# PHẦN 2: LTSPICE

## BUỔI 1

1. **Vẽ mạch nguyên lý sử dụng cổng logic và sử dụng công nghệ CMOS cho các biểu thức sau:**

Text, whiteboard

Description automatically generated

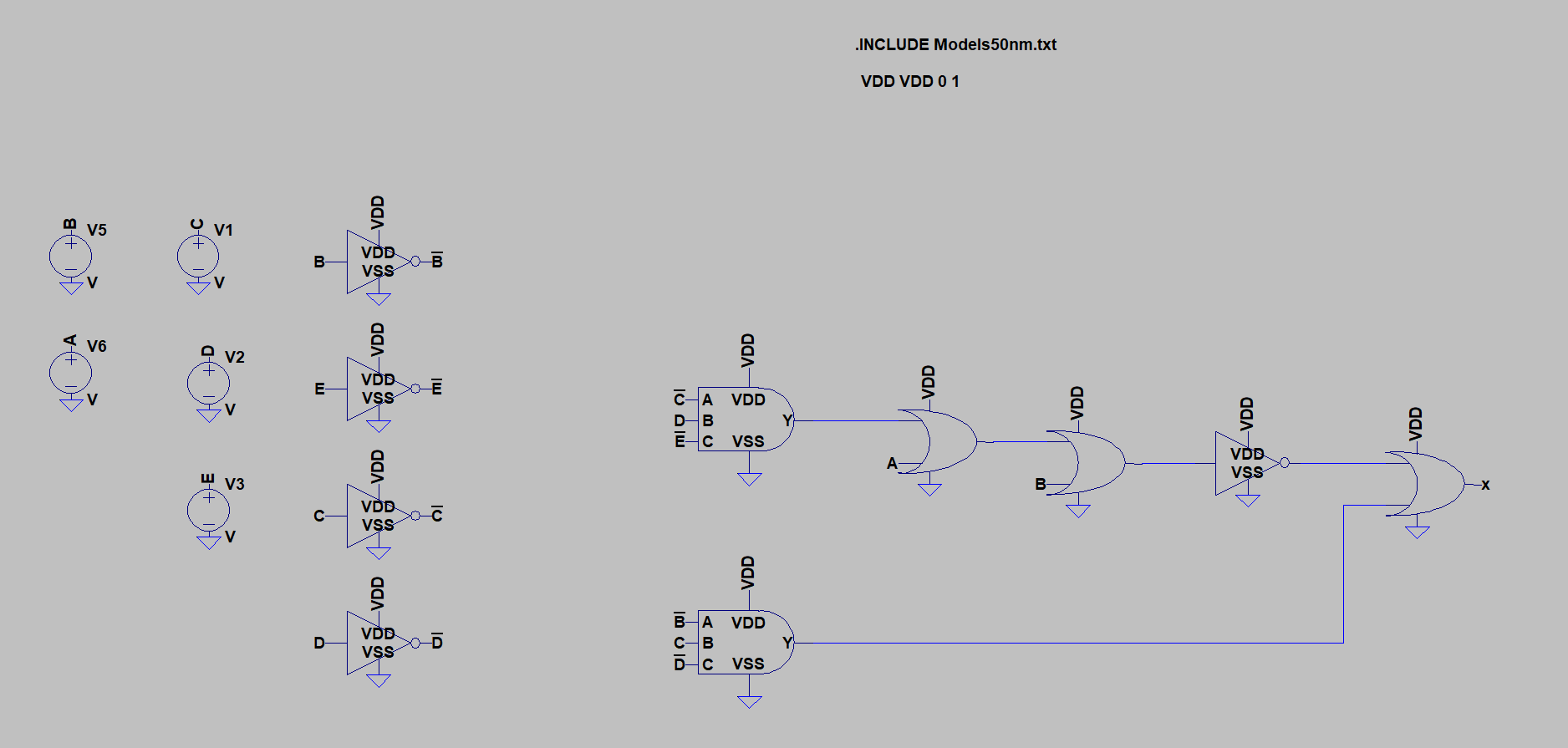




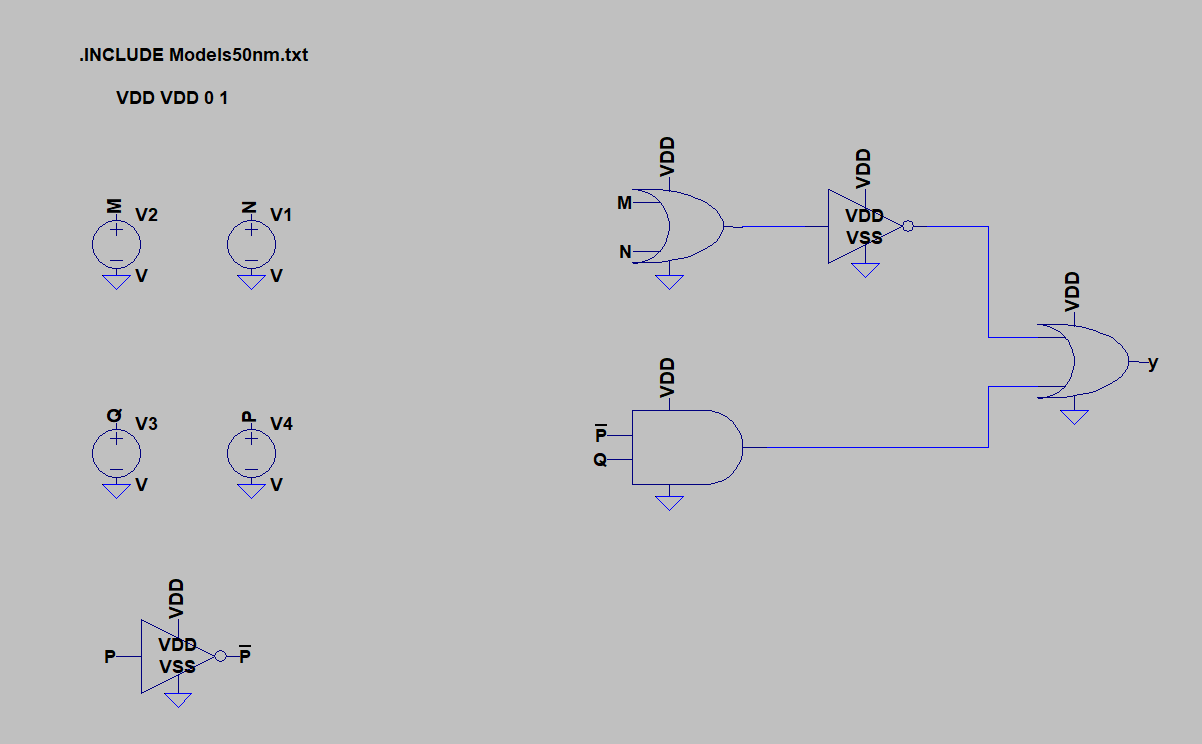




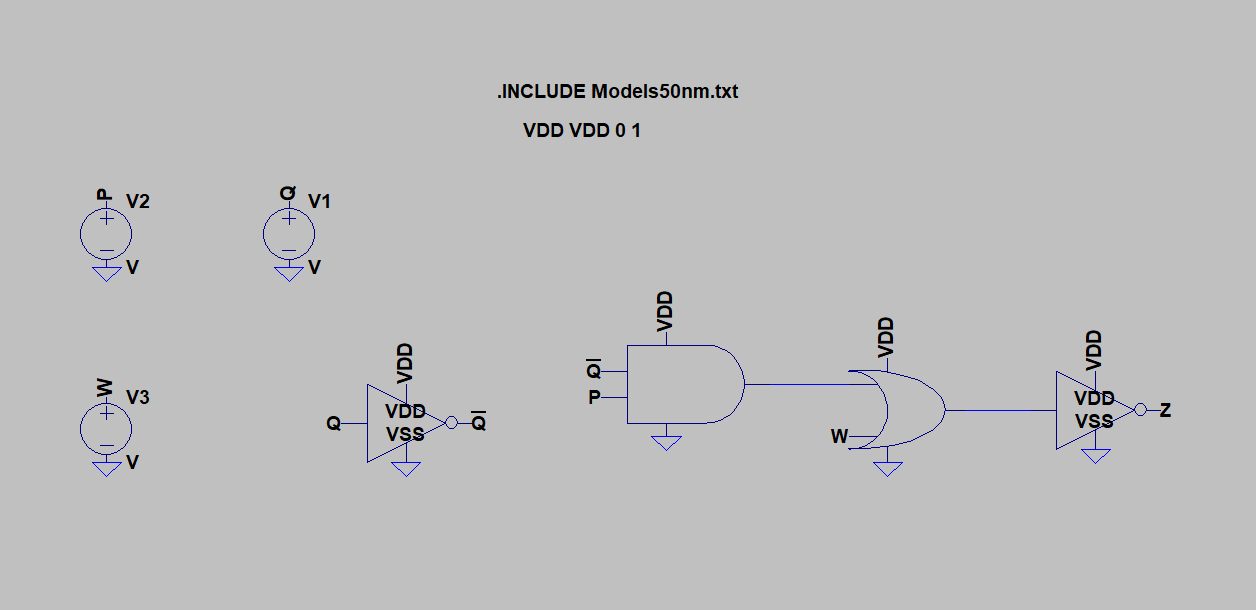
a)



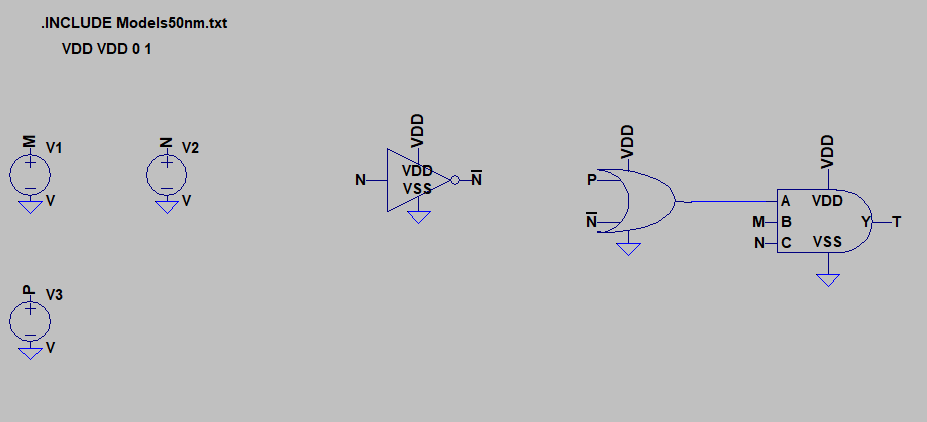
b)



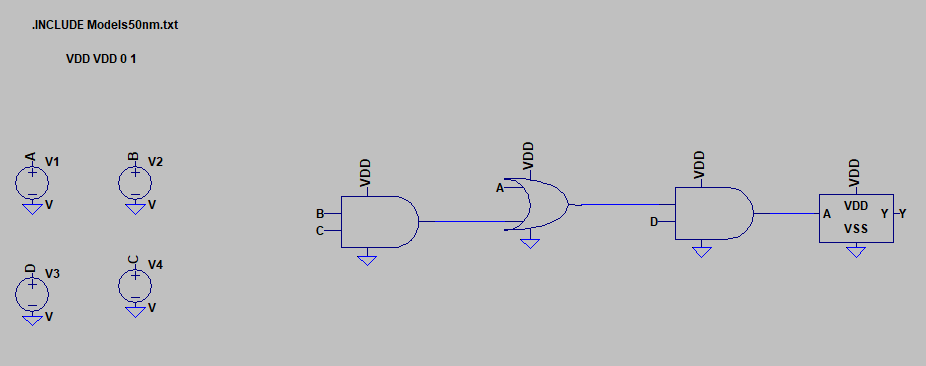
c)



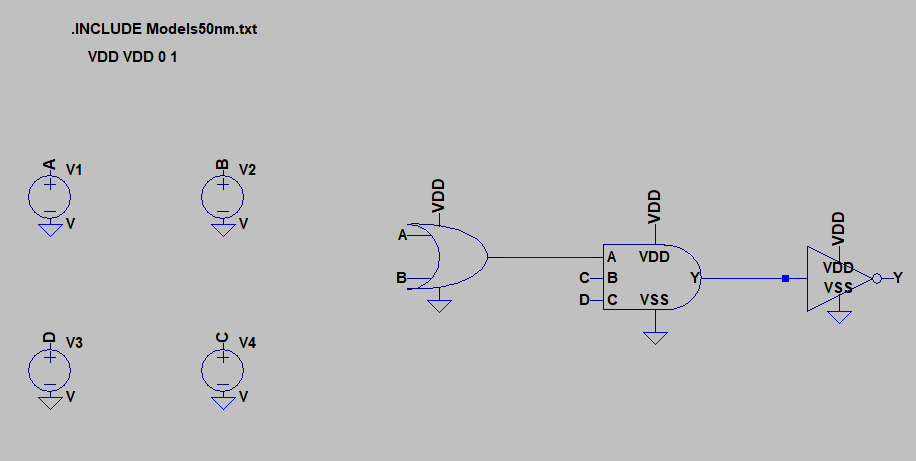
d)



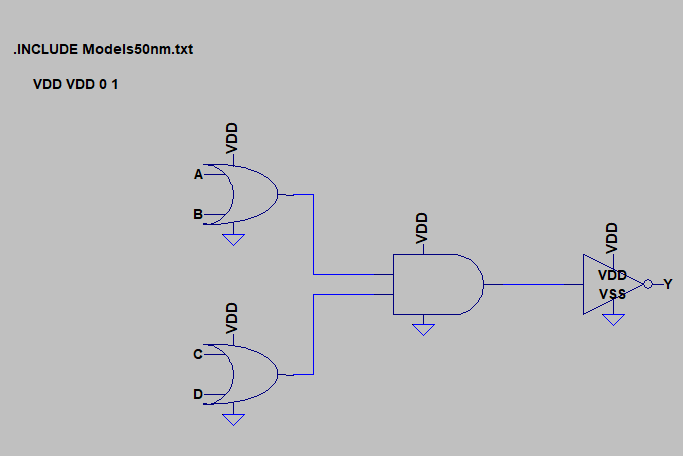
e)



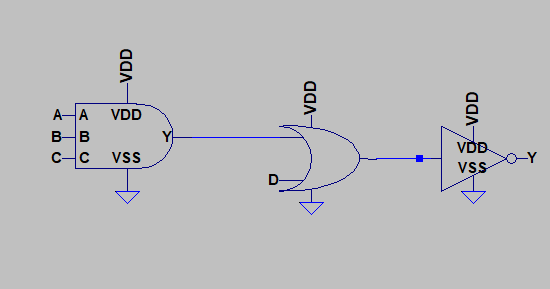
f)



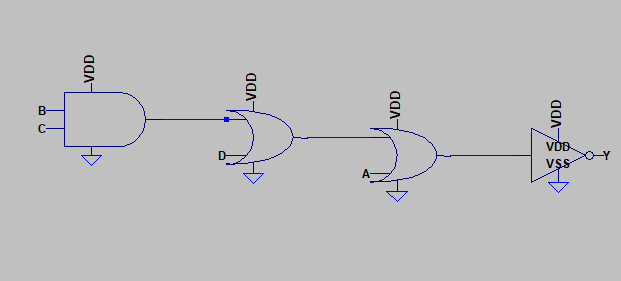
g)



h)



i)



Bài 2

Xác định biểu thức ngõ ra và vẽ mạch logic CMOS cho các mạch sau:

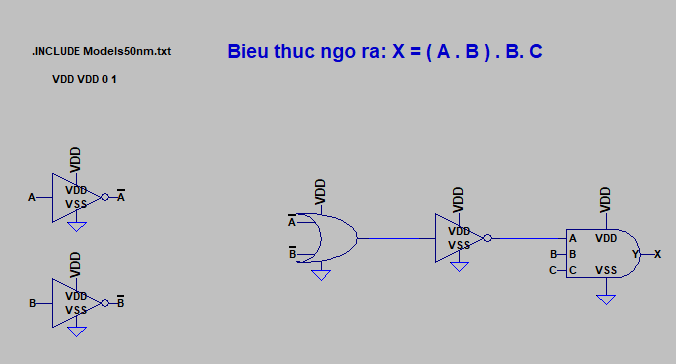
Diagram, schematic

Description automatically generated

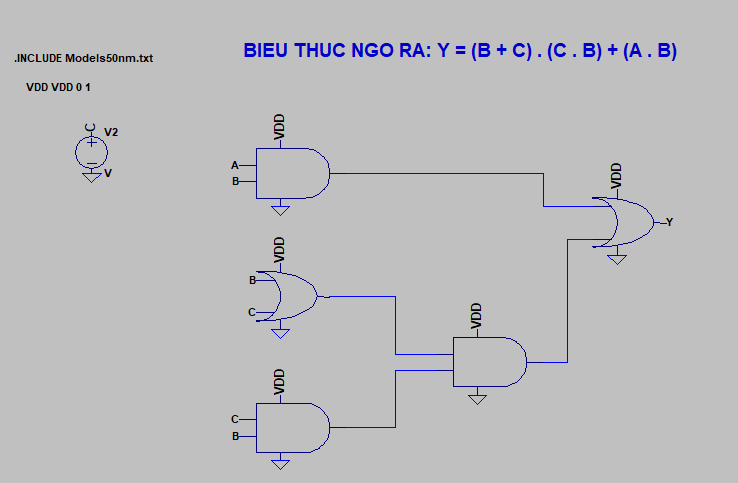
Diagram

Description automatically generated

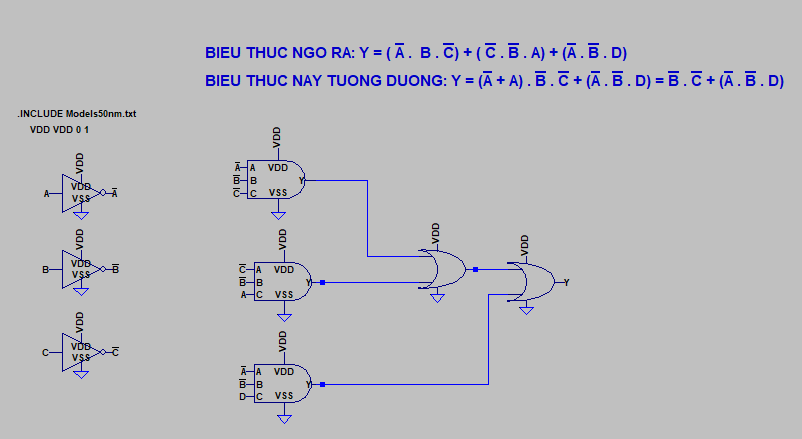
a)



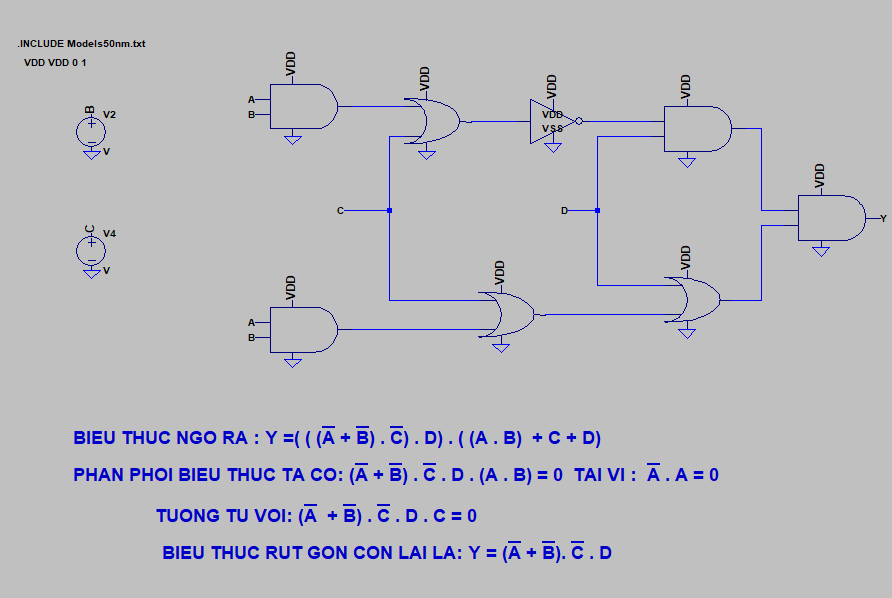
b)



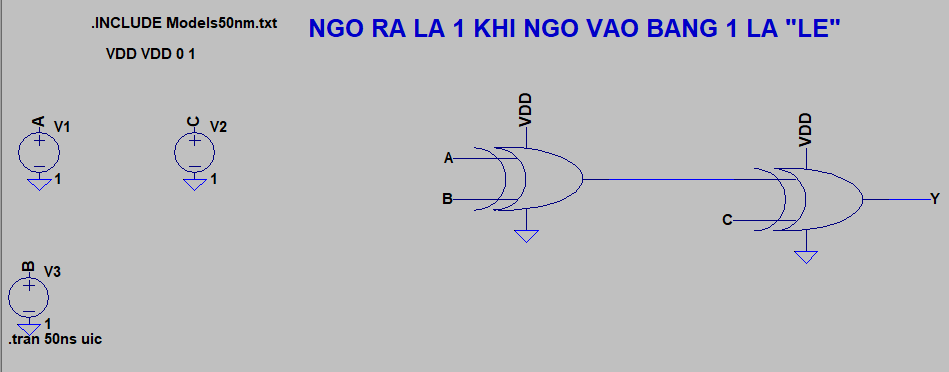
c)



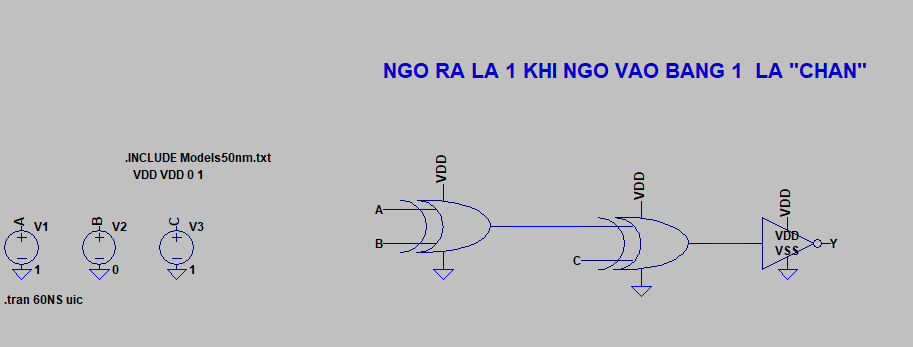
d)



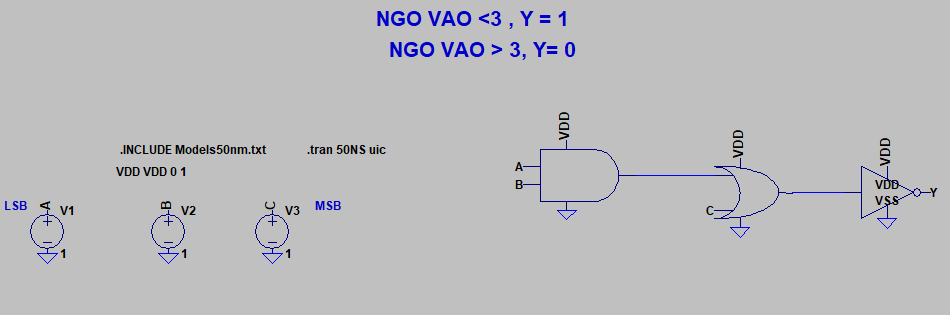
Bài 3 Hãy thiết kế một mạch có 3 ngõ vào, 1 ngõ ra. Ngõ ra lên 1 khi có số lẻ ngõ vào ở mức 1.



Bài 4 Hãy thiết kế một mạch có 3 ngõ vào, 1 ngõ ra. Ngõ ra lên 1 khi có số chẵn ngõ vào ở mức 1.

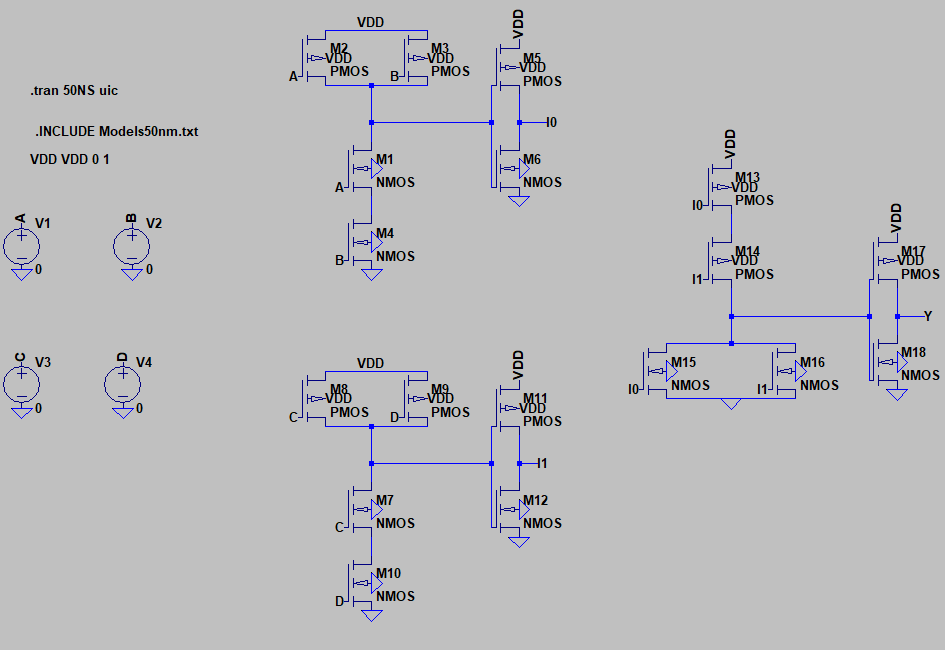


Bài 5 Hãy thiết kế một mạch có 3 ngõ vào, 1 ngõ ra. Ngõ ra lên 1 khi giá trị thập phân của ngõ vào nhỏ hơn 3, các trường hợp còn lại ngõ ra mức 0.

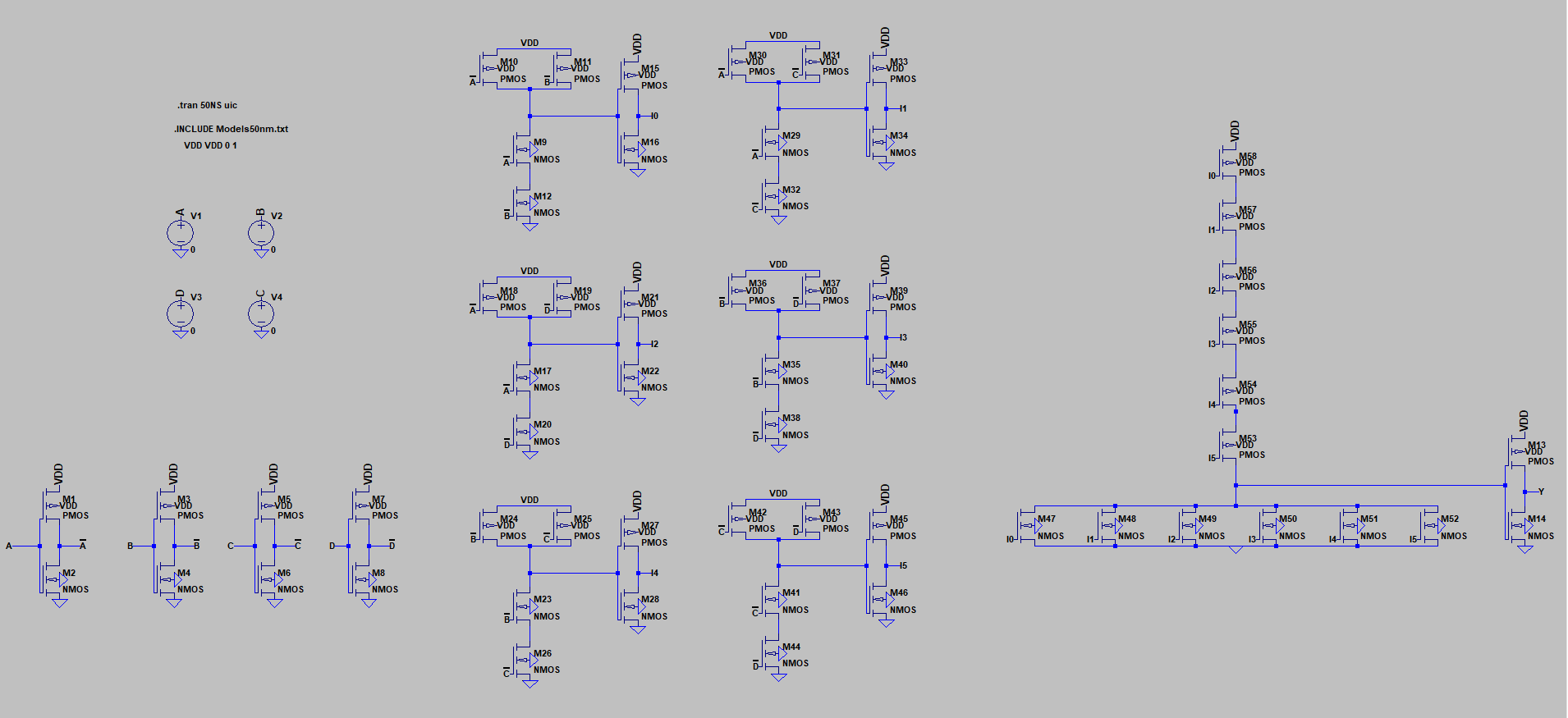


## BUỔI 2

1. **Hãy thiết kế mạch có 4 ngõ vào, 1 ngõ ra. Ngõ ra mức 1 khi A = B = 1 hoặc C = D = 1.**



1. **Hãy thiết kế mạch có 4 ngõ vào, 1 ngõ ra. Ngõ ra mức 1 khi ít nhất hai ngõ vào ở mức 0.**

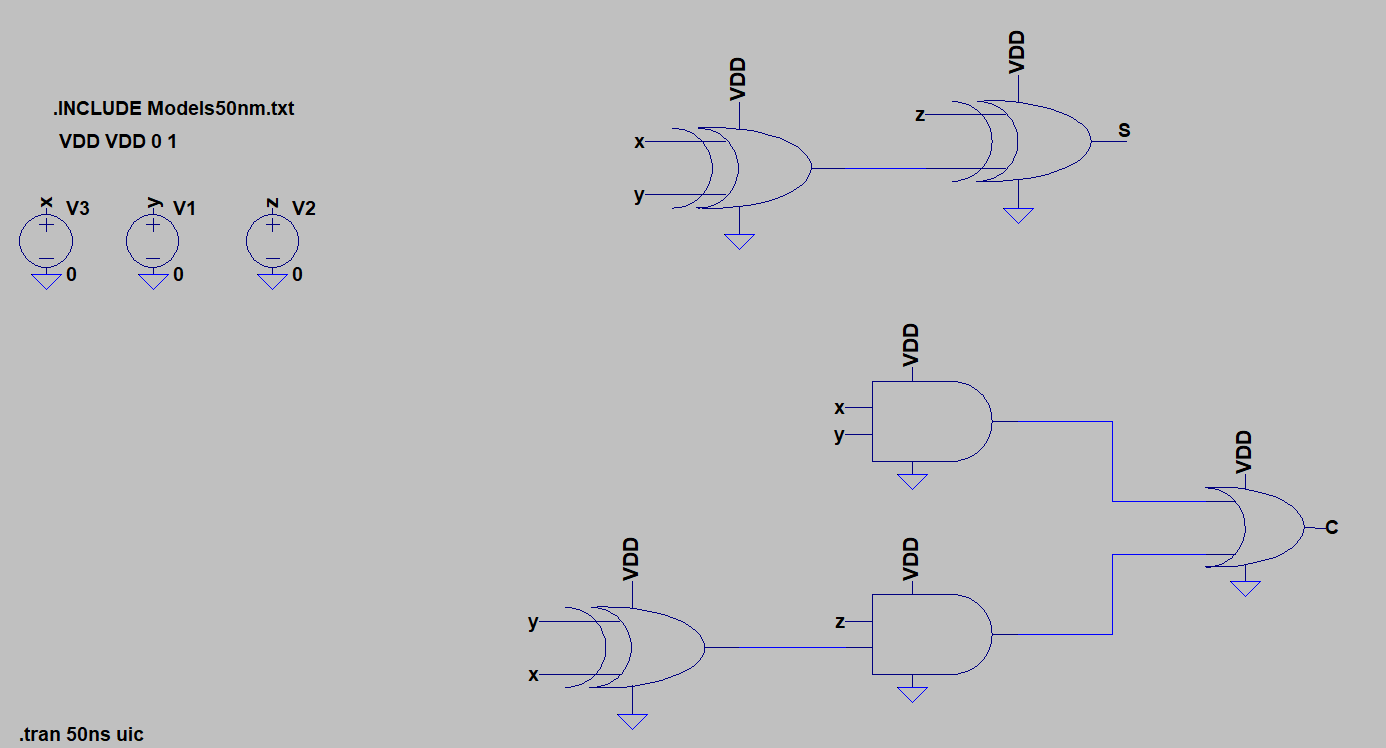


1. **Thiết kế mạch tổ hợp có 3 ngõ vào x, y, z và 2 ngõ ra C và S với quan hệ như sau:**

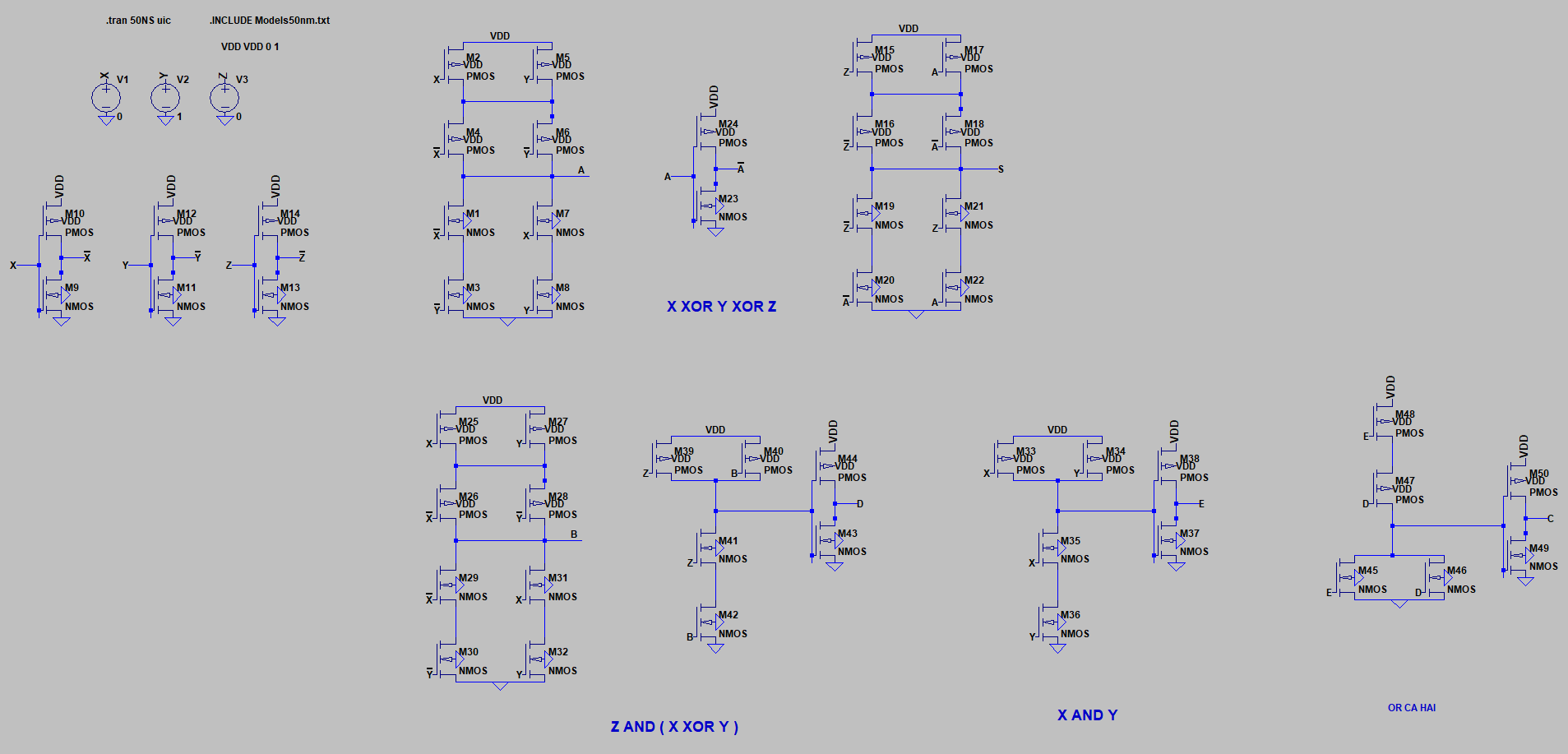
**A group of symbols on a white background

Description automatically generated**

1. **Dùng cổng logic**
2. **Dùng mạch logic CMOS**



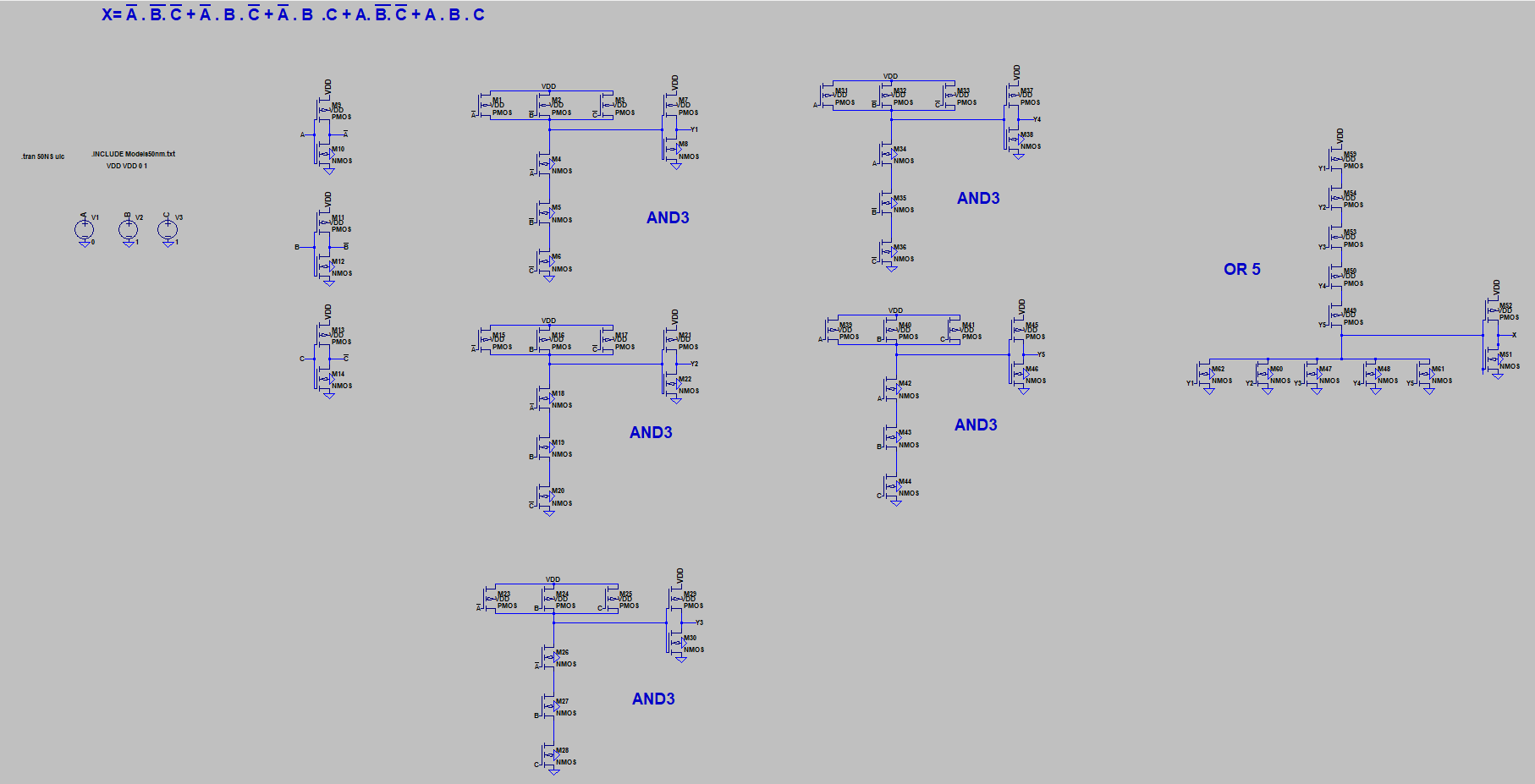
b)



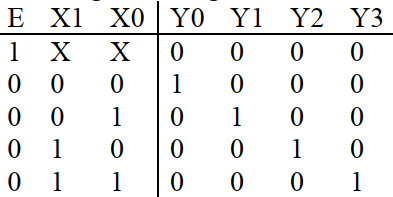
1. Thiết kế mạch có bảng trạng thái như sau:

A grid of numbers and letters

Description automatically generated

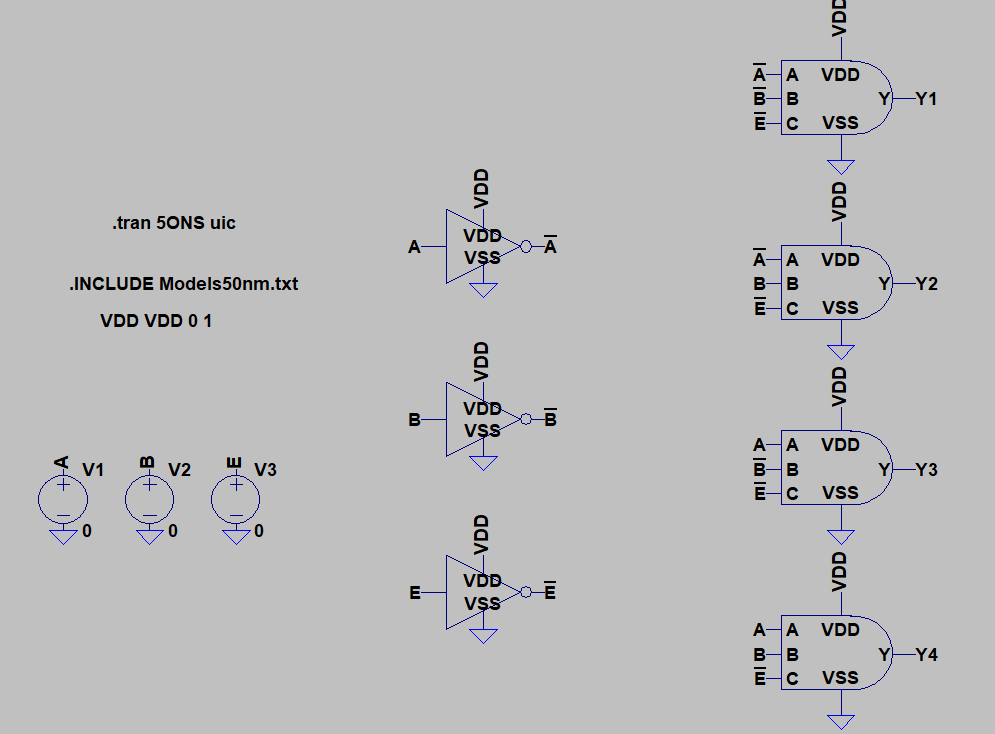


1. **Thiết kế mạch tổ hợp hoạt động theo bảng sau:**

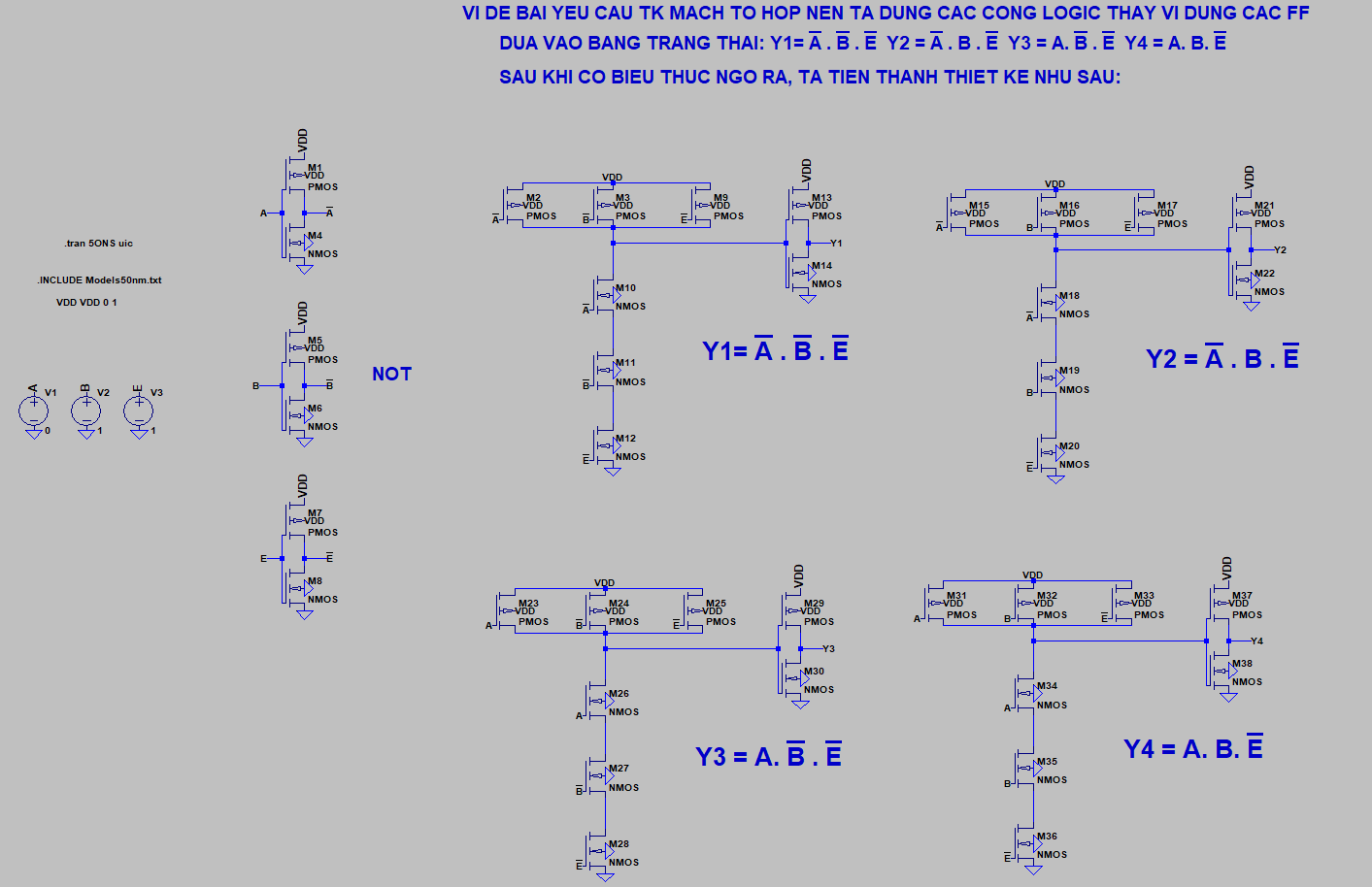
****

1. **Dùng cổng logic**
2. **Dùng mạch logic CMOS**

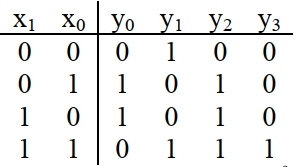
a)

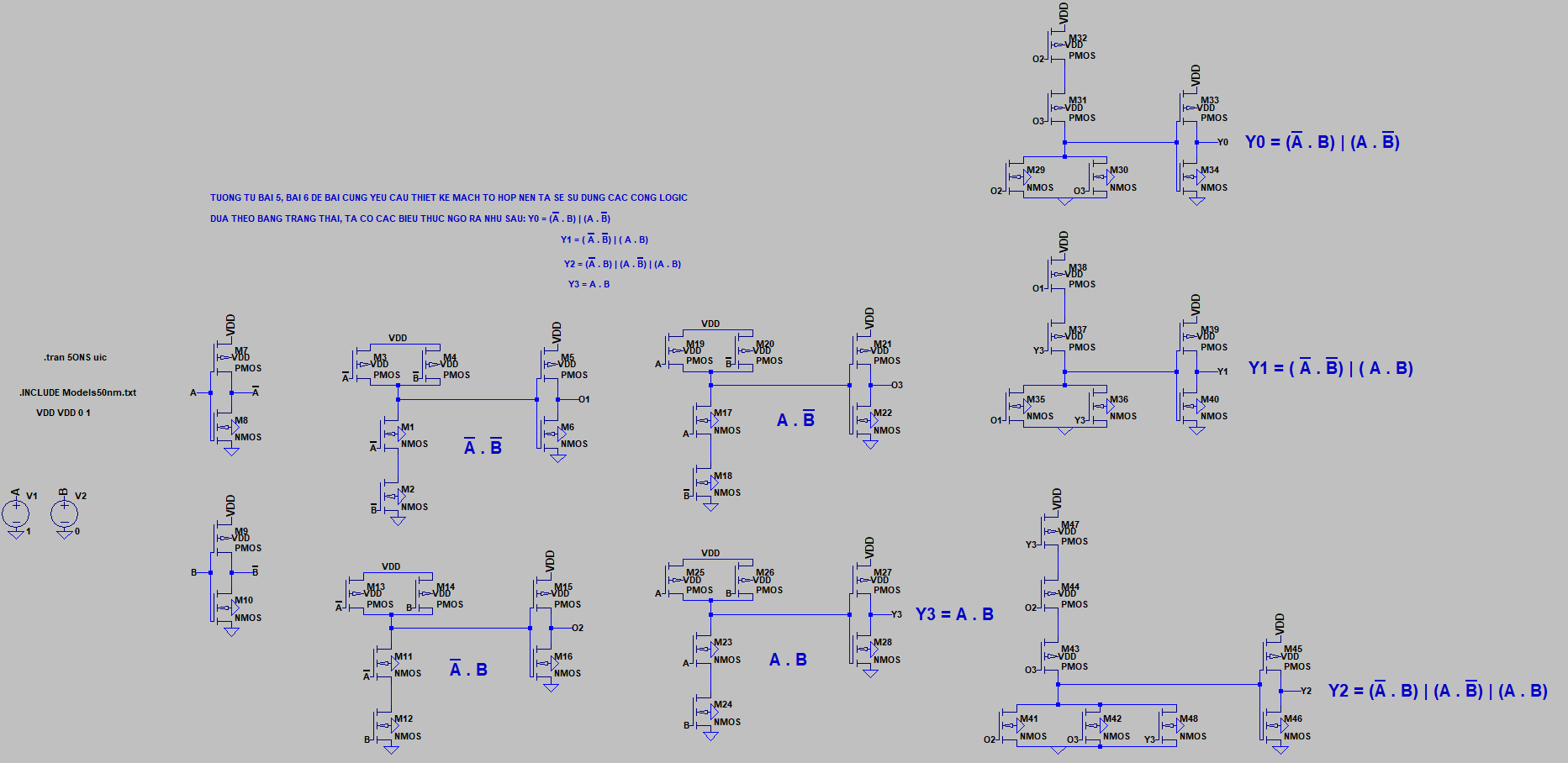


b)

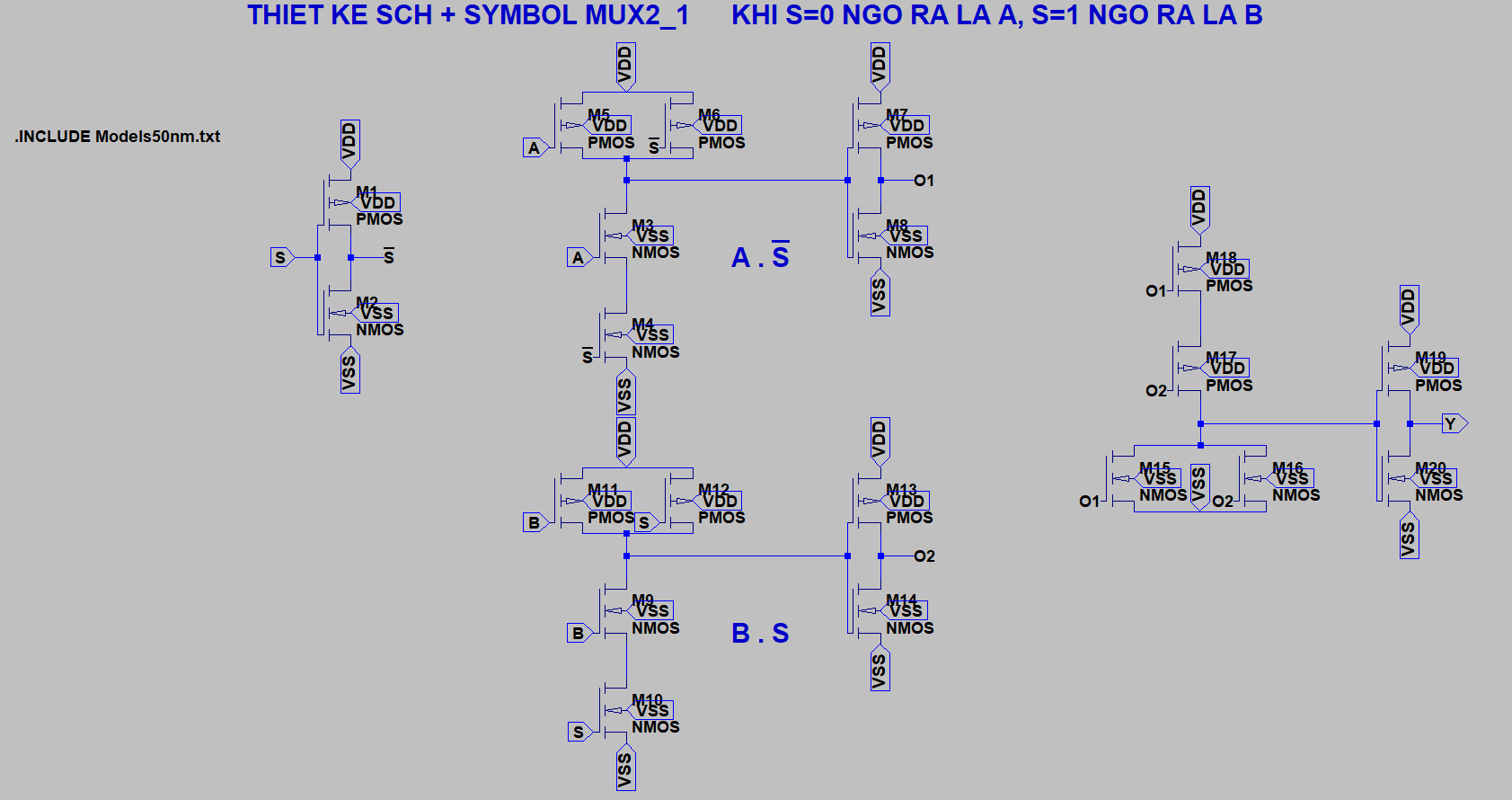


1. **Thiết kế mạch tổ hợp có bảng chân trị sau:**

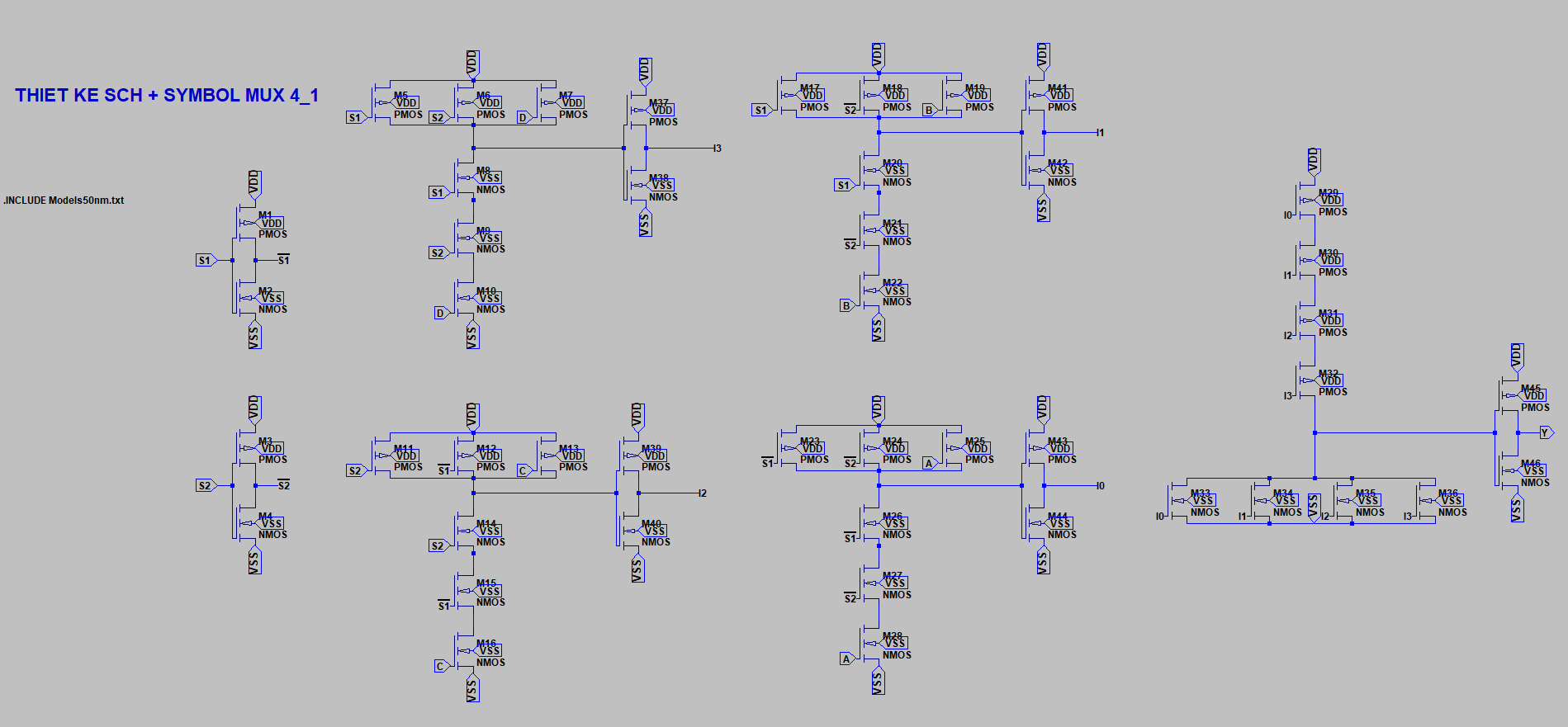




1. **Thiết kế mạch MUX2:1**

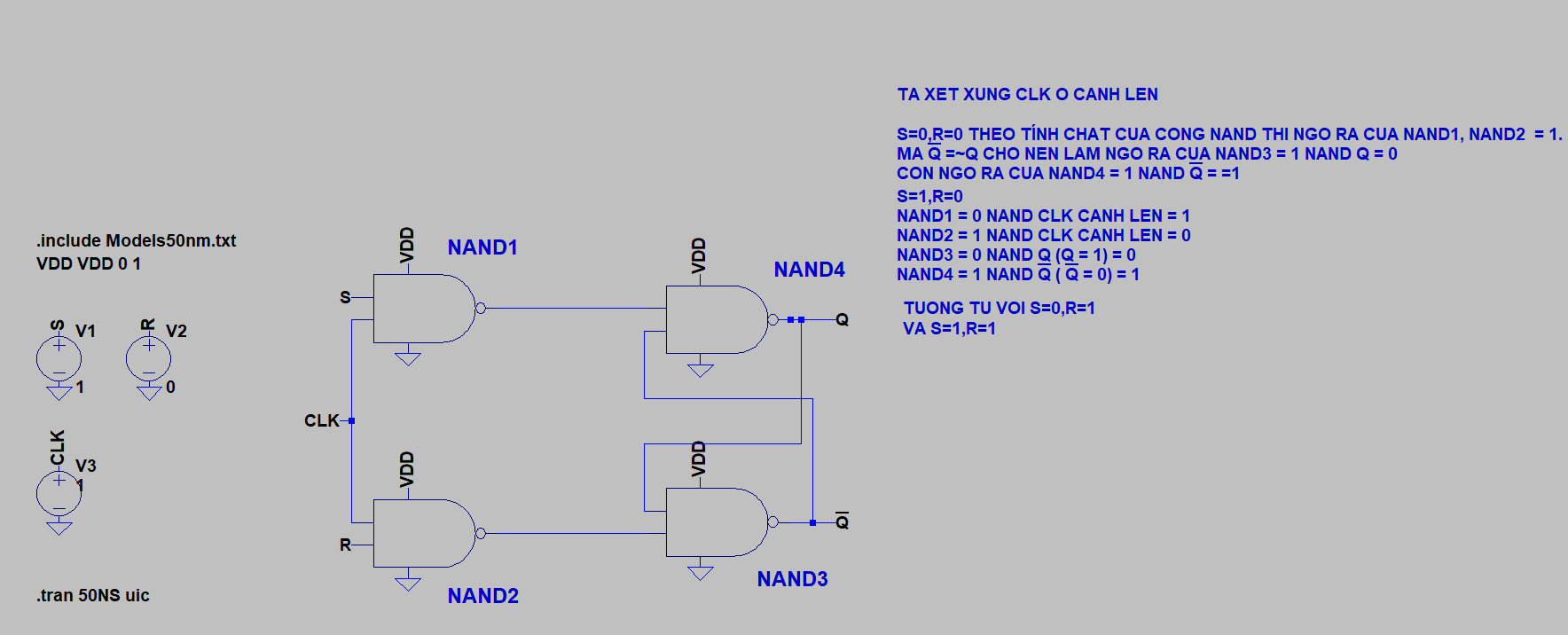


1. **Thiết kế mạch MUX4:1**

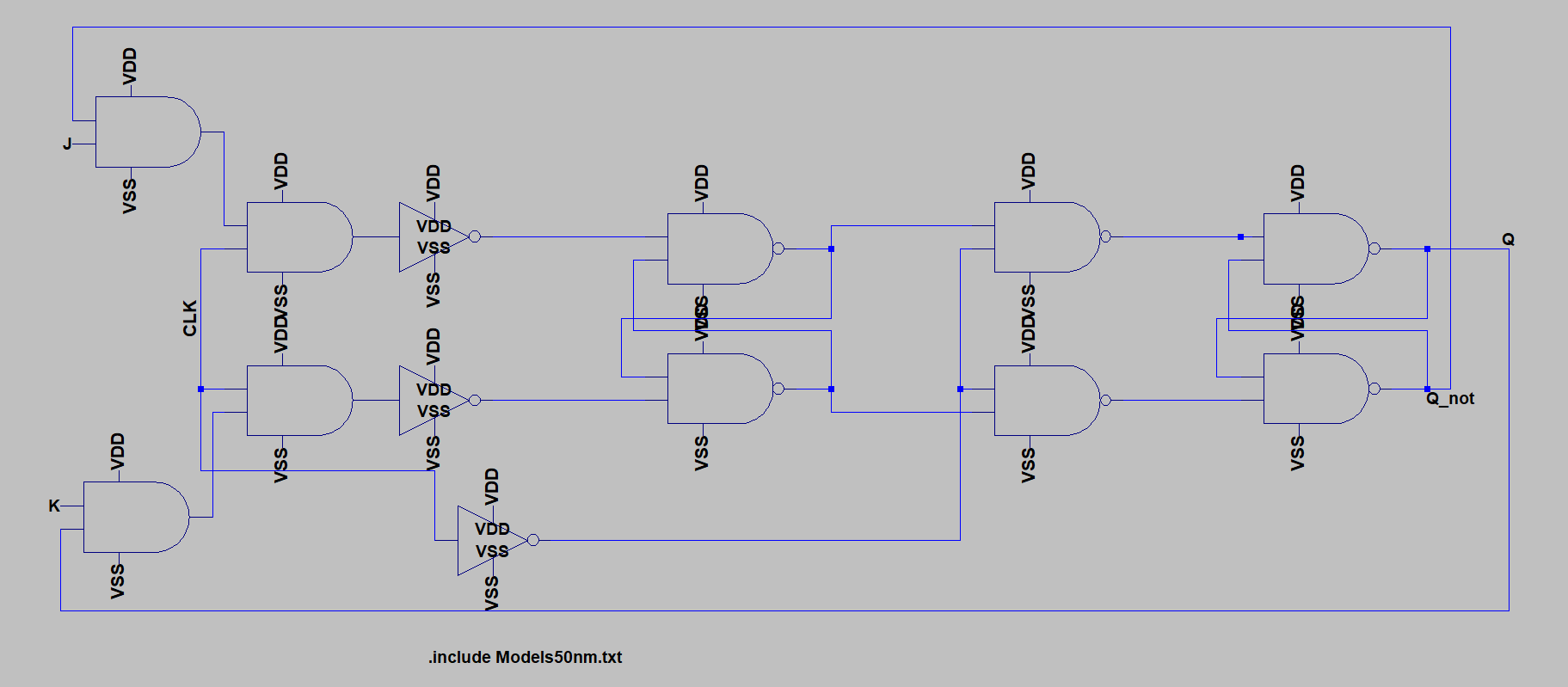


## BUỔI 3

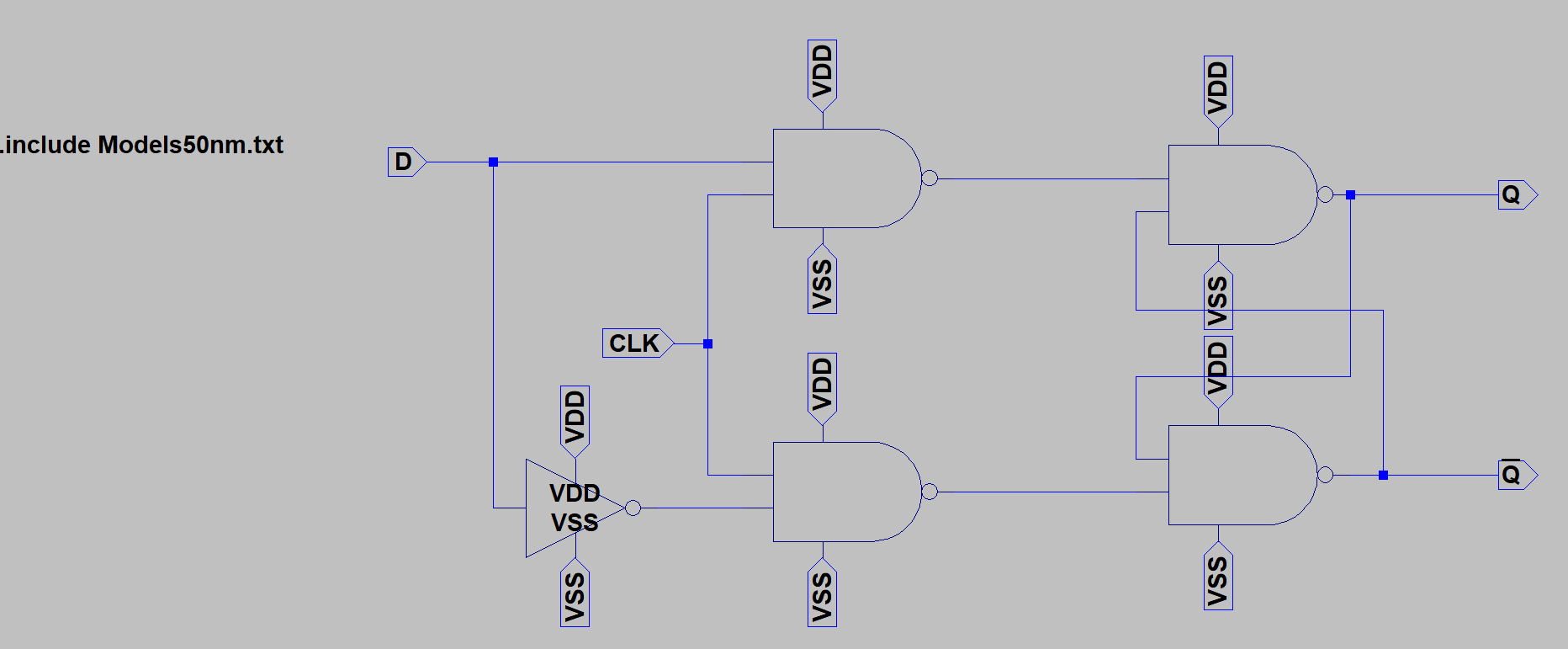
1. **Thiết kế và mô phỏng SR latch và SR flip flop (kèm giải thích nguyên lý)**



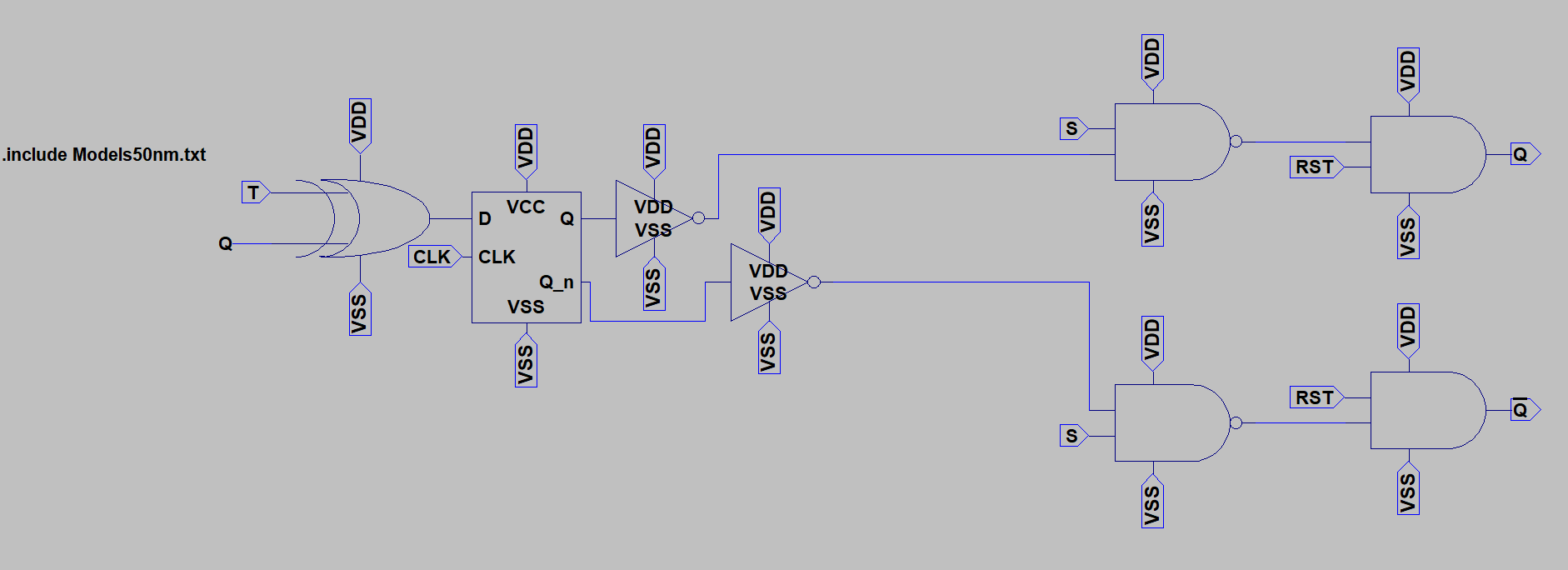
1. **Thiết kế và mô phỏng JK latch và JK flip flop**



1. **Thiết kế và mô phỏng D latch và D flip flop**



1. **Thiết kế và mô phỏng T latch và T flip flop**



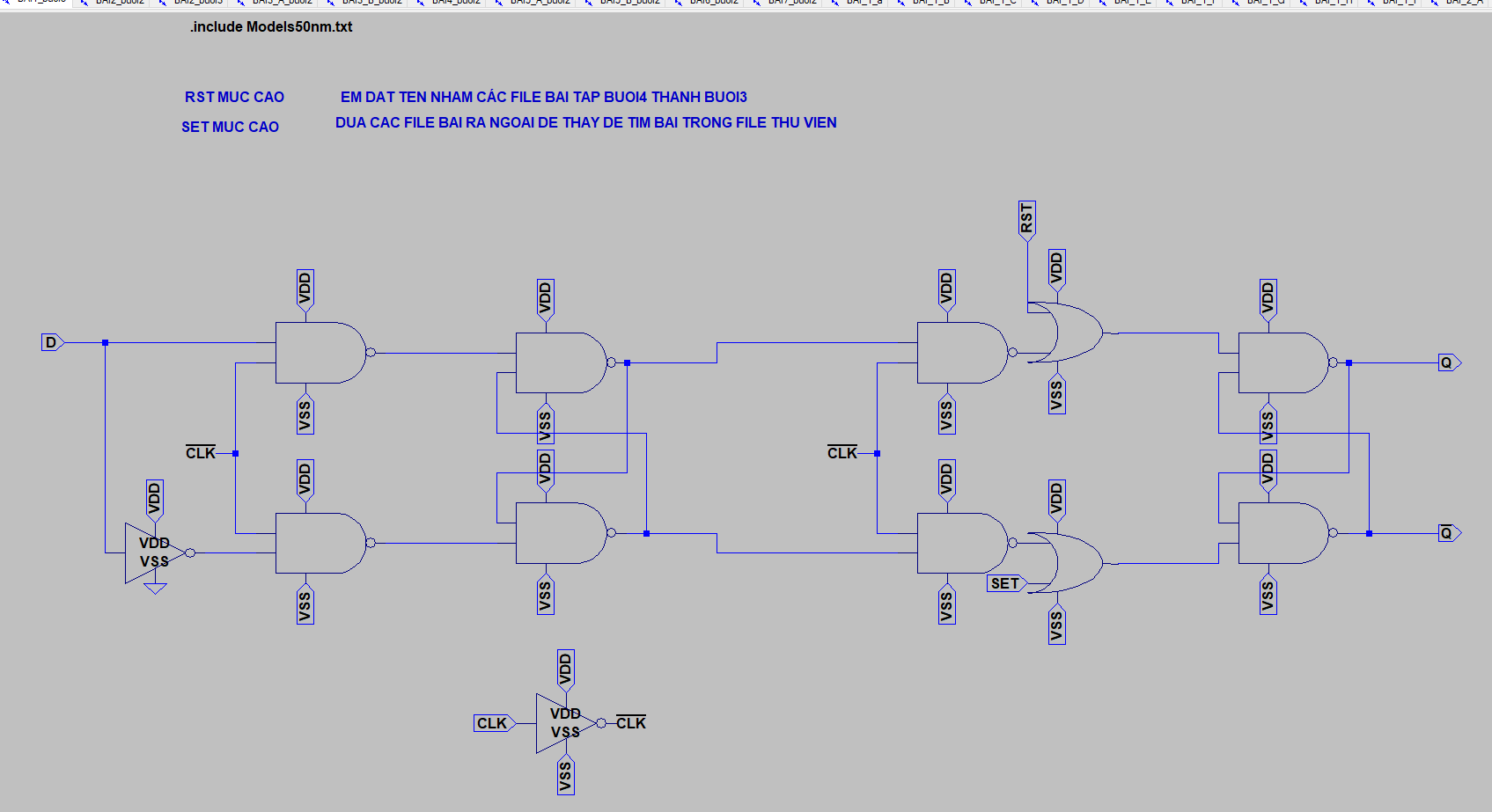
1. **Thiết kế Dlatch/DFlipFlop có thêm chân điều khiển Enable tác động mức thấp.**

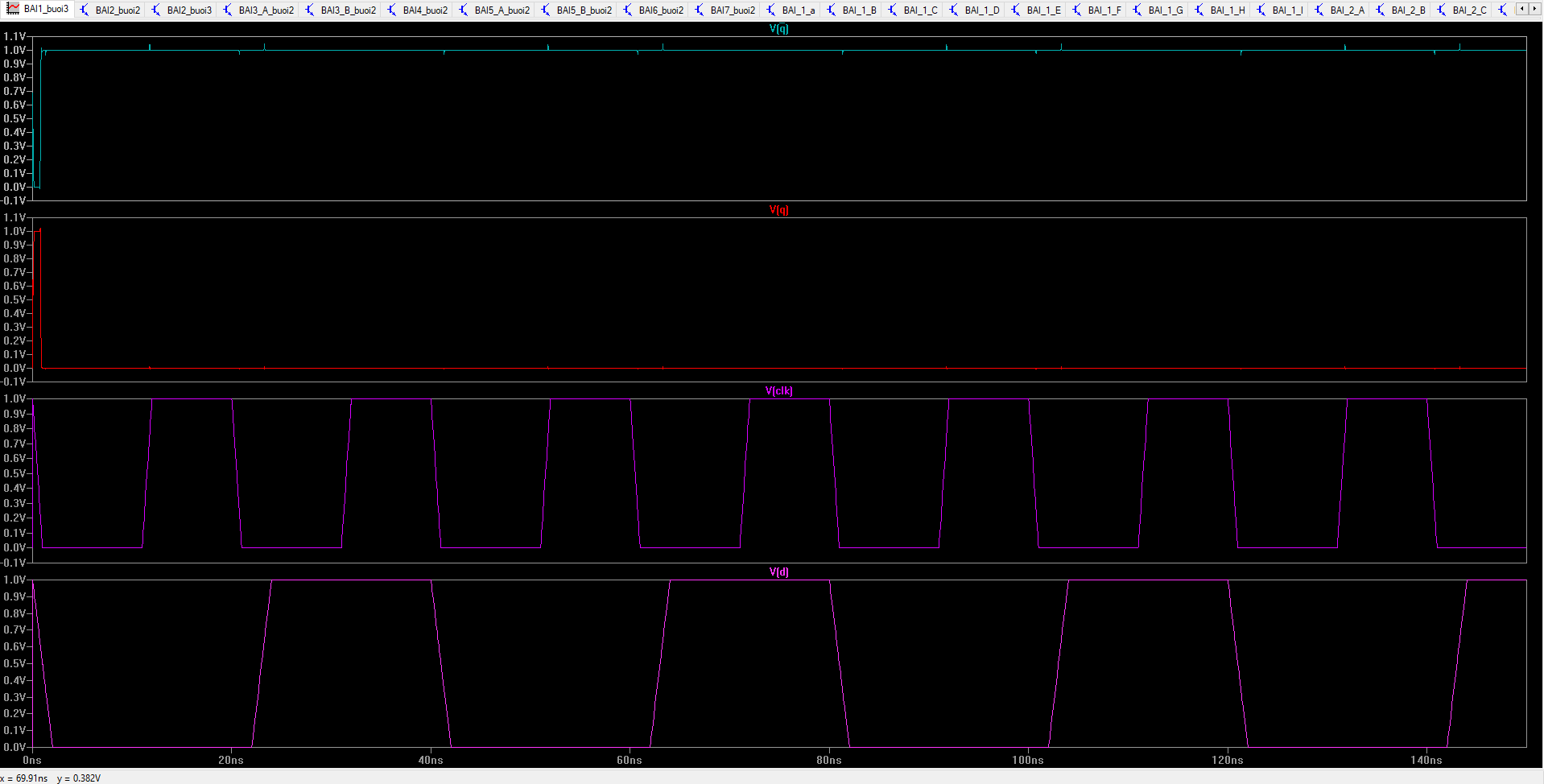
## BUỔI 4

**1. Thiết kế Dlatch HOẶC DFlipFlop có:**

**a. Xung clock cạnh xuống.**

**b. Reset mức cao/thấp đồng bộ/bất đồng bộ với xung clock.**

**c. Set mức cao/thấp đồng bộ/bất đồng bộ với xung clock.**

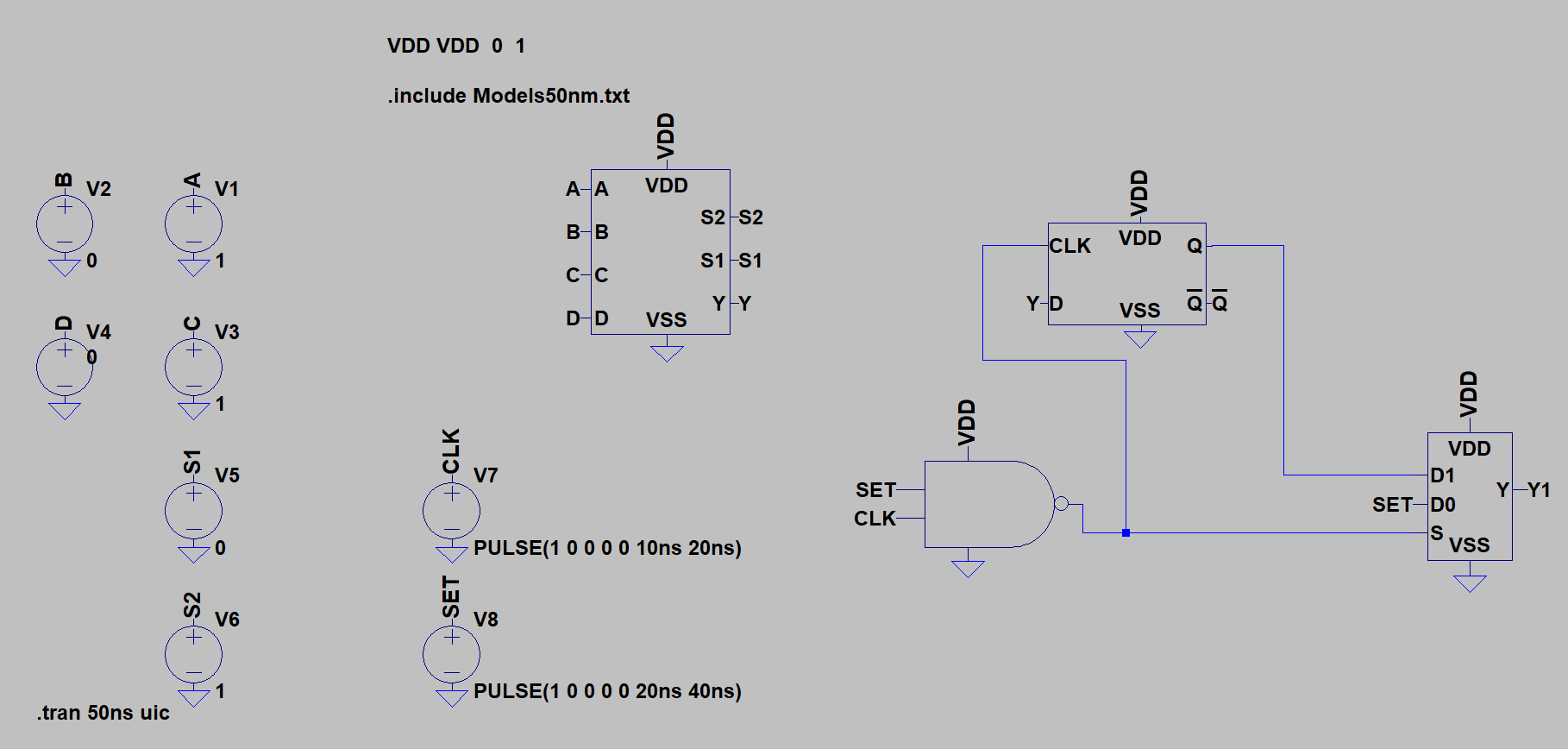


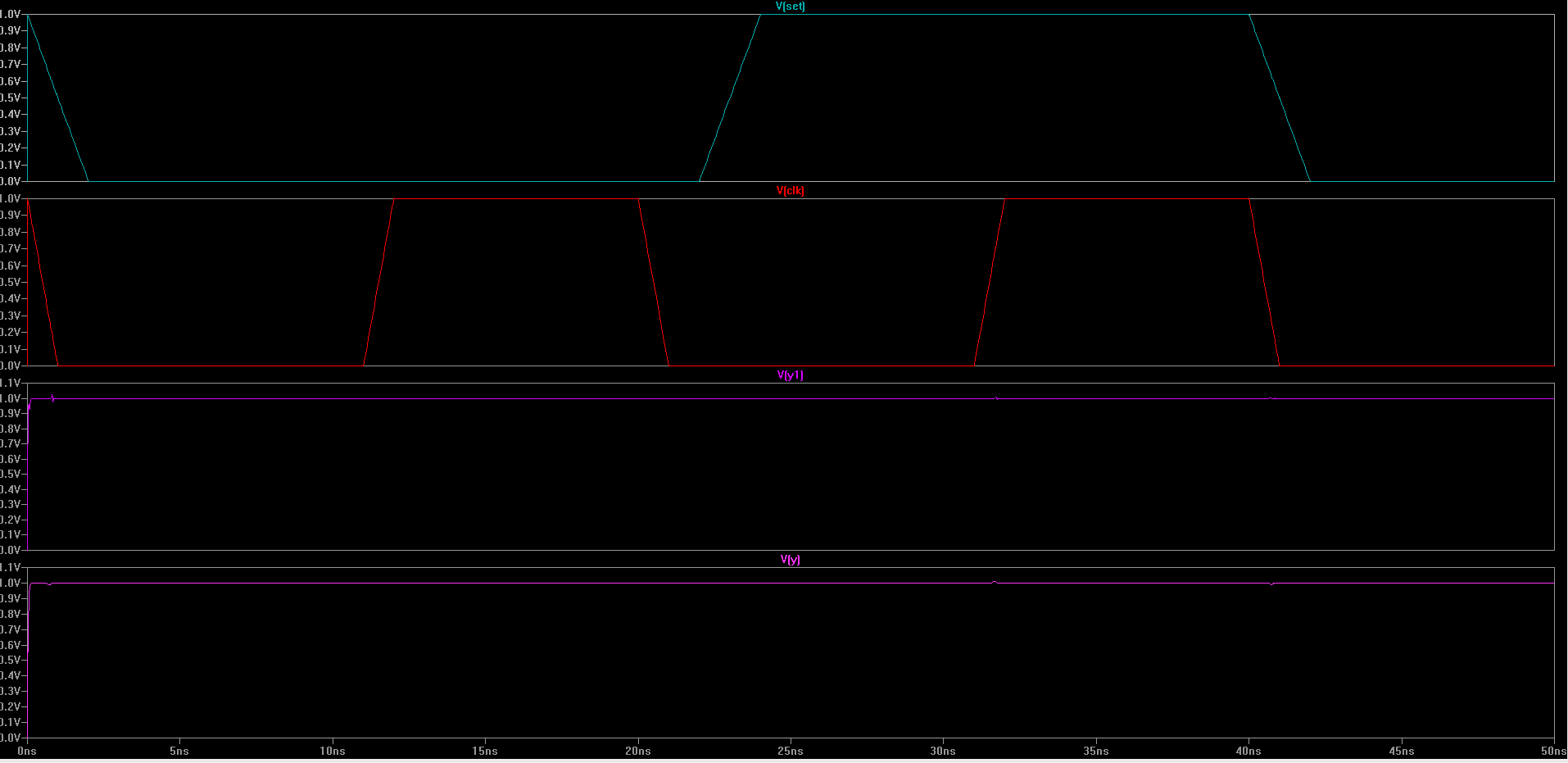
2. Thiết kế Dlatch HOẶC DFlipFlop có:

a. Xung clock cạnh lên.

b. Có 2 chân cho phép S1, S0 chọn một trong 4 giá trị D1, D2, D3, D4 cho ngõ vào D của FF.

c. Set mức cao đồng bộ với xung clock



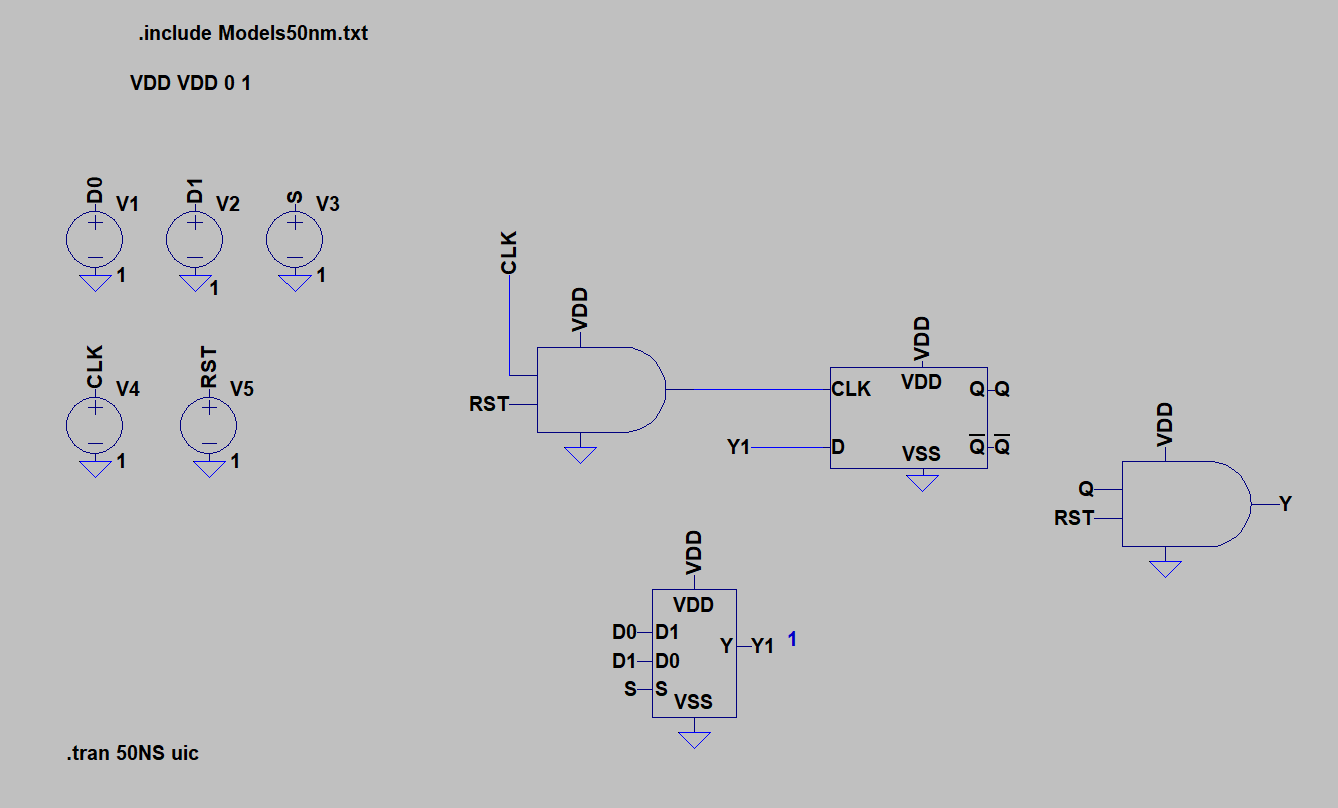


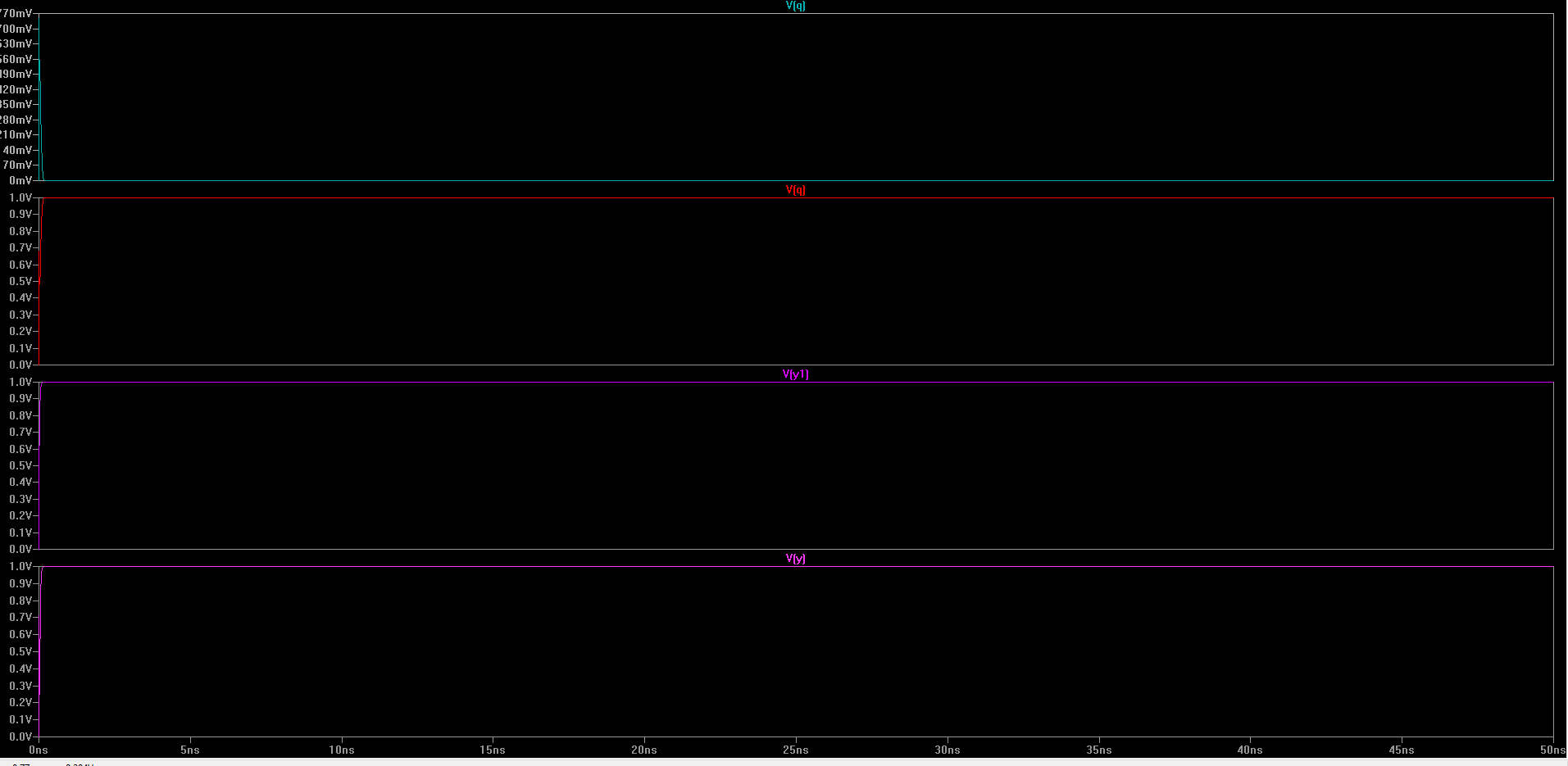
3. Thiết kế Dlatch HOẶC DFlipFlop có:

a. Xung clock cạnh lên.

b. Có chân cho phép S chọn một trong hai giá trị D1, D2 cho ngõ vào D của FF.

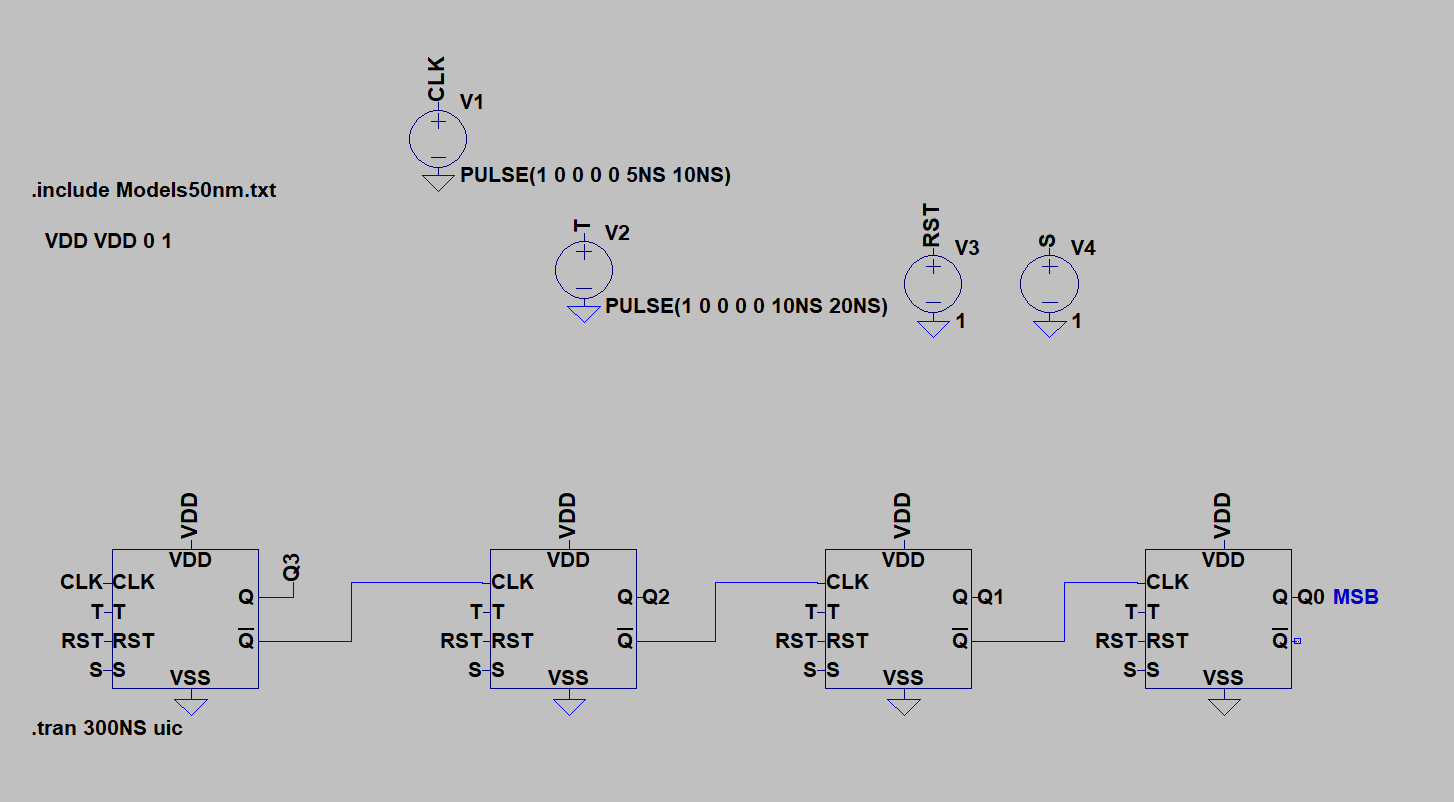
c. Reset mức thấp bất đồng bộ với xung clock.

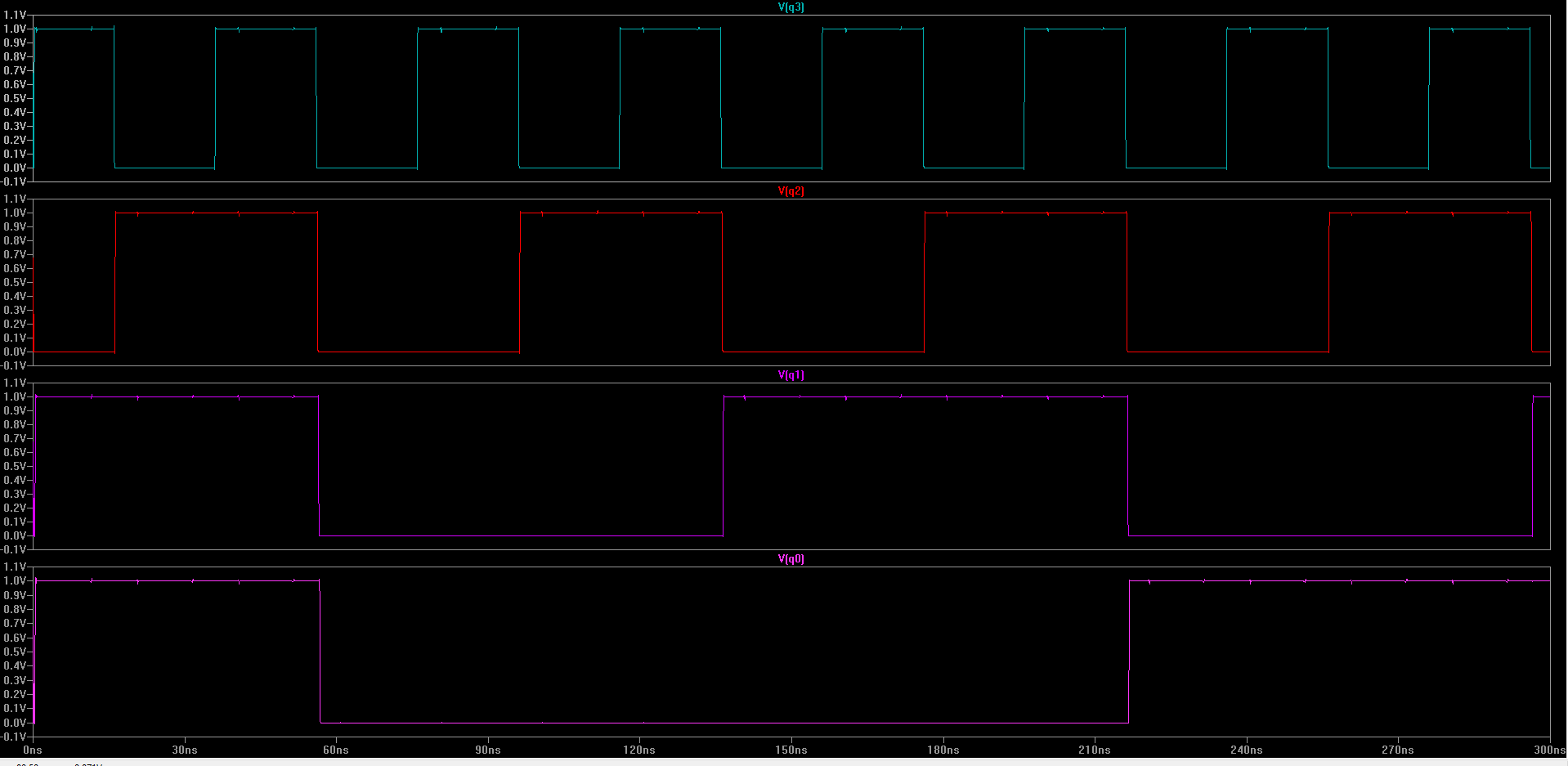




4. Thiết kế mạch đếm nối tiếp mod 16 đếm lên dùng T-FF (xung clock cạnh lên,

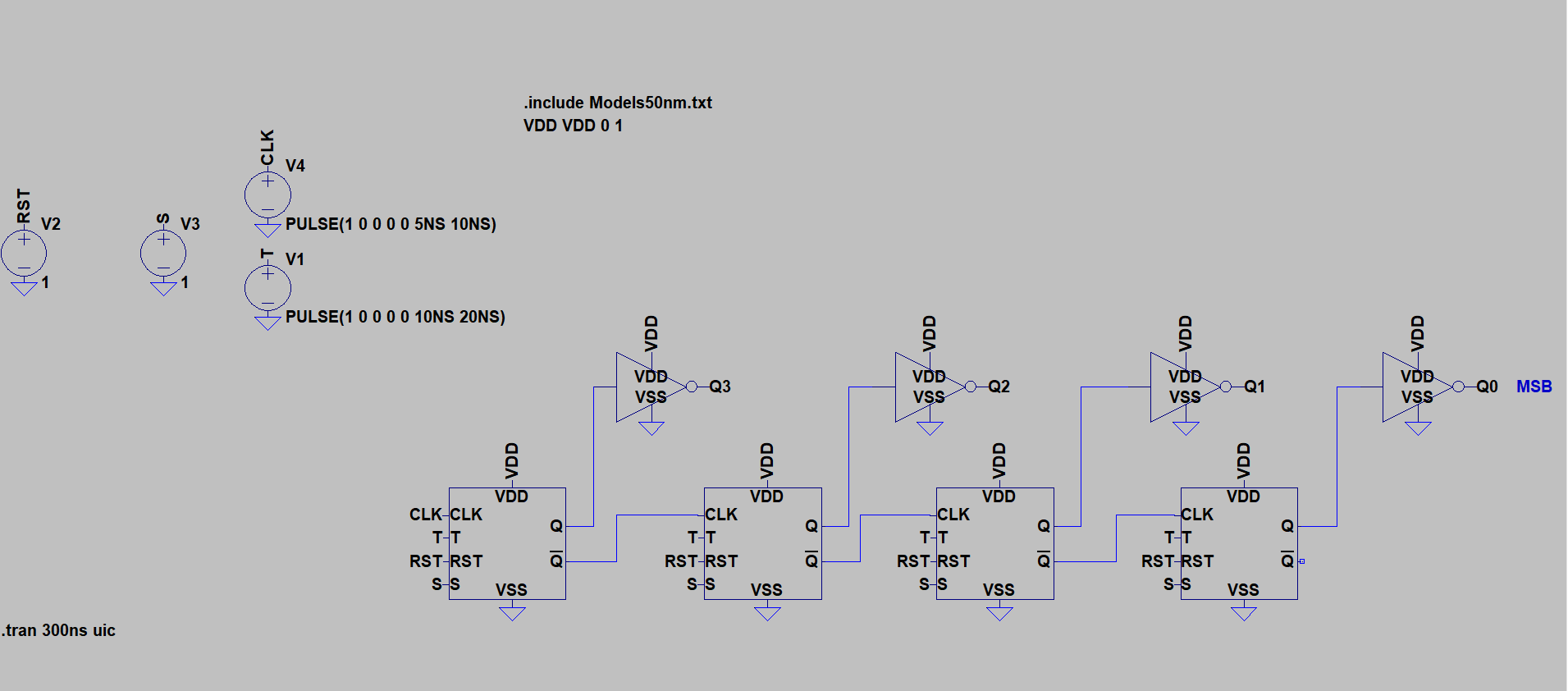
ngõ Pr và ngõ Cl tích cực mức thấp).

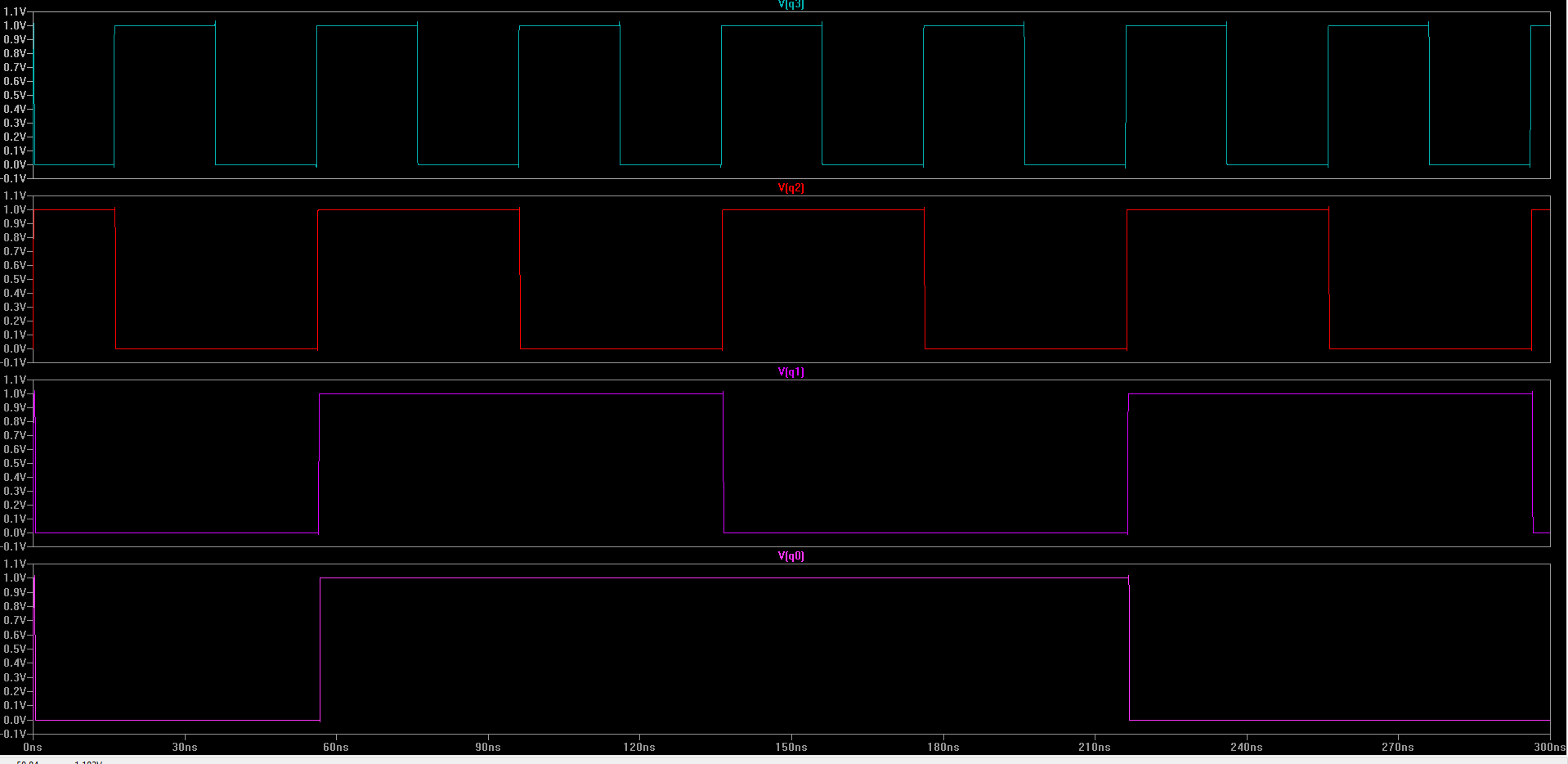


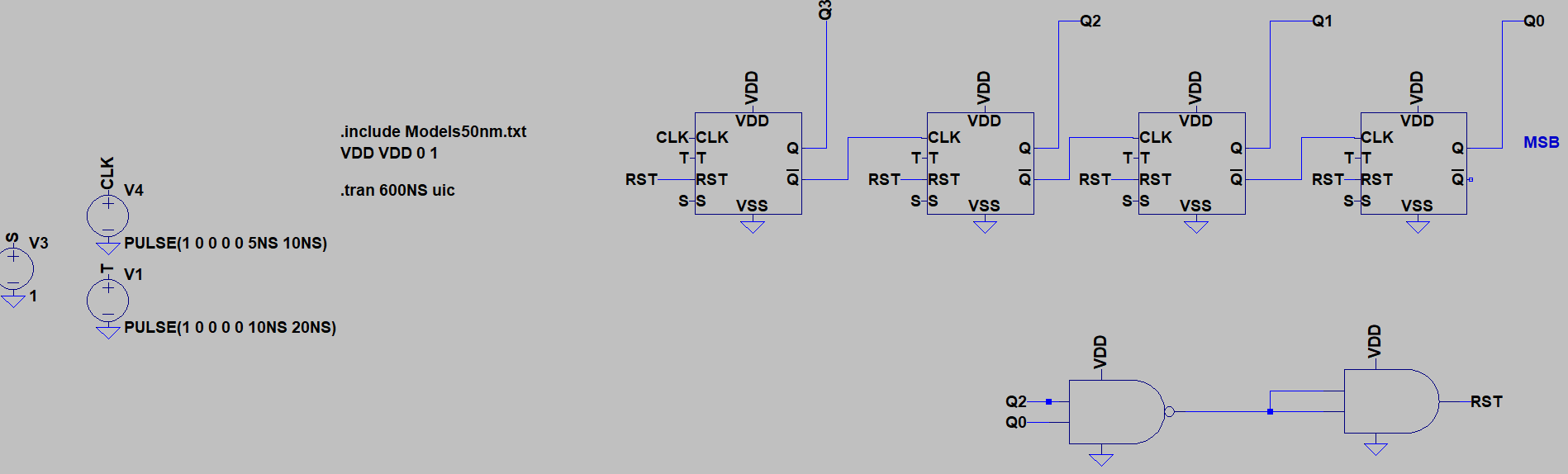
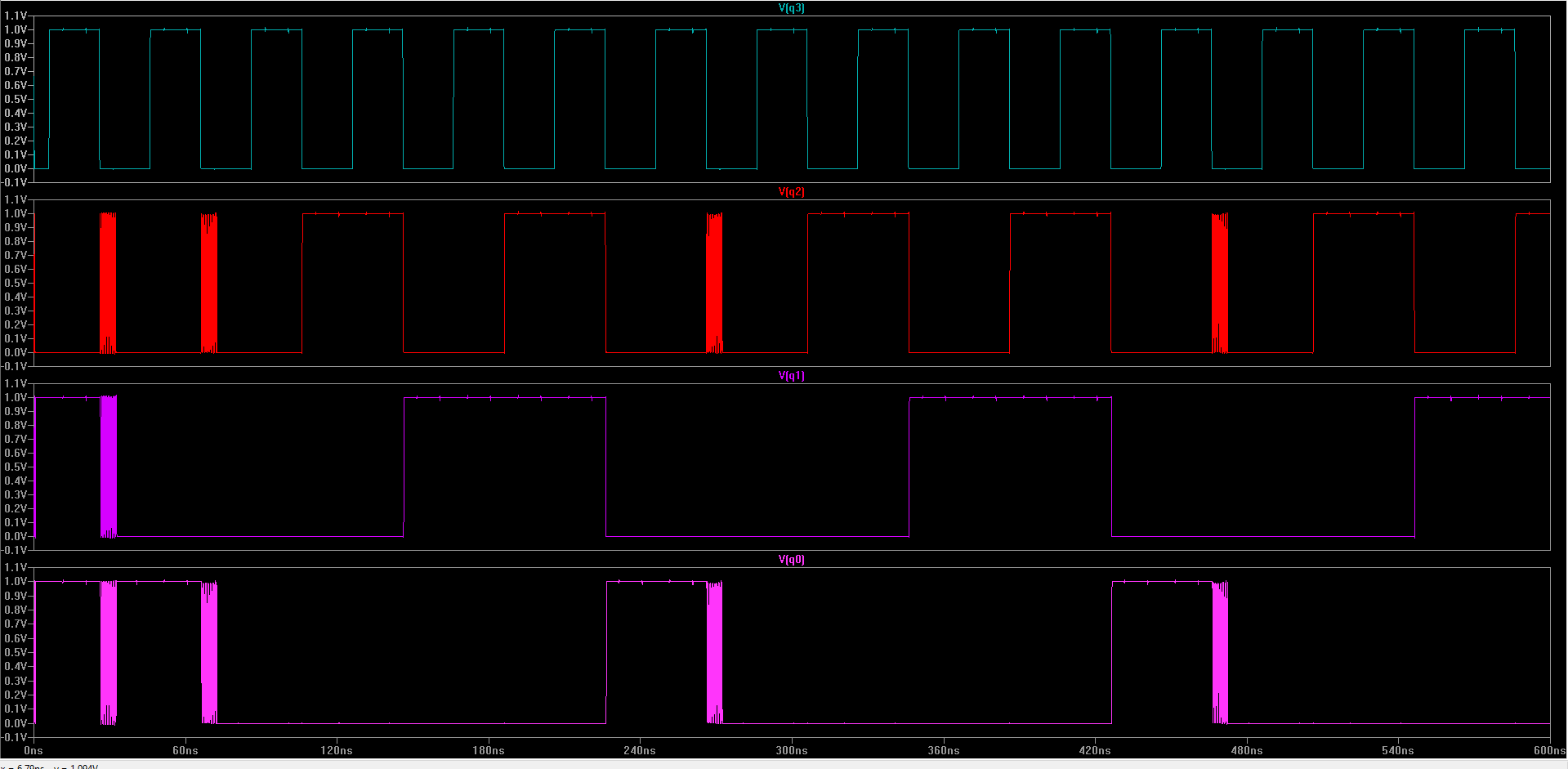


5. Thiết kế mạch đếm nối tiếp mod 16 đếm xuống dùng T-FF (xung clock cạnh

lên, ngõ Pr và ngõ Cl tích cực mức thấp)



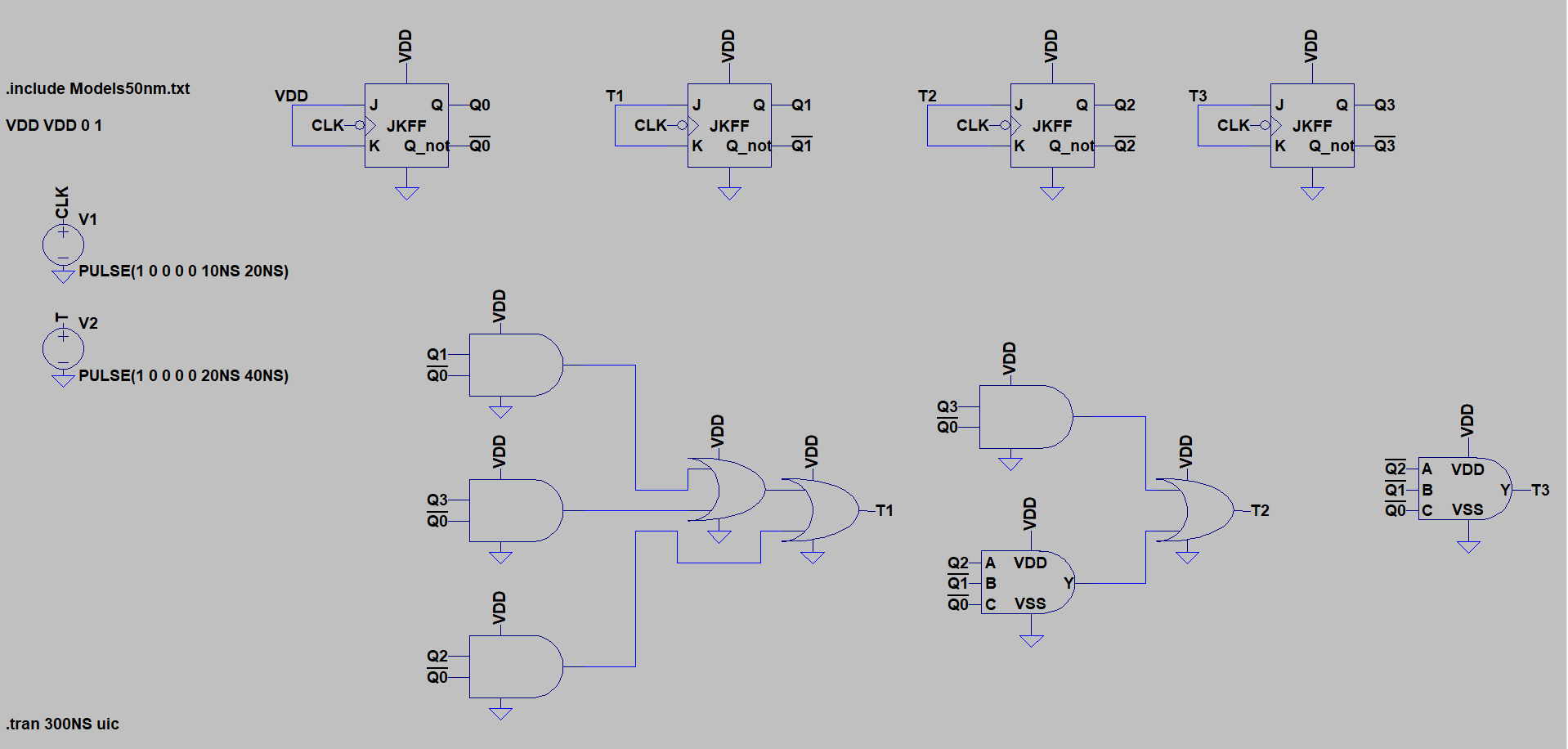


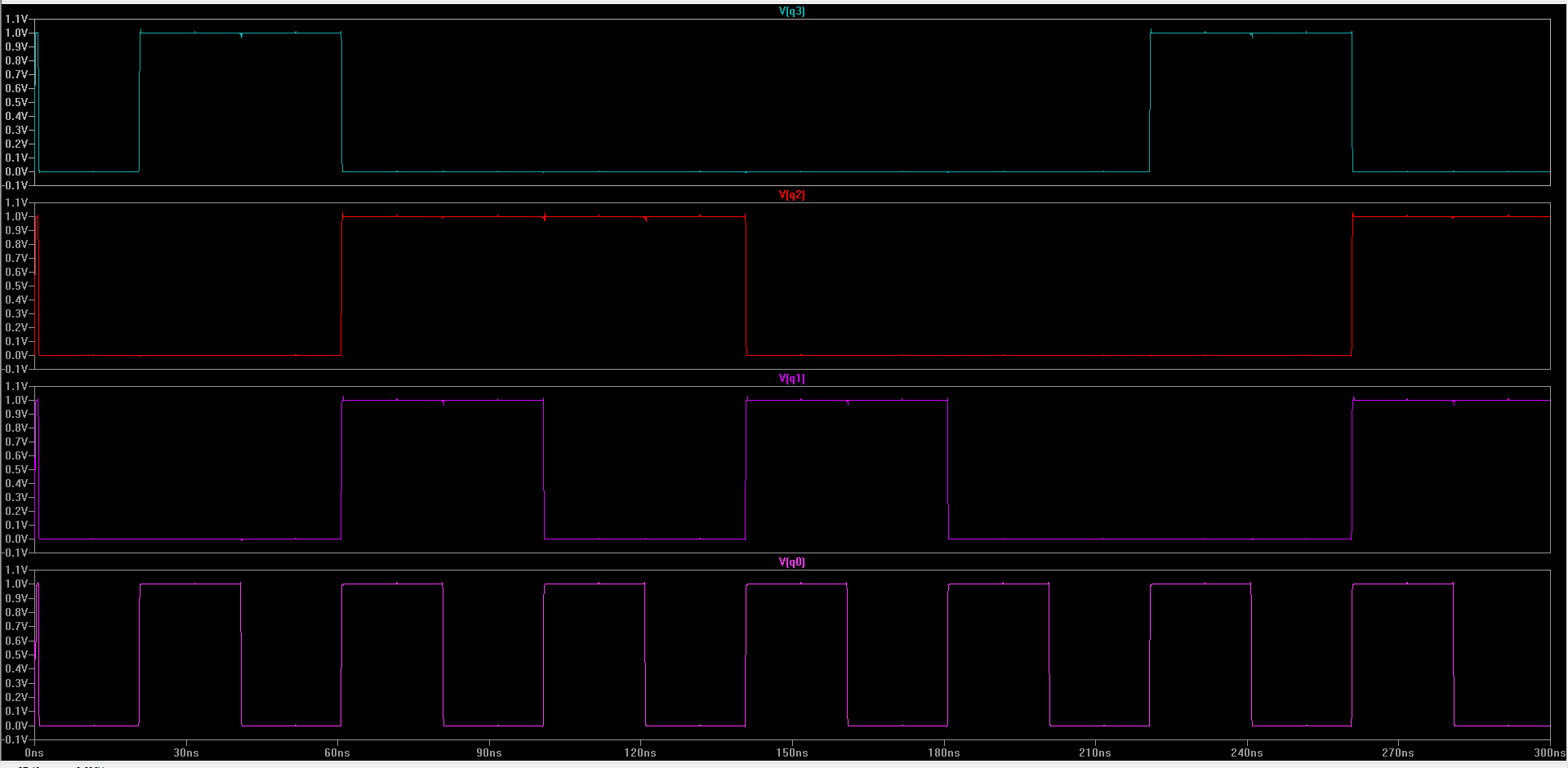
1. Dựa trên kết quả bài 4, thiết kế mạch đếm nối tiếp mod 10 đếm lên 0→1→2→…→9→0→…  
2. 7. Dựa trên kết quả bài 5, thiết kế mạch đếm nối tiếp mod 10 đếm xuống

15→14→13→…→6→15→…

8. Dựa trên kết quả bài 5, thiết kế mạch đếm nối tiếp mod 10 đếm xuống

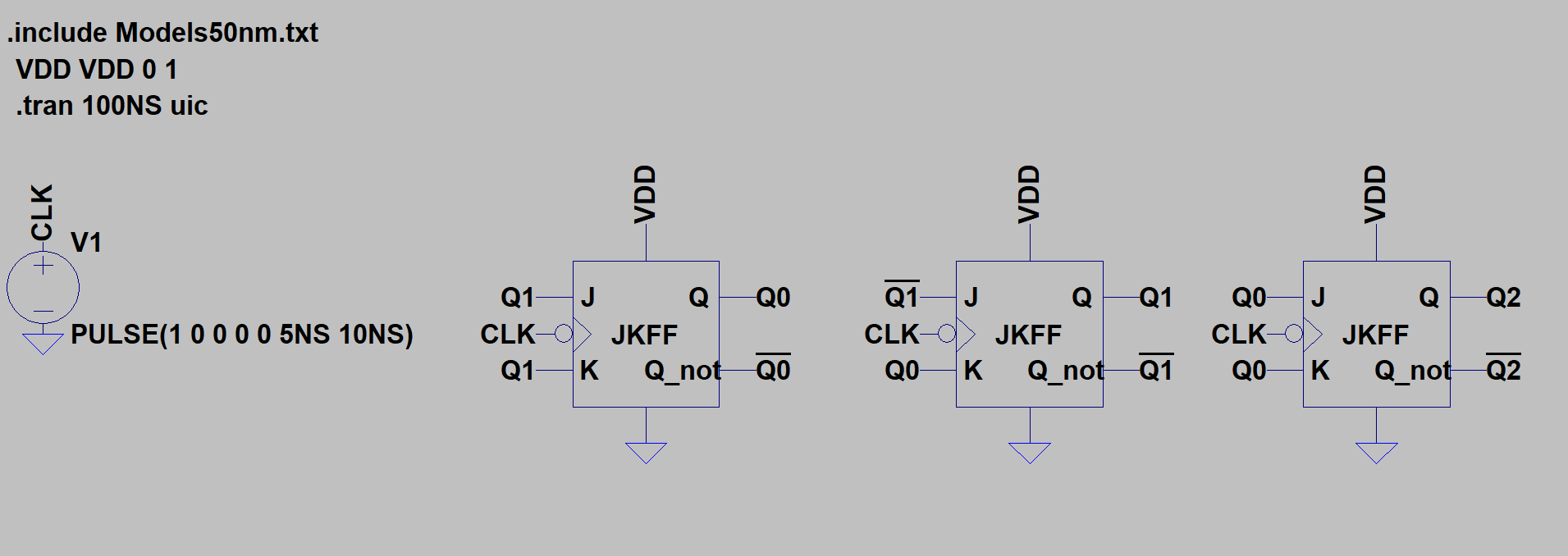
9→8→7→…→0→9→…

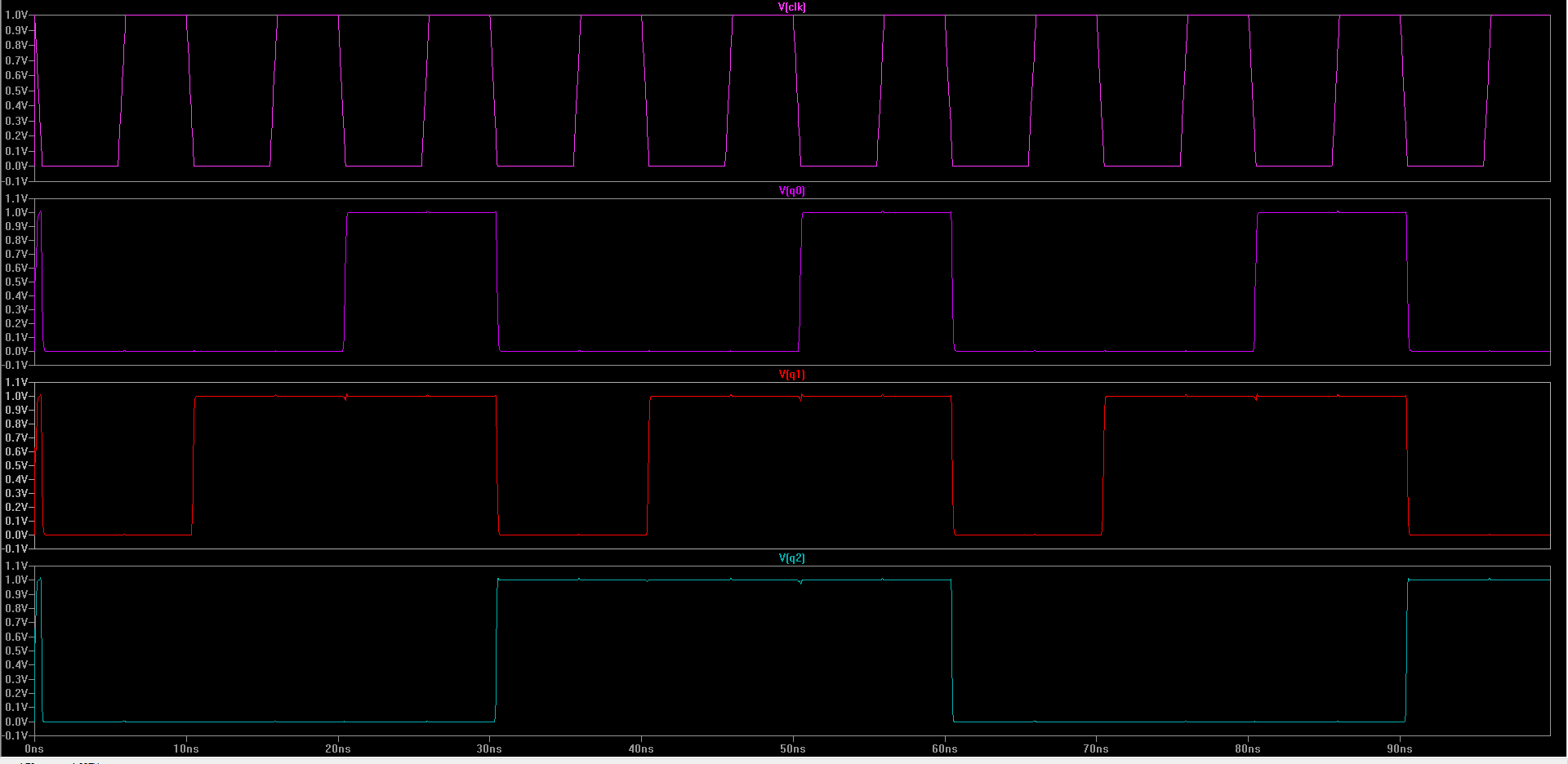




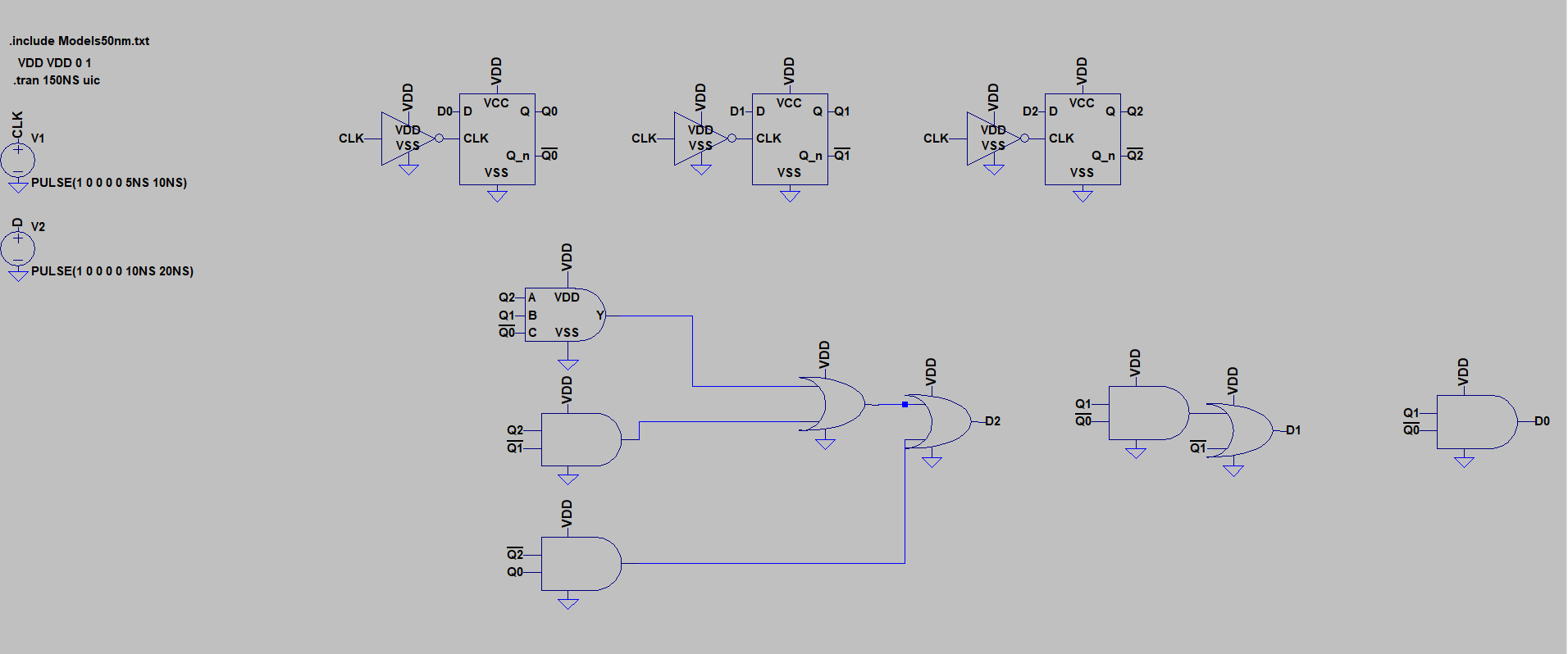
9. Thiết kế mạch đếm song song dùng JK-FF (xung clock cạnh xuống) có dãy

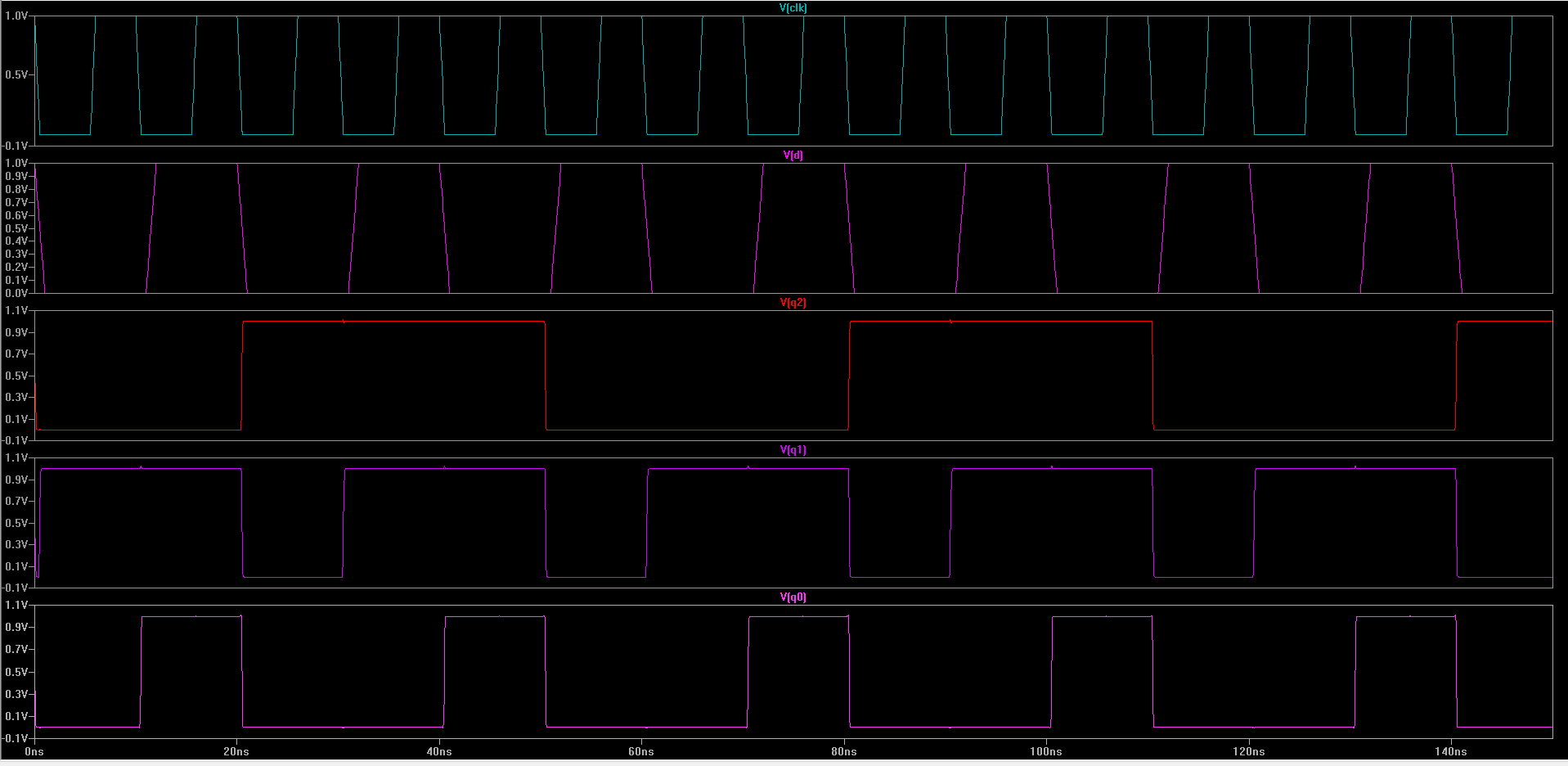
đếm như sau 000→010→011→100→110→111→000→…

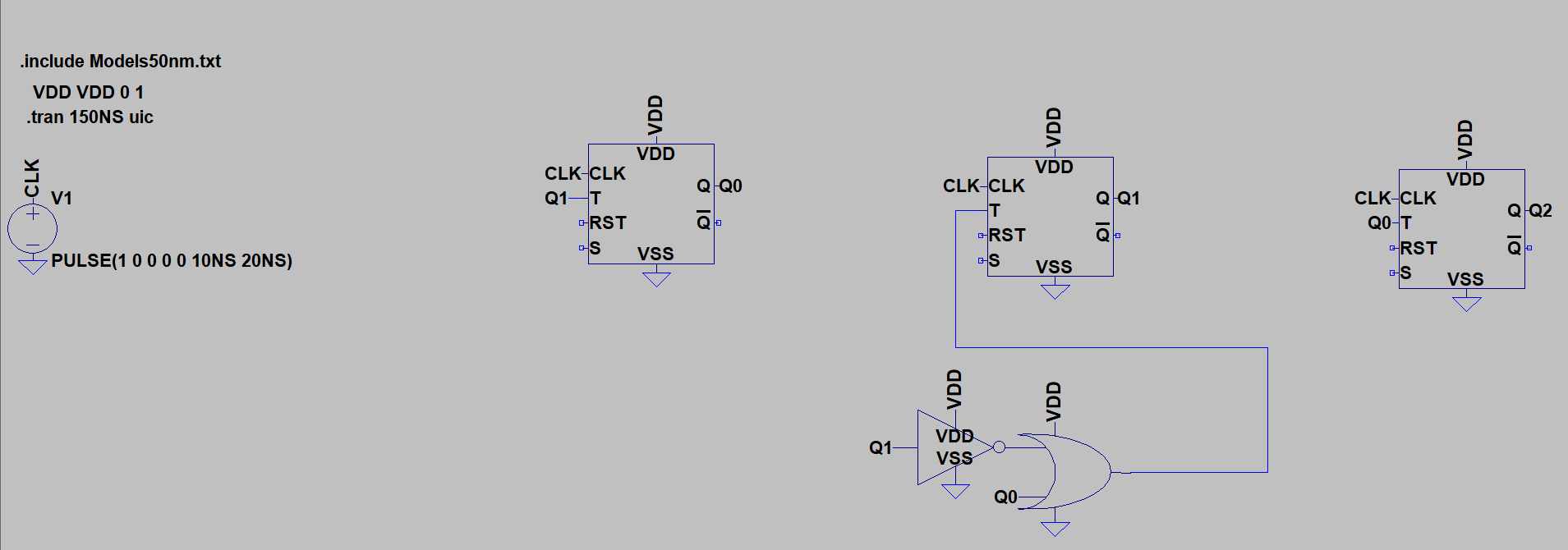


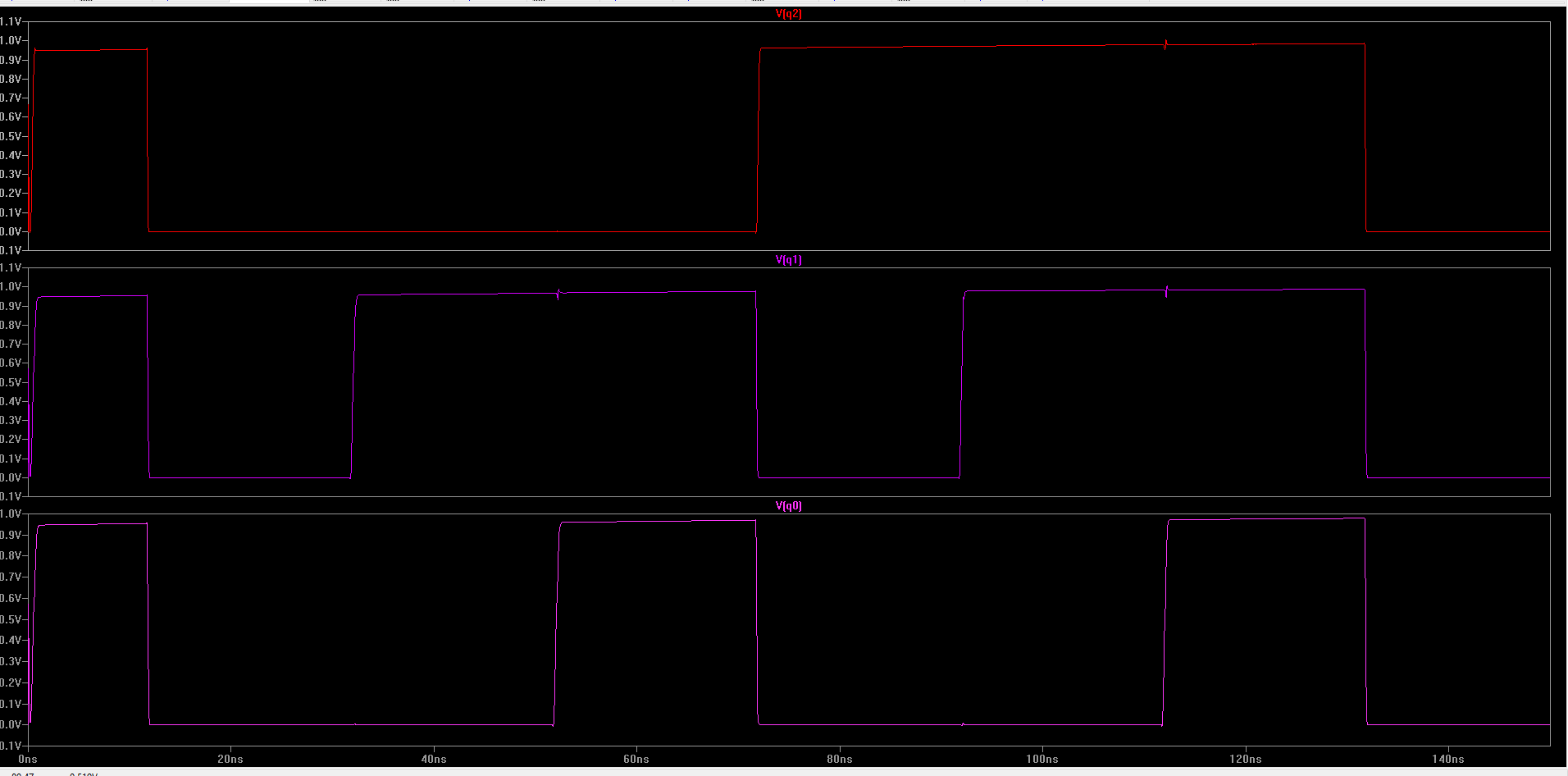


10.Làm lại bài 9 dùng D-FF.

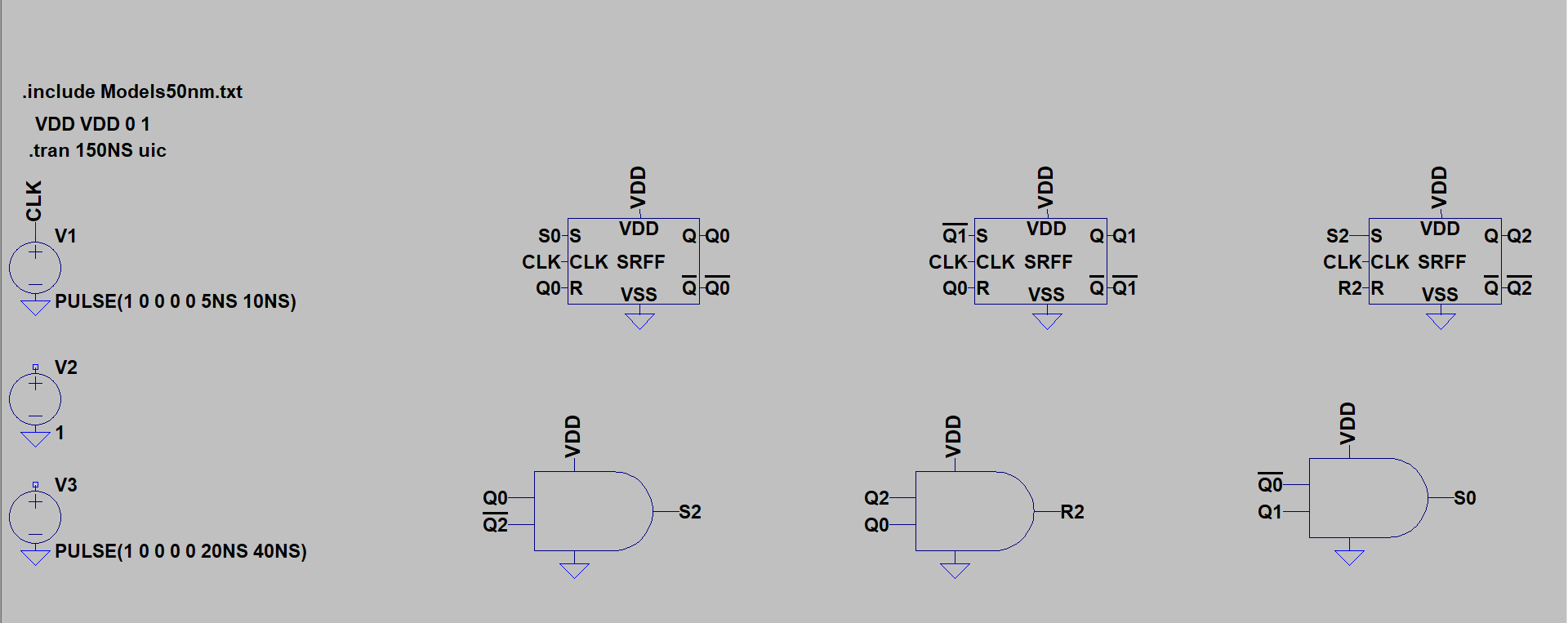


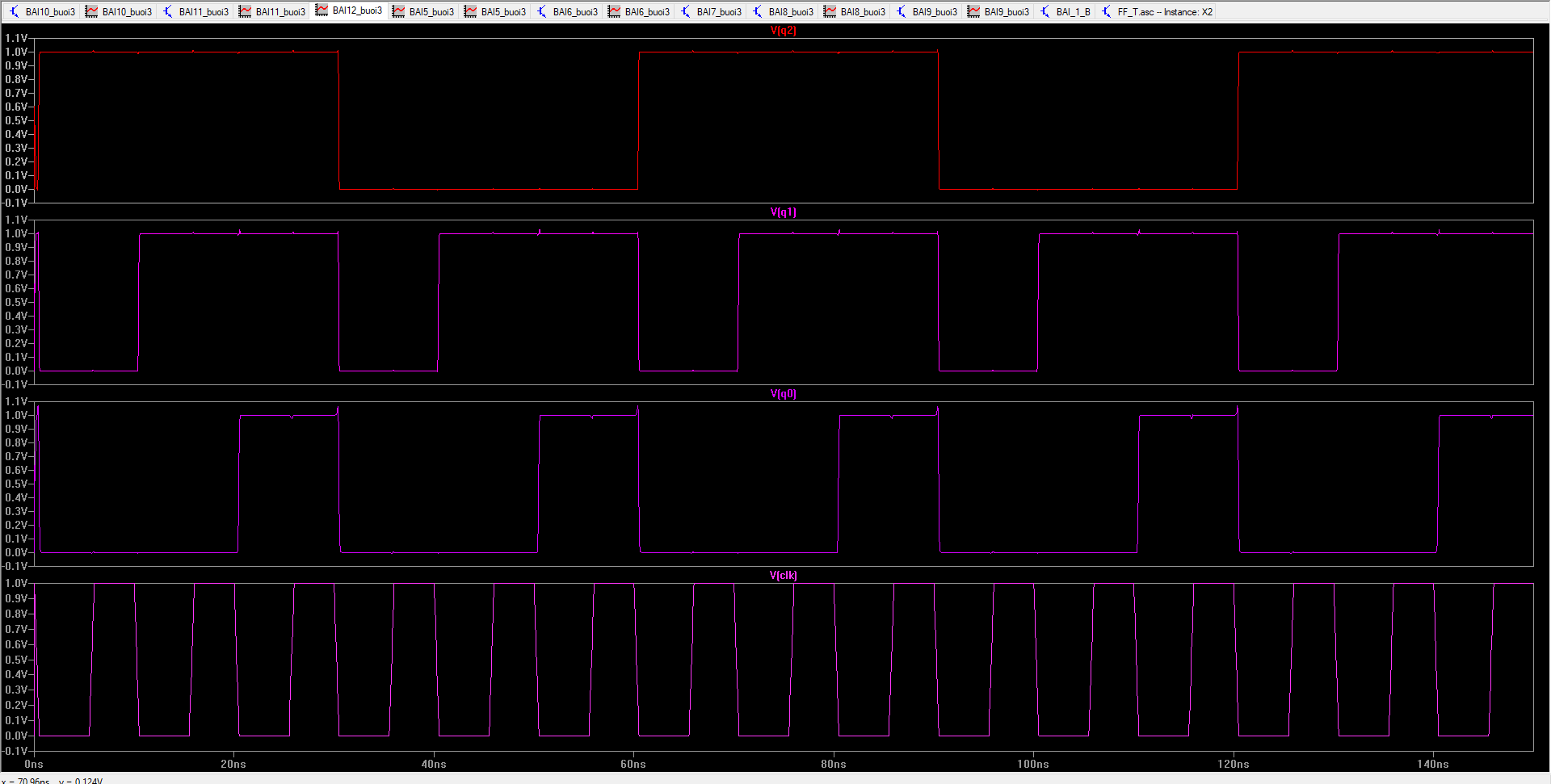
**11. Làm lại bài 9 dùng T-FF**





**12. Làm lại bài 9 dùng SR-FF**





## BUỔI 5

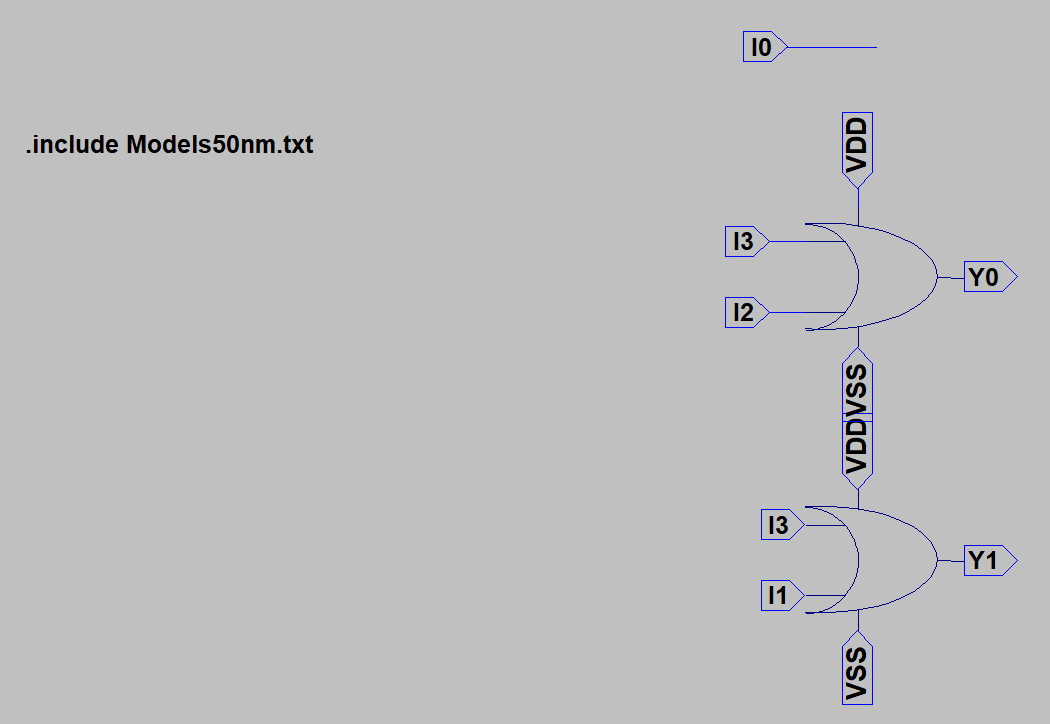
**1. Bộ mã hóa (Encoder). Ứng dụng: Dùng để mã hóa tín hiệu trong các hệ thống**

**truyền thông.**

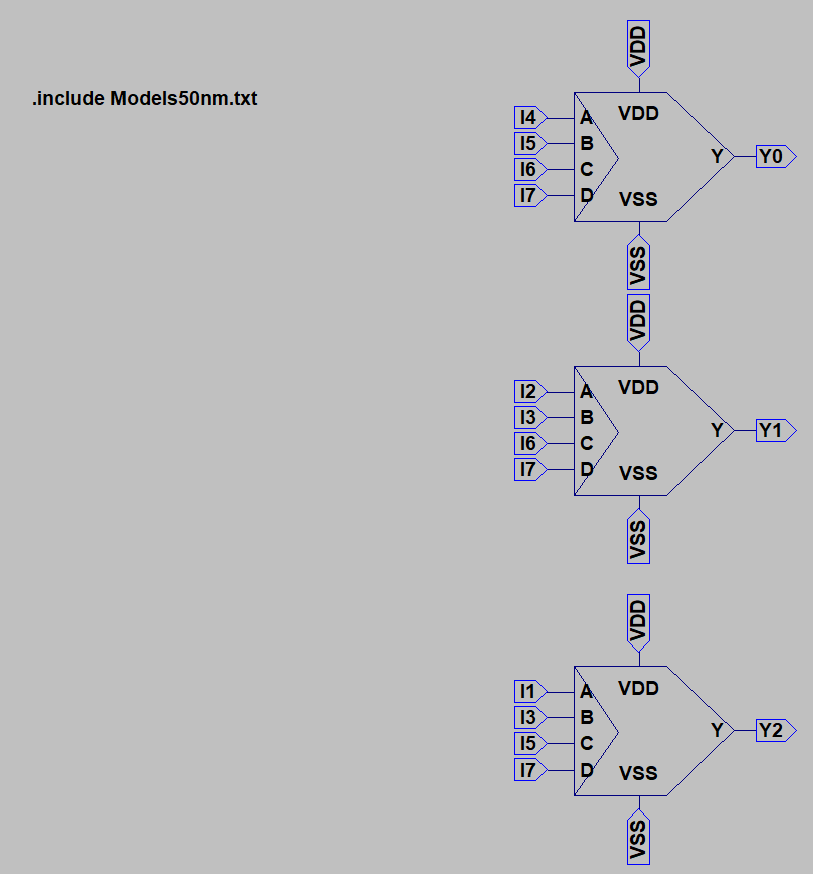
**a. Thiết kế một bộ mã hóa 4-2 (4 đầu vào, 2 đầu ra).**

**b. Thiết kế một bộ mã hóa 8-3 (8 đầu vào, 3 đầu ra)**

a)



b)



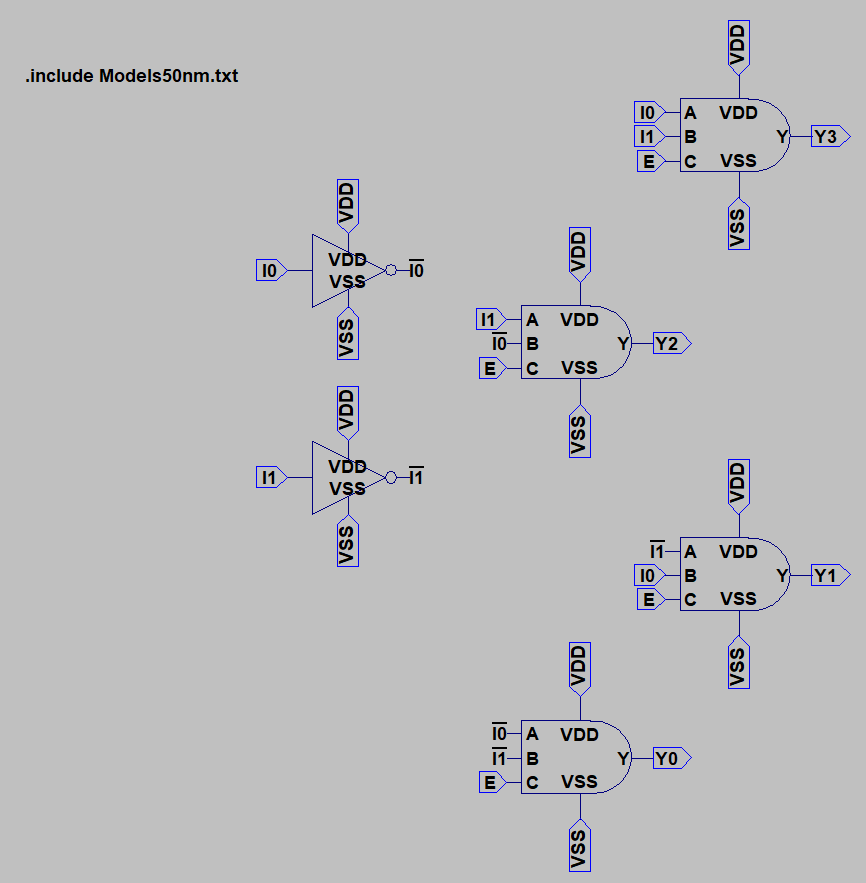
**2. Bộ giải mã (Decoder). Ứng dụng: Dùng trong các thiết bị chọn kênh hoặc điều**

**khiển**

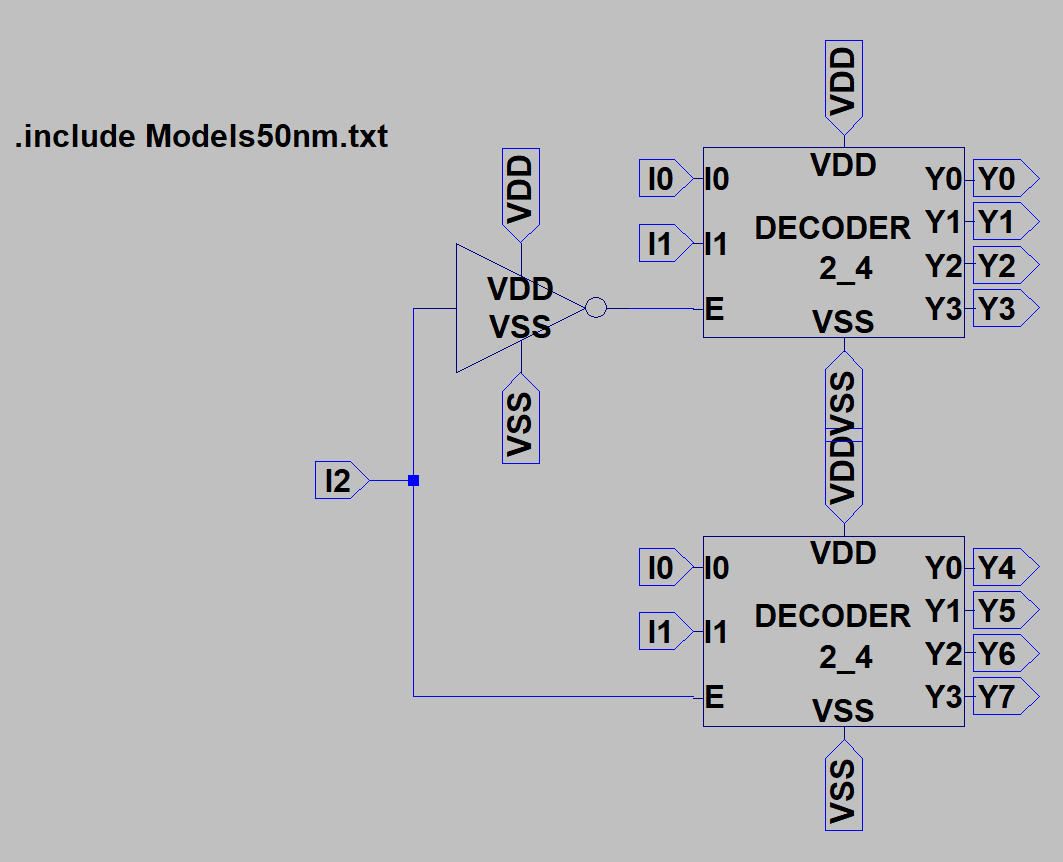
**a. Thiết kế một bộ giải mã 2-4 (2 đầu vào, 4 đầu ra).**

**b. Thiết kế một bộ giải mã 3-8 (3 đầu vào, 8 đầu ra)**

a)



b)

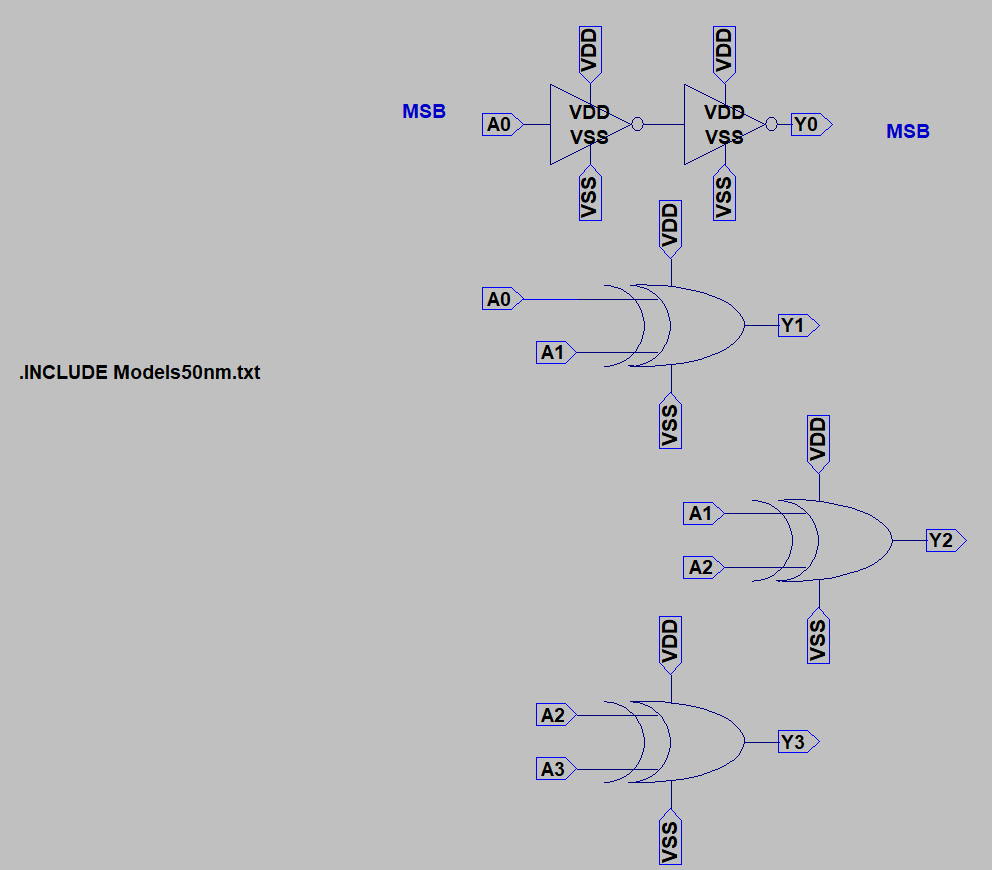


**3. Bộ chuyển đổi mã (Code Converter). Ứng dụng: Dùng để giảm lỗi trong các hệ thống truyền tín hiệu**

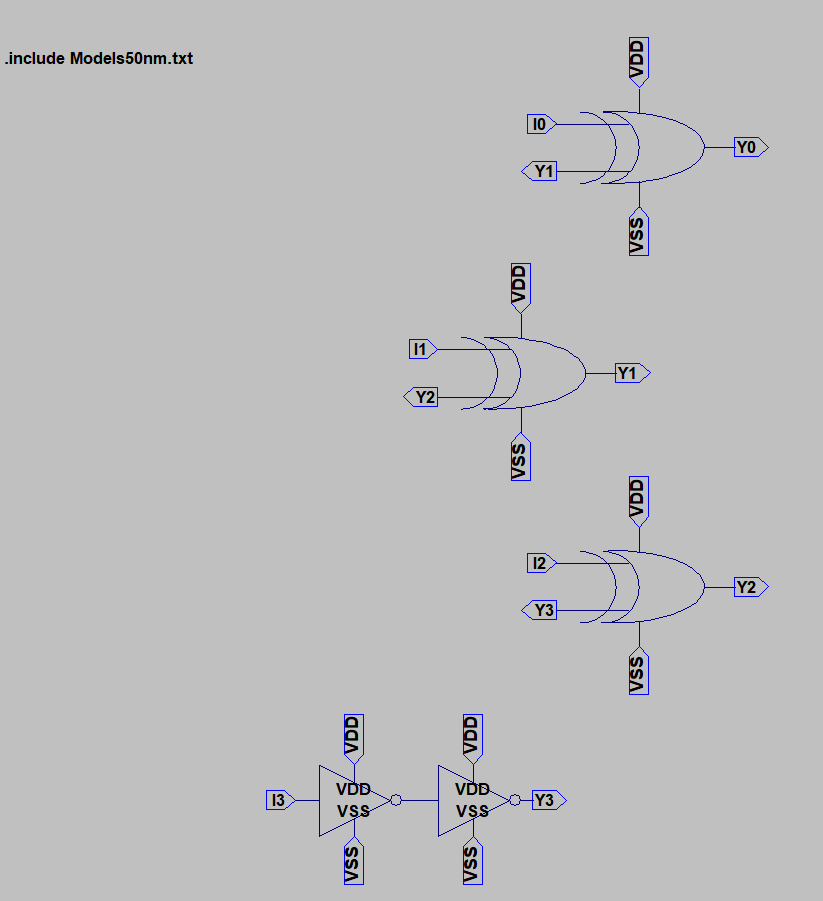
**a. Thiết kế mạch chuyển đổi từ mã nhị phân (Binary) 4 bit sang mã Gray**

**b. Thiết kế mạch chuyển đổi từ mã Gray 4 bit sang mã nhị phân (Binary)**

a)



b)



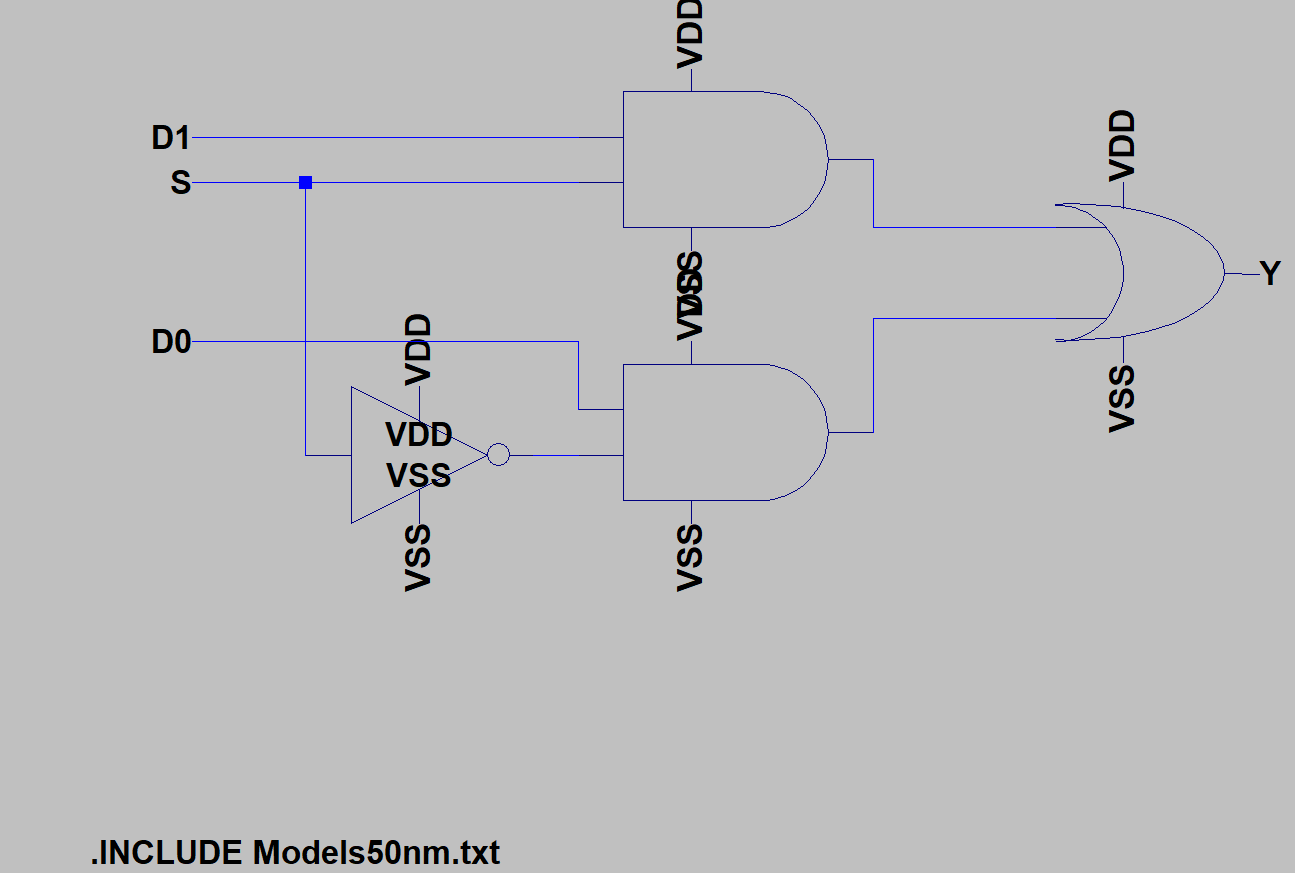
**4. Bộ tạo tín hiệu đa hợp (Multiplexer). Ứng dụng: Dùng trong việc chọn tín hiệu**

**trong các hệ thống truyền thông**

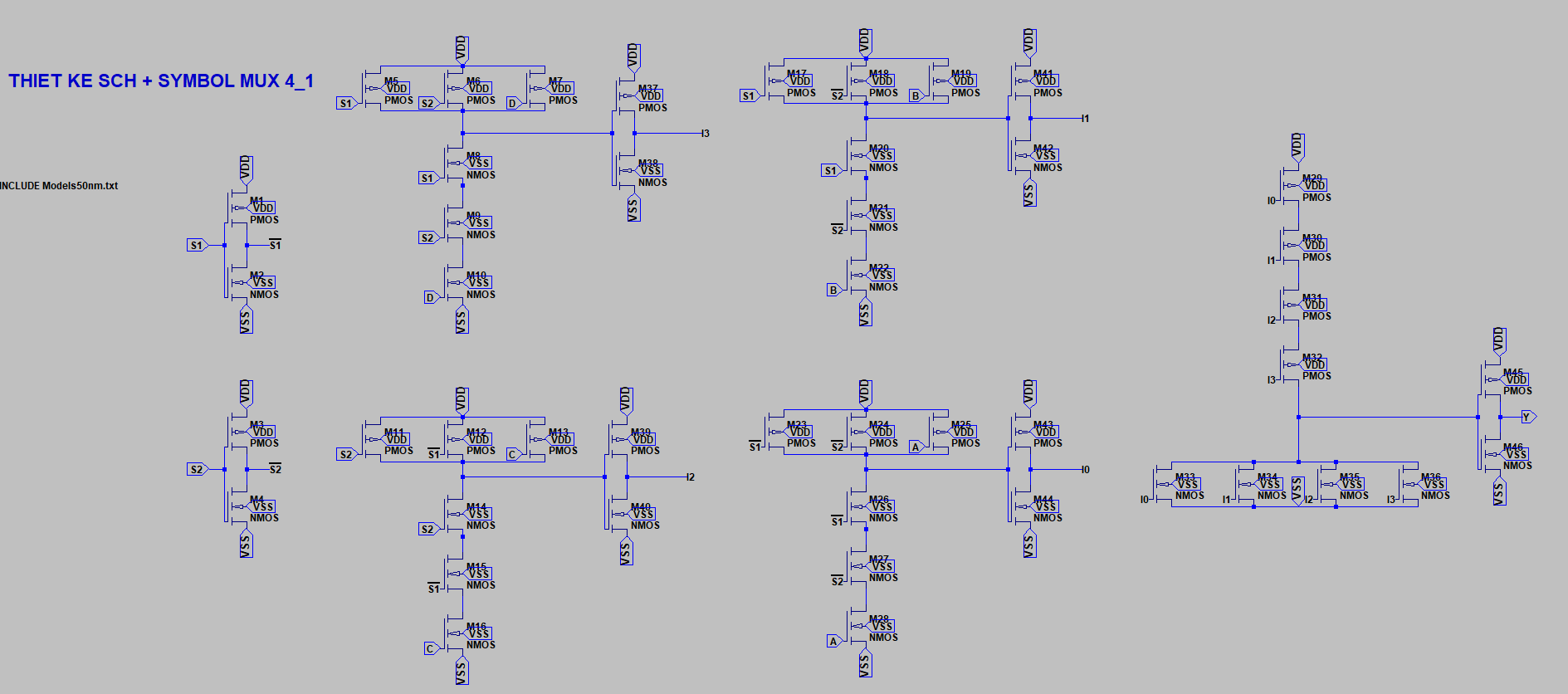
**a. Thiết kế mạch đa hợp 2-1 (Multiplexer 2 đầu vào, 1 đầu ra).**

**b. Thiết kế mạch đa hợp 4-1 (Multiplexer 4 đầu vào, 1 đầu ra)**

a)



b)

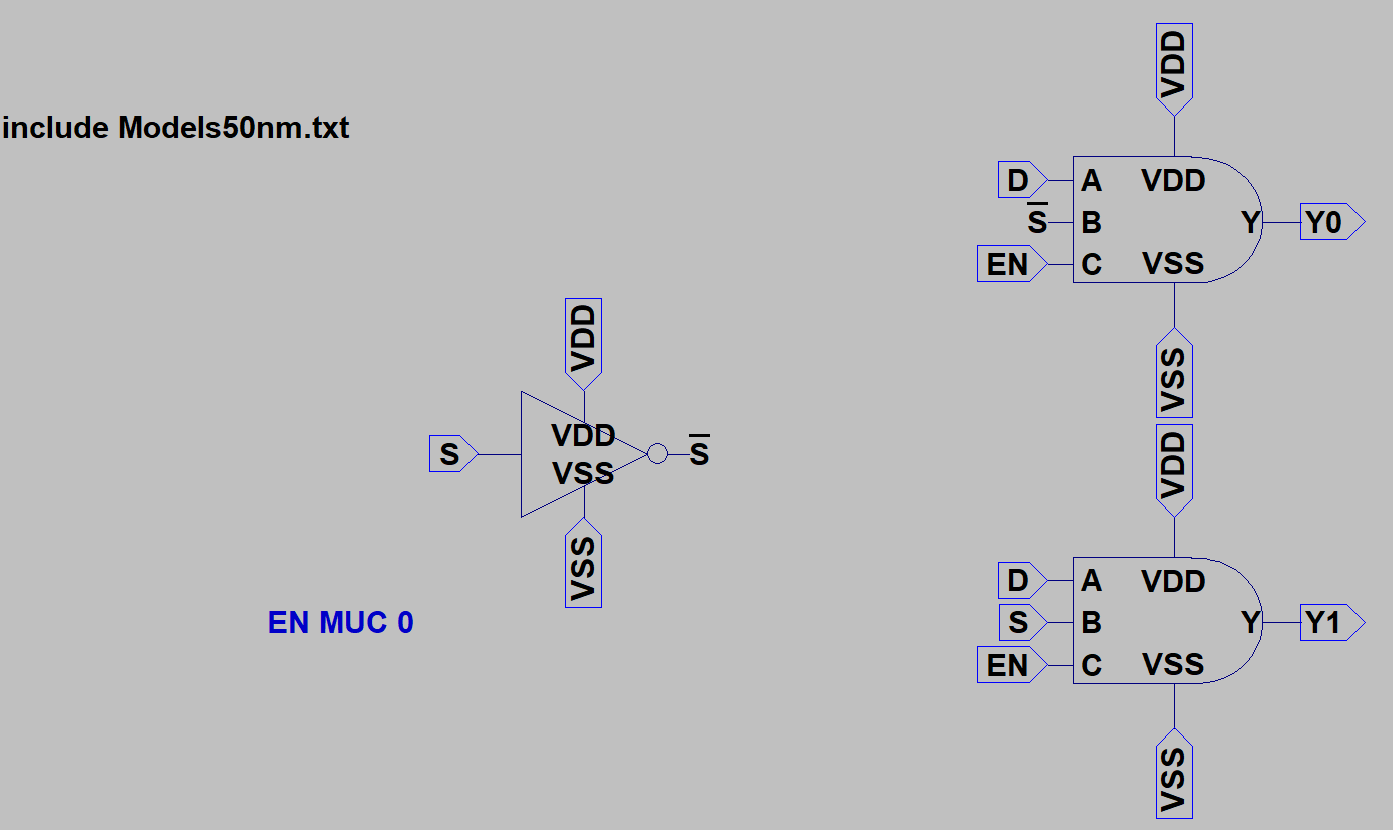


**5. Bộ phân tín hiệu (Demultiplexer). Ứng dụng: Dùng trong các hệ thống điều khiển hoặc phân phối tín hiệu.**

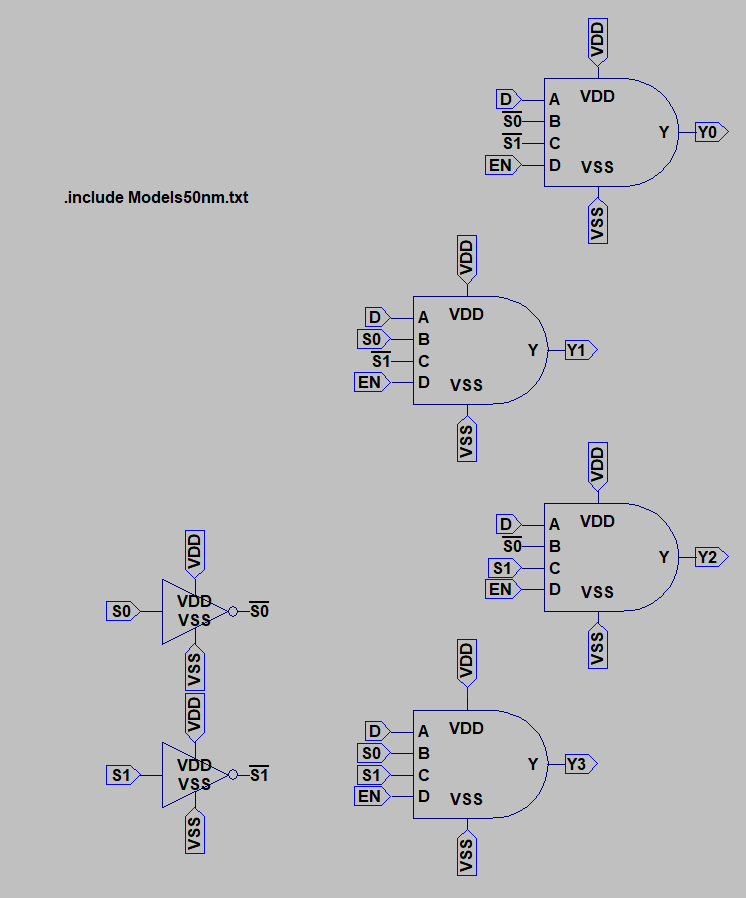
**a. Thiết kế mạch phân tín hiệu 1-2 (Demultiplexer 1 đầu vào, 2 đầu ra)**

**b. Thiết kế mạch phân tín hiệu 1-4 (Demultiplexer 1 đầu vào, 4 đầu ra)**

a)



b)

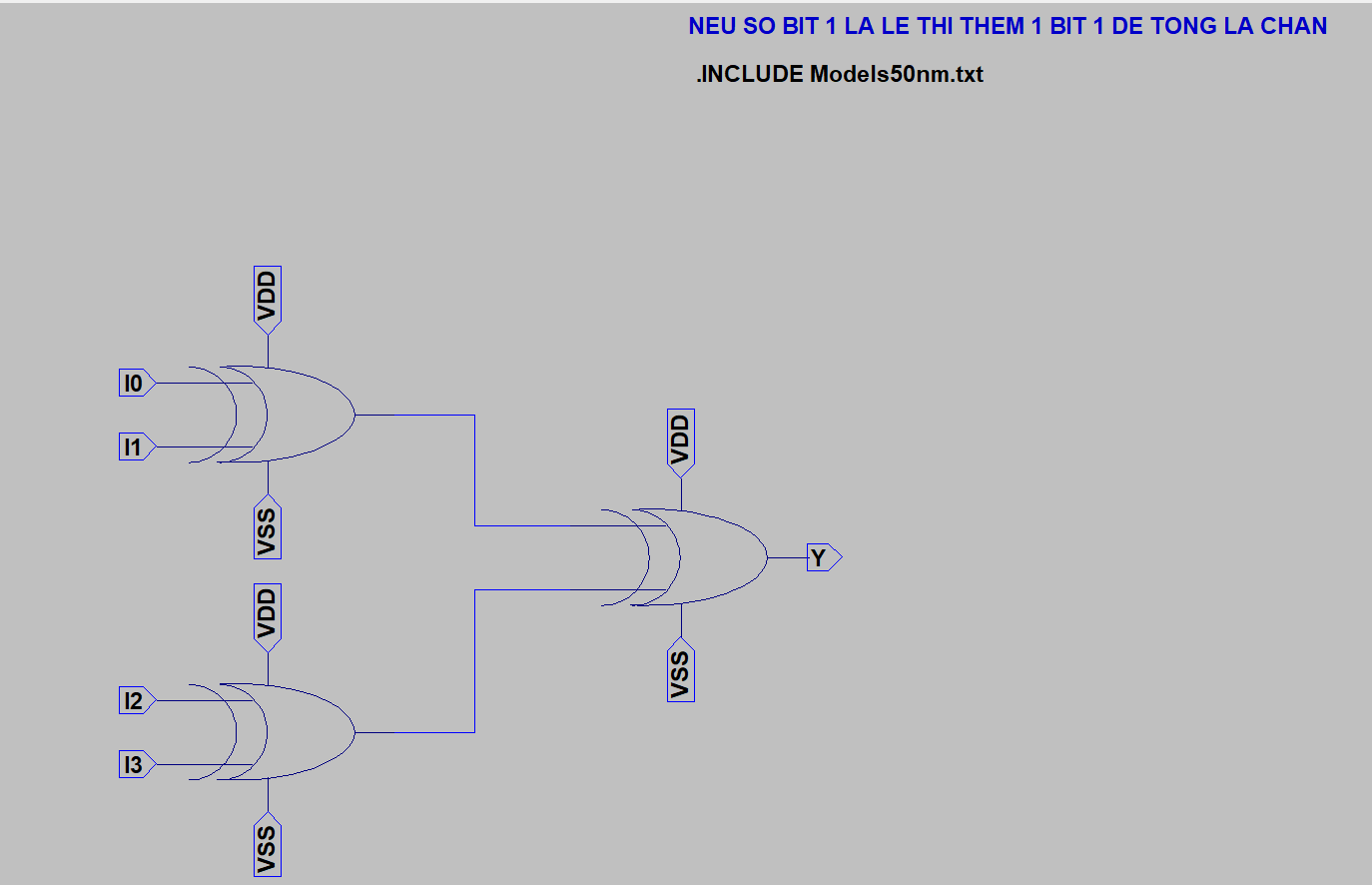


**6. Bộ tạo mã dư (Parity Generator). Ứng dụng: Dùng trong các hệ thống kiểm tra lỗi truyền dữ liệu**

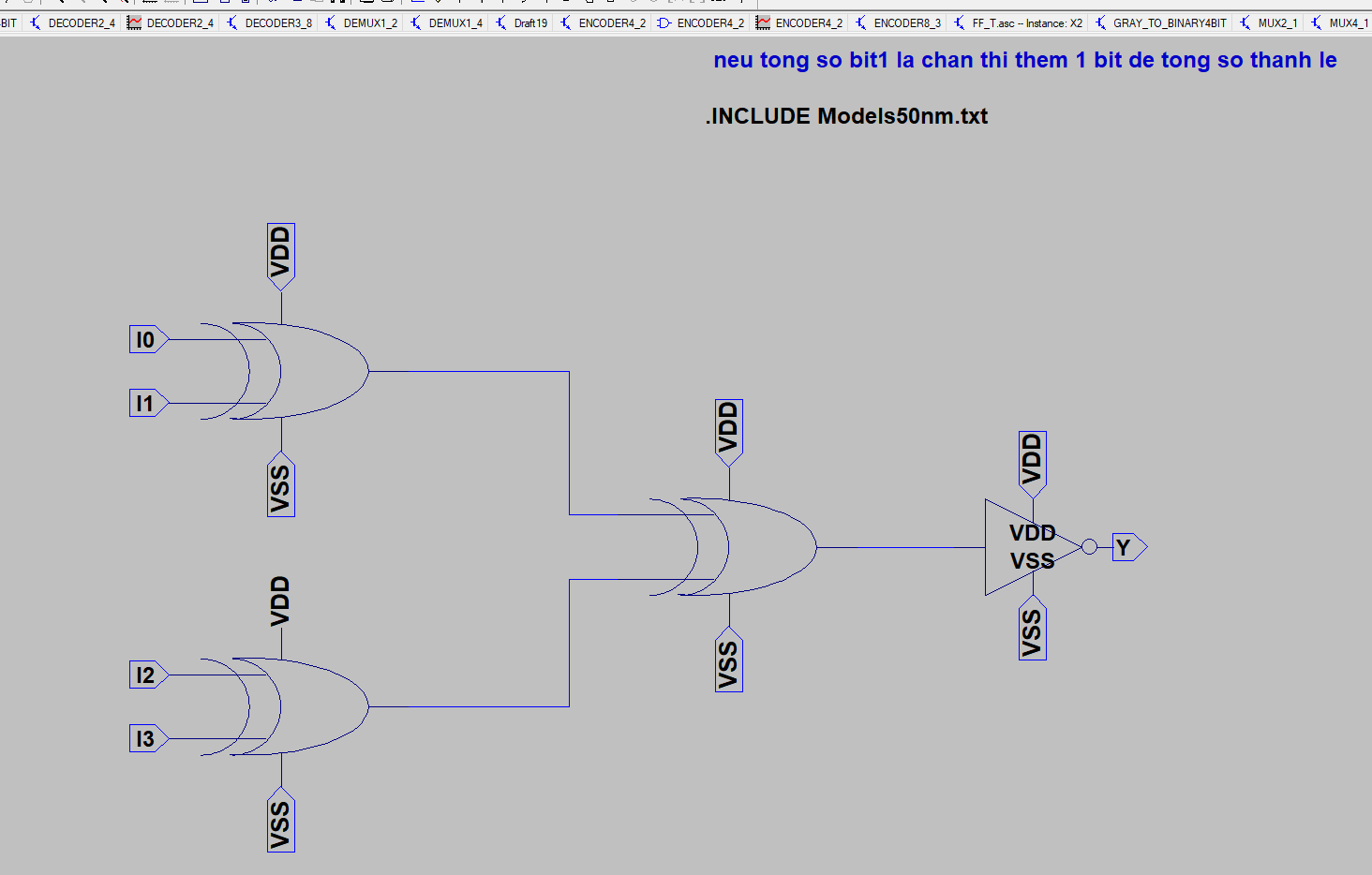
**a. Thiết kế mạch tạo bit chẵn (Parity Bit) cho 4 bit đầu vào. Nếu tổng số bit 1 là lẻ, thêm 1 bit để tổng là chẵn.**

**b. Thiết kế mạch tạo bit lẻ (Parity Bit) cho 4 bit đầu vào. Nếu tổng số bit 1 là chẵn, thêm 1 bit để tổng là lẻ.**

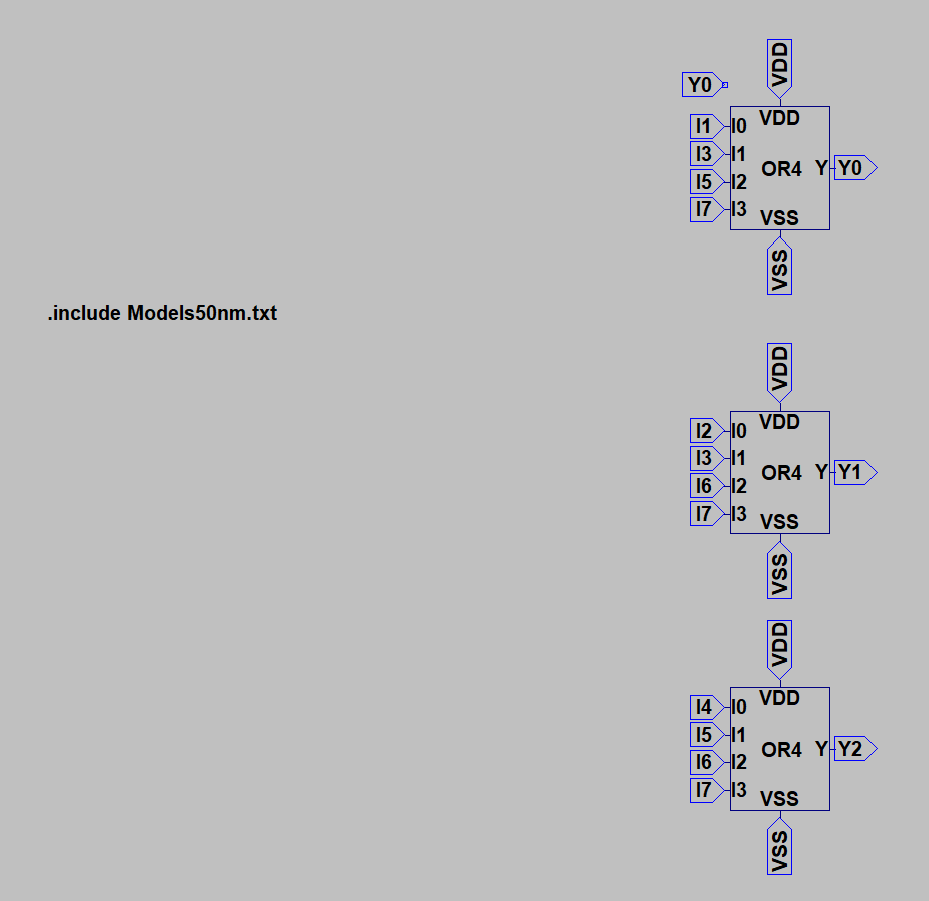
a)

****

b)

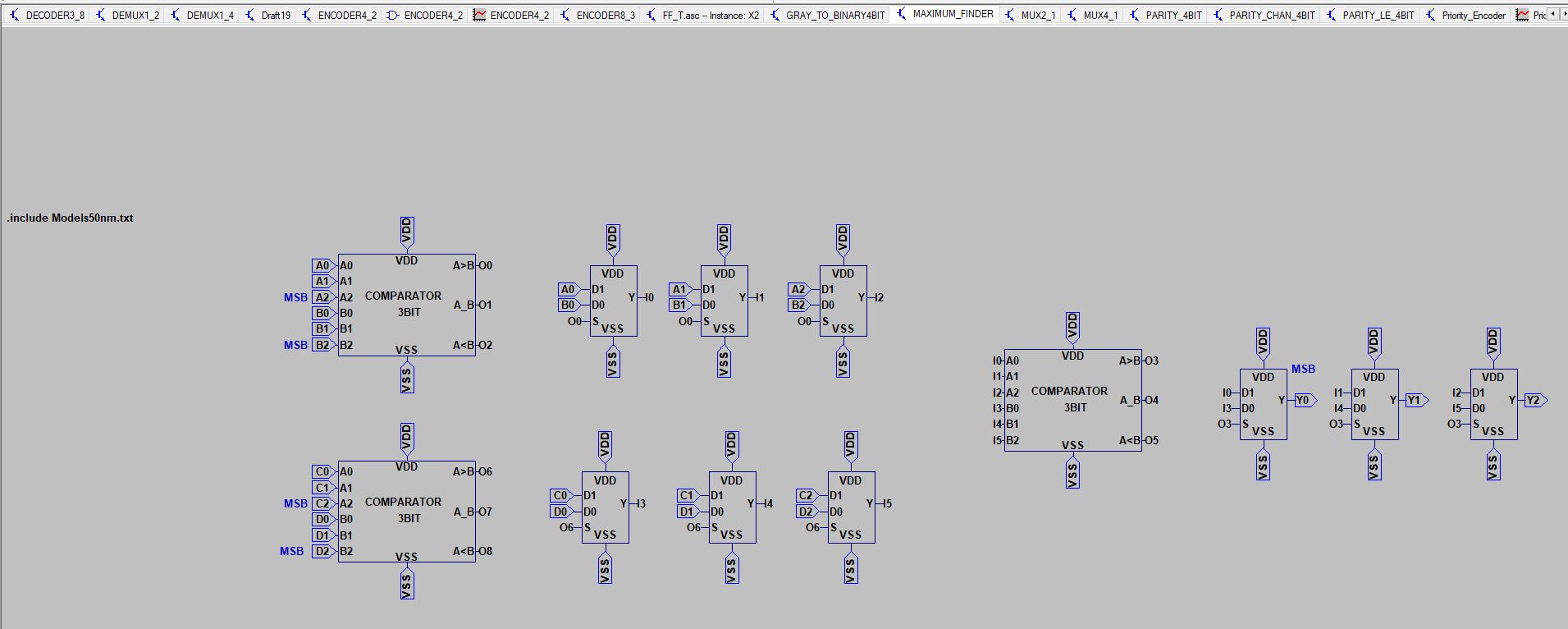


1. **Bộ mã hóa ưu tiên (Priority Encoder). Ứng dụng: Dùng trong các hệ thống xử lý tín hiệu ưu tiên. Thiết kế một bộ mã hóa ưu tiên 8-3. Nếu nhiều đầu vào cùng kích hoạt, mạch sẽ ưu tiên đầu vào cao nhất**



**9. Bộ chọn số lớn nhất (Maximum Finder). Ứng dụng: Dùng trong các hệ thống quyết định hoặc tối ưu hóa**

**Thiết kế mạch chọn số lớn nhất trong 4 số nhị phân 3 bit.**

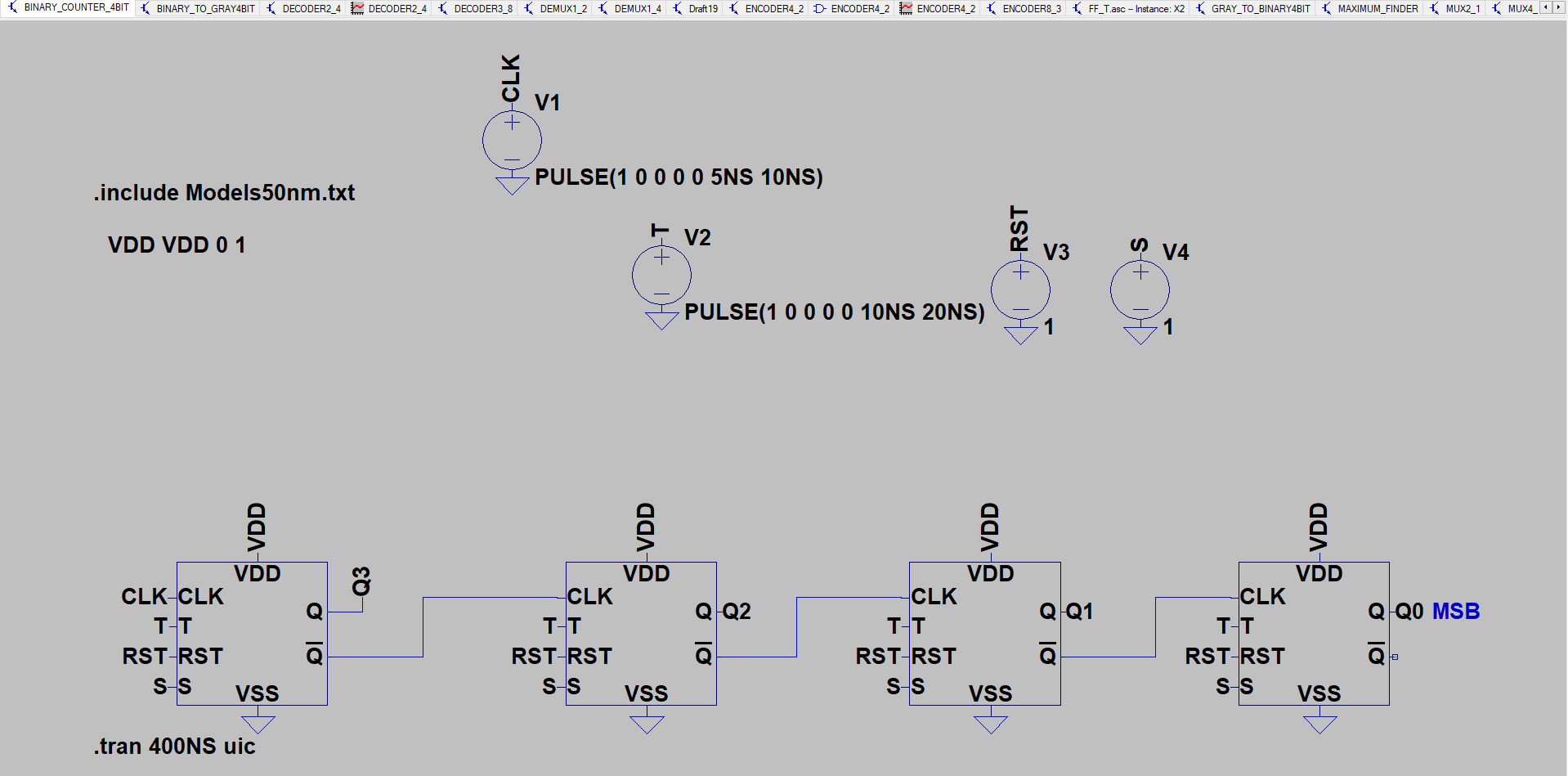


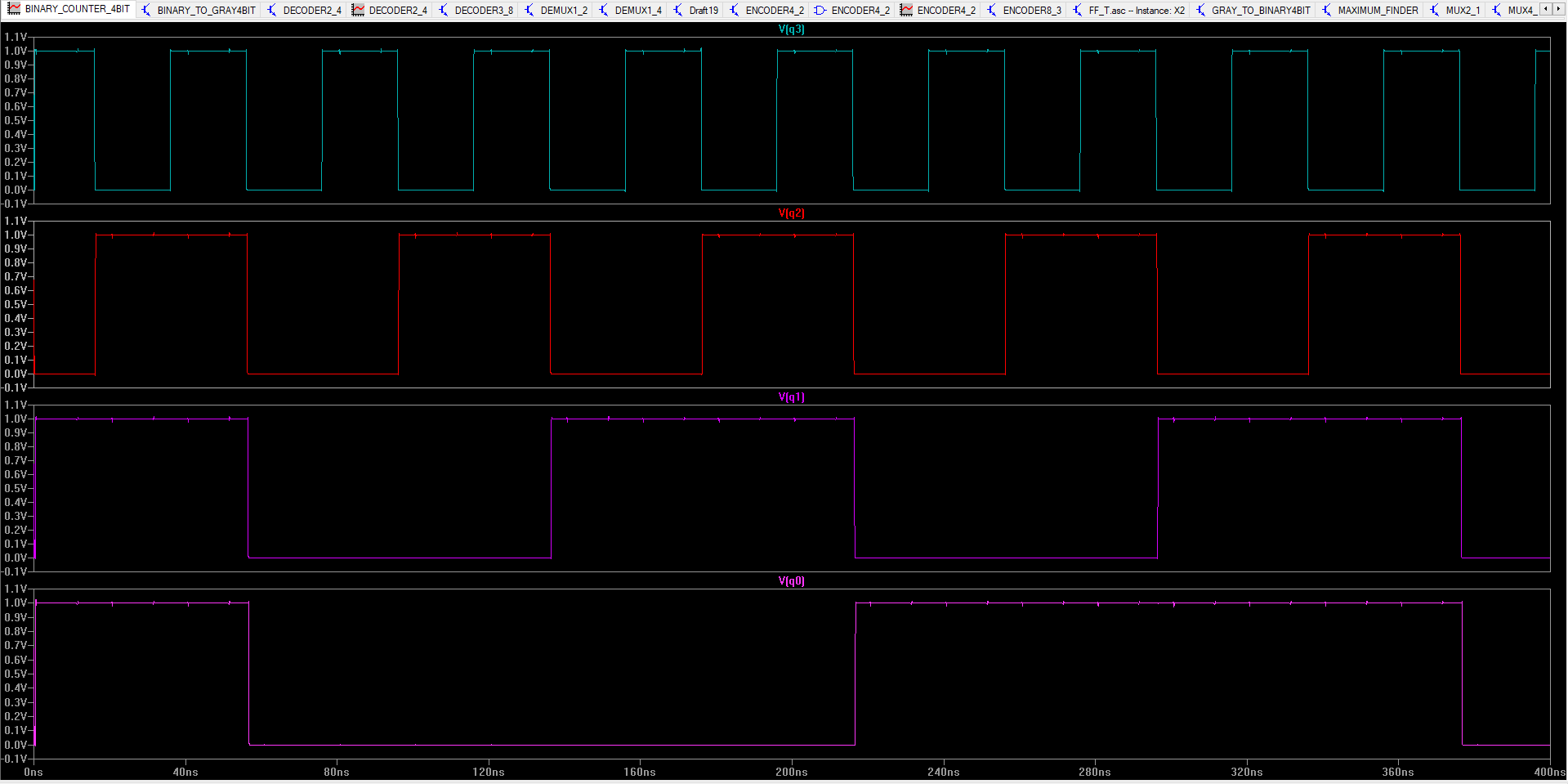
## BUỔI 6

**1. Bộ đếm nhị phân (Binary Counter)**

**Mô tả: Thiết kế một bộ đếm nhị phân 4 bit sử dụng flip-flop. Bộ đếm sẽ tăng giá trị từ 0 đến 15 và lặp lại.**

**Ứng dụng: Dùng trong các hệ thống đo thời gian hoặc đếm sự kiện.**

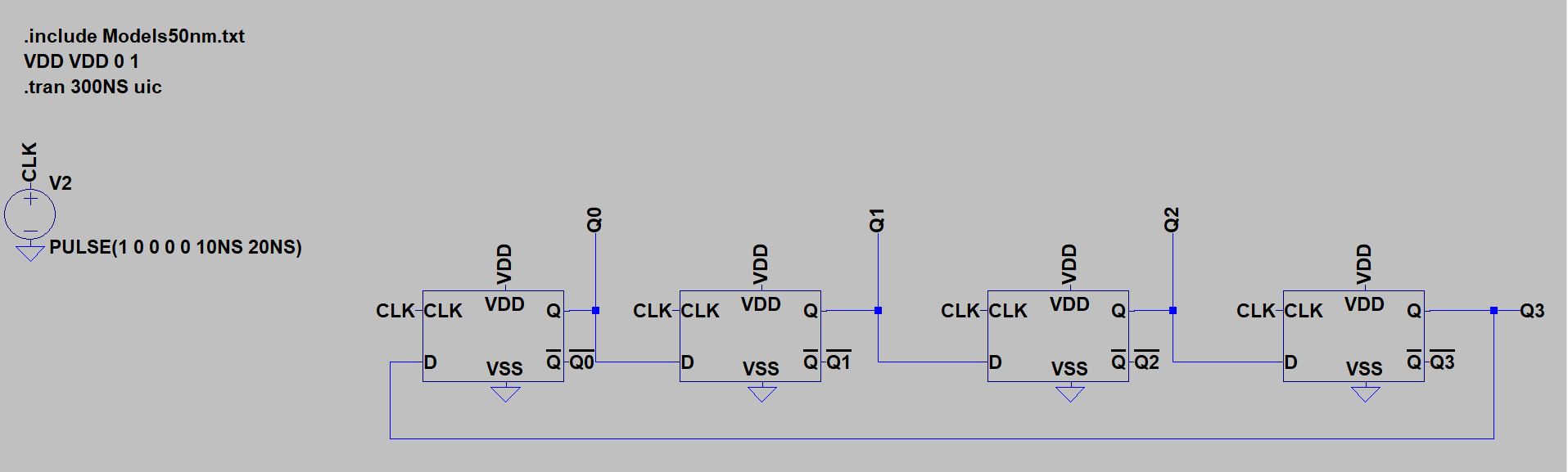




**2. Bộ đếm vòng (Ring Counter)**

**Mô tả: Thiết kế một bộ đếm vòng 4 trạng thái sử dụng flip-flop D. Các trạng thái sẽ luân phiên lặp lại theo chu kỳ.**

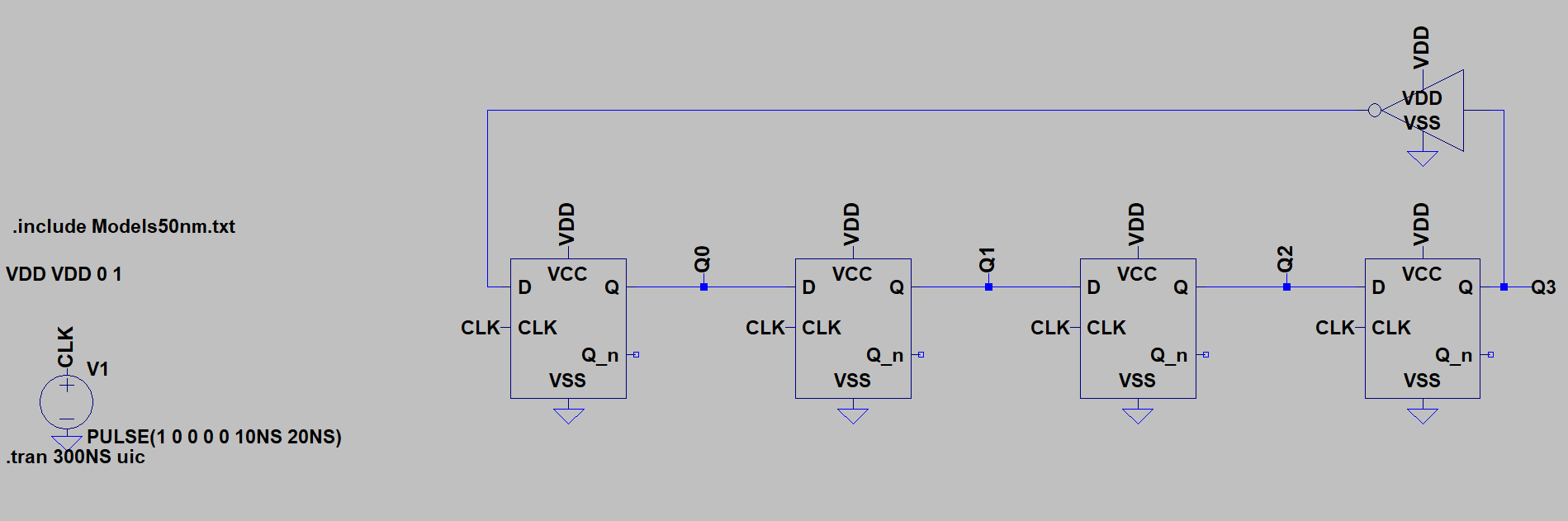
**Ứng dụng: Dùng trong các hệ thống điều khiển trình tự, ví dụ: điều khiển robot**

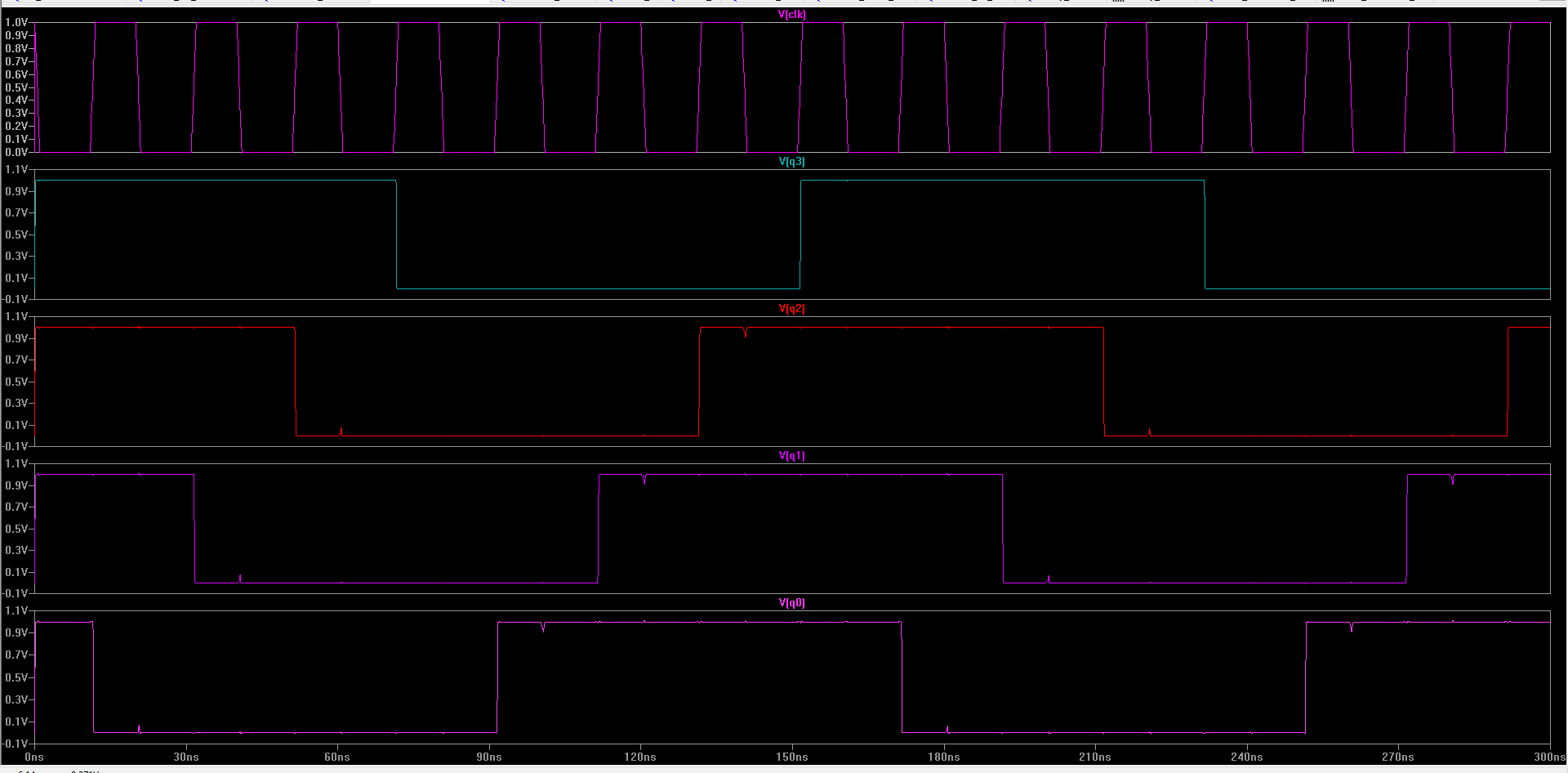


**3. Bộ đếm Johnson (Johnson Counter)**

**Mô tả: Thiết kế một bộ đếm Johnson 4 bit sử dụng flip-flop D. Bộ đếm sẽ tạo ra 8 trạng thái khác nhau và lặp lại.**

**Ứng dụng: Dùng trong các hệ thống tạo tín hiệu điều khiển hoặc mã hóa.**

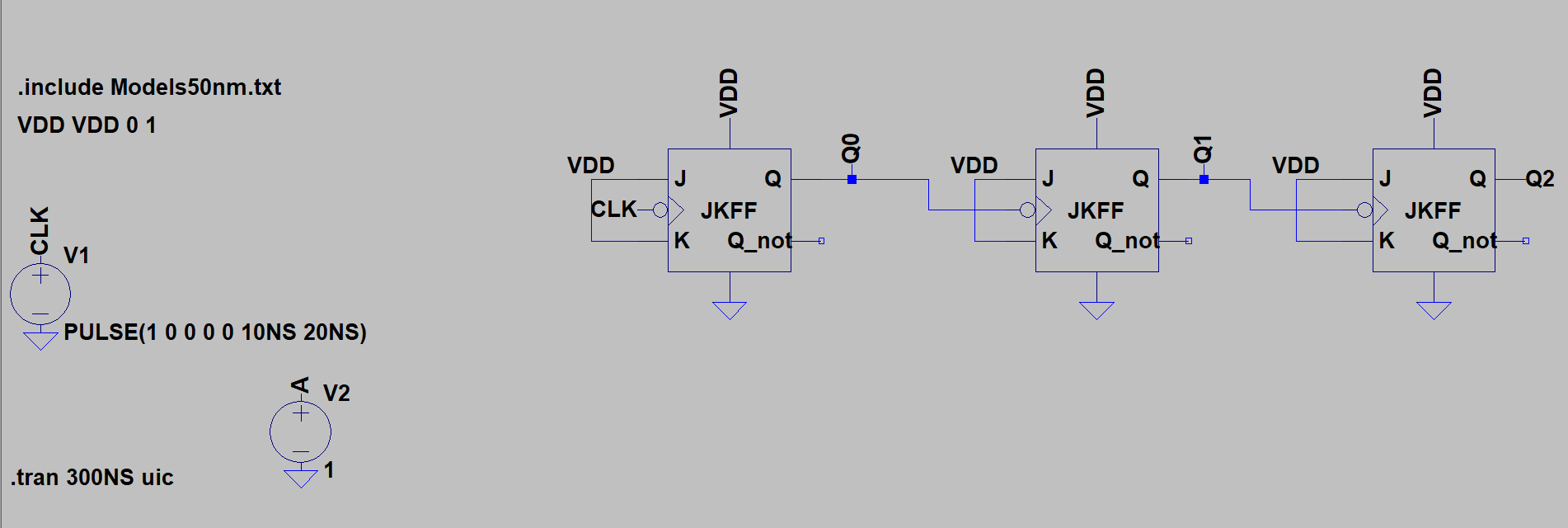


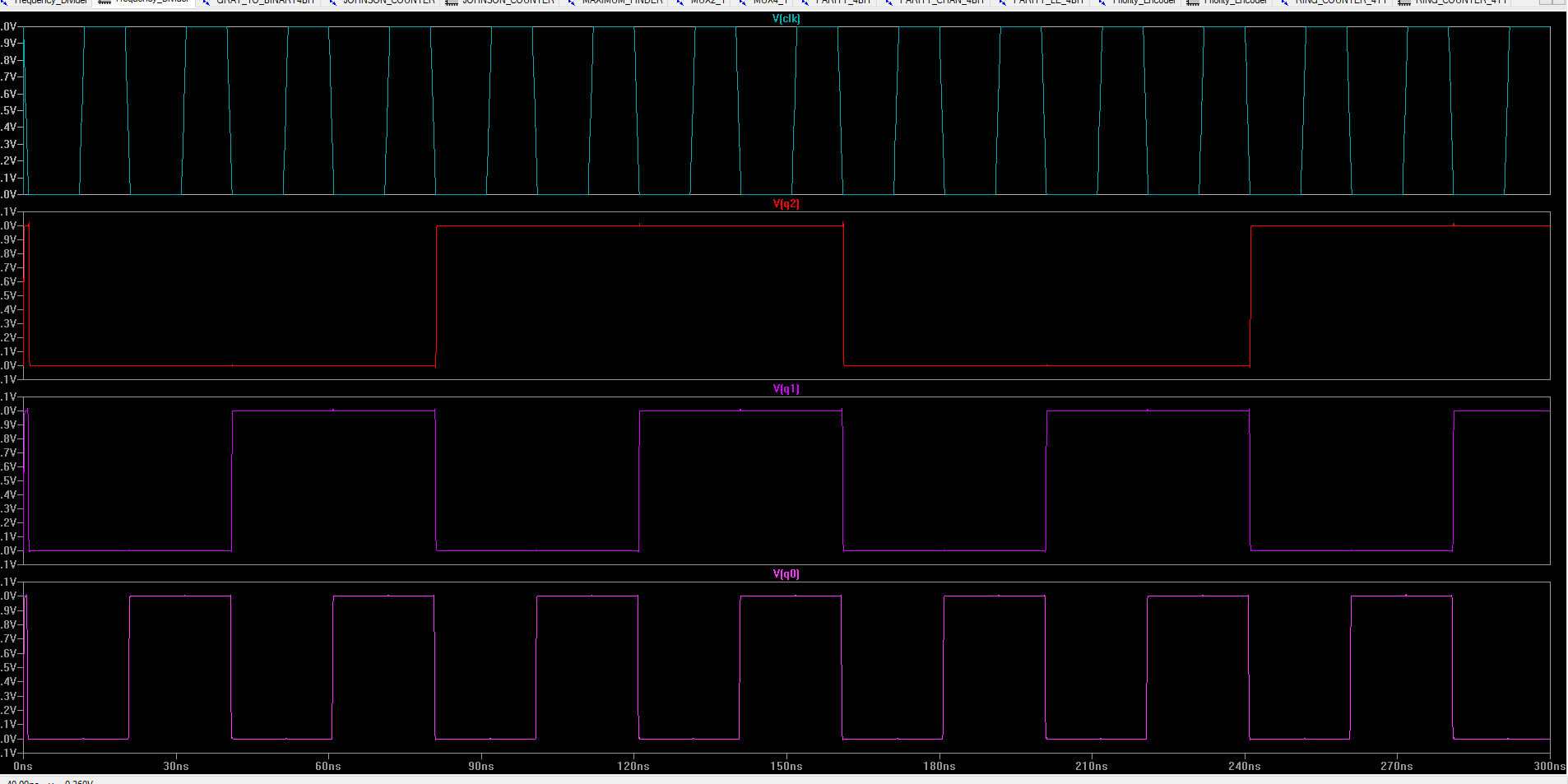


**4. Bộ chia tần số (Frequency Divider)**

**Mô tả: Thiết kế một mạch chia tần số (Frequency Divider) bằng cách sử dụng flip-flop JK. Mạch sẽ chia tần số đầu vào xuống còn 1/2, 1/4, hoặc 1/8 tuỳ thiết kế.**

**Ứng dụng: Dùng trong các hệ thống xử lý tín hiệu hoặc tạo xung clock.**

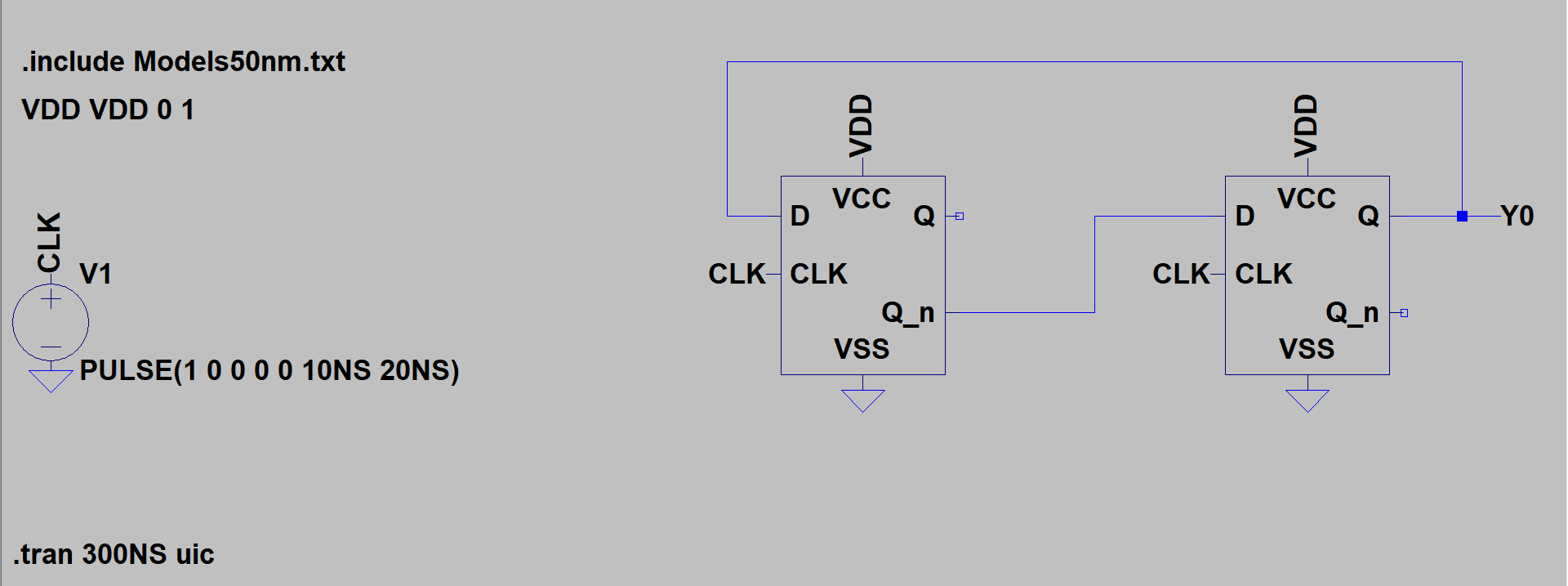


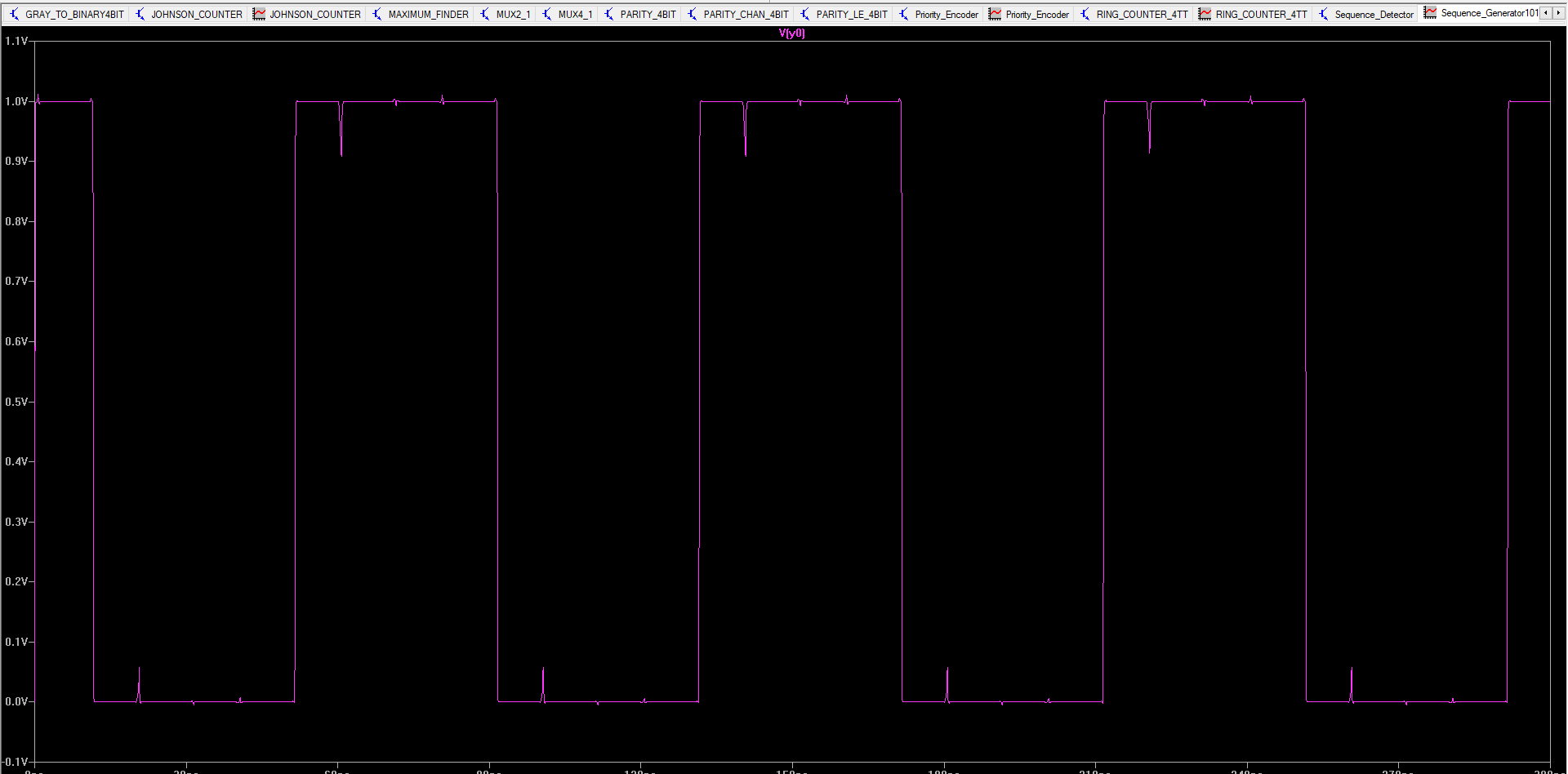


**5. Bộ tạo tín hiệu tuần hoàn (Sequence Generator)**

**Mô tả: Thiết kế một mạch tạo ra chuỗi tín hiệu tuần hoàn 1010 sử dụng flipflop.**

**Ứng dụng: Dùng trong các hệ thống truyền tín hiệu hoặc mã hóa dữ liệu.**

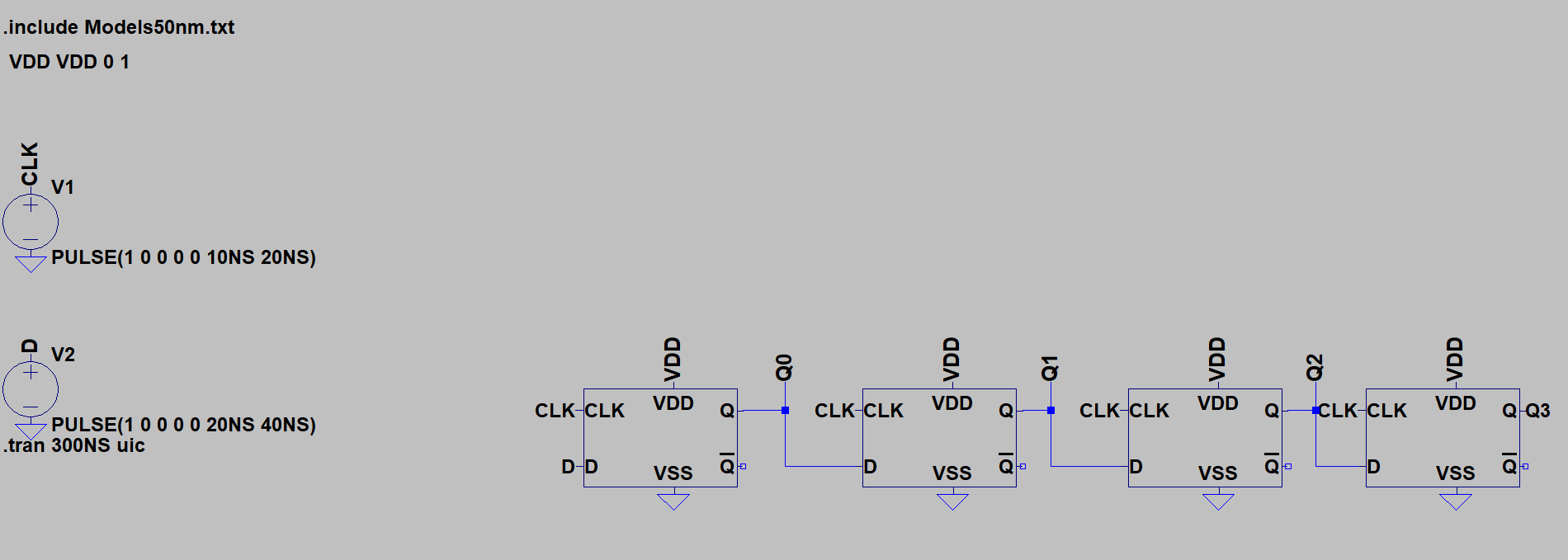




**6. Bộ ghi dịch (Shift Register)**

**Mô tả: Thiết kế một bộ ghi dịch 4 bit (Shift Register) sử dụng flip-flop D. Mạch cho phép dịch trái hoặc dịch phải dữ liệu nhị phân.**

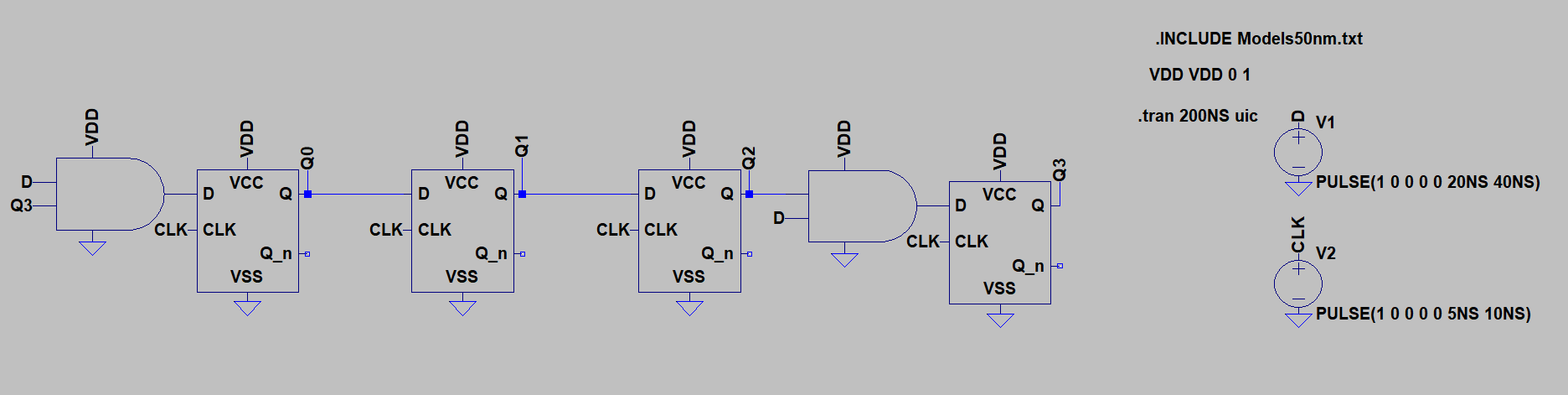
**Ứng dụng: Dùng trong các hệ thống truyền dữ liệu nối tiếp hoặc xử lý tín hiệu.**

****

**7. Bộ ghi dịch xoay vòng (Rotating Shift Register)**

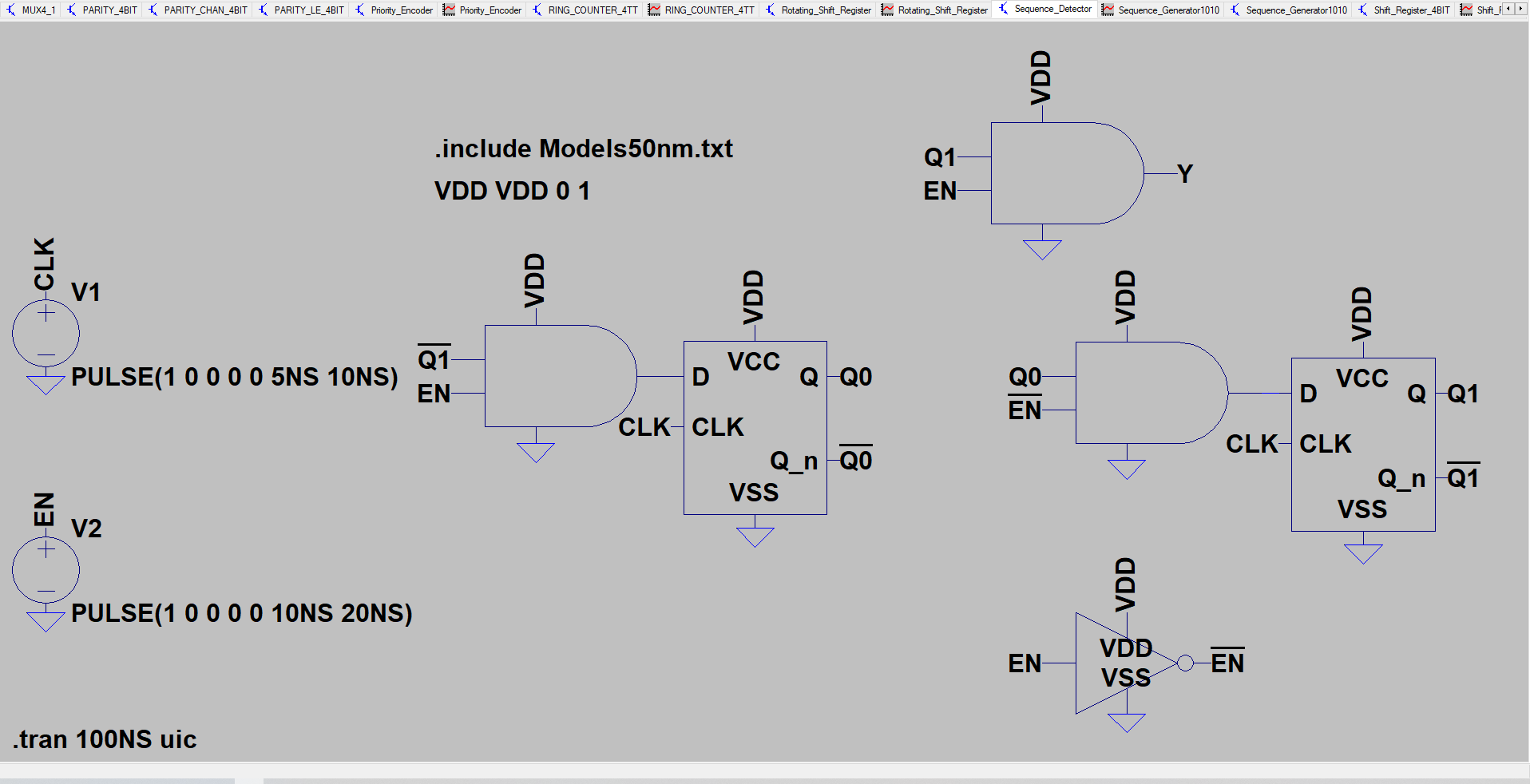
**Mô tả: Thiết kế một bộ ghi dịch xoay vòng 4 bit. Dữ liệu sẽ được dịch vòng liên tục (dịch trái hoặc phải).**

**Ứng dụng: Dùng trong các hệ thống tạo mẫu tín hiệu hoặc điều khiển**

****

**8. Bộ phát hiện chuỗi (Sequence Detector)**

**Mô tả: Thiết kế một mạch phát hiện chuỗi nhị phân "101" trong một luồng dữ liệu đầu vào. Khi chuỗi "101" xuất hiện, mạch sẽ kích hoạt tín hiệu đầu ra. Ứng dụng: Dùng trong các hệ thống nhận dạng tín hiệu hoặc kiểm tra dữ liệu.**

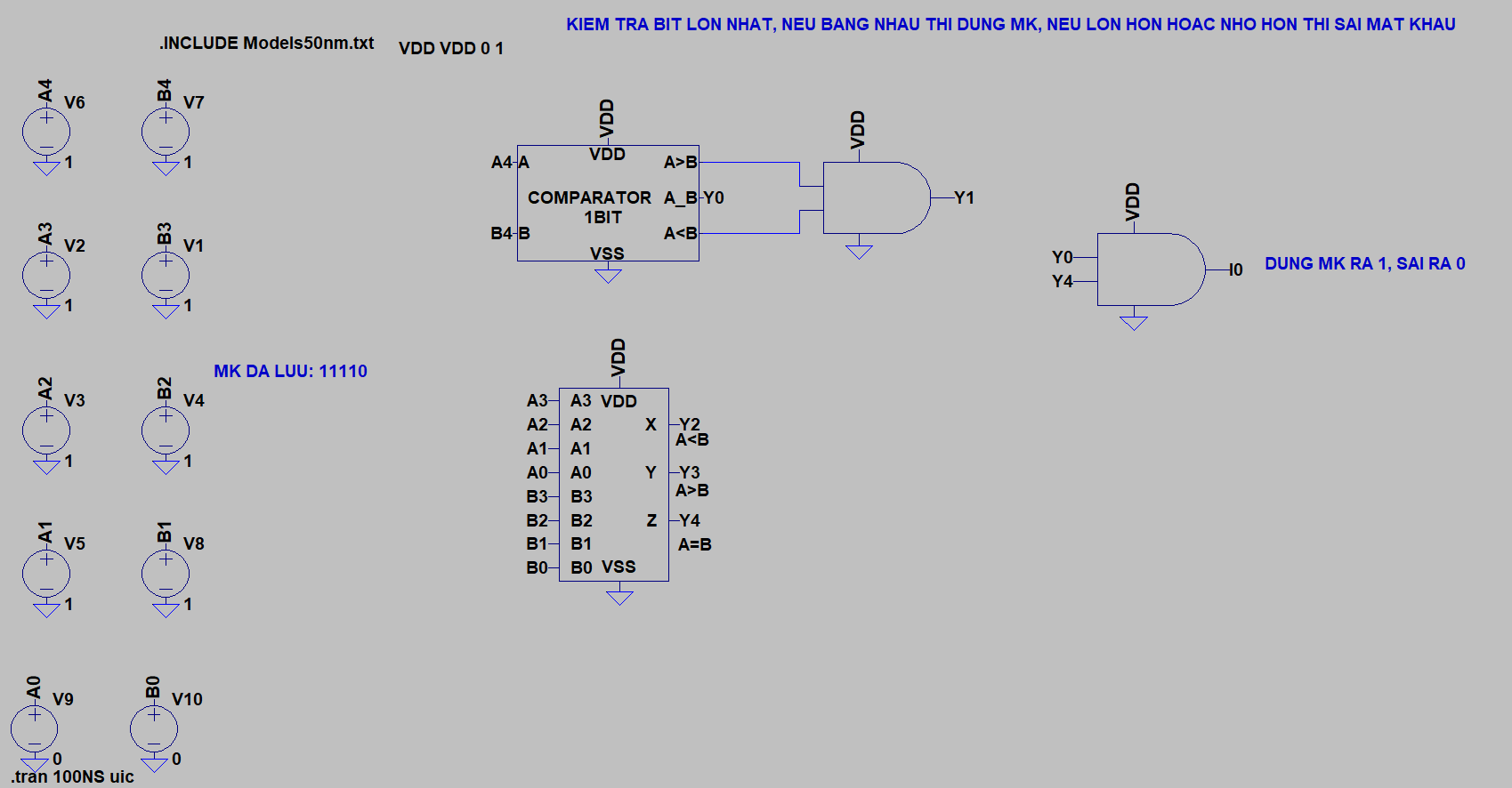
****

## BUỔI 9

**1. Kiểm tra mật mã nhị phân 5 bit (Password Checker)**

• Mạch so sánh để đối chiếu mật mã nhập vào với mật mã lưu trữ.

• Mạch phát hiện 1 hoặc 0 để xác định mật mã đúng hay sai

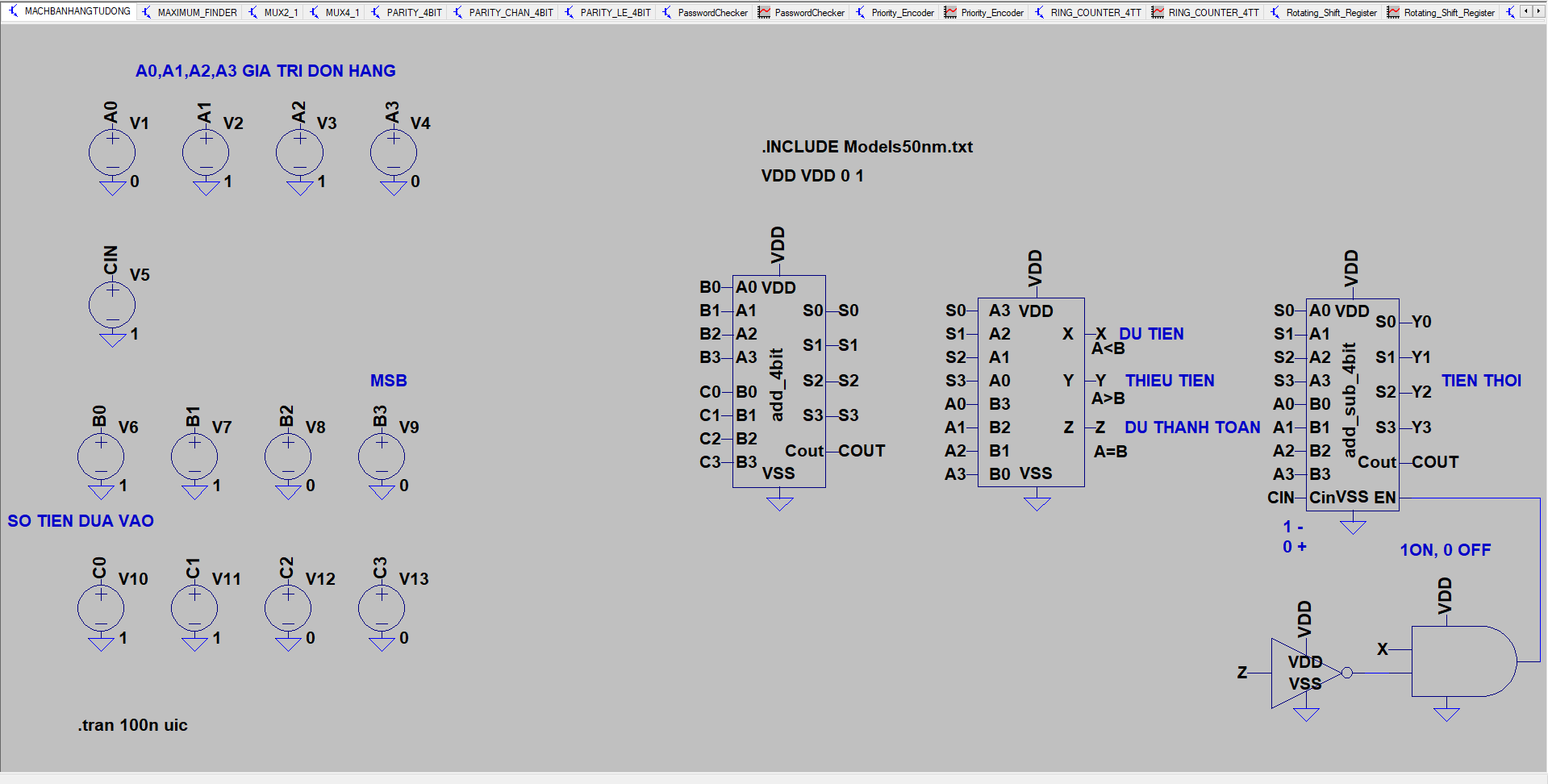


**2. Máy bán hàng tự động (Vending Machine)**

• Mạch cộng để tính tổng tiền đã nhận.

• Mạch so sánh để kiểm tra xem số tiền đã đủ để mua hàng hay chưa.

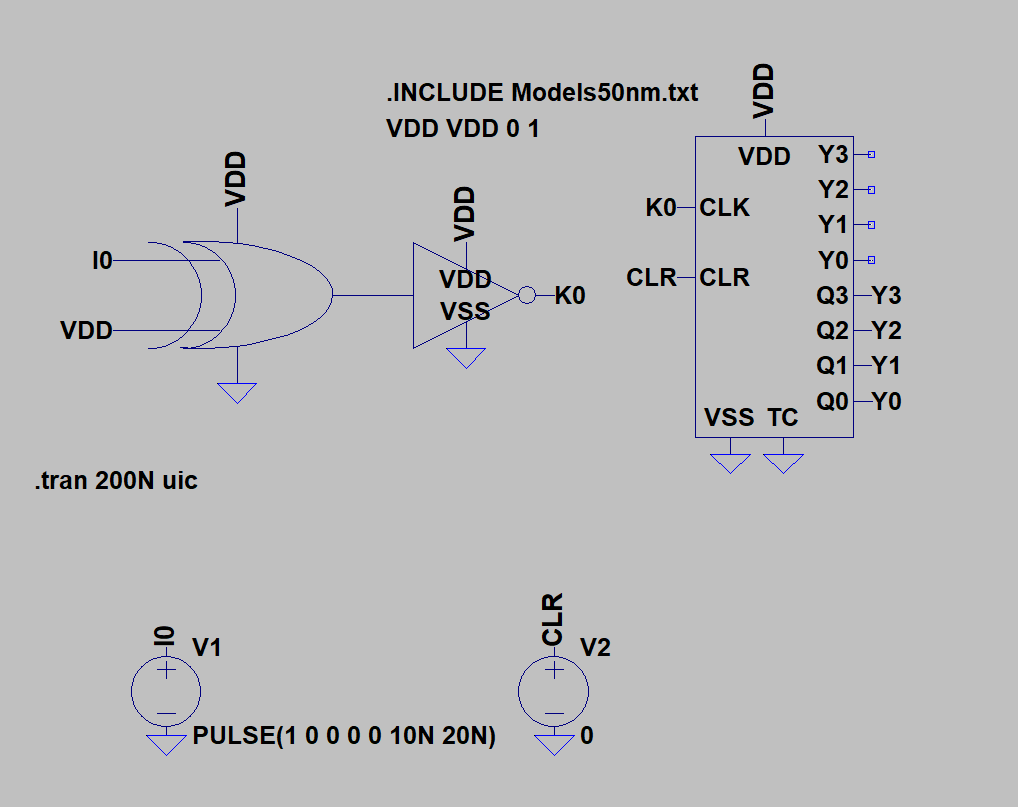
• Mạch trừ để tính số tiền cần trả lại.

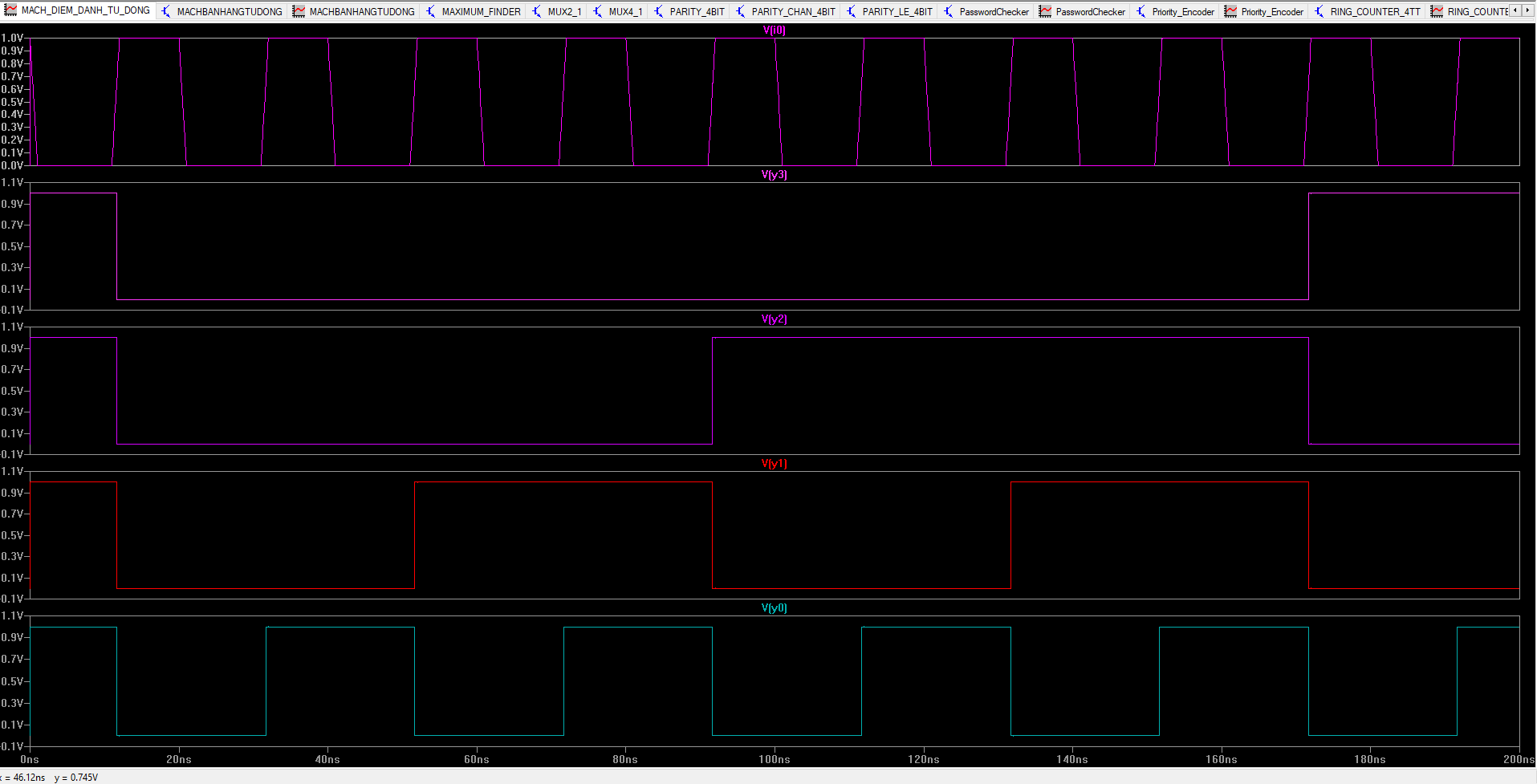


**3. Điểm danh tự động**

• Mạch phát hiện 1 hoặc 0 để kiểm tra trạng thái có mặt/vắng mặt.

• Mạch cộng để tăng số lượng học sinh có mặt.

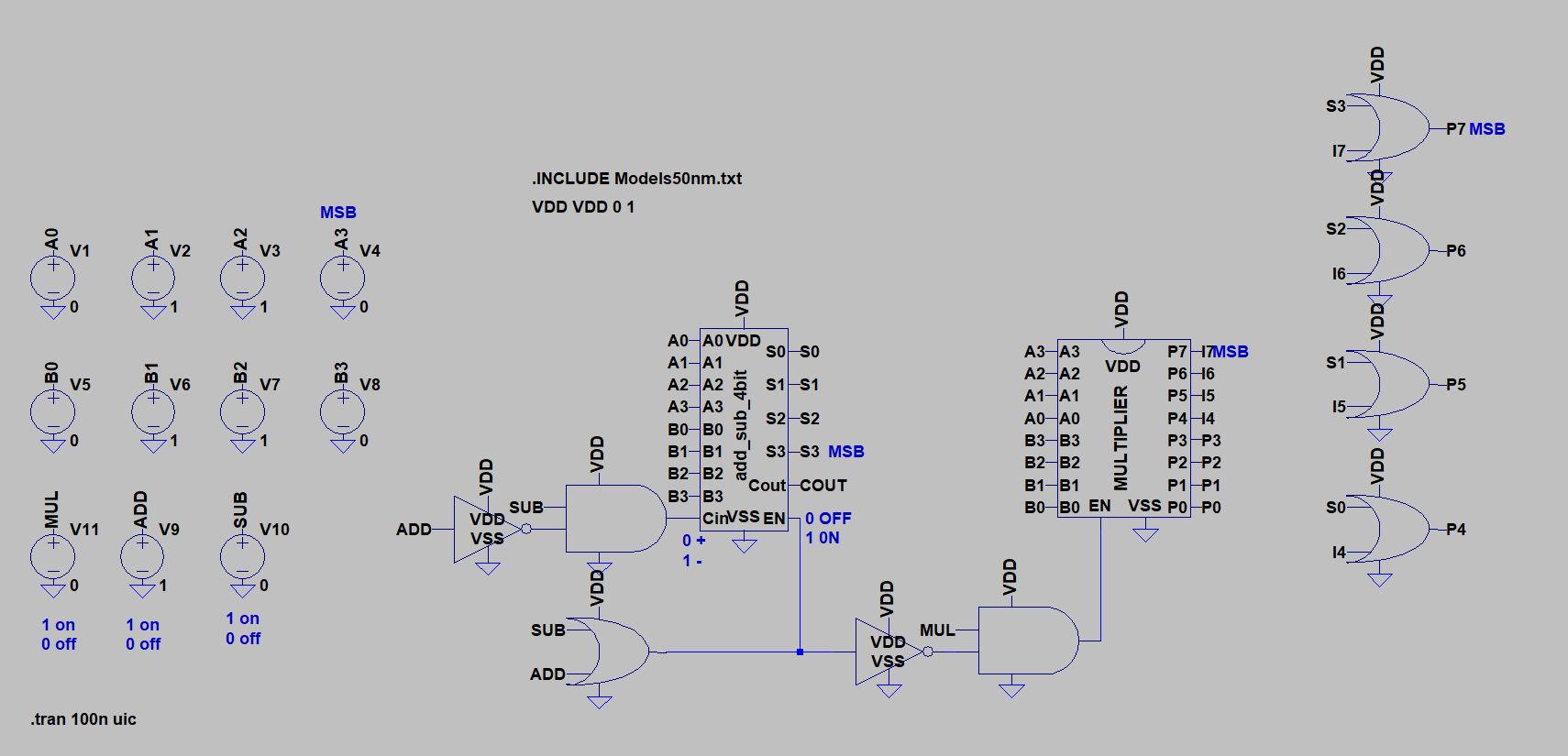




**4. Máy tính đơn giản 4 BIT (Simple Calculator)**

• Mạch cộng và trừ để thực hiện các phép tính số học.

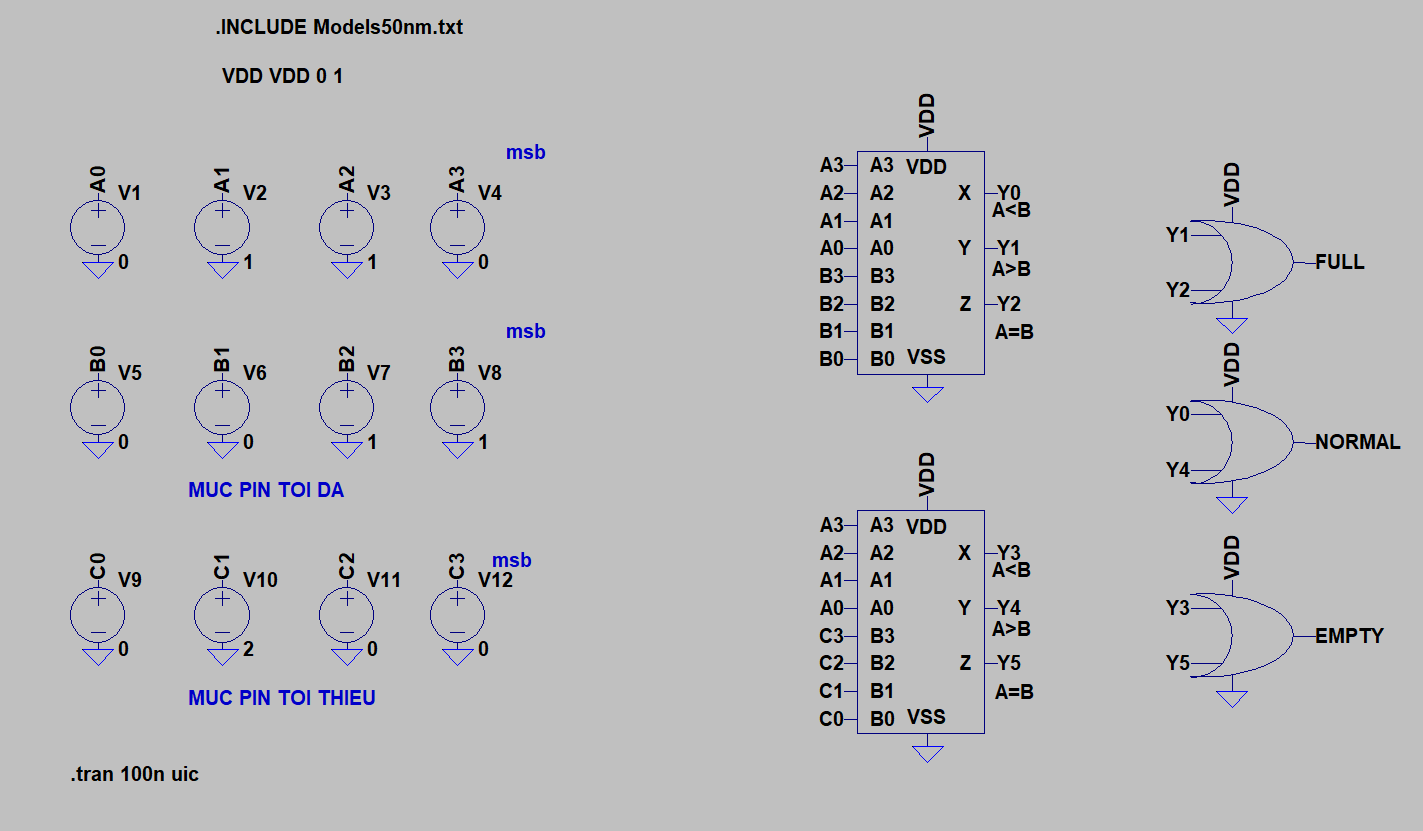
• Mạch nhân để thực hiện phép nhân cơ bản.

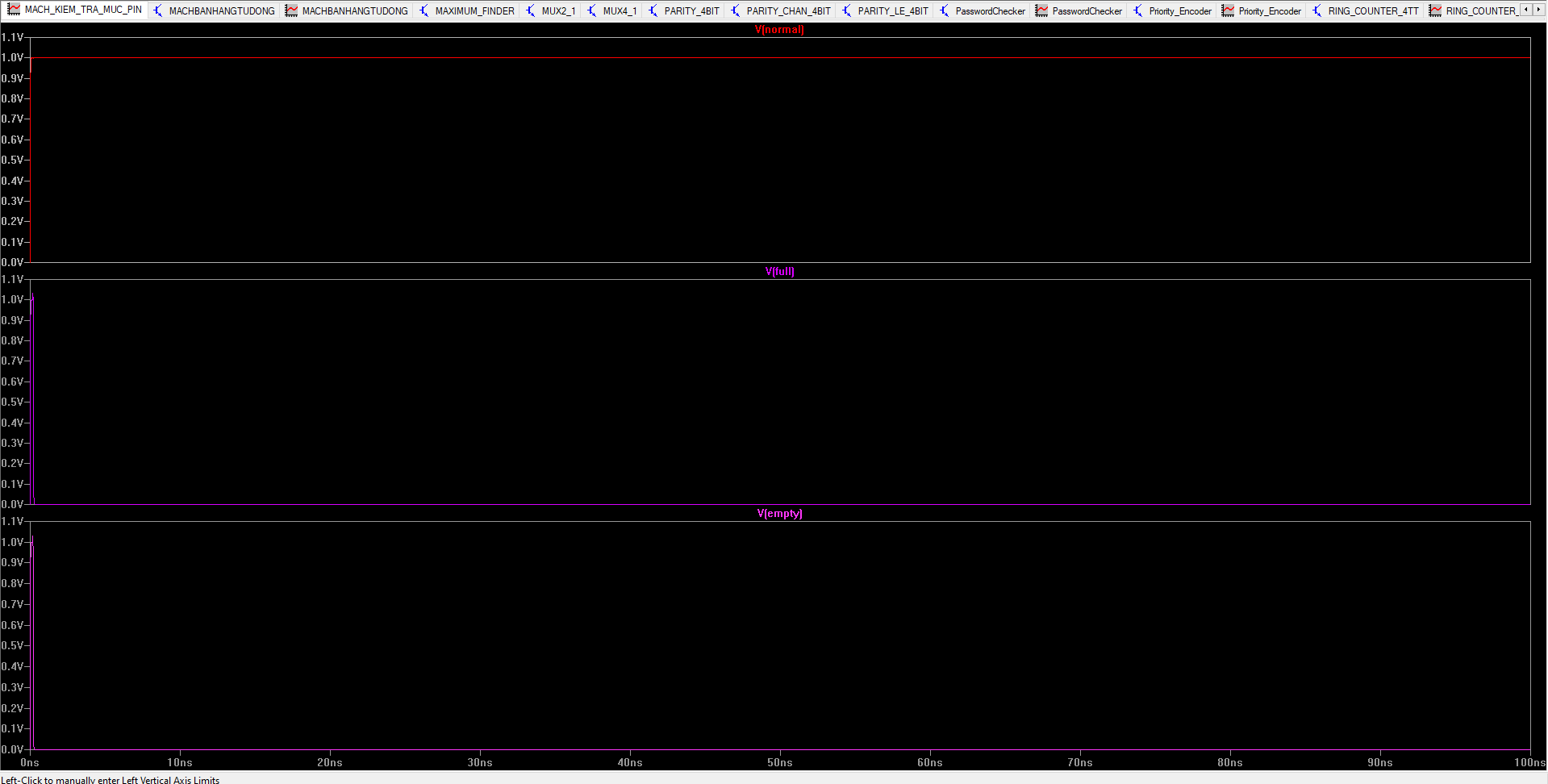


**6. Mạch kiểm tra mức pin**

• Mạch **so sánh** để kiểm tra mức pin hiện tại so với mức tối đa hoặc tối thiểu.

• Mạch **phát hiện 1 hoặc 0** để phát hiện pin yếu.



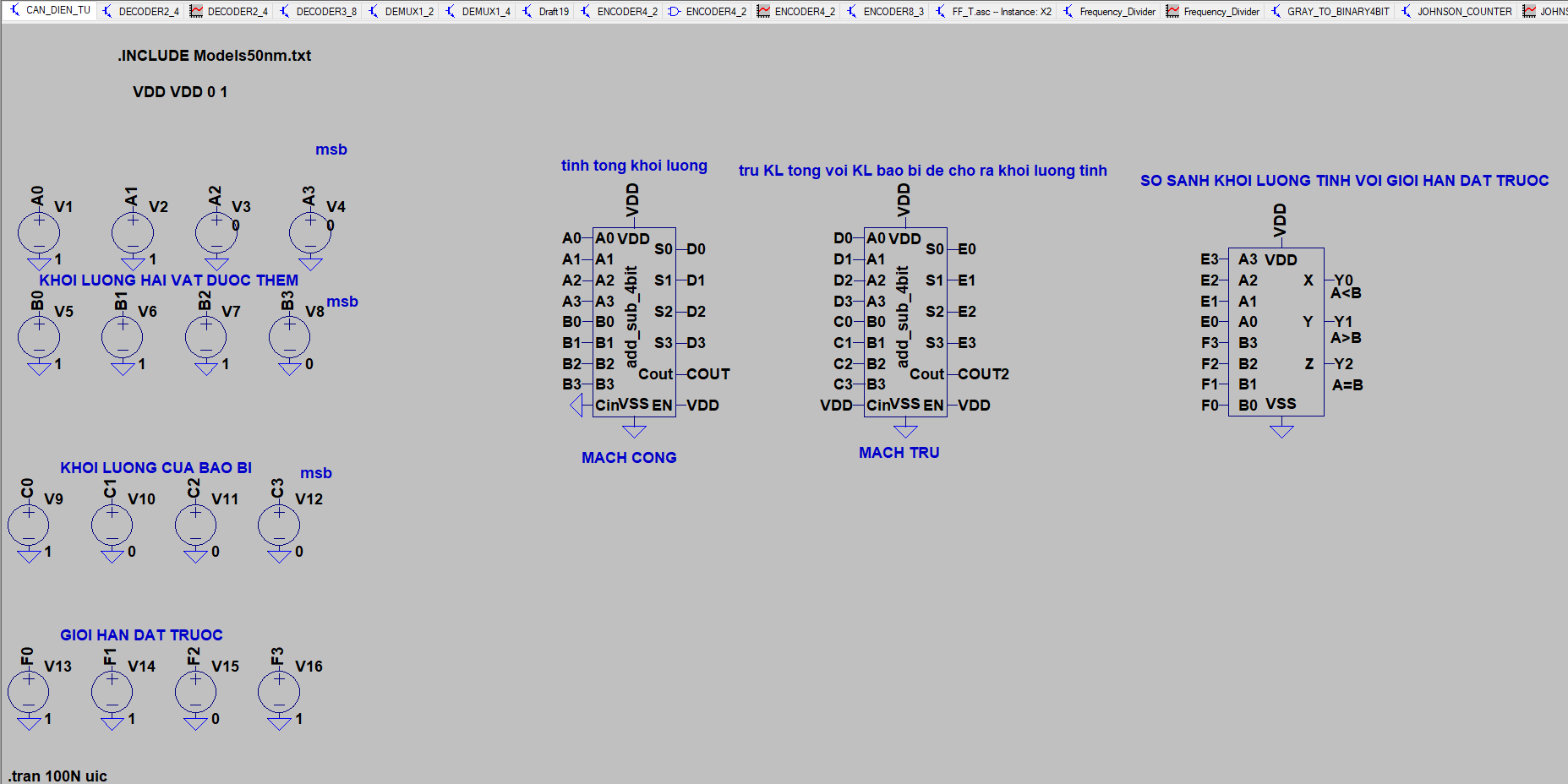


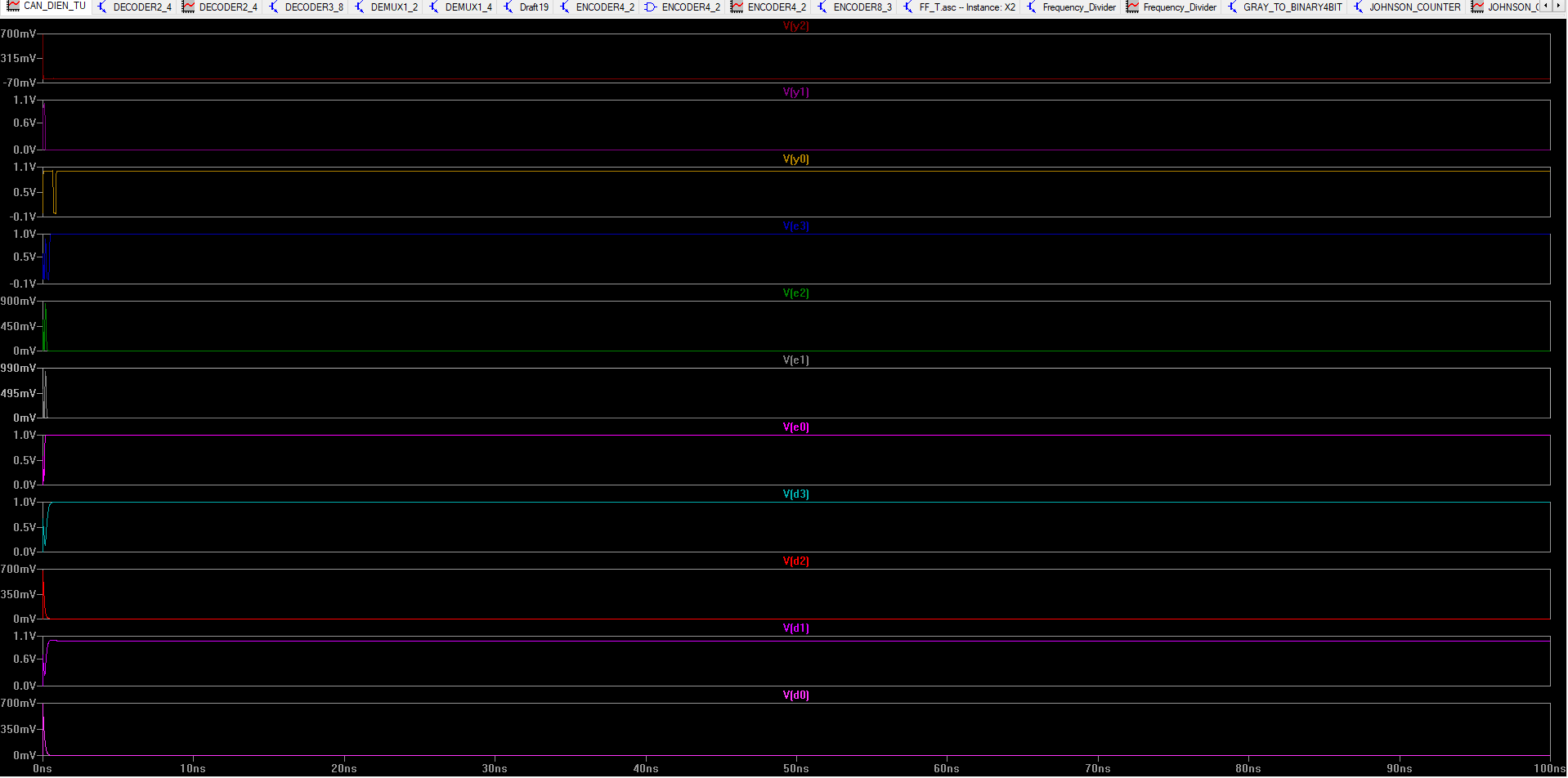
**7. Cân điện tử**

• Mạch **cộng** để tính tổng trọng lượng khi thêm vật.

• Mạch **trừ** để tính khối lượng thực tế khi có bao bì.

• Mạch **so sánh** để kiểm tra trọng lượng với giới hạn đặt trước





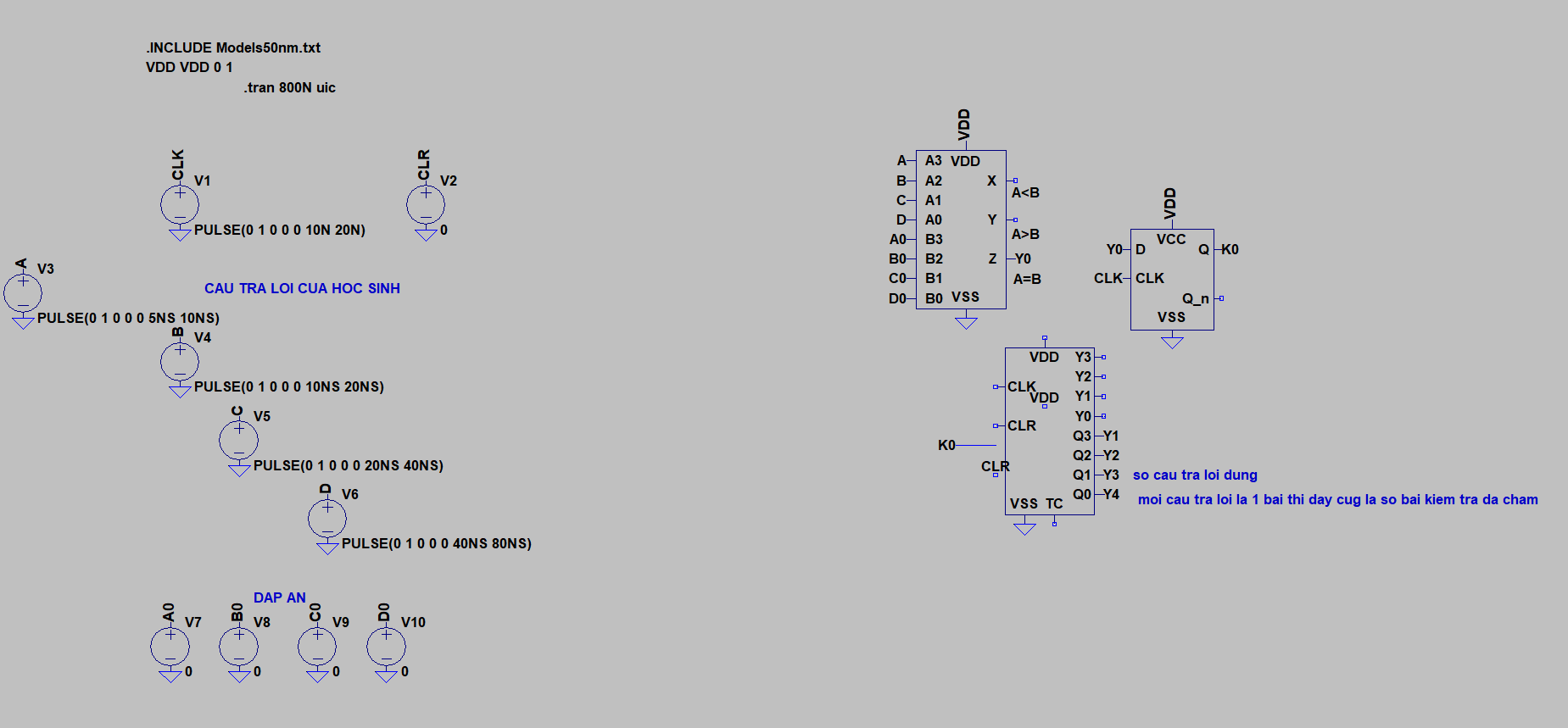
**8. Chấm điểm trắc nghiệm**

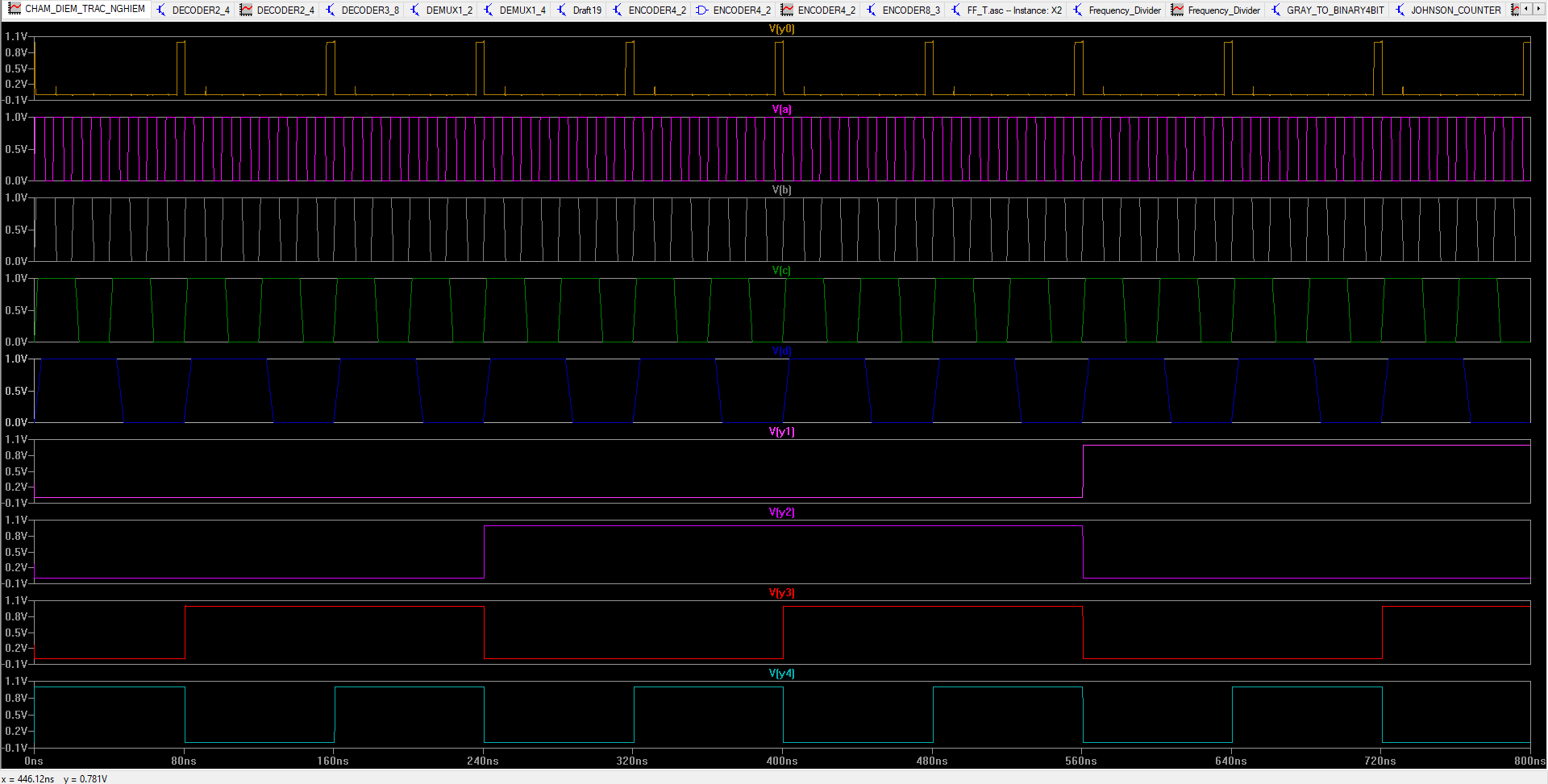
• Câu trả lời đúng là 1, sai là 0

• Mạch **so sánh** để đối chiếu câu trả lời của học sinh với đáp án đúng.

• Mạch **cộng** để tính tổng số câu trả lời đúng.

• Mạch **đếm** để theo dõi tổng số bài kiểm tra đã chấm





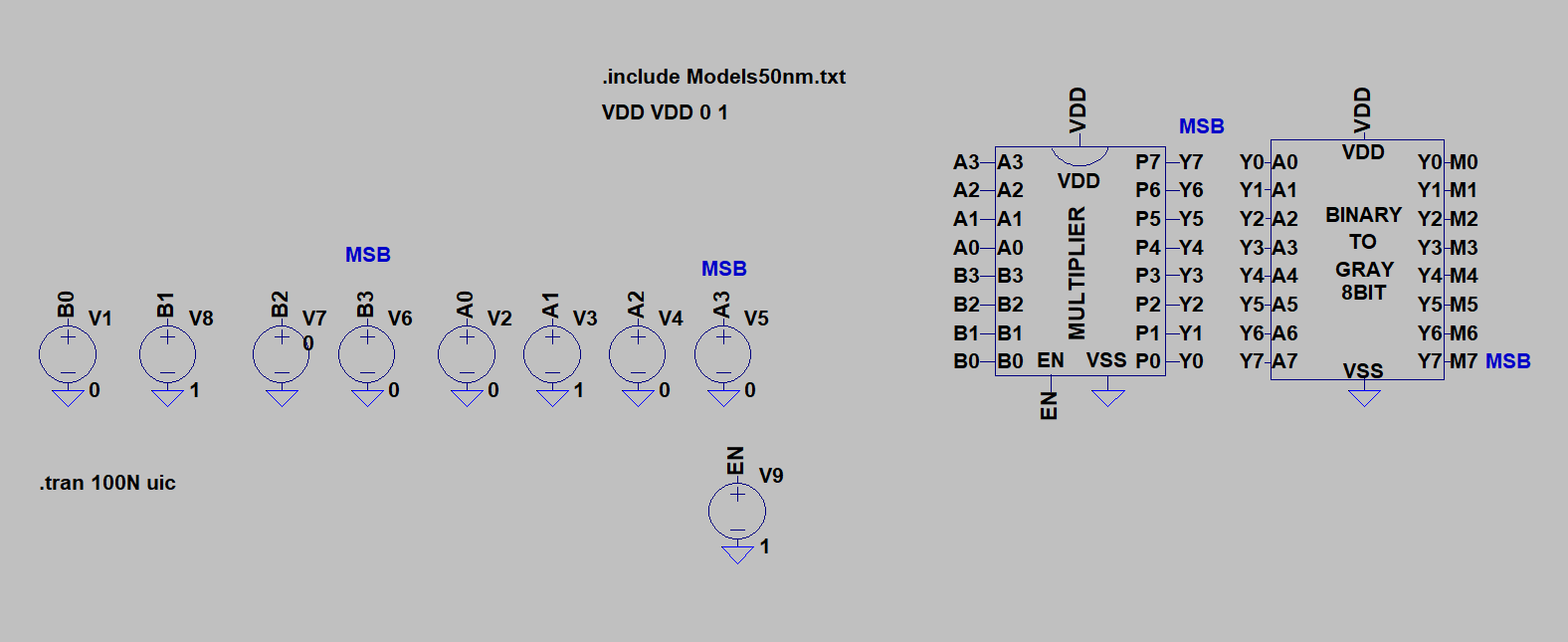
## BUỔI 11

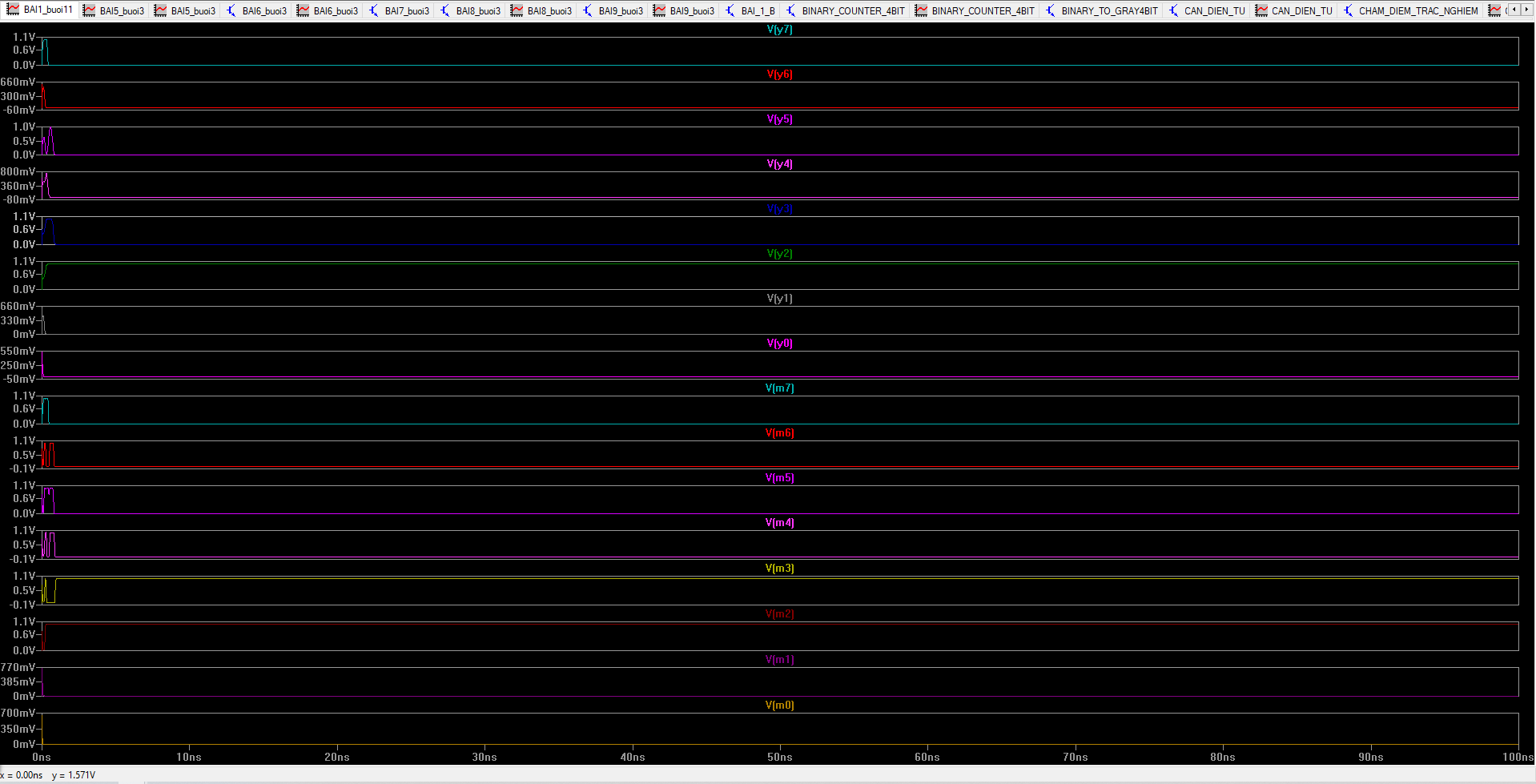
**1. Tính toán và mã hóa**

Mô tả:

Thiết kế mạch vừa tính diện tích hình chữ nhật (mạch nhân), vừa chuyển đổi kết quả từ mã Binary sang Gray để mã hóa dữ liệu.

Đầu vào: Chiều dài và chiều rộng (4-bit mỗi giá trị). Đầu ra: Diện tích (8-bit) và mã Gray của diện tích.



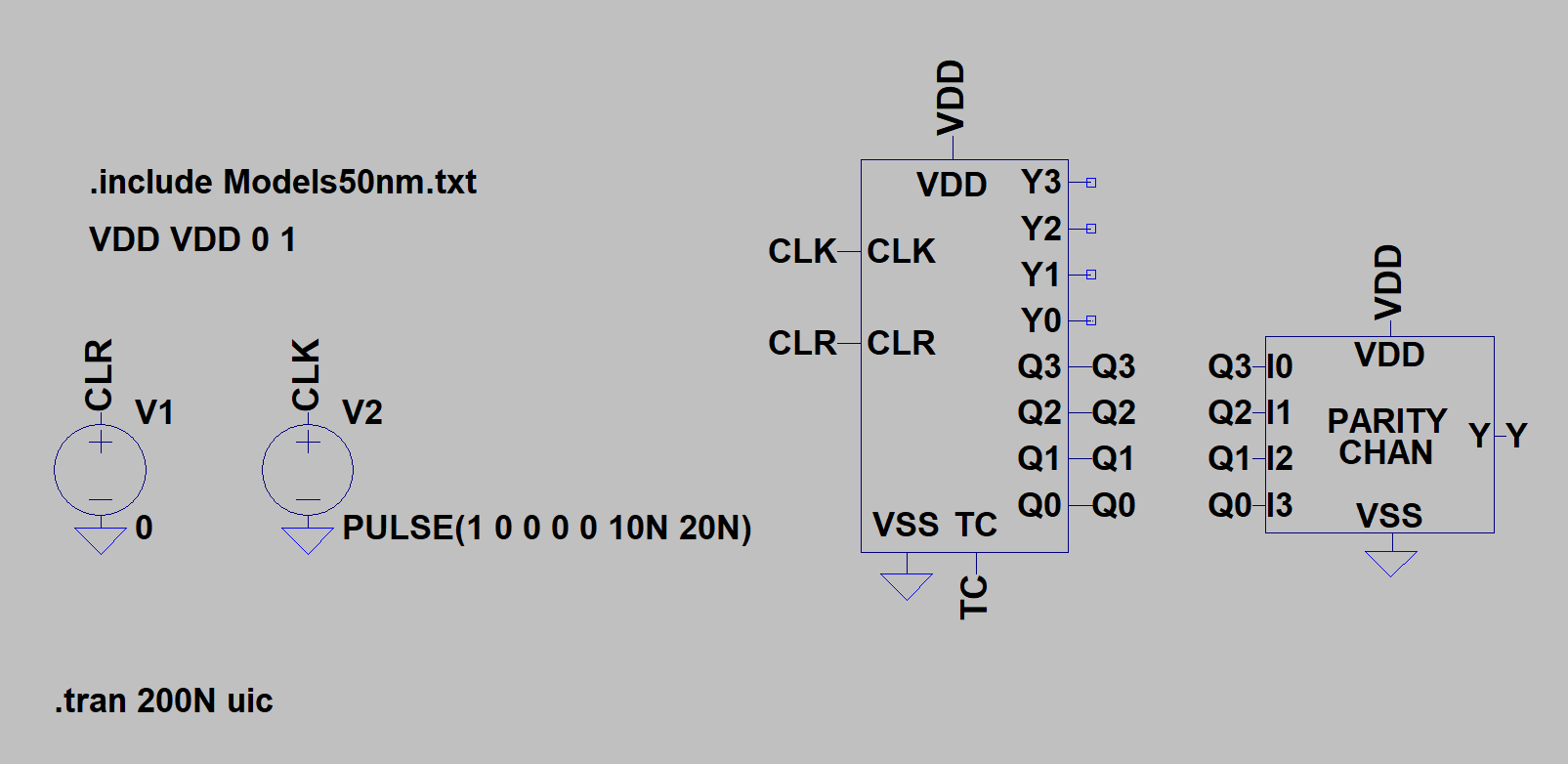


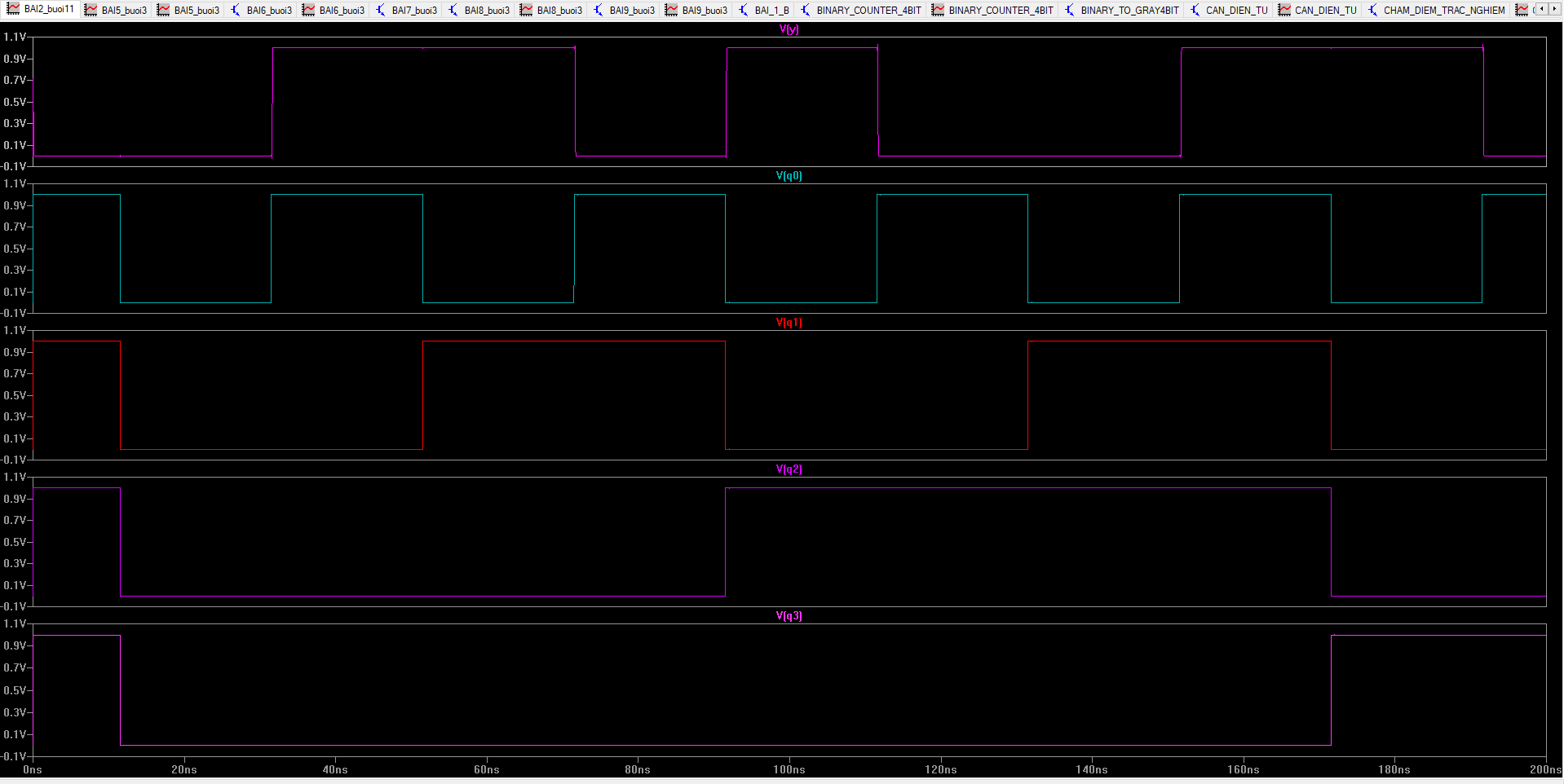
**2. Đếm và kiểm tra lỗi**

**Mô tả:** Thiết kế mạch vừa đếm sự kiện (4-bit), vừa tạo bit parity để kiểm tra lỗi dữ liệu đếm.

• Đầu vào: Tín hiệu xung clock từ cảm biến cửa.

• Đầu ra: Giá trị đếm (4-bit) và bit parity (0: chẵn, 1: lẻ)



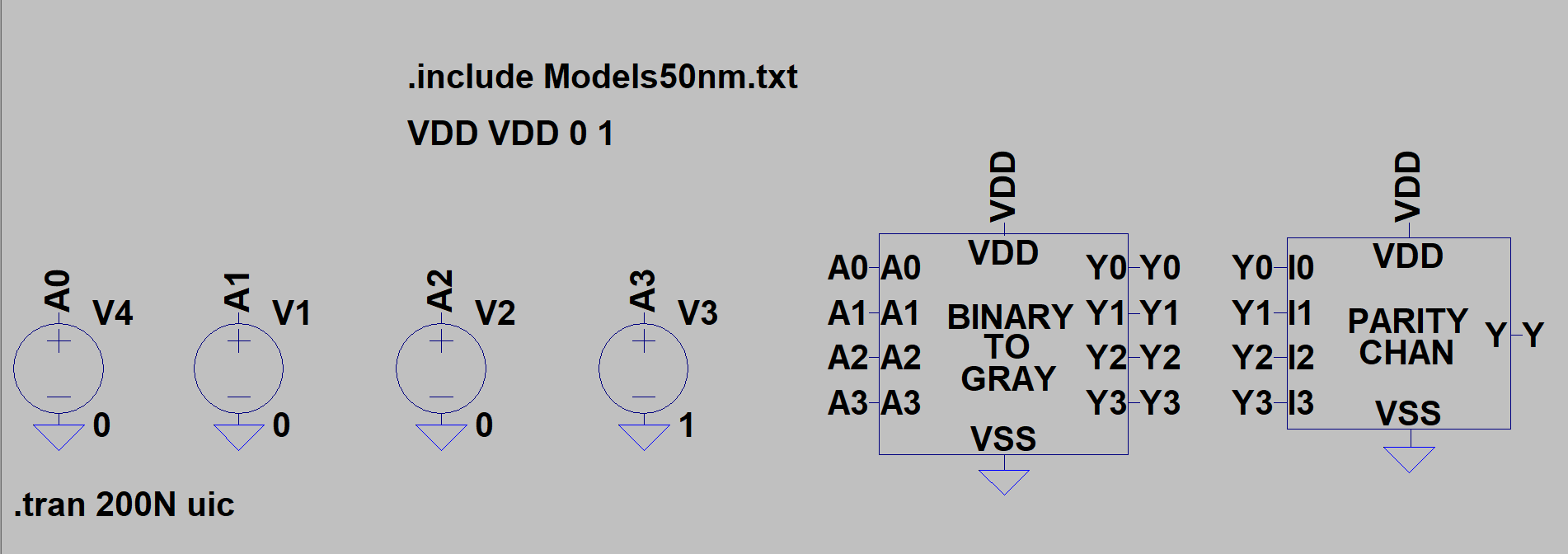


**3. Mã hóa dữ liệu và kiểm tra lỗi**

Mô tả: Thiết kế mạch vừa chuyển đổi mã Binary ↔ Gray (mạch chuyển mã), vừa tạo bit parity để kiểm tra lỗi dữ liệu.

• Đầu vào: Chuỗi dữ liệu 4-bit.

• Đầu ra: Mã Gray 4-bit và bit parity.





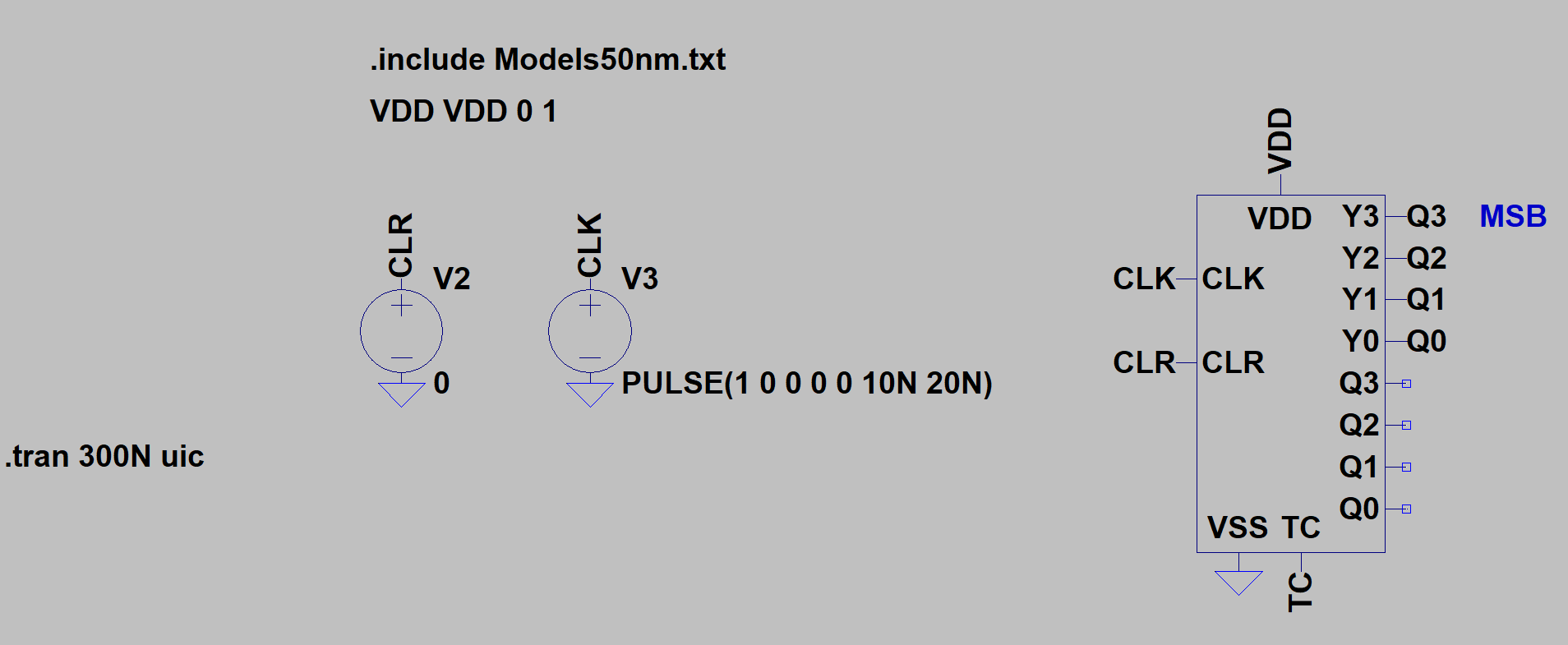
**4. Đếm ngược và phát cảnh báo**

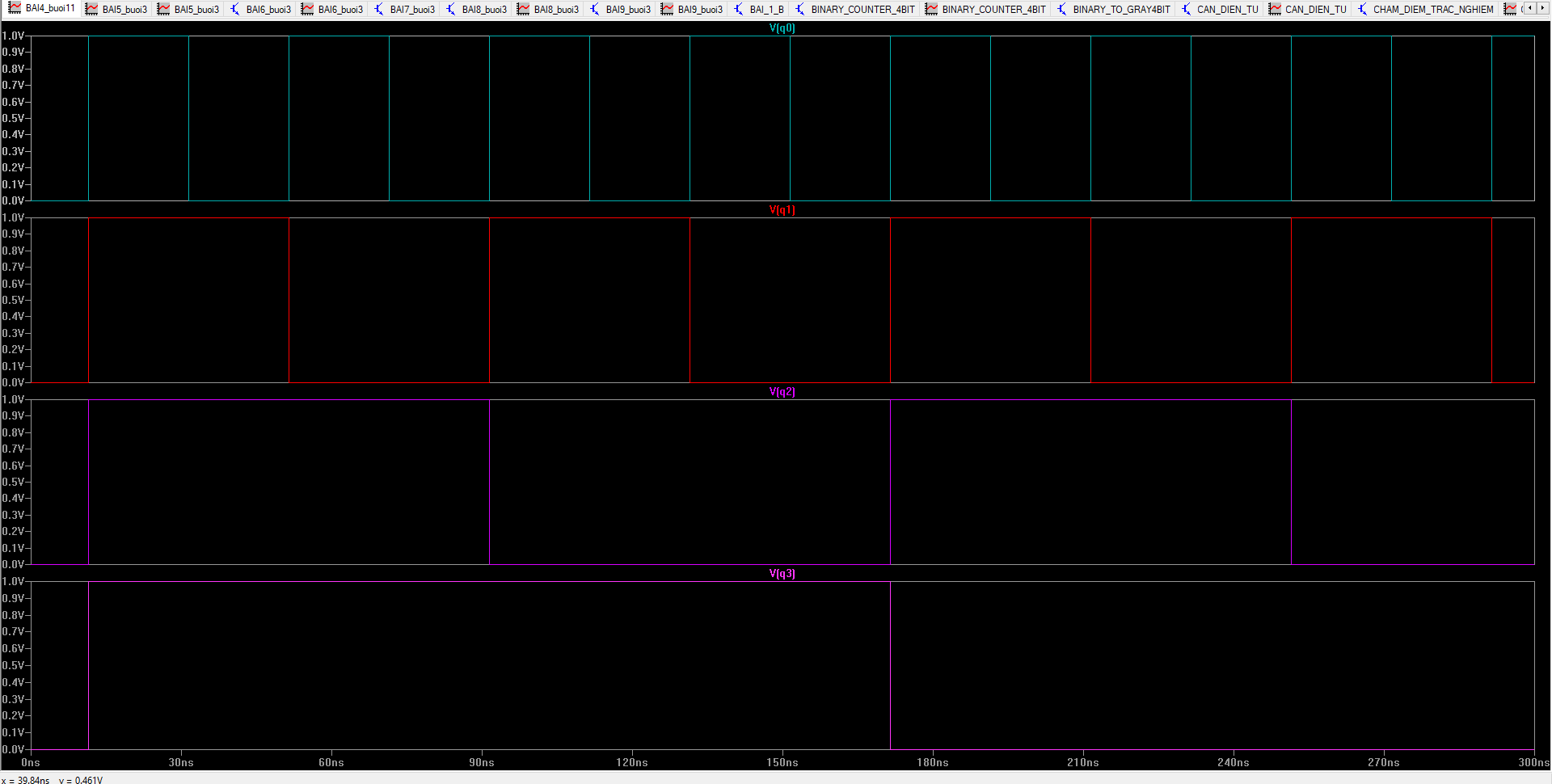
Mô tả:

Thiết kế mạch vừa đếm ngược thời gian (mạch đếm xuống), vừa phát tín hiệu cảnh báo khi thời gian về 0.

Đầu vào: Giá trị thời gian ban đầu (4-bit), tín hiệu xung clock.

Đầu ra: Giá trị thời gian còn lại (4-bit) và tín hiệu cảnh báo (0: chưa về 0, 1: đã về 0)





**5. Kiểm tra và điều chỉnh giá trị cảm biến**

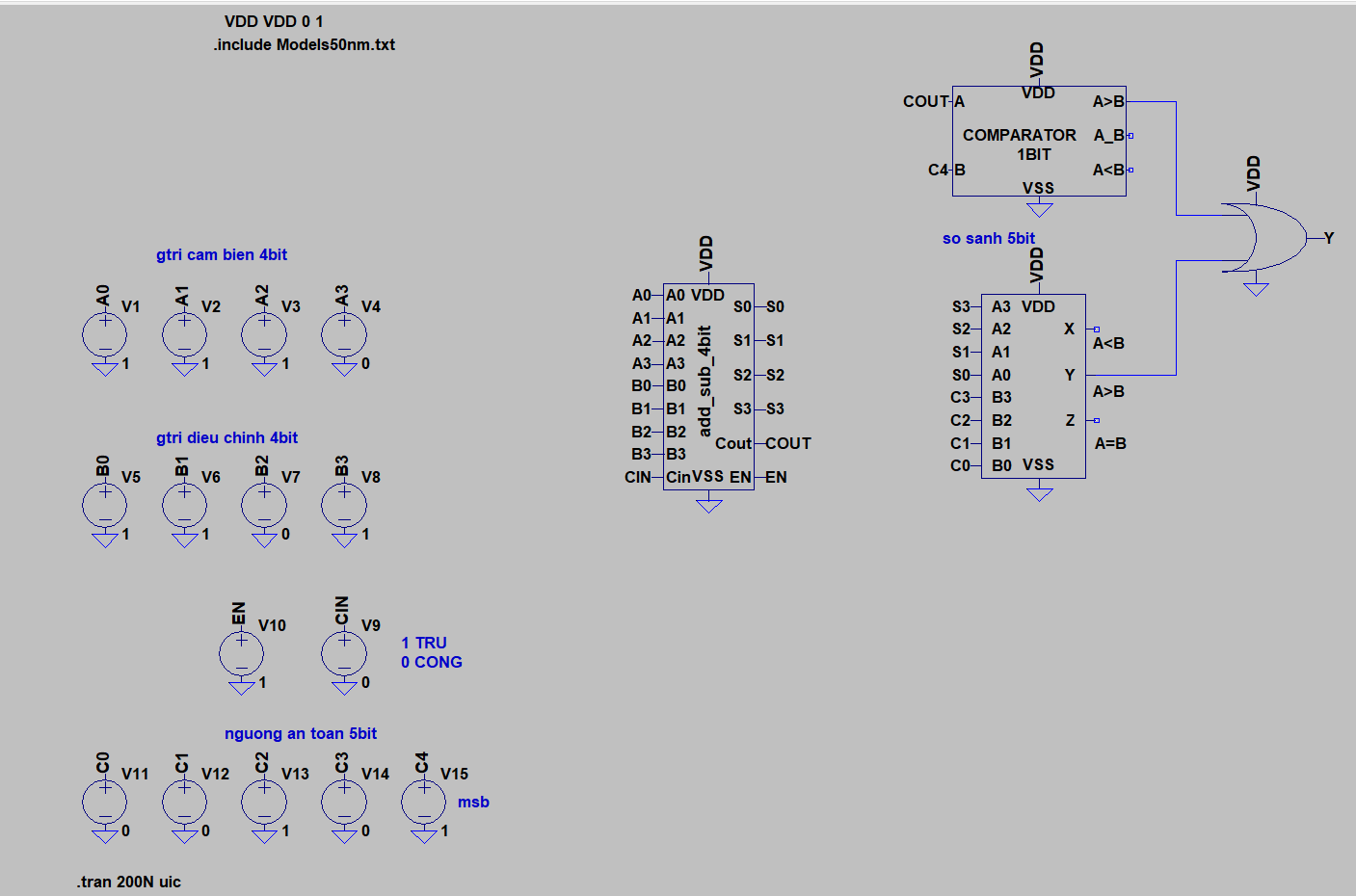
**Mô tả:** Thiết kế mạch điều chỉnh giá trị cảm biến bằng phép cộng/trừ 1 lượng giá trị, vừa kiểm tra giá trị cảm biến vượt ngưỡng (so sánh),

• **Đầu vào:** Giá trị cảm biến (4-bit), giá trị điều chỉnh (4-bit), tín hiệu

chọn chế độ (0: cộng, 1: trừ).

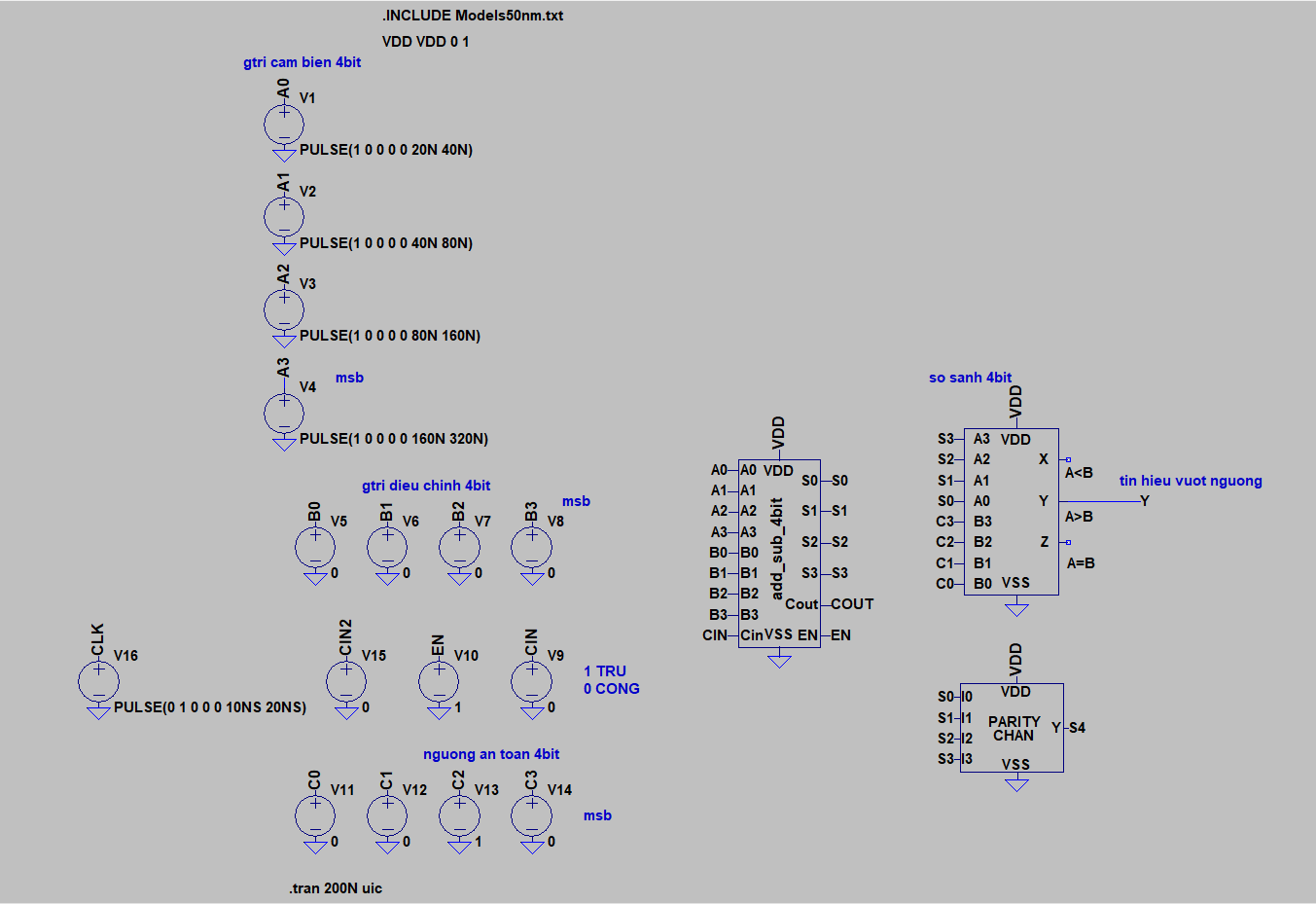
• **Đầu ra:** Giá trị cảm biến mới (5-bit) và tín hiệu kiểm tra trạng thái (0:

bình thường, 1: vượt ngưỡng).

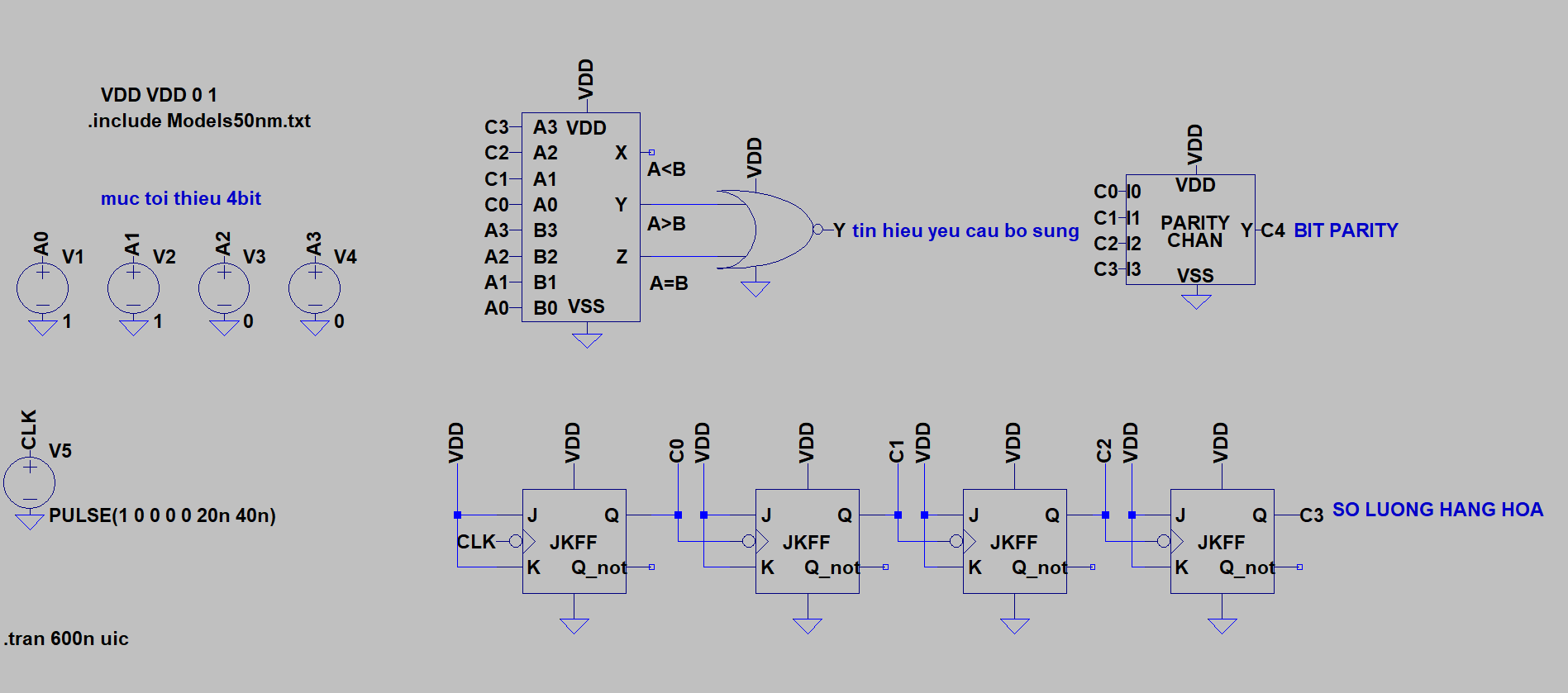


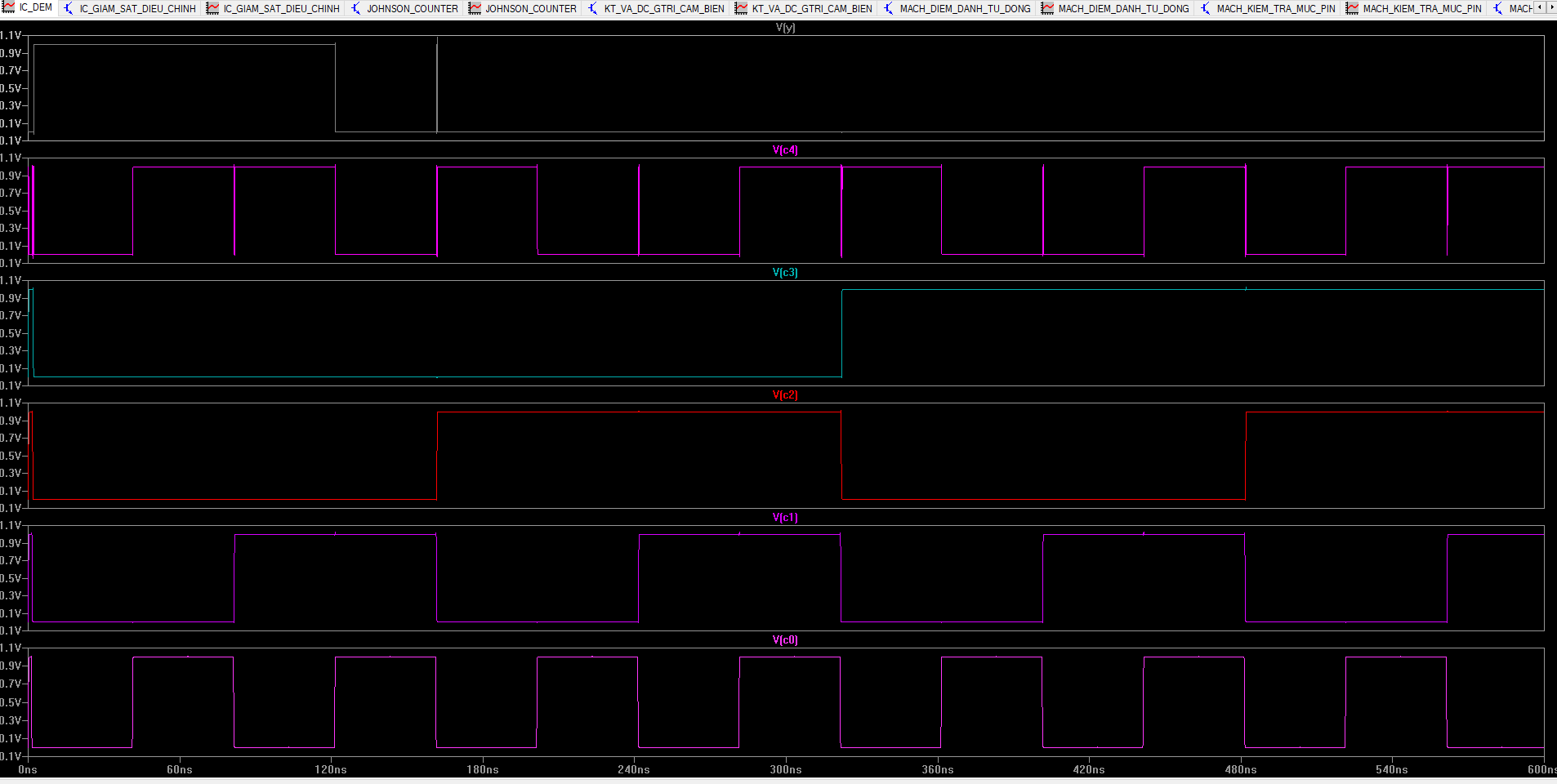


**6. Ic giám sát điều chỉnh Mô tả: Thiết kế mô phỏng IC hoạt động tương tự như bộ giám sát và điều chỉnh lưu lượng nước, bao gồm tính tổng lượng nước, so sánh với ngưỡng, điều chỉnh lưu lượng bằng cách tăng/giảm, và phát tín hiệu cảnh báo nếu vượt ngưỡng. Đầu vào: Lưu lượng nước mỗi giờ (4-bit) so sánh với ngưỡng tối ưu (4-bit), dùng tín hiệu điều chỉnh (0: tăng, 1: giảm) để điều chỉnh tăng, giảm về ngưỡng tối ưu Đầu ra: Lưu lượng nước, tín hiệu cảnh báo (0: bình thường, 1: vượt ngưỡng), và bit parity kiểm tra lỗi.**



**7. IC đếm Mô tả: Thiết kế mô phỏng IC hoạt động tương tự như bộ kiểm soát số lượng hàng hóa trong kho, bao gồm đếm số lượng hàng hóa, so sánh với mức tối thiểu, phát tín hiệu yêu cầu bổ sung, và kiểm tra lỗi dữ liệu. Đầu vào: Số lượng hàng hóa (4-bit), mức tối thiểu (4-bit). Đầu ra: Số lượng hàng hóa hiện tại (4-bit), tín hiệu yêu cầu bổ sung (0: đủ, 1: thiếu), và bit parity kiểm tra lỗi.**

****

****