

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ



BÁO CÁO THÍ NGHIỆM
MÔN: THIẾT KẾ VI MẠCH SỐ
LAB 3: COMBINATIONAL
AND SEQUENTIAL CIRCUIT

Giảng viên lý thuyết: Trương Quang Vinh

Nhóm:14 – Lớp: L04

Sinh viên thực hiện

Họ và tên	MSSV
Lê Đức Lân	2111634
Bùi Khánh Hoàng	2113392
Lê Duy Thức	2110184

Thành phố Hồ Chí Minh, tháng 5, năm 2024

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING

TEAMWORK REPORT

Subject: IC Design

Class: L0 Group: 0

LAB 1: STANDARD CELLS

STT	ID	First Name	Last Name	Percentage Complete
1	2111634	Lê Đức	Lân	100%
2	2113392	Bùi Khánh	Hoàng	100%
3	2110184	Lê Duy	Thức	100%

EXPERIMENT 1

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

Requirments:

- Complete the truth table, schematic, and symbol for 1-bit Full Adder.
- Define the speed of your design.
- Create layouts for each logic gate, then show DRC confirmation and corresponding schematic with proof of LVS.

a. Schematic and symbol

- The truth table of 1-bit Full Adder

INPUT			OUTPUT	
A	B	Cin	Sum	Cout
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

Bảng 1. Truth table 1-bit Full Adder

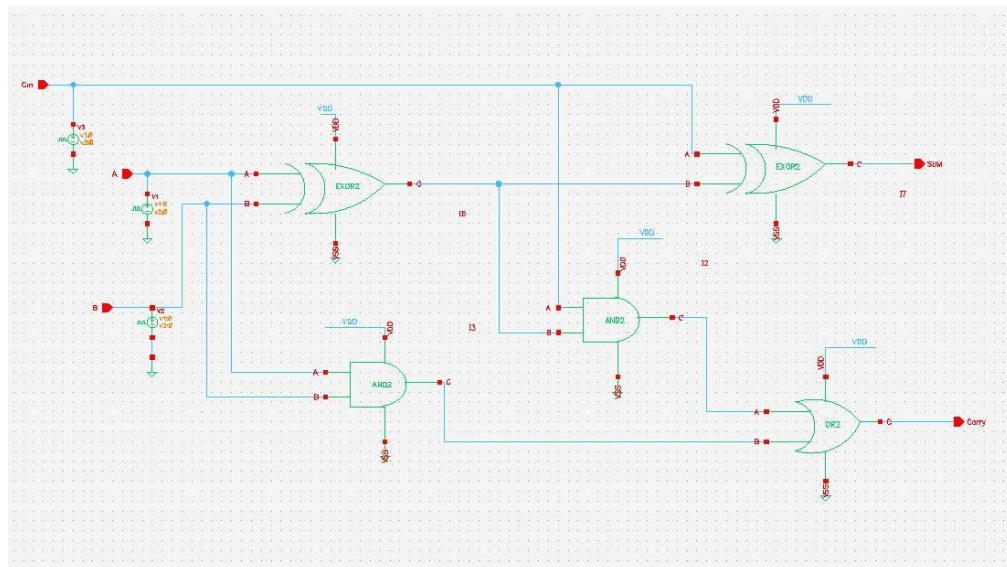
Dựa vào truth table ta có thể suy ra được:

+ Sum = 1 khi chỉ 1 trong 3 inputs (A, B, Cin) bằng 1

=> Sum = $A \oplus B \oplus Cin$

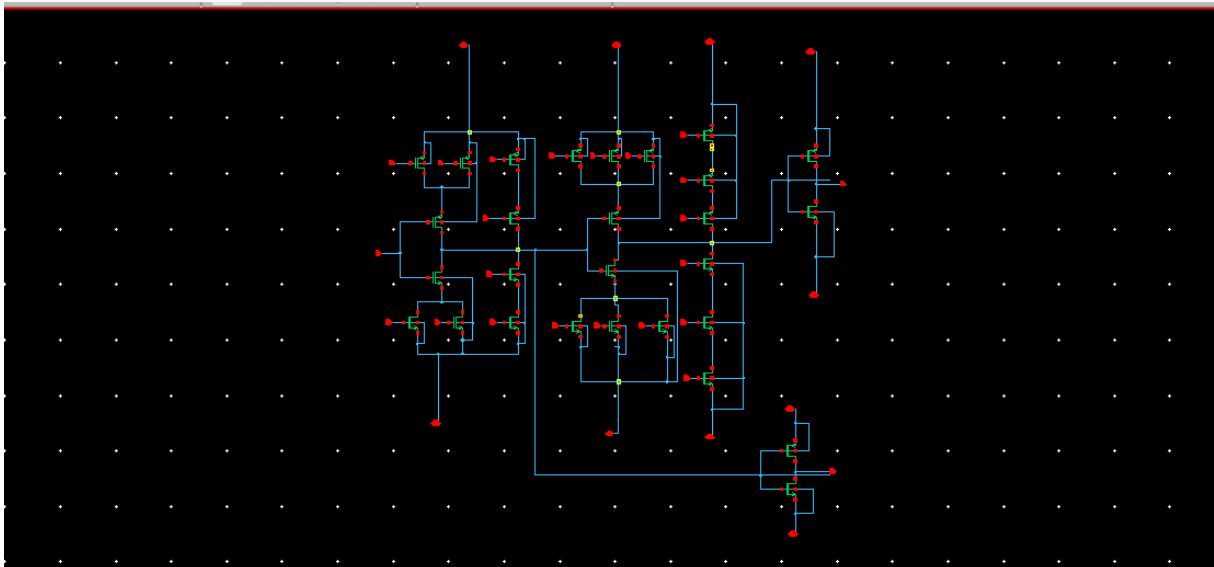
+ Cout = 1 khi có ít nhất 2 inputs bằng 1

=> Cout = $(A \oplus B) \& Cin + A \& B$

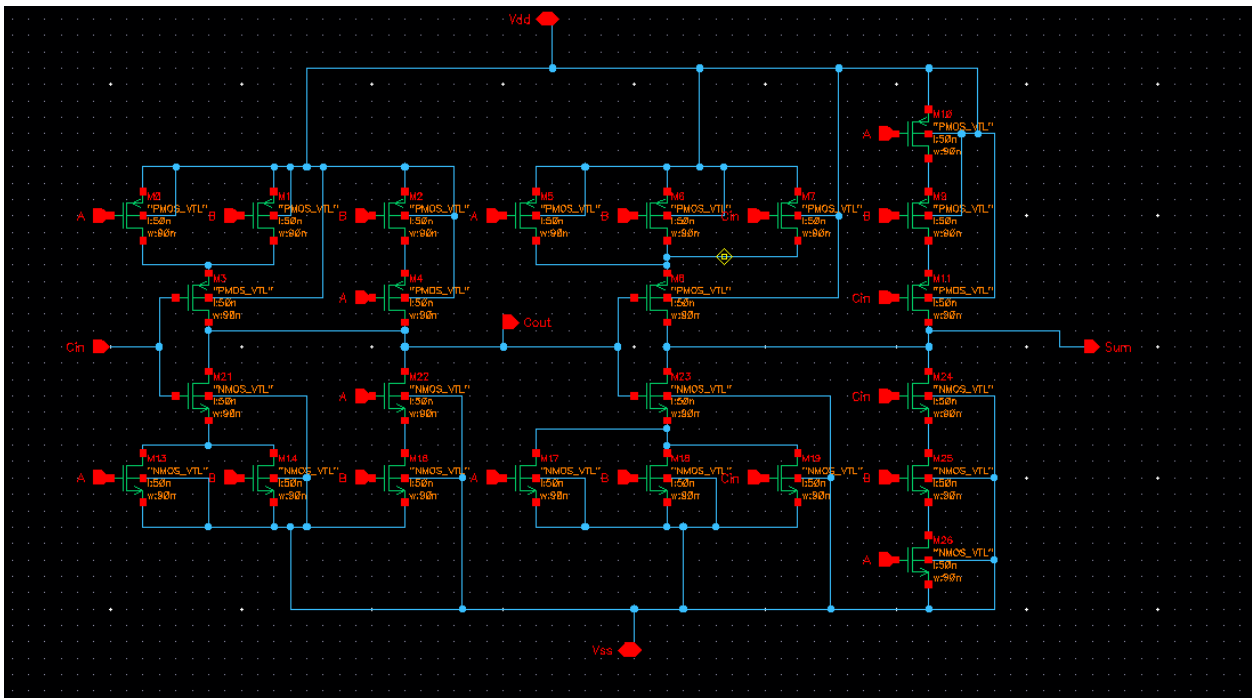


Hình 1. Schematic of 1-bit Full Adder

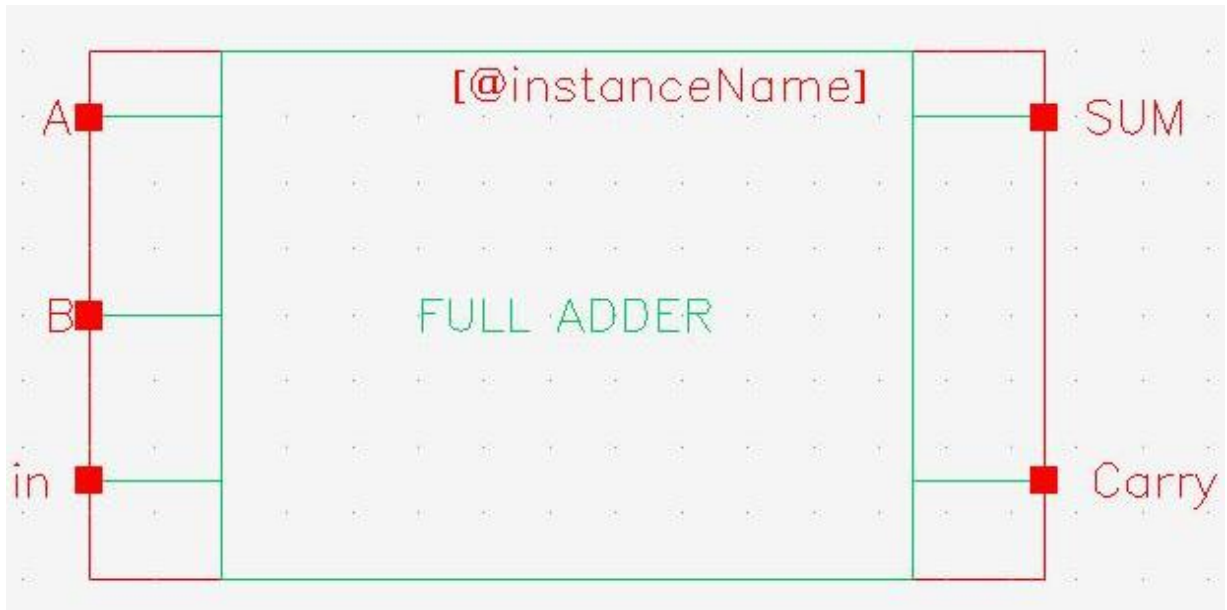
- Schematic of Full Adder 1 bit 28T



- Schematic of Full Adder 1 bit 24T



- Symbol of 1-bit Full Adder
 - + Sau khi đã hoàn thành schematic, hoàn thiện symbol.



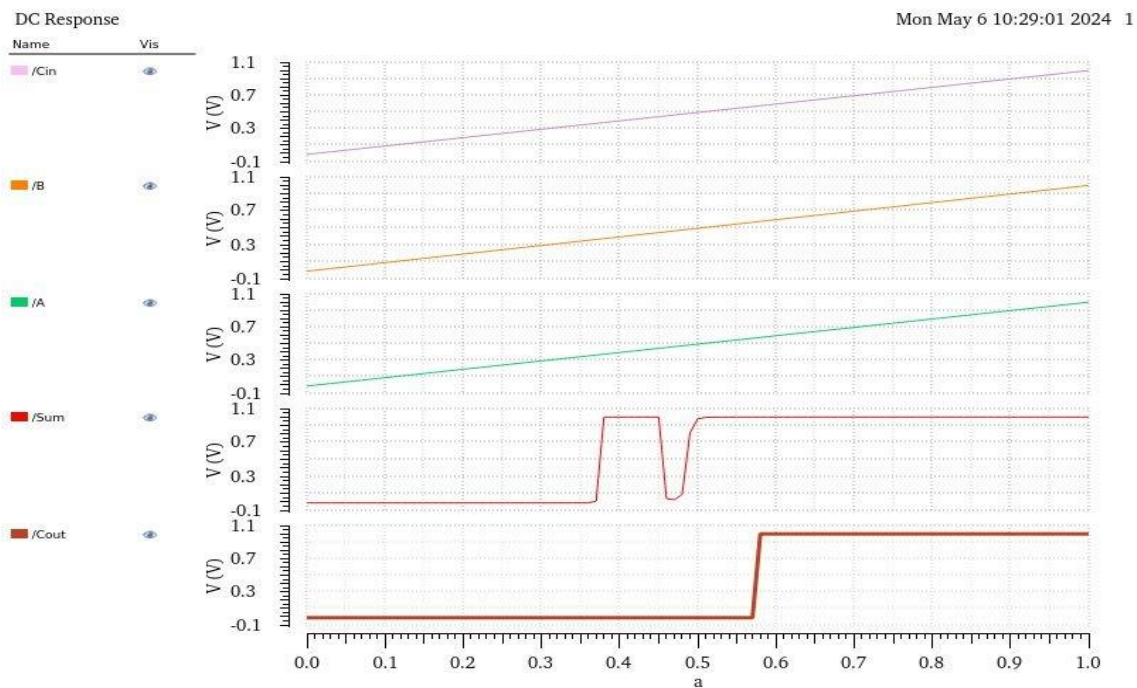
Hình 2. Symbol of 1-bit Full Adder

b.DC Analysis simulation

+ Để tiến hành đo đặc DC ta cần khởi tạo các giá trị cho từng thành phần như hình 4, với bước nhảy của V_{in} là 0.1V.

Parameters	Value
V_{dd}	1
C_{load}	1fF
V_{in}	[0;1]

Bảng 2. Setup parameter values for DC measurement



Hình 3. Kết quả DC analysis

Vin(V)	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Vs(V)	0	0	0	0	1	1	1	1	1	1	1
Vc(V)	0	0	0	0	0	0	1	1	1	1	1

Bảng 3. DC results of 1-bit Full Adder logic gate

Dựa vào kết quả đã đo đạc được đối chiếu với Hình 1 ta có thể có được nhận xét như sau:

+ Khi Vin(V) từ 0.0 đến 0.4: Đầu ra Vs(V) được giữ ở mức thấp (0), không có sự thay đổi trong đầu ra. Điều này cho thấy cổng logic không phản ứng với các mức điện áp đầu vào thấp này.

+ Khi Vin(V) từ 0.5 đến 0.6: Vs(V) vẫn giữ ở mức thấp (0), không có sự chuyển đổi. Điều này có thể được hiểu là cổng logic không phản ứng cho đến khi Vin(V) đạt mức đầu vào quan trọng (critical input) nào đó.

+ Khi Vin(V) là 0.7: Đầu ra Vs(V) chuyển đột ngột từ mức thấp (0) lên mức cao (1). Điều này phản ánh sự chuyển đổi trong chức năng của cổng logic khi đạt mức điện áp đầu vào quan trọng.

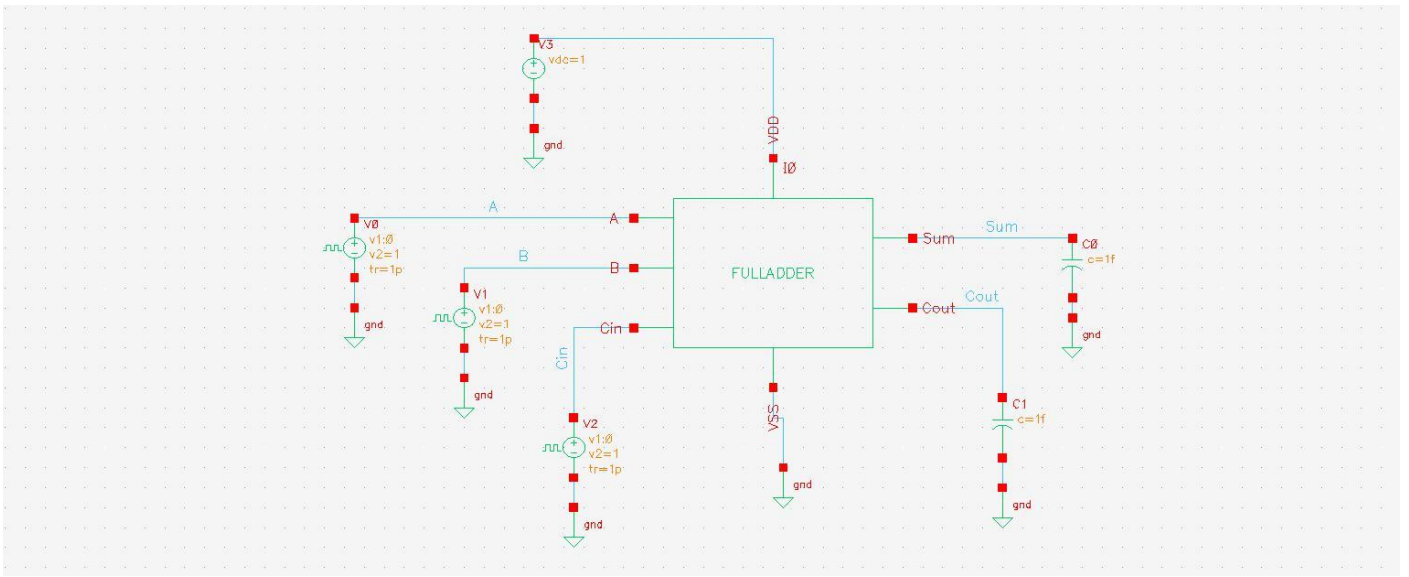
+ Khi Vin(V) từ 0.8 đến 1.0: Vs(V) duy trì ở mức cao (1), không có sự thay đổi trong đầu ra. Điều này cho thấy cổng logic giữ đầu ra ổn định ở mức cao khi mức điện áp đầu vào lớn.

c. Transient simulation:

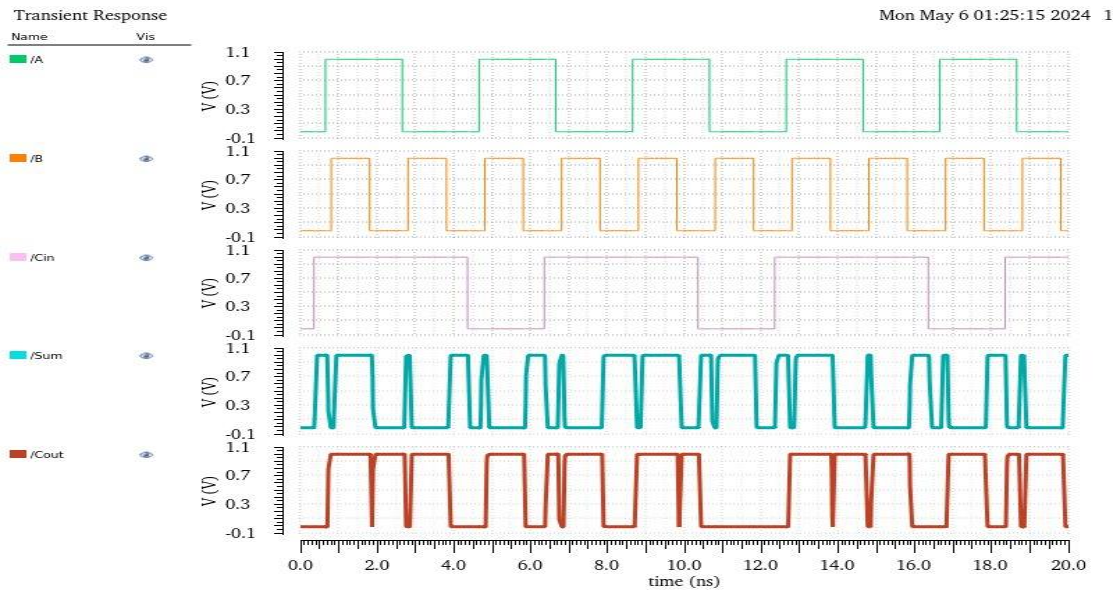
+ Để tiến hành đo đạc transient ta cần khởi tạo các giá trị đã cho như Hình 1.6 ở các thành phần schematic Hình 3.

	In1	In2	In3
Voltage 1	0V	0V	0V
Voltage 2	1V	1V	1V
Peroid	4n	2n	6n
Delay time	0.65n	0.8n	0.35n
Rise time	1p	1p	1p
Fall time	1p	1p	1p
Pulse time	2n	1n	4n

Bảng 4. Setup parameter values for transient measurement



Hình 4. Schematic 1-bit Full Adder for transient simulation



Hình 5. Transient measurement results of 1-bit Full Adder logic gate

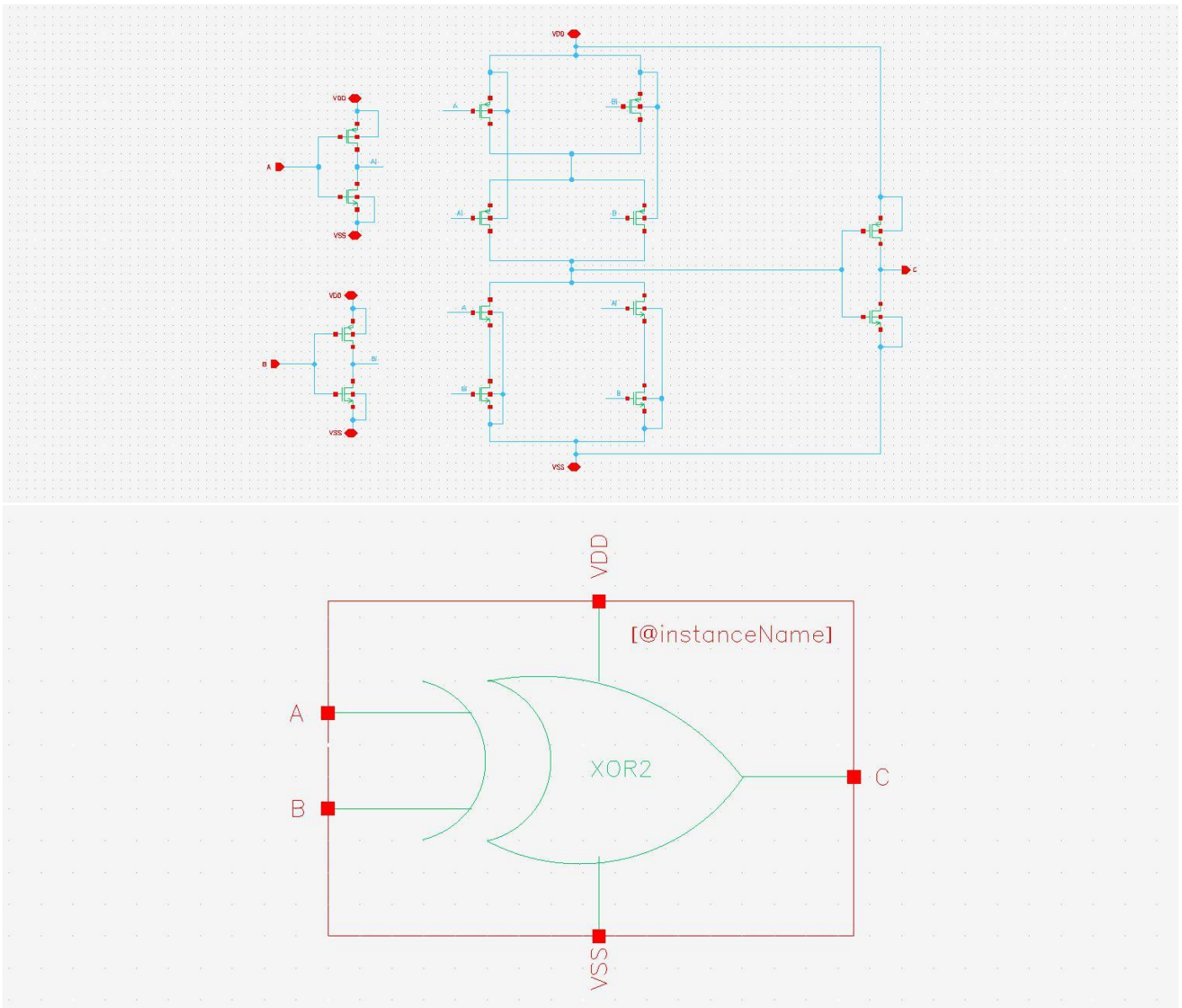
Ở Hình 8 ta có thể thấy quan sát được rõ ràng hơn các ngõ ra ứng với các giá trị input, so sánh với truth table ở Hình 5 thì kết quả cho ra hoàn toàn chính xác.

d. Speed of design

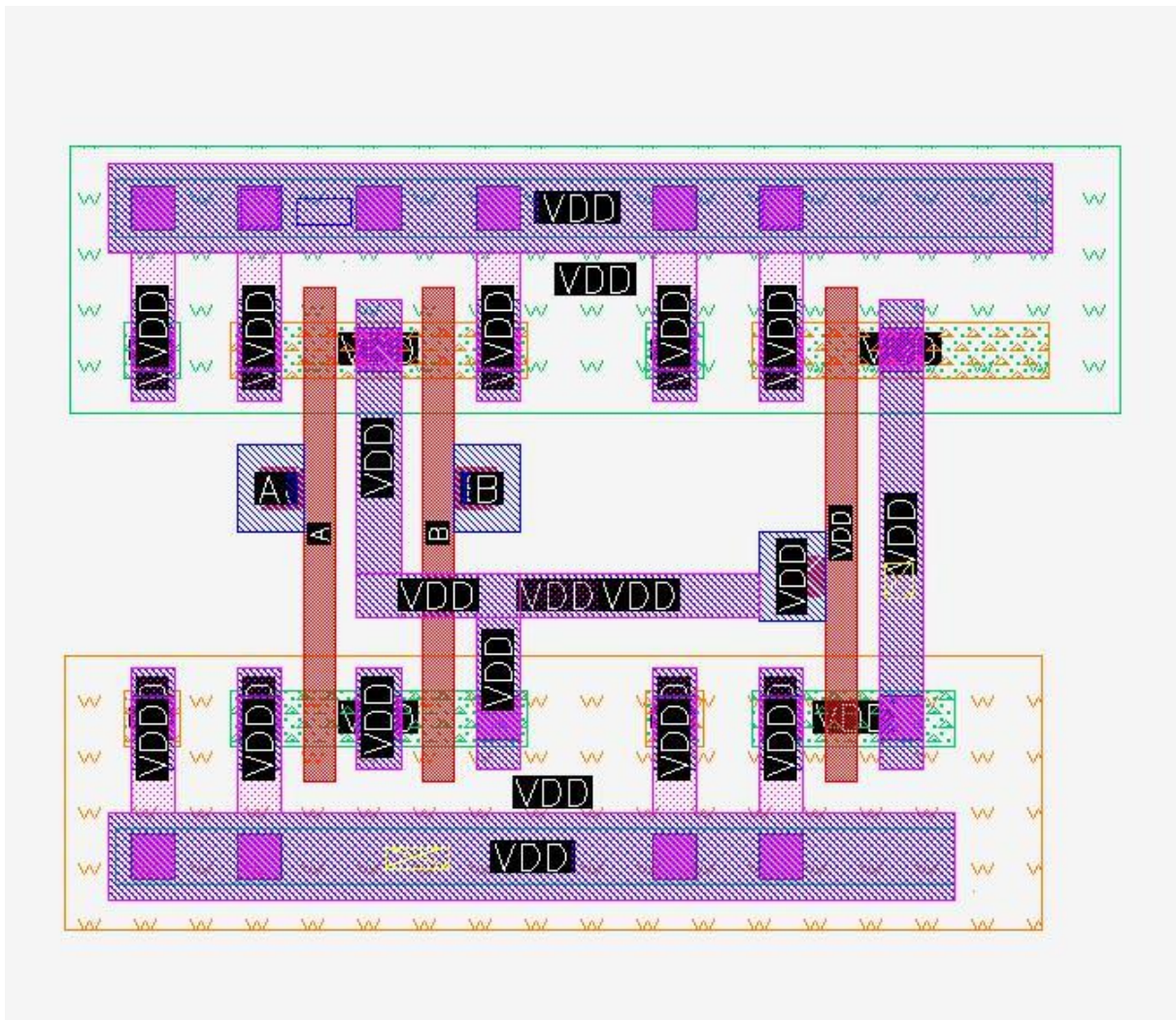
Parameters	Original FA (28T)	FA 24T
t_{rise} -Rising time (10%-90%)	32ps	47ps
t_{fall} – Falling time (90%-10%)	40ps	35ps
t_{pd} – Average propagation delay (50% - 50%)	63ps	70ps
Power consumption	1.61E-10	4.81E-05

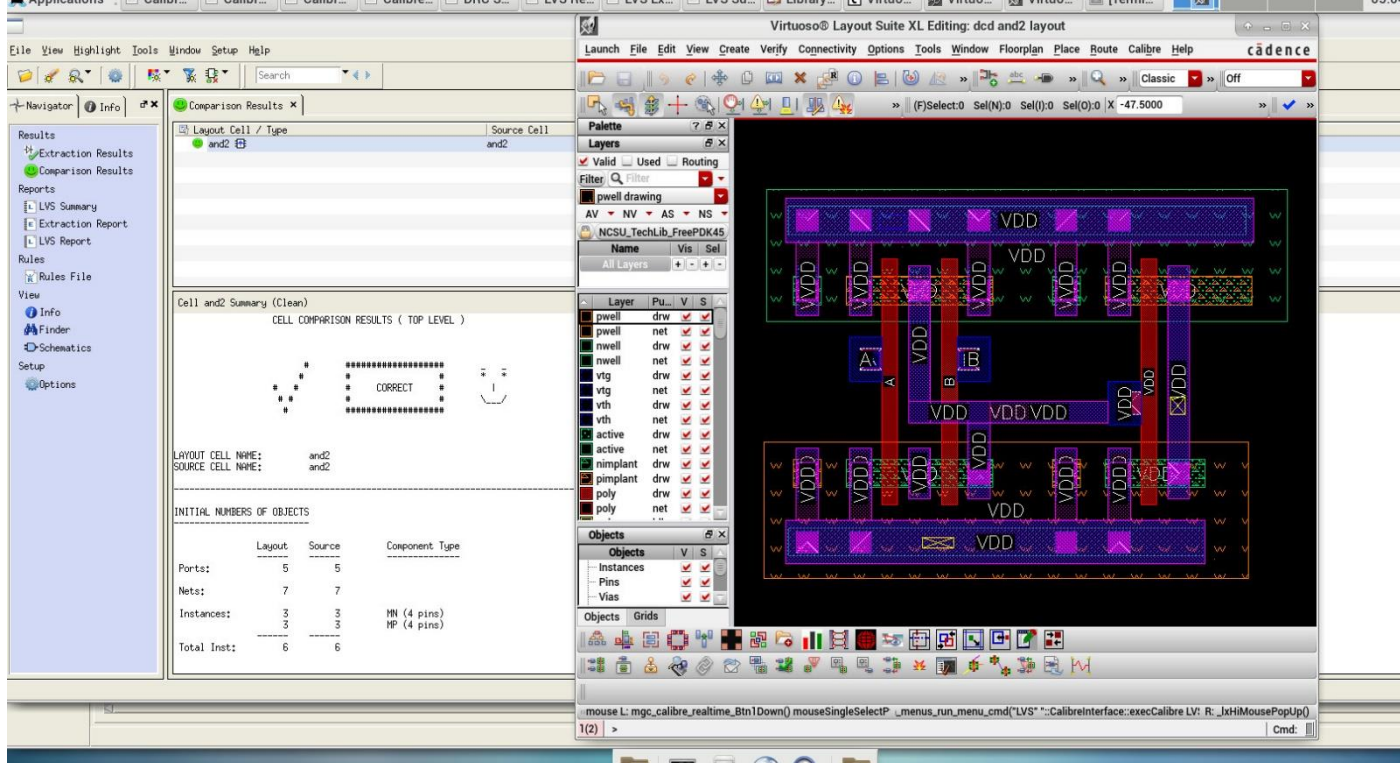
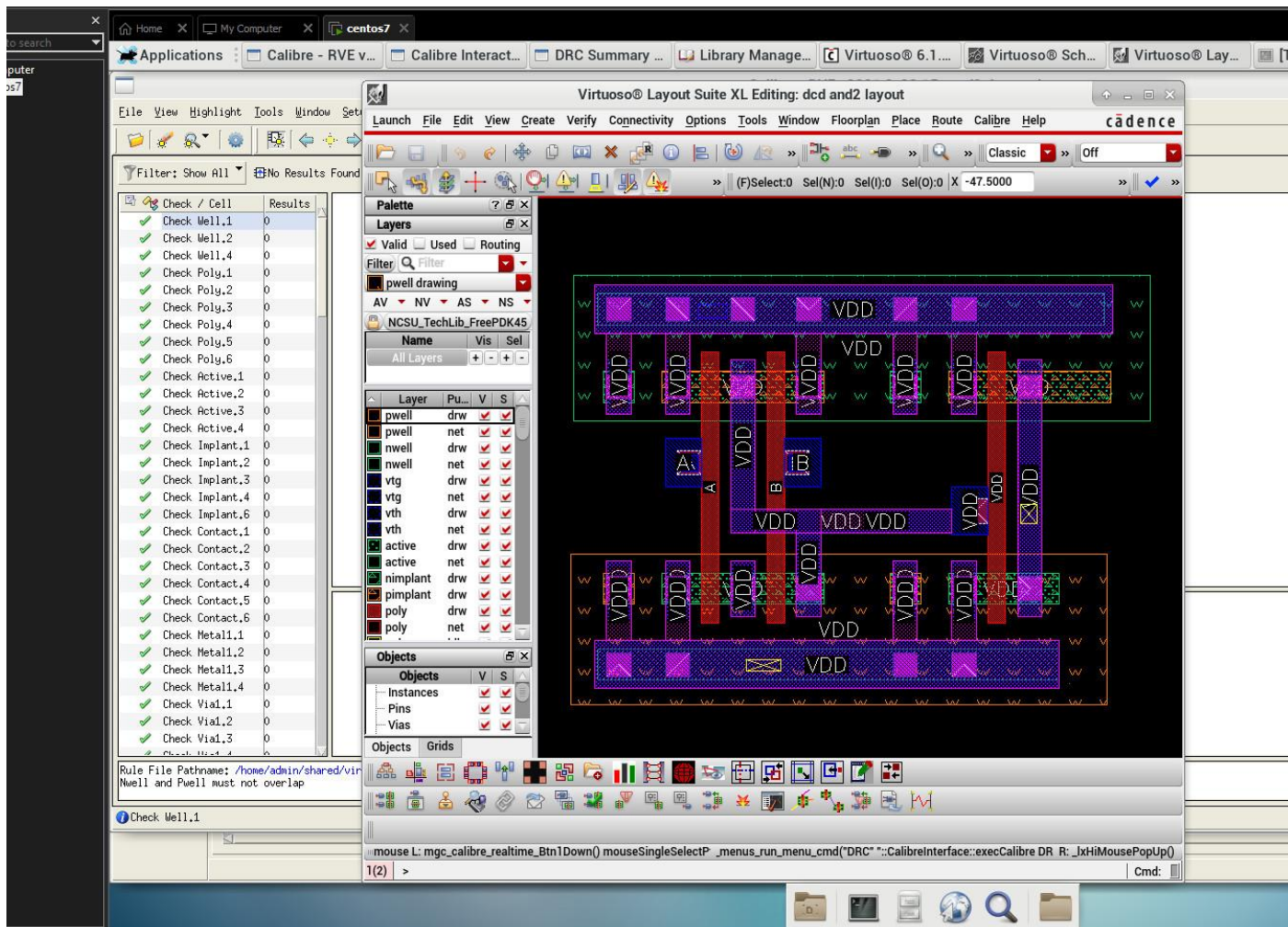
e. Layout of 1-bit Full Adder and each logic inside.

EXOR2: Schematic and symbol.

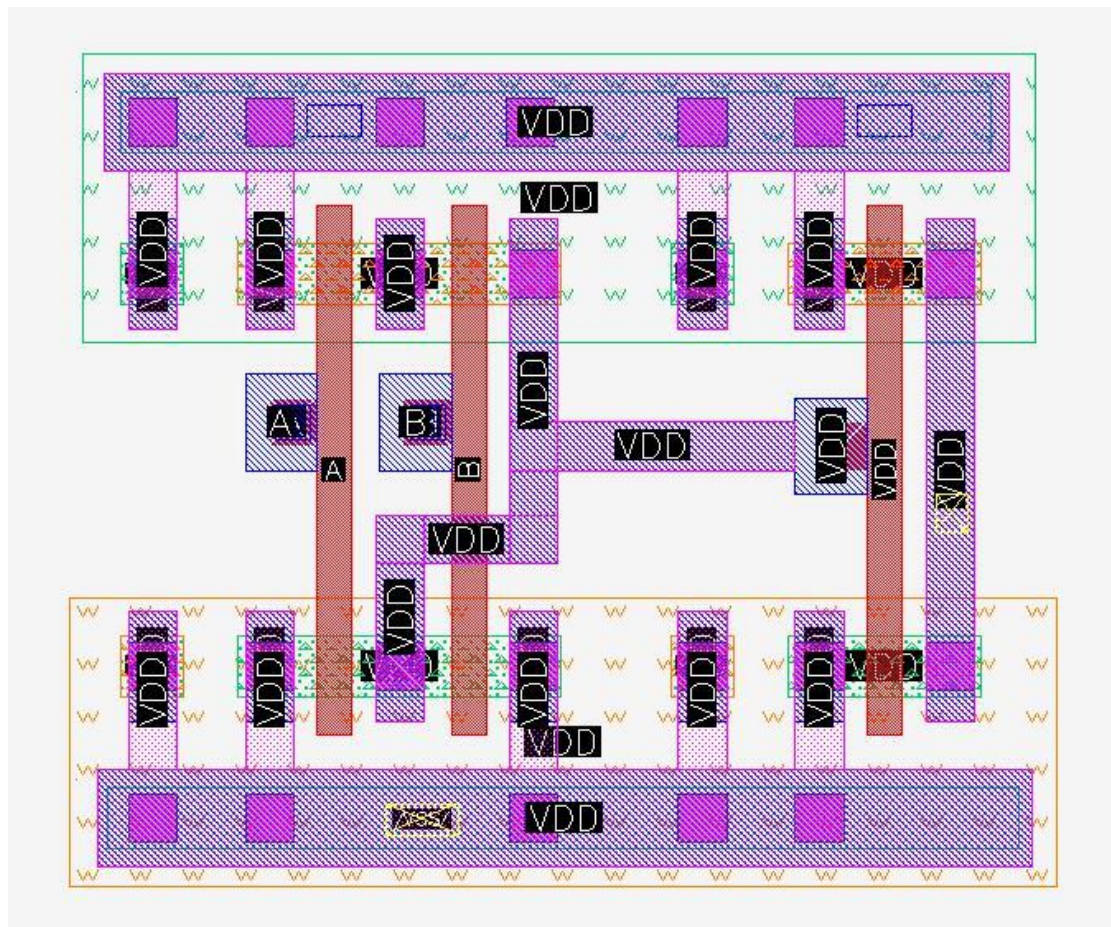


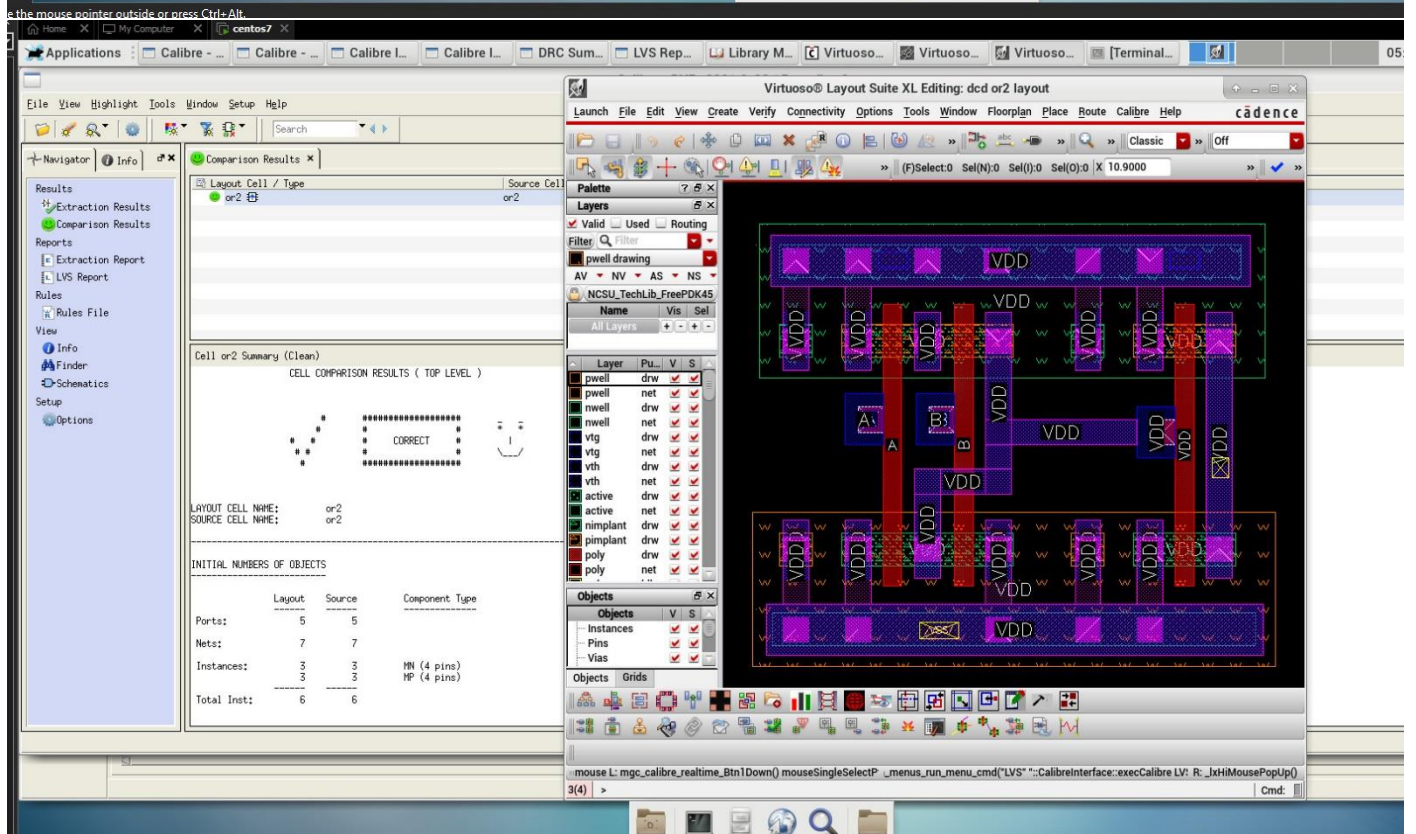
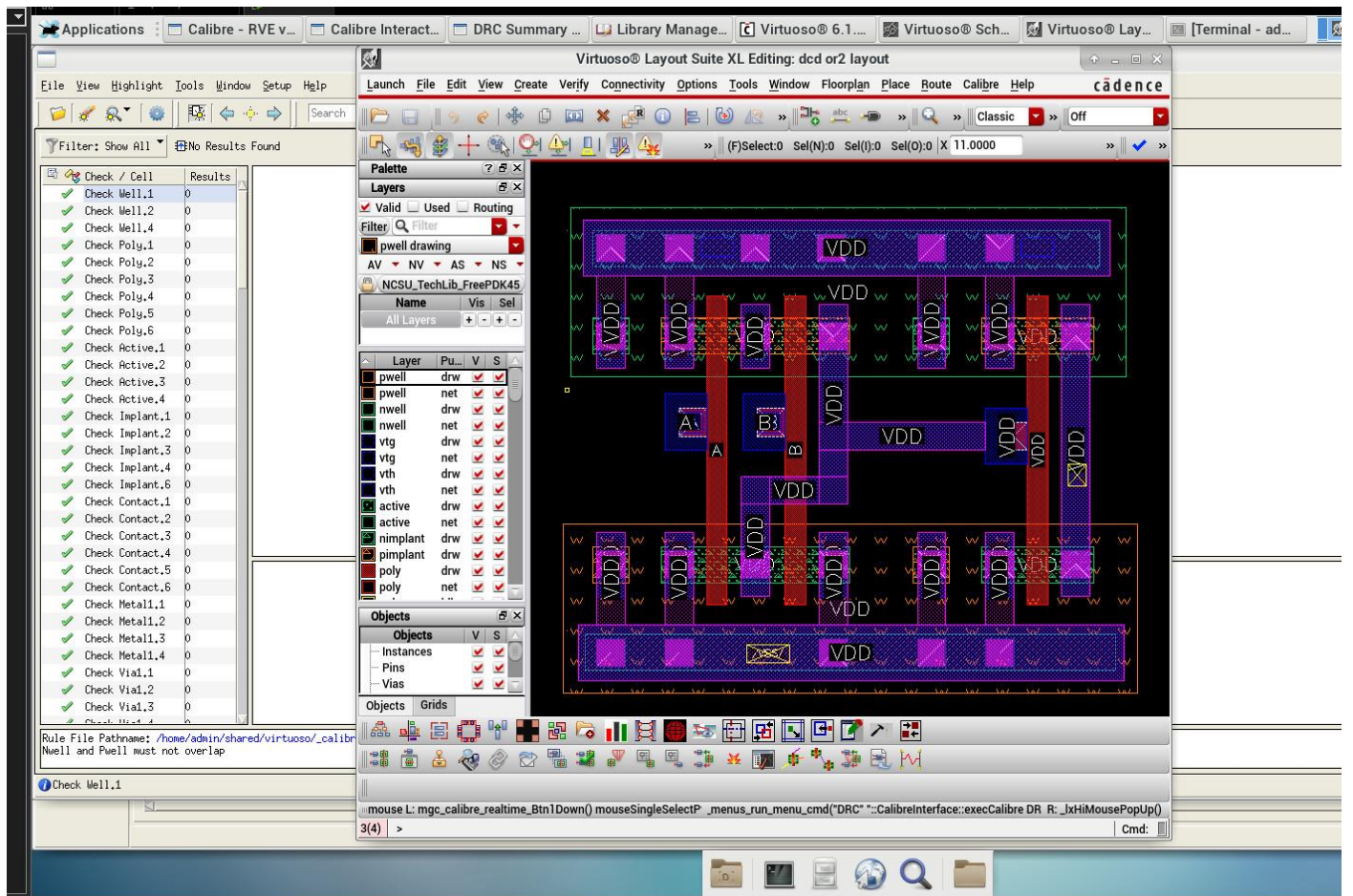
AND2: Layout and check DRC, LVS.





OR2: Layout and check DRC, LVS.





Question: Due to the topology shown in Figure 3, why do people implement the PDN of the first stage using the equation $C_{out} = (A+B)C_{in} + AB$ instead of $C_{out} = AB + AC_{in} + BC_{in}$?

Answer:

The implementation of the Carry-out (C_{out}) function in a full adder using the equation $C_{out} = (A+B)C_{in} + AB$ instead of $C_{out} = AB + AC_{in} + BC_{in}$ is primarily driven by considerations of efficiency, simplicity, and the number of transistors required in CMOS logic design. Here's a detailed explanation of why the former equation is often preferred:

Boolean Simplification

Both equations are logically equivalent, meaning they produce the same output for any given inputs A , B , and C_{in} . To understand why one might be preferred over the other, let's simplify both equations using Boolean algebra.

Original Equations:

$$C_{out} = AB + AC_{in} + BC_{in} \quad C_{out} = AB + AC_{in} + BC_{in}$$

$$C_{out} = (A+B)C_{in} + AB \quad C_{out} = (A+B)C_{in} + AB$$

Simplification of the First Equation:

$$C_{out} = AB + AC_{in} + BC_{in} \quad C_{out} = AB + AC_{in} + BC_{in}$$

$$C_{out} = AB + C_{in}(A+B) \quad C_{out} = AB + C_{in}(A+B) \text{ (factoring out } C_{in})$$

Simplification of the Second Equation:

$$C_{out} = (A+B)C_{in} + AB \quad C_{out} = (A+B)C_{in} + AB$$

As you can see, both equations simplify to the same form:

$$C_{out} = AB + C_{in}(A+B) \quad C_{out} = AB + C_{in}(A+B)$$

Transistor Count and Circuit Complexity

In CMOS logic design, the goal is often to minimize the number of transistors used while maintaining or improving performance. Let's analyze the transistor implementations of both forms.

Equation $C_{out} = (A+B)C_{in} + AB$:

This equation suggests an implementation where you first compute $A+B$, then AND it with C_{in} , and finally OR this result with AB .

(A + B): This requires a NOR gate followed by an inverter, which typically uses 4 transistors.

(A + B) AND C_{in} : This requires an AND gate, typically using 6 transistors.

AB: This requires an AND gate, typically using 6 transistors.

OR: To combine $(A+B)C_{in}$ and AB , we need an OR gate, typically using 6 transistors.

Total transistor count (not considering optimizations): 4 (NOR + inverter) + 6 (AND) + 6 (AND) + 6 (OR) = 22 transistors.

Equation $C_{out} = AB + AC_{in} + BC_{in}$:

This equation suggests three separate AND gates followed by an OR gate to combine their outputs.

AB: This requires an AND gate, typically using 6 transistors.

AC_{in}: This requires an AND gate, typically using 6 transistors.

BC_{in}: This requires an AND gate, typically using 6 transistors.

OR: To combine AB , AC_{in} , and BC_{in} , we need a three-input OR gate, typically implemented with multiple two-input OR gates using 8 transistors (assuming two stages of OR).

gates).

Total transistor count (not considering optimizations): 6 (AND) + 6 (AND) + 6 (AND) + 8 (OR) = 26 transistors.

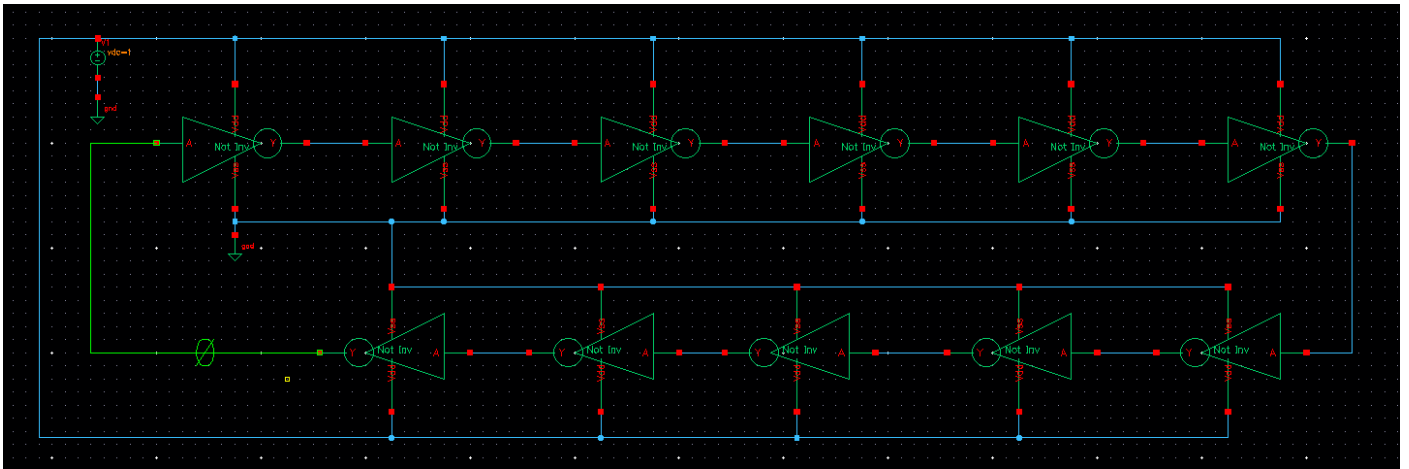
EXPERIMENT 2

Objective: Know how to design a basic sequential circuit.

Requirments:

- You will learn PRBS or pseudo-random binary sequence

a. Schematic NOT logic gate sequetial circuit



b. Waveform

