

Thí Nghiệm Thiết Kế Vi Mạch Số

TÀI LIỆU THỰC HÀNH CƠ BẢN

Để dowload tài liệu thực hành, sinh viên đăng nhập bằng email trường ĐH Bách Khoa
XXX@hcmut.edu.vn

<https://drive.google.com/drive/folders/1UzNZQA7toT6eejjH4pZTDsYthIuITehC?usp=sharing>

Biên soạn: Bùi Lê Quốc Doanh

ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH KHOA ĐIỆN – ĐIỆN TỬ, PHÒNG TN 203 – B3

Email: doanhmanicians2069@gmail.com

Phone: 0327155775

Chỉnh sửa và bổ sung:

Dương Quang Hổ

ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH KHOA ĐIỆN – ĐIỆN TỬ KHÓA K2015

Email: quangho1511184@gmail.com

Nguyễn Minh Hiếu, Mr.

ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH KHOA ĐIỆN – ĐIỆN TỬ, PHÒNG TN 203 – B3

THÍ
NGHIỆM

GIỚI THIỆU CHUNG THIẾT KẾ IC SỐ

0

I. TỔNG QUAN THIẾT KẾ VI MẠCH SỐ.

Trong thế giới vi mạch ngày càng phức tạp như hiện nay, khách hàng và người thiết kế đều mong muốn thời gian đến thị trường ngày càng được rút ngắn. Do đó, việc yêu cầu được đặt ra đó là tối ưu trong quá trình thiết kế và sản xuất. PDK (process design kit) có thể giúp người thiết kế tối ưu hiệu năng của IC và mang thiết kế của mình đến nhà sản xuất. PDK bao gồm các thông số của các cell, thông tin kỹ thuật và rule từ nhà sản xuất.

Các thông số và linh kiện của các cell luận lý được phát triển từ nhà sản xuất để giảm thiểu những sai sót và đảm bảo sự hoạt động chính xác của mạch. Để đơn giản hơn, quá trình thiết kế IC bao gồm standard cell và một số thành phần khác kết hợp tạo nên quy trình thiết kế tổng quát.

Người thiết kế sử dụng symbol từ PDK để vẽ schematic, symbol bao gồm ngõ vào, ngõ ra, nguồn cung cấp và các thông số tùy chỉnh. Sau khi hoàn thành sơ đồ nguyên lý (schematic) sẽ tiến hành mô phỏng bằng một số phần mềm như Hspice, Spectre,... Ngoài ra PDK còn cung cấp layout của các cell, rule file để kiểm tra DRC, LVS và các chiết suất ký sinh để hoàn tất file .gds để đưa đi đến nhà máy sản xuất.

Cấu trúc các thư mục chứa các file và nội dung của các file trong thư viện PDK được trình bày như dưới đây:

Thư mục	File	Nội dung của file
icons	.slib	Thư viện chứa các Symbol. Khi tạo một thiết kế thì Design Compiler sẽ thực hiện “mapping” 1 cell trong file netlist với 1 cell trong thư viện Symbol này
	.sdb	Thư viện chứa các Symbol được biên soạn từ file .slib trong Synopsys database format
models	.lib	Là thư viện bao gồm thông tin characteristics và hàm functions của mỗi cổng. Characeristics bao gồm: tên cổng, tên các chân, diện tích (area), delay arcs, công suất. Ngoài ra, còn chứa các ràng buộc cho thiết kế, điều kiện hoạt động và wire load models,...
	.db	Là thư viện chứa thông số hiệu năng của các cổng, được biên soạn từ file .lib nhờ sử dụng phần mềm DC của Synopsys
plib	.plib	Thư viện vật lý bao gồm thông tin về process và thông tin layout vật lý của các cổng
	.pdb	Thư viện được biên soạn từ file .plib cũng được tạo ra nhờ sử dụng phần mềm DC của hãng Synopsys
tech	.tf	Bao gồm các thông tin kỹ thuật cụ thể như tên, đặc tính vật lý, đặc tính điện cho mỗi lớp Metal và các niêm luật cho thiết kế
tlup	.itf	Mô tả độ dày và thuộc tính vật lý của các lớp dẫn điện và cách điện
	.tluplus	Được sinh ra từ file .itf, là một tập các models được dùng trong việc trích xuất ký sinh của phần mềm Synopsys Place & Route

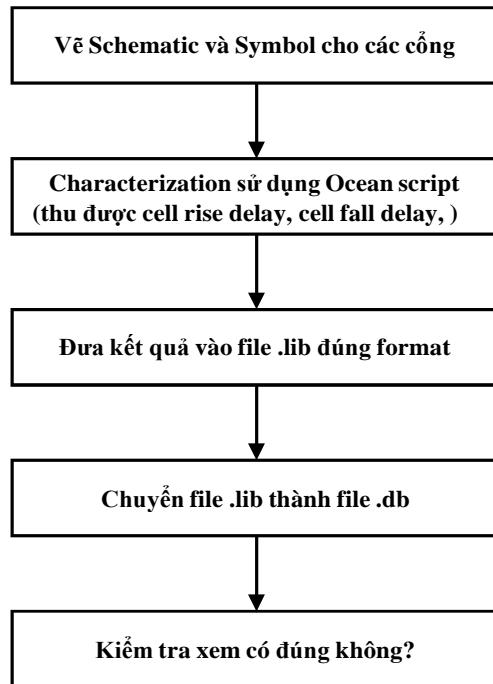
II. QUY TRÌNH XÂY DỰNG MỘT CỔNG LOGIC

Bước 1: Thiết kế thư viện ở cấp độ nguyên lý (Schematic & Simulation)

Ở bước 1, thực hiện tạo thư viện dựa trên các kết quả thu được từ mô phỏng Pre- Layout. Các kết quả này là lý tưởng không chính xác với thực tế khi sản xuất, mục đích của bước này là kiểm tra xem với các số liệu lý tưởng này có thể thực hiện được thư viện hoạt động đúng hay không? Bước này hết sức quan trọng, nếu thành công ở bước này thì mới có thể thực hiện tiếp bước thứ 2. Ngược lại, nếu không thành công thì ta phải tìm ra hướng giải quyết khác để hoàn thành mục tiêu của đề tài. Cụ thể ở bước này được thực hiện như sau:

- Thiết kế sơ đồ nguyên lý cho các cổng trong thư viện và tạo ký hiệu cho các cổng đó.
- Thực hiện Characterization sử dụng Ocean script và mô phỏng Pre-Layout để đo được các thông số cần thiết để tạo thư viện.
- Tiếp theo, tạo ra file LIB chứa các thông số vừa đo được ở bước trên. File LIB này được thiết kế theo chuẩn của phần mềm Synopsys. Đối với phần mềm Cadence thì file này có tên là file TLF.

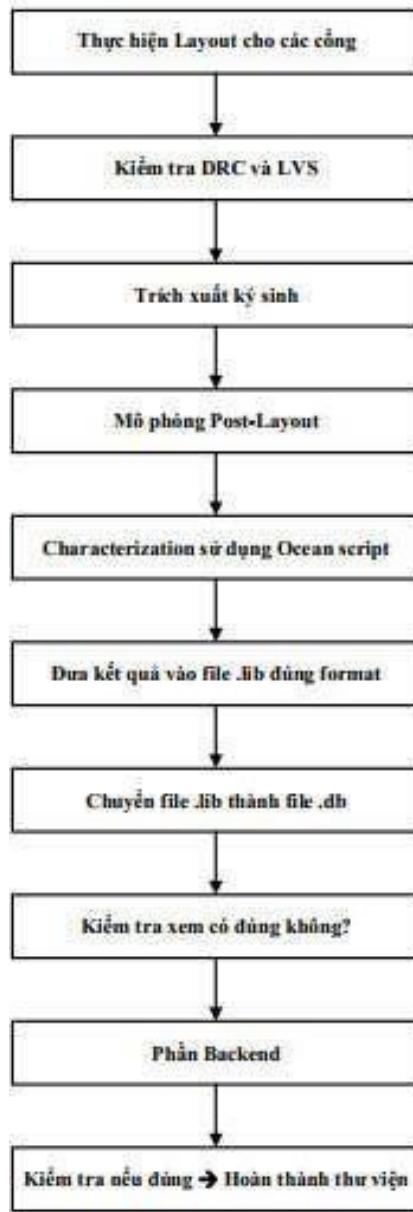
- Chuyển file LIB thành file DB sử dụng phần mềm DC của hãng Synopsys. File DB này là một trong những file rất quan trọng trong thư viện, nó chứa các thông số về hiệu năng của các cổng logic được tạo nên từ các CMOS ở công nghệ nhất định. Kiểm tra thư viện xem có hoạt động đúng không bằng cách thực hiện bước Synthesis. Ở bước này, ta sẽ viết một đoạn code RTL và Synthesis xuống mức cổng. Nếu bước kiểm tra này thành công, ta chuyển sang bước 2 để tạo thư viện cho luận văn. Hình 2-1 dưới đây mô tả các bước thực hiện ở bước 1:



Hình 1-1 Lưu đồ các bước tạo thư viện từ kết quả mô phỏng Pre-Layout

Bước 2: Thiết kế thư viện ở cấp độ vật lý (Layout & Simulation)

Ở bước này, ta thực hiện tạo thư viện Front-end. Đầu tiên, thực hiện thiết kế layout cho tất cả các cổng trong thư viện sau khi đã có sơ đồ nguyên lý ở bước 1. Kiểm tra DRC và LVS cho thiết kế layout, mong muốn đạt được ở bước này là không có lỗi và bất kỳ cảnh báo nào về thiết kế. Nếu không đạt được kết quả đó thì các bước sau sẽ không thể thực hiện tiếp được vì lỗi của phần mềm thiết kế. Tiếp theo, ta thực hiện chạy PEX để trích xuất ký sinh RC do trong quá trình layout gây ra. Sau bước này sẽ thu được một cell view có tên là calibre trong thư mục của cổng đang kiểm tra, sử dụng cell view này để thực hiện mô phỏng Post-Layout. Tương tự như ở bước 1, thực hiện Characterization sử dụng Ocean script nhưng thực hiện dựa trên mô phỏng Post-Layout. Sau khi thu được các thông số cần cho thư viện, tiến tới tạo file LIB và file DB. Tiến hành kiểm tra hoạt động bằng bước Synthesis. Cuối cùng, thực hiện phần Back-end bao gồm STA, Place & Route tạo ra file LEF, ta đã thu được thư viện hoàn chỉnh. Quy trình thực hiện được trình bày trong hình 1-2 dưới đây:



Hình 1-2 Lưu đồ các bước tạo thư viện từ kết quả mô phỏng Post-Layout

III. HƯỚNG DẪN SỬ DỤNG CƠ BẢN HỆ ĐIỀU HÀNH LINUX

Trong mục này, các bước cơ bản trong sử dụng Hệ Điều Hành (HĐH) Linux được giới thiệu. Vì bộ công cụ thiết kế Cadence EDA chỉ hỗ trợ cài đặt trên Linux nên việc sử dụng Hệ Điều Hành này trong quá trình thiết kế là điều hết sức quan trọng. Việc sử dụng Linux tương tự như việc sử dụng chế độ cmd của DOS. Các lệnh của Linux được đánh trong một “shell” hoặc “xterm”.

Bảng dưới tổng hợp một số lệnh cần thiết trong quá trình sử dụng HĐH Linux mà người thiết kế cần phải thành thạo nhằm mục tiêu tối ưu hóa thời gian truy xuất hoặc trích xuất dữ liệu trong quá trình thực hành Thí Nghiệm Thiết Kế Vi Mạch Tương Tự. Tất cả các lệnh trong HĐH Linux được đánh trong một cửa sổ đầu cuối thường gọi là “shell” hoặc “terminal” – tương tự như cửa sổ cmd trong HĐH WINDOW

Lệnh	Chú giải
cd ..	Di chuyển vị trí thư mục hiện tại quay lại thư mục trước đó
cd XXXX	Thay đổi vị trí thư mục hiện tại đến thư mục XXXX
cp XXXX YYYY	Sao chép dữ liệu từ XXXX đến YYYY
mv XXXX YYYY	Di chuyển dữ liệu từ XXXX đến YYYY
rm XXXX	Xóa dữ liệu khỏi XXXX
mkdir XXXX	Tạo một thư mục mới tên XXXX
tar -xf XXXX	Xuất dữ liệu chứa trong file nén XXXX.tar ra ngoài thư mục hiện hành.
virtuoso &	Mở phần mềm Cadence Virtuoso
gedit XXXX &*	Mở trình hiệu chỉnh textfile với tên file là XXXX
lp -dXXXX YYYY	In dữ liệu textfile hoặc postscript của file YYYY đến file tên XXXX, trong đó XXXX có thể là “ipszac” hoặc “hpszac”
top	Kiểm tra phần cứng: nhân xử lý và bộ nhớ
quota -v	Kiểm tra dung lượng ổ đĩa
who grep my_name	Hiển thị trên terminal vị trí đang kết nối
gunzip XXXX	Chuyển file nén XXXX đuôi *.tar.gz thành *.tar để trích xuất dữ liệu nén
ls [-l/-a]	Trình diễn các dữ liệu trong thư mục hiện hành. “-l” trình diễn với đặc tính đi kèm. “-a” trình diễn các dữ liệu ẩn (tên có dấu “.” ở đầu)

IV. CẤU HÌNH MÔI TRƯỜNG THÍ NGHIỆM

Phần mềm chuyên dụng sử dụng trong tài liệu này là bộ Cadence EDA hay còn gọi là Cadence Virtuoso bản 6.1.6 – 64bit với giấy phép hoạt động được lấy thông qua dự án HEAP với 10 user được cấp phép sử dụng phần mềm cho mục đích đào tạo nhân lực ngành thiết kế vi mạch. Đầu tiên trước khi khởi động phần mềm, cần phải cài đặt vùng làm việc đối với từng học viên.

1. Giải nén file **Red Hat Enterprise 4_cadence614_181023.zip** (44 GB)
2. Trong thời gian giải nén, để tránh mất thời gian ta cài đặt phần mềm(giải nén file exe.zip) **VMware Worstation(update to VMWARE 16 PRO)**
Nhập license sau để hoàn thành cài đặt phần mềm(4 license; dùng 1 trong 4 để kích hoạt) :

ZF3R0-FHED2-M80TY-8QYGC-NPKYF

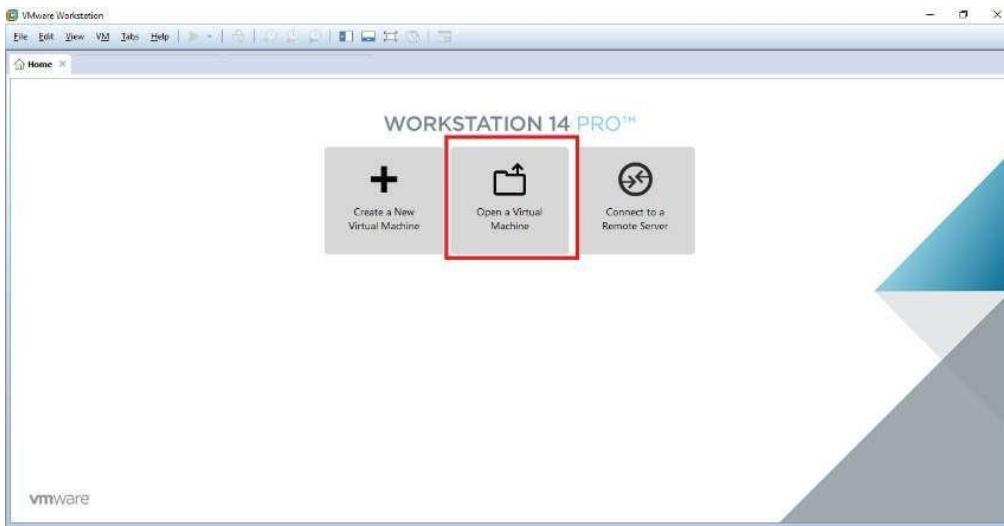
YF390-0HF8P-M81RQ-2DXQE-M2UT6

ZF71R-DMX85-08DQY-8YMNC-PPHV8

FC11K-00DE0-0800Z-04Z5E-MC8T6

3. Hướng dẫn khởi động file máy ảo

- Tại trang chủ Vmware: click **Open the virtual machine**, chọn file **Red Hat Enterprise Linux 4**



- Click icon ➤ để khởi động máy ảo
- Đăng nhập Account

Name: root

Password: 123456

4. Khởi động tool Virtuoso

Tại desktop, click chuột phải chọn **Open Terminal**. Tại cửa sổ Terminal hiện hành, thực hiện gõ các lệnh sau để khởi động Virtuoso Tool:

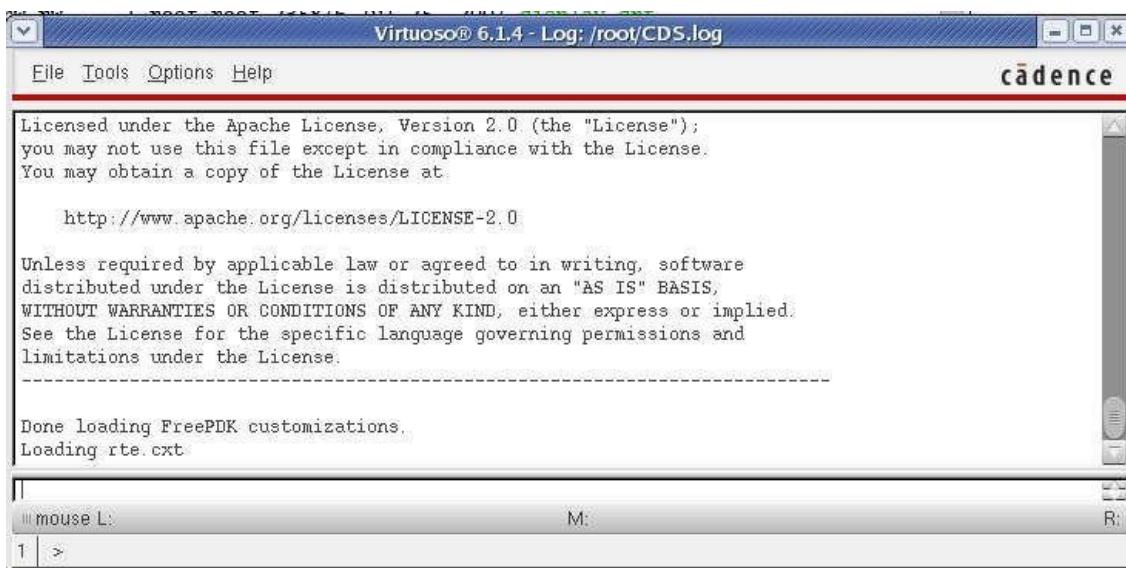
```
cd /root/VIRTUOSO_PDK45      # Di chuyển con trỏ vào thư mục cài đặt thí nghiệm  
license                         # Get license  
virtuoso &                      #Khởi động Virtuoso Tool
```

V. HƯỚNG DẪN SỬ DỤNG TOOLS CADENCE

Để hiểu rõ hơn về cách sử dụng phần mềm Cadence Virtuoso cũng như quy trình thiết kế vi mạch, ví dụ thiết kế công luận lý INVERTER được thực hiện trong phần dưới đây. Trong toàn bộ quy trình sẽ lần lượt giải thích tường minh từng bước sử dụng phần mềm cùng những công cụ phổ biến ứng dụng trong thiết kế vi mạch tương tự, các công cụ mô phỏng, vẽ mạch hoặc ngôn ngữ sử dụng để mô hình hóa một công luận lý INVERTER.

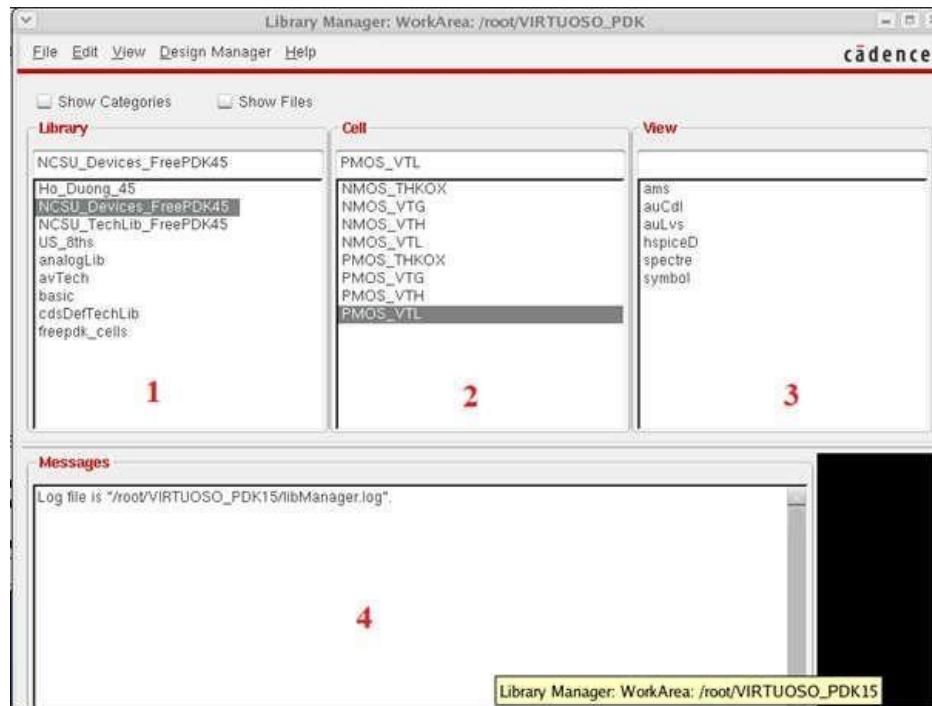
1. THỰC HIỆN TẠO MỘT THƯ VIỆN MỚI

Sau khi khởi động Phần Mềm Cadence Virtuoso, cửa sổ CIW hiện ra như sau:



Trình CIW này hoạt động tương tự như một tập tin *.log dùng để hiển thị tất cả hoạt động trong quá trình thực hiện thiết kế vi mạch sử dụng Cadence EDA. Người thiết kế có thể dùng cửa sổ CIW này để truy nhập bất cứ phần mềm, công cụ tích hợp nào của phần mềm Cadence bằng cách đánh lệnh SKILL trong cửa sổ CIW.

Cửa sổ Library Manager hiện ra như hình dưới.(Nếu không hiện cửa sổ thì chọn Tools > Library Manager...)



1 Library	Các thư viện thiết kế, kể cả PDK tham chiếu đều được hiển thị trong vùng này.
2 Cell	Các khối thiết kế được hiển thị tại vùng này nhằm thông báo cho người thiết kế biết vị trí các khối thiết kế.
3 View	Các khối thiết kế vốn là một thư mục chứa những dữ liệu thiết kế. Các tập tin dữ liệu này được hiển thị trong View. Ví dụ như Sơ Đồ Nguyên Lý (Schematic), Ký Hiệu (Symbol)...
4 Message	Tập tin *.Log kiểm soát hoạt động của Library Manager nhằm thông báo cho người thiết kế biết vị trí các tập tin để kiểm tra lỗi thiết kế.

Thực hiện tạo một thư viện mới: trong cửa sổ Library Manager thực thi:

File > New > Library...

Cửa sổ tạo thư viện mới xuất hiện



Thực hiện nhập tên thư viện mới vào và chọn OK. Để thực hiện thiết kế trên nền một thư viện ta cần phải tham chiếu đến một PDK thiết kế chứa những linh kiện bán dẫn mà cụ thể ở đây là *NCSU_TechLib_FreePDK45*

Thực hiện chọn:

Attach to an existing technology library > NCSU_TechLib_FreePDK45 > OK



2. XÂY DỰNG MÔ HÌNH LÝ TƯỞNG

Tiếp theo, để thực hiện một thiết kế mạch nguyên lý mới, trong cửa sổ Library Manager chọn

vào thư viện vừa tạo và chọn:

File > New > Cell View...

Cửa sổ tạo mạch nguyên lý mới hiện ra như sau:



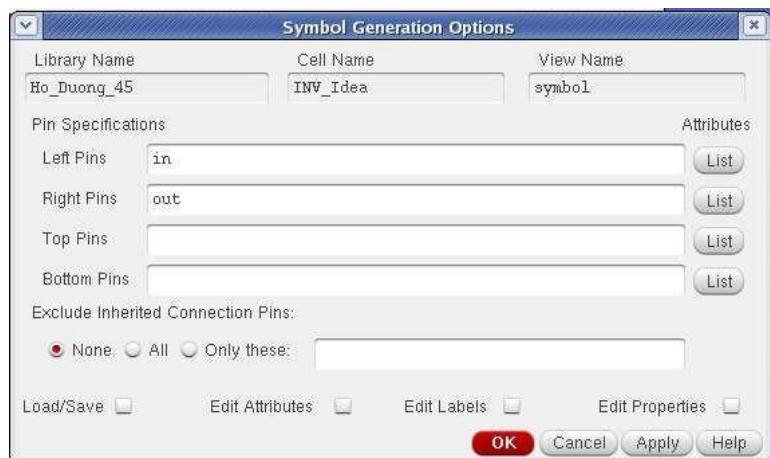
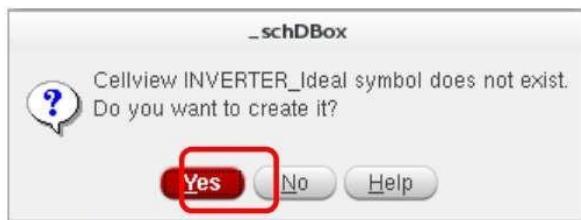
Thực hiện nhập Tên của mạch thiết kế, và chọn loại thiết kế. Trong vị trí chọn loại thiết kế này chứa nhiều dạng. Đầu tiên, thực hiện xây dựng mô hình lý tưởng hay macro cho mạch công luận lý đảo bằng ngôn ngữ Verilog Analog.

Cửa sổ soạn thảo hiện ra, nhập code Verilog-A cho cổng Inverter_Idea như sau:

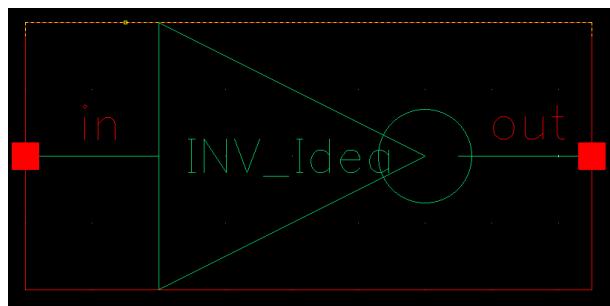
```
// VerilogA for Ho_Duong_45, INV_Idea, veriloga
`include "constants.vams"
`include "disciplines.vams"
module INV_idea(in, out);
input in;
output      out;
electrical in,out;
parameter real vout_high = 1;
parameter real vout_low = 0;
parameter real vth = 0.5;
parameter real tdelay = 1n; //propagation delay
parameter real trise = 1p; //rise time
parameter real tfall = 1p; //fall time
real val;
analog begin
  @( cross( V(in) - vth, -1 ) ) val = vout_low;
  @( cross( V(in) - vth, +1 ) ) val = vout_high;
  V(out) <+ transition(val, tdelay, trise, tfall);
end
endmodule
```

Tương tự như ngôn ngữ Verilog, Verilog-A cũng bao gồm những thành phần khai báo tương tự như trong soạn thảo ngôn ngữ Verilog. Tuy nhiên trong quá trình thực hiện khai báo của Verilog A bên cạnh việc khai báo những mức áp, còn phải khai báo thông số thời gian bao gồm thời gian lên, thời gian xuống và thời gian trì hoãn của cỗng luận lý.

Sau khi thực hiện xây dựng mô hình lý tưởng của cỗng Đảo bằng ngôn ngữ Verilog-A, công việc tiếp theo là biên dịch.(**Save** và **Close** trình soạn thảo).



Thực hiện chọn **OK** màn hình chuyển đến vị trí thực hiện vẽ ký hiệu cho cỗng luận lý. Việc sử dụng trình này rất dễ dàng để tạo ra những ký hiệu đại diện cho hoạt động của cỗng Đảo và các mạch khác. Xét về mặt đồ họa thì Cadence đã cung cấp đủ cho chúng ta toàn bộ các dụng cụ để vẽ được cỗng Đảo. Thực hiện vẽ cỗng đảo dựa vào những công cụ đồ họa như trong hình.



3. XÂY DỰNG SƠ ĐỒ NGUYÊN LÝ

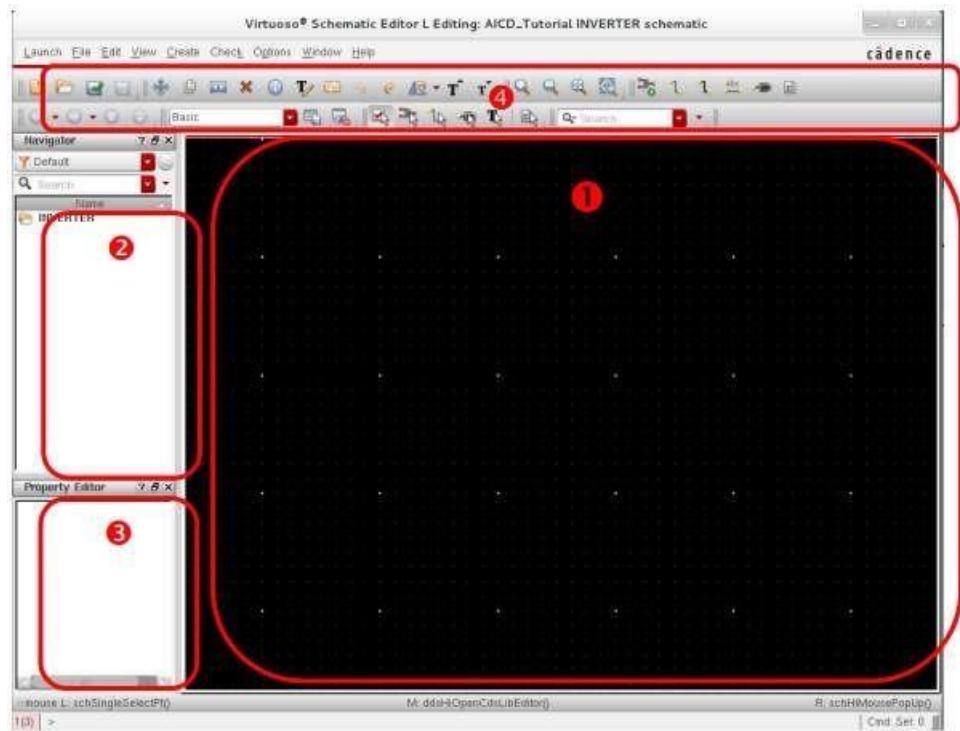
Bước kế tiếp, thực hiện tạo sơ đồ nguyên lý cho mạch – tức thực hiện thiết kế mạch ở cấp transistors. Trong cửa sổ Library Manager, thực hiện:

File > New > Cellview...

Cửa sổ tạo một mạch mới hiển thị như sau:



Trong phần **Type** chọn **schematic** và open with **schematics L**. Lúc này, cửa sổ hiển thị thiết kế mạch nguyên lý hiển thị:

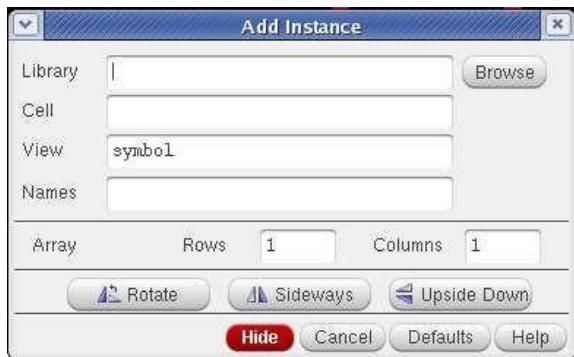


①	Workspace	Vùng đồ họa dùng để vẽ mạch và chứa những phần tử linh kiện lấy ra từ trong thư viện thiết kế PDK.
②	Name	Những kết nối xuất hiện trong quá trình thiết kế mạch nguyên lý sẽ được hiển thị ở đây. Vì vậy, việc quản lý kết nối khá dễ dàng trong những thiết kế phức tạp. Vùng này còn được dùng để lựa chọn cũng như hiển thị kết nối trong những mạch phức tạp.
③	Properties Editor	Bảng hiển thị thuộc tính của từng linh kiện. Trong quá trình sử dụng phần mềm VSE để hiển thị nhanh vùng làm việc có thể dựa vào Properties Editor để chỉnh sửa, thêm thông số cho những linh kiện sử dụng trong thiết kế được nhanh hơn.
④	Tools Bar	Thanh công cụ cho phép người thiết kế sử dụng những chức năng cơ bản của việc thiết kế đồ họa mạch nguyên lý. Tuy nhiên, để thiết kế nhanh hơn và chuyên nghiệp hơn, người thiết kế thường sử dụng những phím tắt thay cho việc chọn biểu tượng trên thanh công cụ.

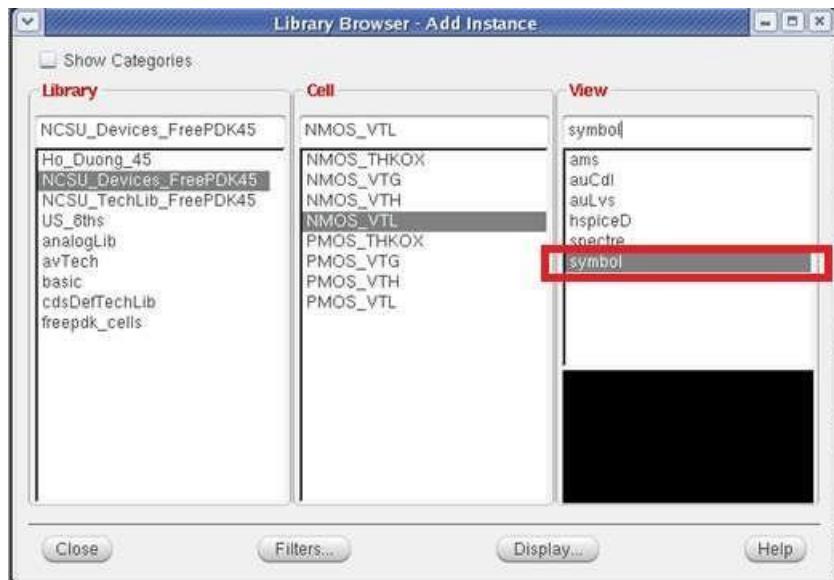
Để sử dụng tốt quy trình vẽ mạch nguyên lý trong VSE, những phím tắt cần được sử dụng một cách thành thạo. Bảng dưới chỉ ra những phím tắt cơ bản trong sử dụng VSE, đồng thời giải thích ý nghĩa của những phím tắt này.

Phím Tắt	Chú giải
Q	Xen thuộc tính của một phần tử
C	Sao chép. Nhấn c sau đó chọn phần tử rồi di chuyển đến vị trí cần sao chép đến.
M	Di chuyển phần tử. Nhấn m sau đó chọn phần tử rồi di chuyển phần tử
Delete	Xóa phần tử
ESC	Dừng thực thi lệnh
F	Phóng lớn toàn màn hình
I	Thêm phần tử vào vùng làm việc
L	Thêm nhãn cho các đường dây
W	Nối dây các phần tử
P	Thêm chân cho phần tử (dùng cho tạo ký hiệu)
E	Chọn cây thư mục hiện tại
R	Xoay một phần tử
X	Lưu lại mạch đang thiết kế
Shift + R	Lấy đối xứng một phần tử
U	Lùi lại trạng thái trước
Z	Phóng lớn trong hộp – dùng cho những thiết kế phức tạp
F1	Giúp đỡ thiết kế
F6	Vẽ lại mạch
8	Thu nhỏ
9	Phóng lớn
Shift + Y	Lưu toàn bộ các cây thư mục
Shift + X	Lưu mạch hiện tại đang thiết kế

Để thêm linh kiện mới cho mạch nguyên lý thực hiện chọn phím tắt **I** bảng thông báo thêm linh kiện hiển thị. Việc lựa chọn thông số cho linh kiện được thực hiện trong bảng thông báo này.



Chọn **Browser** bảng thông báo **Library Browser – Add Instance** hiển thị



Cần phân biệt cửa sổ **Library Browser** và **Library Manager**. Hai cửa sổ này tuy gần nhau nhưng chức năng, thuộc tính hoàn toàn khác nhau.

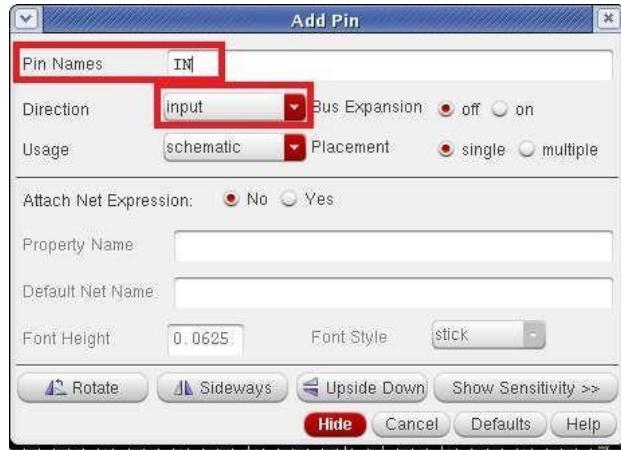
Chọn thư viện **NCSU_Devices_FreePDK45** rồi chọn linh kiện **NMOS_VTL**, **PMOS_VTL** và đưa ra ngoài màn hình thực hiện vẽ mạch nguyên lý của công Đảo. Trong việc chọn lựa linh kiện có những linh kiện được sử dụng nhiều lần và thường hữu ích trong quá trình mô phỏng hoặc thiết kế mô hình kiểm tra cho mạch nguyên lý. Các phần tử sử dụng nhiều trong các bài thí nghiệm được cho như trong bảng bên dưới.

Tên Phần tử	Thu Viện	Phần Tử
vdd	analogLib	Nguồn
gnd	analogLib	Đất
vdc	analogLib	Nguồn áp DC
idc	analogLib	Nguồn dòng DC
vsin	analogLib	Nguồn áp sine
isin	analogLib	Nguồn dòng sine
vpulse	analogLib	Nguồn áp xung
ipulse	analogLib	Nguồn dòng xung
vcvs	analogLib	Nguồn áp điều khiển được
res	analogLib	Điện trở lý tưởng
cap	analogLib	Tụ điện lý tưởng
ind	analogLib	Cuộn dây lý tưởng
NMOS_VTL	NCSU_Devices_FreePDK45	N-MOSFET 1v 45nm
PMOS_VTL	NCSU_Devices_FreePDK45	P-MOSFET 1v 45nm

Thực hiện kết nối dây bằng phím tắt **W** và vẽ hoàn chỉnh sơ đồ thiết kế của công Inverter. Để đặt nhãn cho linh kiện thực hiện chọn phím tắt **L** cửa sổ đặt tên dây hiện ra, lần lượt đánh ào vị trí đặt tên dây các tên: input, output như hình vẽ bên dưới. Có thể chọn kích thước, màu, kểu chữ, tùy thuộc vào mục tiêu ứng dụng nhưng phải dễ hiểu.



Thực hiện đặt tên chân nhằm mục tiêu tạo ký hiệu dùng trong quá trình mô phỏng linh kiện. Chọn phím tắt **P** màn hình hiển thị tạo chân kết nối cho mạch xuất hiện.



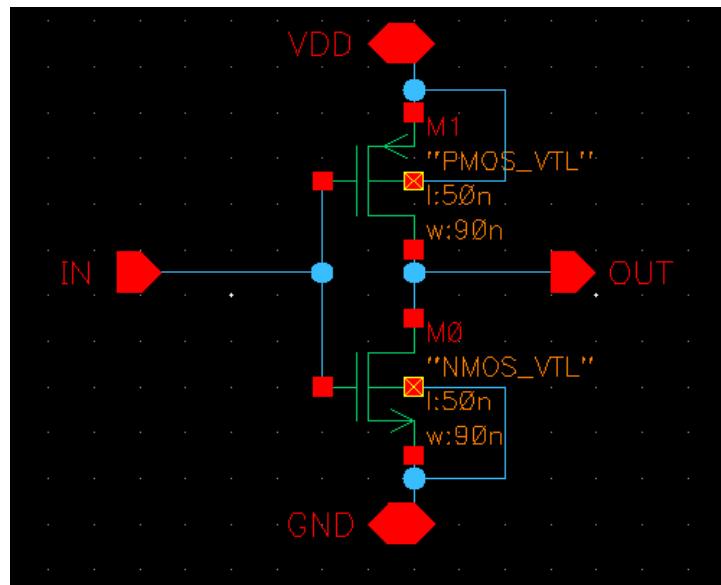
Lần lượt tạo các PIN: IN, OUT, VDD, GND

Trong đó, hướng của mỗi dạng chân được quy định trong Direction với những kiểu như sau:

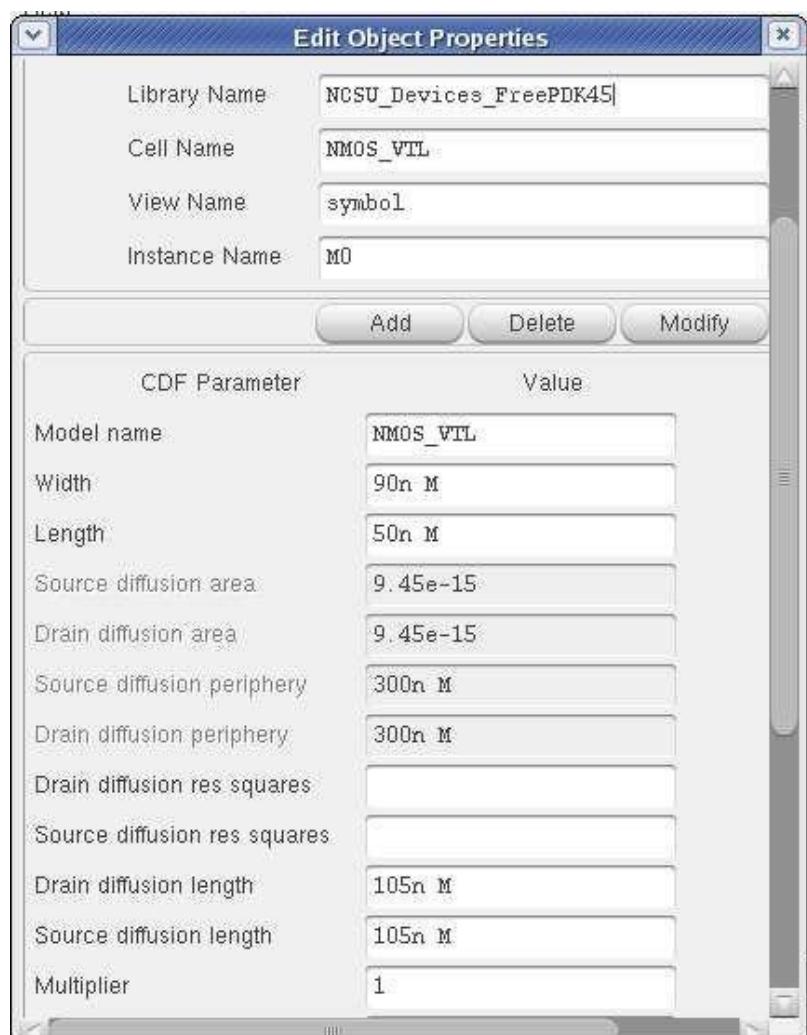
Tên chân	Dạng chân
VDD	inputOutput
GND	inputOutput
IN	input
OUT	output

Kết quả thu được dạng mạch nguyên lý của cỗng Đảo như trong hình bên dưới.

Bố trí các phần tử trong mạch được chú thích như trong hình.



Để hiệu chỉnh thuộc tính của các linh kiện đã thêm vào trong mạch, thực hiện chọn **Q** rồi chọn linh kiện muốn chỉnh sửa. Bảng thuộc tính của các linh kiện chứa những thông số mô hình của linh kiện đó cụ thể là thông số tính toán của mô hình mô phỏng thuộc phần mềm SPECTRE. Từ những thông số hình học cơ bản, phần mềm Cadence sẽ thực hiện tính toán những thông số như tụ điện ký sinh, điện trở ký sinh ... Để đưa vào mô hình mô phỏng sử dụng trong trình mô phỏng SPECTRE. Đối với từng linh kiện như Điện trở, Tụ Điện, MOSFET, BJT ... những thông số mô hình này khác nhau đối với từng PDK cụ thể và phục thuộc vào mô hình toán học trích xuất, tính toán ký sinh cho từng linh kiện. Trong tài liệu này chỉ tóm tắt lại những thông số linh kiện quan trọng phổ biến trong thiết kế và các bài thí nghiệm.



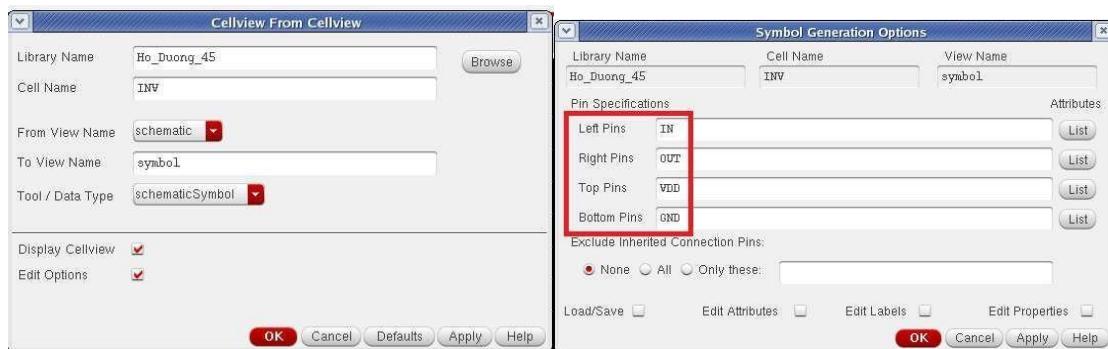
Sau khi vẽ xong, chọn **"File->Check and Save"** để kiểm tra lỗi và lưu, hoặc có thể sử dụng hotkey F8.

 <p>Static Parameters</p>	<p>Khai báo tên và vị trí thư viện thiết kế.</p> <p>Những thông số tĩnh của linh kiện. Những thông số này cần phải được nhập vào sau khi quá trình tính toán bằng tay được thực hiện</p>
 <p>MOSFET</p>	<p>Những thông số động được tính toán dựa vào mô hình toán học với tham số lấy từ những thông số tĩnh.</p>
<p>Bảng thông số, thuộc tính quan trọng của những linh kiện cơ bản được cho như bên dưới:</p>	
 <p>Resistor</p>	<p>Model Name – Mô hình mô phỏng SPECTRE Multiplier – Số lượng linh kiện MOS mắc song song Length (M) – Chiều dài kenh Total Width (M) – Độ rộng cực cổng Finger Width – Độ rộng nhánh của MOSFET Fingers – Số lượng nhánh của mỗi MOSFET</p> <p>Model Name – Mô hình mô phỏng SPECTRE Segments – Số lượng nhánh mắc nối tiếp hoặc song song Segment Connection – Lựa chọn mắc nối tiếp hoặc song song Calculated Parameter – Lựa chọn kiểu thông số tính toán Resistance – Trở kháng toàn phần của điện trở Segment Width – Độ rộng nhánh Segment Length – Độ dài nhánh Effective Width – Độ rộng nhánh hiệu dụng Effective Length – Độ dài nhánh hiệu dụng</p>
 <p>Moscap</p>	<p>Model Name – Mô hình mô phỏng SPECTRE Multiplier – Số lượng MOS mắc song song Calculated Parameter – Tính toán thông số hình học Capacitance – Dung kháng của tụ điện Length (M) – Chiều dài cực cổng Total Width (M) – Độ rộng toàn phần Finger Width – Độ rộng nhánh Fingers – Số lượng nhánh của tụ điện MOS</p>
 <p>Bipolar</p>	<p>Model Name – Mô hình mô phỏng Device Area – Diện tích Emitter Width – Độ rộng cực phát Multiplier – Số lượng BJT mắc song song</p>

5**Model name** - Mô hình mô phỏng**Calculated Parameter** – Lựa chọn thông số “diện tích”, “độ dài hay “độ rộng”**Device Area** – Diện tích môi nối tiếp giáp**Length (M)** – Độ rộng của Diode**Width (M)** – Độ dài của Diode**Multiplier** – Số lượng Diode mắc song song

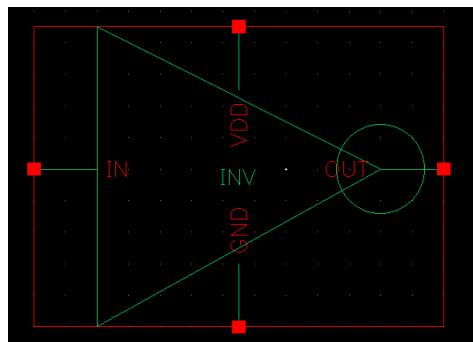
- Tạo Symbol cho Inverter:**

Create > Cellview > From Cellview...

Bảng thông báo Cellview From Cellview hiện ra, chọn **OK**Bảng thông báo Symbol Generation Options hiện ra, điều chỉnh các tên Pin, chọn **OK**

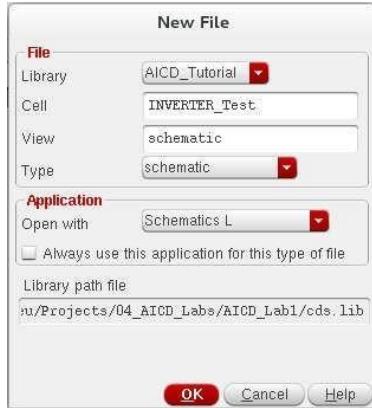
Màn hình chuyển đến vị trí thực hiện vẽ ký hiệu cho công luận lý. Việc sử dụng trình này rất dễ dàng để tạo ra những ký hiệu đại diện cho hoạt động của công Đảo và các mạch khác. Xét về mặt đồ họa thì Cadence đã cung cấp đủ cho chúng ta toàn bộ các dụng cụ để vẽ được công Đảo. Thực hiện vẽ công đảo dựa vào những công cụ đồ họa như trong hình.

Sử dụng các trên thanh công cụ để vẽ ký hiệu mạch như sau:

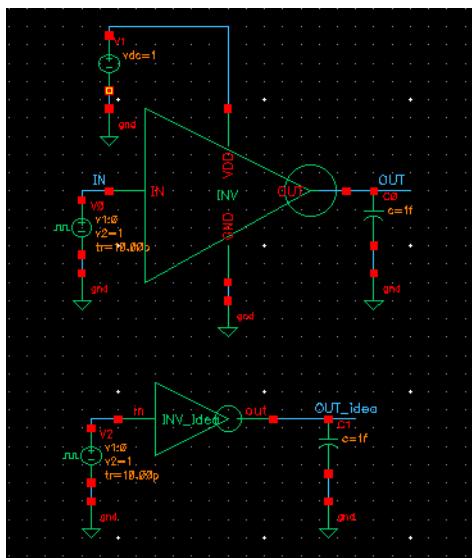


4. MÔ PHỎNG PRE-LAYOUT

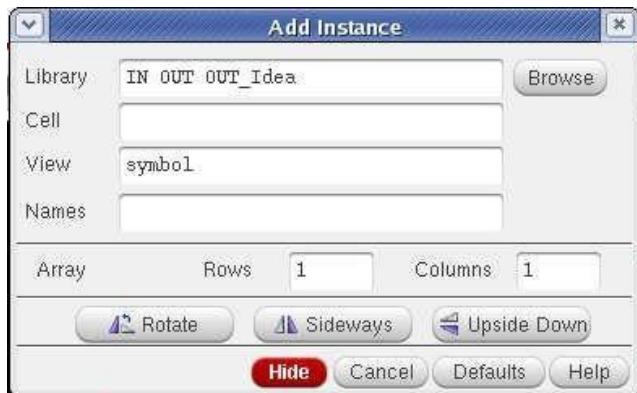
Trong cửa sổ Library Manager, thực hiện chọn:
File > New > Cellview...



Tạo một mạch mới với tên gọi INVERTER_Test và xây dựng mạch như trong Hình dưới.



Cài đặt Label cho các net để dễ phân biệt khi mô phỏng.(Sử dụng Hot key “L”)



Các thông số cần thiết:

Thông số	Giá trị
V_{dd}	1 V
C_{load}	1 fF
Voltage 1	0 V
Voltage 2	1 V
Rise time	1 ps
Fall time	1 ps
Delay	0 ns
Pulse width	1 ns
Period	2ns

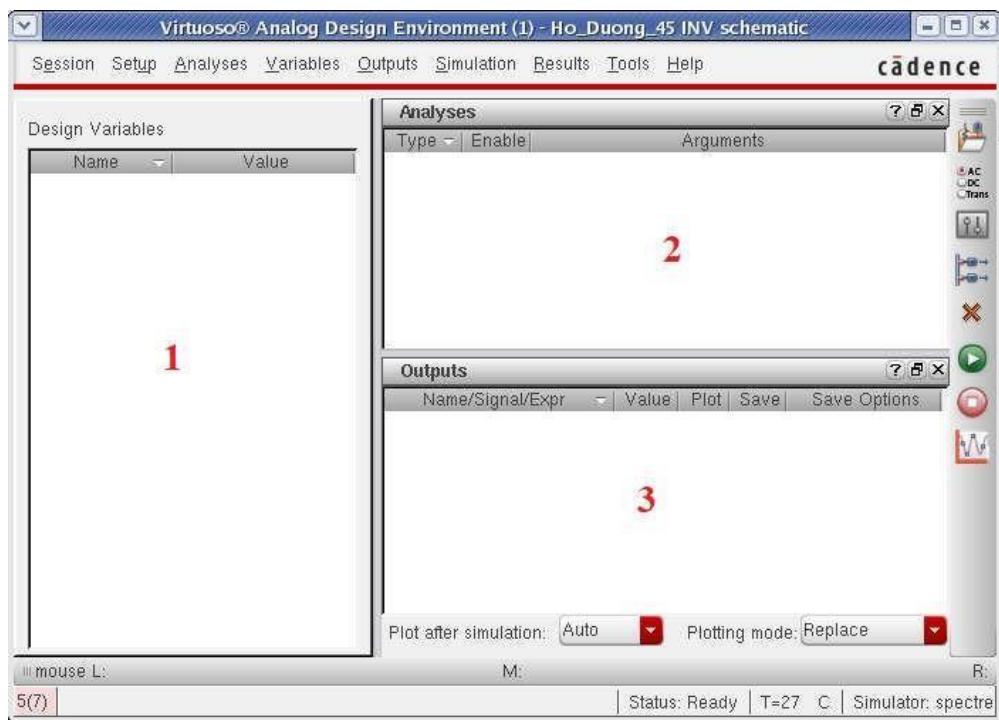
Mạch trên bao gồm hai phần một phần dùng để kiểm tra hoạt động của công Đảo thiết kế trên CMOS, mạch còn lại dùng để mô phỏng dạng sóng lý tưởng của công Đảo.

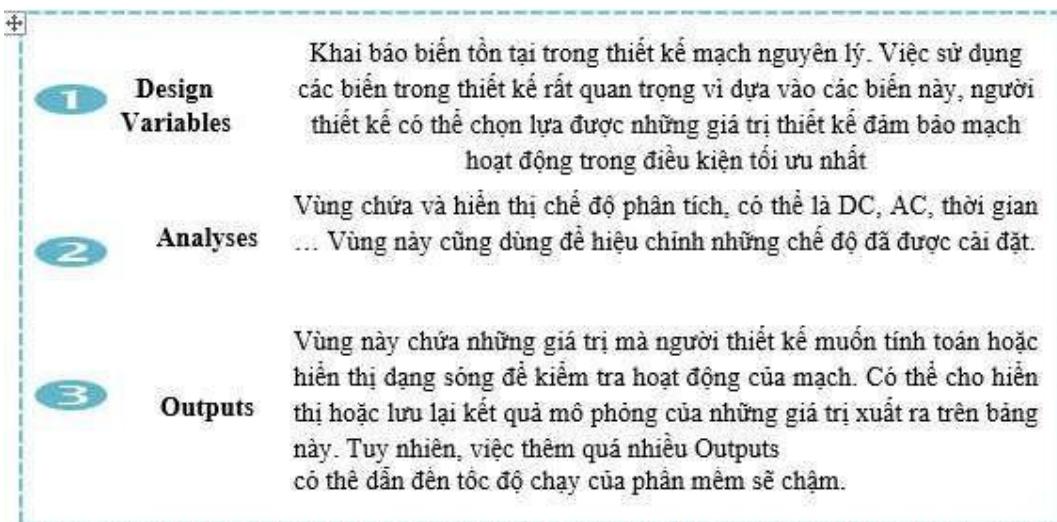
Trong môi trường Cadence Virtuoso, trình mô phỏng được chạy trên giao diện đồ họa Analog Design Environment (ADE). Trình này được khởi động bằng cách:

Trong cửa sổ thiết kế mạch nguyên lý, thực hiện chọn:

Launch > ADE L

Trình mô phỏng có dạng như Hình sau





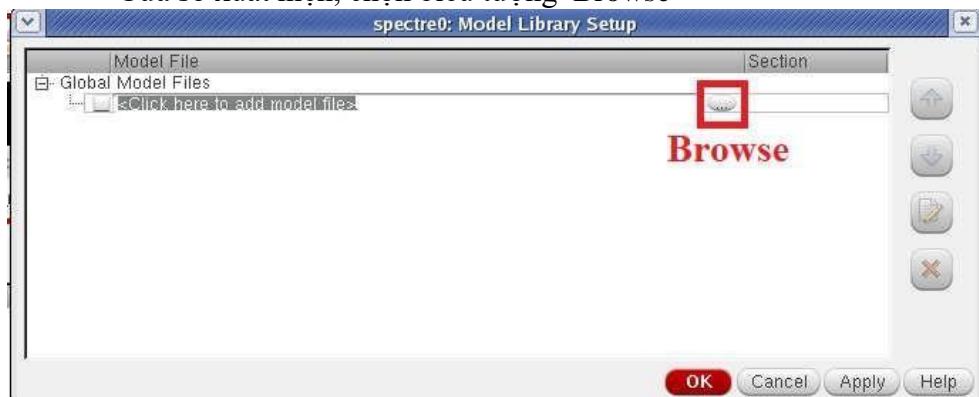
Trong trình mô phỏng này, công việc đầu tiên là thực hiện lựa chọn loại phần mềm mô phỏng. Trong Cadence có tích hợp nhiều bộ công cụ mô phỏng với những tính năng khác nhau. Tùy thuộc vào mục tiêu mô phỏng, ứng dụng của mạch mà lựa chọn công cụ mô phỏng sau cho phù hợp. Các mục tiêu mô phỏng có thể chia ra như sau: phương pháp phân tích, tốc độ, lấy gần đúng khi thực hiện những hệ thống vi sai. Những phương pháp phân tích có thể được sử dụng bao gồm: Phân tích DC, mô hình tương đương tín hiệu nhỏ, phân tích thời gian, phân tích nhiễu, phân tích hài và phân tích độ nhạy của vi mạch. Trong khóa học thí nghiệm này, tài liệu tập trung vào những phương pháp phổ biến, được sử dụng trong toàn bộ những thiết kế vi mạch tương tự bao gồm: Phân tích DC, AC và Thời gian.

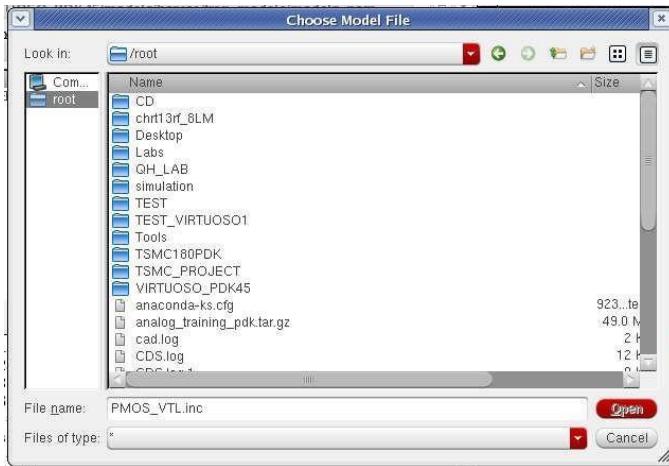
Thông thường, quy trình cài đặt mô phỏng một vi mạch được chia làm bốn phần: setup model library, phân tích, cài đặt tín hiệu xuất, cài đặt biến thiết kế.

a) Setup Model Library

Chọn Setup > Model Library

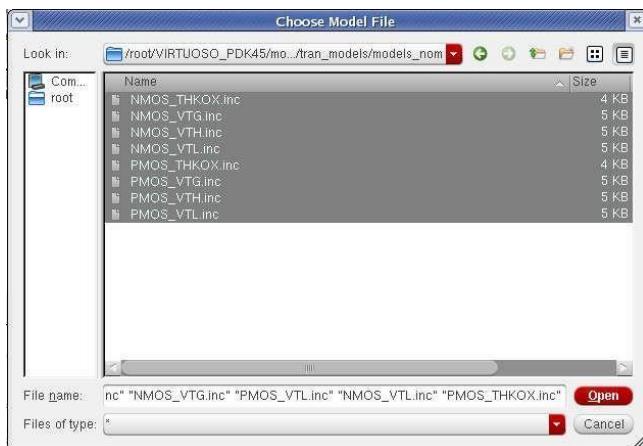
Cửa sổ xuất hiện, chọn biểu tượng Browse





Chọn theo đường dẫn sau

/root/VIRTUOSO_PDK45/models/hspice/tran_models/models_nom
Open tất cả các Model có trong file "models_nom"

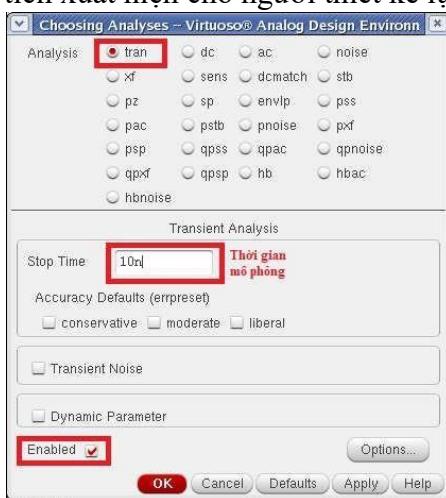


>> Chọn OK để Save

b) Thực hiện chọn chế độ phân tích, trong cửa sổ ADE chọn:

- Chọn **Analyses > Choose**

Bảng các chế độ phân tích xuất hiện cho người thiết kế lựa chọn như hình dưới:



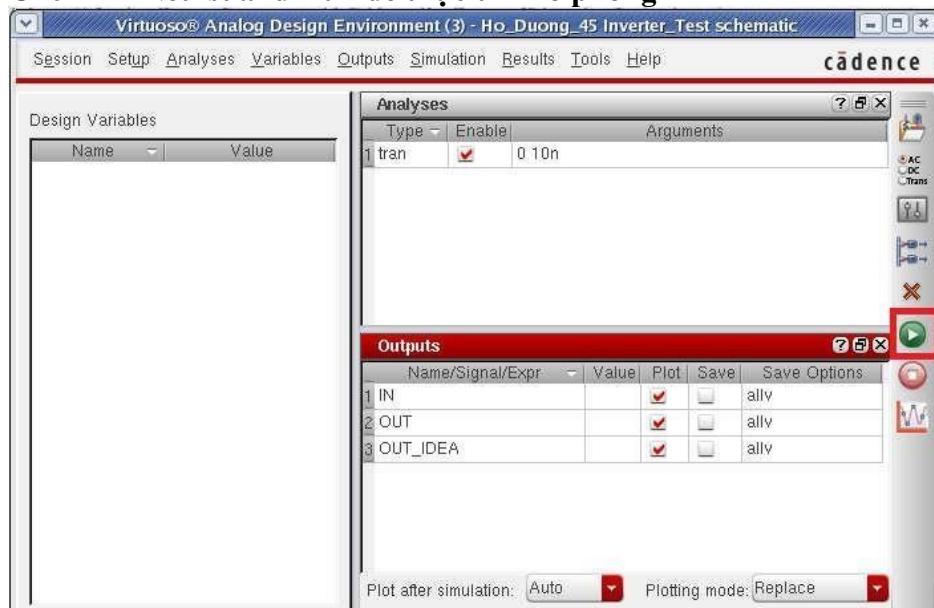
Những chế độ phân tích được chi tiết hóa như sau:

 DC	<p>Phân Tích DC tương tự như phân tích SPICE, người thiết kế có thể kiểm tra hoạt động trong miền DC của mạch có thể kiểm tra điểm hoạt động của từng linh kiện trong mạch một cách chính xác. Ví dụ: Kiểm tra xem tất cả các Transistor trong mạch đã hoàn toàn đi vào vùng hoạt động mong muốn hay chưa? Trong đó, người thiết kế có thể mô phỏng tương quan giữa những thông số để tìm ra điểm hoạt động tối ưu cho mạch như tìm dòng phân cực cho mạch khuếch đại sao cho toàn bộ các Transistor trong mạch đi vào vùng bão hòa sâu nhằm đạt được độ lợi tín hiệu cao.</p>
 Transient	<p>Chế độ phân tích đáp ứng của mạch trong miền thời gian. Phương pháp phân tích này được dùng để phân tích những thay đổi của dạng tín hiệu khi chịu tác động của nhiễu hoặc nén tín hiệu. Ví dụ, phân tích méo hài. Mà những thay đổi này không thể thực hiện mô phỏng bằng phân tích AC.</p>

- Chọn dáng sóng ngõ ra:
Chọn **Outputs > To be plotted > Select on schematic**

Màn hình schematic xuất hiện, click chọn các net muốn đo điện áp (ví dụ: net IN, OUT, OUT_idea) > nhấn ESC

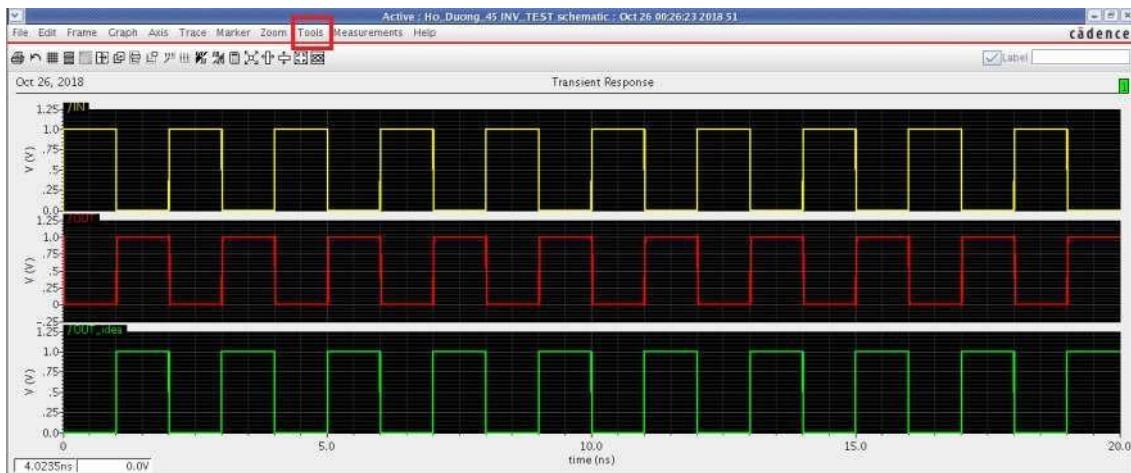
Click Netlist and Run để thực thi mô phỏng



Lưu ý:

Ở các lần mô phỏng sau, để tránh lỗi khi mô phỏng, mỗi khi thay đổi trên schematic, bắt buộc phải xóa các dạng song ngõ ra đã chọn và thực hiện chọn lại. (ví dụ: xóa và chọn lại In, Out, Out_idea)

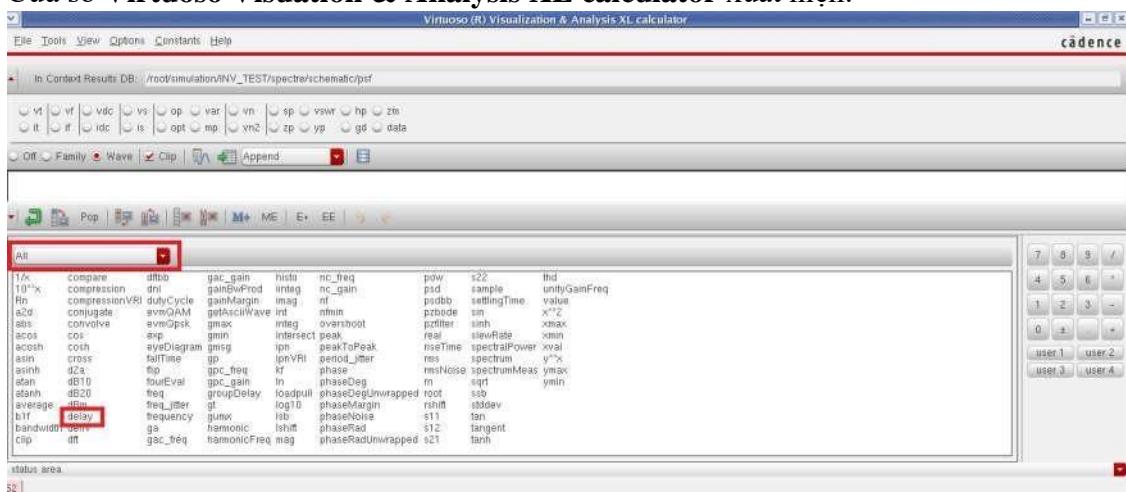
Kết quả mô phỏng được xuất thành dạng sóng quan sát bởi trình Virtuoso (R) Visualizations & Analysis XL như hình dưới.



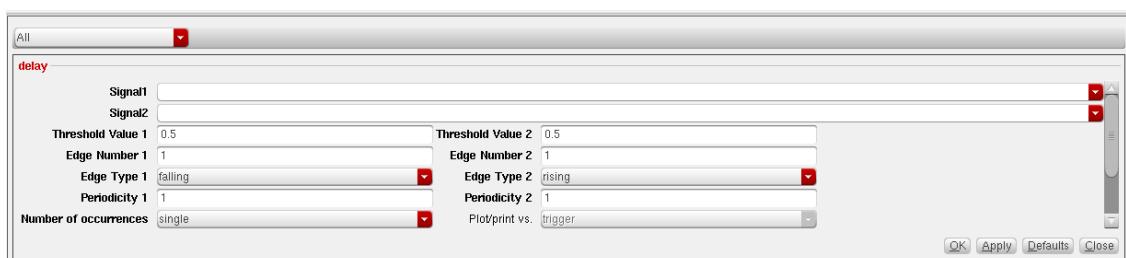
Hướng dẫn đo delay:

Chọn Tool >> Calculator

Cửa sổ Virtuoso Visualisation & Analysis XL calculator xuất hiện:



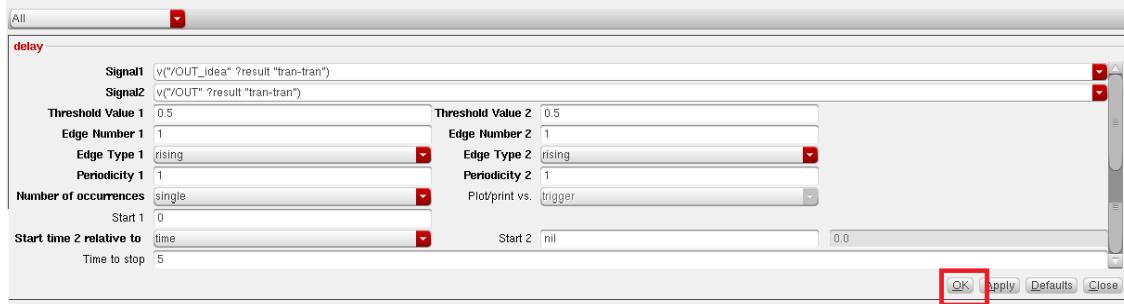
Click chọn All và Delay, cửa sổ để calculate Delay hiện ra:



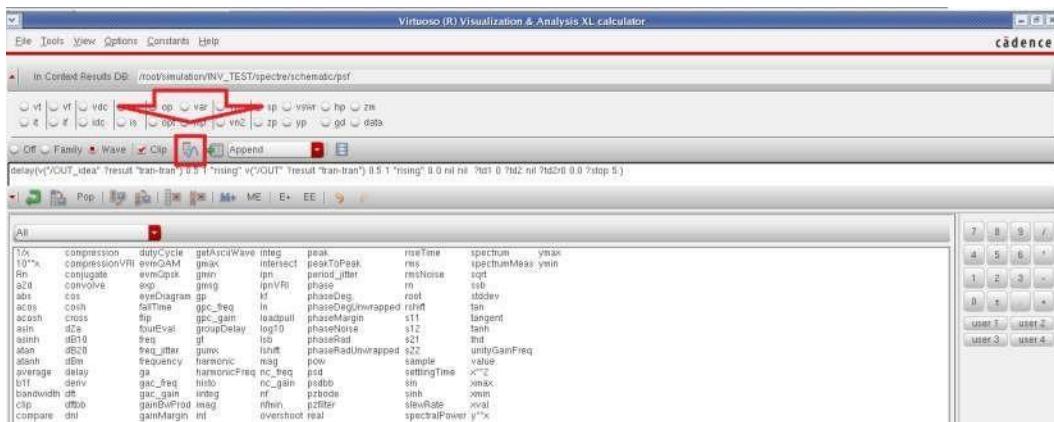
Set Signal1 & Signal2:

- Click vào ô Signal 1, chuyển sang tab chứa dạng sóng vừa mô phỏng, click chuột vào dạng sóng cần thêm (ví dụ Output_idea)
- Thực hiện tương tự để set Signal 2 (ví dụ Output)

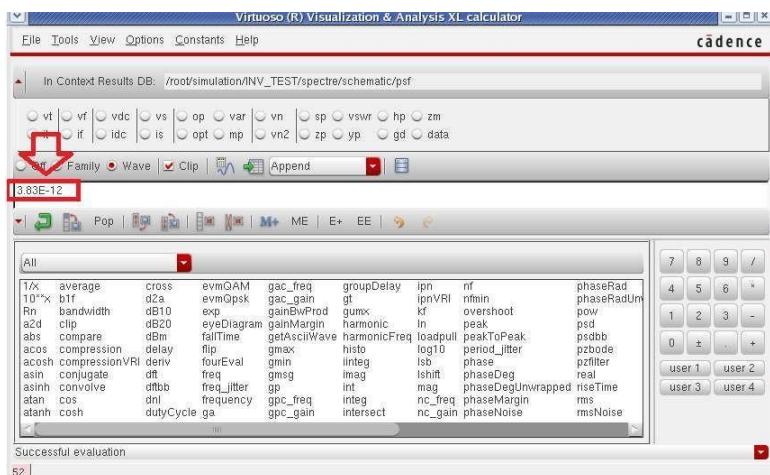
Cài đặt các thông số còn lại như sau:



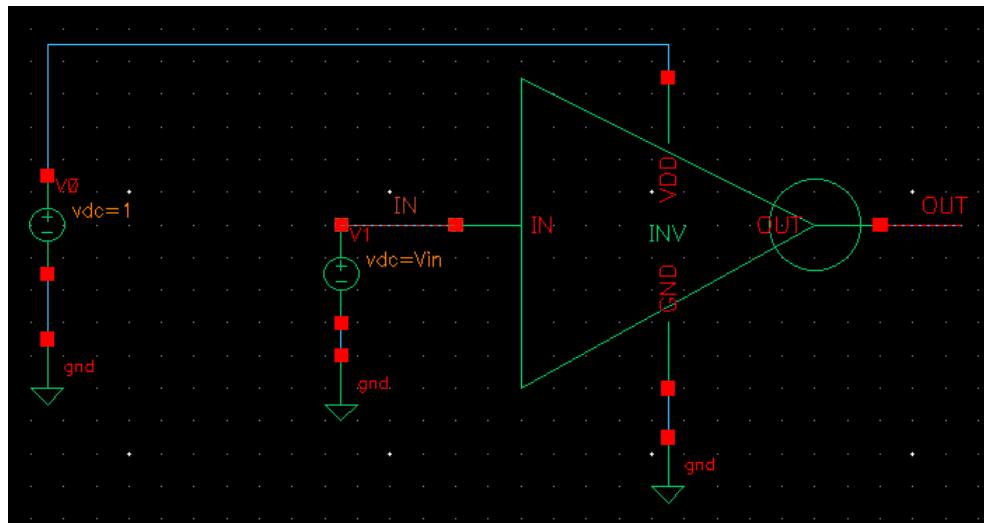
>> Chọn OK và chọn để tool thực hiện tính toán.



Kết quả tính toán được hiển thị :

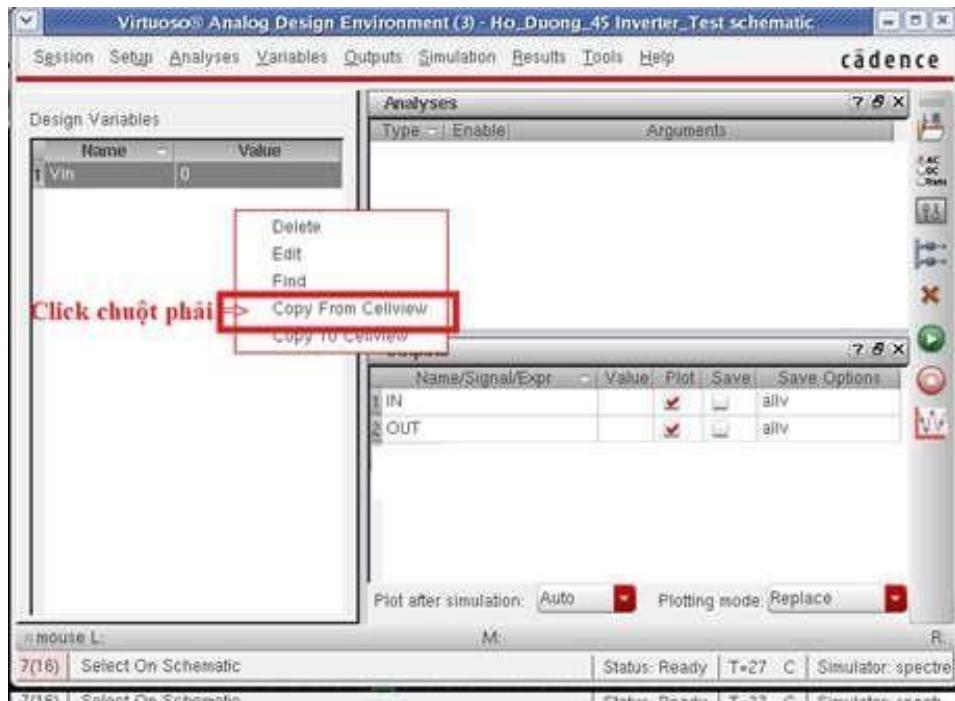


c) Phân tích DC cho công Inverter



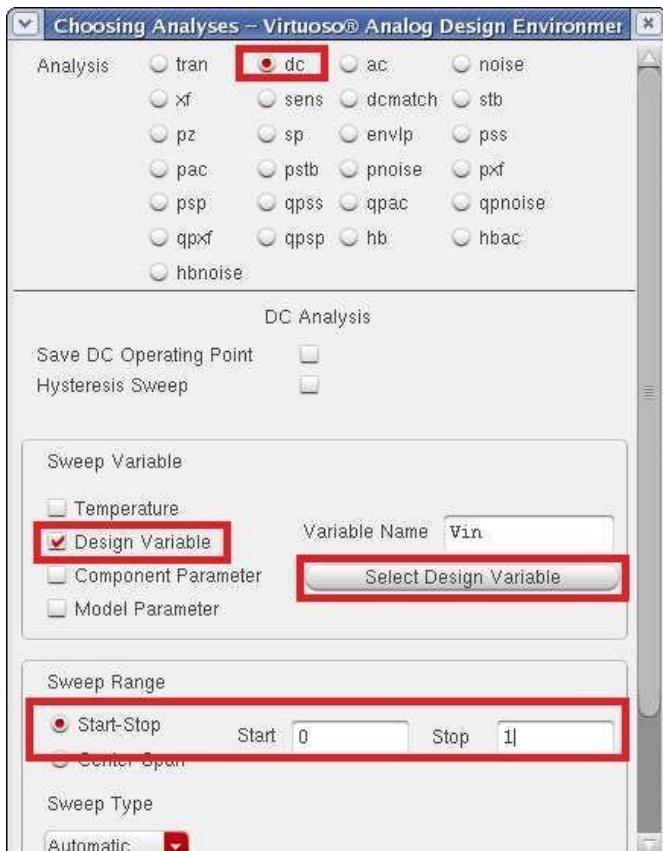
Sơ đồ Schematic mô phỏng DC

- Đặt biến Vin là điện áp ngõ vào
- Cài đặt giá trị đầu bất kỳ cho biến Vin =0 (mục đích: tránh lỗi khi mô phỏng)



Cài đặt DC analyses:

Chọn **Analyses > Choose** và cài đặt như sau:



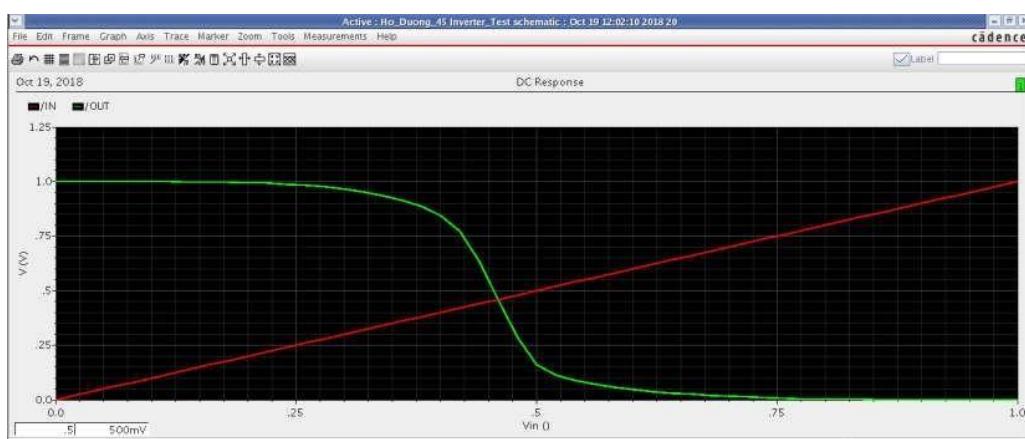
Chọn dáng sóng ngõ ra:

Chọn **Outputs > To be plotted > Select on schematic**

Màn hình schematic xuất hiện, click chọn các net muốn đo điện áp (ví dụ: net IN, OUT, OUT_idea) sau đó nhấn ESC

Click Netlist and Run để thực thi mô phỏng

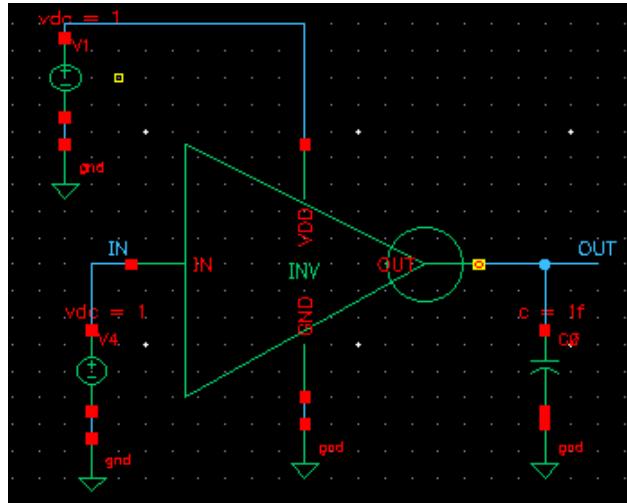
Kết quả mô phỏng được xuất thành dạng sóng quan sát bởi trình Virtuoso (R) Visualization & Analysis XL như hình dưới.



❖ ĐO STATIC, DYNAMIC, TOTAL POWER CONSUMPTION

$$\begin{aligned} \text{Dynamic Power} &= CL * VDD^2 * freq \\ \text{Static Power} &= I_{stat} * VDD \\ \text{Total Power} &= \text{Dynamic Power} + \text{Static Power} \end{aligned}$$

a. Đo Static Power:



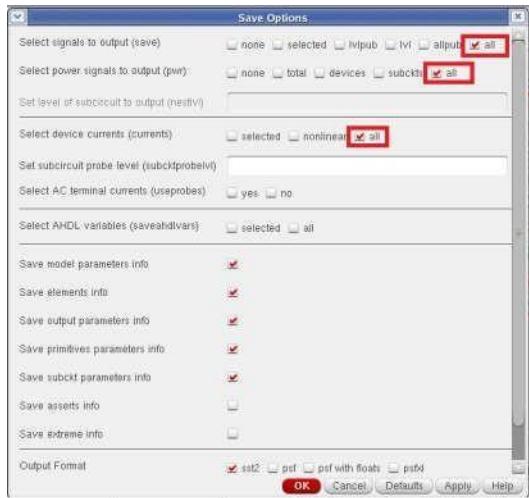
Sơ đồ schematic

a.1. Đo Static Power với Vin = 0 (dựa trên Pmos Device)

Tương tự cài đặt Model Library như trên và loại mô phỏng là DC như sau:



Trên cửa sổ ADEL, thực hiện chọn **Output >> Save All**, cửa sổ **Save Option** xuất hiện, cài đặt như sau và nhấn **OK**



Tiến hành chạy mô phỏng (chọn netlist and Run

Sau khi mô phỏng thành công, thực hiện chọn **Tool > Result Browser**

Cửa sổ sau xuất hiện, click chuột vào **dcOpInfo-info >> V1**(giá trị của nguồn DC) để xem kết quả



$$P_{\text{stat(PMOS)}} = 25,2028 \text{ nW}$$

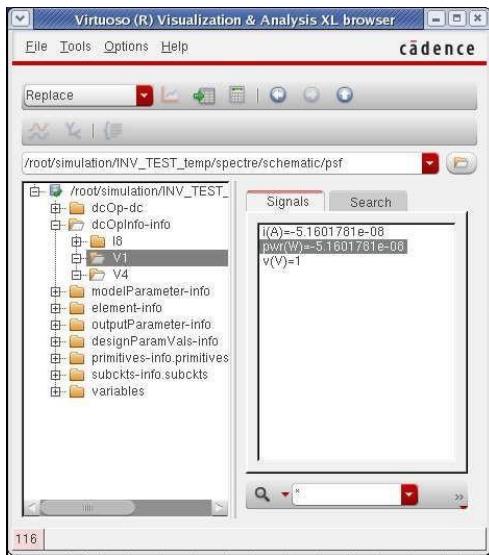
Có thể kiểm chứng lại kết quả bằng:

$$\text{Static Power} = I_{\text{sat}} * \text{VDD} = 2.52028 \times 10^{-8} \text{ A} \times 1\text{V} = 25,2028 \text{ nW}$$

a.2. Đo Static Power với Vin = 1V (dựa trên Nmos Device)

Thực hiện tương tự như đo $P_{stat}(PMOS)$.

Kết quả mô phỏng:



Vậy $P_{stat(NMOS)} = 51.6 \text{ nW}$

Có thể kiểm chứng lại kết quả bằng:

$$\boxed{\text{Static Power} = I_{sat} * VDD}$$

b. Đo Total Power consumption:

Các thông số cần thiết:

Thông số	Giá trị
V_{dd}	1 V
C_{load}	400 fF
Voltage 1	0 V
Voltage 2	1 V
Rise time	400 ps
Fall time	400 ps
Delay	0 ns
Pulse width	10 ns
Period	20ns

- Cài đặt Transient Analysis 1 chu kỳ (Stop time = 20ns)



- Cài đặt DC Analysis:



Trên cửa sổ ADEL, thực hiện chọn **Output > Save All**, cửa sổ **Save Option** xuất hiện, cài đặt như sau và click **OK**

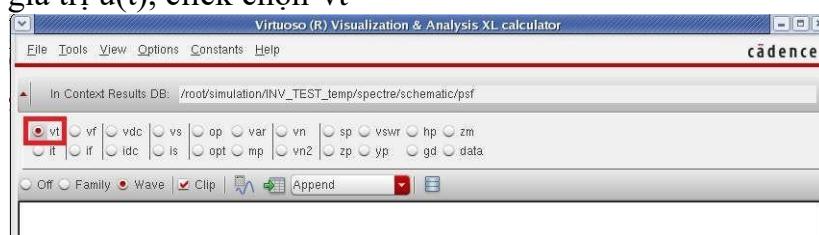


Tiến hành chạy mô phỏng (netlist and Run

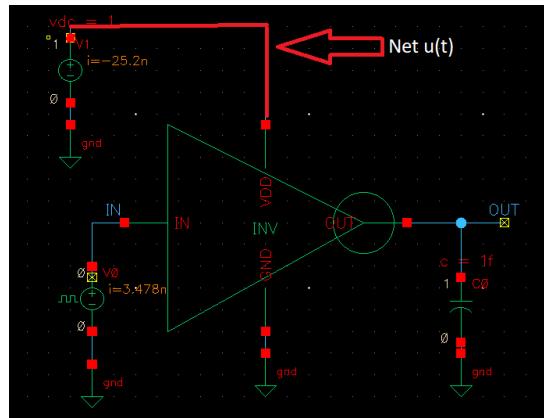
Chọn **Tool > Calculator**

$$\text{Total Power consumption} = \text{Average}[v(t) * i(t)]$$

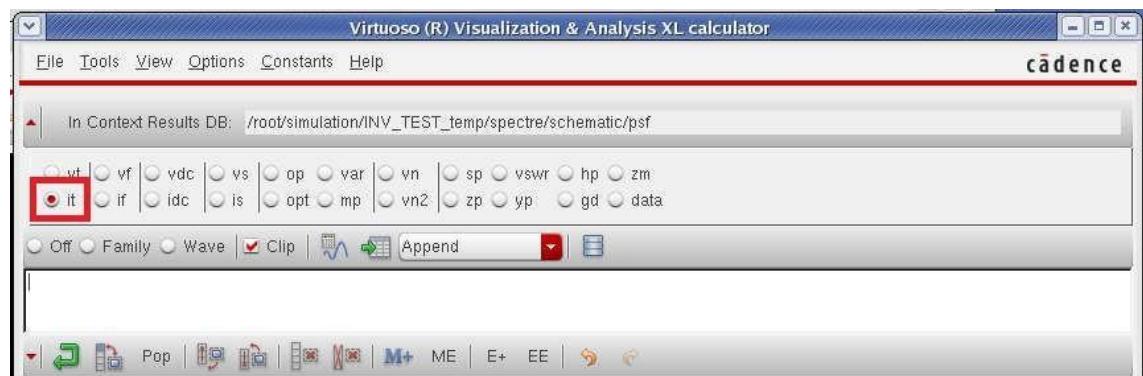
- Để lấy giá trị $u(t)$, click chọn vt



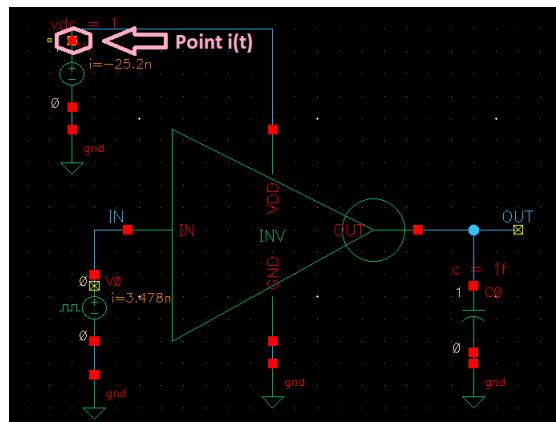
Click vào net $u(t)$



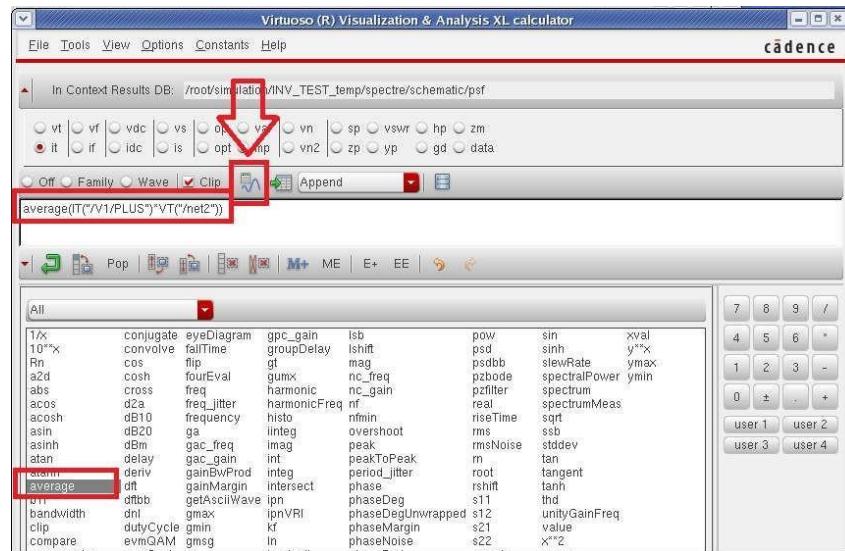
- Để lấy giá trị $i(t)$, click chọn it



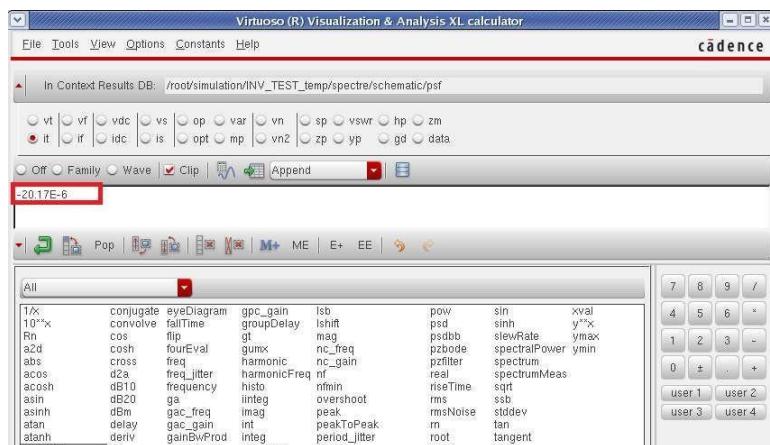
Click vào point $i(t)$



Tiếp tục chọn chức năng **Average** để lấy giá trị trung bình



Click để tool thực hiện tính toán.



Kết quả: Total Power consumption = 20.17 uW

c. Tính toán Dynamic power

$$\text{Dynamic Power} = \text{Total Power} - \text{Static Power}$$

$$= 20.17 \text{ uW} - 25,2028 \text{ nW}$$

$$= 0.2 \text{ uW}$$

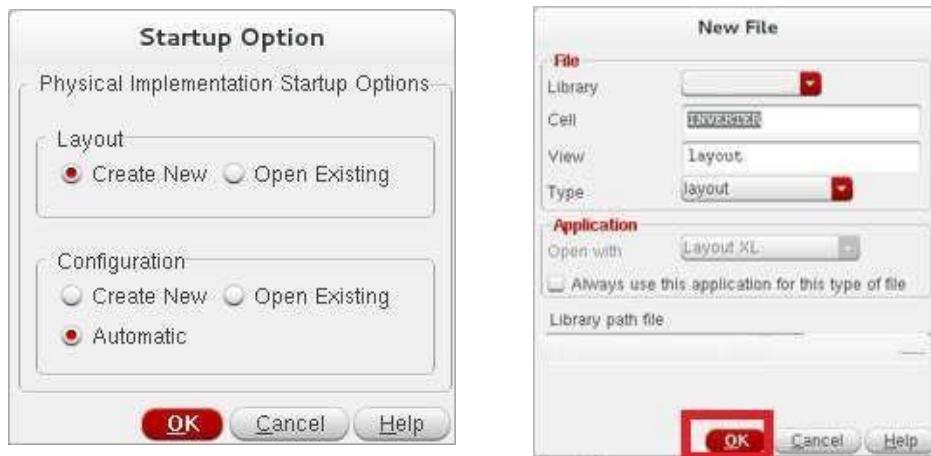
Ta có thể kiểm chứng lại kết quả bằng:

$$\text{Dynamic Power} = CL * VDD^2 * freq$$

5. THỰC HIỆN THIẾT KẾ VẬT LÝ

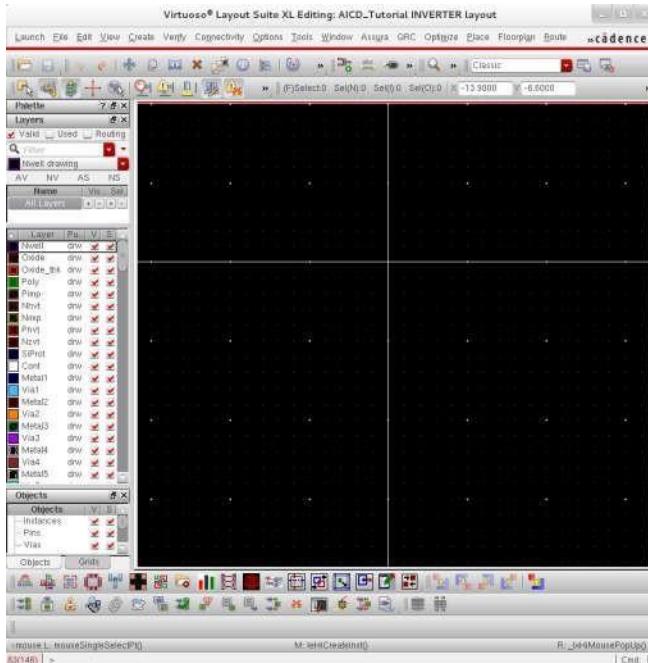
5.1 Các Bước vẽ Layout

Đầu tiên, từ cửa sổ mạch nguyên lý thực hiện chọn **Launch > Layout XL**



Bảng hiển thị yêu cầu lựa chọn khởi động trình thiết kế vật lý hiện ra như hình trên. Thực hiện tạo mới toàn bộ thiết kế chọn **Create New** và để chế độ tạo tự động cài đặt trong **Configuration**. Cửa sổ tạo thiết kế bật lý hiện ra như Hình bên cạnh.

Thực hiện chọn OK, màn hình Virtuoso Layout Suite XL Editing xuất hiện như hình dưới.



Thực hiện chuyển thiết kế từ mạch nguyên lý sang môi trường thiết kế vật lý.
Chọn **Connectivity > Generate > All from source**

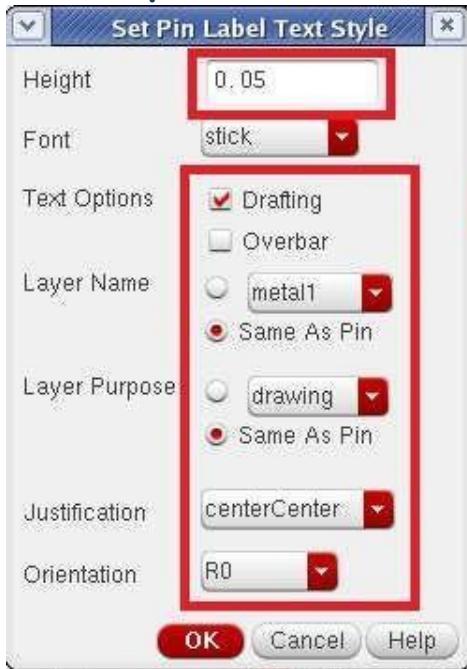


- Instance: phần tử linh kiện có trong mạch.
- I/O pin: Chân ngõ vào và ngõ ra.
- PR Boundary: Biên thiết kế.

Tiếp tục thực hiện cài đặt I/O pin như sau:



Thực hiện chọn lớp cho các chân linh kiện. Tạo nhãn tên cho chân linh kiện nhằm phân biệt dễ dàng hơn trong quá trình thực hiện vẽ Layout cho mạch và khi run LVS
 Chọn **Options**: > **Same As Pin** > chọn **OK**



Để dễ dàng hơn trong vẽ Layout ta cài đặt Display Options như sau:

Options > Display



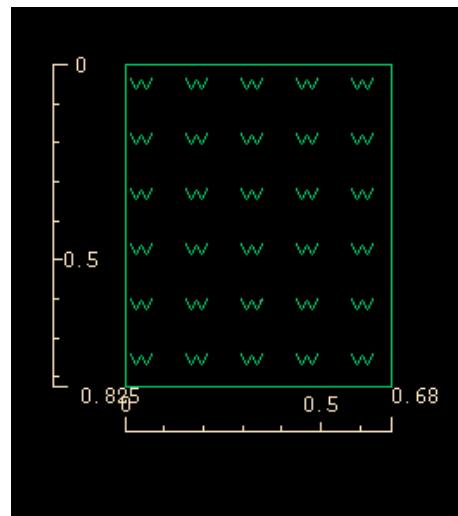
Các phím tắt cần thiết để vẽ Layout linh hoạt hơn

HotKey	Function
K	Create Ruler
Shift + K	Remove all rulers
R	Rectangle
Shift + P	Polygon
S	Stretch
M	Move
C	Copy
P	Path
Shift + M	Merge
Shift + C	Chop
I	Instance Cell
O	Contact
L	Label
Ctrl + S	Split
U	Undo
F	Fit all

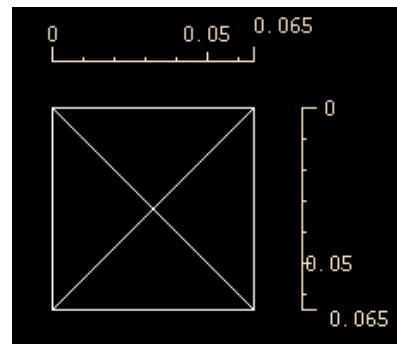
Bước 1: Thực hiện thiết kế PMOS

PMOS không xét cực B được tạo thành từ các lớp: DIFF, PIMP, POLY1, NWELL, CONT, METAL1. Thực hiện vẽ PMOS theo các bước sau:

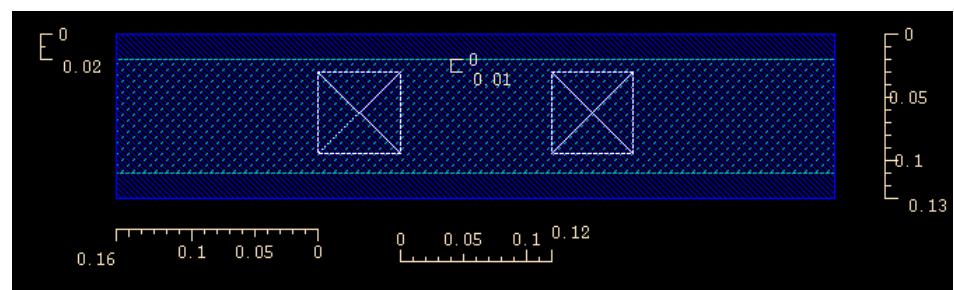
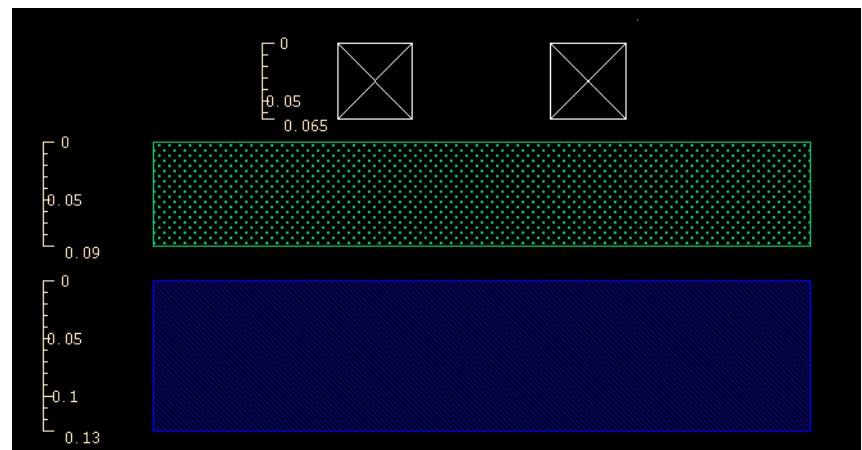
a) Vẽ lớp NWELL



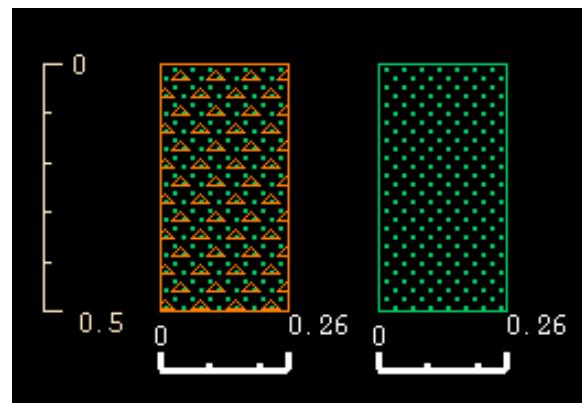
b) Vẽ lớp CONT



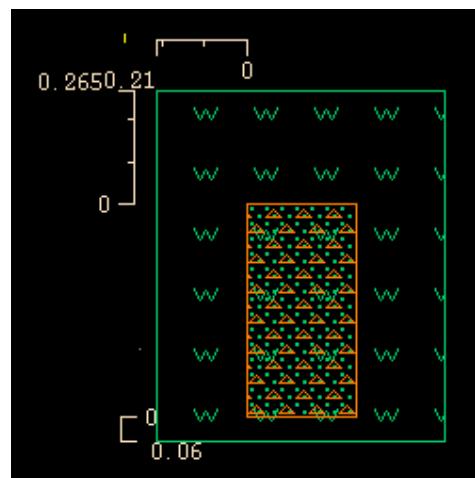
c) Vẽ đường VDD cho mạch



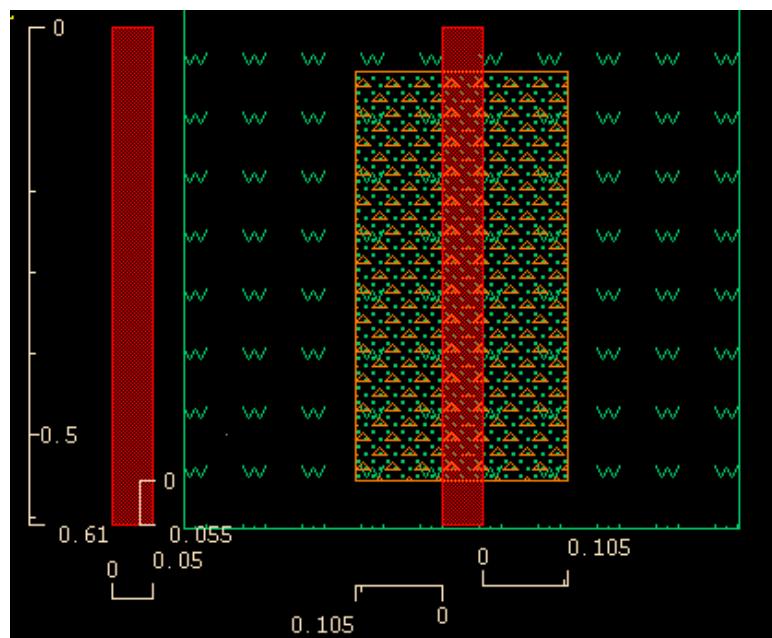
d) Vẽ lớp Pimplant và lớp active



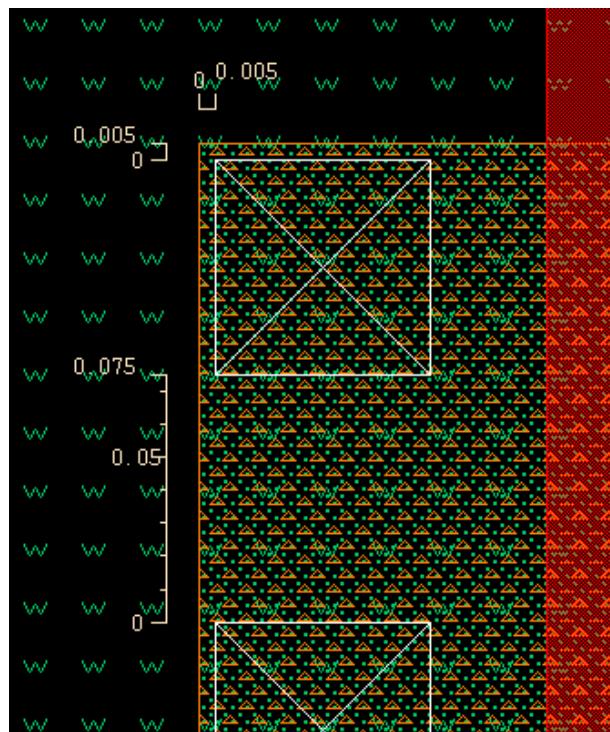
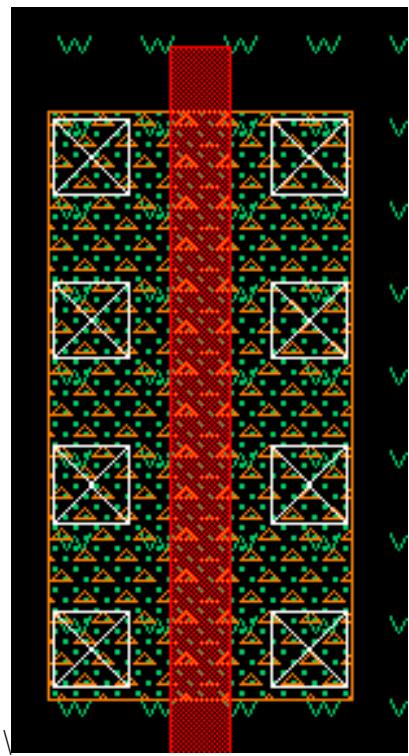
e) Đặt hai lớp Pimplant và Active chồng lên nhau



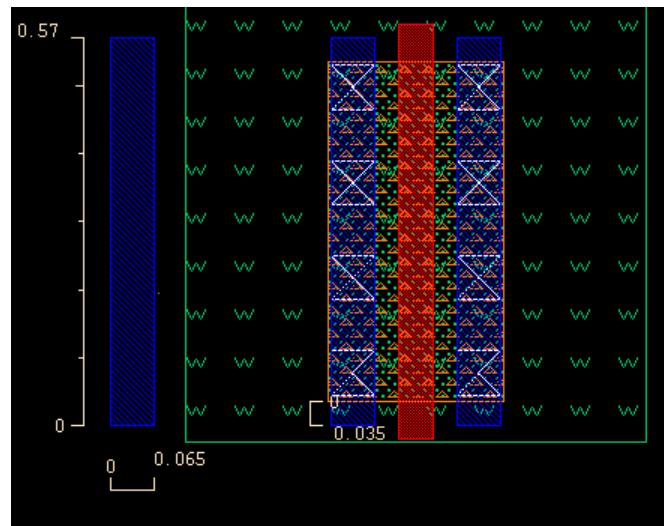
f) Vẽ lớp POLY



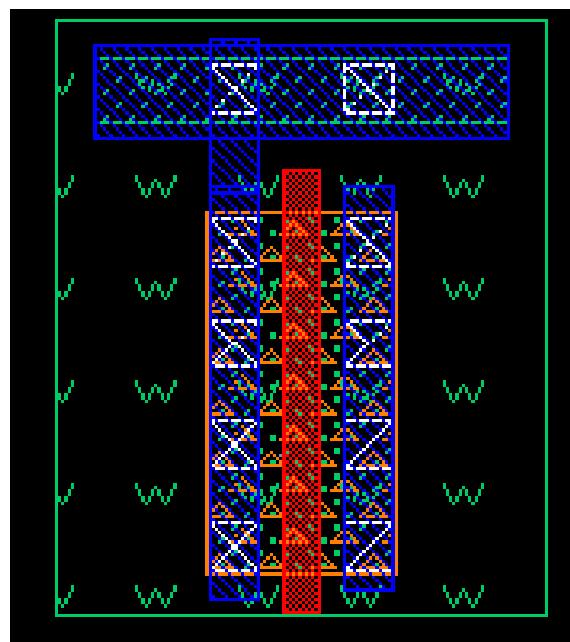
g) Vẽ llop CONT cho Pmos



h) Vẽ lớp Metal1



i) Hình ảnh dưới đây minh họa phần nửa trên của INV sau khi hoàn thành

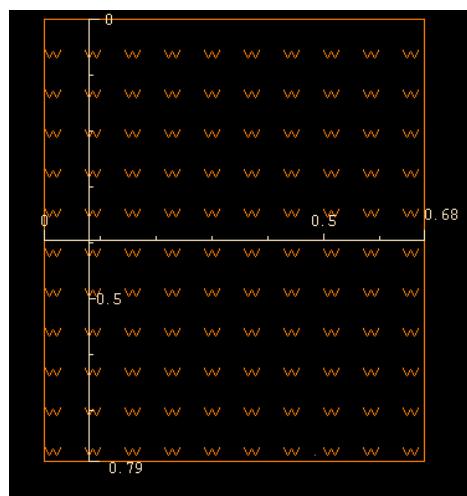


Bước 2: Thực hiện thiết kế NMOS

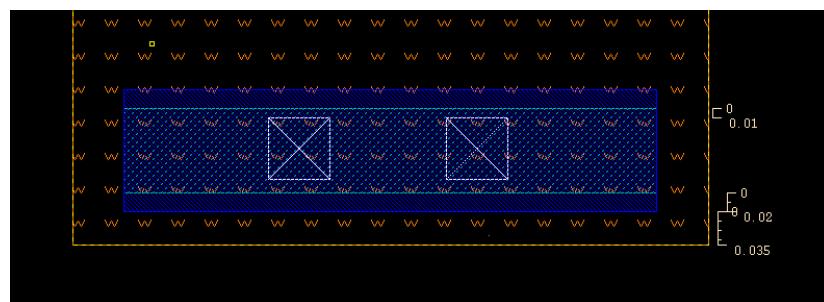
NMOS không xét cực B được tạo thành từ các lớp: DIFF, NIMP, POLY1, CONT, METAL1. Tùy vào PDK sử dụng cho việc layout mà có các định nghĩa về tên của các lớp khác nhau.

Thực hiện vẽ NMOS theo các bước sau:

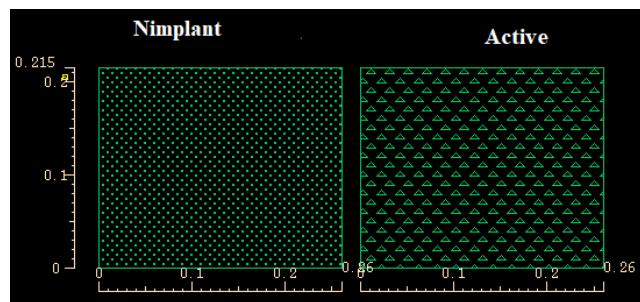
- Vẽ lớp Pwell



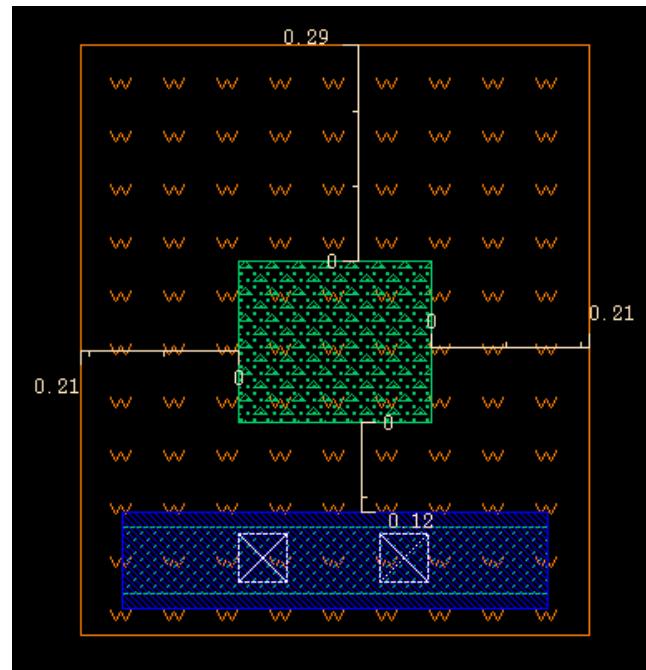
- Vẽ đường GND tương tự VDD



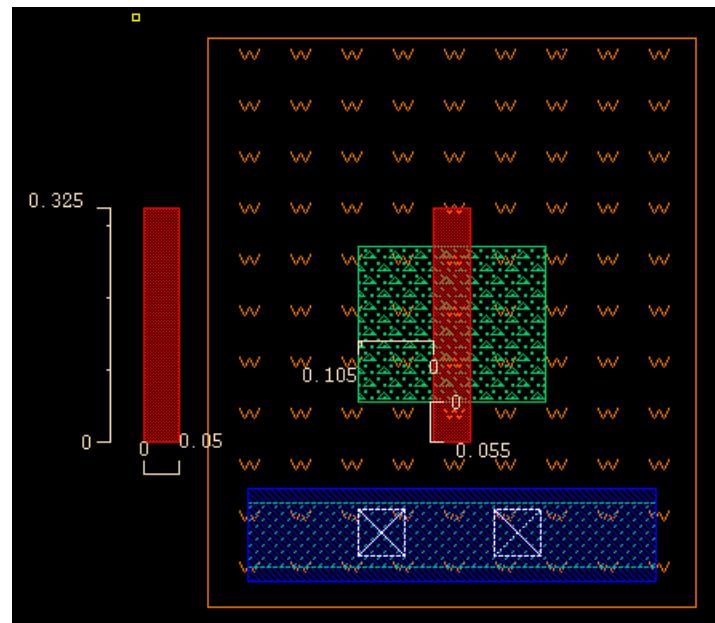
- Vẽ lớp Active và Nimplant



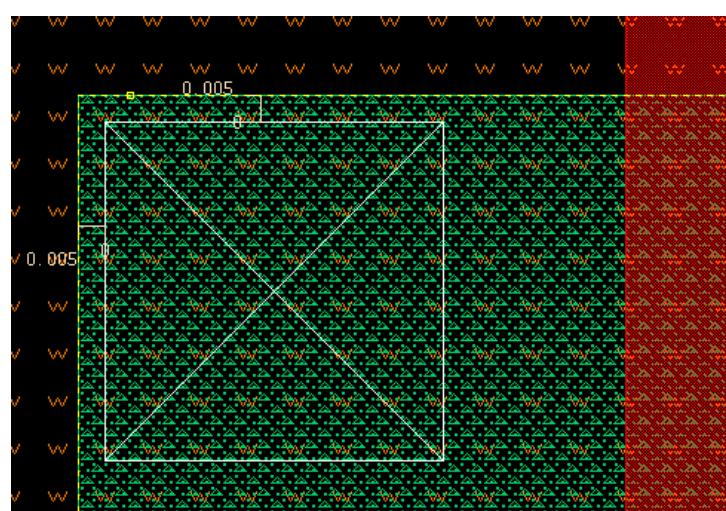
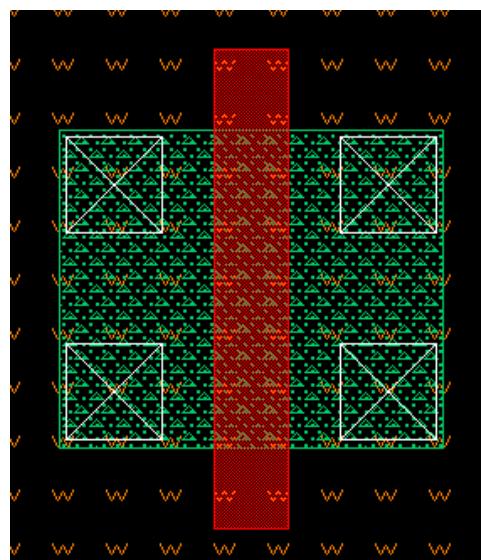
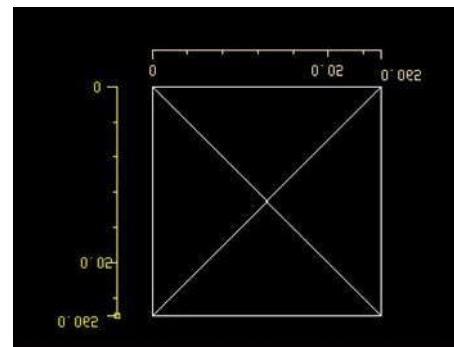
d) Xếp hai lớp Nimplant và Active chồng lên nhau



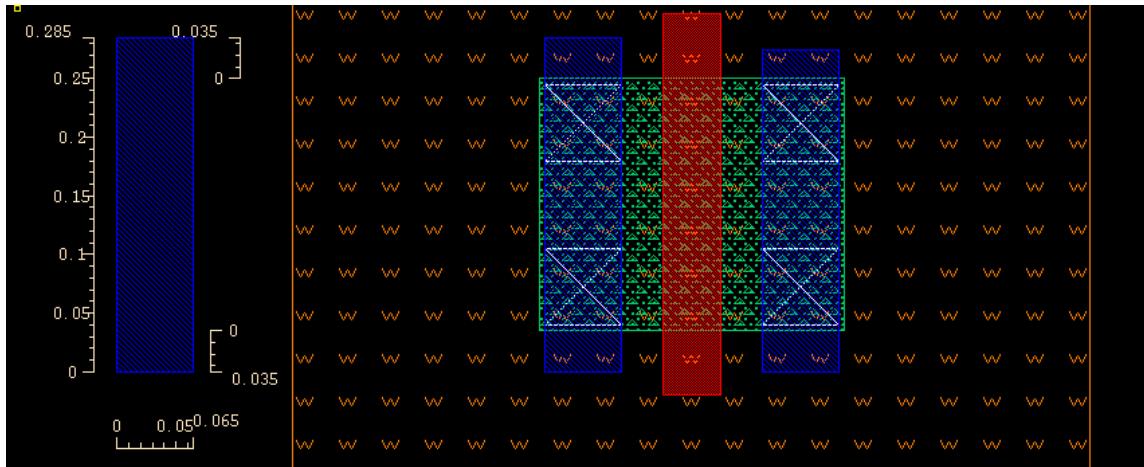
e) Vẽ lớp POLY1



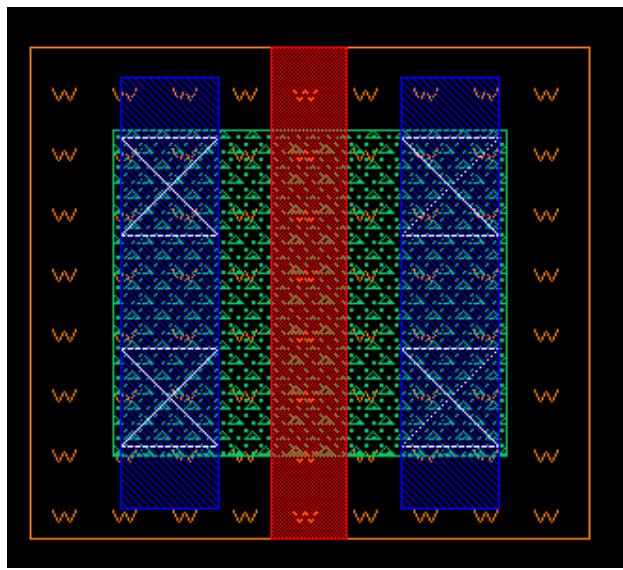
f) Vẽ lớp CONT cho Nmos



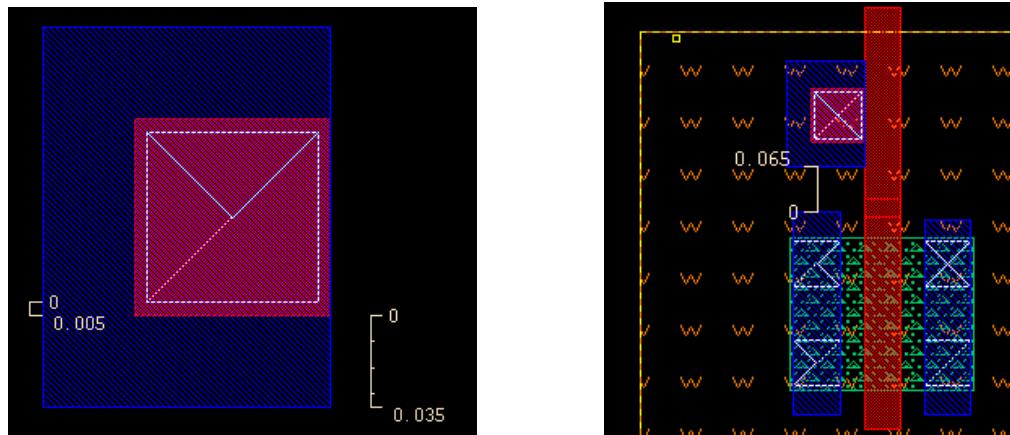
g) Vẽ lớp Metal 1



Hình ảnh dưới đây minh họa NMOS sau khi được thiết kế hoàn thành



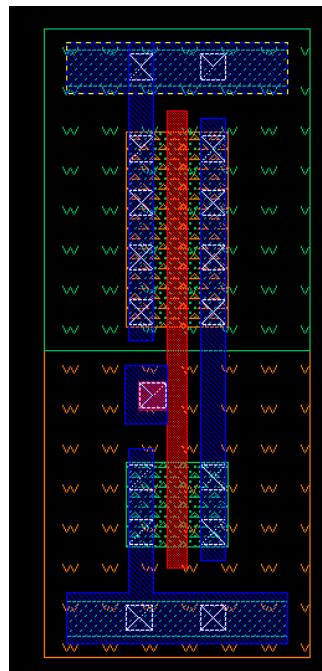
h. Tạo Via cho kết nối giữa lớp Poly & Metal1 (mục đích phục vụ cho Port In)



Lưu ý: Để tạo VIA này các bạn chỉ cần dùng 3 lớp CONT, POLY và METAL1 xếp chồng lên nhau. Không được sử dụng via có sẵn trong thư viện.

Bước 3: Sắp xếp và đi dây

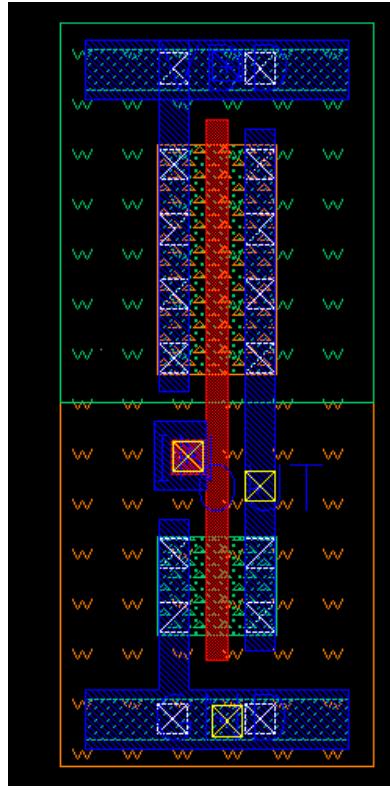
Tiếp theo, ta thực hiện sắp xếp PMOS và NMOS bên trong không gian giữa hai đường nguồn và đất. PMOS được đặt ở gần nguồn và NMOS được đặt ở gần đất. Thực hiện nối hai cực G của NMOS và PMOS lại với nhau bằng cách sử dụng lớp POLY1, cực D của PMOS nối với cực D của NMOS sử dụng lớp METAL1, cực S của PMOS nối lên nguồn và cực S của NMOS nối xuống đất nhờ sử dụng lớp METAL1.



Bước 4: Đặt các PIN I/O, nguồn, đất

Sau khi các bước trên đã được thực hiện xong, chúng ta đặt các PIN ngõ vào, ngõ ra, nguồn, đất giống như bên sơ đồ mạch nguyên lý của cỗng.

Cuối cùng điều chỉnh lại độ rộng của thiết kế layout sao cho giá trị này là nhỏ nhất có thể thực hiện được. Kết quả ta có thiết kế hoàn chỉnh của cỗng NOT như hình bên dưới:



Một số chú ý khi thực hiện layout các mạch phức tạp đó là:

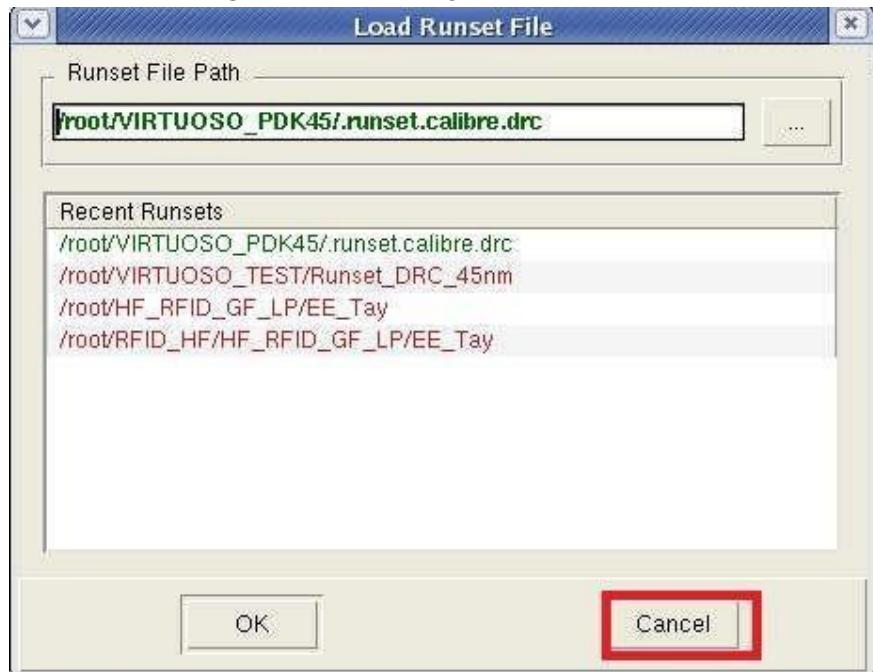
Đường POLY1 kết nối hai cực G của PMOS và NMOS lại với nhau phải thẳng từ trên xuống dưới, không được đi gấp khúc để tránh hiện tượng IR – hiện tượng rót áp một lượng ΔV khi qua mỗi gấp khúc.

Ở những thiết kế phức tạp nên sử dụng thêm lớp METAL2 bởi vì tụ điện ký sinh giữa lớp POLY1 với lớp này nhỏ hơn giữa lớp POLY1 với lớp METAL1, dẫn đến giảm ký sinh cho mạch và việc thực hiện mô phỏng Post layout sẽ chính xác hơn.

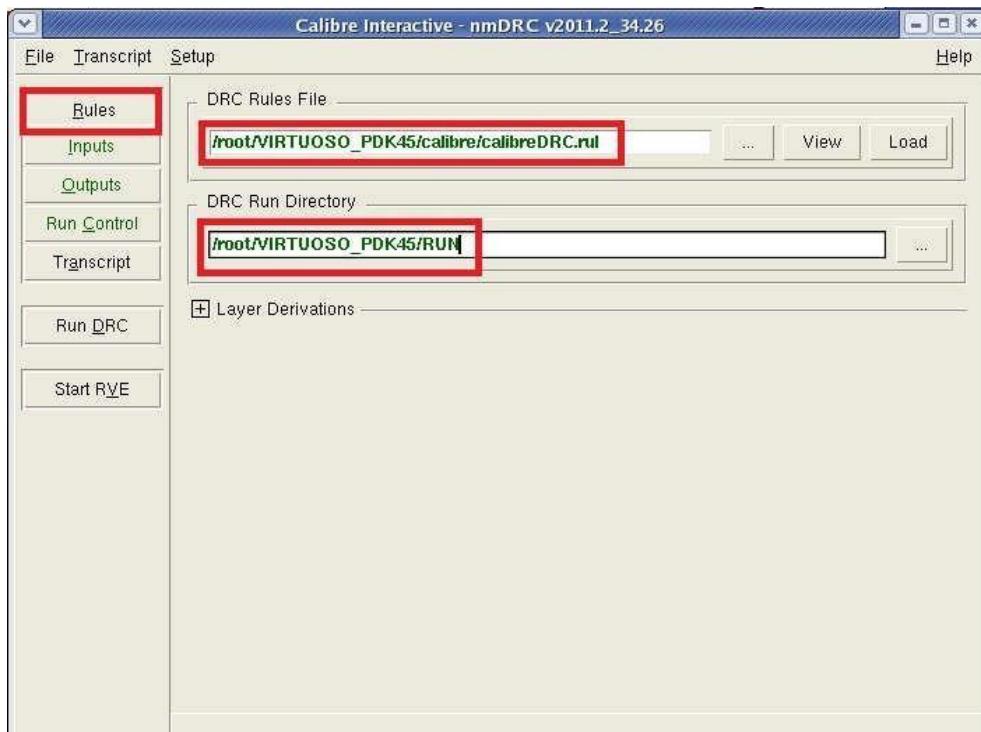
4.2. DESIGN RULE CHECK (DRC)

Chọn Calibre >> DRC

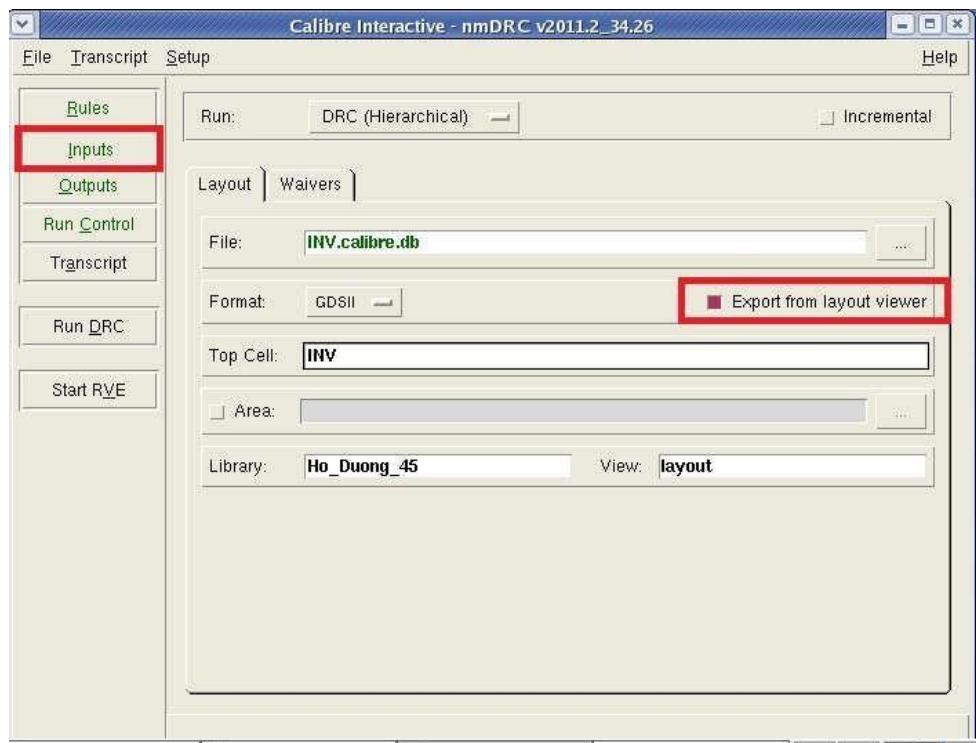
- Cửa sổ Calibre Interactive-nmDRC xuất hiện.
- Bởi vì là lần run DRC đầu tiên nên có thể chọn “cancel” khi được hỏi về “runset file”. Một runset file sẽ lưu tất cả những cài đặt của chúng ta về run DRC.



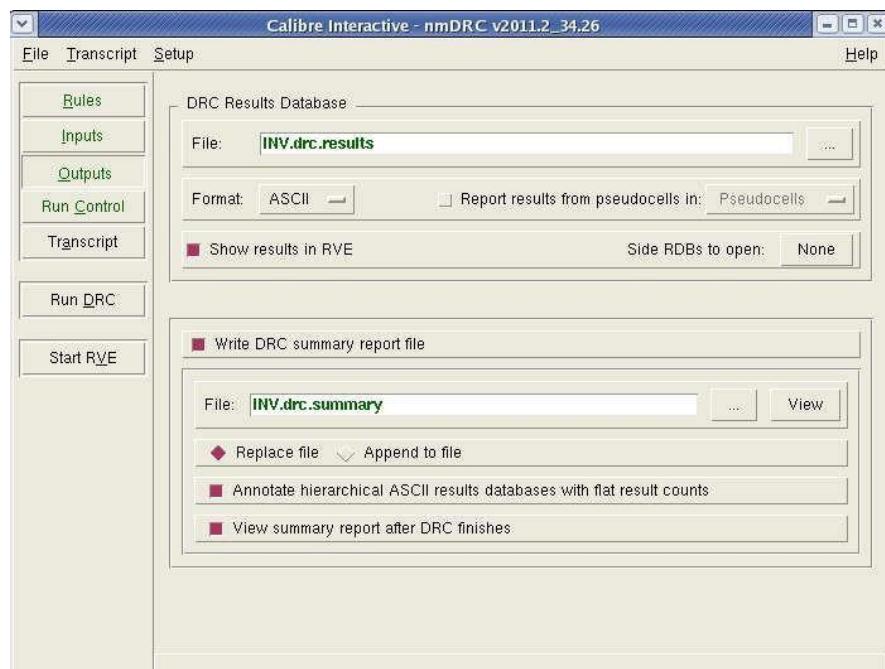
Chọn Rules tab ở bên trái và chọn đường dẫn tuyệt đối đến file Rule ở mục DRC Rules File.



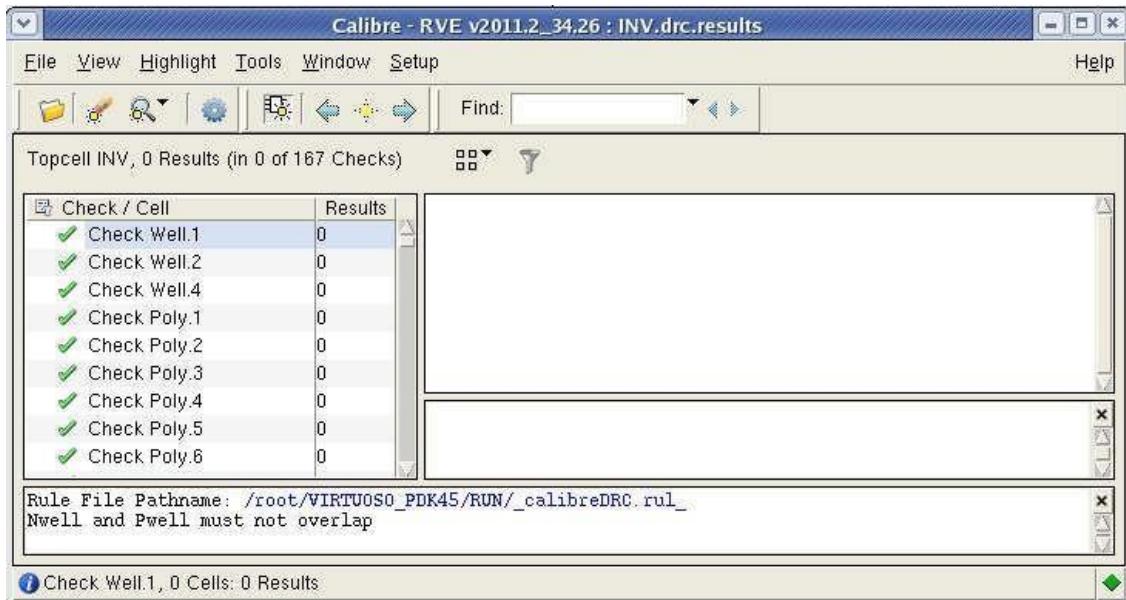
- Chọn Input tab, sau đó chọn Netlist tab ở chính giữa. Check “Export from Schematic viewer”.



- Chọn Output tab có hình ảnh như dưới đây:

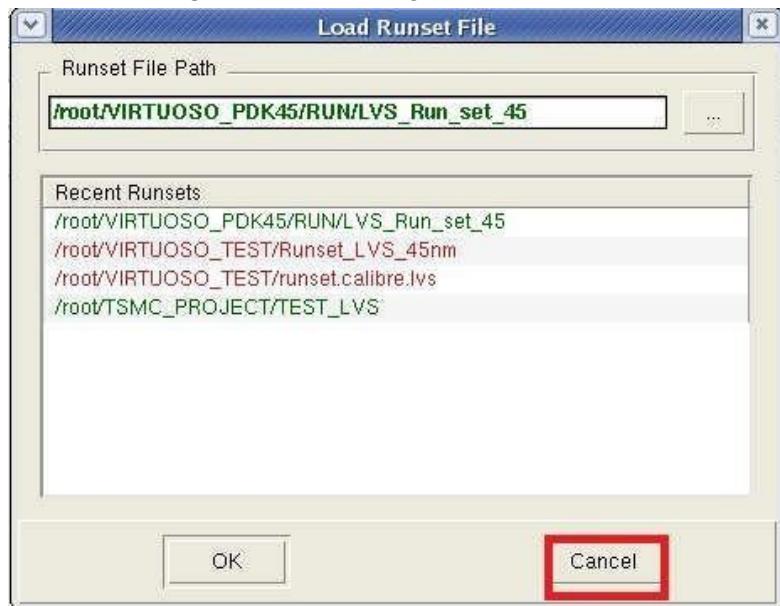


- Bây giờ DRC đã được cài đặt xong, Chọn Run DRC ở bên trái.
- Nếu gặp các cửa sổ thông báo hiện ra thì Check “Don’t show this dialog again” và chọn OK. Cửa sổ kết quả hiện ra, kiểm tra và sửa lỗi nếu có xuất hiện bất kỳ lỗi nào.



4.3. LAYOUT- VS-SCHEMATIC (LVS)

- Để mở LVS, chọn **Calibre → Run LVS**
- Cửa sổ Calibre Interactive-nmLVS xuất hiện.
- Bởi vì là lần run LVS đầu tiên nên có thể chọn “cancel” khi được hỏi về “runset file”. Một runset file sẽ lưu tất cả những cài đặt của chúng ta về run LVS.



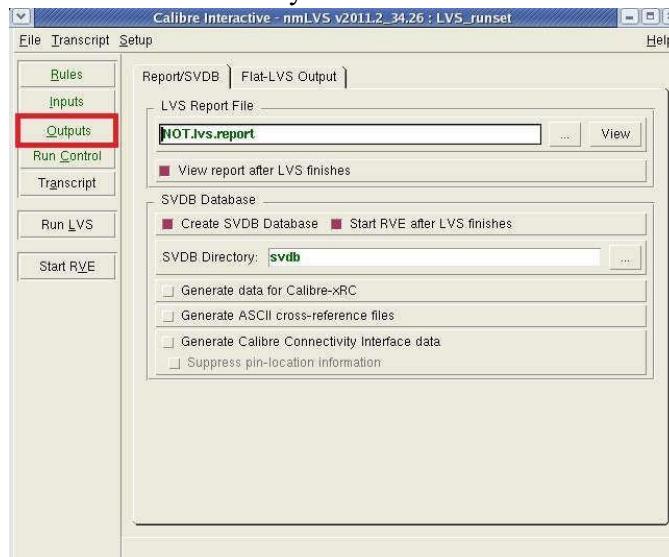
Chọn Rules tab ở bên trái và chọn đường dẫn tuyệt đối đến file Rule ở mục DRC Rules File.



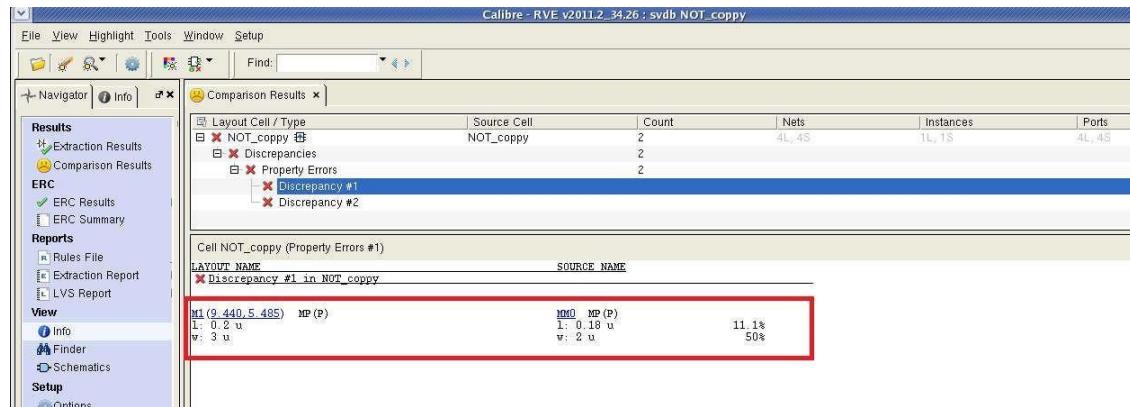
- Chọn Input tab, sau đó chọn Netlist tab ở chính giữa. Check “Export from Schematic viewer”.



- Chọn Output tab có hình ảnh như dưới đây:

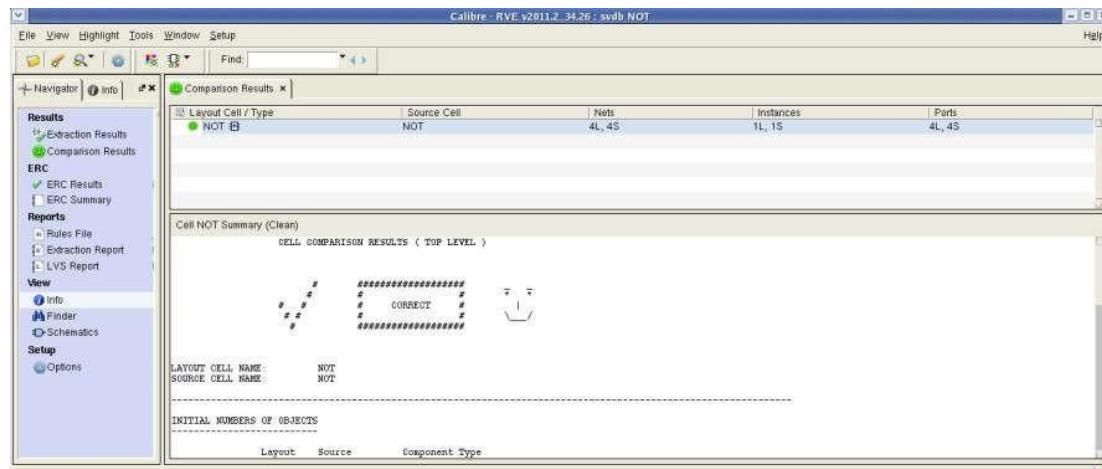


- Bây giờ LVS đã được cài đặt xong, Chọn Run LVS ở bên trái.
- Nếu gặp các cửa sổ thông báo hiện ra thì Check “Don’t show this dialog again” và chọn OK.
- Khi LVS chạy hoàn thành, cửa sổ Calibre-RVE sẽ xuất hiện như là một file LVS Report. Lỗi thường gặp:



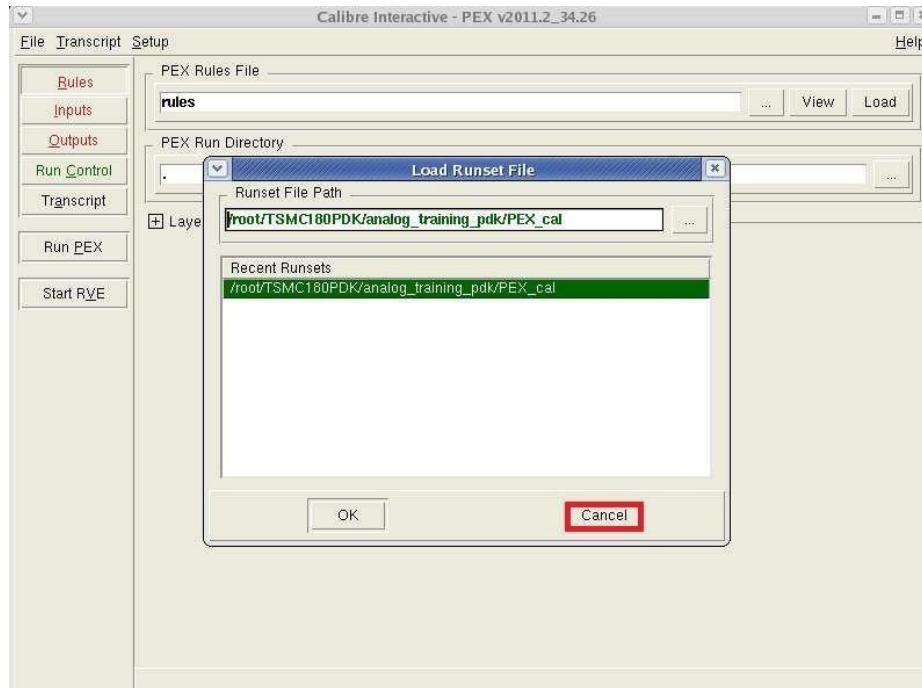
Lỗi này do Width và Length của nmos và pmos ta vừa Layout không phù hợp với thư viện nên ta phải quay lại schematic sửa lại W và L cho phù hợp.

- Nếu mọi thứ đều đúng, chúng ta sẽ thấy một Check Mark và một Happy Face giống như hình dưới đây.
- Chọn vào Schematics ở bên trái để thấy được hình ảnh schematic của layout vừa tạo ra và schematic ban đầu.

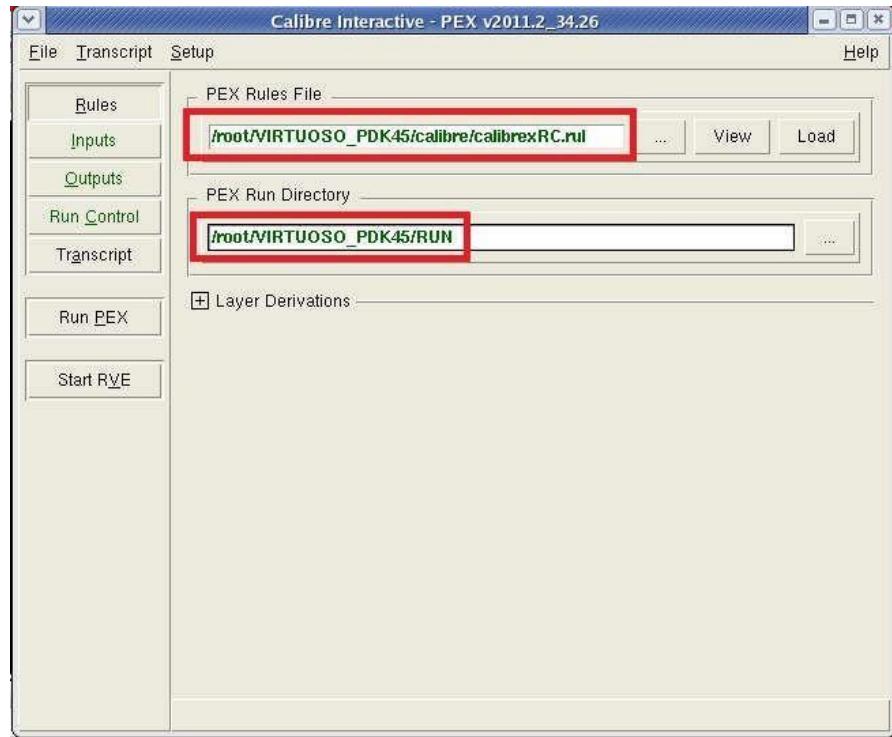


4.4. PARASITIC EXTRACTION (PEX):

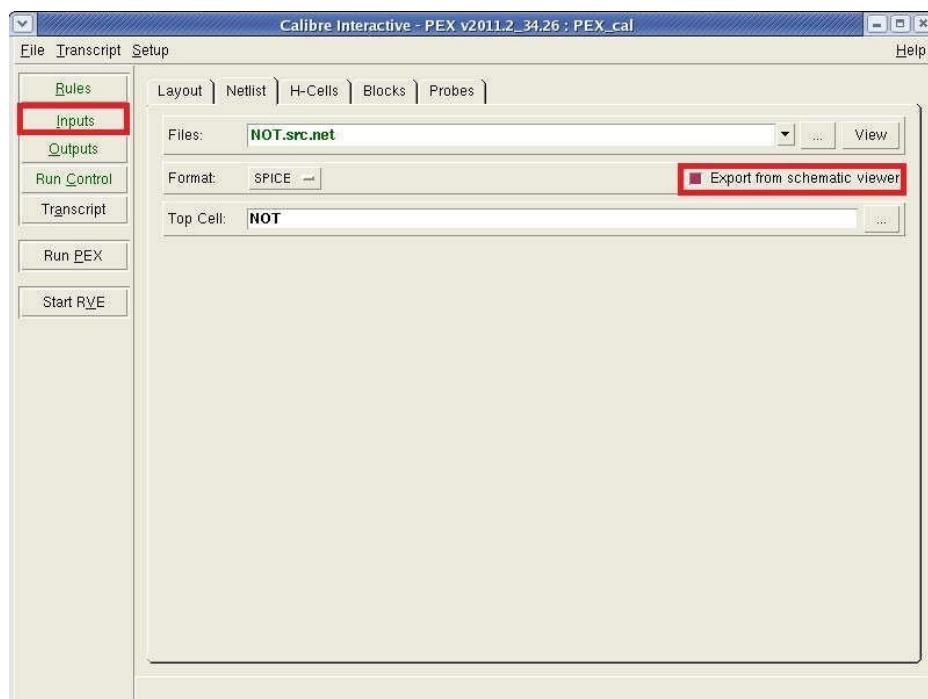
- Trước khi bắt đầu, chúng ta phải chắc chắn rằng đã run Calibre LVS một cách chính xác với 0 error và 0 warning. Một warning bất kì có thể dẫn đến sau này môn phỏng Postlayout không thực hiện được.
- Để run Calibre PEX, chọn menu **Calibre → Run PEX**.
- Bởi vì là lần run PEX đầu tiên nên có thể chọn “cancel” khi được hỏi về “runset file”. Một runset file sẽ lưu tất cả những cài đặt của chúng ta về run PEX. Ở những lần chạy sau, khi đã lưu runset file thì ta chọn OK.



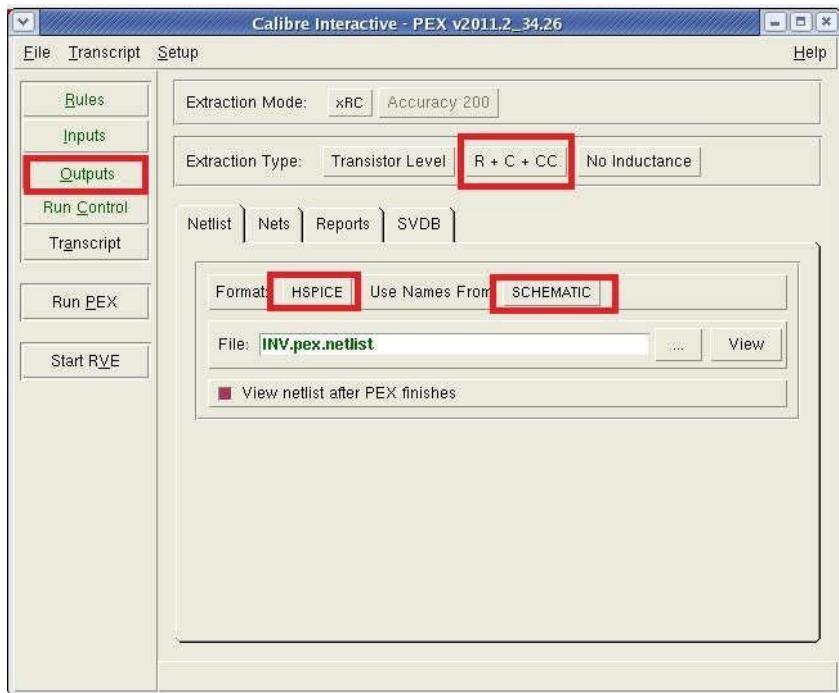
- Trên PEX form, Chọn Rules tab và tiếp theo là PEX Rules File, click vào “...”. Chọn đi đến thư mục chứa file Rules
/root/VIRTUOSO_PDK45/calibre/calibrexRC.rul
- Chọn thư mục “checkpex” vào PEX Run Directory.
- Thủ được kết quả như hình dưới đây:



- Đi đến Input tab, chọn Netlist và check “Export from schematic viewer”.



- Đi đến Output tab, thay đổi Format từ “ELDO” thành “HSPICE”. Thay đổi Use Names From thành “SCHEMATIC”. Thiết lập Extraction Type thành “R+C+CC” để trích xuất cả điện trở ký sinh và tụ điện ký sinh.



- Bây giờ lưu lại tất cả các cài đặt trên bằng cách chọn File → Save Runset. Cho vào một tên như là PEX_cal và chọn OK ở cửa sổ tiếp theo, chúng ta có thể sử dụng lại cài đặt giống như trên ở lần run PEX tiếp theo bằng cách load file runset này.
- Chọn Run PEX.

Sau khi Run PEX thành công, cửa sổ hiện thị tập tin PEX report:

- INV.pex.netlist - netlist được trích xuất từ layout cùng với tụ điện và điện trở ký sinh.

```

PEX Netlist File - INV.pex.netlist
File Edit Options Windows
* File: INV.pex.netlist
* Created: Tue Oct 23 23:12:52 2018
* Program "Calibre xRC"
* version "v2011.2_34.26"
*
.include "INV.pex.netlist.pex"
.subckt INV IN GND VDD OUT
*
* OUT OUT
* VDD VDD
* GND GND
* IN IN
MM0 N OUT MM0 d N IN MM0 g N GND MM0_s N GND MM0_b NMOS_VIL L=5e-08 W=2.15e-07
+ AD=2.2575e-14 AS=2.2575e-14 PD=6.4e-07 PS=6.4e-07
MM1 N OUT MM1 d N IN MM1 g N VDD MM1_s N VDD MM1_b PMOS_VIL L=5e-08 W=5e-07
+ AD=5.25e-14 AS=5.25e-14 PD=1.21e-06 PS=1.21e-06
*
.include "INV.pex.netlist.INV.pxi"
*
.ends
*

```

Từ file INV.pex.netlist này, chúng ta sẽ sử dụng phần mềm HSPICE (synopsis) để mô phỏng Post Layout, đo đặc delay, power,...

**THÍ
NHIỆM**

THIẾT KẾ VÀ ĐO ĐẶC CÔNG NOT

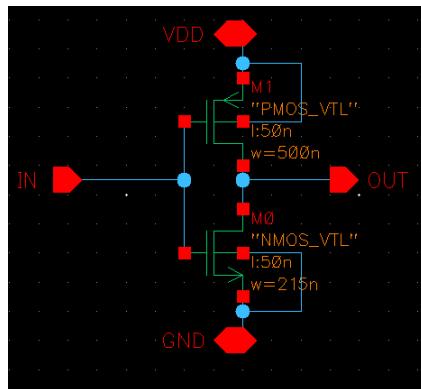
1

1.1 THIẾT KẾ SƠ ĐỒ NGUYÊN LÝ

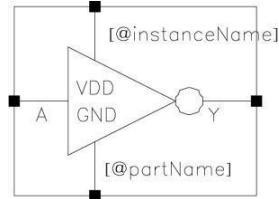
Bảng 3-1 Bảng sự thật của công NOT

A	Y
0	1
1	0

Sơ đồ nguyên lý & Ký hiệu công



Hình 1-1 Sơ đồ nguyên lý công NOT



Hình 1-2 Ký hiệu cồng NOT

1.2 THỰC HIỆN MÔ PHỎNG ĐÁP ỨNG DC

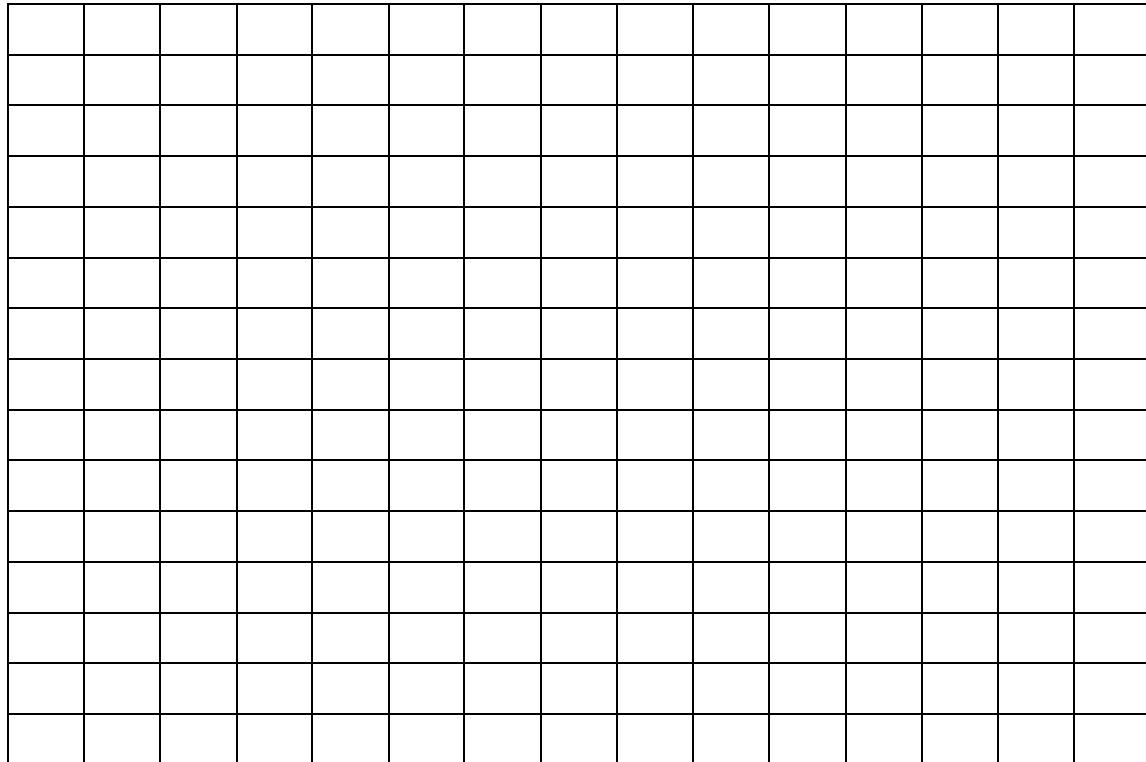
Sử dụng ADEL để thực hiện mô phỏng đáp ứng DC của cồng NOT, thực hiện cấp tín hiệu input dạng xung RAMP tầm điện áp thay đổi từ 0 đến 1V và khảo sát đáp ứng ngõ ra.

Thông số	Giá trị
V_{dd}	1 V
C_{load}	1 fF
Vin	0 V – 1V

Thực hiện đo điện áp ngõ ra tại các giá trị Vin sau:

Vin	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Vout										

Thực hiện vẽ lại dạng sóng Vout



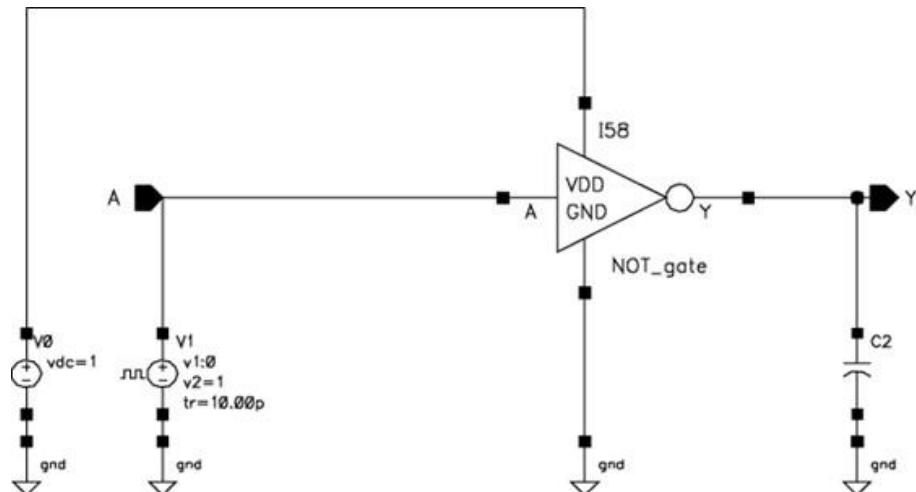
Có nhận xét gì về dạng sóng này từ đó giải thích hoạt động của mạch?

1.3 THỰC HIỆN MÔ PHỎNG ĐÁP ÚNG TRANSIENT

Sử dụng ADE L thực hiện mô phỏng theo thời gian (mô phỏng transient) để kiểm tra hoạt động theo bảng sự thật của công NOT, kết quả sẽ thu được dạng sóng như hình bên dưới, cùng với các thiết lập cho mạch testbench ở bảng 3-2. Tương tự cho các trường hợp kiểm tra hoạt động cho các công khác có nhiều chân ngõ vào hơn thì ta chỉ việc thêm các nguồn xung vào với các thông số thiết lập giống như trong trường hợp này và chỉ khác giá trị chu kỳ và độ rộng xung.

Thông số	Giá trị
V_{dd}	1 V
C_{load}	1 fF
Voltage 1	0 V
Voltage 2	1 V
Rise time	0.001 ns
Fall time	0.001 ns
Delay	0 ns
Pulse width	1 ns
Period	2 ns

Mạch testbench và kết quả dạng sóng được trình bày trong hình dưới đây (hình 3-3, 3-4)



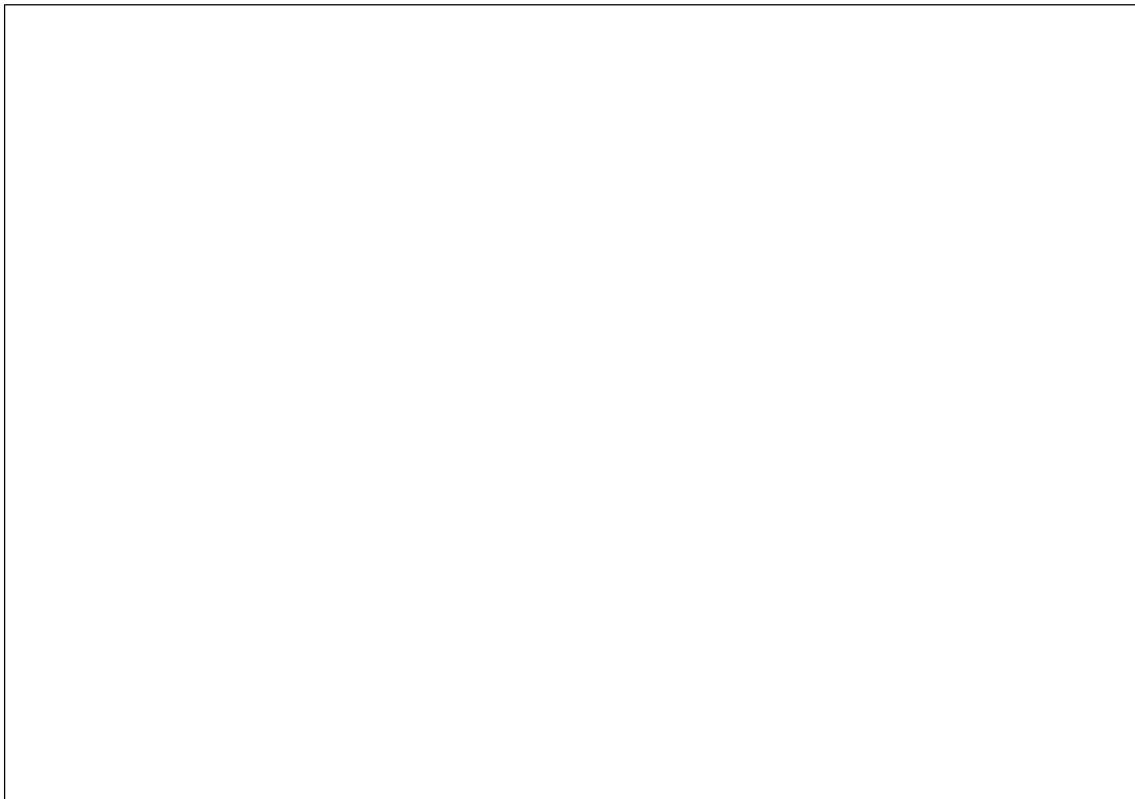
Mạch kiểm tra hoạt động của công NOT

Thực hiện đo những giá trị sau của đáp ứng ngõ ra:

Thông số	Kết quả
Trise – Rising Time (20% - 80%)	
Tfall – Falling Time (80% - 20%)	
Trise – Rising Time (10% - 90%)	
Tfall – Falling Time (90% - 10%)	
Dynamic Power	
Switching Power	

1.4 THỰC HIỆN VẼ LAYOUT CHO CỔNG NOT

- Hoàn thành stick diagram



- Hoàn thành layout của cổng NOT

THÍ NGHIỆM 2: THIẾT KẾ VÀ ĐO ĐẶC CỘNG NAND2, NOR2 VÀ ỨNG DỤNG THIẾT KẾ BỘ FULL-ADDER 4 BITS

2.1 THIẾT KẾ SƠ ĐỒ NGUYÊN LÝ CỘNG NAND2

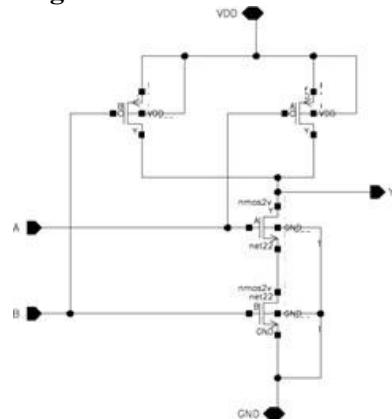
Bảng 2-1 Bảng sự thật của cỗng NAND2

A	B	Y
0	X	
X	0	
1	1	

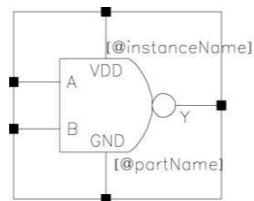
Bảng 2-2 Bảng giá trị kích thước của NMOS và PMOS

W_n	
L_n	
W_p	
L_p	

Sơ đồ nguyên lý & Ký hiệu cỗng



Hình 2-3 Sơ đồ nguyên lý cỗng NAND2



Hình 2-4 Ký hiệu công NAND2

2.2 MÔ PHỎNG ĐÁP DC CỔNG NAND2

Sử dụng ADEL để thực hiện mô phỏng đáp ứng DC của cổng NAND2, thực hiện cấp tín hiệu input dạng xung RAMP tầm điện áp thay đổi từ 0 đến 1V và khảo sát đáp ứng ngo ra.

Thông số	Giá trị
V_{dd}	1 V
C_{load}	1 fF
Vin1	0 V – 1V
Vin2	0 – 1V

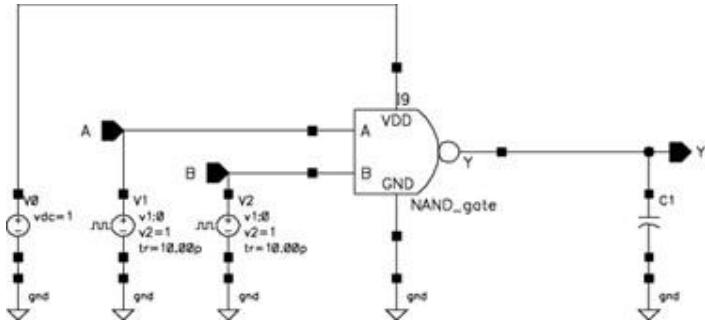
Thực hiện đo điện áp ngõ ra tại các giá trị Vin sau:

Thực hiện vẽ lại dạng song Vout

Có nhận xét gì về dạng song này từ đó giải thích hoạt động của mạch?

2.3 MÔ PHỎNG ĐÁP ÚNG TRANSIENT CỘNG NAND2

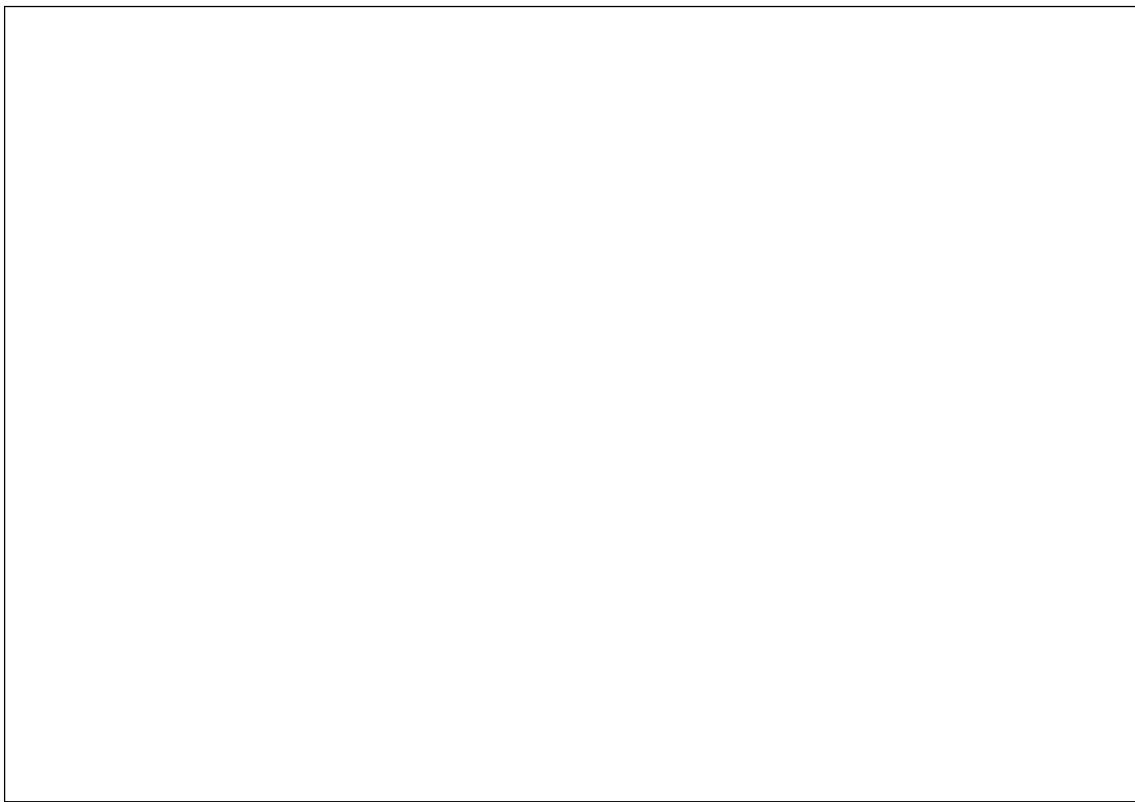
Sử dụng ADE L để kiểm tra hoạt động của cỗng NAND2 với các thiết lập cho mạch testbench như sau: Vdd= 1 V, Cload= 1f, Vpulse cho ngõ vào A (voltage 1 = 0 V, voltage 2 = 1 V, delay = 0 ns, rise time = fall time = 1 ps, pulse width = 1 ns, period = 2 ns), Vpulse cho ngõ vào B (voltage 1 = 0 V, voltage 2 = 1 V, delay = 0 ns, rise time = fall time = 1 ps, pulse width = 2.5 ns, period = 5 ns) và thời gian mô phỏng transient là 8 ns. Mạch Testbench:



Hình 2-5 Mạch kiểm tra hoạt động của cỗng NAND2

2.4 THIẾT KẾ LAYOUT CỘNG NAND2

- Hoàn thành stick diagram



- Hoàn thành layout của cỗng NOT

2.5 THỰC HIỆN TƯƠNG TỰ CÁC BƯỚC TRÊN CHO CỘNG NOR2

2.6 THIẾT KẾ BỘ FULL ADDER 4 BITS.

Yêu cầu: Dựa vào các cỗng logic cơ bản đã thiết kế trong các thí nghiệm trước, thiết kế và mô phỏng một bộ Full adder 4 bit và trình bày kết quả schematic(transistor level), symbol và waveform.

Thí Nghiệm Thiết Kế Vi Mạch Số

Bùi Lê Quốc Doanh
Thí Nghiệm Thiết Kế Vi Mạch Số

THÍ NGHIỆM 3: THIẾT KẾ D-FLIP FLOP, ỦNG DỤNG THIẾT KẾ THANH GHI 4 BITS. TÌM HIỂU VỀ CELL CHARACTERIZATION VÀ PHÂN TÍCH STA (STATIC TIMING ANALYSIS)

3.1 THIẾT KẾ SƠ ĐỒ NGUYÊN LÝ DFF

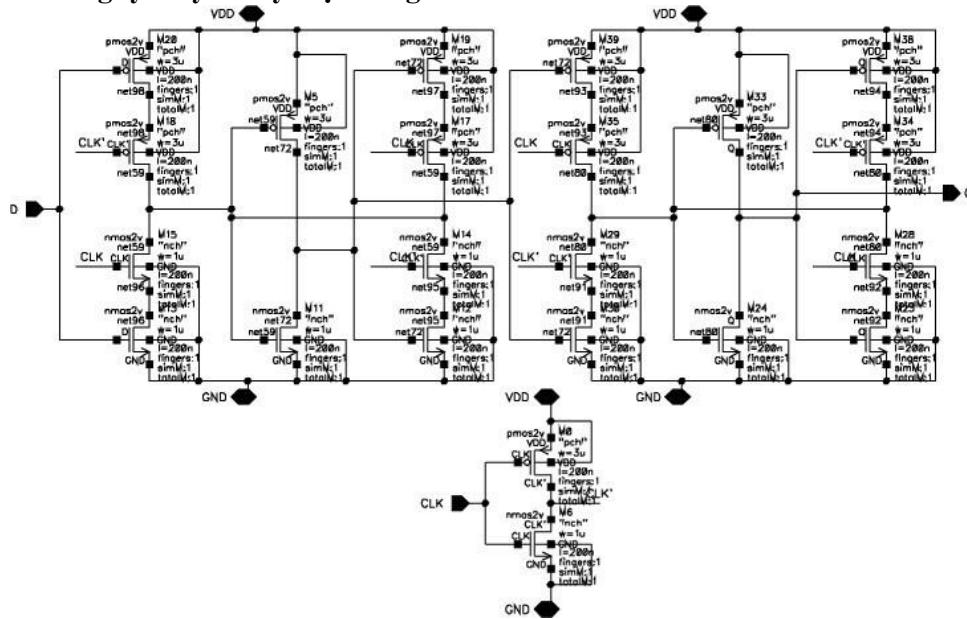
Bảng 3-1 Bảng sự thật của công DFFNEG

CLK	D	Q	Qnext
0, 1, ↑	X	0	
0, 1, ↑	X	1	
↓	0	X	
↓	1	X	

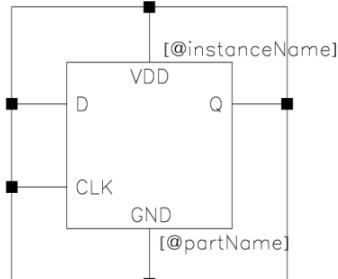
Bảng 3-22 Bảng giá trị kích thước của NMOS và PMOS

W_n	
L_n	
W_p	
L_p	

Sơ đồ nguyên lý & Ký hiệu công



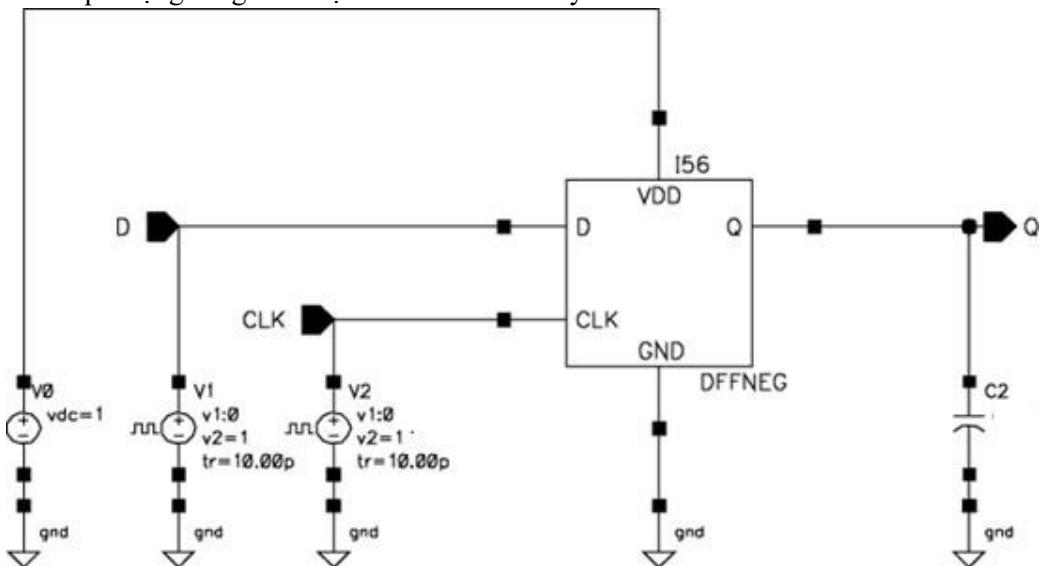
Hình 3-3 Sơ đồ nguyên lý công DFFNEG



Hình 3-4 Ký hiệu công DFFNEG

3.2 KIỂM TRA ĐÁP ỨNG TRANSIENT

Sử dụng ADE L để kiểm tra hoạt động của công DFFNEG với các thiết lập cho mạch testbench như sau: Vdd= 1 V, Cload = 1 f , Vpulse cho ngõ vào D (voltage 1 = 0 V, voltage 2= 1V, delay = 0 ns, rise time = fall time = 1 ps, pulse width = 2 ns, period = 4 ns), Vpulse cho ngõ vào CLK (voltage 1 = 0 V, voltage 2 = 1 V, delay = 0 ns, rise time = fall time = 1 ps, pulse width = 1 ns, period = 2 ns) và thời gian mô phỏng transient là 20 ns. Mạch testbench và kết quả dạng sóng thu được như hình dưới đây:



Hình 3-5 Mạch kiểm tra hoạt động của công DFFNEG

3.3 THỰC HIỆN CHARACTERIZATION

Trong phần này, ta sẽ thực hiện characterization cho tất cả các cổng đã thực hiện layout ở phần trên. Các đại lượng cần thực hiện trong bước characterization đó là: cell rise delay, cell fall delay, rise transistion, fall transistion, rise power, fall power, leakage power và input capacitance.

Để thực hiện characterization luận văn sẽ sử dụng OCEAN script và công cụ Calculator của phần mềm Cadence. Tất cả các cổng trong thư viện sẽ được dùng chung giá trị tụ điện C_{load} là 2 fF, 5 fF, 6 fF, 7 fF, 8 fF, 9 fF, 9.5 fF và giá trị thời gian lên (rise time), thời gian xuống (fall time) của nguồn xung ở các ngõ vào của cổng là 0.01 ns, 0.02 ns, 0.04 ns, 0.06 ns, 0.08 ns, 0.09 ns, 0.095ns.

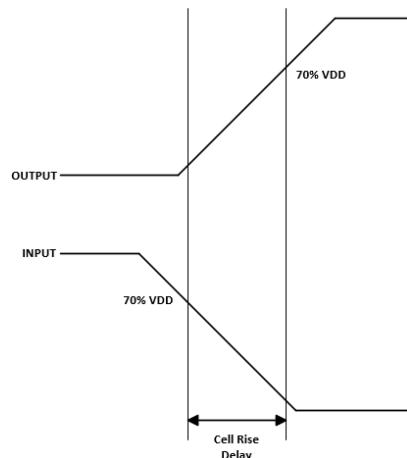
Khi sử dụng OCEAN script để đo cell rise delay, cell fall delay, rise transistion, fall transistion, rise power, fall power của cổng thì OCEAN script sẽ chạy 7x7 lần và cho ra bảng 7x7 giá trị đúng với format của file LIB mong muốn tạo ra. Đặc biệt, đối với trường hợp đo rise power và fall power cho các chân ở ngõ vào, giá trị này chỉ phụ thuộc vào rise time, fall time của nguồn xung nên chỉ sử dụng giá trị rise time, fall time ở trên mà không sử dụng giá trị tụ C_{load} . Vì vậy ta sẽ thu được kết quả công suất ở từng trường hợp và cho ra bảng 1x7 giá trị.

Sử dụng công cụ Calculator để đo Input capacitance và Leakage power, lý do không sử dụng script để đo bởi vì đây chỉ là một giá trị cần tìm và script thường dùng để đo một dãy các giá trị như thời gian trễ, công suất dynamic,...

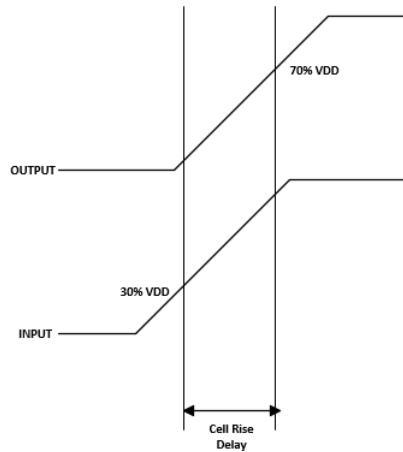
Tất cả các mô phỏng cho việc đo này đều là mô phỏng Post-Layout, không phải là mô phỏng Pre-Layout. Phần dưới đây sẽ trình bày chi tiết hơn về các đại lượng cần đo nói trên, cũng như là các hàm trong phần mềm để sử dụng tính toán.

3.2.1 Cell rise delay và Cell fall delay

Cell rise delay: là khoảng thời gian tính từ 70% giá trị điện áp của VDD tại cạnh lên đầu tiên ở ngõ ra và 70% giá trị điện áp của VDD tại cạnh xuống đầu tiên ở ngõ vào nếu cổng mà chúng ta đang xét thuộc loại “negative_unate”. Đối với trường hợp cổng đang xét là “positive_unate” thì giá trị cell rise delay được định nghĩa là khoảng thời gian tính từ 70% giá trị của VDD tại cạnh lên đầu tiên ở ngõ ra và 30% giá trị của VDD tại cạnh lên đầu tiên ở ngõ vào. Hình 3-105 và 3-106 dưới đây minh họa khái niệm thời gian này.

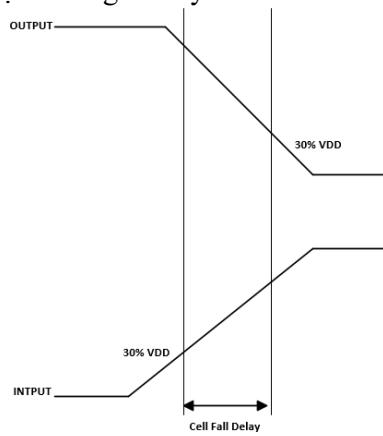


Hình 3-6 Cell rise delay khi xét trường hợp negative_unate của cổng

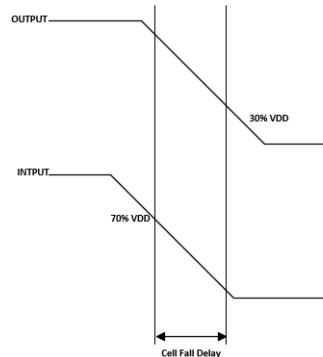


Hình 3-7 Cell rise delay khi xét trường hợp positive_unate của cổng

Cell fall delay: là khoảng thời gian tính từ 30% giá trị điện áp của VDD tại cạnh xuống đầu tiên ở ngõ ra và 30% giá trị điện áp của VDD tại cạnh lên đầu tiên ở ngõ vào nếu đang xét cổng thuộc loại “negative_unate”. Trường hợp cổng thuộc loại “positive_unate” thì giá trị thời gian này là khoảng thời gian được tính từ 30% giá trị điện áp của VDD tại cạnh xuống đầu tiên ở ngõ ra và 70% giá trị điện áp của VDD tại cạnh xuống đầu tiên ở ngõ vào. Hình 3-107 và 3-108 dưới đây minh họa khái niệm thời gian này.



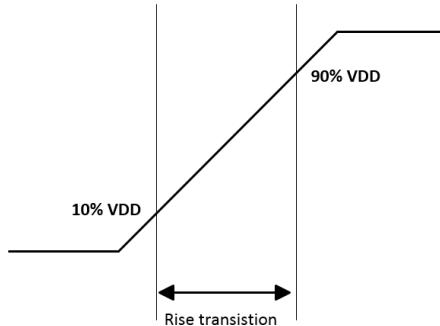
Hình 3-8 Cell fall delay khi xét trường hợp negative_unate của cổng



Hình 3-9 Cell fall delay khi xét trường hợp positive_unate của cổng

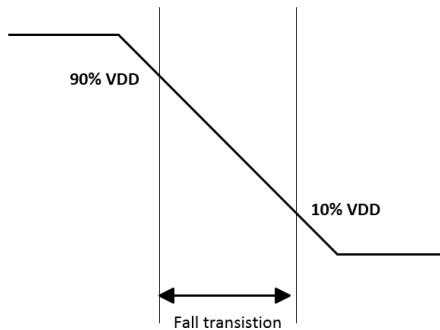
3.2.2 Rise transistion và Fall transistion

Rise transistion: hay còn gọi là thời gian lên, trong phần characterization xét thời gian lên ở ngõ ra của công. Tại cạnh lên ở ngõ ra, giá trị thời gian này khoảng thời gian giữa 90% giá trị điện áp của VDD và 10% giá trị điện áp của VDD.



Hình 3-10 Rise transistion

Fall transistion: hay còn gọi là thời gian xuống, tương tự như thời gian lên, giá trị này là khoảng thời gian giữa 90% giá trị của VDD và 10% giá trị của VDD. Tuy nhiên giá trị này được tính ở cạnh xuống của ngõ ra.



Hình 3-11 Fall transistion

3.2.3 Rise power và Fall power

Thực hiện đo hai công suất này ở các chân ngõ vào và của công. Khái niệm dưới đây xét trường hợp công suất của công, trường hợp ở ngõ vào được thực hiện tương tự.

Rise power: là công xuất của công được tính trong khoảng thời gian ngõ ra rising tức là chuyển từ mức thấp lên mức cao ($0\text{ V} \rightarrow \text{VDD}$).

Fall power: là công xuất của công được tính trong khoảng thời gian ngõ ra falling tức là chuyển từ mức cao xuống mức thấp ($\text{VDD} \rightarrow 0\text{ V}$).

Sử dụng hàm tích phân “integ” của công cụ Calculator vào OCEAN script để tính công suất này. Công thức tính tổng quát như sau:

"VDD * integ(i("/V0/PLUS"), t, t + Δt)/(1e - 9)".

là khoảng thời gian ngõ ra rising hoặc falling phụ thuộc vào việc đang đo công suất nào.

3.2.4 Leakage power

Leakage power là công suất do dòng điện rò $I_{leakage}$ gây ra. Ta thực hiện tính công suất này trong hai trường hợp mạch ở mạng pull-up và ở mạng pull-down. Lý do chỉ thực hiện ở hai trường hợp này là tại vì chúng ta mong muốn tìm giá trị công suất trong trường hợp “worst case” là đủ. Chọn ra giá trị lớn hơn làm giá trị công suất rò (Leakage power) cần tìm. Để đo được công suất này bằng công cụ Calculator ta sử dụng công thức sau:

"average(VT(/VDD) * IT("/V0/PLUS"))"

Trong đó: giả sử **V0** là nguồn cung cấp VDD cho cổng, **average** là hàm tính giá trị trung bình

từ kết quả tích của dòng $I_{leakage}$ và điện áp VDD = 1 V cung cấp cho cổng.

3.2.5 Input capacitance

Giá trị điện dung ngõ vào được tính toán dựa vào công thức: $I = C \frac{dV}{dt}$ (1), khi cho một nguồn xung vào chân ngõ vào của cổng sẽ sinh ra dòng điện ở ngõ vào tại thời điểm điện áp ngõ vào thay đổi từ mức thấp lên mức cao hoặc ngược lại. Sở dĩ có dòng điện này là nhờ vào quá trình nạp xả của tụ điện ngõ vào (Input capacitance) ở chân ngõ vào tương ứng. Dựa vào công thức (1) suy ra:

$$Idt = CdV \quad (2)$$

Lấy tích phân 2 về phương trình (2) trong khoảng thời gian $Δt$ là thời gian điện áp ngõ vào chuyển từ mức thấp lên mức cao. Ta có:

$$\int_t^{t+Δt} Idt = \int_t^{t+Δt} CdV \quad (3)$$

Do đó:

$$C = \frac{\int_t^{t+Δt} Idt}{\int_t^{t+Δt} dV} = \frac{\int_t^{t+Δt} Idt}{VDD}$$

Từ đó sử dụng công thức sau trên công cụ Calculator để tính được giá trị tụ ở các chân ngõ vào như sau:

"integ(i("/V1/PLUS"), t, t + Δt)/(VDD * (1e - 9))"

Trong đó: giả sử V1 là nguồn xung ở chân ngõ vào và $Δt$ là thời gian nguồn xung ngõ vào rising hoặc falling.

3.2.6 Thực hiện thí nghiệm

Thực hiện characteristic cho một trong các cổng đã thực hiện NOT, NAND2, NOR2 hoặc cả DFF và điều kết quả vào các bảng sau:

Bảng 3-12 Leakage power của công

Leakage power (pW)

Bảng 3-13 Cell rise delay của công (ns)

$C_{load}(fF)$	2	5	6	7	8	9	9.5
0.01							
0.02							
0.04							
0.06							
0.08							
0.09							
0.095							

Bảng 3-14 Cell fall delay của công (ns)

$C_{load}(fF)$	2	5	6	7	8	9	9.5
0.01							
0.02							
0.04							
0.06							
0.08							
0.09							
0.095							

Bảng 3-15 Rise transistion của công (ns)

$C_{load}(fF)$	2	5	6	7	8	9	9.5
0.01							
0.02							
0.04							
0.06							
0.08							
0.09							
0.095							

Bảng 3-16 Fall transistion của công(ns)

$C_{load}(fF)$	2	5	6	7	8	9	9.5
0.01							
0.02							
0.04							
0.06							
0.08							
0.09							
0.095							

Bảng 3-17 Rise power của công(W)

$C_{load}(fF)$	2	5	6	7	8	9	9.5
0.01							
0.02							
0.04							
0.06							
0.08							
0.09							
0.095							

Bảng 3-18 Fall power của công(W)

$C_{load}(fF)$	2	5	6	7	8	9	9.5
0.01							
0.02							
0.04							
0.06							
0.08							
0.09							
0.095							

3.4) Thiết kế thanh ghi 4 bits.

*Mô tả: Dựa vào thiết kế DFF đã làm ở phần 3.1, sinh viên thực hiện thiết kế 1 thanh ghi 4 bits và mô phỏng hoạt động của thanh ghi trong 2 chu kì liên tiếp(Lưu ý: 4 bits data ở 2 chu kì là khác nhau). Mô tả hoạt động thanh ghi và trình bày kết quả dưới dạng schematic, waveform(không cần layout).

Lưu ý: Thanh ghi có các dạng nhận và truyền data khác nhau(4 dạng chính: SISO,PISO,SIPO,PIPO). Sinh viên tìm hiểu và chọn một dạng để làm cho thiết kế của mình và giải thích lý do chọn loại thanh ghi đó.

THÍ NGHIỆM 4: THIẾT KẾ MEMORY SRAM

4.1) Lý thuyết

Sinh viên nghe giảng hoặc tự đọc trong tài liệu hướng dẫn đính kèm.

4.2) Yêu cầu thiết kế.

*Mô tả:

1) Sinh viên thiết kế 1 cell SRAM và kiểm tra hoạt động của cell SRAM đó ở 2 chế độ WRITE và READ. Sau đó sử dụng mạch sense amplifier(SA) để tiến hành lại quá trình READ và so sánh với trường hợp không dùng mạch SA. Trình bày kết quả dưới dạng schematic, waveform và số liệu đo đạc. Nhận xét kết quả thu được.

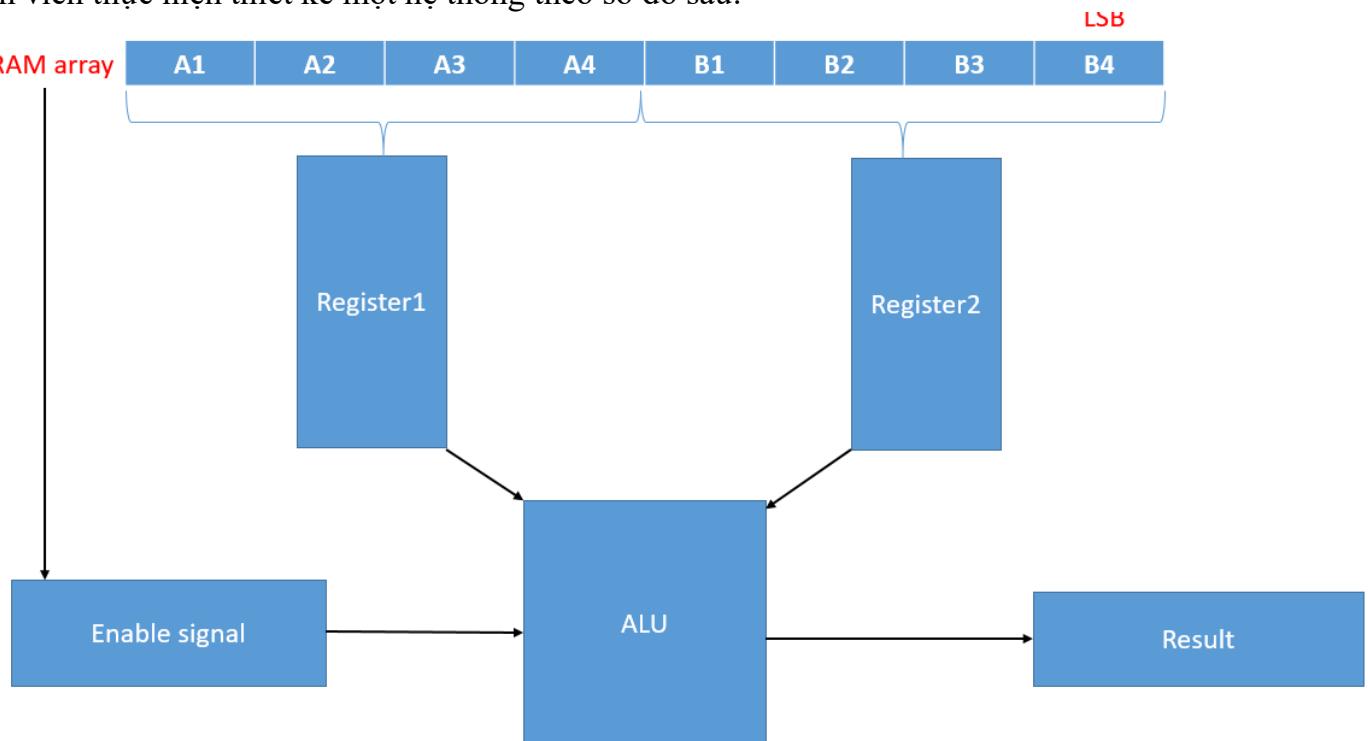
2) Dựa trên cell SRAM đã thiết kế, sinh viên thiết kế SRAM array 4x4 và kiểm tra hoạt động của mảng SRAM đã thiết kế ở 2 chế độ READ và WRITE(không cần dùng mạch SA cho trường hợp thiết kế này). Trình bày kết quả dưới dạng schematic, waveform. Nhận xét kết quả thu được.

*Lưu ý : Sinh viên có thể lựa chọn tùy ý data lưu vào SRAM (ngoại trừ trường hợp toàn bit 0 hoặc 1). Tụ dùng trong mạch được sử dụng với giá trị C = 1-10(fF).

BÀI TẬP LỚN

5.1) Mô tả thiết kế

Sinh viên thực hiện thiết kế một hệ thống theo sơ đồ sau:



***SRAM array:** Tạo một mảng SRAM đơn giản 1x8 và lưu giá trị 8 bit theo quy ước sau.

Số thứ tự nhóm(STT) là số lẻ: $(STT+100)_{10} \Leftrightarrow (A1A2A3A4B1B2B3B4)_2$

Số thứ tự nhóm(STT) là số chẵn: $(STT+200)_{10} \Leftrightarrow (A1A2A3A4B1B2B3B4)_2$

***Register:** Thanh ghi chứa 4 bit data. Sinh viên có thể dùng thanh ghi đã thiết kế ở phần thí nghiệm trước hoặc thiết kế lại cho phù hợp.

***ALU:** Khối tính toán chính, có chức năng thực hiện các phép toán đơn giản như cộng hoặc trừ.

***Enable Signal:** Khối tín hiệu chọn cho khối ALU. Quy ước: nếu tổng số bit 1 trong SRAM array là lẻ thì Enable Signal cho phép khối ALU thực hiện phép **cộng** 4 bit ($A_1A_2A_3A_4 + B_1B_2B_3B_4$), nếu tổng số bit 1 trong SRAM array là **chẵn** thì Enable Signal cho phép khối ALU thực hiện phép **trừ** 4 bit ($A_1A_2A_3A_4 - B_1B_2B_3B_4$).

5.1) Yêu cầu thực hiện

Dựa trên mô tả của BTL, sinh viên dựa trên STT nhóm mình thực hiện thiết kế và mô phỏng hệ thống trên và chứng minh bằng kết quả waveform giống với kết quả tính toán lý thuyết.

KẾT THÚC THÍ NGHIỆM
