

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT  
KHOA ĐIỆN-ĐIỆN TỬ  
BỘ MÔN KT MÁY TÍNH – VIỄN THÔNG**

**BÁO CÁO THỰC TẬP  
MÔN HỌC:**

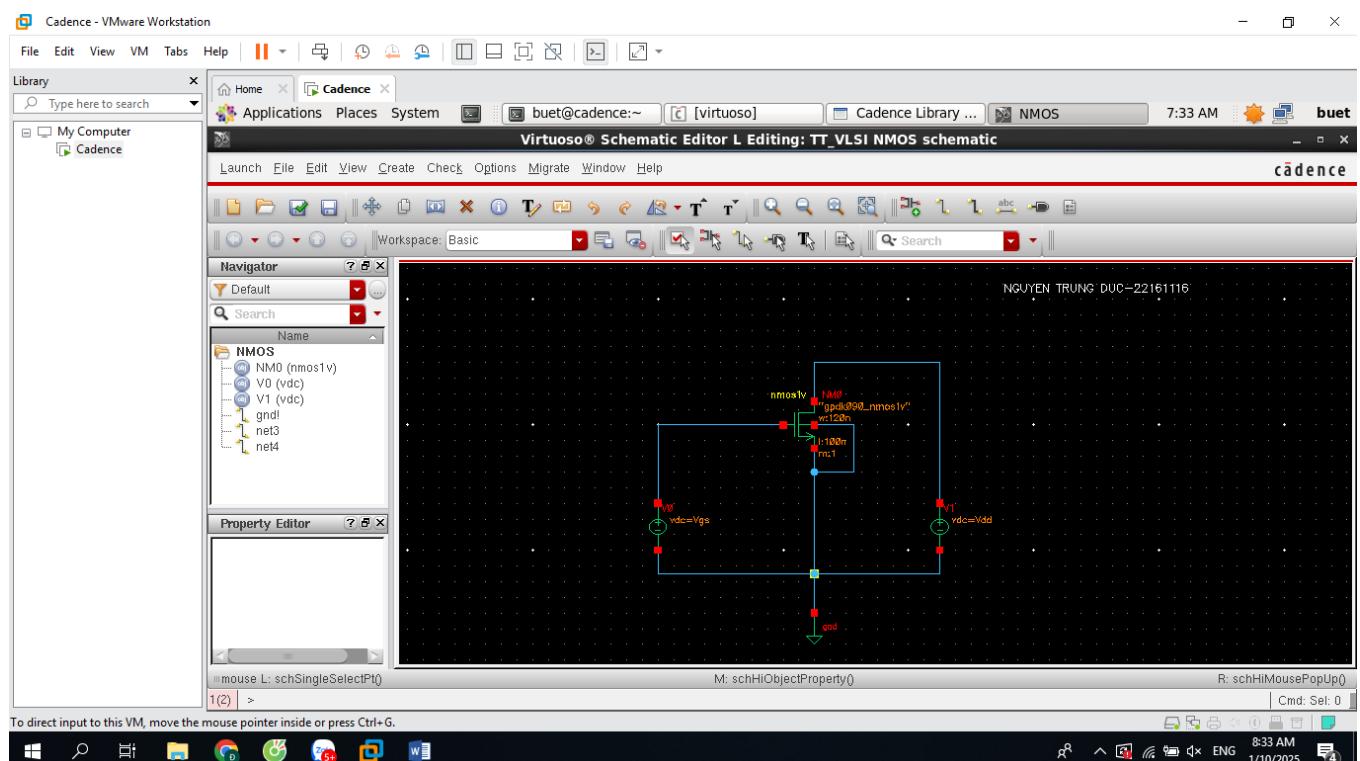
Họ tên: Nguyễn Trung Đức	MSSV: 22161116	Nhóm: (Sáng thứ 6, tuần 1-8)

**1. Khảo sát đặc tuyến của nMOS và pMOSS công nghệ 90nm**

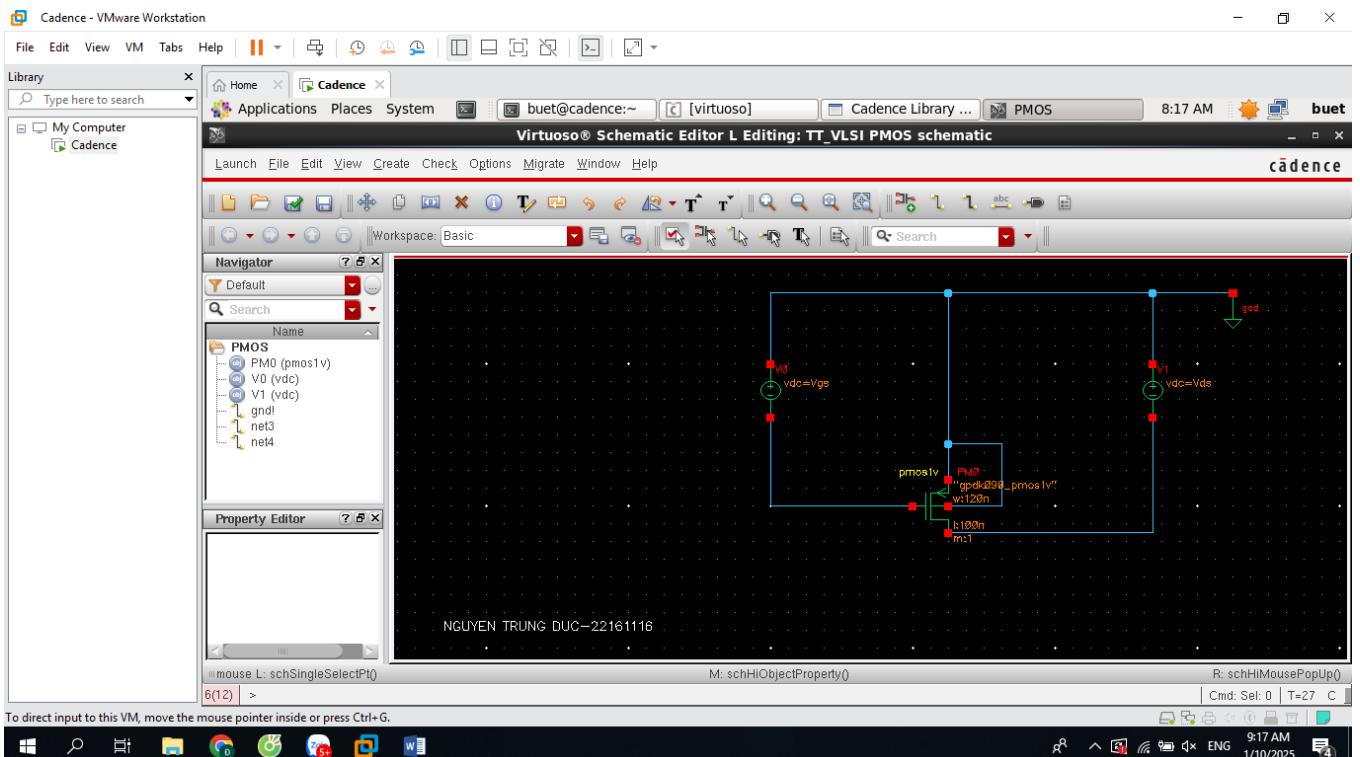
- Vẽ đặc tuyến của NMOS và PMOS với điện áp cung cấp 1.2V, các giá trị Vgs lần lượt là 0.2. 0.4, 0.6, 0.8, 1V.

- Sinh viên trình bày sơ đồ nguyên lý mạch

**NMOS**



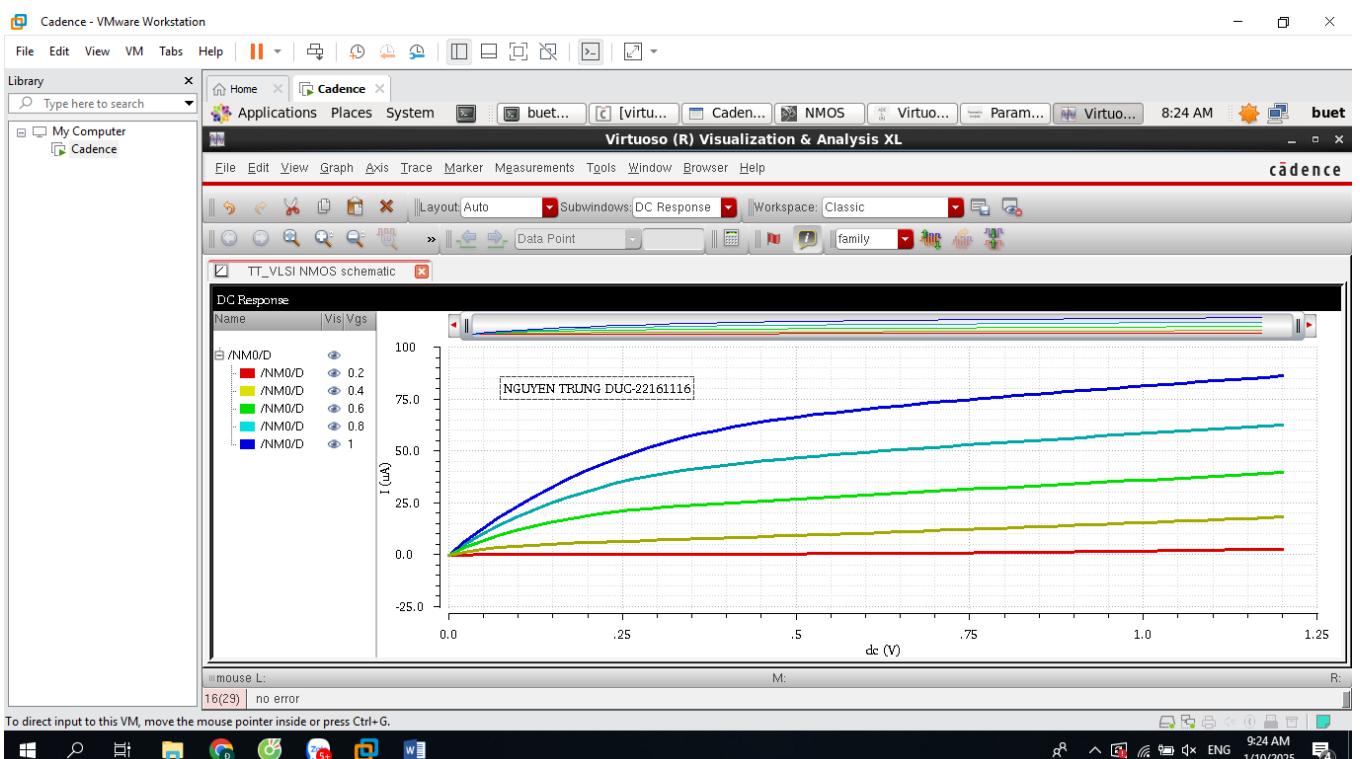
**PMOS**



Lưu ý: chú ý rằng cực body của NMOS sẽ được nối xuống GND, còn cực body của PMOS sẽ được nối lên VDD.

- Chụp hình kết quả, trong cửa sổ phải có text ghi tên sinh viên.

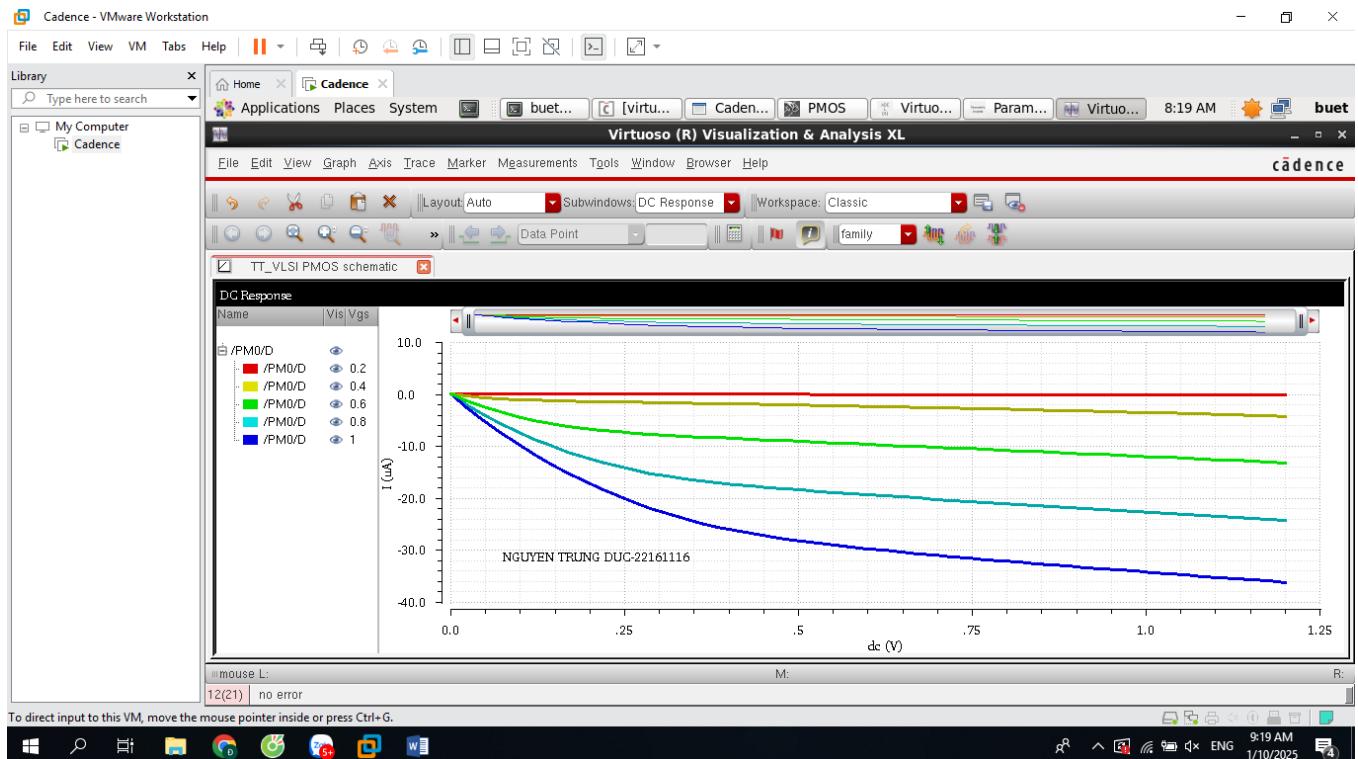
## NMOS



Phân tích:

- Dòng  $I_D$  tăng khi  $V_{DS}$  tăng:
  - Ở vùng nhỏ, dòng tăng tuyến tính (Vùng ohmic).
  - Khi  $V_{DS}$  lớn, đường cong dần bão hòa (vùng bão hòa).
- $V_{GS}$  ảnh hưởng mạnh đến  $I_D$ :
  - Khi  $V_{GS}$  tăng, dòng  $I_D$  cũng tăng đáng kể.
  - Nếu  $V_{GS}$  quá nhỏ (nhỏ hơn ngưỡng  $V_{th}$ ), dòng  $I_D$  gần như bằng 0, transistor ở trạng thái tắt.

## PMOS

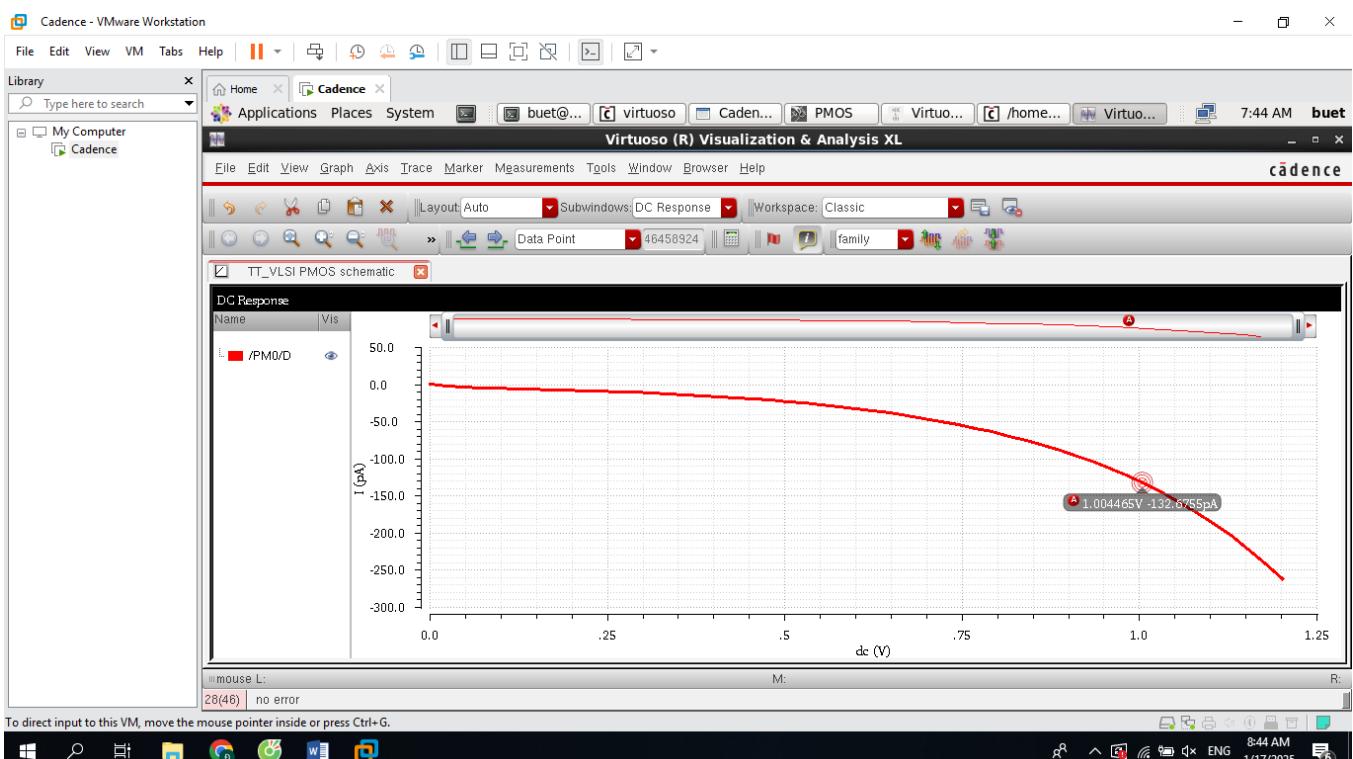
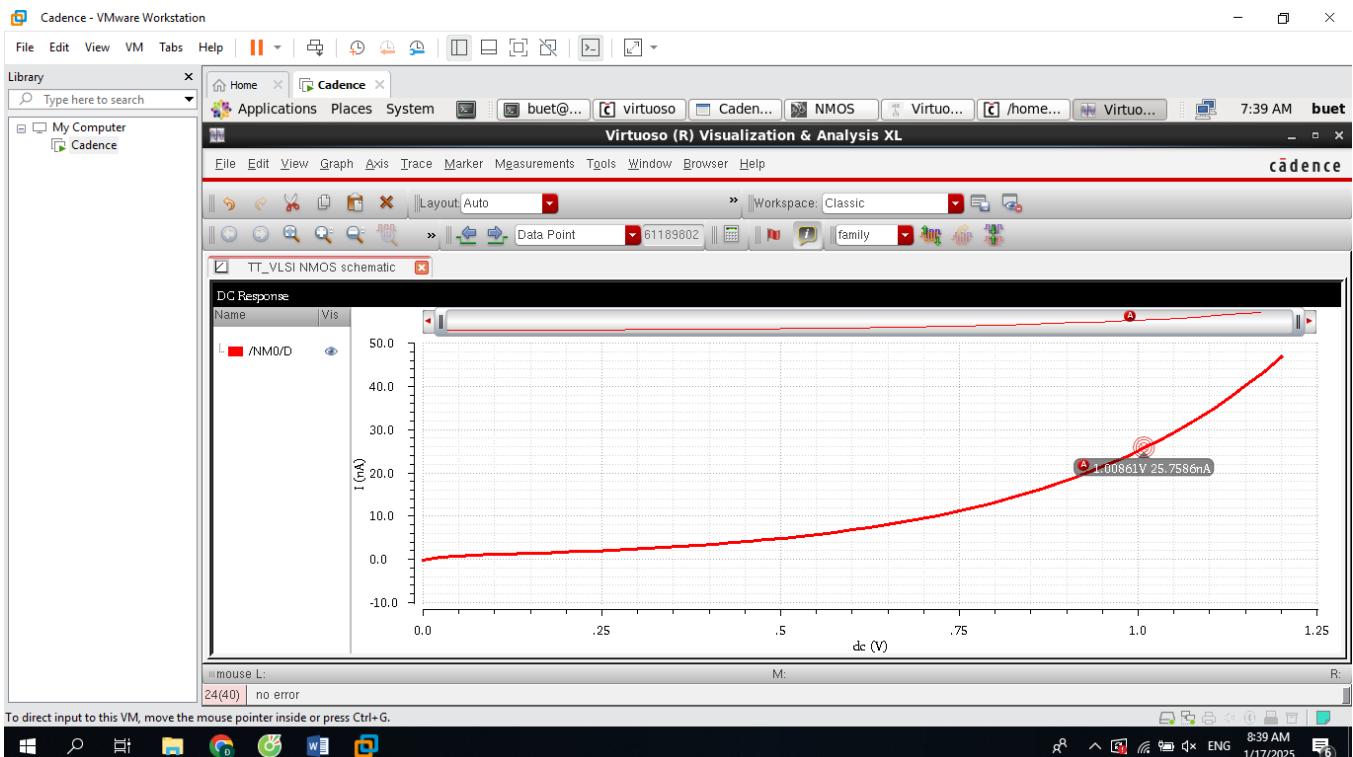


## Phân tích:

- Dòng  $I_D$  giảm khi  $V_{DS}$  tăng:
  - Khi  $V_{DS}$  giảm (hướng về giá trị âm), dòng  $I_D$  tăng lên về độ lớn (theo hướng âm).
  - Khi  $V_{DS}$  lớn hơn một giá trị nhất định, transistor đi vào vùng bão hòa, dòng gần như không đổi.
- $V_{GS}$  ảnh hưởng đến  $I_D$ :
  - Khi  $V_{GS}$  giảm (theo hướng âm), dòng  $I_D$  tăng mạnh hơn.
  - Nếu  $V_{GS}$  không đủ âm (lớn hơn điện áp ngưỡng  $V_{th}$ ), transistor không dẫn hoặc dẫn yếu.
- So sánh với NMOS:
  - Đặc tuyến PMOS ngược dấu so với NMOS (dòng điện âm, điện áp điều khiển âm).
  - Vận hành trong mạch số và tương tự, thường dùng trong CMOS kết hợp với NMOS.

## 2. Xác định điện trở on/off của nMOS, pMOS

- Đo và xác định điện trở OFF của nMOS, pMOS sử dụng công nghệ 90nm



Thành phần	$V_{gs}$ (V)	$V_{DD}$ (V)	$I_{DS}$ (nA)	$R_{on} = V_{DD}/I_{DS}$ ( $\Omega$ )
N_MOS	0	1	25.7586	38,821,985.67
P_MOS	0	1	-0.1326755	7,537,186,594

Phân tích:

- Điện trở OFF của NMOS (Roff\_NMOS):
  - Với  $V_{GS} = 0V$ ,  $V_{DD} = 1V$ , dòng rò  $I_{DS} = 25.7586nA$
  - Điện trở OFF được tính bằng:

$$R_{off} = V_{DD}/I_{DS} = 1V/25.7586 \cdot 10^{-9} A = 38,821,985.67 \Omega$$

- Giá trị này khá lớn, phản ánh rằng khi NMOS ở trạng thái tắt (OFF), dòng rò rất nhỏ, phù hợp với mong đợi.
- Điện trở OFF của PMOS ( $R_{off\_PMOS}$ ):

  - Với  $V_{GS} = 0V$ ,  $V_{DD} = 1V$ , dòng rò  $I_{DS} = -0.1326755nA$
  - Điện trở OFF được tính bằng:
$$R_{off} = V_{DD}/|I_{DS}| = 1V/0.1326755 \cdot 10^{-9} A = 7,537,186,594 \Omega$$
  - Giá trị này lớn hơn rất nhiều so với NMOS, cho thấy PMOS có dòng rò nhỏ hơn đáng kể.

Nhận xét:

- PMOS có điện trở OFF lớn hơn nhiều so với NMOS (~7.537GΩ so với ~38.82MΩ), chứng tỏ dòng rò của PMOS nhỏ hơn NMOS. Điều này phù hợp với thực tế vì trong công nghệ CMOS, PMOS thường có dòng rò thấp hơn NMOS do sự khác biệt trong cơ chế dẫn điện.
- Dòng rò của NMOS tương đối lớn (~25.76nA), điều này có thể ảnh hưởng đến hiệu suất tổng thể của mạch nếu không được kiểm soát tốt.
- Sự chênh lệch giữa NMOS và PMOS về điện trở OFF có thể ảnh hưởng đến thiết kế CMOS, đặc biệt trong việc tối ưu hóa mức tiêu thụ điện năng và rò rỉ tĩnh.
- Đo và xác định điện áp ON của nMOS, pMOS, với  $V_{GS}$  lần lượt bằng 0.6V và 1.0V,  $V_{DD} = 1.0V$   
Sinh viên trình bày kết quả một cách khoa học và dễ đọc (có thể sử dụng bảng, chart, .... )





Thành phần	$V_{gs}$ (V)	$V_{DD}$ (V)	$I_{DS}$ ( $\mu A$ )	$R_{on} = V_{DD}/I_{DS}$ ( $\Omega$ )
N_MOS	0.6	1	35.6489	28,051.36
N_MOS	1	1	81.1331	12,325.42
P_MOS	0.6	1	-11.9803	83,470.36
P_MOS	1	1	-34.1613	29,272.89

Phân tích:

- NMOS có điện trở ON nhỏ hơn PMOS:
  - Khi  $V_{GS} = 0.6V$ , NMOS có  $R_{on} = 28,051.36 \Omega$ , còn PMOS có  $R_{on} = 83,470.36 \Omega$ .
  - Khi  $V_{GS} = 1V$ , NMOS giảm xuống  $12,325.42 \Omega$ , trong khi PMOS giảm xuống  $29,272.89 \Omega$ .
- Điện trở ON giảm khi  $V_{GS}$  tăng:
  - Điều này phù hợp với lý thuyết MOSFET: khi  $V_{GS}$  lớn hơn ngưỡng, kênh dẫn mở rộng hơn, dẫn đến điện trở nhỏ hơn.
- PMOS có điện trở lớn hơn NMOS:
  - Do lỗ trống có độ linh động thấp hơn electron, nên PMOS thường có điện trở lớn hơn NMOS trong cùng điều kiện.

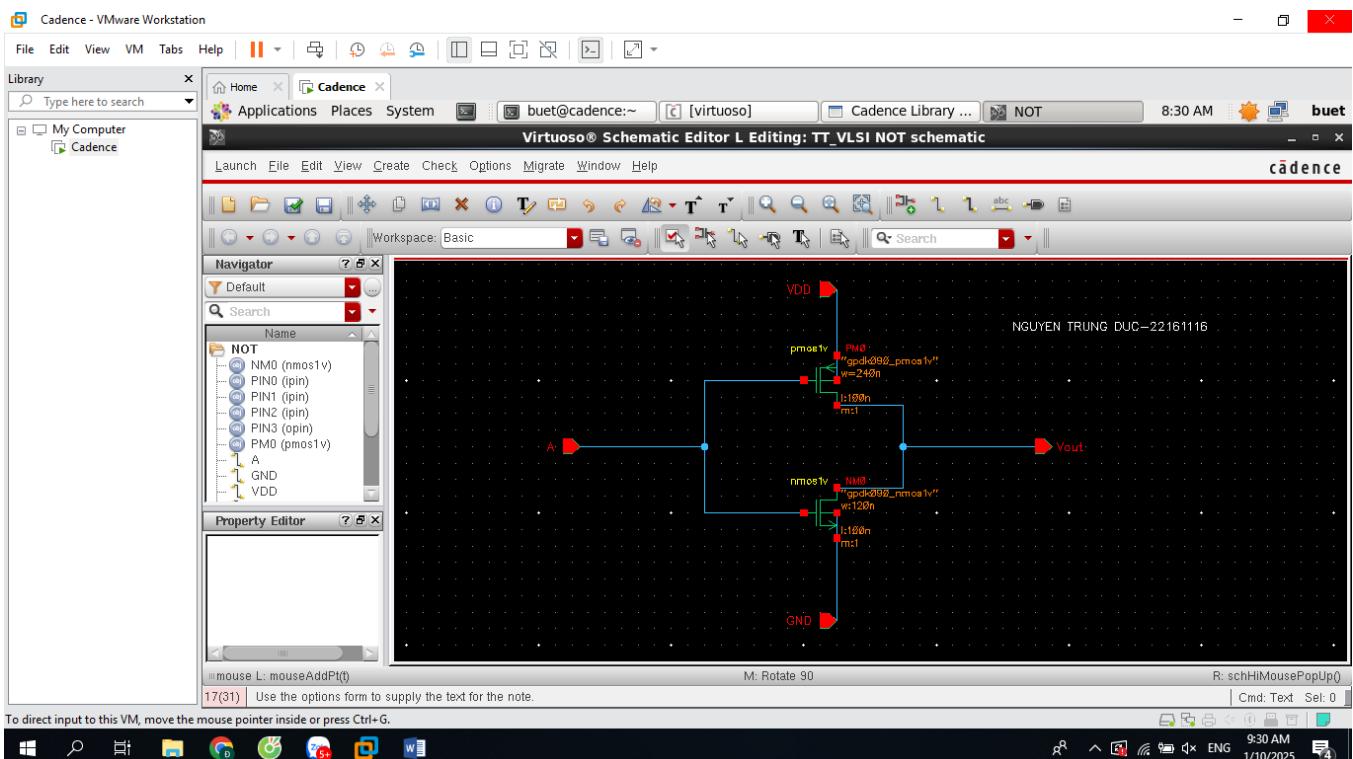
Nhận xét:

- Khi thiết kế mạch, NMOS thường được sử dụng làm công tắc chính do điện trở thấp hơn.
- PMOS thường ít hiệu quả hơn trong dẫn điện, nhưng vẫn cần thiết trong thiết kế CMOS để tạo cặp bổ sung.
- Điện trở ON phụ thuộc vào  $V_{GS}$ , do đó trong các ứng dụng cần giảm thiểu tiêu hao điện năng, nên điều chỉnh mức  $V_{GS}$  phù hợp.

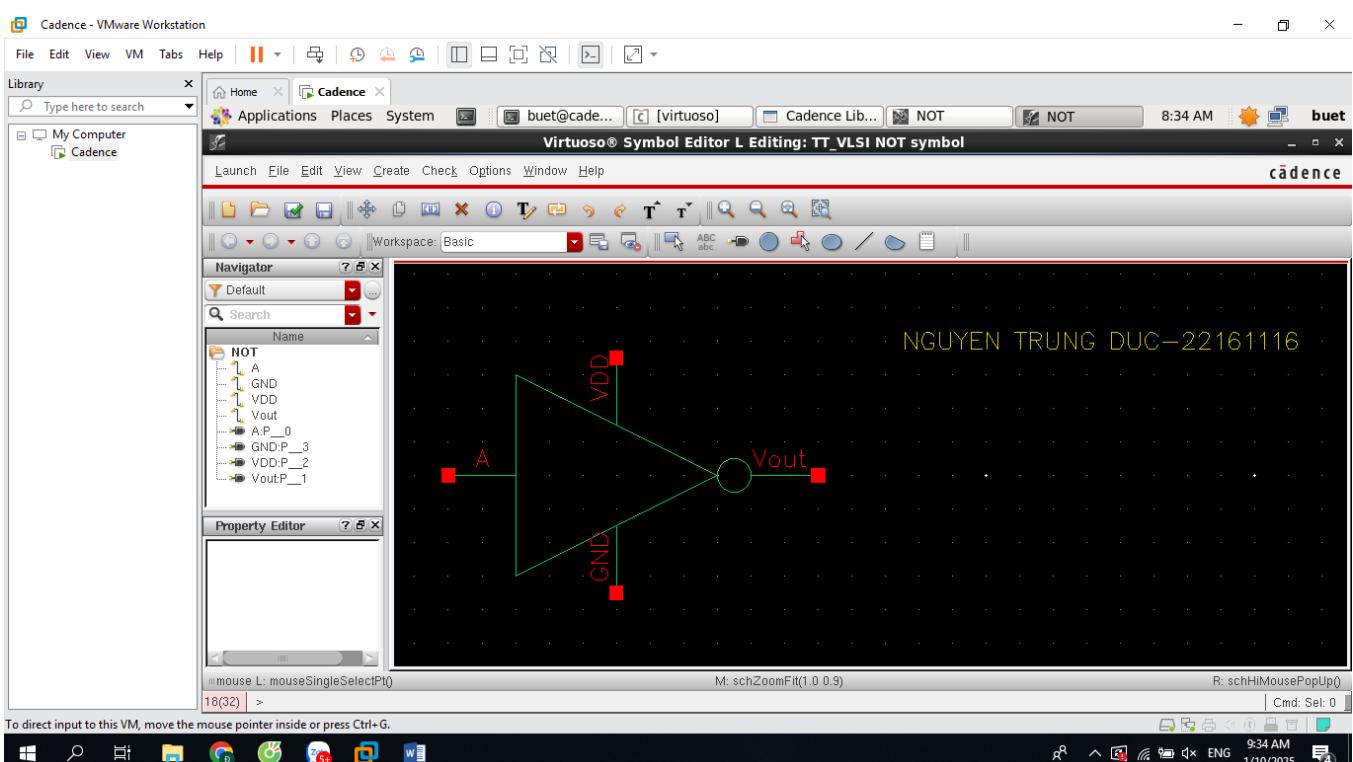
### 3. Thiết kế và mô phỏng inverter

- Mô phỏng logic

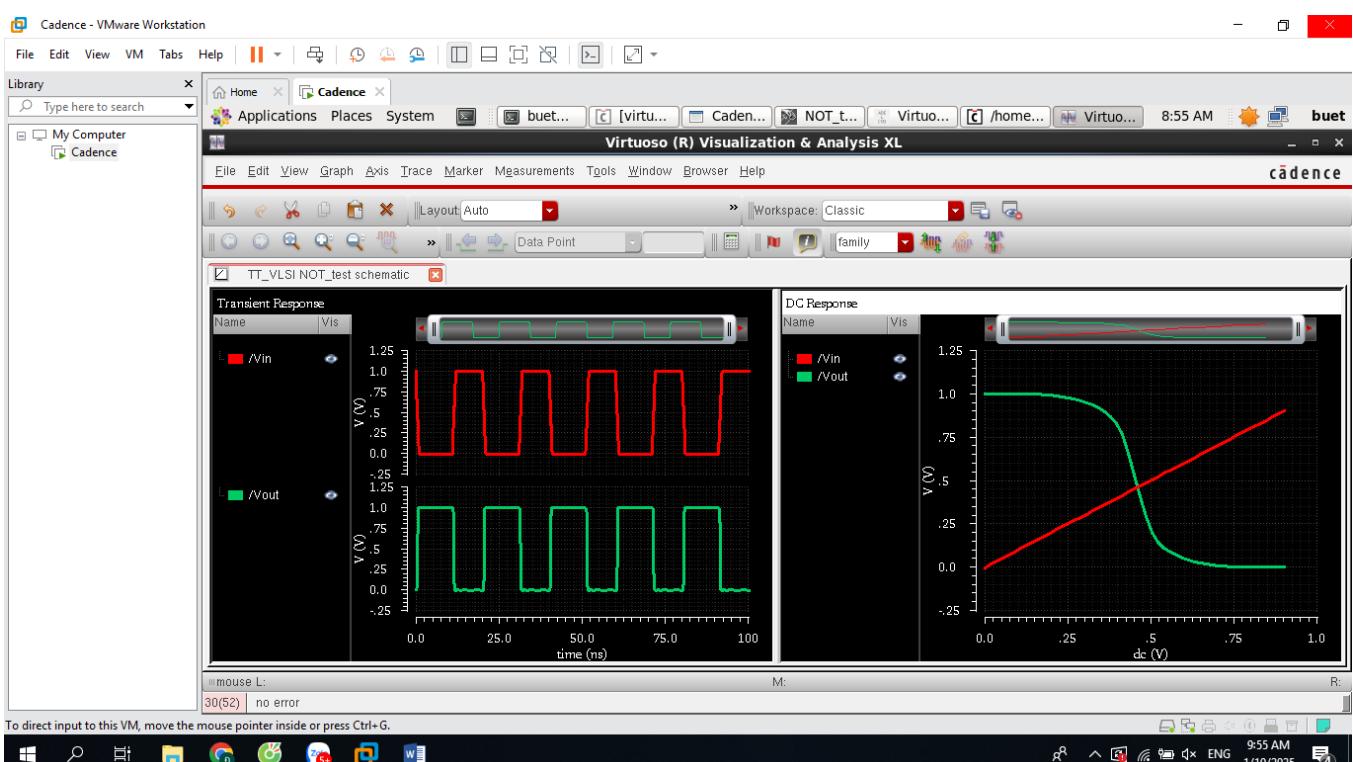
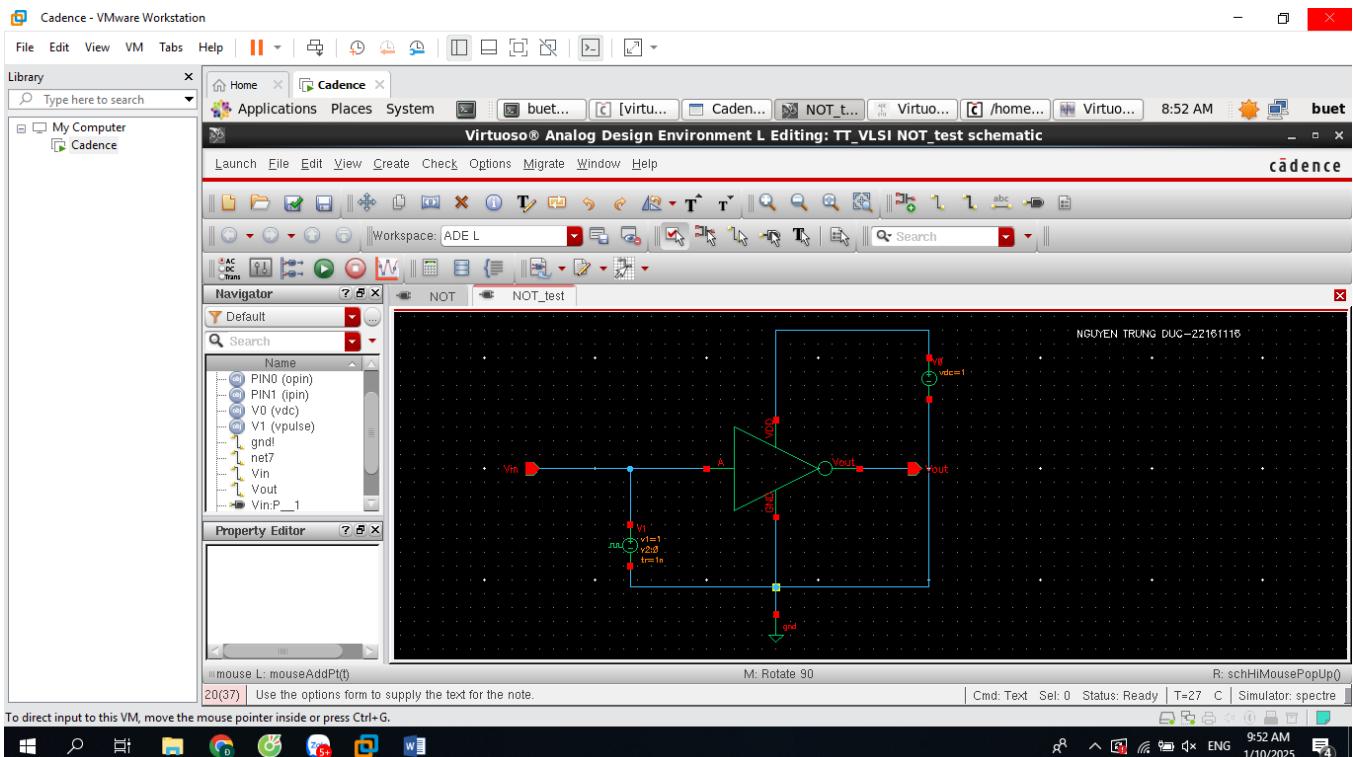
+ Trình bày sơ đồ nguyên lý mạch



+ ) Đóng gói mạch

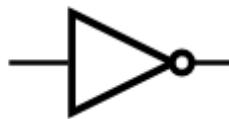


Kết quả



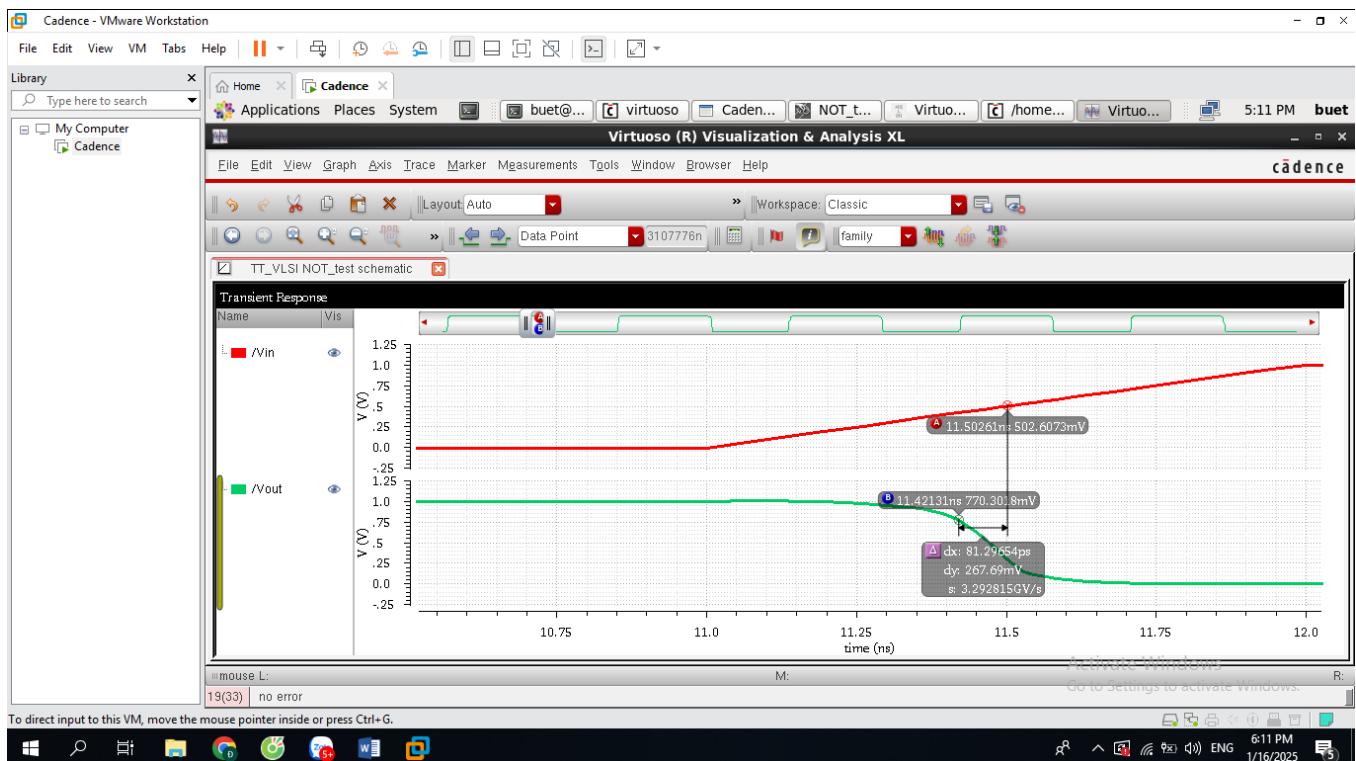
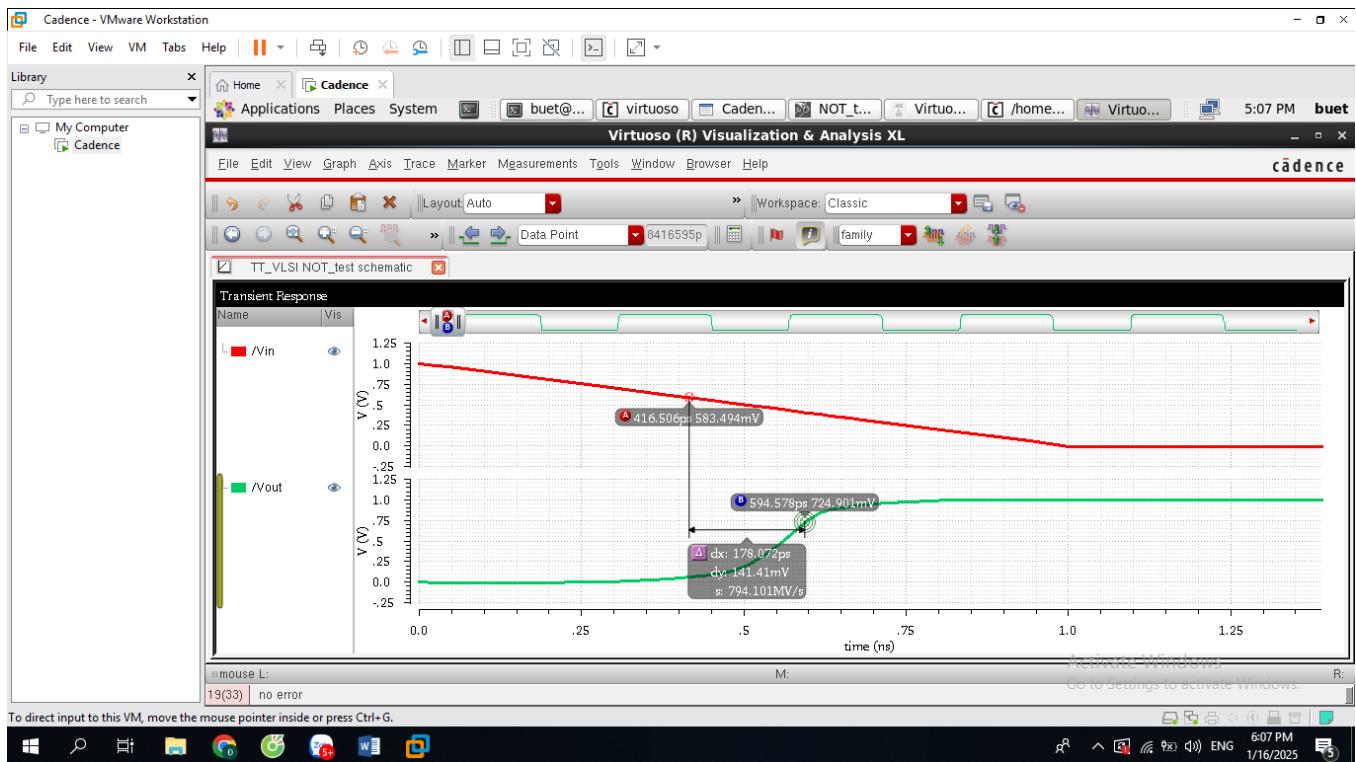
Phân tích

## Cổng NOT



Bảng chân lý - NOT Gate	
Đầu vào A	Đầu ra Q
0	1
1	0

- Ta có hai dạng sóng màu đỏ và màu xanh:
  - Vin (đường đỏ): Dạng sóng đầu vào (xung vuông).
  - Vout (đường xanh lá): Dạng sóng đầu ra, là tín hiệu đảo ngược của đầu vào.
- Phân tích chi tiết:
  - Ban đầu ta cho điện áp ngõ vào (Vin) ở mức thấp tức là bằng 0V thì Vout lên mức cao tức logic 1 (ở hình kết quả là 1V).
  - Sau đó, ta cho điện áp ngõ vào (Vin) lên mức cao tức là bằng 1V thì Vout xuống mức thấp tức logic 0.
  - Do bên mạch vẽ schematic ta xét thông số có biên độ là 1V nên ta thấy khi Vin hay Vout lên mức logic 1 thì chúng sẽ lên 1V, còn khi ở schematic ta xét thông số khác thì lúc đó ngõ ra sẽ có điện áp mức logic 1 khác.
  - Tín hiệu đầu ra có dạng xung vuông, không bị méo hoặc trễ đáng kể.
- DC Response (Phản hồi DC - bên phải)
  - Đồ thị này thể hiện đặc tuyến truyền của cổng NOT (Vout theo Vin).
  - Đường màu đỏ (Vin) là điện áp đầu vào thay đổi từ 0V đến 1.2V.
  - Đường màu xanh (Vout) là điện áp đầu ra tương ứng.
  - Khi Vin nhỏ, Vout cao, và ngược lại.
  - Điểm giữa đường cong là điện áp ngưỡng (Vth) của cổng NOT, nơi Vout chuyển đổi nhanh.
- Mô phỏng delay



Phân tích:

- Ở hình thứ nhất:
  - *V<sub>in</sub>* chuyển mức từ cao xuống thấp tại khoảng 416.5 ps.
  - *V<sub>out</sub>* thay đổi rõ rệt tại 594.57 ps.
  - Khoảng thời gian trễ giữa *V<sub>in</sub>* và *V<sub>out</sub>* có thể tính bằng hiệu của hai thời điểm này.
- Ở hình thứ hai:
  - *V<sub>in</sub>* tăng từ mức thấp lên mức cao tại 11.5 ns.
  - *V<sub>out</sub>* có thay đổi rõ rệt sau đó tại 11.41 ns.

- Delay có thể xác định từ độ dốc của đồ thị và sự dịch chuyển giữa hai tín hiệu.
- Nhận xét kết quả:

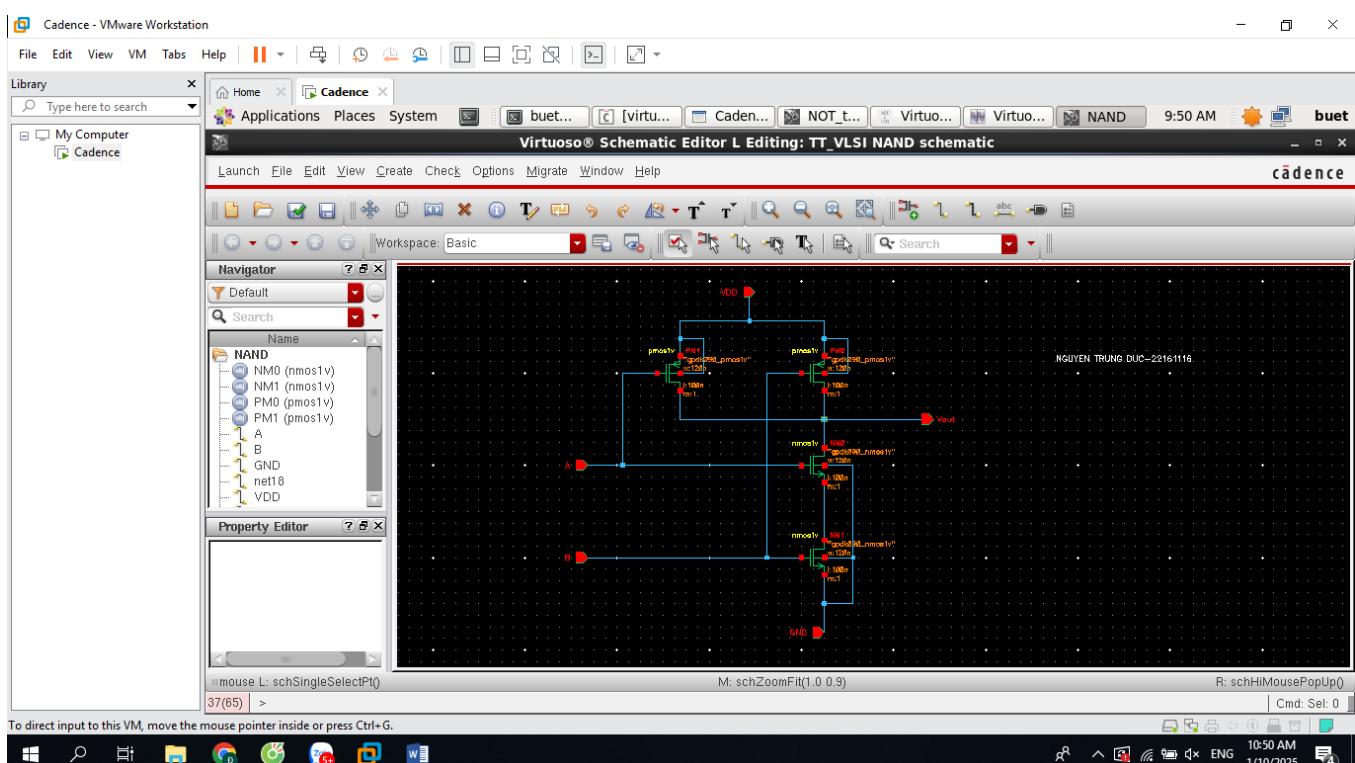
- Khoảng thời gian trễ (Propagation Delay) được đo từ khi đầu vào thay đổi đến khi đầu ra đạt được 50% giá trị chuyển đổi.
- Delay này phản ánh tốc độ truyền mạch của công nghệ logic trong công nghệ bán dẫn 90nm.
- Nếu độ trễ lớn, có thể cần tối ưu hóa thiết kế bằng cách thay đổi kích thước transistor hoặc điều chỉnh điện áp nguồn.

- Thiết kế, tạo symbol và mô phỏng logic cho các công NAND, AND, NOR, OR, EX-NOR, EX-OR.

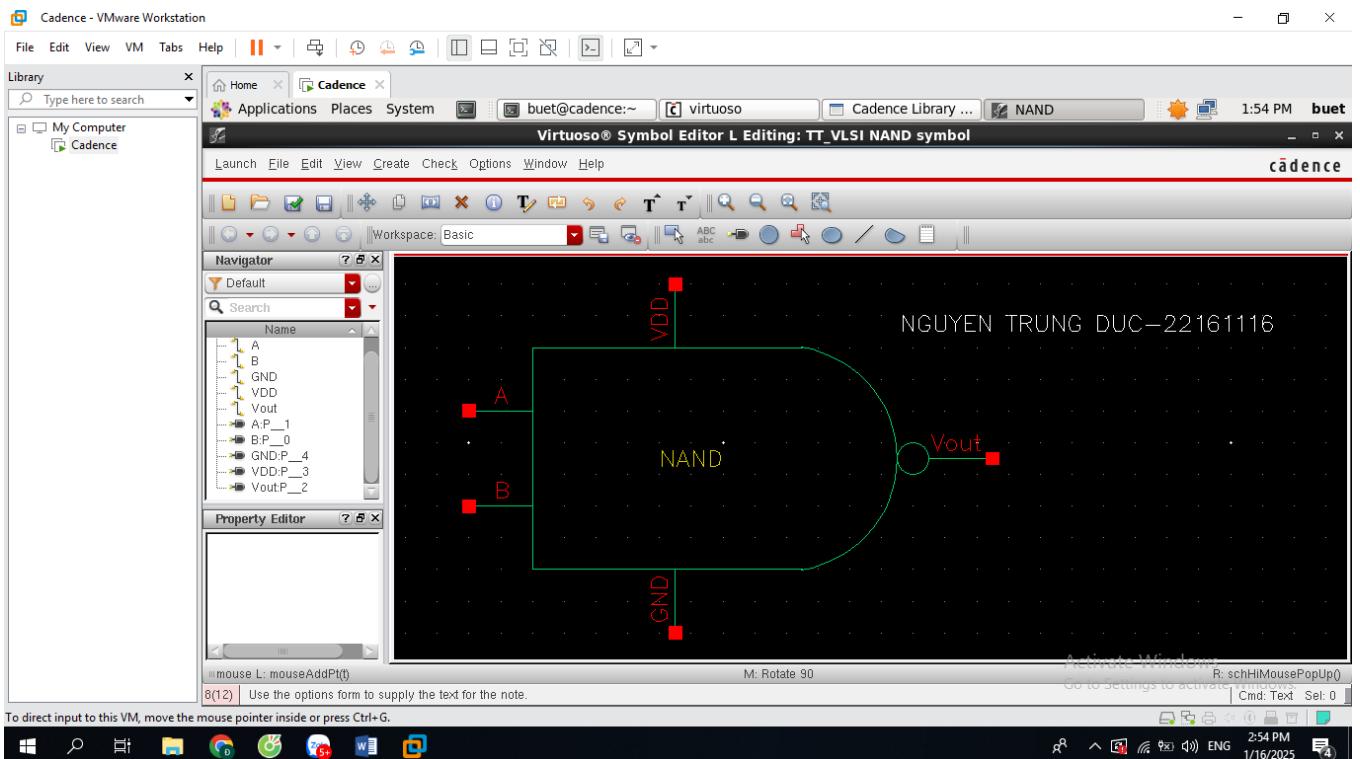
Sinh viên trình bày schematic

NAND

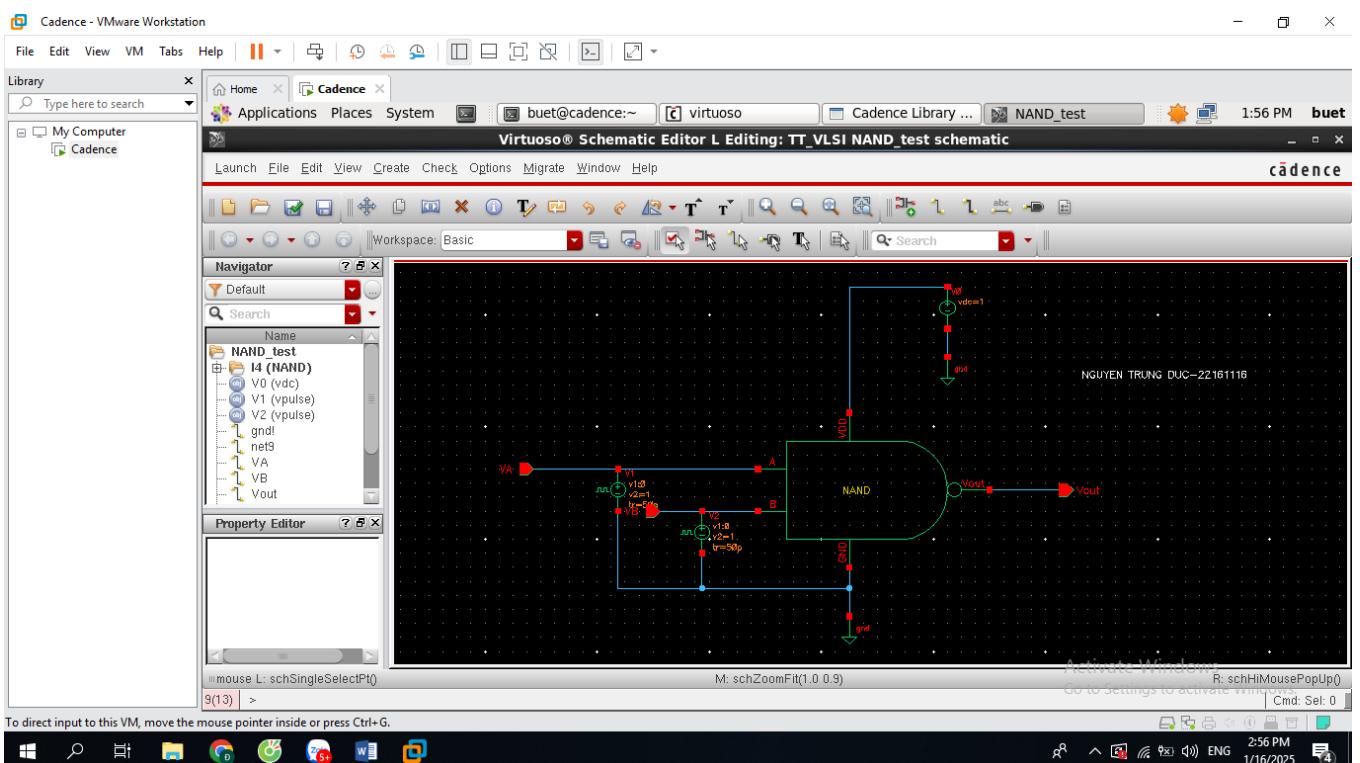
+ Trình bày sơ đồ nguyên lý mạch

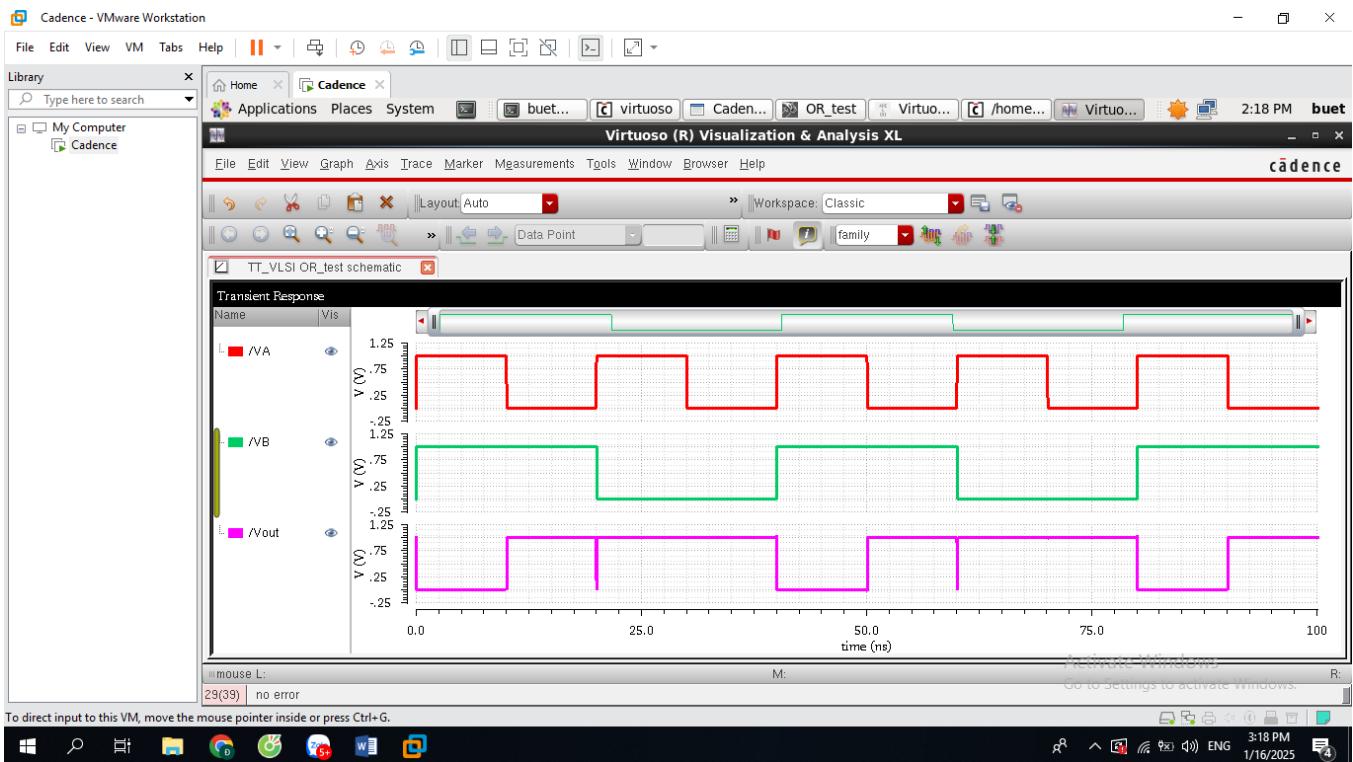


+ Đóng gói mạch



### Kết quả mô phỏng có phân tích





Phân tích

Công NAND có phép toán logic:

$$V_{out} = \overline{A \cdot B}$$

## Công NAND



Bảng chân lý - Công NAND

Đầu vào A	Đầu vào B	Đầu ra Q
0	0	1
0	1	1
1	0	1
1	1	0

Ta có thể thấy kết quả :

- Ban đầu ta cho VA (màu đỏ) ở mức 1, VB (màu xanh) ở mức 1 thì cho ra Vout (màu hồng) xuống mức 0.

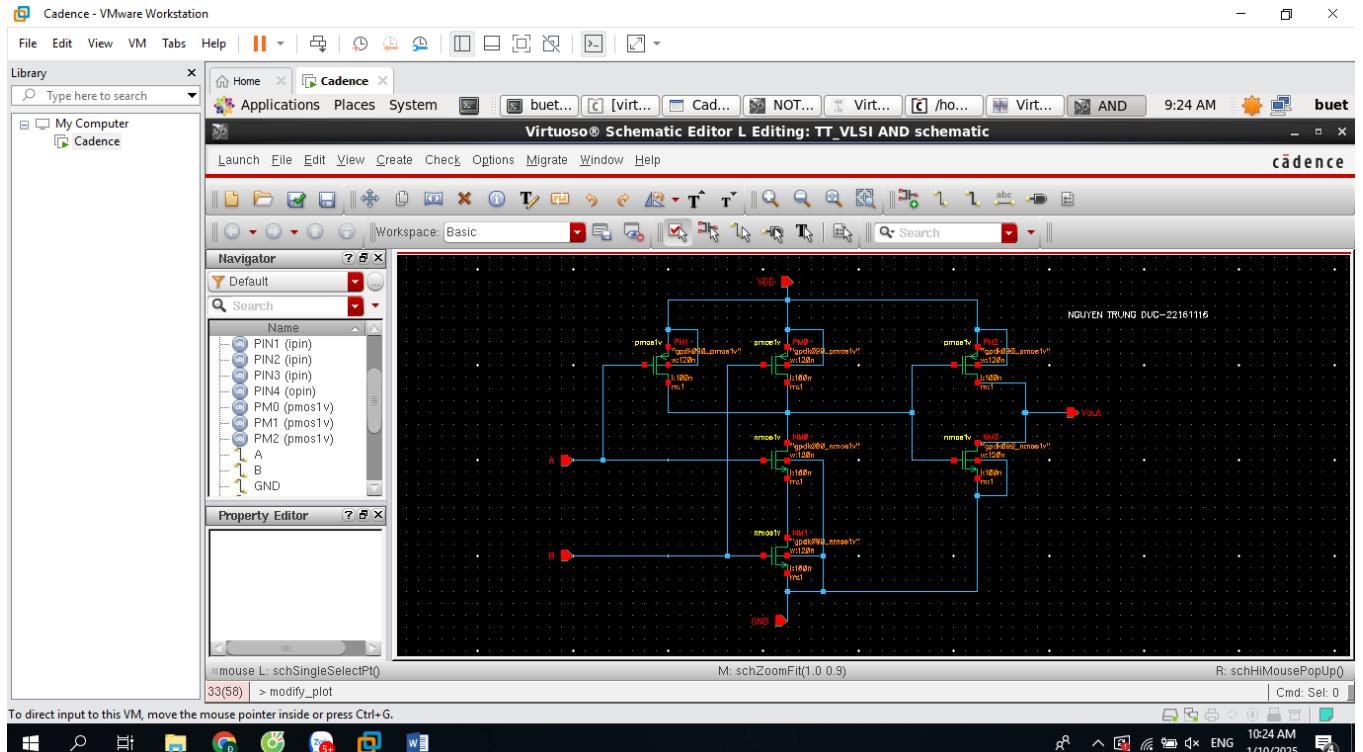
- Sau đó ta cho VA (màu đỏ) xuống mức 0, VB (màu xanh) vẫn giữ ở mức 1 thì cho ra Vout (màu hồng) lên mức 1.
- Tiếp tục ta lại cho VA (màu đỏ) lên mức 1, VB (màu xanh) xuống mức 0 thì cho ra Vout (màu hồng) ở mức 1.
- Cuối cùng ta cho cả VA và VB xuống mức 0 thì thấy rằng Vout lên mức 1.

Vậy: - Mạch NAND hoạt động chính xác theo nguyên lý logic.

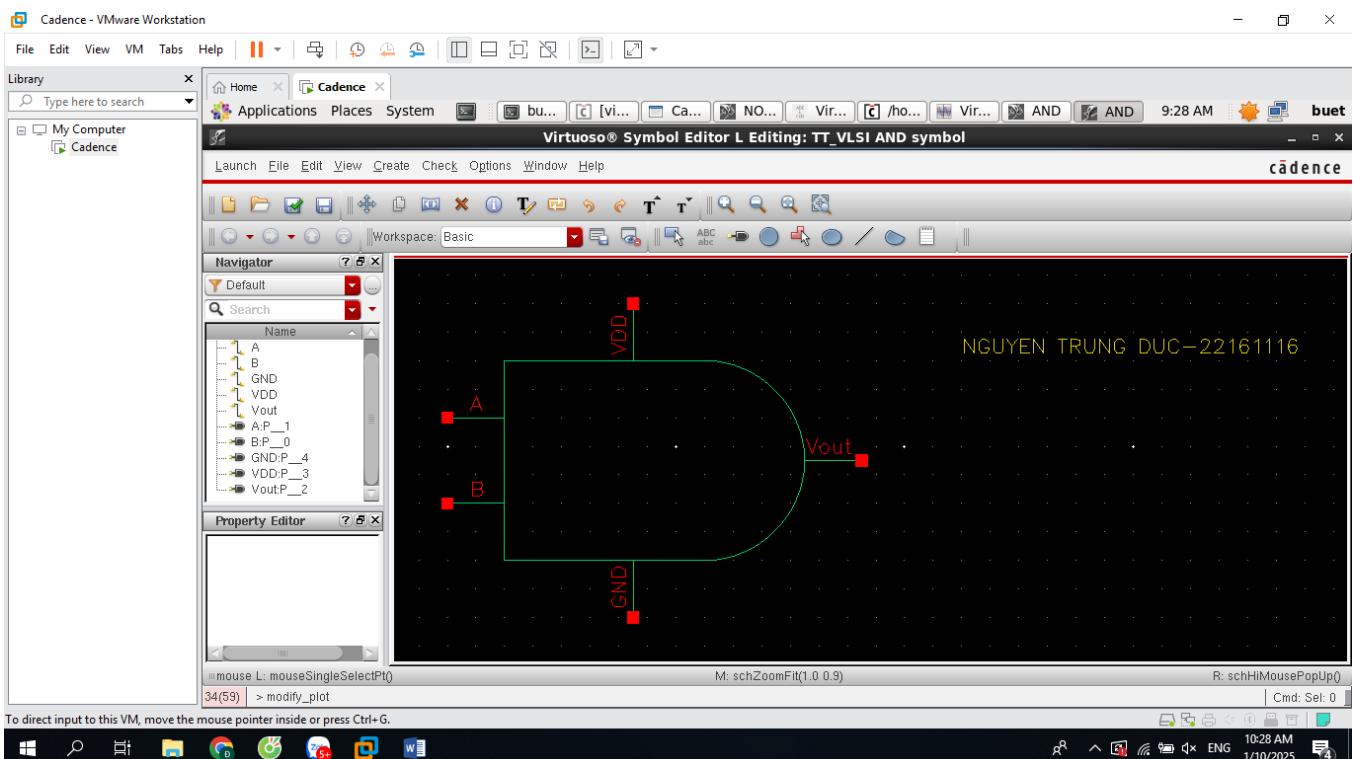
- Có độ trễ nhỏ, có thể đo đặc chính xác nếu cần tối ưu hóa.
- Dữ liệu dạng sóng hợp lý, phản ánh đúng tính chất của NAND.

## AND

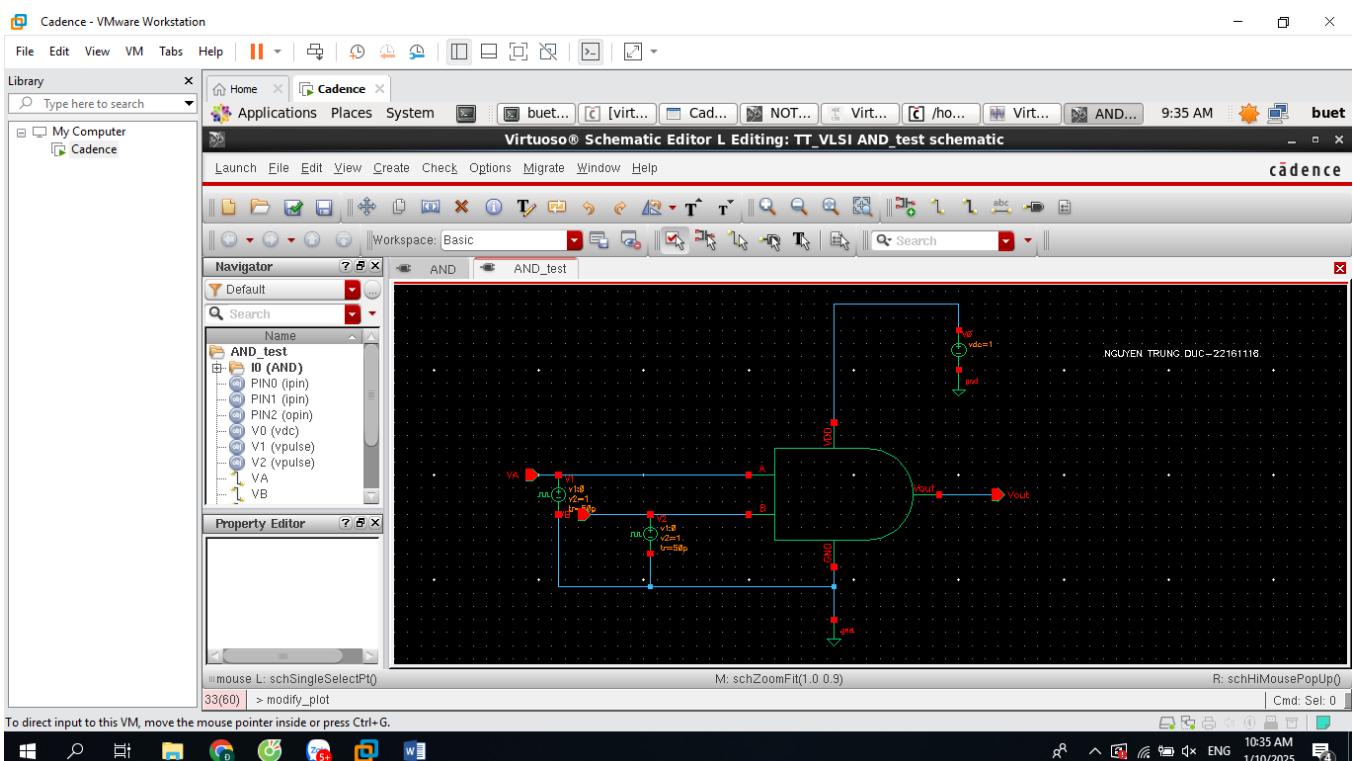
### + Trình bày sơ đồ nguyên lý mạch

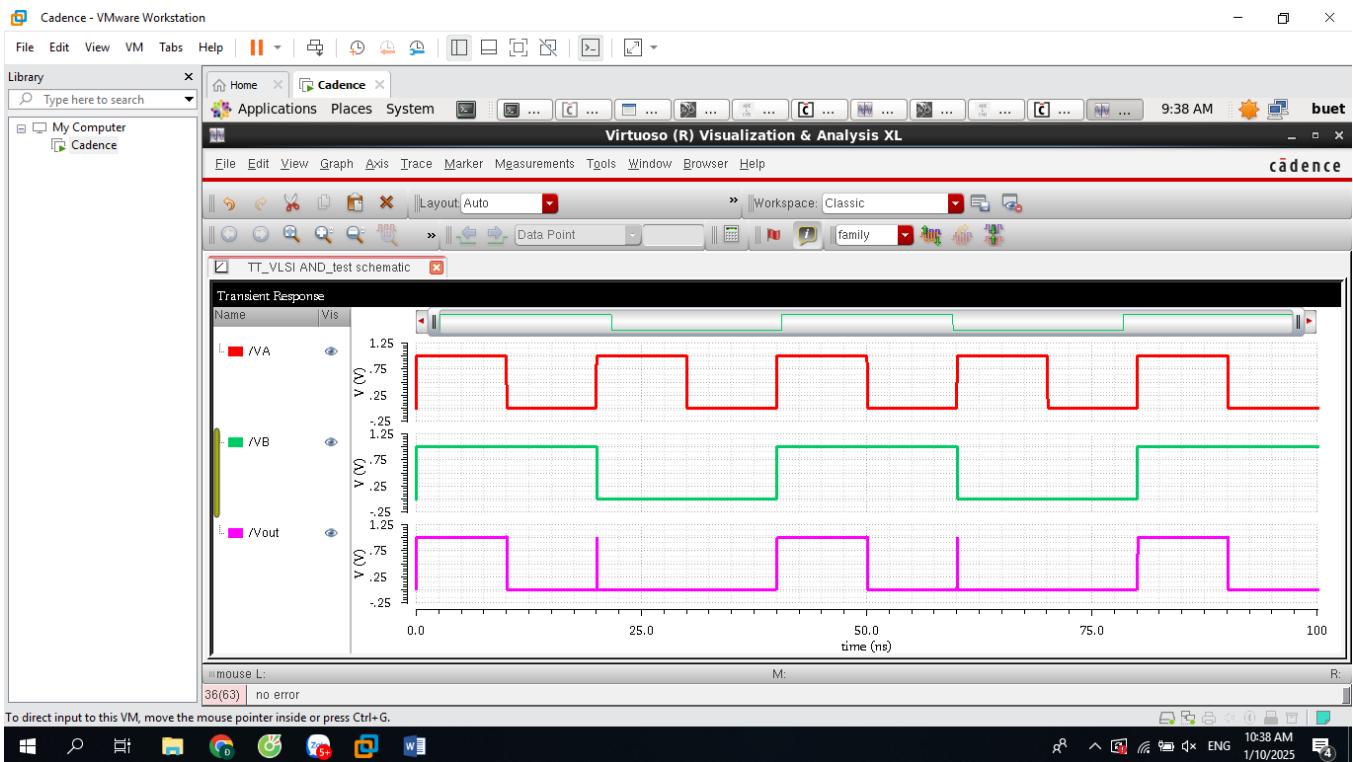


### + Đóng gói mạch



### Kết quả mô phỏng có phân tích

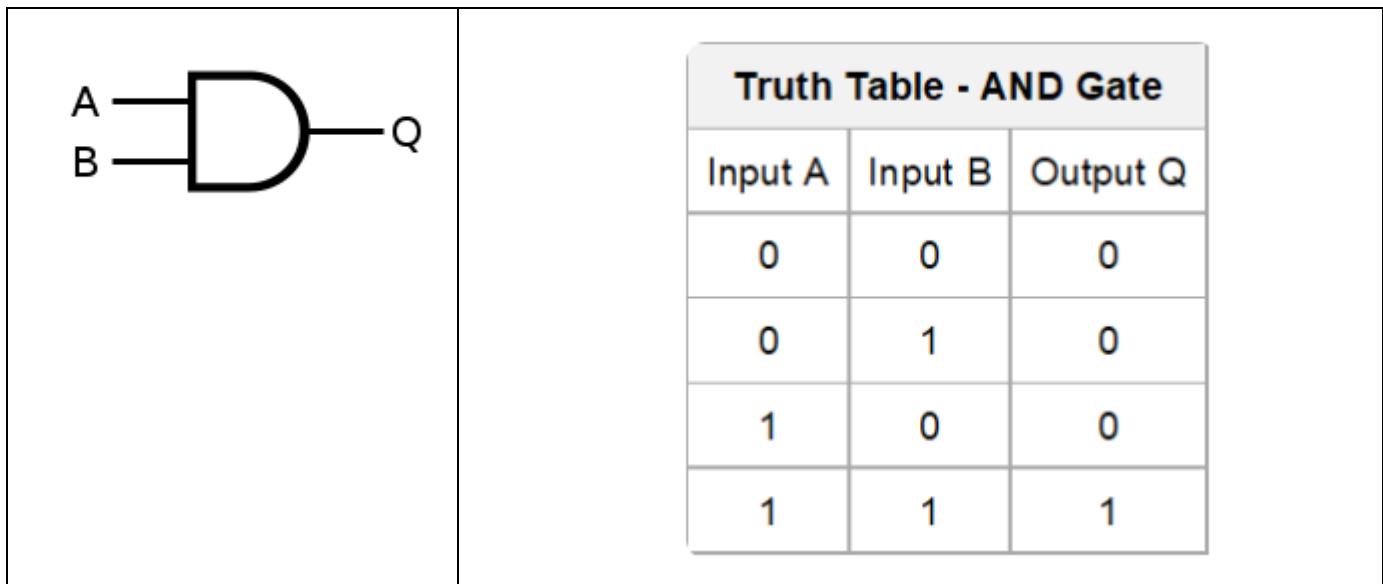




Phân tích

Công AND thực hiện phép toán logic:

$$V_{\text{out}} = A \cdot B$$



Ta có thể thấy kết quả :

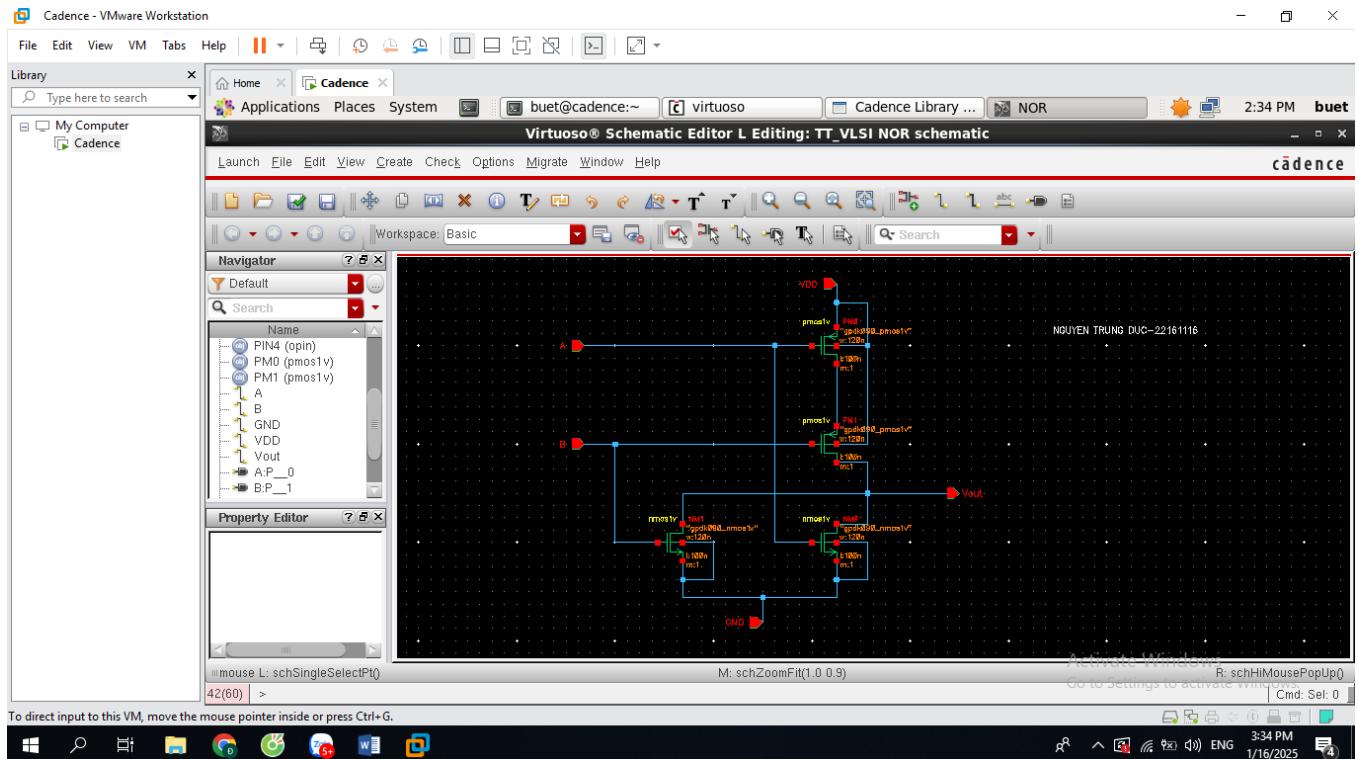
- Ban đầu ta cho VA (màu đỏ) ở mức 1, VB (màu xanh) ở mức 1 thì cho ra Vout (màu hồng) ở mức 1.
- Sau đó ta cho VA (màu đỏ) xuống mức 0, VB (màu xanh) vẫn giữ ở mức 1 thì cho ra Vout (màu hồng) xuống mức 0.
- Tiếp tục ta lại cho VA (màu đỏ) lên mức 1, VB (màu xanh) xuống mức 0 thì cho ra Vout (màu hồng) xuống mức 0.
- Cuối cùng ta cho cả VA và VB xuống mức 0 thì thấy rằng Vout xuống mức 0.

Vậy: - Mạch AND hoạt động chính xác theo nguyên lý logic.

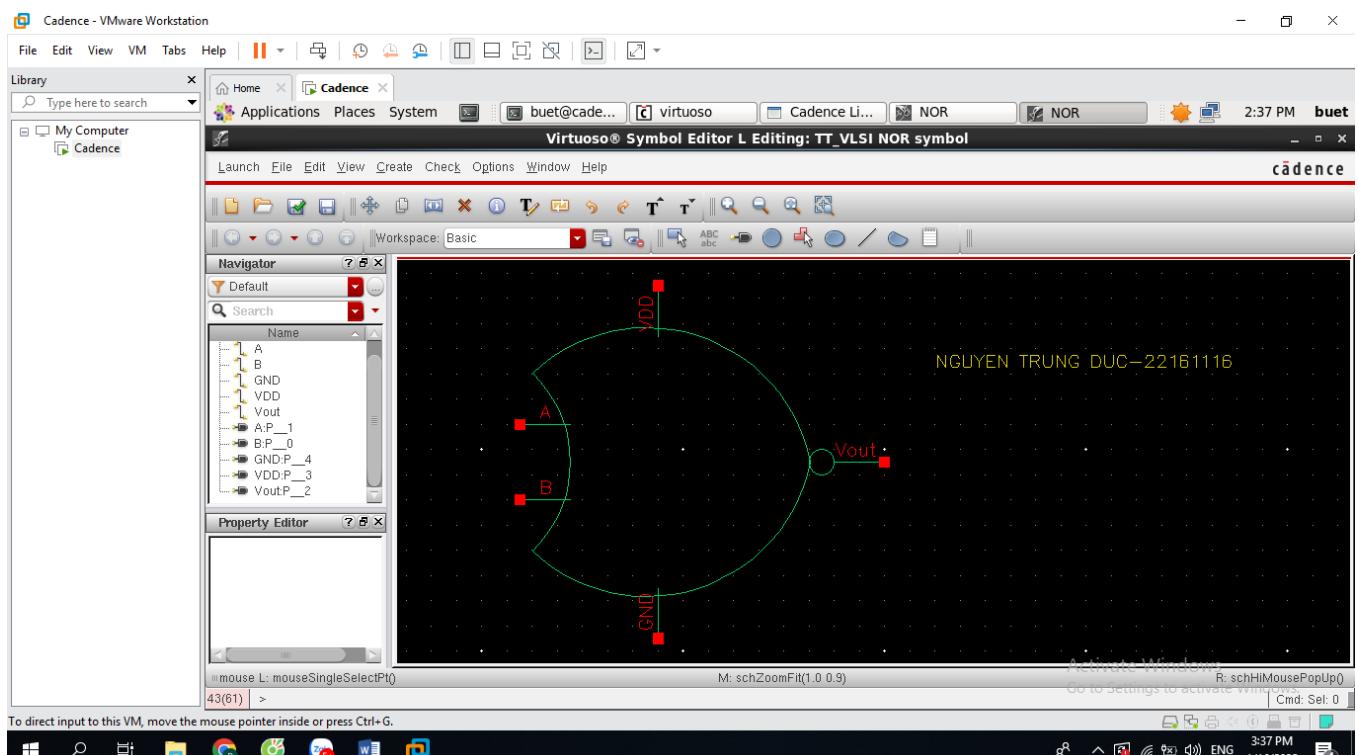
- Có độ trễ nhỏ, có thể đo chính xác hơn nếu cần tối ưu hóa.
- Dữ liệu dạng sóng hợp lý, phản ánh đúng tính chất của AND.

## NOR

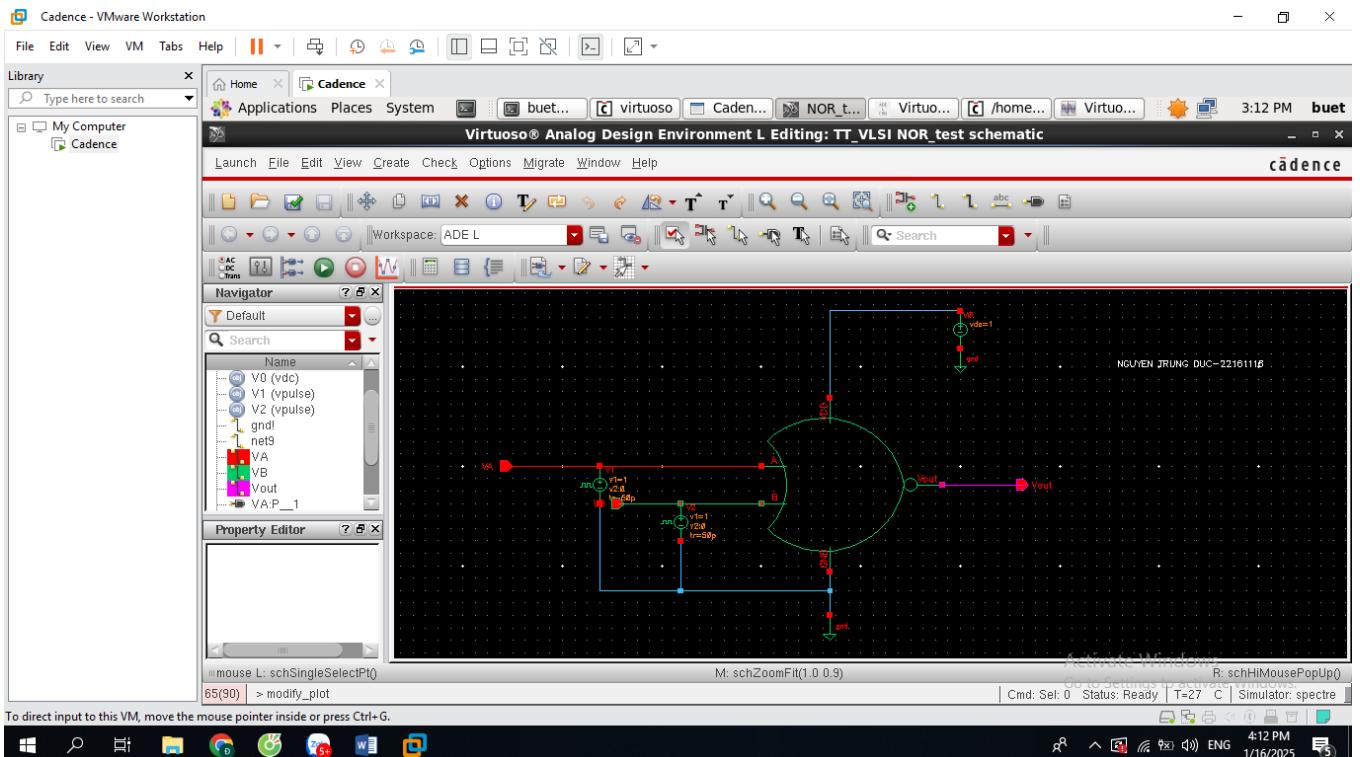
+ ) Trình bày sơ đồ nguyên lý mạch



+ ) Đóng gói mạch



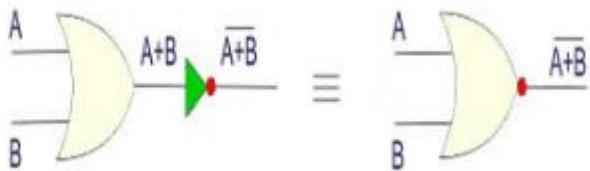
## Kết quả mô phỏng có phân tích



## Phân tích

Công NOR thực hiện phép toán logic:

$$V_{\text{out}} = \overline{A + B}$$



Inputs		Output
A	B	X = $\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

Ta có thể thấy kết quả :

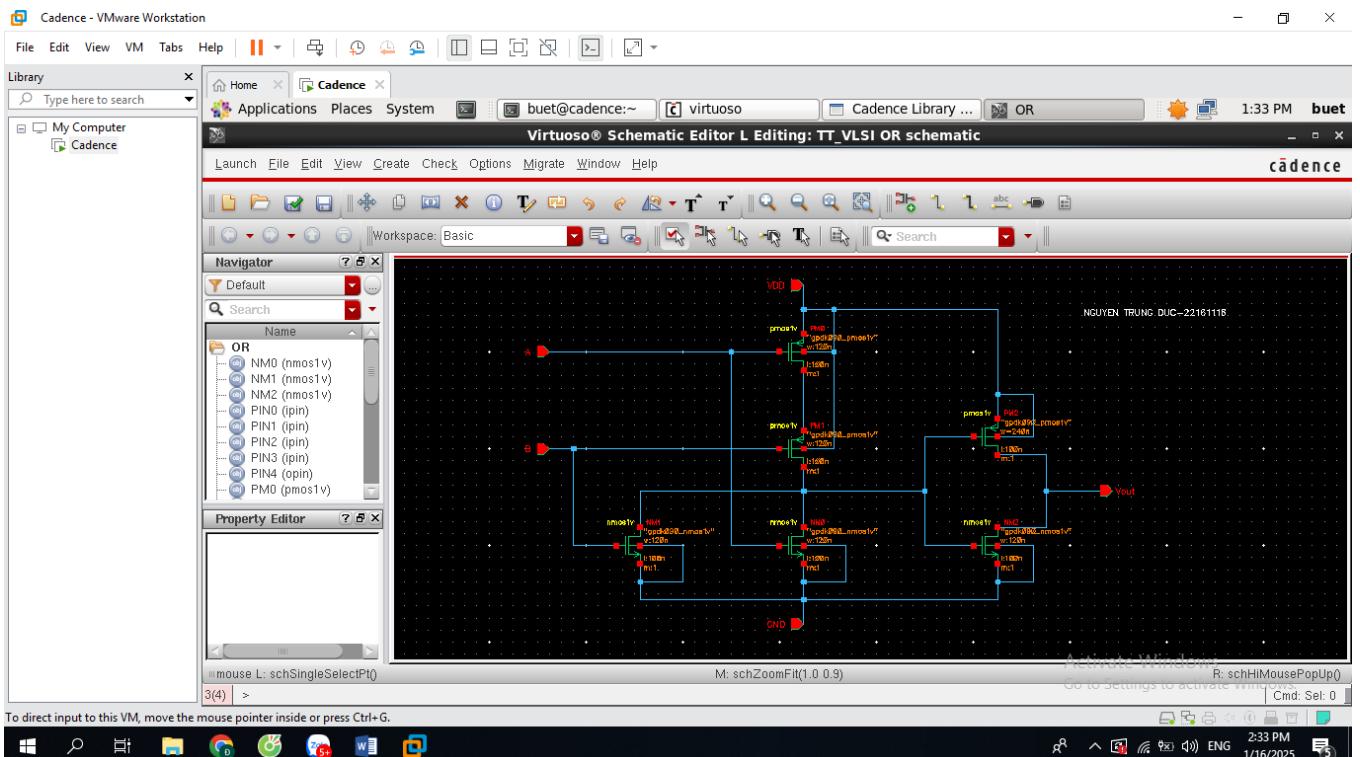
- Ban đầu ta cho VA (màu đỏ) ở mức 0, VB (màu xanh) ở mức 0 thì cho ra Vout (màu hồng) xuống mức 1.
- Sau đó ta cho VA (màu đỏ) lên mức 1, VB (màu xanh) vẫn ở mức 0 thì cho ra Vout (màu hồng) xuống mức 0.
- Tiếp tục ta lại cho VA (màu đỏ) ở mức 1, VB (màu xanh) lên mức 1 thì cho ra Vout (màu hồng) ở mức 0.
- Cuối cùng ta cho VA (màu đỏ) xuống mức 0, VB (màu xanh) vẫn ở mức 1 thì cho ra Vout (màu hồng) ở mức 0.

Vậy: - Mạch NOR hoạt động chính xác theo nguyên lý logic.

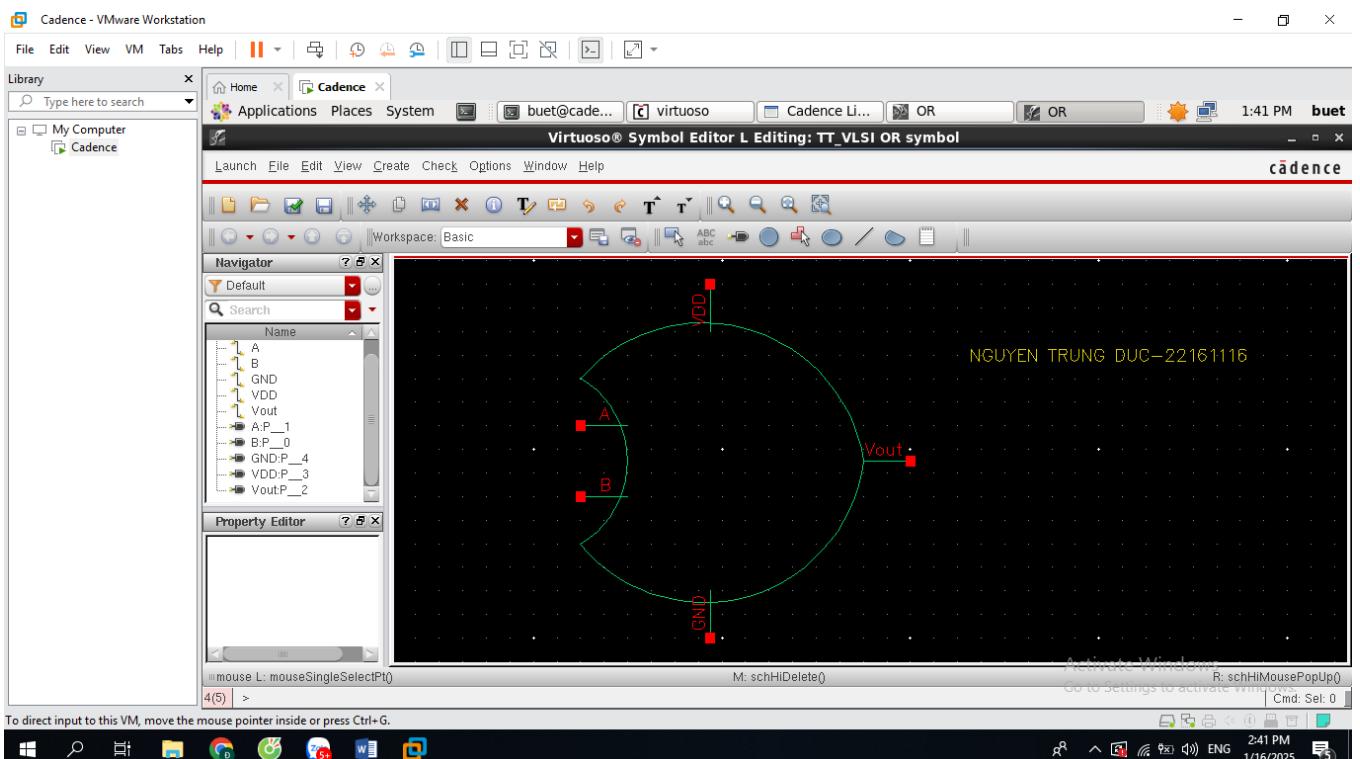
- Có độ trễ nhỏ, có thể đo chính xác hơn nếu cần tối ưu hóa.
- Dữ liệu dạng sóng hợp lý, phản ánh đúng tính chất của NOR.

OR

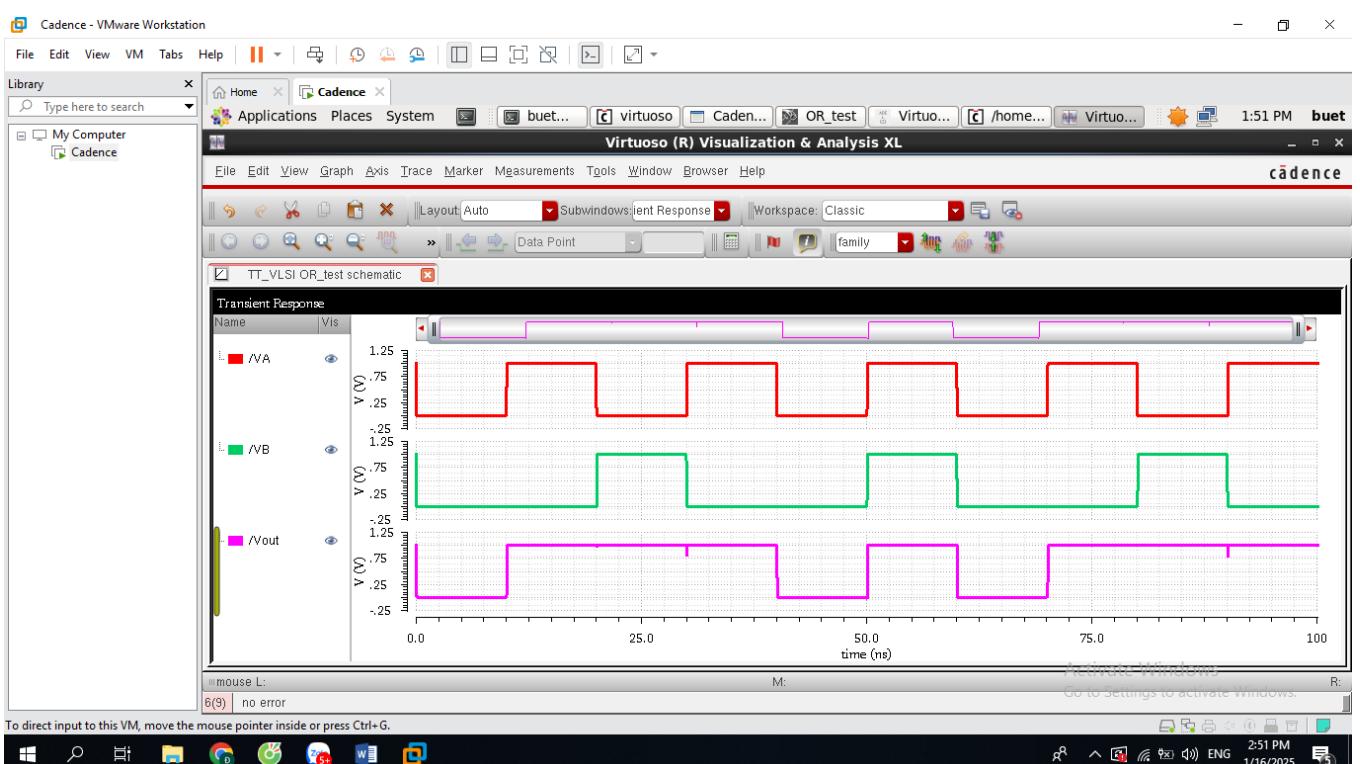
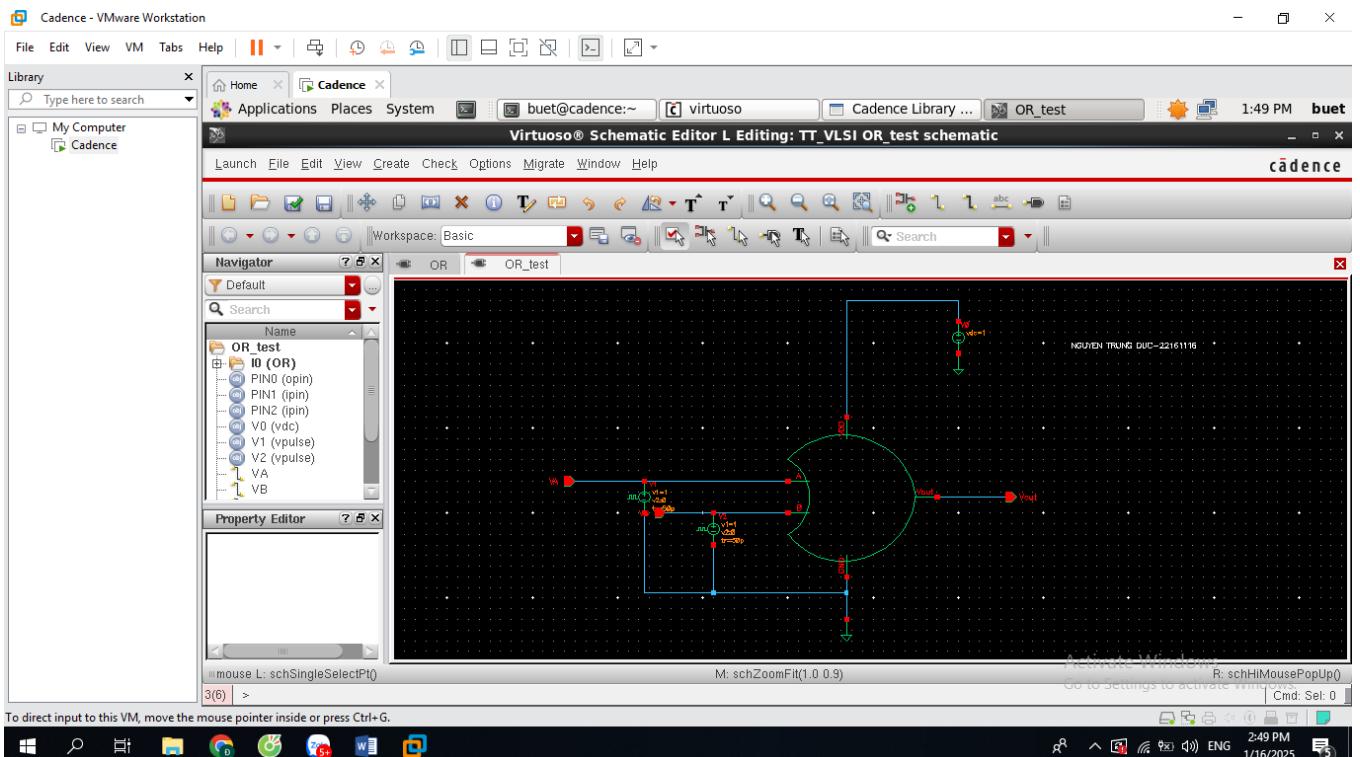
+ ) Trình bày sơ đồ nguyên lý mạch



+ ) Đóng gói mạch



Kết quả mô phỏng có phân tích



Phân tích  
Circuit OR thực hiện phép toán logic:

$$V_{\text{out}} = A + B$$

## Cỗng OR



Bảng chân lý - Cỗng OR		
Đầu vào A	Đầu vào B	Đầu ra Q
0	0	0
0	1	1
1	0	1
1	1	1

Ta có thể thấy kết quả :

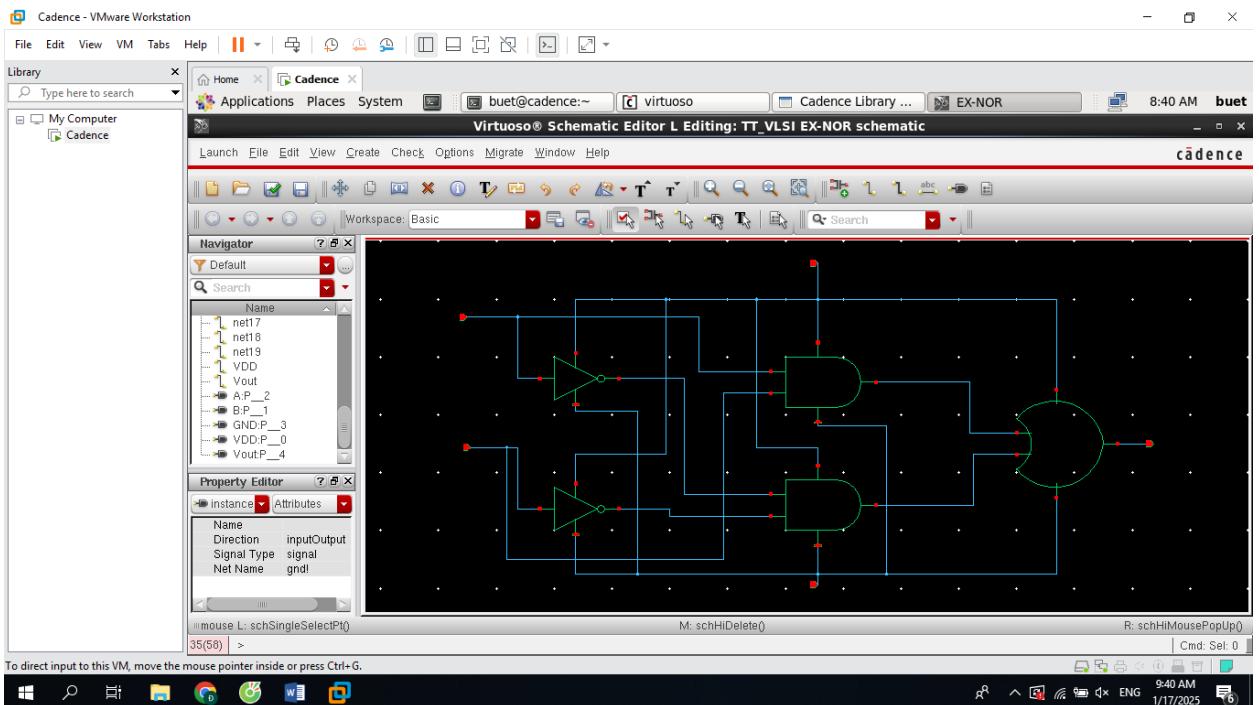
- Ban đầu ta cho VA (màu đỏ) ở mức 0, VB (màu xanh) ở mức 0 thì cho ra Vout (màu hồng) ở mức 0.
- Sau đó ta cho VA (màu đỏ) lên mức 1, VB (màu xanh) vẫn ở mức 0 thì cho ra Vout (màu hồng) lên mức 1.
- Tiếp tục ta lại cho VA (màu đỏ) xuống mức 0, VB (màu xanh) lên mức 1 thì cho ra Vout (màu hồng) lên mức 1.
- Cuối cùng ta cho VA (màu đỏ) lên mức 1, VB (màu xanh) xuống mức 0 thì cho ra Vout (màu hồng) lên mức 1.

Vậy: - Mạch OR hoạt động chính xác theo nguyên lý logic.

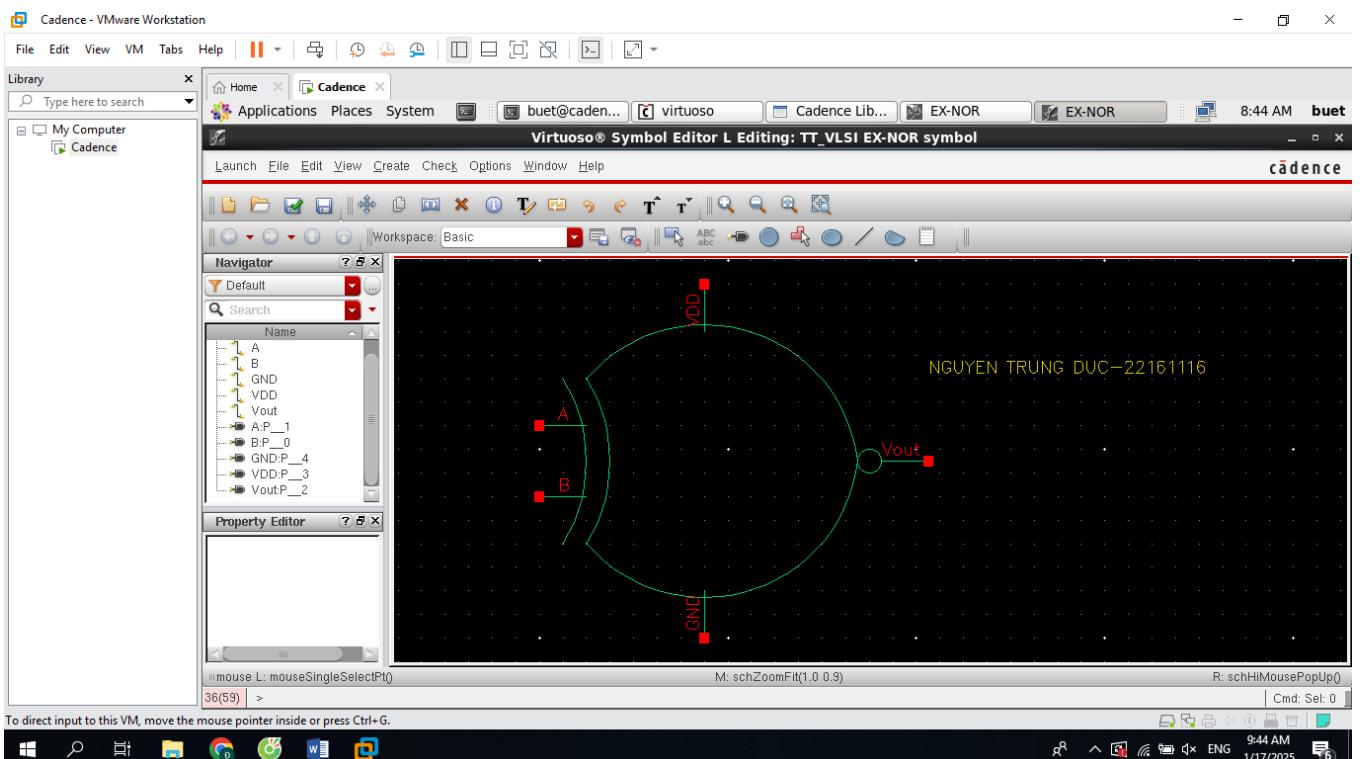
- Có độ trễ nhỏ, nhưng không ảnh hưởng đáng kể đến kết quả logic.
- Dữ liệu dạng sóng hợp lý, phản ánh đúng tính chất của OR.

## EX-NOR

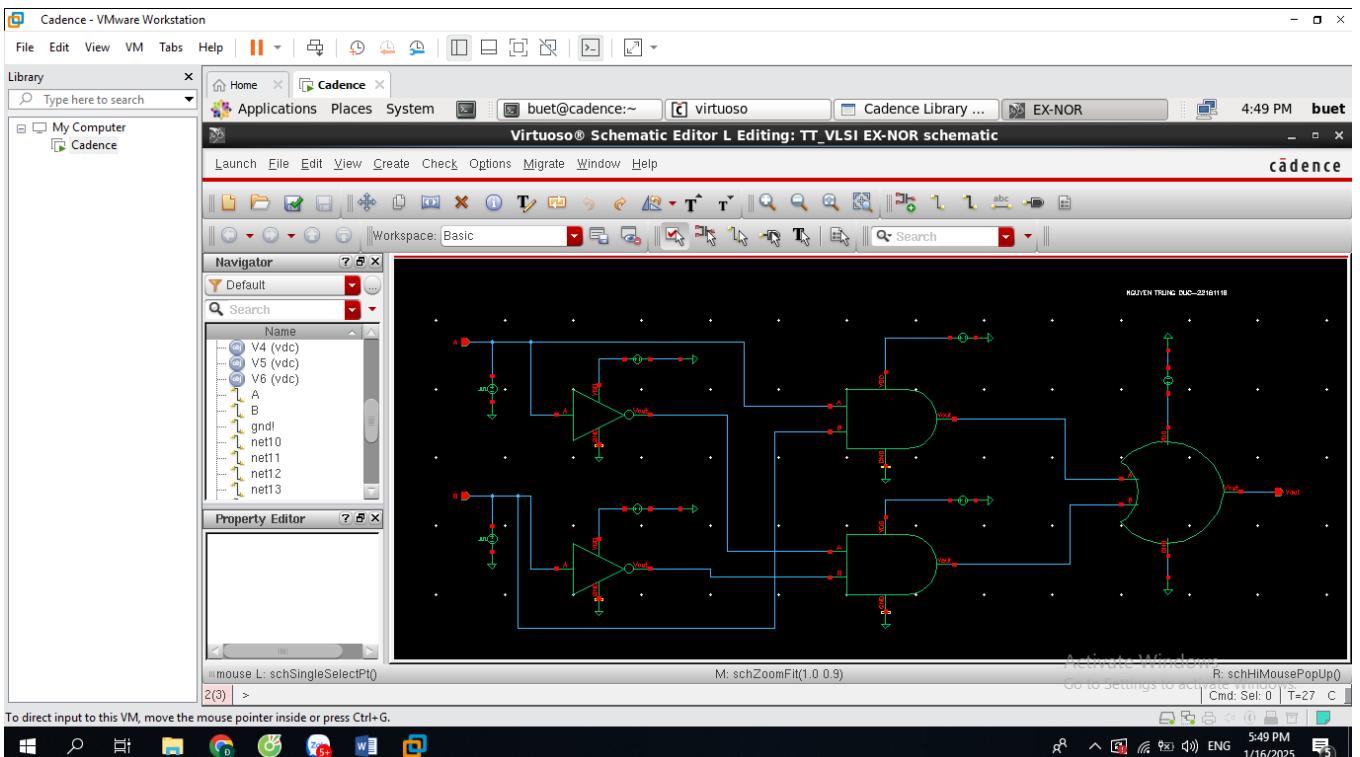
+ ) Trình bày sơ đồ nguyên lý mạch



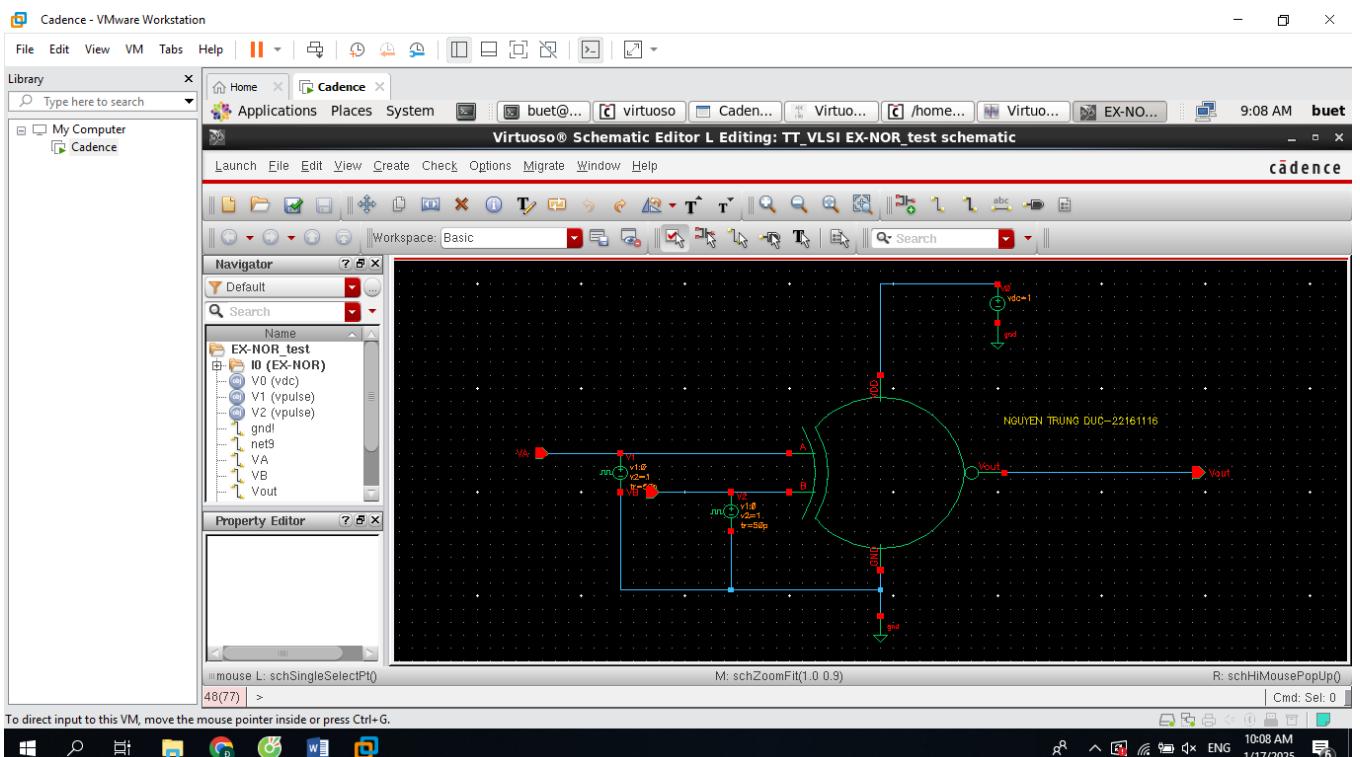
+ Đóng gói



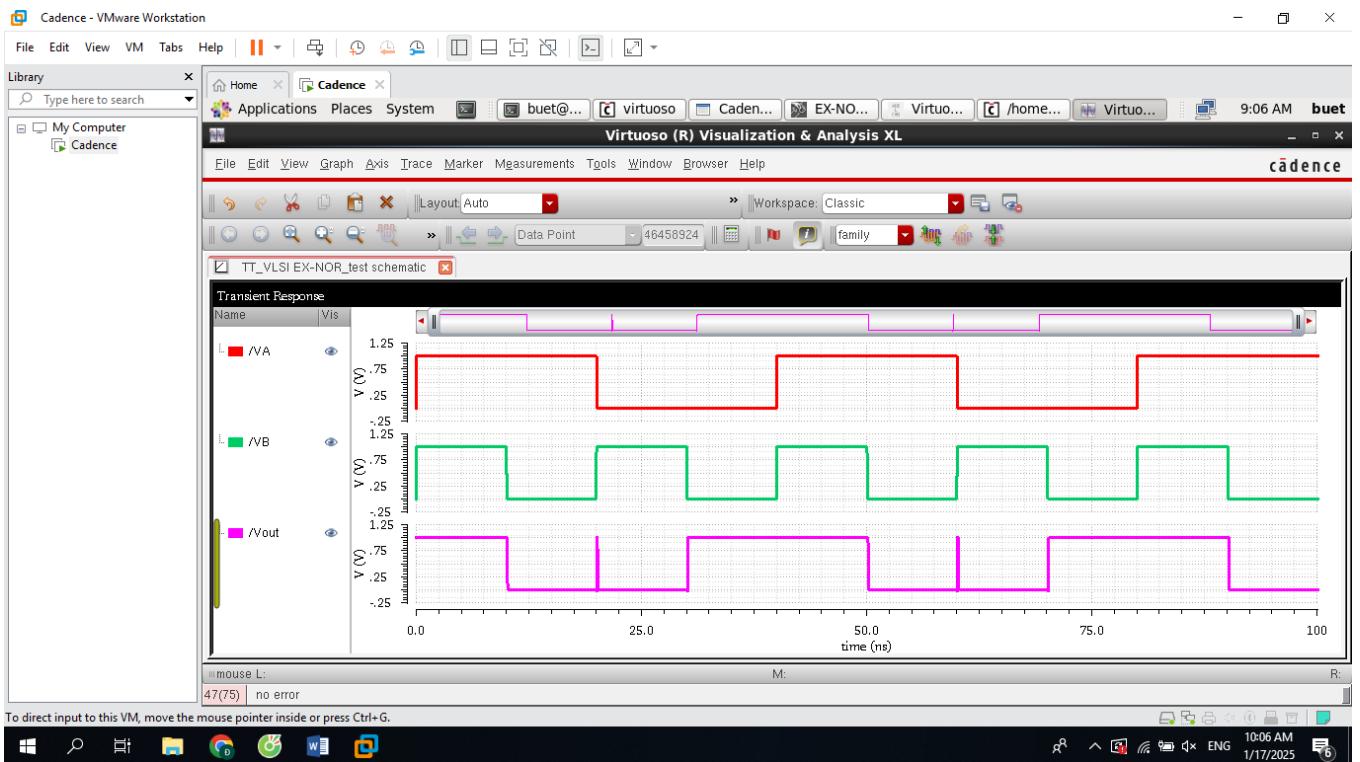
Kết quả mô phỏng có phân tích  
- Vẽ trực tiếp



- Vẽ qua đóng gói

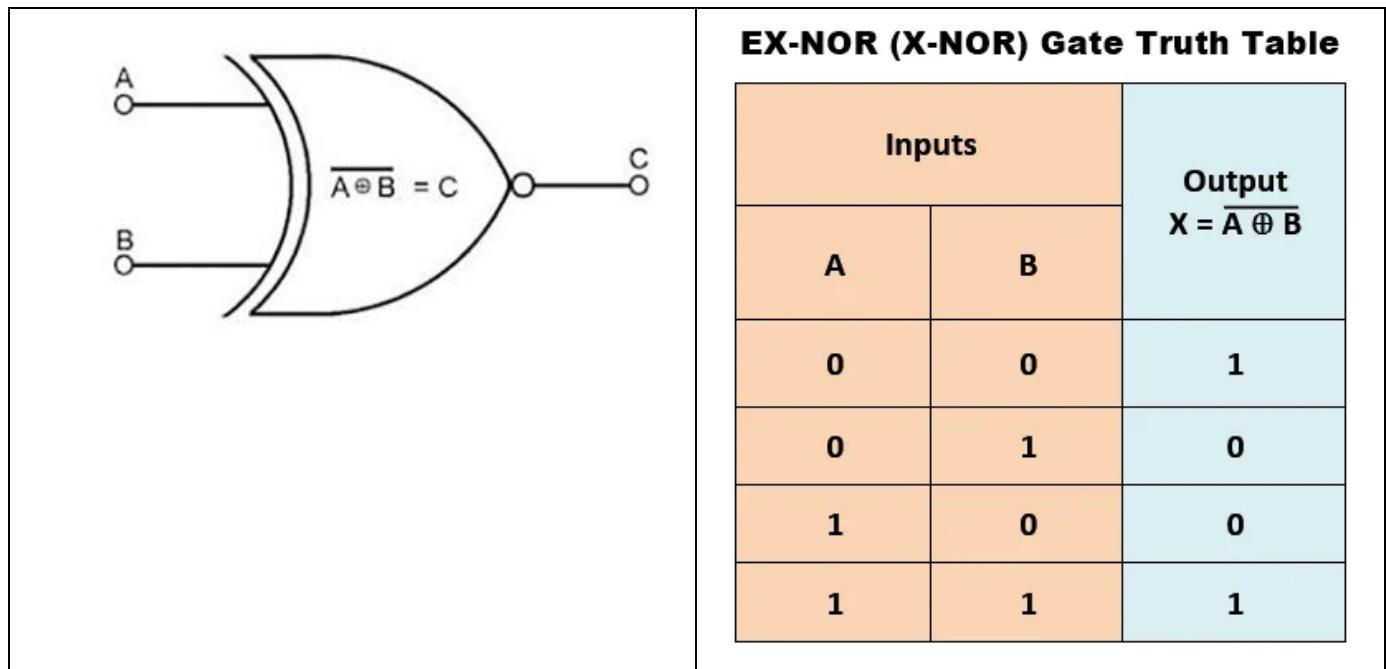


Kết quả



Phân tích  
Công EX-NOR thực hiện phép toán logic:

$$V_{out} = \overline{A \oplus B}$$



Ta có thể thấy kết quả :

- Ban đầu ta cho VA (màu đỏ) ở mức 1, VB (màu xanh) ở mức 1 thì cho ra Vout (màu hồng) ở mức 1.

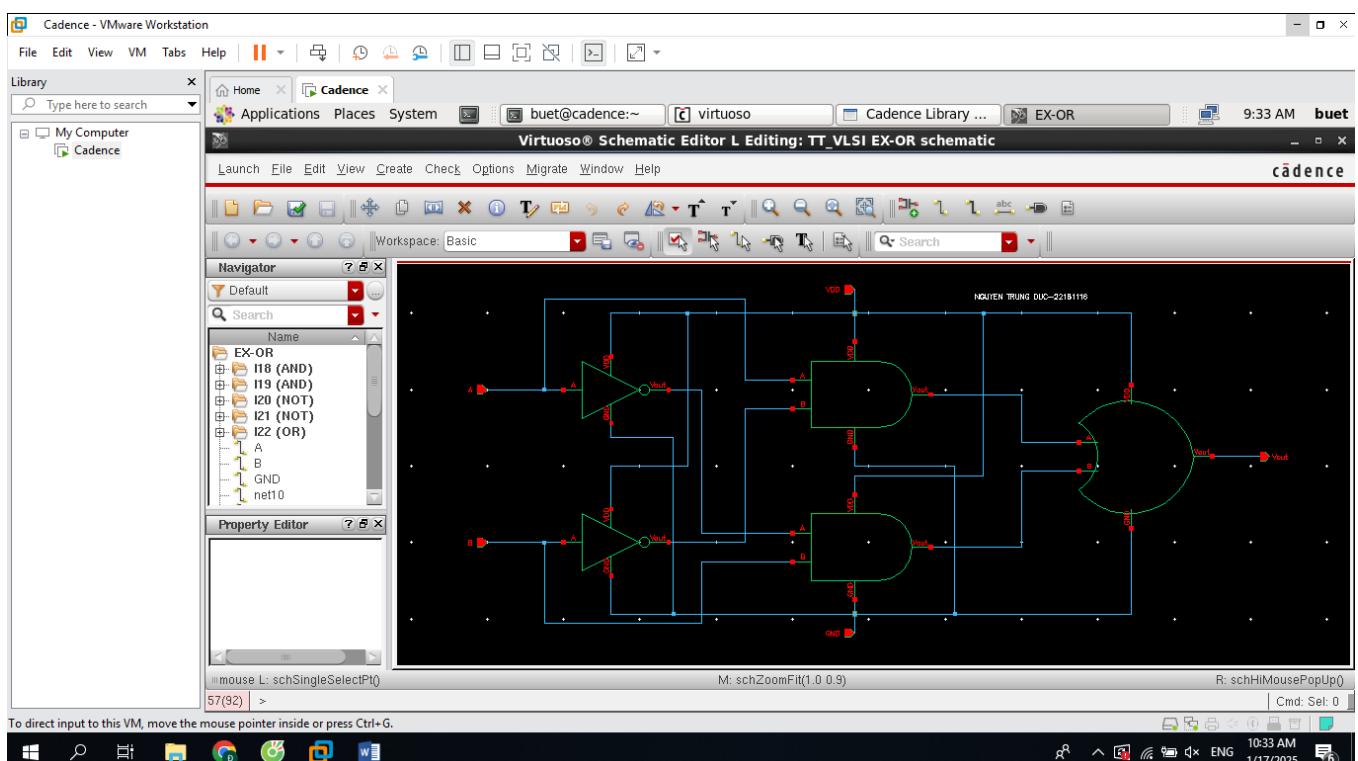
- Sau đó ta cho VA (màu đỏ) ở mức 1, VB (màu xanh) xuống mức 0 thì cho ra Vout (màu hồng) xuống mức 0.
- Tiếp tục ta lại cho VA (màu đỏ) xuống mức 0, VB (màu xanh) lên mức 1 thì cho ra Vout (màu hồng) xuống mức 0.
- Cuối cùng ta cho VA (màu đỏ) và VB (màu xanh) đều ở mức 0 thì ngõ ra Vout (màu hồng) lên mức 1.

Vậy: - Mạch EX-NOR hoạt động chính xác theo nguyên lý logic.

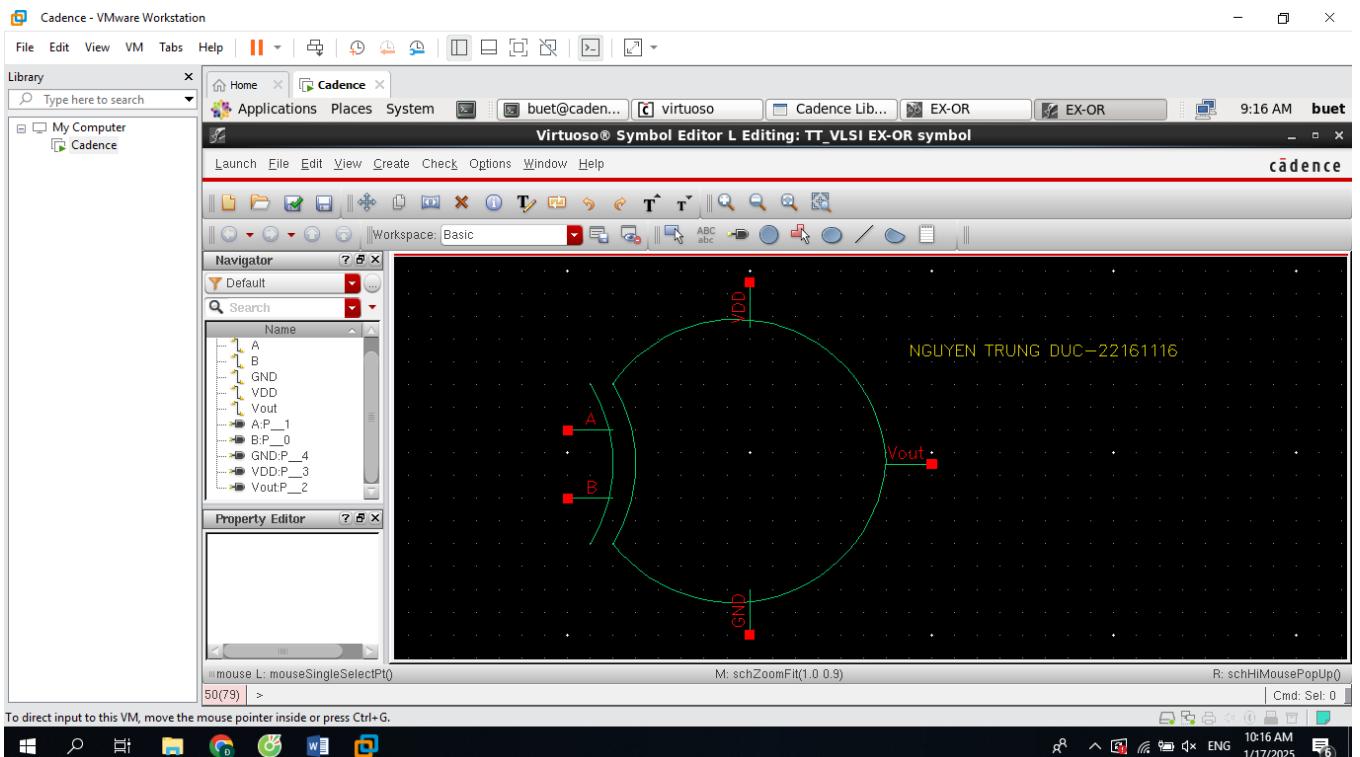
- Có độ trễ nhỏ, có thể cải thiện bằng cách tối ưu thiết kế mạch.
- Dữ liệu dạng sóng hợp lý, phản ánh đúng tính chất.

## EX-OR

+ Trình bày sơ đồ nguyên lý mạch

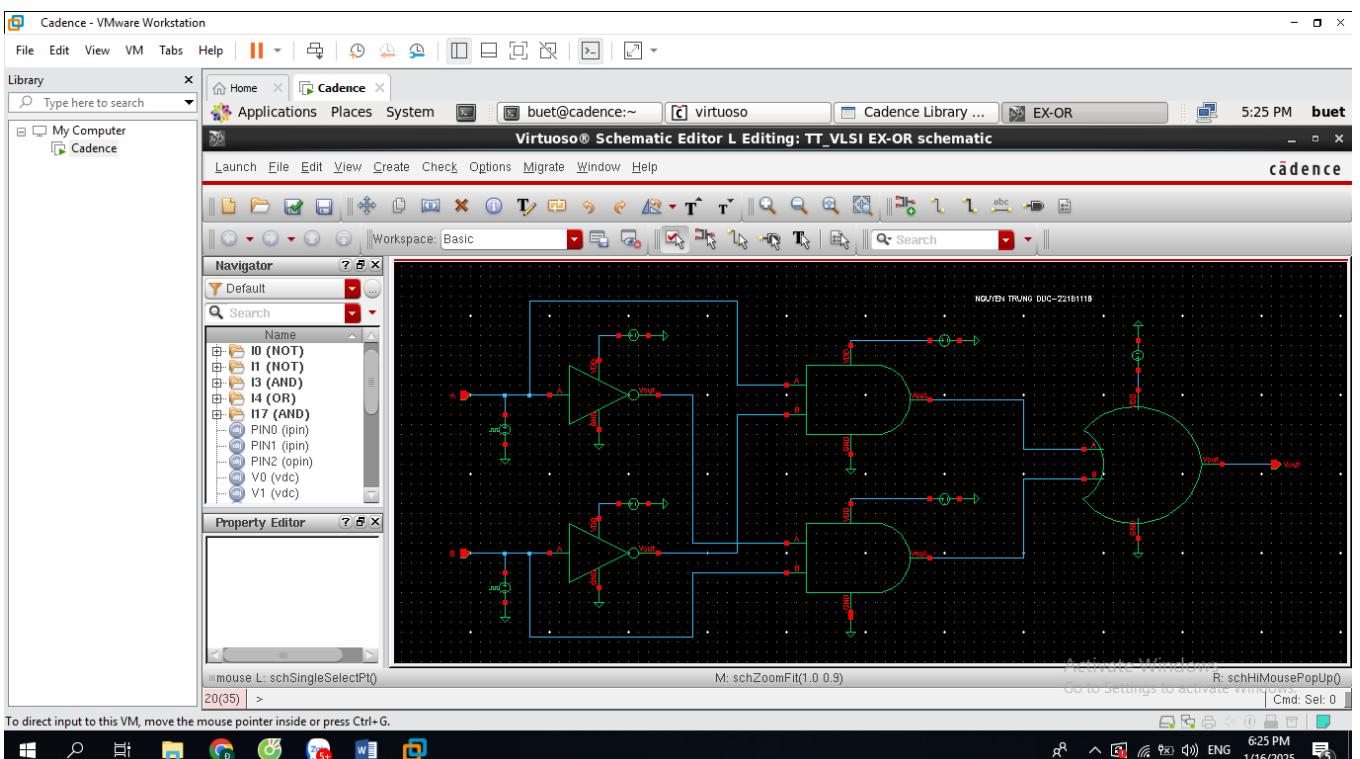


+ Đóng gói

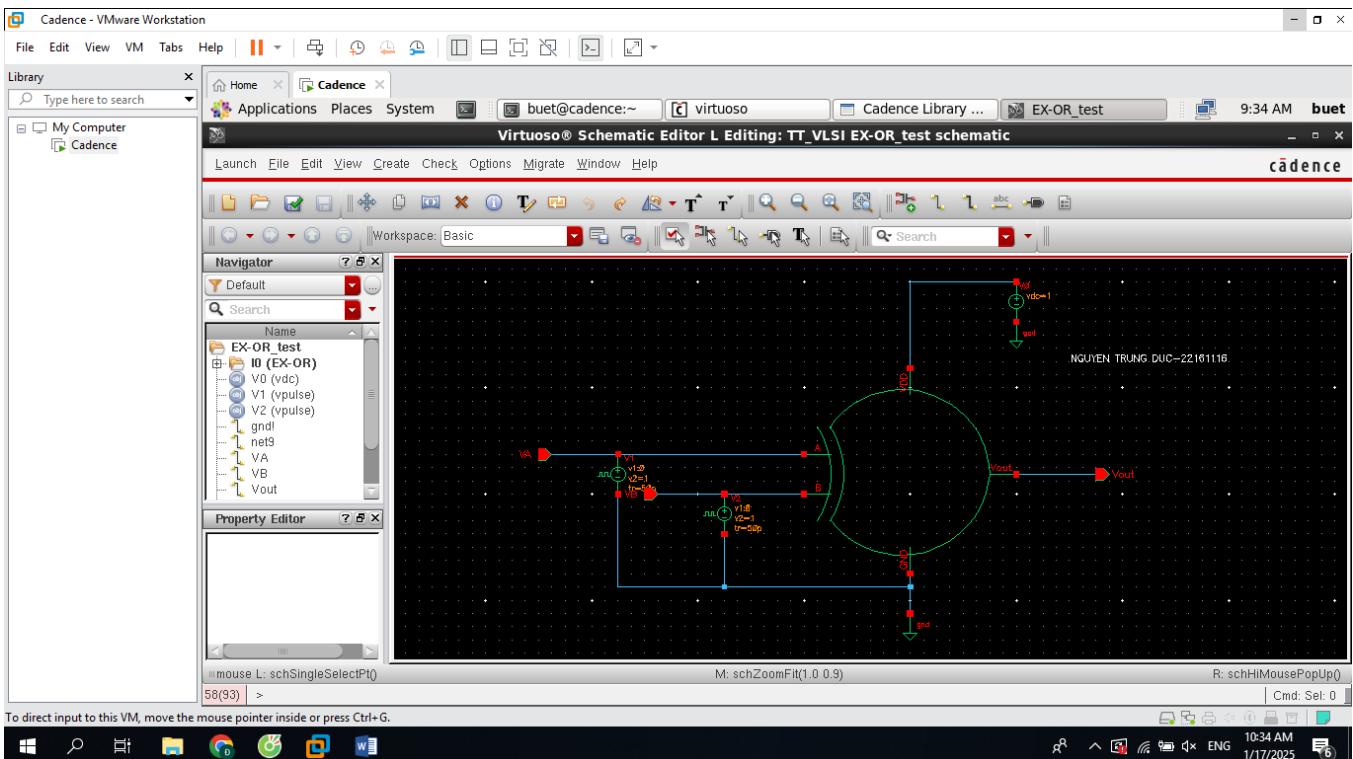


Kết quả mô phỏng có phân tích

- Vẽ trực tiếp



- Vẽ qua đóng gói



## Kết quả



Phân tích  
Công EX-OR thực hiện phép toán:

$$V_{\text{out}} = A \oplus B$$

## Cỗng XOR



Bảng chân lý - Cỗng XOR		
Đầu vào A	Đầu vào B	Đầu ra Q
0	0	0
0	1	1
1	0	1
1	1	0

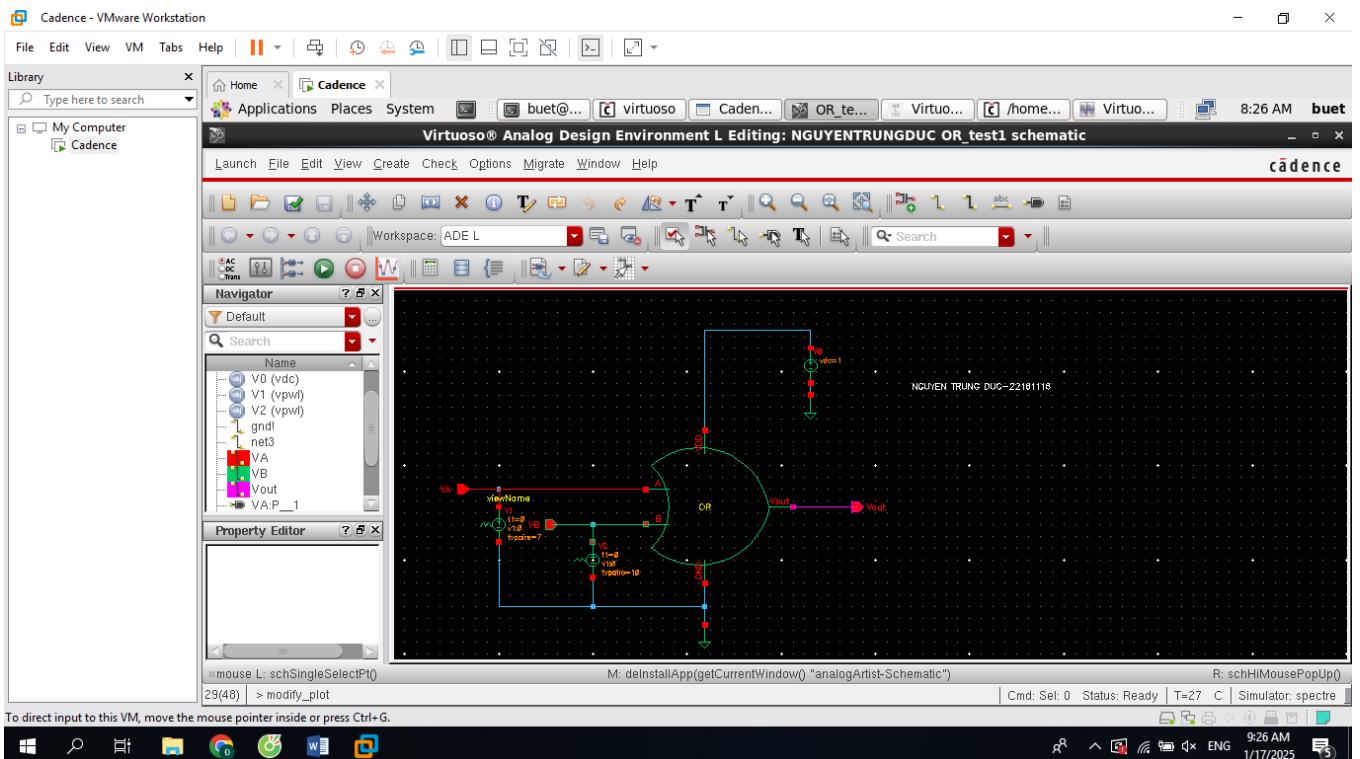
Ta có thể thấy kết quả :

- Ban đầu ta cho VA (màu đỏ) ở mức 1, VB (màu xanh) ở mức 1 thì cho ra Vout (màu hồng) ở mức 0.
- Sau đó ta cho VA (màu đỏ) ở mức 1, VB (màu xanh) xuống mức 0 thì cho ra Vout (màu hồng) lên mức 1.
- Tiếp tục ta lại cho VA (màu đỏ) xuống mức 0, VB (màu xanh) lên mức 1 thì cho ra Vout (màu hồng) lên mức 1.
- Cuối cùng ta cho VA (màu đỏ) và VB (màu xanh) đều ở mức 0 thì ngõ ra Vout (màu hồng) xuống mức 0.

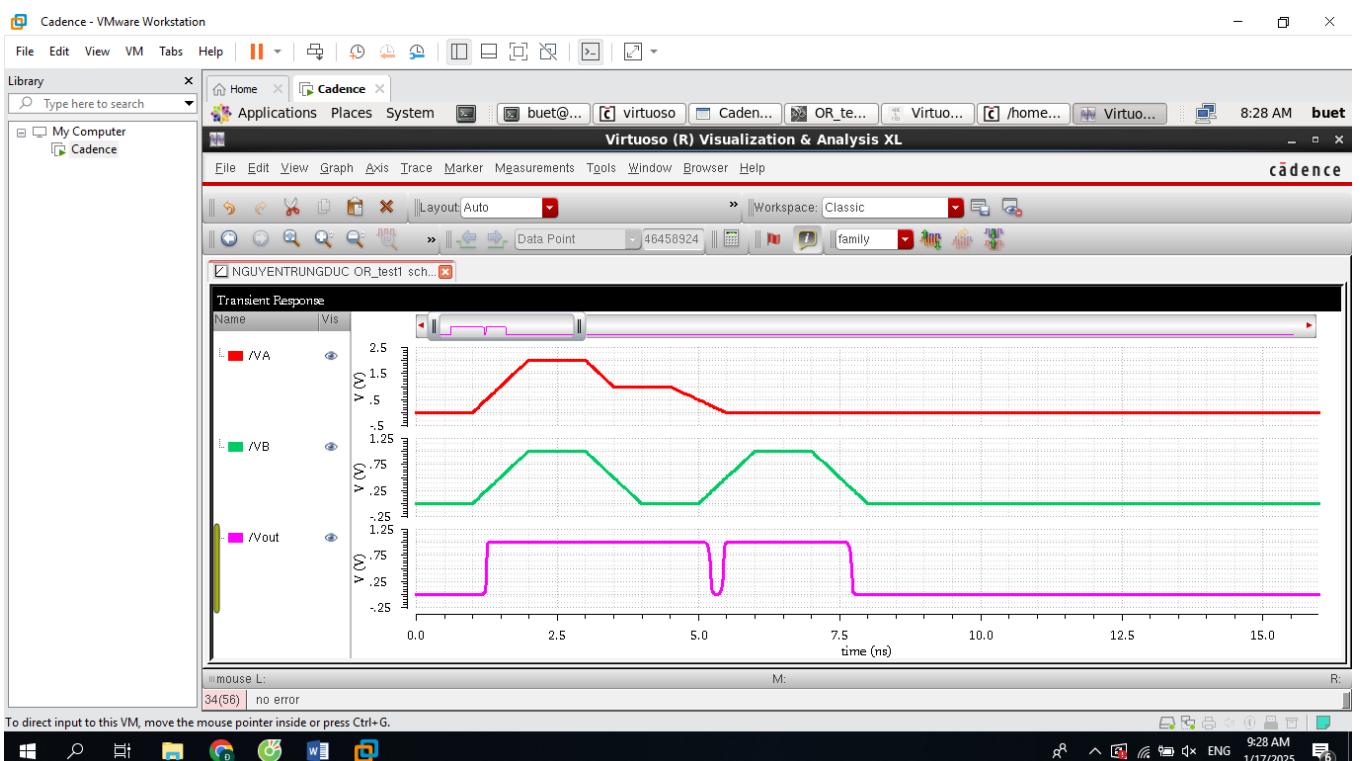
Vậy: - Mạch EX-OR hoạt động chính xác theo nguyên lý logic.

- Có thể cải thiện độ trễ, nếu cần tối ưu hiệu suất.
- Dữ liệu dạng sóng hợp lý, phản ánh đúng tính chất EX-OR.

5. Thay thế vpulse thành vpwl để chỉnh dạng sóng tùy ý
- Trình bày sơ đồ nguyên lý mạch



## Kết quả mô phỏng



Nhận xét: Khi ta sử dụng vpwl thay vì sử dụng vpulse thì sẽ tạo được độ linh hoạt cho dạng sóng ngõ vào từ đó giúp quan sát một cách chi tiết hơn dạng sóng ngõ ra.