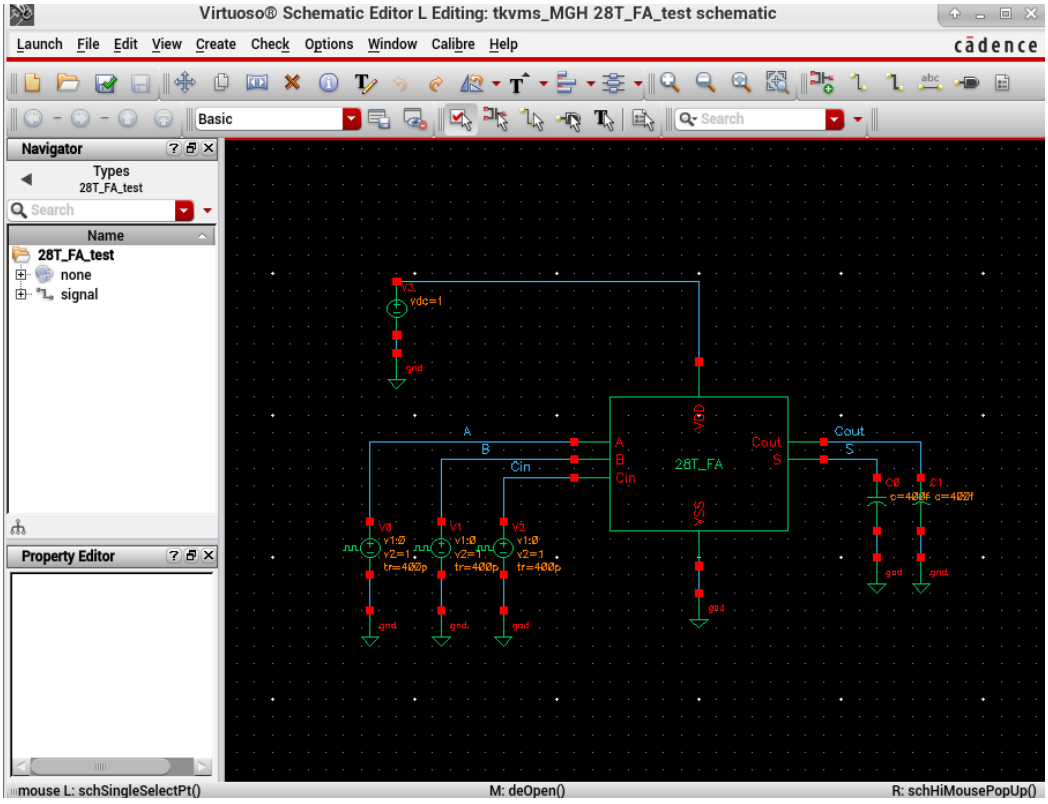


Figure 3.1.4: Testbench for 28T Full Adder

1.2 Layout:

The next thing that our group will introduce is about the layout of two topology above and show the checking result using DRC and LVS:

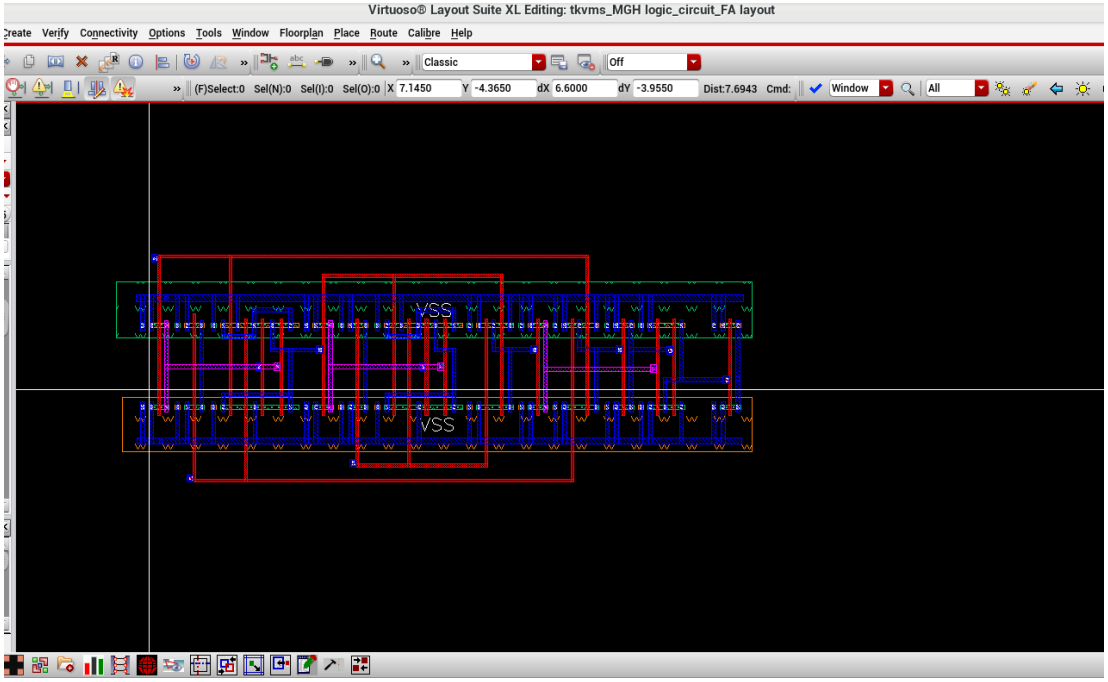


Figure 3.1.5: Layout of logic circuit Full Adder

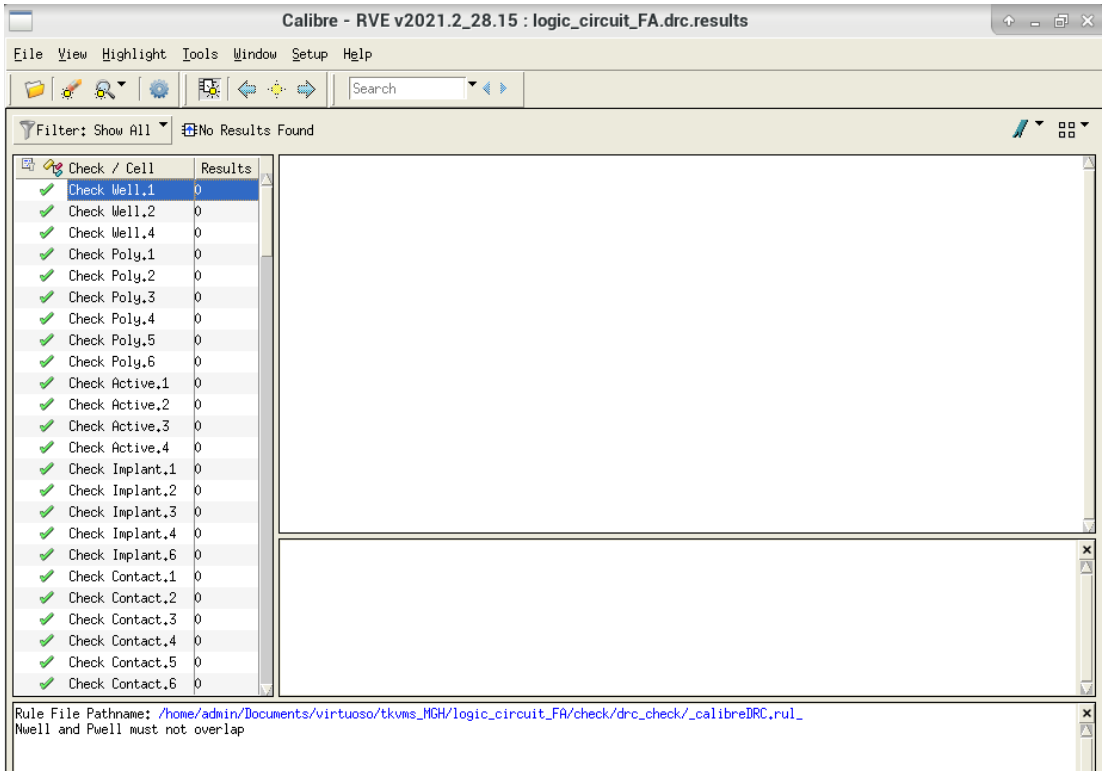


Figure 3.1.6: Result of DRC check

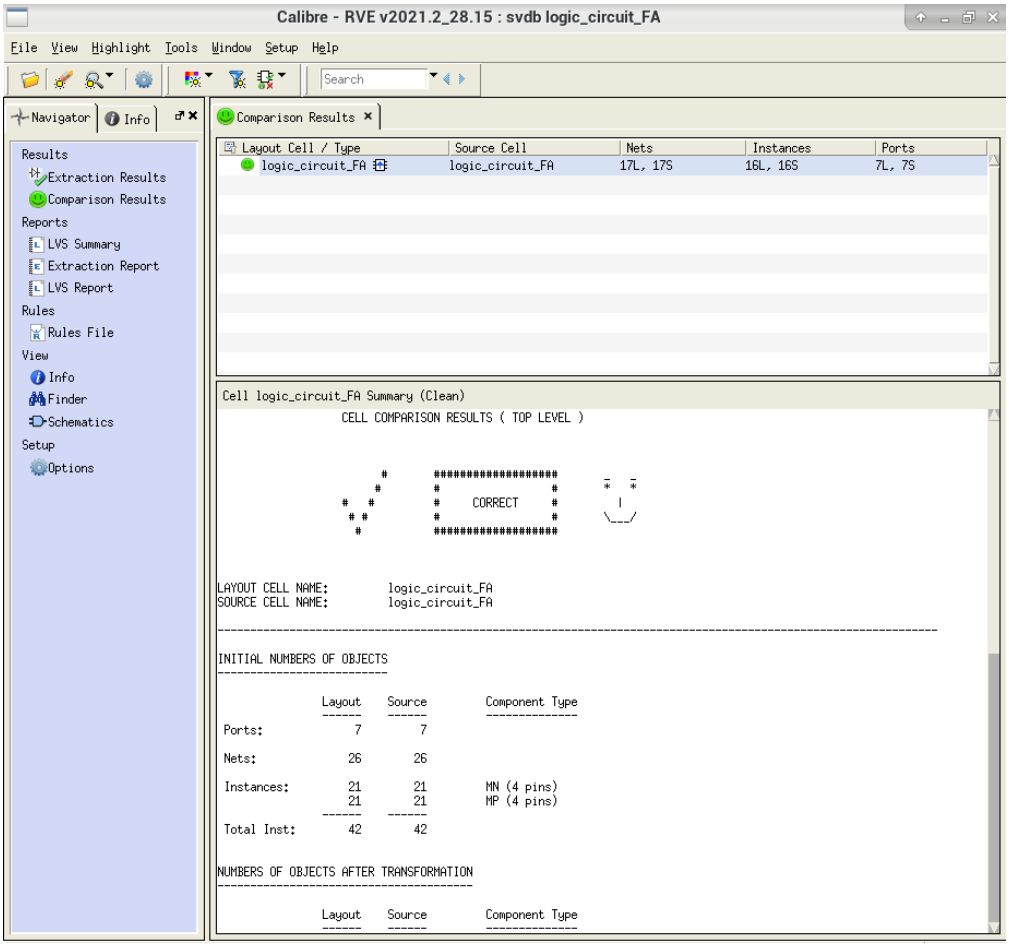


Figure 3.1.7: Result of LVS check

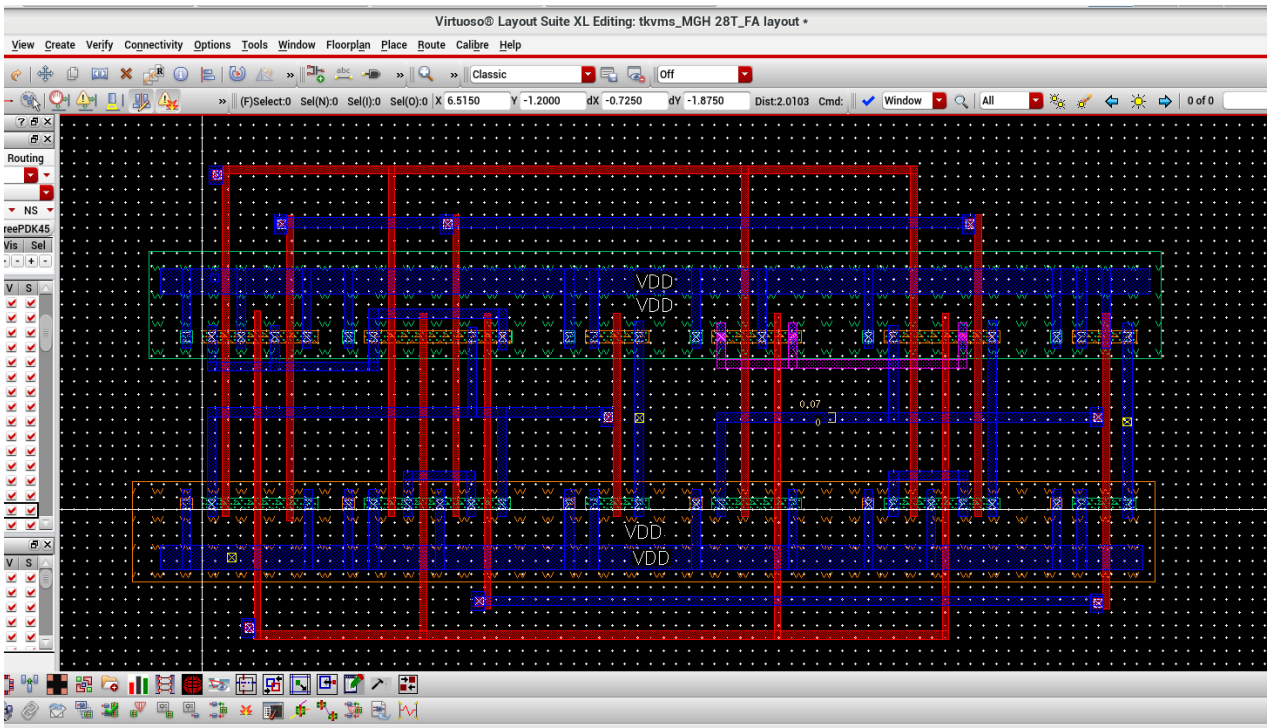


Figure 3.1.8: Layout of 28T Full Adder

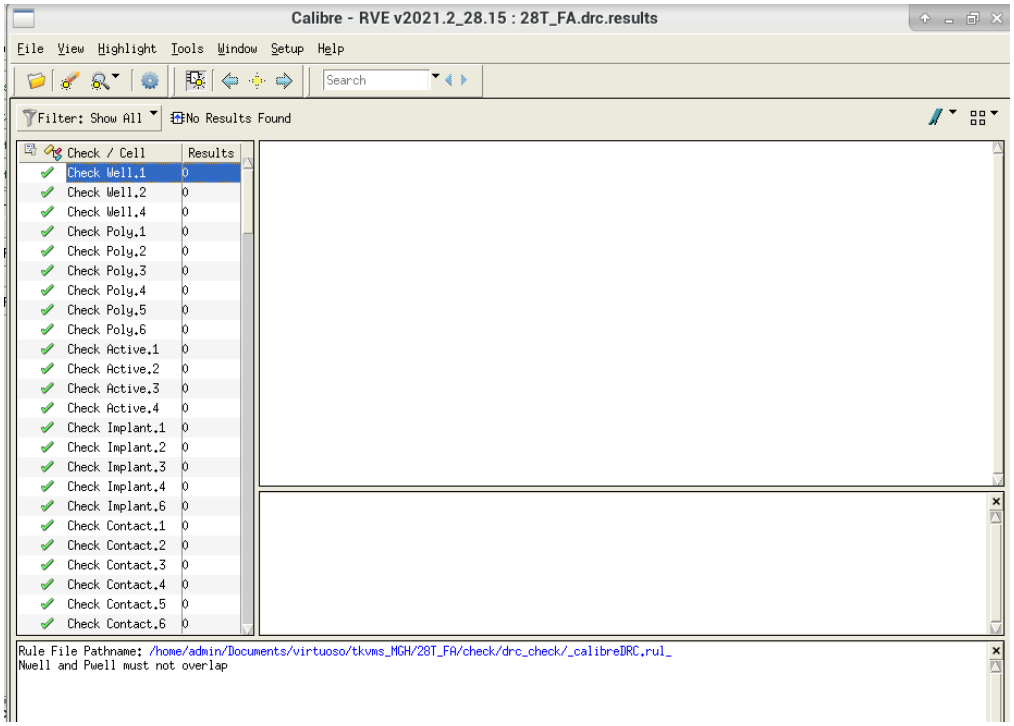


Figure 3.1.9: Result of DRC check

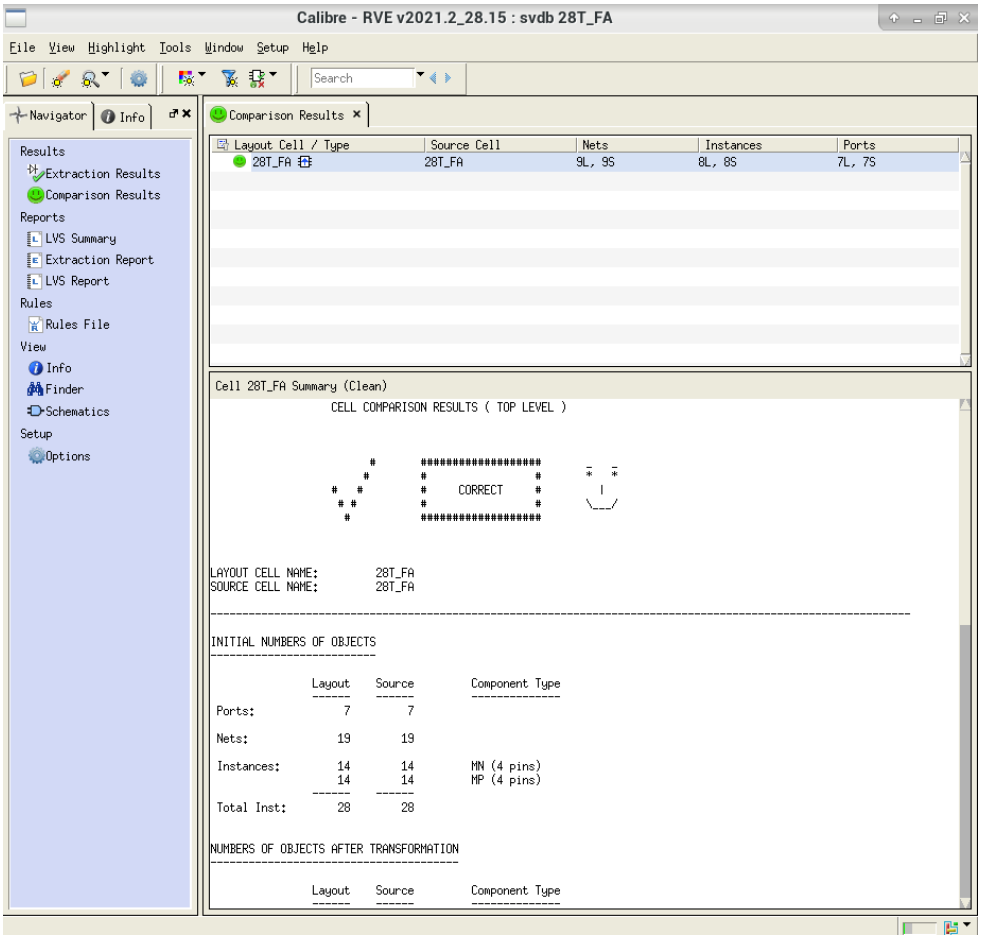


Figure 3.1.10: Result of LVS check

1.3 Transient Analysis Simulation:

A 28T (Twenty-Eight Transistor) Full Adder includes dynamic modeling and analysis of the adder's behavior during operation. This full adder architecture typically consists of multiple stages of logic gates, including XOR and AND gates, implemented using CMOS (Complementary Metal-Oxide-Semiconductor) technology. In the transient analysis simulation, various input scenarios representing different binary inputs (A, B, and carry-in) are applied to the full adder, and the transient response of the output signals (sum and carry-out) is observed over time. The simulation helps validate the correctness and functionality of the full adder design, assessing its performance metrics such as propagation delay, power consumption, and signal integrity. Additionally, transient simulations allow designers to optimize the full adder's architecture and transistor sizing for speed, area efficiency, and power consumption, ensuring it meets the desired specifications for arithmetic and logic operations in digital circuits.

	A	B	Cin
Voltage 1	0	0	0
Voltage 2	1	1	1
Period	4n	2n	2n
Delay time	0.65n	0.8n	0.4n
Rise time	1p	1p	1p
Fall time	1p	1p	1p
Pulse width	2n	1n	0.5n

Table 3.1.2: Parameters for transient simulation

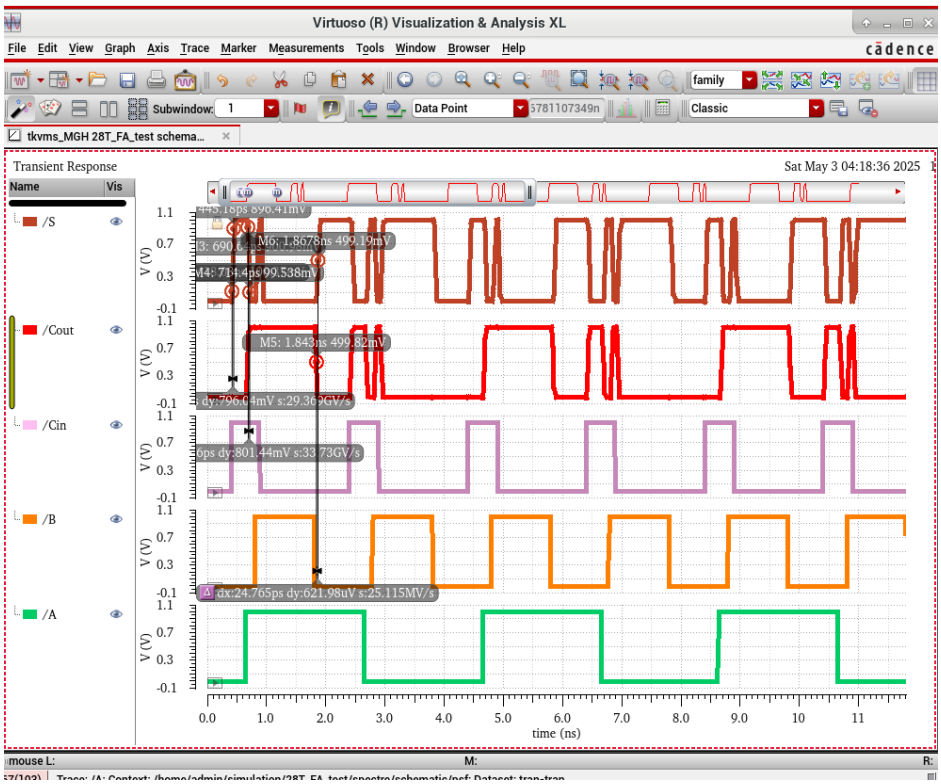


Figure 3.1.11: Transisent result of 28T Full Adder

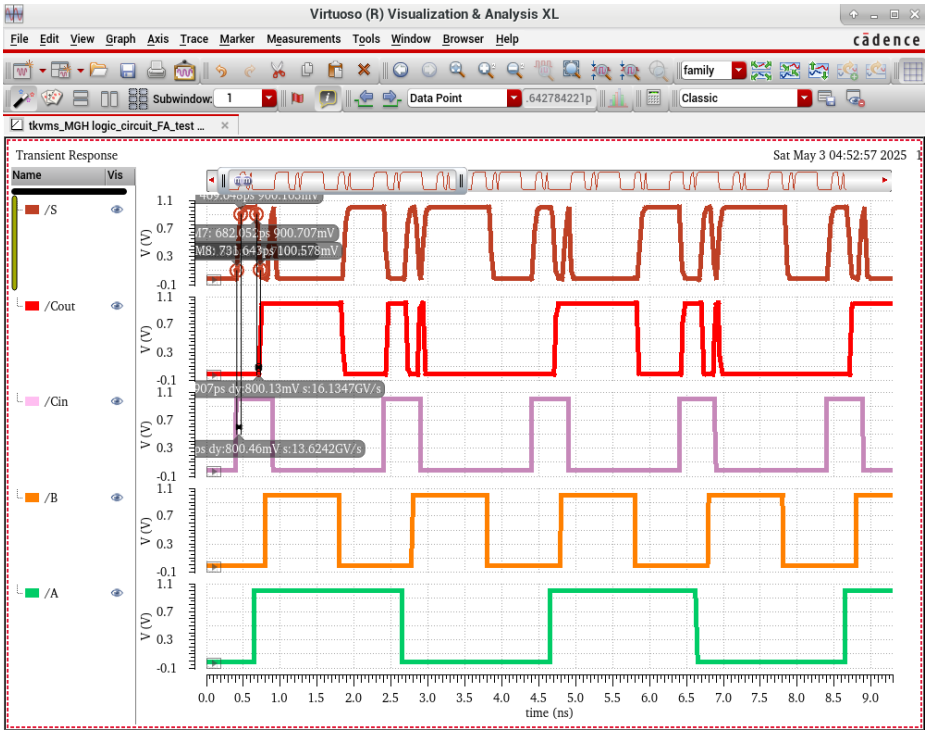


Figure 3.1.12: Transisent result of logic ciruit Full Adder

Parameter	Original FA (Figure 2)	FA 28T (Figure 3)
t_{rise} - Rising time (10%-90%)	58.75 ps	27.11 ps
t_{fall} – Falling time (90% - 10%)	49.64 ps	23.76 ps
t_{pd} – Average propagation delay (50% - 50%)	27.92 ps	24.76 ps
Power consumption	37.48 uW	39.85 uW

Table 3.1.3: Measurement requirements for both Full Adder

Question: Due to the topology shown in Figure 3, why do people implement the PDN of the first stage using equation Cout = (A + B)Cin + AB instead of Cout = AB + ACin + BCin?

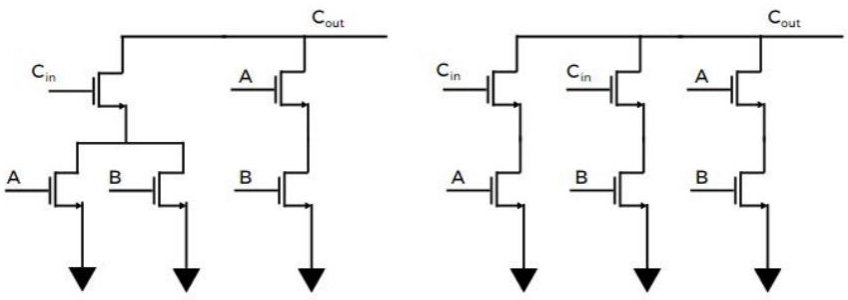


Figure 3.4 : The PDN using $C_{out} = (A+B) * C_{in} + AB$ *instead of* $C_{out} = AB + AC_{in} + BC_{in}$

In the design of CMOS full adders, particularly for the carry-out logic, the implementation of the Pull-Down Network (PDN) plays a crucial role in determining the overall efficiency in terms of area, speed, and power consumption.

Figure 3.4 illustrates two different implementations of the PDN based on these expressions. The implementation corresponding to Cout = (A+B)Cin + AB results in a PDN composed of two main branches. These two branches are then connected in parallel to form the final PDN. This structure inherently limits the maximum number of transistors stacked in series to two, which is beneficial for reducing both resistance and delay.

In contrast, the alternative expression Cout = AB + ACin + BCin requires the construction of three distinct branches, each with different series combinations. Consequently, this approach increases the number of required transistors and can result in more complex routing and larger silicon area, which may degrade performance. Two figures below illustrate the delay in 2 different conducted circuits. As we can see, figure 1 will create more capacitances which can extend the delay. Therefore, for optimizing the schematic people usually choose PDN using Cout = (A+B)Cin + AB rather than Cout = AB + ACin + BCin.

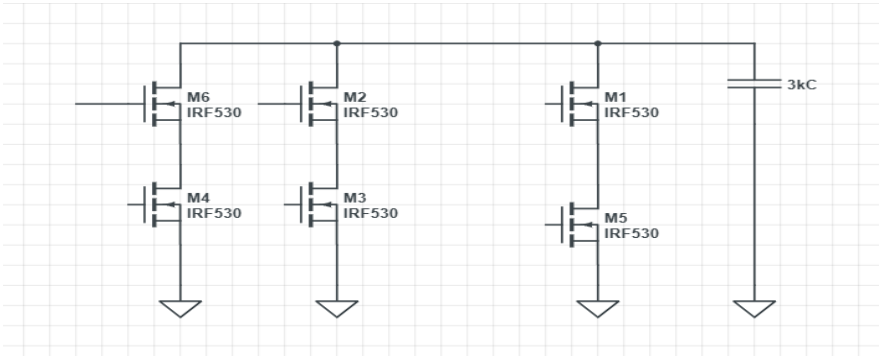


Figure 1: RC circuit of Cout = AB + ACin + BCin

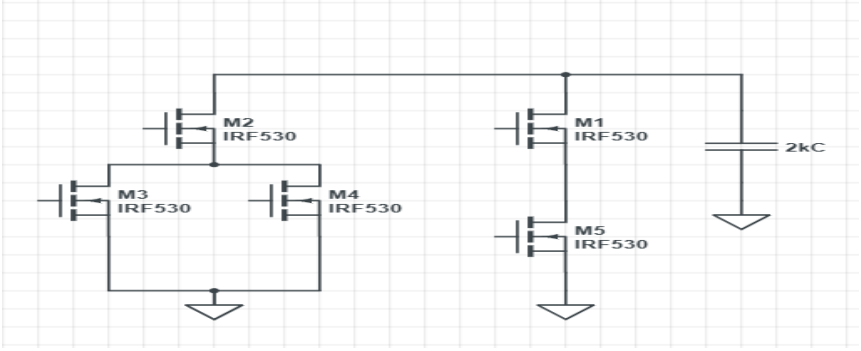


Figure 2: RC circuit of Cout = (A+B)Cin + AB

2. EXPERIMENT 2

Objective: Know how to design a basic sequential circuit.

Requirement: You will learn PRBS or pseudo-random binary sequence

A pseudo-random binary sequence (PRBS) which is a type of algorithm-generated random signal is often used as model data to test high-speed serial interface devices for emulating a mission mode. It is a mathematically randomized bit stream so that is well-neutralized and balanced data. A PRBS bit stream can be generated by using a **linear feedback shift register (LFSR)**.

2.1. The schematic, symbol and waveform of PRBS:

- Draw schematic of PRBS including: LFSR using positive-edge-triggered Modified TSPC DFF with Preset and XOR gate.

+ Schematic and symbol of **TSPC DFF**:

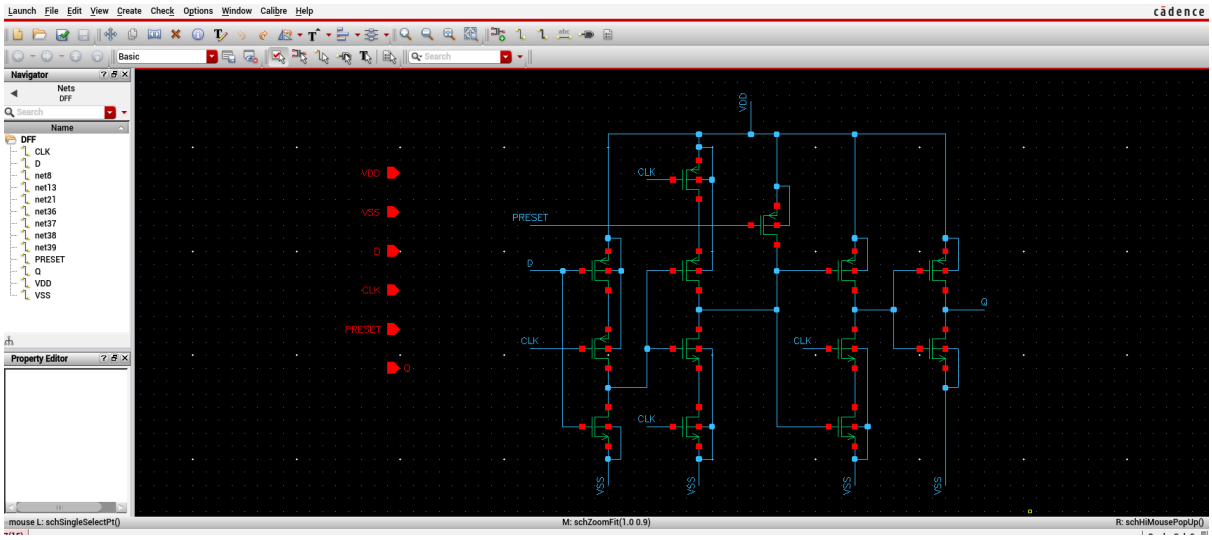


Figure 2.1: Schematic of TSPC DFF

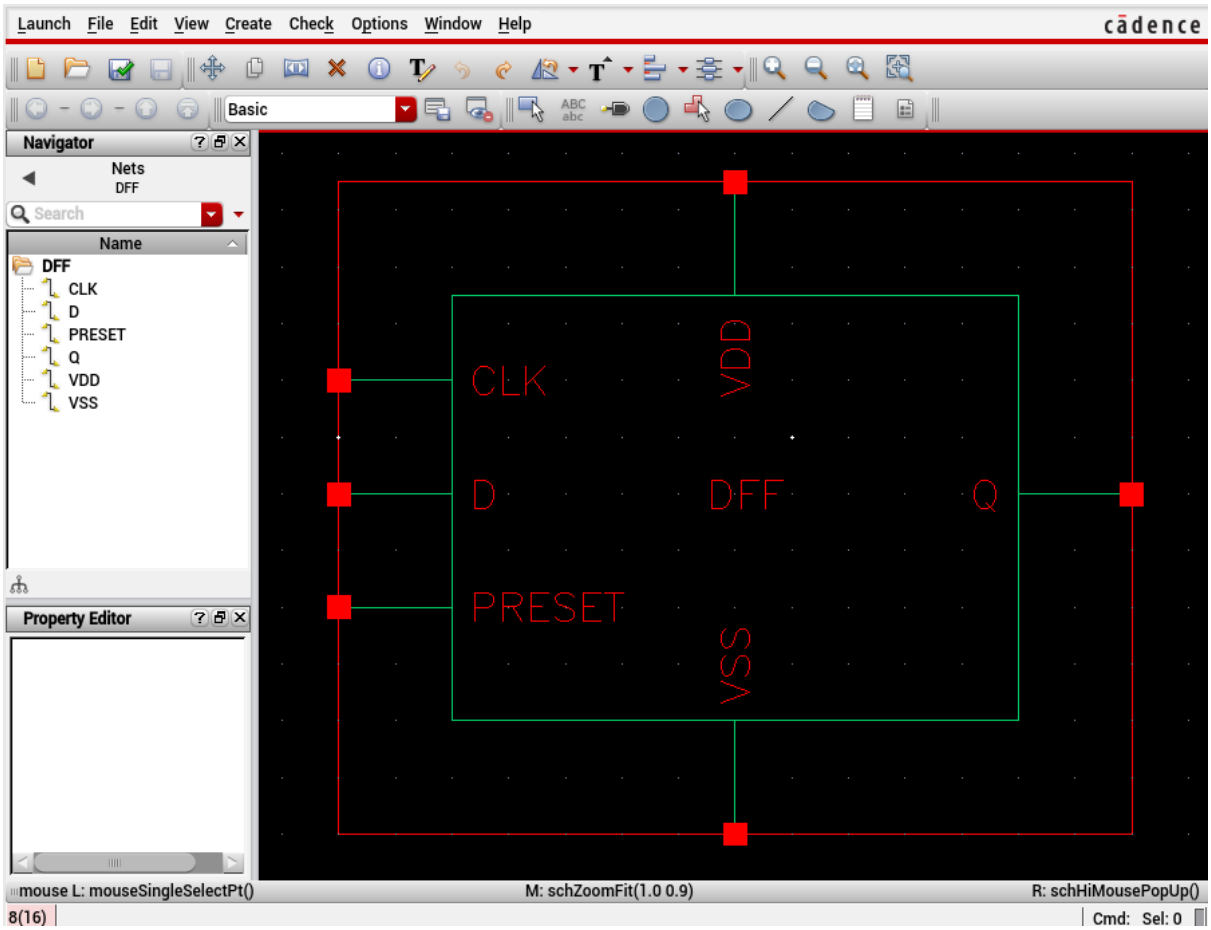


Figure 2.2: Symbol of TSPC DFF

+ Schematic and symbol of **XOR** gate:

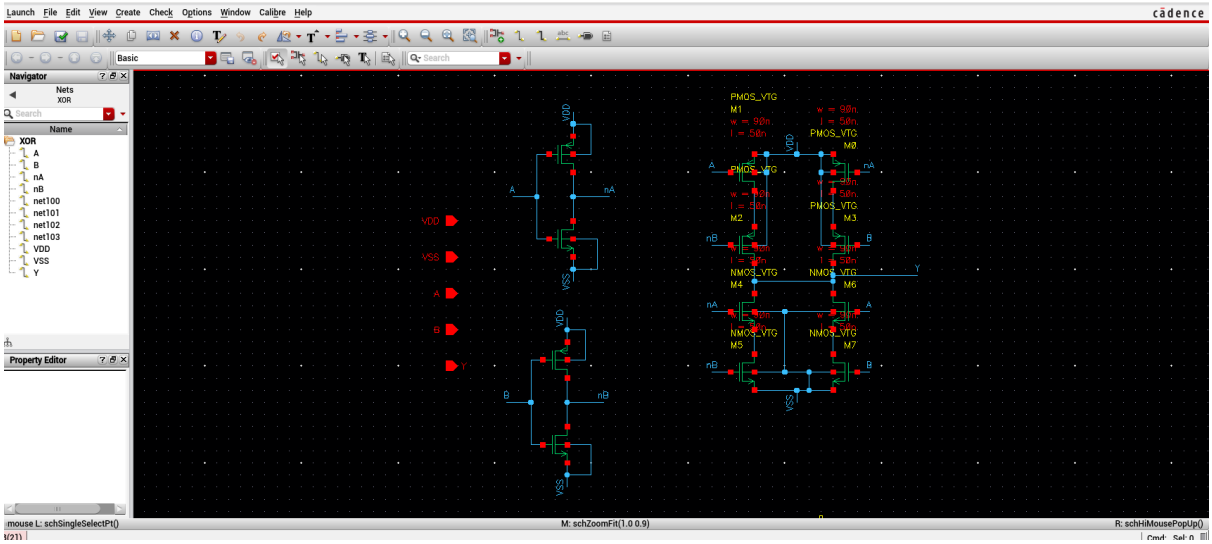


Figure 2.3: Schematic of XOR Gate

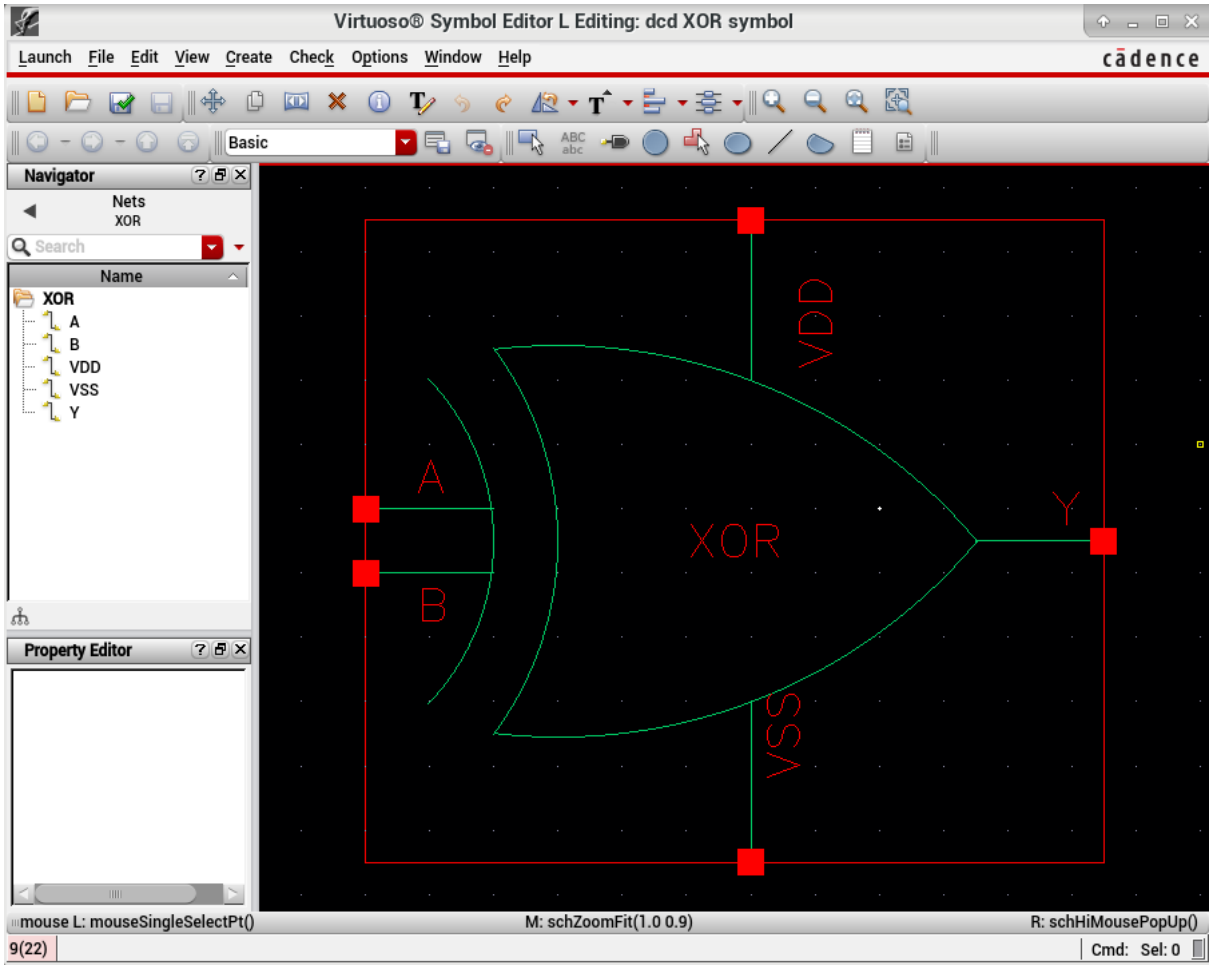


Figure 2.4: Symbol of XOR

+ Schematic of the Completed PRBS Circuit:

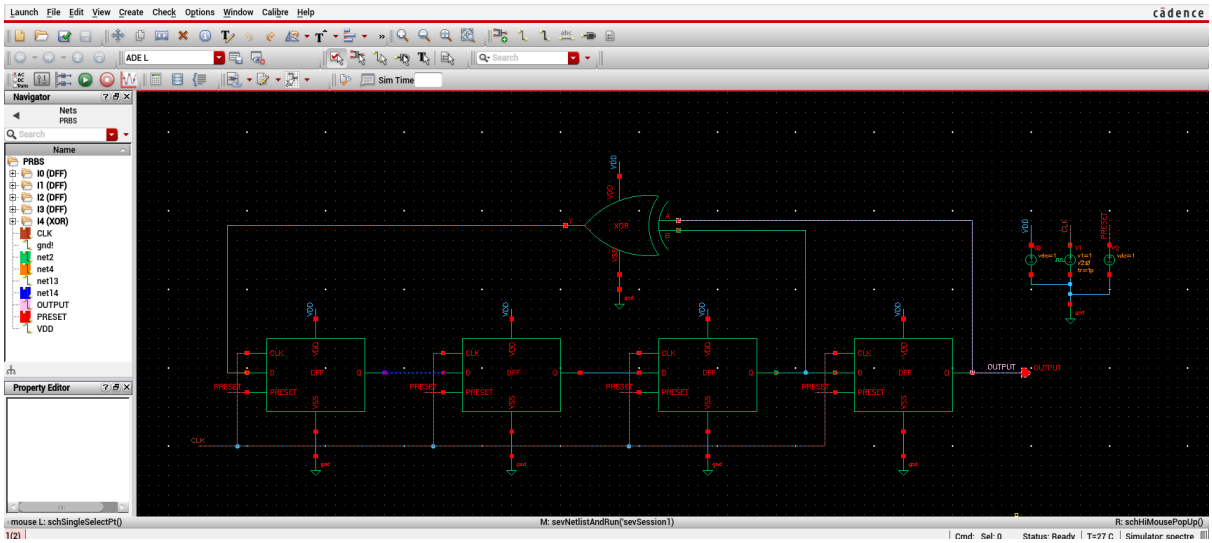


Figure 2.5: Schematic of PRBS

- Set the Clock (CLK) Signal:

Figure 2.6: Set the clock (CLK) signal

- Set up some initial conditions: Set the initial state to 1111.

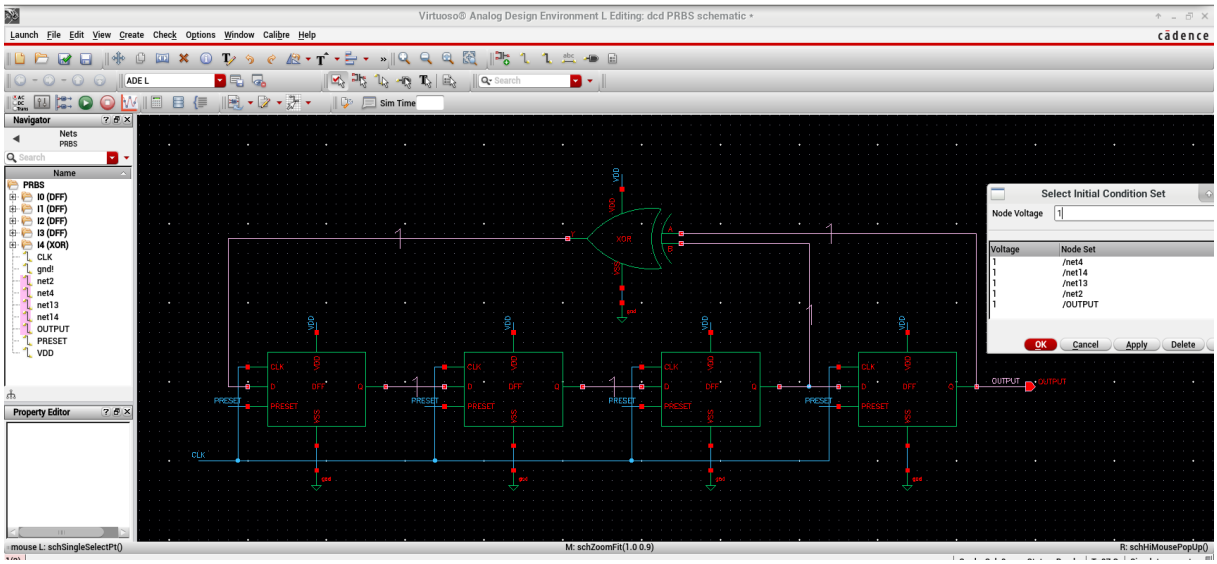


Figure 2.7: Initial Condition of PRBS

- Set parameters to simulate the waveform of PRBS:

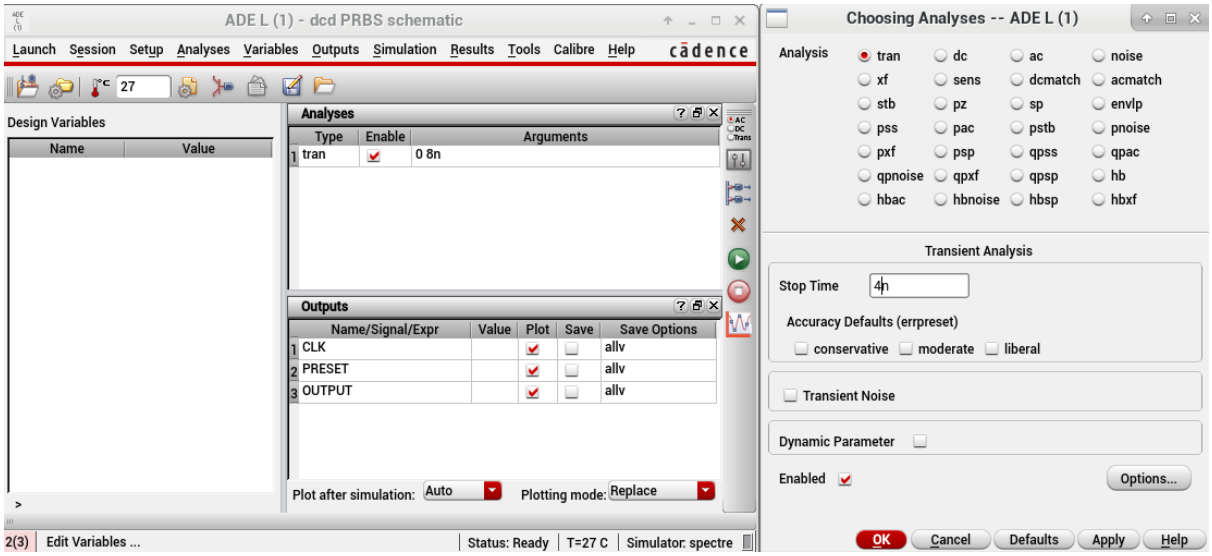
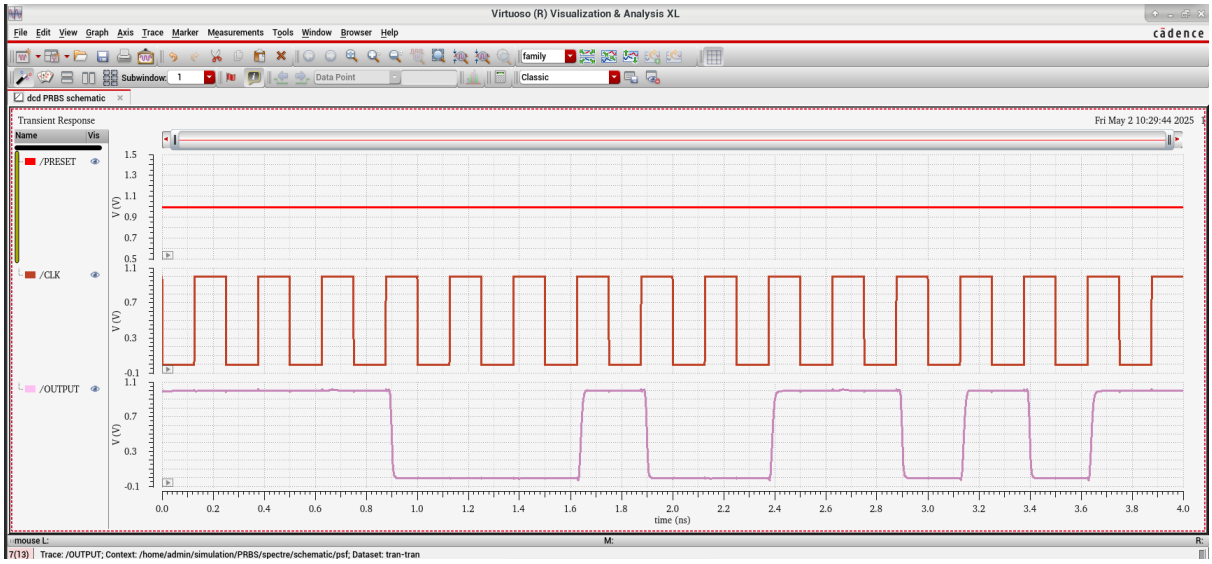


Figure 2.8: Set parameters for simulation



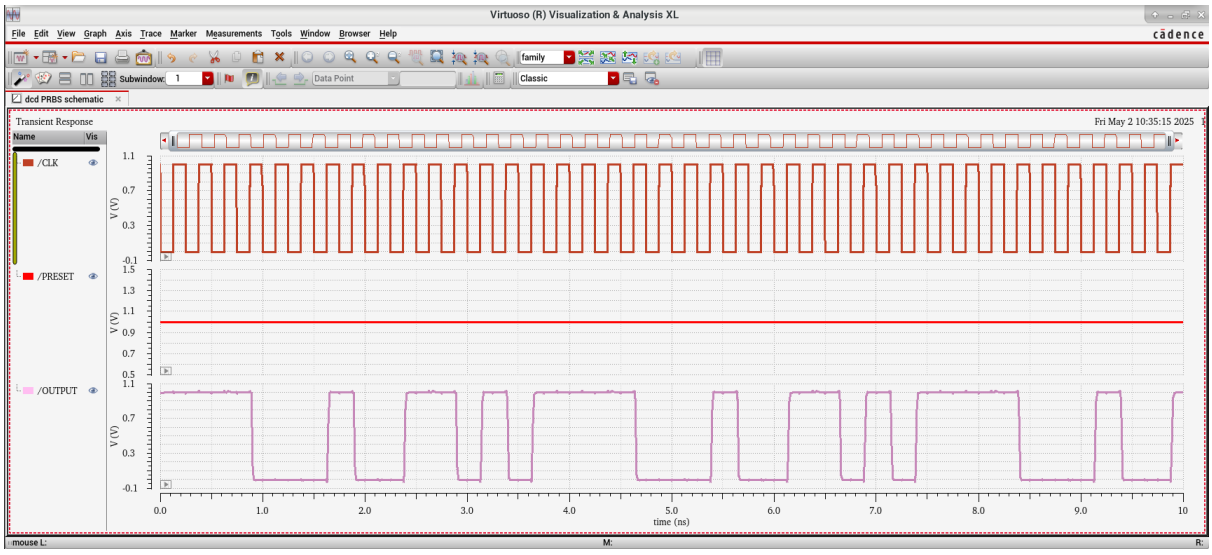


Figure 2.9+2.10: Result of PRBS's waveform

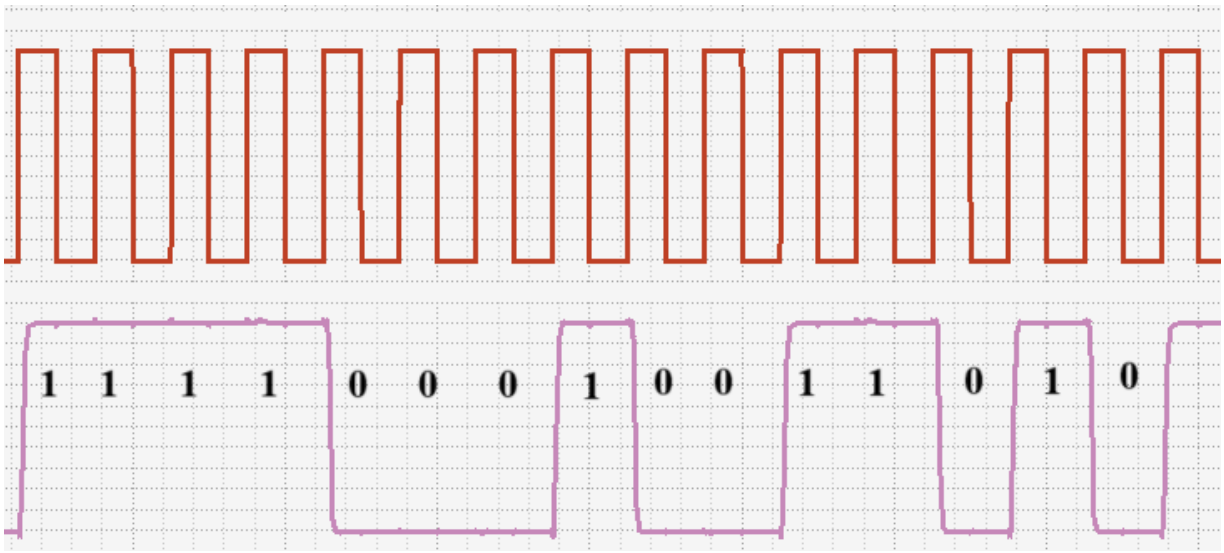


Figure 2.11: Bitstream of the PRBS

=> The waveform accurately reflects the functionality of the PRBS:

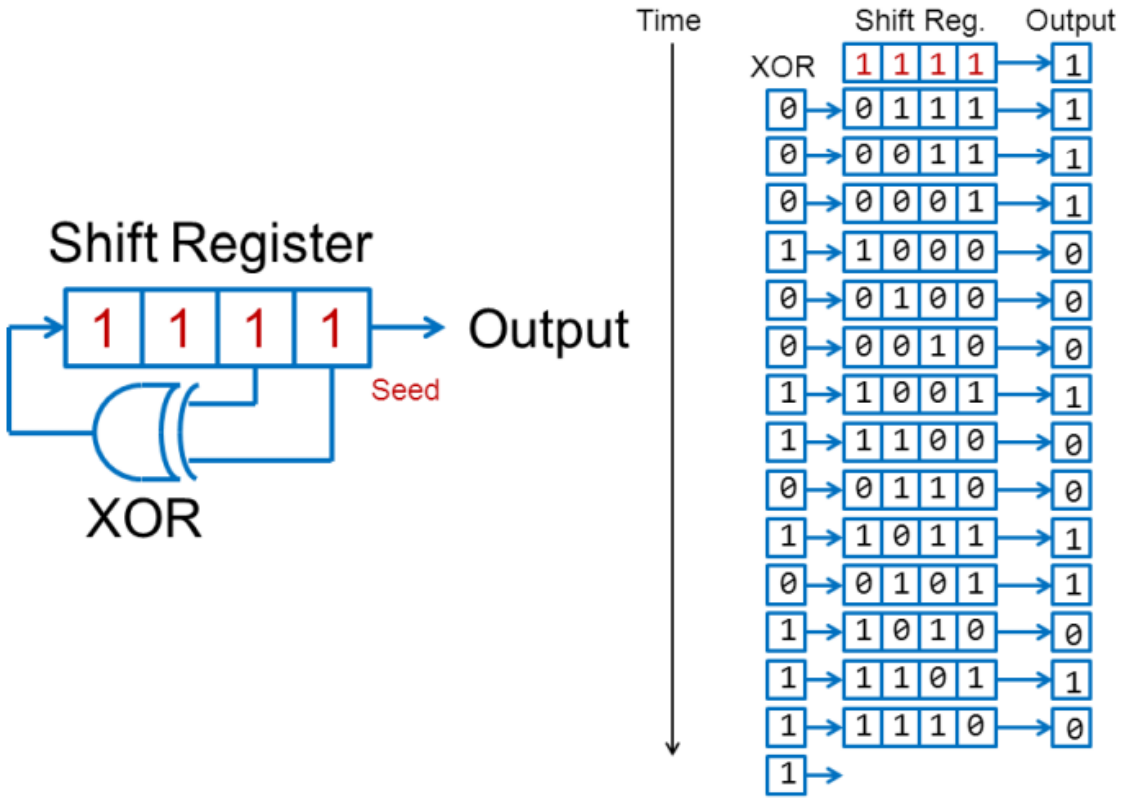


Figure 2.12: Linear Feedback Shift Register (LFSR)

- The LFSR waveform is generated according to theoretical expectations:

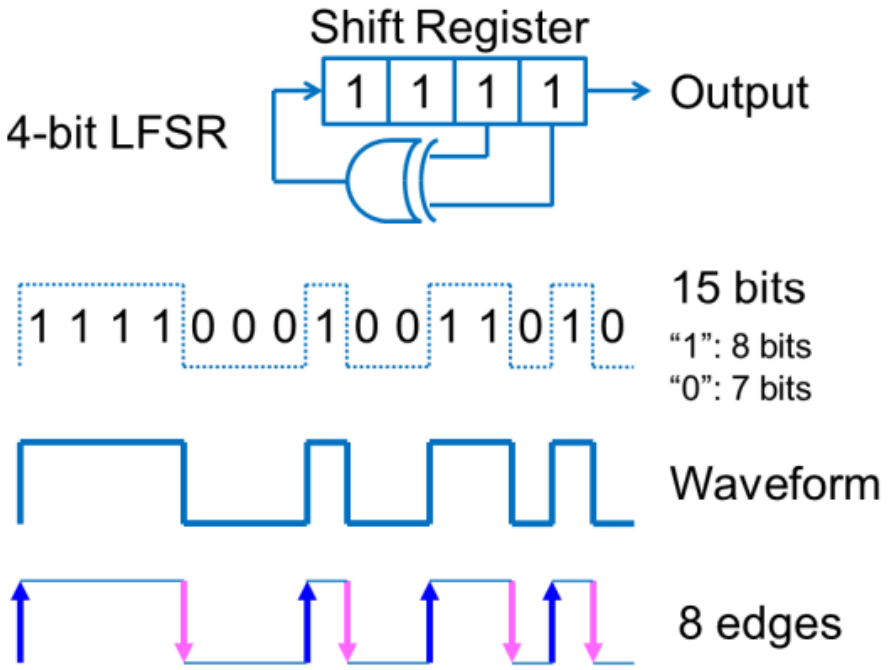


Figure 2.13: Waveform of PRBS by 4-bit LFSR

2.2. PRBS's Clock Frequency:

- Calculate the Clock Frequency:

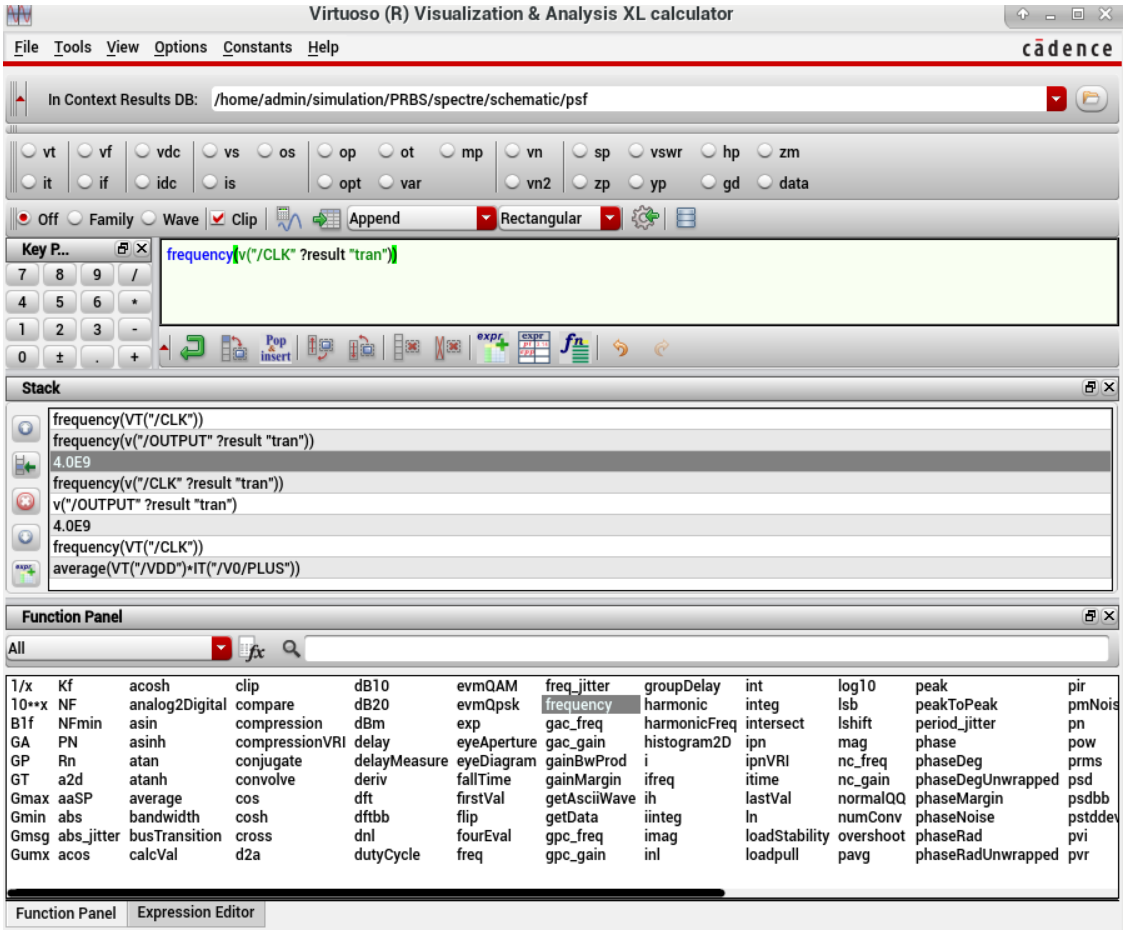


Figure 2.14: Using calculator to calculate clock frequency

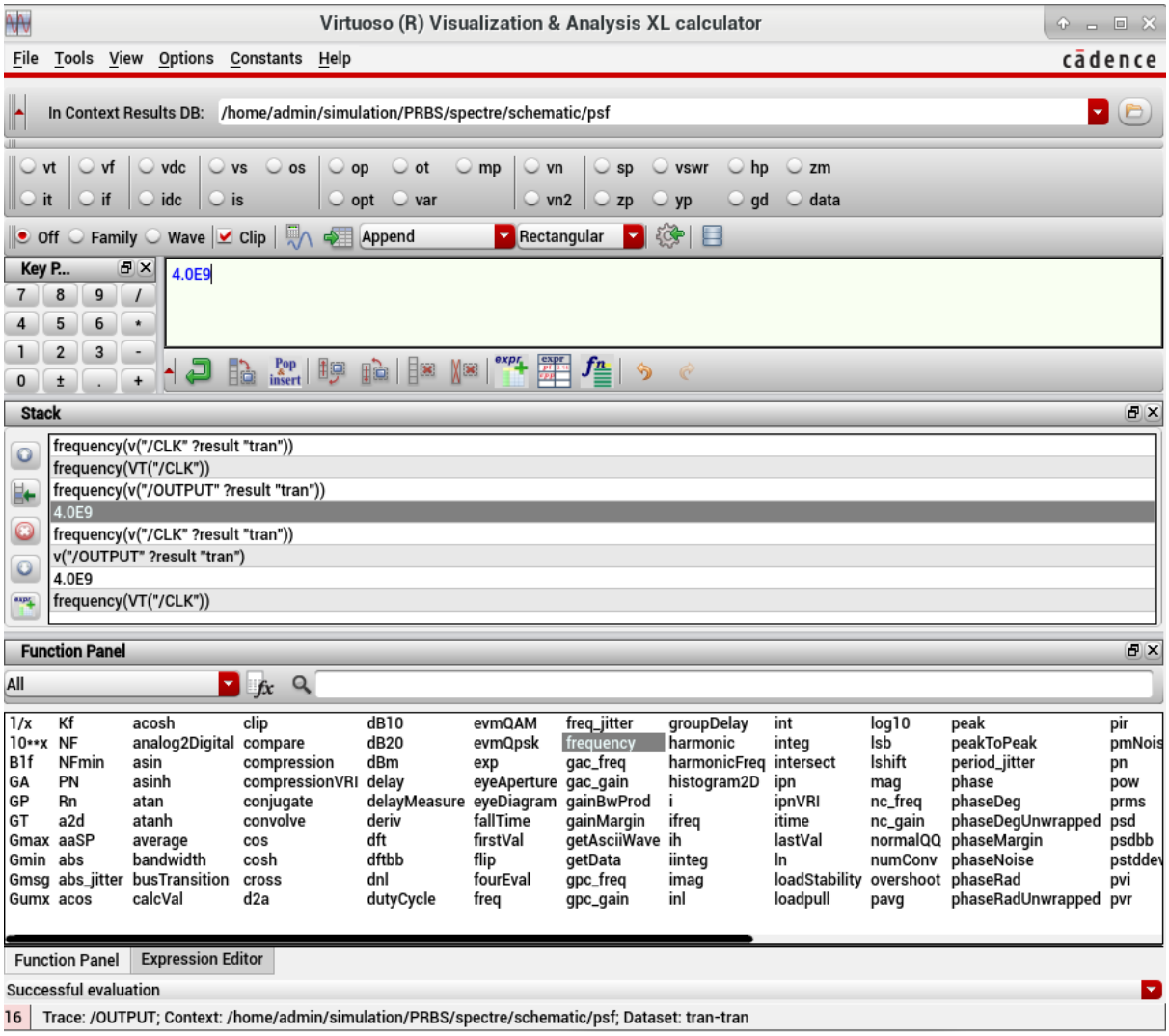


Figure 2.15: Result of Calculator

=> $f_{CLK} = 4\ GHz$.