

BÁO CÁO

TỔNG HỢP VÀ SO SÁNH CÔNG SUẤT,

DỘ TRỄ HAI MẠCH NHÂN 8-BIT

MỤC LỤC

1. Tổng hợp Verilog HDL bằng Yosys.....	1
1.1. Giới thiệu về Yosys.....	1
1.2. Cài đặt Yosys.....	1
1.2.1. Cài đặt các gói phụ thuộc.....	1
1.2.2. Sao chép Yosys từ kho lưu trữ Github.....	1
2.3. Xây dựng các cấu hình.....	2
2.4. Thực hiện biên dịch Yosys.....	2
2.5. Thêm thư mục gốc yosys vào biến môi PATH.....	2
1.3. Tổng hợp Verilog RTL với Yosys.....	2
1.4. Hiển thị sơ đồ khối với Yosys.....	4
1.5. Nhập tệp verilog đã tổng hợp vào Cadence.....	6
2. Thêm thư viện FreePDK45 vào trong Cadence Virtuoso.....	9
2.1. Giới thiệu về FreePDK45.....	9
2.2. Sao chép FreePDK45.....	10
2.3. Thêm FreePDK45, NangateOpenCellLib vào Cadence Virtuoso.....	10
2.3.1. Sao chép thư mục chứa FreePDK45, NangateOpenCellLib vào thư mục làm việc (cds_work).....	10
2.3.2. Thiết lập thư viện FreePDK45.....	11
2.3.3. Thêm thư viện NangateOpenCellLibrary.....	12
2.3.4. Thiết lập Cadence Virtuoso.....	12
2.3.5. Thiết lập Cadence Virtuoso (DRF).....	14
2.3.6. Thiết lập Cadence Virtuoso (TF).....	17
2.4. Kiểm tra thư viện FreePDK45.....	17
2.4.1. Mô phỏng một biểu thức Y.....	17
2.4.1.1. Sơ đồ nguyên lý.....	17
2.4.1.2. Thiết lập mô phỏng.....	18
2.4.1.3. Kết quả mô phỏng.....	20
3. Tổng hợp hai bộ nhân 8-bit sử dụng Yosys.....	21
3.1. Tổng hợp bộ nhân dùng CLA.....	21
3.1.1. Chuẩn bị.....	21
3.1.2. Xây dựng TCL script để tổng hợp mạch nhân với CLA.....	22
3.1.3. Chạy TCL Script để tổng hợp mạch nhân.....	23
3.2. Tổng hợp bộ nhân SCDM.....	24
3.2.1. Chuẩn bị.....	24
3.1.2. Xây dựng TCL script để tổng hợp mạch nhân SCDM.....	25
3.1.3. Chạy TCL Script để tổng hợp mạch nhân SCDM.....	25
4. So sánh công suất, độ trễ hai bộ nhân 8-bit.....	26
4.1. Điều kiện mô phỏng.....	26
4.1.1. Thông số môi trường.....	26
4.1.2. Thông số nguồn DC.....	26
4.1.3. Thông số nguồn AC (nguồn xung).....	26
4.2. Mô phỏng mạch nhân sử dụng CLA.....	27
4.2.1. Sơ đồ lý nguyên mạch nhân sử dụng CLA.....	27
4.2.2. Sơ đồ nguyên lý dùng cho mô phỏng mạch nhân sử dụng CLA.....	27
4.2.3. Quá trình mô phỏng.....	28
4.2.4. Kết quả mô phỏng mạch nhân với CLA.....	29
#t=0us.....	29
#t=8,400025us.....	30
#t=9,580025us.....	31

#t=19,200025us.....	31
#t=25,200025us.....	32
#t=33,600025us.....	32
4.3. Mô phỏng mạch nhân SCDM.....	33
4.3.1. Sơ đồ nguyên lý mạch nhân SCDM8_73.....	33
4.3.2. Sơ đồ nguyên lý dùng cho mô phỏng mạch nhân SCDM8_73.....	34
4.2.3. Quá trình mô phỏng.....	34
4.2.4. Kết quả mô phỏng mạch nhân SCDM.....	35
#t=0us.....	35
#t=8,400025us.....	36
#t=9,500025us.....	36
#t=19,200025us.....	37
#t=25,200025us.....	37
#t=33,600025us.....	38
4.4. So sánh, kết luận.....	39
4.4.1. So sánh độ trễ lan truyền.....	39
4.4.2. So sánh công suất tiêu thụ.....	39
4.4.3. So sánh thời gian mô phỏng.....	40

1. Tổng hợp Verilog HDL bằng Yosys

1.1. Giới thiệu về Yosys

Đây là một framework dành cho các công cụ tổng hợp RTL. Hiện tại, nó hỗ trợ rộng rãi Verilog-2005 và cung cấp một tập hợp cơ bản các thuật toán tổng hợp cho nhiều lĩnh vực ứng dụng khác nhau.

Yosys có thể được tùy biến để thực hiện bất kỳ công việc tổng hợp nào bằng cách kết hợp các bước xử lý (thuật toán) hiện có thông qua các tập lệnh tổng hợp và thêm các bước xử lý bổ sung nếu cần bằng cách mở rộng mã nguồn C++ của yosys.

Yosys là phần mềm miễn phí (mã nguồn GitHub: <https://github.com/YosysHQ/yosys>) được cấp phép theo giấy phép ISC (một giấy phép tương thích với GPL, tương tự như giấy phép MIT hoặc giấy phép BSD 2 điều khoản).

Phần mềm của bên thứ ba được phân phối kèm theo phần mềm này được cấp phép theo các giấy phép tương thích. Vui lòng tham khảo các thư mục con abc và libs để biết điều khoản giấy phép của chúng.

1.2. Cài đặt Yosys

1.2.1. Cài đặt các gói phụ thuộc

Để cài đặt YoSys cần có trình biên dịch C++ hỗ trợ C++17 cũng như một số công cụ chuẩn như GNU Flex, GNU Bison, Make và Python. Một số công cụ bổ sung: readline, libffi, Tcl và zlib; là tùy chọn nhưng được bật theo mặc định.

```
sudo apt-get install gperf build-essential bison flex \
libreadline-dev gawk tcl-dev libffi-dev git graphviz \
xdot pkg-config python3 libboost-system-dev \
libboost-python-dev libboost-filesystem-dev zlib1g-dev
```

1.2.2. Sao chép Yosys từ kho lưu trữ Github

Tạo một bản sao mã nguồn mở của Yosys thông qua **git clone** với cờ **--recurse-submodules** để đảm bảo các thư mục, tập tin bên trong được tải xuống đầy đủ. Lệnh bên dưới sẽ tạo một bản sao của mã nguồn Yosys tại thư hiện tại (khuyến nghị clone vào ~ hay /home/<username>/). Sau đó di chuyển con trỏ thư mục (thư mục làm việc) hiện tại vào bên trong thư mục mã nguồn mở của Yosys. (Lưu ý rằng đường dẫn có thể sẽ thay đổi, hoặc không còn khả dụng, vui lòng kiểm tra lại đường dẫn trước khi thực hiện các bước tiếp theo.

```
git clone --recurse-submodules https://github.com/YosysHQ/yosys.git
```

Kết quả:

```
ngxxfus@ngxxfus-X409FA [10:59:46] [-]
$ cd Downloads
ngxxfus@ngxxfus-X409FA [10:59:52] [Downloads]
$ git clone --recurse-submodules https://github.com/YosysHQ/yosys.git
Cloning into 'yosys'...
remote: Enumerating objects: 90894, done.
remote: Counting objects: 100% (1528/1528), done.
remote: Compressing objects: 100% (537/537), done.
remote: Total 90894 (delta 1317), reused 1011 (delta 991), pack-reused 89366 (from 5)
Receiving objects: 100% (90894/90894), 30.61 MiB | 15.94 MiB/s, done.
Resolving deltas: 100% (67111/67111), done.
Submodule 'abc' (https://github.com/YosysHQ/abc) registered for path 'abc'
Submodule 'cxxopts' (https://github.com/jarro2783/cxxopts) registered for path 'libs/cxxopts'
Cloning into '/mnt/sda1/Linux/Downloads/yosys/abc'...
remote: Enumerating objects: 68347, done.
remote: Counting objects: 100% (26820/26820), done.
remote: Compressing objects: 100% (1494/1494), done.
remote: Total 68347 (delta 25504), reused 25347 (delta 25326), pack-reused 41527 (from 2)
Receiving objects: 100% (68347/68347), 52.51 MiB | 14.82 MiB/s, done.
Resolving deltas: 100% (67111/67111), done.
Cloning into '/mnt/sda1/Linux/Downloads/yosys/libs/cxxopts'...
remote: Enumerating objects: 1917, done.
remote: Counting objects: 100% (225/225), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 1917 (delta 196), reused 169 (delta 169), pack-reused 1692 (from 3)
Receiving objects: 100% (1917/1917), 849.79 KiB | 6.44 MiB/s, done.
Resolving deltas: 100% (1124/1124), done.
Submodule path 'abc': checked out 'e55d316c9a7f2a84a76eda555aa6ec083c9d0d'
Submodule path 'libs/cxxopts': checked out '4bf61f08097b110d9e3991864650a405b3dd515d'
ngxxfus@ngxxfus-X409FA [11:00:14] [Downloads]
$ cd yosys
ngxxfus@ngxxfus-X409FA [11:00:59] [yosys]
$
```

Hình . Tạo bản sao mã nguồn yosys.

Ghi chú: Trong hình, bản sao sẽ được lưu ở ~ hay /home/ngxxfus. Đường dẫn đến mã nguồn yosys là: ~/yosys - chính là đường dẫn thư mục gốc của yosys (yosys root dir).

2.3. Xây dựng các cấu hình

Việc biên dịch Yosys hoàn toàn dựa trên các tệp Makefile và sử dụng một số biến môi trường. Phương thức xây dựng các cấu hình để biên dịch Yosys là sử dụng các quy tắc được xây dựng sẵn có trong Makefile.conf. Dòng lệnh bên dưới sẽ dọn dẹp thư mục và tạo một tệp cấu hình ban đầu:

```
make config-gcc
```

2.4. Thực hiện biên dịch Yosys

Từ thư mục gốc yosys (yosys root dir), hãy gọi các lệnh sau:

```
sudo make -j8
sudo make -j8 install
```

GHI CHÚ:

- + `'-j8` cho Make biết rằng nó có nhiều lõi song song hơn để chạy cùng một lúc.
- + Sau khi quá trình cài đặt hoàn tất, một tệp thực thi `yosys` sẽ được tạo ra để sử dụng.

2.5. Thêm thư mục gốc yosys vào biến môi PATH

Mở ~/.bashrc (hoặc ~/.zshrc), sau đó thêm dòng bên dưới. Lưu ý rằng hãy khởi động lại Terminal sau khi thêm đường dẫn mới vào biến PATH. Điều này làm yosys có thể được gọi ở bất kỳ đâu với một lệnh rất ngắn là yosys. Lưu ý rằng có thể bỏ qua bước này, tuy nhiên nếu bỏ qua bước này, khi sử dụng yosys, sẽ cần chỉ rõ tập tin thực thi (exec file) của yosys đang ở đâu.

```
export PATH=$PATH:~/yosys
```

1.3. Tổng hợp Verilog RTL với Yosys

Yosys hỗ trợ chạy bằng lệnh của Yosys hoặc chạy qua file tcl. Chạy qua file tcl được khuyến nghị vì tập lệnh TCL (.tcl) cho phép tạo các biến giúp thuận tiện hơn trong việc tái sử dụng, lập trình, trong khi tập lệnh Yosys (.ys) không cho phép. Ví dụ tổng hợp một mạch đếm lên 8-bit có tín tạm dừng SS, top-module là UP_CNT8b:

```
/* Module name: 8-bit up counter
```

Author: Nguyen Thanh Phu */

```
module UP_CNT8b (
    input clk, rst, SS,
    output [7:0] OUT
);
    reg [7:0] OUT;
    always @(posedge clk)
    begin
        if (rst == 1)
            OUT <= 8'b0;
        else if (SS == 1)
            OUT <= OUT + 1;
    end
endmodule
```

Chú ý rằng nếu top-module có chứa các phiên bản (instance) của các module con thì phải nhập kèm định nghĩa của các module con này vào cùng!

Đoạn mã bên dưới được viết bằng TCL-Script để tổng hợp module UP_CNT8b ở trên. Lưu ý rằng các thư mục phải kết thúc bằng dấu / và phải liệt kê tất cả các file Verilog được dùng trong thiết kế và bên trong **verilog_files**. Danh sách các file trong **verilog_files** kết thúc bằng dấu xuống dòng và chấp nhận đường dẫn thư mục con ở trong **verilog_files** (VD: sub-folder0/sub-folder1/my-part-x.v). Ngoài ra, để tổng hợp được, Yosys cần file *.lib là thư viện cell tiêu chuẩn (VD: Nangate Open Cell Library 45nm):

```
# IMPORTANT: Every directory in this script must be absolute paths
# and end with a '/'
# E.g:
#   `/home/user/dir/ -----> `/home/user/dir/synth_output.v'
#   `/home/user/dir` -----> `/home/user/dirsynth_output.v'
# HOME directory
set working_dir
"/mnt/sda1/DOC/HCMUTE_DOCs/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/
yosys/"

# Set output directory
set output_dir
"/mnt/sda1/DOC/HCMUTE_DOCs/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/
yosys/"
set lib_path
"/mnt/nvme0n1p3/Shared/.my_vmware/CentOS_Exchange/NanGate_45nm_OCL_v2010_12/pdk_v1.3_v2010_12/Nangate
OpenCellLibrary_PDKv1_3_v2010_12/NangateOpenCellLibrary_PDKv1_3_v2010_12/Front_End/Liberty/NLDM/Nangat
eOpenCellLibrary_typical.lib"
# Set output file names
set synth_output "synth_output_8bit_up_counter.v"
set mapped_output "mapped_output_8bit_up_counter.v"
# Set top module name
set top_module "UP_CNT8b"
# Define variable for Verilog files using set
set verilog_files {
    _8bit_up_counter.v
}
# Read all Verilog files using Tcl foreach
foreach file $verilog_files {
```

```

yosys read_verilog $working_dir$file
}
# Set the top module
yosys hierarchy -check -top $top_module
# Synthesize the design with flatten
yosys synth -top $top_module -flatten
# Write unmapped synthesized output
yosys write_verilog -noattr $output_dir$synth_output
# Read the liberty file (replace with your library path)
yosys read_liberty -lib $lib_path
yosys opt
# Map the design to the library
yosys dfllibmap -liberty $lib_path
yosys abc -liberty $lib_path
yosys techmap -map +/techmap.v
yosys check

yosys write_verilog -noattr $output_dir$mapped_output

```

Sử dụng tập lệnh TCL cho phép sử dụng lại tập lệnh TCL với cùng cấu trúc như dự án cũ. Để chạy tập lệnh TCL trong Yosys, sử dụng cờ -c để chỉ định đầu vào là file TCL (VD: yosys -c tcl_tên_tệp.tcl):

```
yosys -c ./synthe_8bit_up_counter.tcl
```

```

ngxxfus@ngxxfus-X409FA [11:20:39] [yosys]
$ yosys -c ./synthe_8bit_up_counter.tcl

|-----\
| Yosys --- Yosys Open Synthesis Suite
| Copyright (C) 2012 - 2025 Claire Xenia Wolf <claire@yosyshq.com>
| Distributed under an ISC-like license, type "license" to see terms
| \-----\

Yosys 0.52+45 (git sha1 19845be85, g++ 13.3.0-6ubuntu2-24.04 -fPIC -O3)

1. Executing Verilog-2005 frontend: /mnt/sda1/DOC/HOMUTE_DOCS/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/8bit_up_counter.v
Parsing Verilog input from /mnt/sda1/DOC/HOMUTE_DOCS/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/8bit_up_counter.v to AST representation.
Generating RTLLIL representation for module 'UP_CNT8'.
Successfully finished Verilog frontend.

2. Executing Verilog-2005 frontend: /mnt/sda1/DOC/HOMUTE_DOCS/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/#
Parsing Verilog input from /mnt/sda1/DOC/HOMUTE_DOCS/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/# to AST representation.
Successfully finished Verilog frontend.

3. Executing Verilog-2005 frontend: /mnt/sda1/DOC/HOMUTE_DOCS/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/AND_GATE.v
Parsing Verilog input from /mnt/sda1/DOC/HOMUTE_DOCS/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/AND_GATE.v to AST representation.
Successfully finished Verilog frontend.

4. Executing Verilog-2005 frontend: /mnt/sda1/DOC/HOMUTE_DOCS/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/#
Parsing Verilog input from /mnt/sda1/DOC/HOMUTE_DOCS/NH_2024_2025_HK_02/DA1/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/# to AST representation.
Successfully finished Verilog frontend.

```

Hình . Chạy tập lệnh TCL với Yosys.

Lưu ý rằng: Trong Yosys Shell không tồn tại lệnh cd hoặc pwd, tất cả các đường dẫn đều là đường dẫn **tuyệt đối**. Dựa trên tập lệnh TCL ở trên, sau khi chạy, Yosys sẽ tổng hợp hai tệp, nhưng một trong số chúng được ánh xạ tới một thư viện!

Mặc định, yosys chỉ in trạng thái hoạt động ra Terminal, để in ra một file (VD: yosys_synth.log) cần sử dụng một tiện ích dòng lệnh được sử dụng để đọc dữ liệu từ stdin - tee. Ngoài ra còn có thể thêm cờ -a với tee thêm vào cuối file được chỉ định thay vì ghi đè.

```
yosys -c tcl_tên_tệp.tcl | tee yosys_synth.log
```

1.4. Hiển thị sơ đồ khối với Yosys

Ngoài việc tổng hợp Verilog HDL, Yosys còn cung cấp trình vẽ sơ đồ khối hiển thị thông qua xdot. Lệnh bên dưới mô tả cách sử dụng chức năng vẽ sơ đồ khối của Yosys. Cờ -p cho Verilog biết rằng chuỗi liền sau là Yosys-Shell.

```
yosys -p "read_verilog /<đường dẫn>/<dេn>/<verilog_filename.v>; show"
```

Kết quả:

```

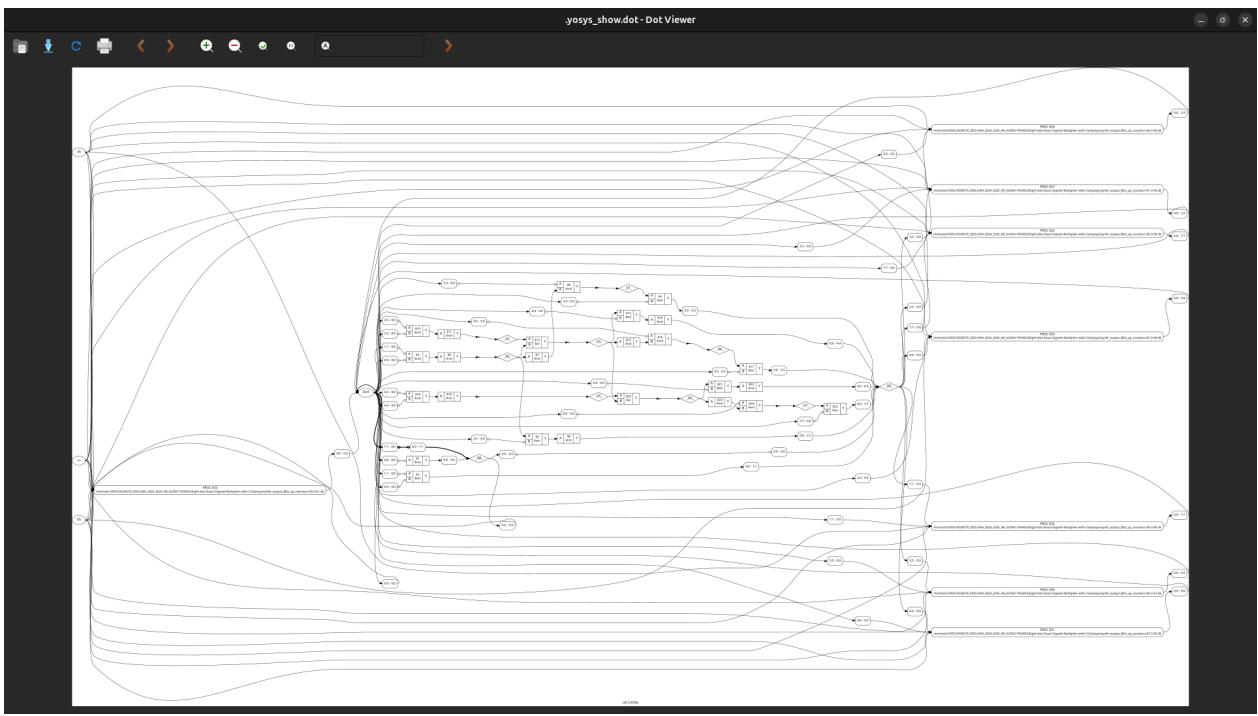
ngxxfus@ngxxfus-X409FA [11:33:50] [yosys]
$ yosys -p "read_verilog /mnt/sda1/DOC/HOMUTE_DOCs/MH_2024_2025_HK_02/DAL/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/synth_output_8bit_up_counter.v; show"
-----
| yosys -- Yosys Open Synthesis Suite
| Copyright (C) 2012 - 2025 Claire Xenia Wolf <claire@yosys.hq.com>
| Distributed under an ISC-like license. Type "license" to see terms
| https://yosys.hq.com/LICENSE
Yosys 0.52+45 (git sha1 19845be85, g++ 13.3.0-6ubuntu2-24.04 -fPIC -O3)
-- Running command 'read_verilog /mnt/sda1/DOC/HOMUTE_DOCs/MH_2024_2025_HK_02/DAL/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/synth_output_8bit_up_counter.v; show' --
1. Executing Verilog frontend: /mnt/sda1/DOC/HOMUTE_DOCs/MH_2024_2025_HK_02/DAL/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/synth_output_8bit_up_counter.v
Parsing Verilog input from '/mnt/sda1/DOC/HOMUTE_DOCs/MH_2024_2025_HK_02/DAL/TASK02/Eight-bits-Exact-Signed-Multiplier-with-CLA/yosys/synth_output_8bit_up_counter.v' to AST representation.
Generating RTLIL representation for module 'UP_CNTB'.
Successfully finished Verilog frontend.

2. Generating Graphviz representation of design.
Writing dot description to '/home/ngxxfus/.yosys_show.dot'.
Dumping module UP_CNTB to page 1.
Exec: { test -f '/home/ngxxfus/.yosys_show.dot.pid' && fuser -s '/home/ngxxfus/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >63; exec xdot '/home/ngxxfus/.yosys_show.dot'; ) 3> '/home/ngxxfus/.yosys_show.dot.pid' &

End of script. Logfile hash: b55649feef, CPU: user 0.01s system 0.00s, MEM: 15.44 MB peak
Yosys 0.52+45 (git sha1 19845be85, g++ 13.3.0-6ubuntu2-24.04 -fPIC -O3)
Time spent: 654.2x read_verilog (0 sec), 34.1x show (0 sec)
ngxxfus@ngxxfus-X409FA [11:34:07] [yosys]
$ ./usr/lib/python3/dist-packages/xdot/dot/elements.py:174: UserWarning: Font family 'Times-Roman' is not available, using 'Ubuntu Sans 11'
  warnings.warn(msg)

```

a.

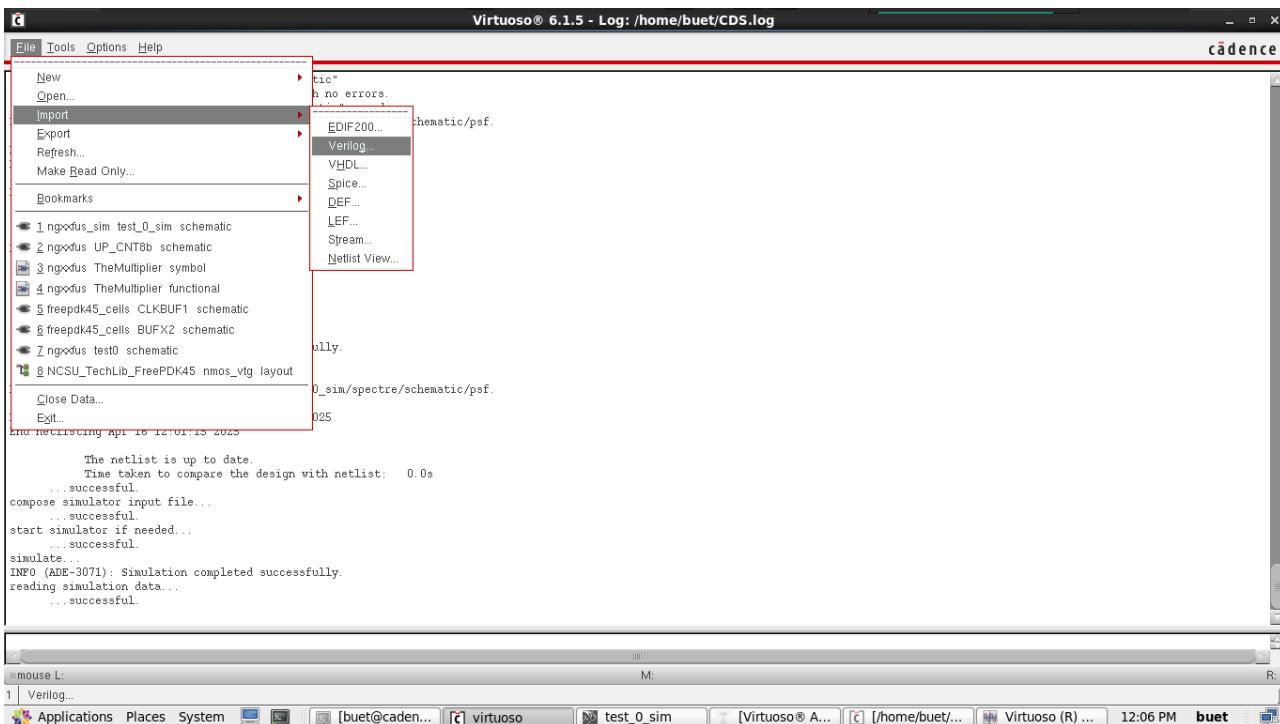


b.

Hình . (a, b) log và kết quả sau khi chạy lệnh show.

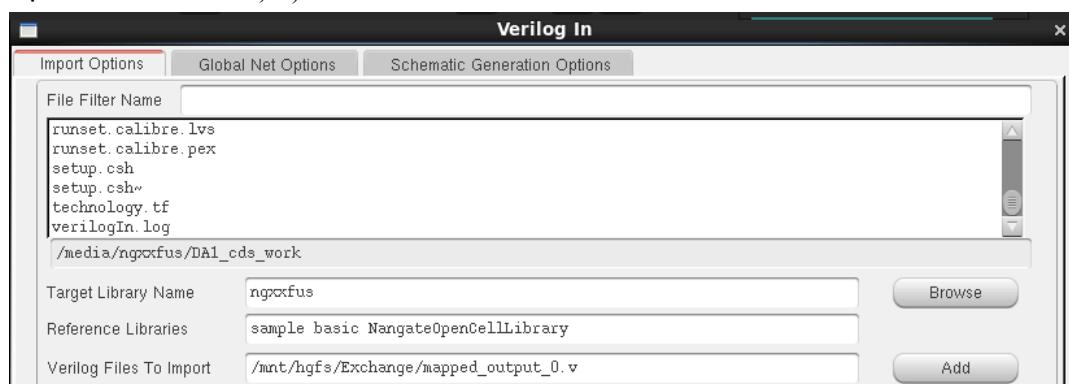
1.5. Nhập tệp verilog đã tổng hợp vào Cadence

Phần 1.5 minh họa cách nhập một file Verilog sau khi tổng hợp bởi Yosys và kiểm thử file Verilog được tổng hợp bởi Yosys. Để bắt đầu quá trình nhập file Verilog, trong Cadence Virtuoso, chọn File > Import > Verilog ... để mở hộp thoại Verilog In.



Bên trong hộp thoại Verilog In, điền vào 03 trường bắt buộc gồm:

- + Tên thư viện đích (Target Library Name): Thư viện nơi mô-đun sẽ được nhập vào.
- + Thư viện tham chiếu (Reference Libraries): Tất cả các thư viện mà Cadence sẽ tham chiếu tới.
- + Các tập tin Verilog cần nhập (Verilog Files To Import): Tập tin Verilog. (Lưu ý rằng các file Verilog phải là các file đã được ánh xạ đến thư viện ở mục **Reference Libraries**, nếu không sẽ không nhập được, hoặc nếu nhập được sẽ thiếu Schematic, ...)



- + Các trường khác là tùy chọn!

Nhấn OKE/YES để tiếp tục. Khi quá trình nhập hoàn tất, một hộp thoại khác sẽ xuất hiện để yêu cầu bạn hiển thị nhật ký khi nhập.



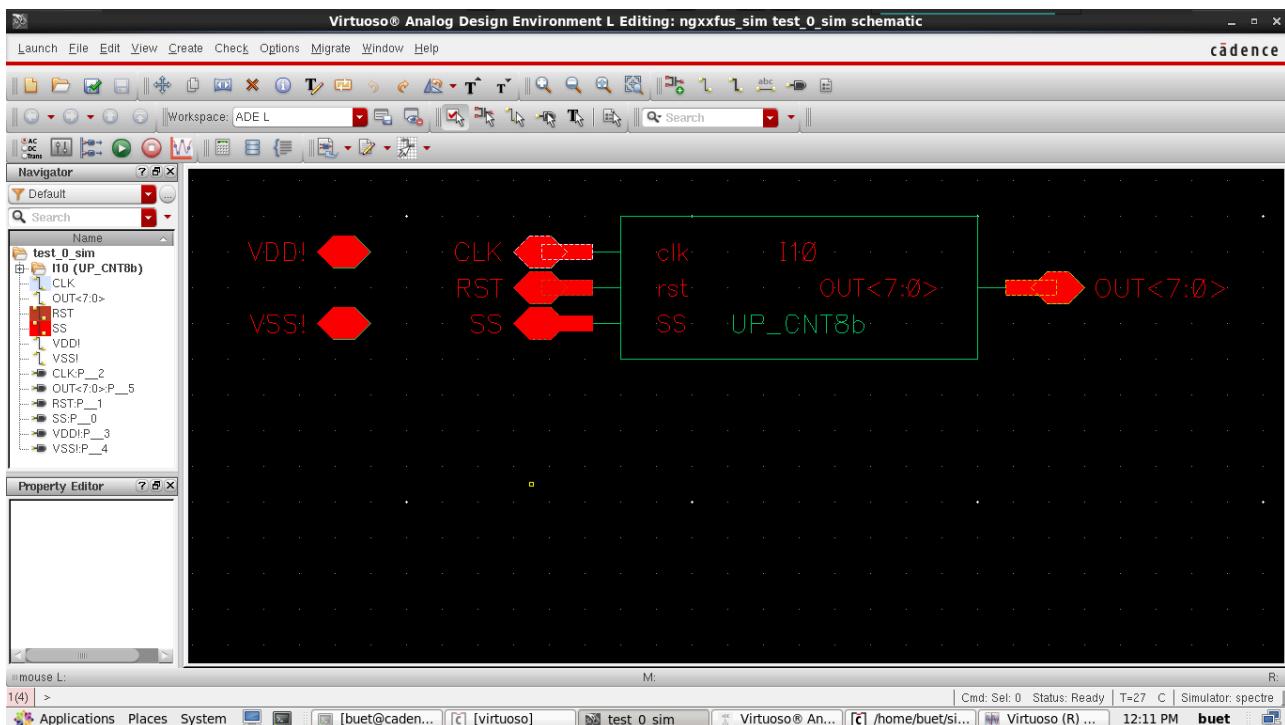
Nhập vào Có để đọc log.

```

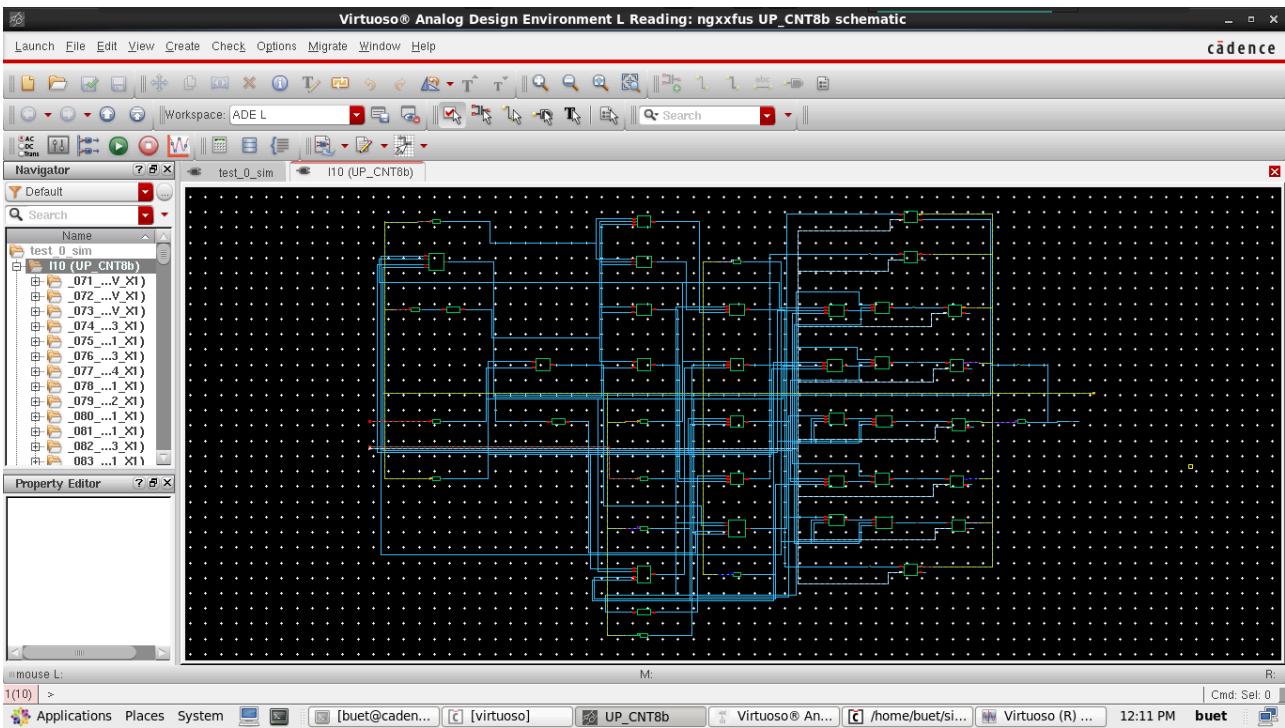
@(#) $CDS: ihdl version 6.1.5 12/12/2012 15:26 (sjfd1229) $ Wed Apr 16 12:10:09 2025
WARNING (VERILOGIN-19): The library, sample, is not specified in the cds.lib file. Add the library in the
cds.lib file and run the command again.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module NAND2_X1. Therefore using library
NangateOpenCellLibrary, cell NAND2_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module XNOR2_X1. Therefore using library
NangateOpenCellLibrary, cell XNOR2_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module INV_X1. Therefore using library
NangateOpenCellLibrary, cell INV_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module NOR2_X1. Therefore using library
NangateOpenCellLibrary, cell NOR2_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module OR2_X1. Therefore using library
NangateOpenCellLibrary, cell OR2_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module OAI21_X1. Therefore using library
NangateOpenCellLibrary, cell OAI21_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module OAI22_X1. Therefore using library
NangateOpenCellLibrary, cell OAI22_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module AND3_X1. Therefore using library
NangateOpenCellLibrary, cell AND3_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module AND2_X1. Therefore using library
NangateOpenCellLibrary, cell AND2_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module OR3_X1. Therefore using library
NangateOpenCellLibrary, cell OR3_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module MUX2_X1. Therefore using library
NangateOpenCellLibrary, cell MUX2_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module NOR3_X1. Therefore using library
NangateOpenCellLibrary, cell NOR3_X1, and view symbol as the symbol.
INFO (VERILOGIN-126): Unable to find the Verilog definition for module AOI221_X1. Therefore using library
NangateOpenCellLibrary, cell AOI221_X1, and view symbol as the symbol.
INFO (VERILOGIN-575): Symbol 'TheMultiplier' was not imported because it already exists in library 'ngxxfus'.
To overwrite the symbol, select an appropriate option from the Overwrite Symbol
Views drop-down list and try again.
INFO (VERILOGIN-376): Schematic for TheMultiplier exists in the library ngxxfus. Not imported since the overwrite
option is set to off.
INFO (VERILOGIN-206): End of Logfile.

```

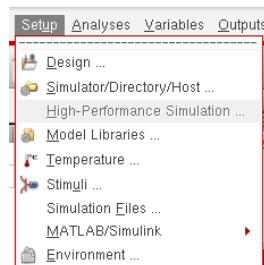
Tạo phiên bản (instance) của mô-đun đã nhập và bắt đầu mô phỏng bằng cách khởi chạy ADE L (Lưu ý rằng các cell bên trong thư viện cell chuẩn được cấp nguồn bởi nguồn global - kết thúc bằng dấu ! nên sẽ không xuất hiện chân VDD/VSS trên symbol, để cấp nguồn cho các cell này, chỉ cần tạo các chân (pin) và đặt tên VDD!/VSS! và cấp nguồn vào các chân này thì cũng sẽ cấp nguồn cho các cell bên trong). Hình bên dưới mô tả symbol của module UP_CNT8b:



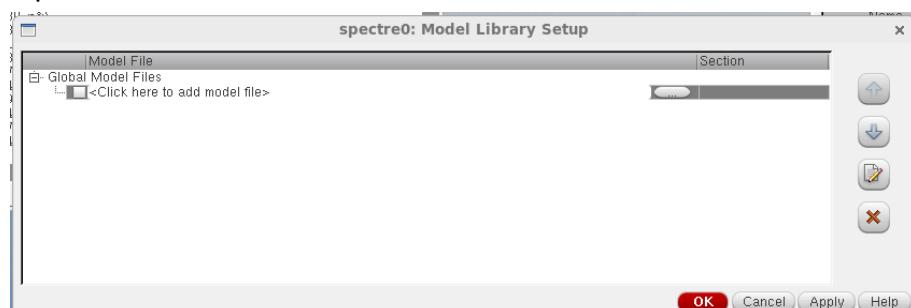
Sơ đồ nguyên lý sau khi nhập module UP_CNT8b:



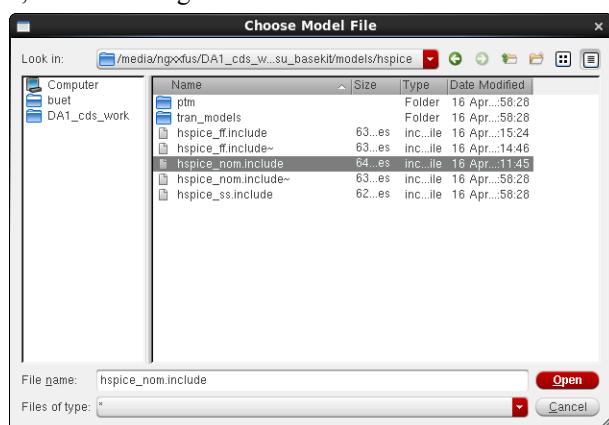
Để mô phỏng, trong ADE L, đầu tiên phải thiết lập model (Setup>Model Libraries):



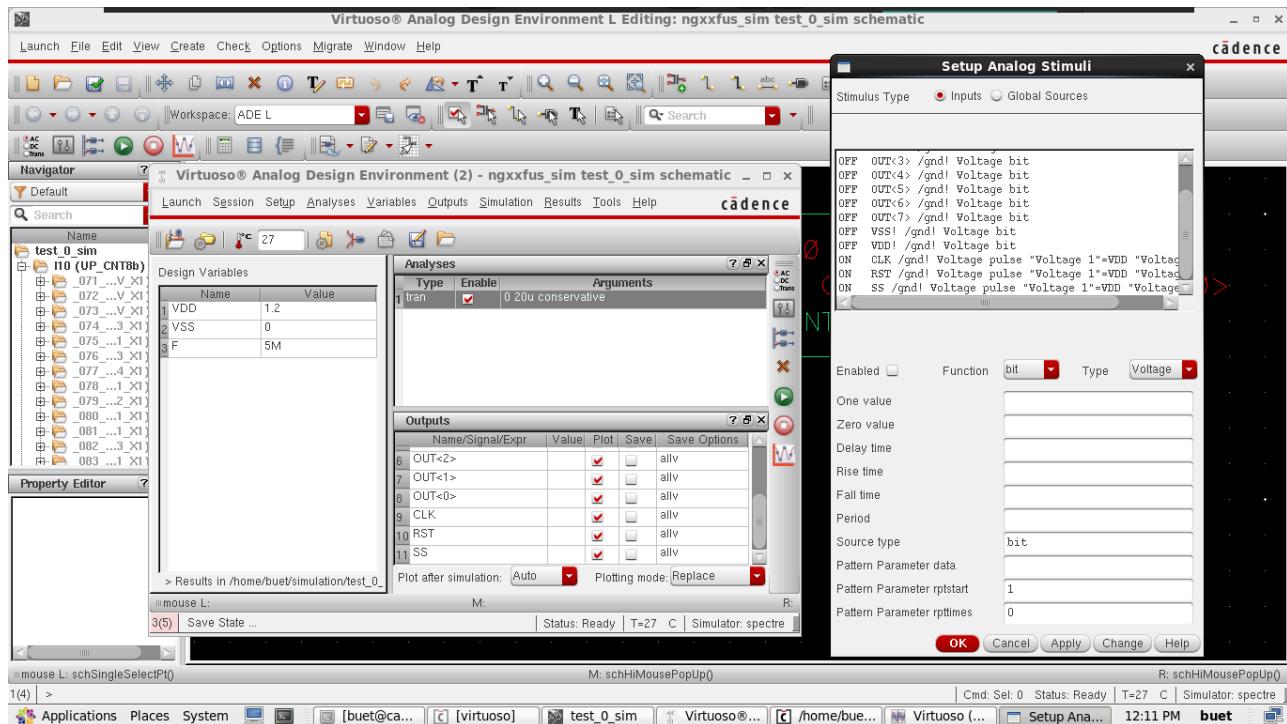
Sau đó chọn vào mục ... :



Một hộp thoại duyệt file hiện lên, tìm đến đường dẫn chứa model:



Cuối cùng thiết lập chế độ đánh giá, thời gian mô phỏng, ngõ ra và thực hiện mô phỏng:



Kết quả:



Ghi chú: Chi tiết về quá trình thêm thư viện cell tiêu chuẩn (Nangate 45nm Open Cell Library) và thư viện MOS/Model... (basekitNCSU).

2. Thêm thư viện FreePDK45 vào trong Cadence Virtuoso

2.1. Giới thiệu về FreePDK45

FreePDK45 là một Bộ công cụ thiết kế quy trình (PDK) giáo dục mã nguồn mở do Đại học NC State phát triển. Đây là một công nghệ trùu tượng nhằm mô phỏng quy trình chế tạo 45nm mà không bị ràng buộc với bất kỳ

xưởng đúc thực tế nào. Thư viện cho phép sinh viên, nhà nghiên cứu và nhà phát triển thực hành các luồng thiết kế ASIC kỹ thuật số/tương tự mà không cần NDA hoặc các hạn chế độc quyền.

2.2. Sao chép FreePDK45

Tạo bản sao thư viện FreePDK45 từ kho lưu trữ GitHub (<https://github.com/baichen318/FreePDK45.git>) của tác giả: baichen318. Sau khi tạo bản sau, cây thư mục của FreePDK45 được mô tả như bên dưới.

```
git clone https://github.com/baichen318/FreePDK45.git
Cloning into 'FreePDK45'...
remote: Enumerating objects: 1329, done.
remote: Counting objects: 100% (1329/1329), done.
remote: Compressing objects: 100% (703/703), done.
remote: Total 1329 (delta 544), reused 1317 (delta 534), pack-reused 0 (from 0)
Receiving objects: 100% (1329/1329), 8.42 MiB | 11.80 MiB/s, done.
Resolving deltas: 100% (544/544), done.
cd FreePDK45
tree -L 3
FreePDK45
├── APACHE-LICENSE-2.0.txt
│   ├── ncsu_basekit
│   │   ├── cdssetup
│   │   ├── doc
│   │   ├── examples
│   │   ├── gentech
│   │   ├── lib
│   │   ├── models
│   │   ├── README.txt
│   │   ├── skill
│   │   └── techfile
│   └── osu_soc
│       ├── flo
│       ├── lib
│       ├── README
│       └── ref_design
│           ├── README.txt
│           └── SVRF_EULA_06Feb09.txt
└── ncsu-FreePDK45-1.4.tar.gz.sig
    ├── README
    └── README.md

15 directories, 8 files
```

Hình . Sao chép FreePDK45 từ Github.

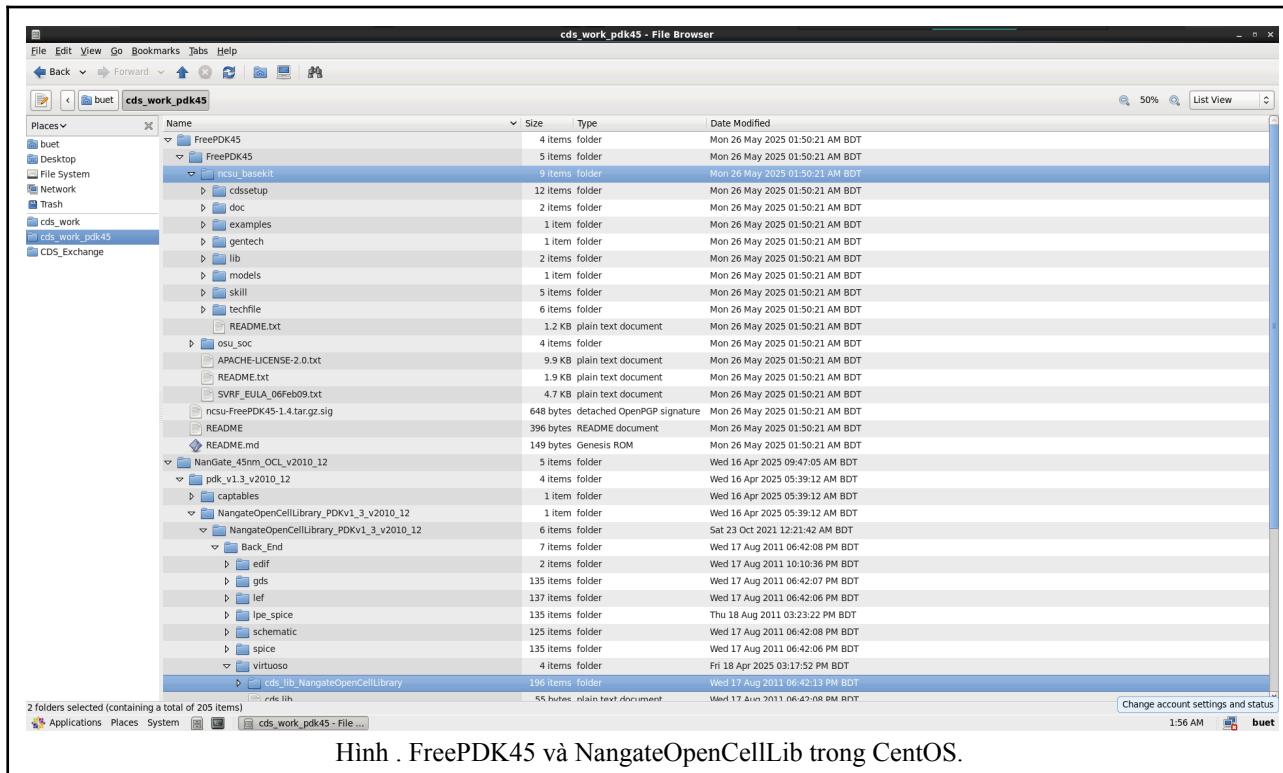
GHI CHÚ: Phiên bản này, tất cả các cổng (cell) trong thư viện cell tiêu chuẩn đi kèm có chế độ Symbol. Để xem được mạch nguyên lý được tổng hợp bởi Yosys cần sử dụng cell tiêu chuẩn khác.

(<https://si2.org/open-cell-and-free-pdk-libraries/>). Lúc này thư viện FreePDK45 ở trên chỉ sử dụng phần basekitNCSU cho việc mô phỏng MOSFET-level

2.3. Thêm FreePDK45, NangateOpenCellLib vào Cadence Virtuoso

2.3.1. Sao chép thư mục chứa FreePDK45, NangateOpenCellLib vào thư mục làm việc (cds_work)

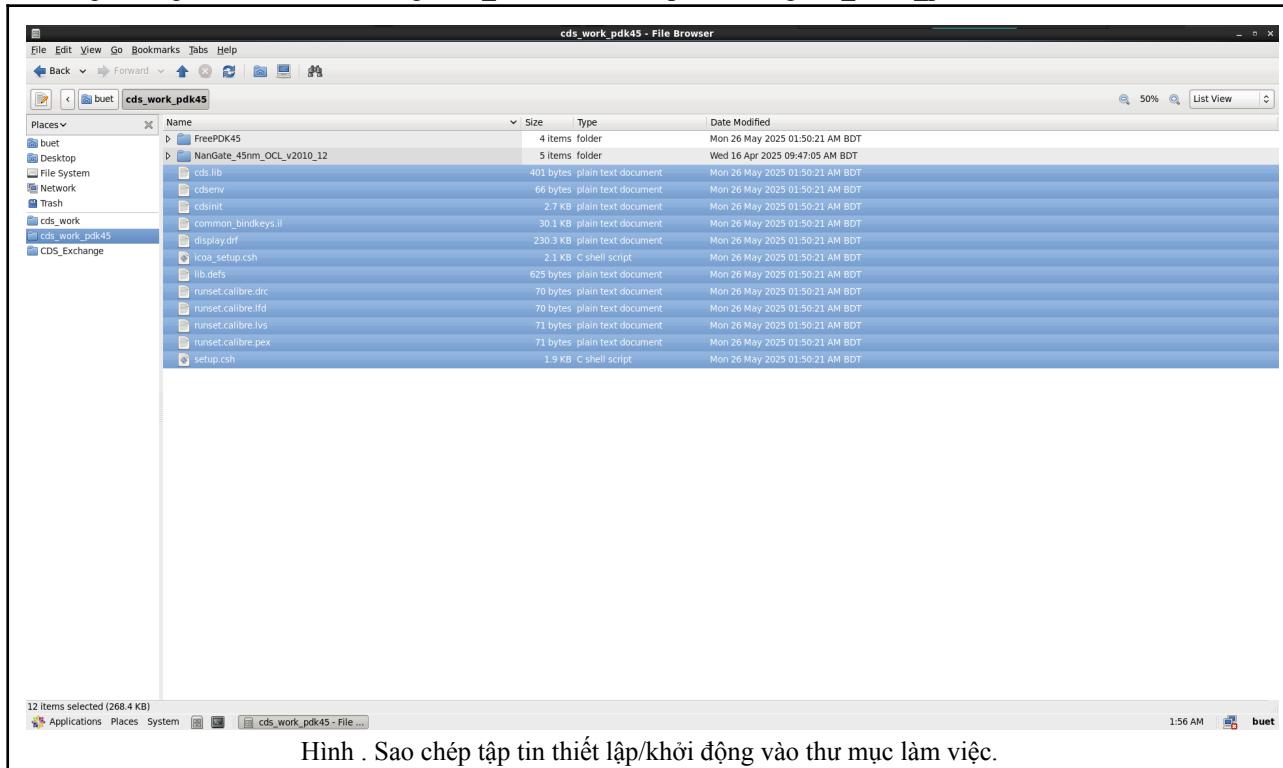
Sao chép thư mục FreePDK45 và NangateOpenCellLib vào CentOS (bằng cách kéo và thả vào cửa sổ VMware hoặc sao chép vào thư mục được chia sẻ).



Hình . FreePDK45 và NangateOpenCellLib trong CentOS.

2.3.2. Thiết lập thư viện FreePDK45

Sao chép các tập tin được lưu trữ trong **ncsu_basekit/cdssetup/** vào trong **cds_work_pdk45/**



Hình . Sao chép tập tin thiết lập/khởi động vào thư mục làm việc.

Thiết lập giá trị cho các biến môi trường cho thư viện FreePDK45:

- **CDSHOME** : /home/buet/cadence/IC615
- **PDK_DIR**: /home/buet/cds_work_pdk45/FreePDK45/FreePDK45
- **MGC_CALIBRE_DRC_RUNSET_FILE**=/media/ngxxfus/cds_work_pdk45/.runset.calibre.drc
- **MGC_CALIBRE_LVS_RUNSET_FILE**=/media/ngxxfus/cds_work_pdk45/.runset.calibre.lvs
- **MGC_CALIBRE_PEX_RUNSET_FILE**=/media/ngxxfus/cds_work_pdk45/.runset.calibre.pex

- PYTHONPATH=\$PYTHONPATH:\$PDK_DIR/ncsu_basekit/techfile/cni

Ghi chú:

- + Các thiết lập đường dẫn này dựa trên file **ncsu_basekit/cdsetup/setup.csh**.
- + Tùy thuộc vào vị trí lưu thư viện, thư mục làm việc mà hiệu chỉnh để phù hợp.
- + CDSHOME là biến môi trường đã được thiết lập từ trước, vui lòng kiểm tra kỹ trước khi thiết lập.

Đoạn mã bên dưới mô tả việc thêm các biến môi trường này vào .bashrc (Bash Shell) hoặc .zshrc (Zsh Shell):

```
export PDK_DIR=/media/ngxxfus/DA1_cds_work/FreePDK45/FreePDK45
export MGC_CALIBRE_DRC_RUNSET_FILE=/media/ngxxfus/cds_work_pdk45/.runset.calibre.drc
export MGC_CALIBRE_LVS_RUNSET_FILE=/media/ngxxfus/cds_work_pdk45/.runset.calibre.lvs
export MGC_CALIBRE_PEX_RUNSET_FILE=/media/ngxxfus/cds_work_pdk45/.runset.calibre.pex
export PYTHONPATH=$PDK_DIR/ncsu_basekit/techfile/cni
```

2.3.3. Thêm thư viện NangateOpenCellLibrary

Do cds.lib được sao chép từ FreePDK45, nên trong cds.lib chưa có thư viện NangateOpenCellLib, vì vậy cần sửa cds.lib để bổ sung thư viện NangateOpenCellLib. Bên trong thư mục chứa thư viện NangateOpenCellLib, tìm đến Back_End/virtuoso, thư mục này chứa một thiệp lập cho Cadence Virtuoso tương tự cái trong ncsu_basekit/cdsetup/ nhưng cds.lib trong Back_End/virtuoso đơn giản hơn nhiều. Lúc này sao chép đường dẫn đến thư viện NangateOpenCellLibrary (... /Back_End/virtuoso/cds_lib_NangateOpenCellLibrary) và thêm vào cds.lib ở cds_work_pdk45. Nội dung file cds.lib được mô tả bên dưới:

```
DEFINE analogLib $CDSHOME/tools/dfII/etc/cdslib/artist/analogLib
DEFINE US_8ths $CDSHOME/tools/dfII/etc/cdslib/sheets/US_8ths
DEFINE basic $CDSHOME/tools/dfII/etc/cdslib/basic
DEFINE cdsDefTechLib $CDSHOME/tools/dfII/etc/cdsDefTechLib
DEFINE NCSU_TechLib_FreePDK45 $PDK_DIR/ncsu_basekit/lib/NCSU_TechLib_FreePDK45
DEFINE NCSU_Devices_FreePDK45 $PDK_DIR/ncsu_basekit/lib/NCSU_Devices_FreePDK45

DEFINE NangateOpenCellLibrary
/home/buet/cds_work_pdk45/NanGate_45nm_OCL_v2010_12/pdk_v1.3_v2010_12/NangateOpenCellLibrary_PDK
v1_3_v2010_12/NangateOpenCellLibrary_PDKv1_3_v2010_12/Back_End/virtuoso/cds_lib_NangateOpenCellLibrary
```

GHI CHÚ: Để rút ngắn đường dẫn thư viện NangateOpenCellLibrary, có thể đặt biến môi trường hoặc di chuyển thư mục cds_lib_NangateOpenCellLibrary ra ngoài thư mục làm việc (cds_work_pdk45).

2.3.4. Thiết lập Cadence Virtuoso

File cdsinit sao chép từ ncsu_basekit/cdsetup/ sẽ làm cho Cadence Virtuoso trở nên rất "khác" vì file cdsinit này sẽ thay đổi các cấu hình phím tắt (key-blinding). Hiệu chỉnh cdsinit để đặt keyblind về mặt định. Ngoài ra cũng cần sửa file cdsinit để Cadence Virtuoso tải file display.drf ngay trong thư mục làm việc (cds_work_pdk45) thay vì trong ncsu_basekit/cdsetup/ như ban đầu. File cdsinit mới như sau:

```
envSetVal( "graphic" "drfPath" 'string "./display.drf")

if( getShellEnvVar("MGC_HOME") then
  loadi( strcat( getShellEnvVar("MGC_HOME") "/shared/pkgs/icv/tools/queryskl/calibre.OA.skl"))
) ;if

loadi( strcat( getShellEnvVar("PDK_DIR") "/ncsu_basekit/skill/loadSkill.il"))

(envLoadVals ?envFile "./cdsenv" ?tool "layout")

printf( strcat(
```

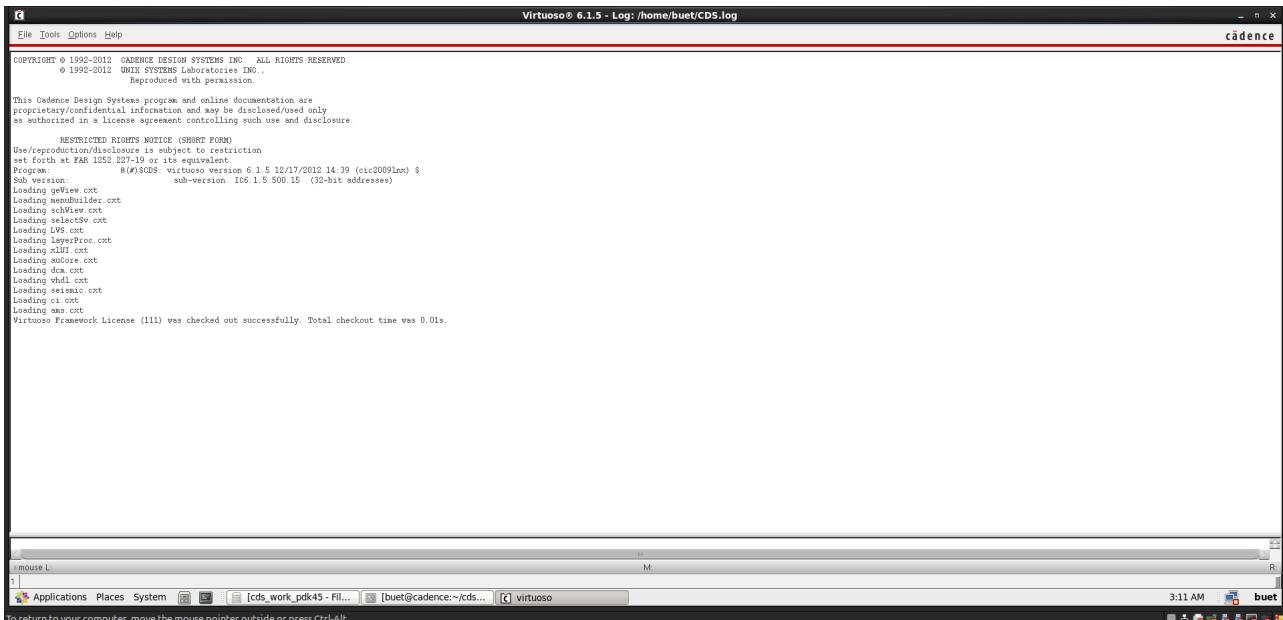
```
"-----\n"
"Welcome to the FreePDK 45nm Free, Open-Source Process Design Kit\n"
"\n"
"This initiative is brought to you by the Semiconductor Research\n"
"Corporation (SRC), the National Science Foundation (NSF), Silicon\n"
"Integration Initiative (Si2), Mentor Graphics, and Synopsys.\n"
"\n"
"This version of the kit was created by Rhett Davis, Paul Franzon,\n"
"Michael Bucher, Sunil Basavarajaiah, and Harun Demircioglu\n"
"of North Carolina State University, and James Stine and Ivan Castellanos\n"
"of Oklahoma State University.\n"
"\n"
"Contributions and modifications to this kit are welcomed and encouraged.\n"
"\n"
"Copyright 2007 - W. Rhett Davis, Paul Franzon, Michael Bucher,\n"
"and Sunil Basavarajaiah, North Carolina State University\n"
"Copyright 2008 - W. Rhett Davis, Michael Bucher, and Sunil Basavarajaiah,\n"
"North Carolina State University (ncsu_basekit subtree)\n"
"James Stine, and Ivan Castellanos,\n"
"and Oklahoma State University (osu_soc subtree)\n"
"Copyright 2011 - W. Rhett Davis, and Harun Demircioglu,\n"
"North Carolina State University\n"
"\n"
"Licensed under the Apache License, Version 2.0 (the \"License\");\n"
"you may not use this file except in compliance with the License.\n"
"You may obtain a copy of the License at\n"
"\n"
"  http://www.apache.org/licenses/LICENSE-2.0\n"
"\n"
"Unless required by applicable law or agreed to in writing, software\n"
"distributed under the License is distributed on an \"AS IS\" BASIS,\n"
"WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n"
"See the License for the specific language governing permissions and\n"
"limitations under the License.\n"
"-----\n"
"\n"
"Done loading FreePDK customizations.\n"
))
```

Sau khi sửa cdsinit, có thể loại bỏ một số file không còn dùng nữa.

2.3.5. Thiết lập Cadence Virtuoso (DRF)

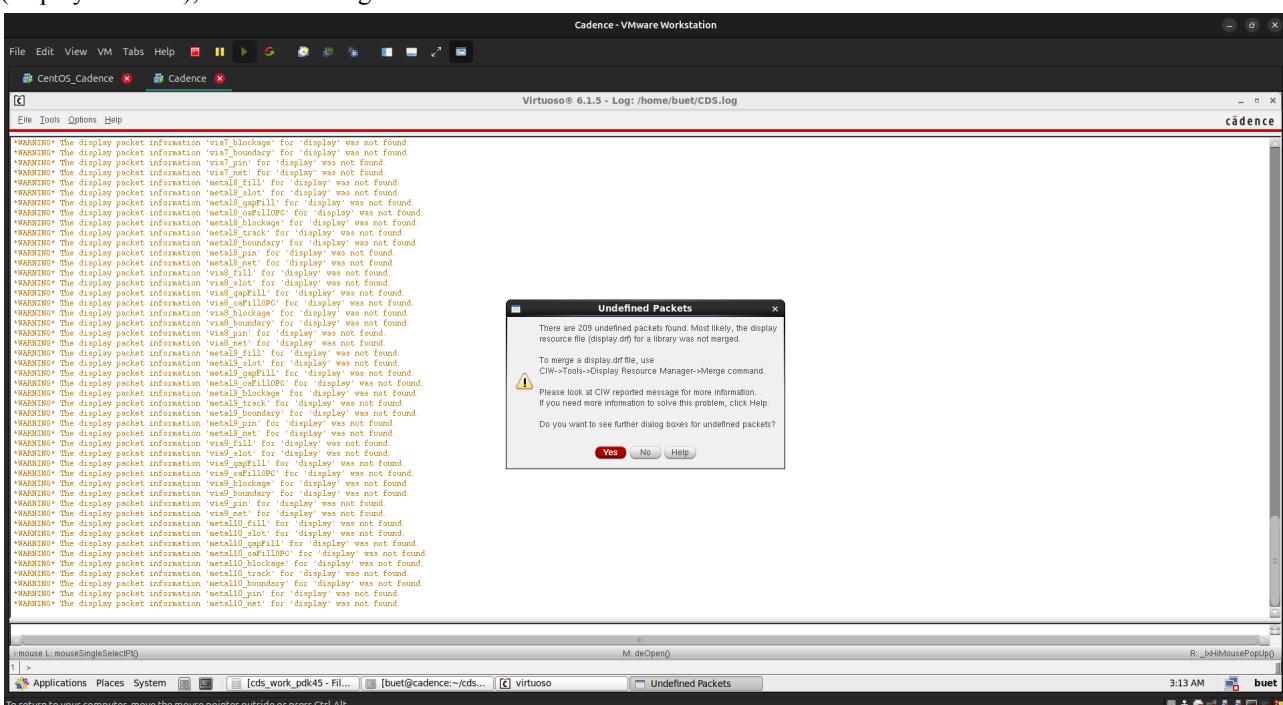
Sau khi thực hiện các bước từ 2.3.1 đến 2.3.4, về cơ bản có thể mở phông sử dụng hai thư viện trên Cadence Virtuoso, nhưng trong một số trường hợp sẽ gặp phải cảnh báo về file display.drf hoặc techfile.tf. Điều này xảy ra vì hai thư viện có đôi file display.drf và techfile.tf cho riêng mình, nên phải thực hiện việc gộp hai file này lại.

Đầu tiên, khởi động Cadence Virtuoso, mở thử mạch nguyên lý (schematic) của một con MOSFET bất kỳ hoặc một công logic bất kỳ để đảm bảo rằng các phần trước thiết lập đúng.



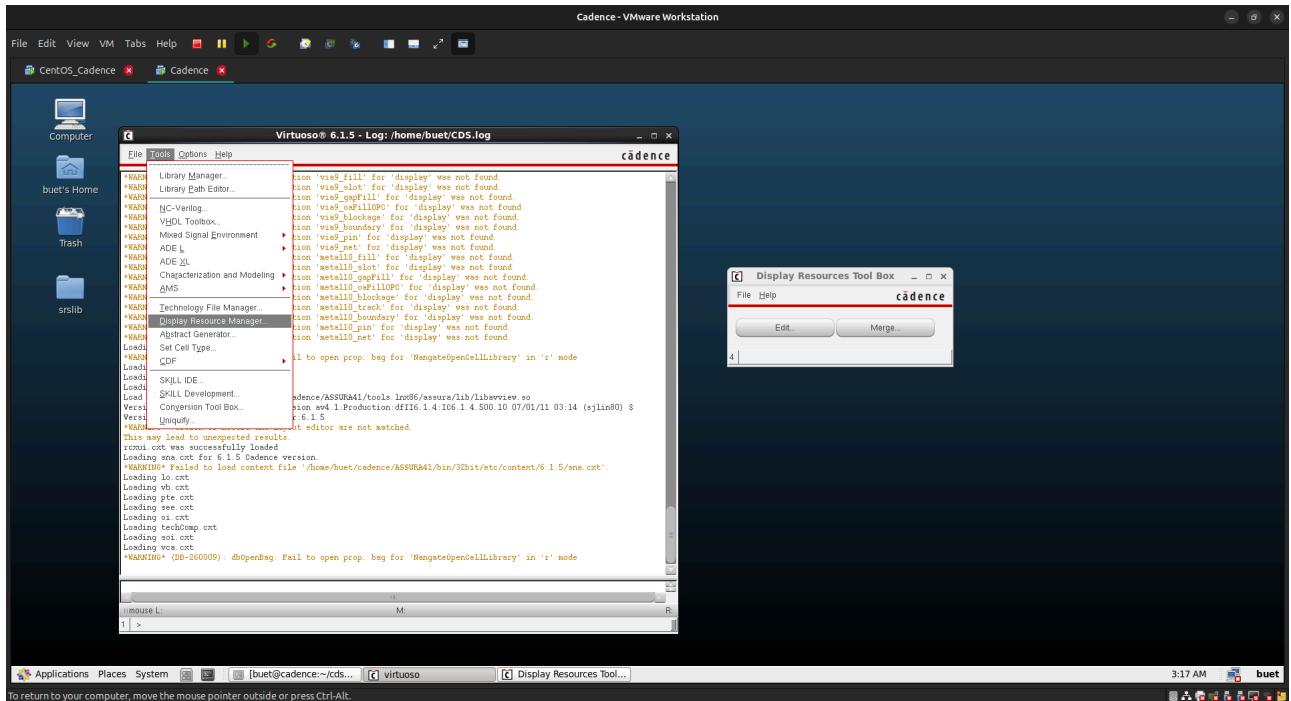
Hình . Giao diện chính của Cadence Virtuoso.

Thứ mở Abstract của một công AND3_X4, sẽ nhận thấy một hộp thoại cảnh báo về tài nguyên hiển thị (display resource), kèo theo hướng dẫn sửa:



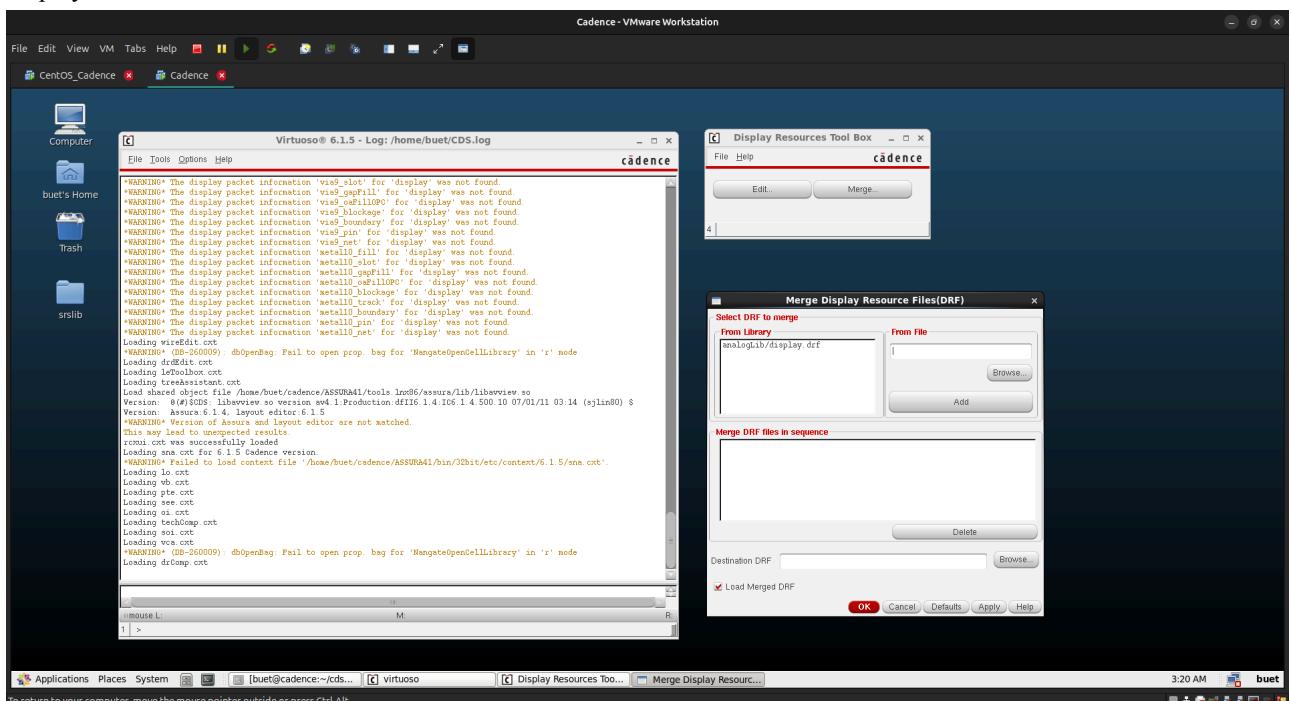
Hình . Hộp thoại thông báo thiếu file display.drf.

Dựa trên hướng dẫn của hệ thống: CIW > TOOLS > DISPLAY RESOURCE MANAGER > MERGE COMMAND. Trong đó CIW là cửa sổ chính của Virtuoso. Hình bên dưới mô tả cách mở hộp thoại Display Resource Tool Box.



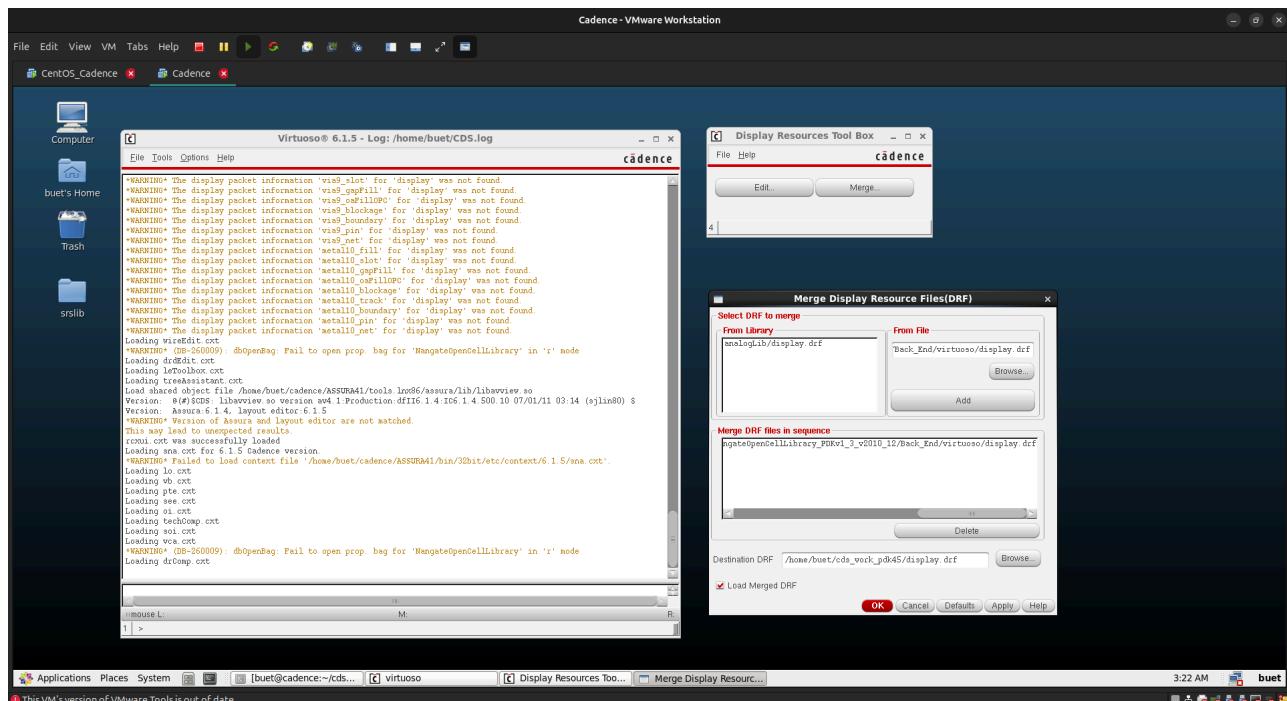
Hình . Mở hộp thoại Display Resources Tool Box.

Bên trong hộp thoại Display Resource Tool Box, chọn Merge, sau đó chọn OKE, sau đó hộp thoại Merge Display Resource Files hiện ra.



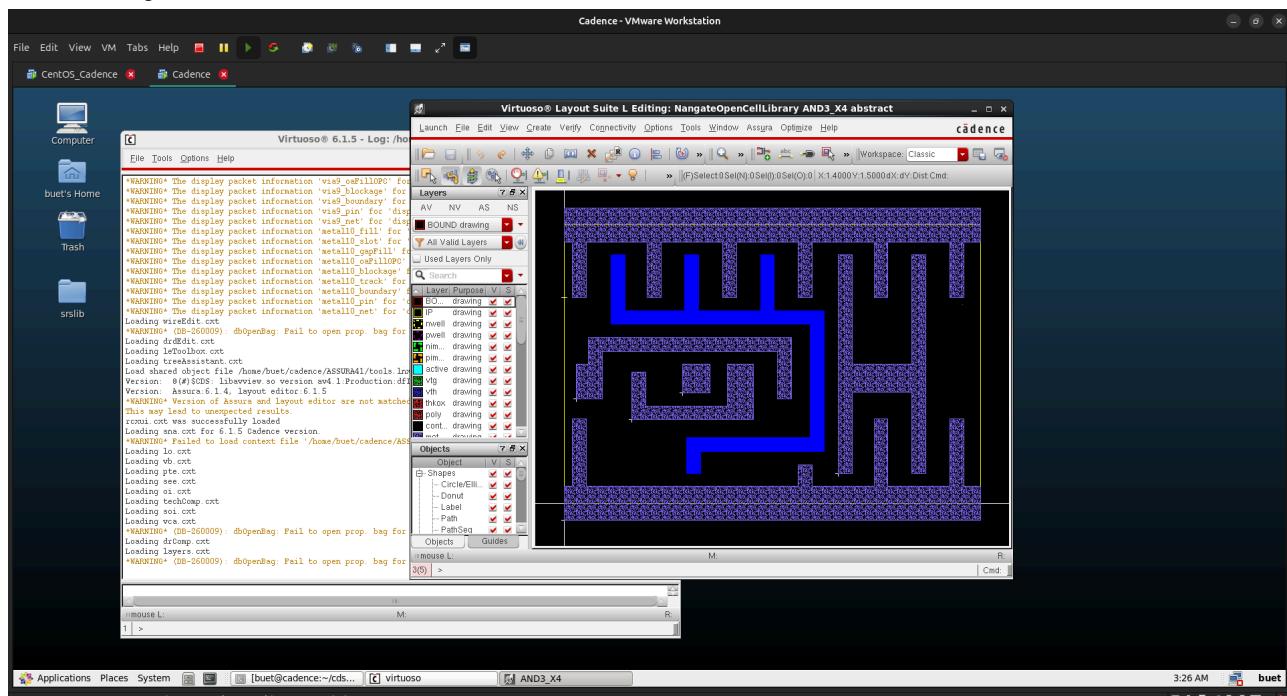
Hình . Mở hộp thoại Merge Display Resource Files (DRF).

Trong ô From File, chọn Browse, và chọn đến file display.drf trong Back_End/virtuoso/ sau đó bấm Add. Trong ô Destination DRF, chọn Browse, dẫn đến file display.drf ở thư mục làm việc hiện tại (cds_work_pdk45).



Hình . Chọn thư drf file trong hộp thoại Merge Display Resource Files (DRF).

Cuối cùng chọn OKE (tiếp tục chọn OKE nếu có hộp thoại hỏi về việc ghi đè). Mở lại công AND3_X4 để kiểm tra kết quả.

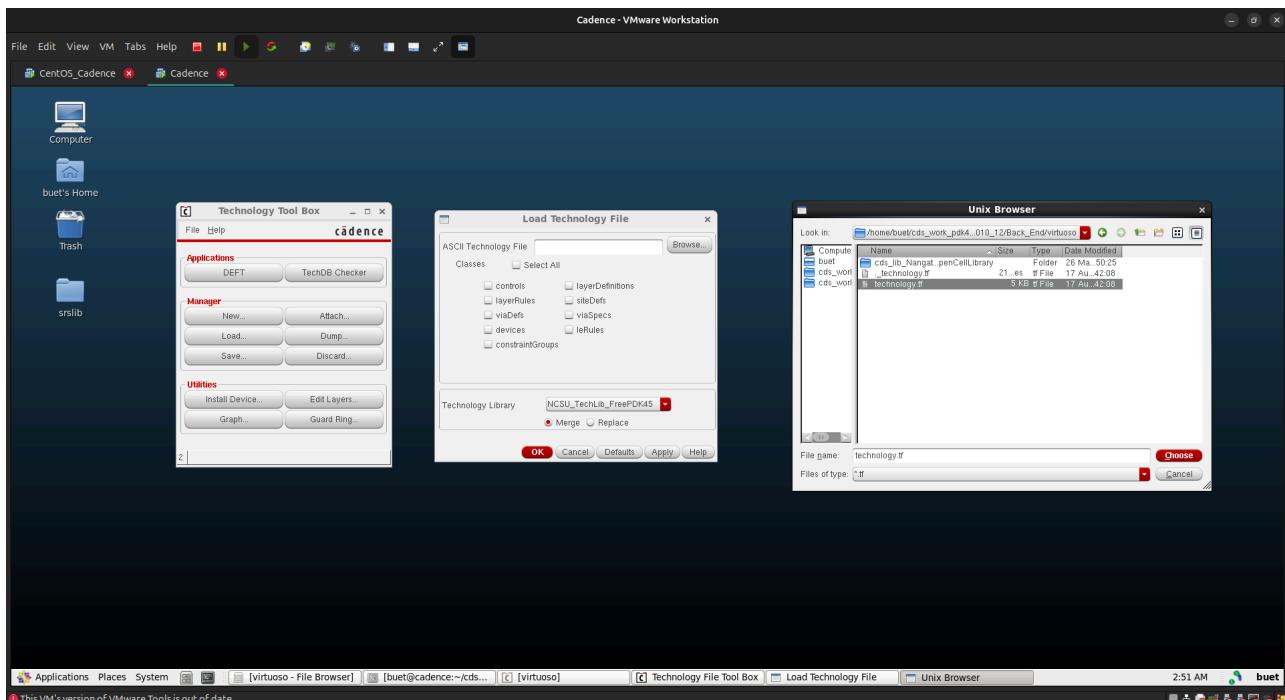


Hình . Abstract của cỗng AND3_X4.

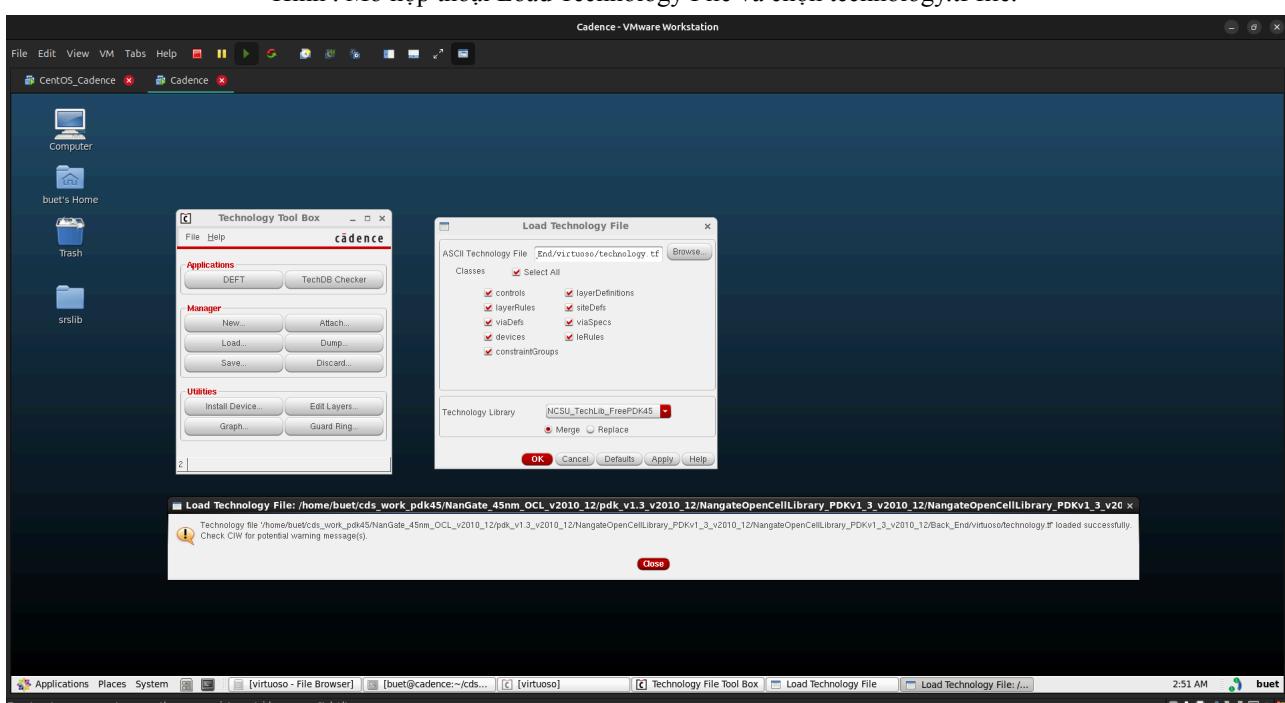
GHI CHÚ: Lúc này abstract của AND3_X4 đã có màu (khác với khi chưa thêm display.drf phù hợp thì chỉ hiện viền màu vàng và không có màu). Ngoài mở lớp abstract, có thể thử với layout.

2.3.6. Thiết lập Cadence Virtuoso (TF)

Việc gộp techfile cũng quan trọng như việc gộp display.drf, các bước thực hiện là tương đương. Các hình bên dưới mô tả vắng tắt quá trình gộp techfile.tf của thư viện NangateOpenCellLibrary vào FreePDK45.



Hình . Mở hộp thoại Load Technology File và chọn technology.tf file.



Hình . Gộp technology.tf file trong hộp thoại Load Technology File.

2.4. Kiểm tra thư viện FreePDK45

Thực hiện mô phỏng biểu thức $Y = (A\# \cdot B \cdot C) \#$ sử dụng cổng AND3_X4 trong thư viện NangateOpenCellLibrary và cổng Inverter được thiết kế ở phần sau.

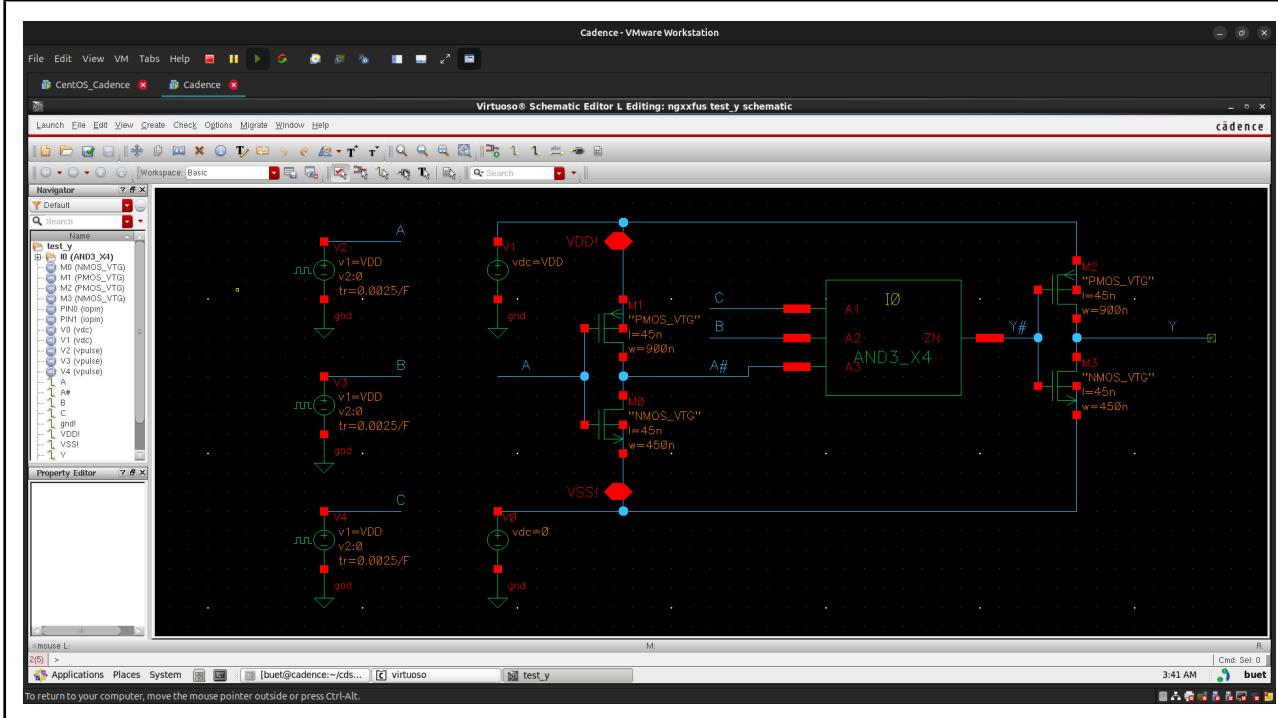
2.4.1. Mô phỏng một biểu thức Y

2.4.1.1. Sơ đồ nguyên lý

Bảng . Thuộc tính CMOS cho cỗng Inverter:

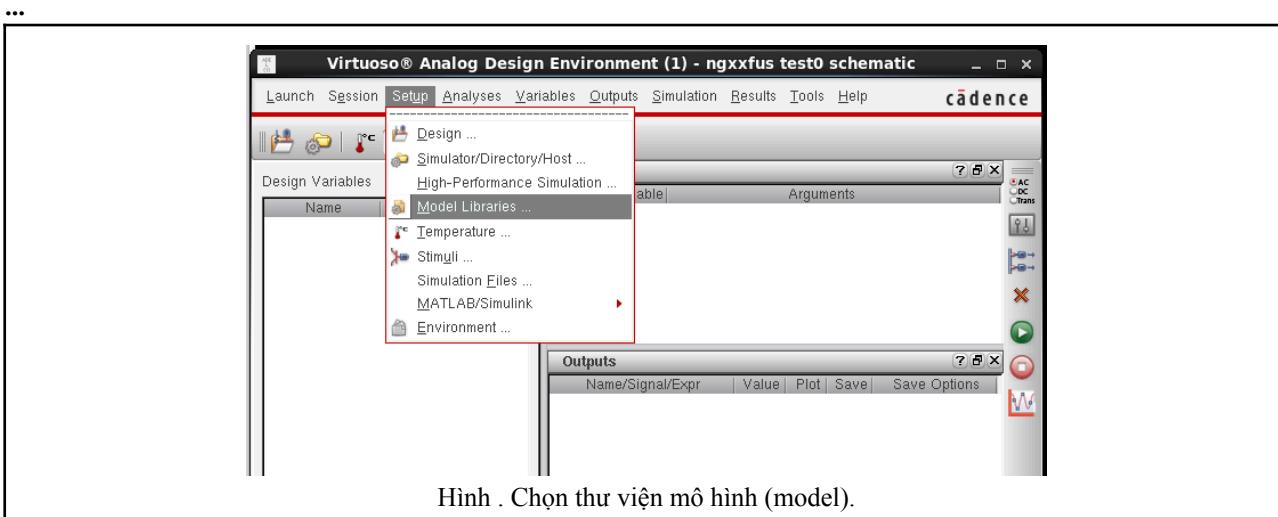
	W (nm)	L (nm)
pMOS	900	45
nMOS	450	45

Vẽ sơ đồ của NOT-GATE dựa trên bảng tính chất ở trên.



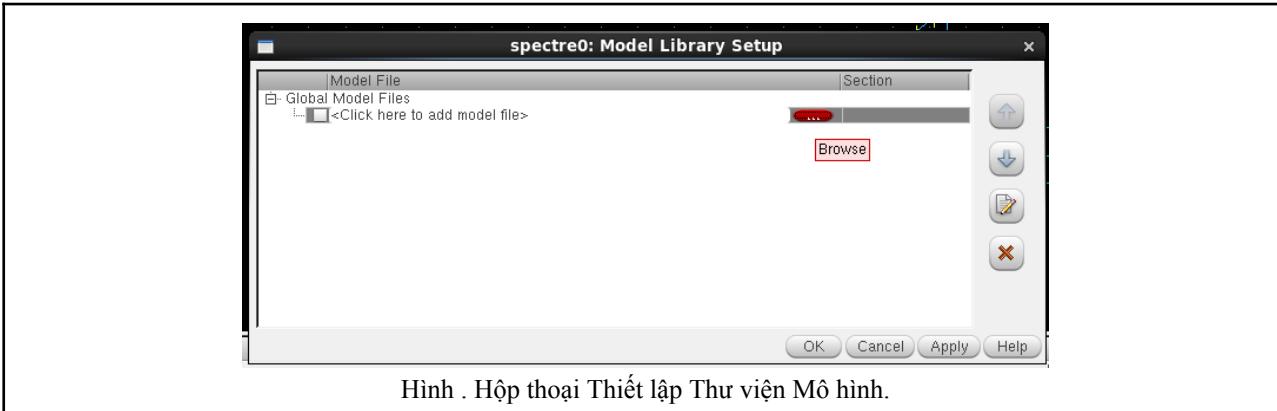
2.4.1.2. Thiết lập mô phỏng

Bước đầu tiên, khởi chạy `ADE L`, sau đó chọn mô hình bằng cách nhấp vào tab Setup, sau đó chọn Model Libraries ...



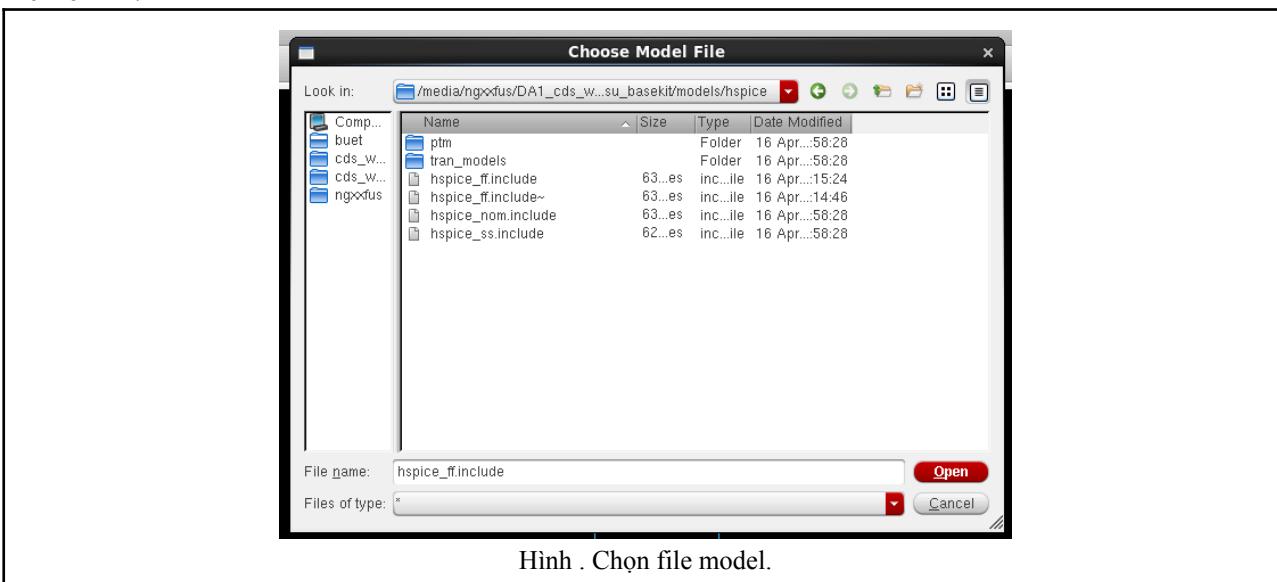
Hình . Chọn thư viện mô hình (model).

Một hộp thoại (Model Library Setup) sẽ xuất hiện, sau đó nhấp vào “...” để mở hộp thoại trình duyệt.



Hình . Hộp thoại Thiết lập Thư viện Mô hình.

Chọn file model theo đường dẫn `ncsu_basekit/models/hspice/` và chọn một tập tin mô hình. Sau đó chọn Open để mở file mô hình.



Hình . Chọn file model.

Nếu sau khi chọn mô hình và thiết lập mô phỏng như bình thường thì sẽ không thể mô phỏng được, và lỗi xuất ra ở log như hình bên dưới.

```
Cadence - VMware Workstation
File Edit View VM Tabs Help
CentOS_Cadence Cadence
/home/buet/simulation/test_y/spectre/schematic/psf/spectre.out
File Help
Includes RSA BSAFE(R) Cryptographic or Security Protocol Software from RSA Security, Inc.

User: buet Host: cadence HostID: Tf0100 PID: 6944
Memory available: 1536MB CPU: 1.0GHz
CPU Type: Intel(R) Core(TM) i5-3550M CPU @ 1.0GHz
Processor PhysicalID: CoreID Frequency: 0 0 1800 0
spectre 6944
Simulation 'input.scs' on cadence at 3:46:21 AM, Mon May 26, 2005 (process id: 6944).
Current working directory: /home/buet/simulation/test_y/spectre/schematic/netlist.
Command line: /home/buet/cadence/MNSIM121/tools/lncdf/spectre/spectre/J2005/spectre \
    -mnsim121 -cscd -nolog -noexit -nocheck -nocheckreg \
    -npnoption=spectrcd_e699_1 -format pifcd -rsw -p2 \
    -lqmax 900 -axm 5 -asm 5
spectre 6944
Loading /home/buet/cadence/MNSIM121/tools/lncdf/cmlib5/0/libbfinsen.cmlib5 ...
Loading /home/buet/cadence/MNSIM121/tools/lncdf/cmlib5/0/libphiphi_o.sh ...
Loading /home/buet/cadence/MNSIM121/tools/lncdf/cmlib5/0/libphiphi_sh.so ...
Loading /home/buet/cadence/MNSIM121/tools/lncdf/cmlib5/0/libtdmodels_sh.so ...
Loading /home/buet/cadence/MNSIM121/tools/lncdf/cmlib5/0/libtdmodels_sh.so ...
Reading file /home/buet/cadence/MNSIM121/tools/lncdf/spectre/schematic/netlist/input.scs
Reading file /home/buet/cadence/MNSIM121/tools/lncdf/spectre/schematic/netlist/input.scs
spectre 6944
Error found by spectre during circuit read-in.
ERROR (SPK-668): '/home/buet/cds_work/pd45/freePD45/ncsu_basekit/hndels/hspice/hspice_nom.include' 2: Unable to open input file '/home/buet/cds_lib_FreePD45/ncsu_basekit/models/hspice/tran_models/models_ncm/NMOS_VT0.inc'
No such file or directory.
ERROR (SPK-668): '/home/buet/cds_work/pd45/freePD45/ncsu_basekit/models/hspice/hspice_nom.include' 5: Unable to open input file '/home/buet/cds_lib_FreePD45/ncsu_basekit/models/hspice/tran_models/models_ncm/NMOS_VTL.inc'
No such file or directory.
ERROR (SPK-668): '/home/buet/cds_work/pd45/freePD45/ncsu_basekit/models/hspice/hspice_nom.include' 8: Unable to open input file '/home/buet/cds_lib_FreePD45/ncsu_basekit/models/hspice/tran_models/models_ncm/NMOS_VTH.inc'
No such file or directory.
ERROR (SPK-668): '/home/buet/cds_work/pd45/freePD45/ncsu_basekit/models/hspice/hspice_nom.include' 11: Unable to open input file '/home/buet/cds_lib_FreePD45/ncsu_basekit/models/hspice/tran_models/models_ncm/NMOS_THROX.inc'
No such file or directory.

Time for parsing: CPU = 584.13 ms, elapsed = 1.32035 s.
The accumulated CPU = 584.13 ms, total elapsed = 1.32035 s.
Peak resident memory used = 30.5 Mbytes.

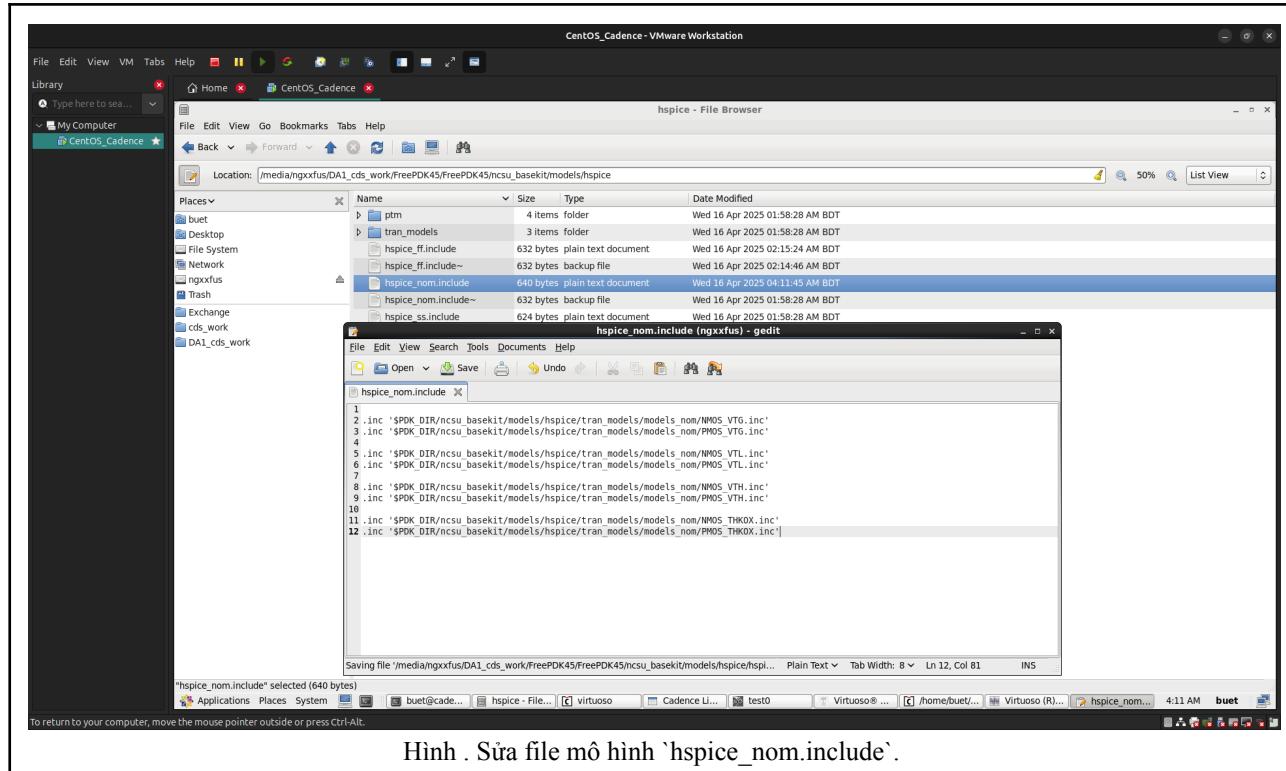
Time for parsing: CPU = 0 s, elapsed = 39.8159 us.
The accumulated CPU = 0 s, total elapsed = 1.320076 s.
Peak resident memory used = 30.5 Mbytes.

Aggregate audit (3:46:23 AM, Mon May 26, 2005)
Time spent in SPK: 1.32035 s, total = 41.98 s.
Time spent in licensing: elapsed = 480 us, percentage of total = 34%.
Peak memory used = 30.5 Mbytes.
Start time: 3:46:21 AM, Mon May 26, 2005, ended at: 3:46:23 AM, Mon May 26, 2005, with elapsed time (wall clock): 1.35 s.
spectre completes with 4 errors, 0 warnings, and 0 notices.
spectre terminated prematurely due to fatal error.

[  ] Applications Places System [ ] buet [ ] virtuoso [ ] test_y [ ] Virtuoso® Analog Des... [ ] /home/buet/simulation...
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
3:46 AM
```

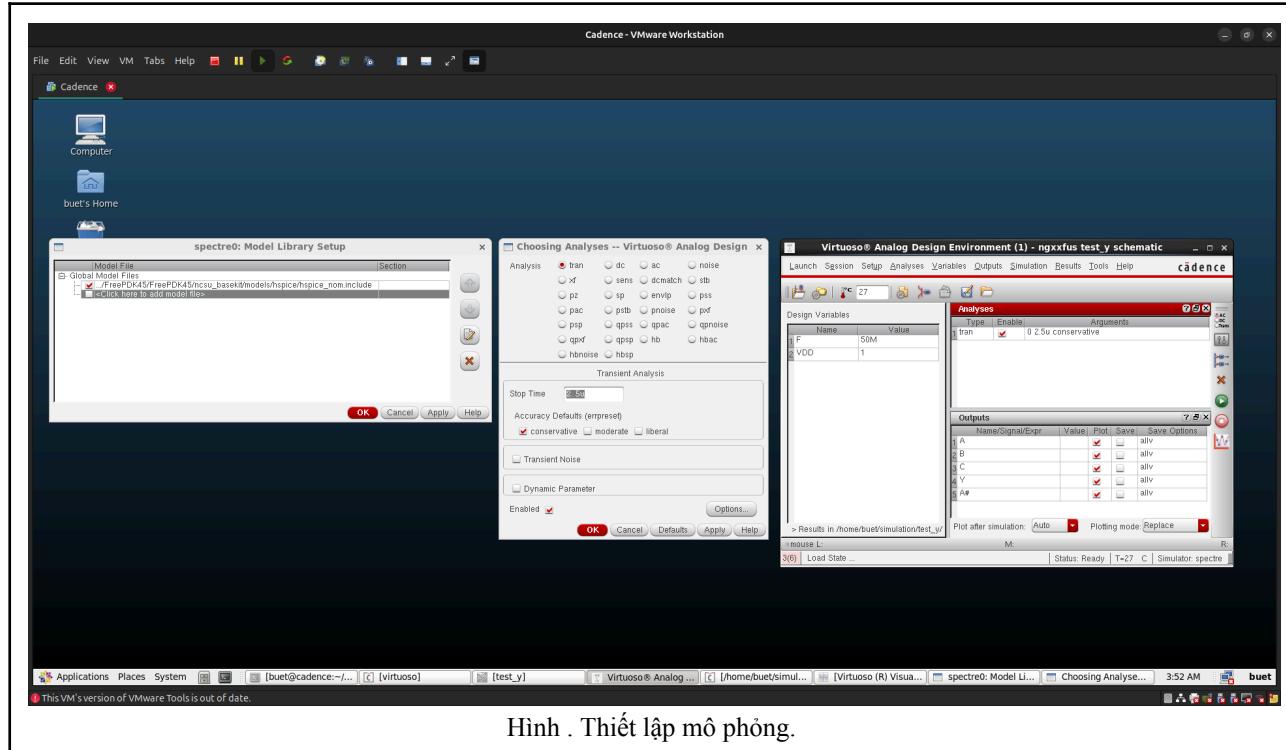
Hình . Thông báo lỗi từ file model vừa chọn ở trên.

Đây mà một lỗi do quên đóng dấu ' trong file model.include. Chỉ cần mở file model.include và bổ sung dấu ' còn thiếu thì có thể mô phỏng được. Phiên bản đã sửa cho file mô hình được hiển thị trong Hình bên dưới.



Hình . Sửa file mô hình 'hspice_nom.include'.

Cuối cùng, chọn Analyses (2,5u - Conservative); Variables (F=50M, VDD=1); và mô phỏng với spectre (ADE).



Hình . Thiết lập mô phỏng.

2.4.1.3. Kết quả mô phỏng

Kết quả của mô phỏng:



Hình . Dạng sóng ngõ ra của hàm Y trên Cadence Virtuoso.

3. Tổng hợp hai bộ nhân 8-bit sử dụng Yosys

3.1. Tổng hợp bộ nhân dùng CLA

3.1.1. Chuẩn bị

Trước khi sử dụng Yosys để tổng hợp, cần chuẩn bị các thông tin sau:

- + Tên top-module: Điều này rất quan trọng, vì nếu không có sẽ không tổng hợp được.
- + Danh sách các file Verilog có trong thiết kế. Yosys yêu cầu nhập tất cả các file .v liên quan.
- + Thư viện cell tiêu chuẩn, yosys có thể không cần thư viện cell tiêu chuẩn để tổng hợp, nhưng sẽ cần cell tiêu chuẩn để ánh xạ sau này, thuận tiện cho việc mô phỏng.

Có thể lấy cá danh sách file Verilog thông qua lệnh find của CentOS, hình bên dưới mô tả cách sử dụng lệnh find để liệt kê tất cả các file Verilog có trong thư mục hiện tại (dấu * trong chuỗi so khớp cho biết rằng bất cứ tên nào kết thúc bằng .v đều khớp):

```
find -name "*.v"
```

```
root@xxfus-X409PA: [14:49:32] [Eight-bits-Exact-Signed-Multiplier-with-CLA]
ls
AND_GATE.v      CarryLookaheadAdder4bits.prj      FirstRow.v      GroupA_8x4.v      PartialProductUnit.prj      TestMultiplier.v
CarryLookahead4bitsSumComponent.prj   CarryLookaheadAdder4bits.stx      FirstRow.xst     GroupA_8x4.xst      PartialProductUnit.stx      TheMultiplier.cmd_log
CarryLookahead4bitsSumComponent.stx   CarryLookaheadAdder4bits.syr      FullAdder1bit.cmd_log    GroupB_8x4.prf      PartialProductUnit.v      TheMultiplier.envsettings.html
CarryLookahead4bitsSumComponent.v     CarryLookaheadAdder4bits.v      FullAdder1bit.ensettings.html GroupB_8x4.xst      PartialProductUnit.kst      TheMultiplier.iso
CarryLookahead4bitsSumComponent.xst   CarryLookaheadAdder4bits.xst      FullAdder1bit.grp      GroupB_8x4.xst      Row0fPartialProductUnit.kst  TheMultiplier.ngc
CarryLookaheadAdder2bits.cmd_log     CarryLookaheadAdder4bits.xst      FullAdder1bit.ngc      GroupB_8x4.xst      Row0fPartialProductUnit.v  TheMultiplier.ngf
CarryLookaheadAdder2bits.ngc        CarryLookaheadAdder4bits.xrpt     FullAdder1bit.ngr      HalfAdder1bit.v      Row0fPartialProductUnit.v  TheMultiplier.prj
CarryLookaheadAdder2bits.ngv        CarryLookaheadAdder4bits.xrpt     FullAdder1bit.prj      ignore_all          Row0fPartialProductUnit.xst TheMultiplier.stx
CarryLookaheadAdder2bits.lso        CarryLookaheadAdderLogicCircuit.prj FullAdder1bit.stx      ignore_all          Row0fPartialProductUnit.xst TheMultiplier.summary.html
CarryLookaheadAdder2bits.ngt        CarryLookaheadAdderLogicCircuit.v FullAdder1bit.v      isim.cmd            TestCL12bits_bmh.prj  TheMultiplier.tcl
CarryLookaheadAdder2bits.ngv        CarryLookaheadAdderLogicCircuit.v FullAdder1bit.v      isim.log             TestCL12bits_isim_bmh.exe TheMultiplier.tsr
CarryLookaheadAdder2bits.ngt        CarryLookaheadPandCreate.prj     FullAdder1bit.v      isim.log             TestCL12bits_isim_bmh.wb  TheMultiplier.xst
CarryLookaheadAdder2bits.ngv        CarryLookaheadPandCreate.v      FullAdder1bit.v      isim.log             TestCL12bits_isim_bmh.wb  TheMultiplier.xt
CarryLookaheadAdder2bits.summary.html CarryLookaheadPandCreate.v      FullAdder1bit.v      LastRowOfGroupB.prj  TestCL12bits_stx_Beh.prj  TestCL12bits_stx_xrpt
CarryLookaheadAdder2bits.syr        CarryLookaheadPandCreate.xt      FullAdder1bit.v      LastRowOfGroupB.stx  TestCL12bits_v           webtalk_pk.xml
CarryLookaheadAdder2bits.v         CarryLookaheadPandCreate.xt      FullAdder1bit.xst     LastRowOfGroupB.v   Test4k4bits_isim_beh.exe xilinxsim.ini
CarryLookaheadAdder2bits.v         Eight-bits-Exact-Signed-Multiplier-with-CLA.gise FullAdder1bit.xst     LastRowOfGroupB.xst  Test4k4bits_isim_beh.wb  xilinxsim.v
CarryLookaheadAdder2bits.v         Eight-bits-Exact-Signed-Multiplier-with-CLA.kst FullAdder1bit.xst     LastRowOfGroupB.xst  Test4k4bits_stx_Beh.prj
CarryLookaheadAdder2bits.xt.krt     Eight-bits-Exact-Signed-Multiplier-with-CLA.xise fuse.log      NAND_GATE.v       TestCL12bits_isim_bmh.prj
CarryLookaheadAdder2bits.cmd_log   FirstRow.cmd_log                fuseRelaunch.cmd    PartialProductUnit.N.prj TestMultiplier.bah.prj
CarryLookaheadAdder2bits.lso       FirstRow.lso                  fuse.xmsgs          PartialProductUnit.N.stx TestMultiplier.isim_bah.exe
CarryLookaheadAdder2bits.ngc       FirstRow.ngc                  GroupA_8x4.prf      PartialProductUnit.N.v  TestMultiplier.isim_bah.wb
CarryLookaheadAdder2bits.ngt       FirstRow.ngt                  GroupA_8x4.stx      PartialProductUnit.N.xst TestMultiplier.isim_bah.wb
CarryLookaheadAdder2bits.ngv       FirstRow.ngv                  GroupB_8x4.prf      TestMultiplier.stx_Beh.prj
CarryLookaheadAdder2bits.ngt       FirstRow.ngt                  GroupB_8x4.stx      TestMultiplier.stx_Beh.prj
CarryLookaheadAdder2bits.ngv       FirstRow.ngv                  GroupC_8x4.prf      TestMultiplier.stx_xrpt
CarryLookaheadAdder2bits.ngt       FirstRow.ngt                  GroupC_8x4.stx      TestMultiplier.stx_xrpt
CarryLookaheadAdder2bits.ngv       FirstRow.ngv                  GroupD_8x4.prf      TestMultiplier.stx_v
CarryLookaheadAdder2bits.ngt       FirstRow.ngt                  GroupD_8x4.stx      TestMultiplier.stx_v
CarryLookaheadAdder2bits.ngv       FirstRow.ngv                  GroupE_8x4.prf      TestMultiplier.v
CarryLookaheadAdder2bits.ngt       FirstRow.ngt                  GroupE_8x4.stx      TheMultiplier.v
root@xxfus-X409PA: [14:49:35] [Eight-bits-Exact-Signed-Multiplier-with-CLA]
```

Hình . Sử dụng lệnh find để liệt kê tất cả các file Verilog.

Tuy nhiên, đây chưa phải đầu ra mong muốn, cần loại bỏ thêm ./ bằng cách sử dụng steam-editor (sed). Hình bên dưới mô tả cách loại bỏ ./ bằng lệnh sed. Ý nghĩa của `s|.|/||` bên trong lệnh sed là thay thế (substitution) ./ bởi chuỗi rỗng.

```
find -name "*.v" | sed "s|.|/||"
```

```
nxxfus@nxxfus-X409FA [14:52:40] [Eight-bits-Exact-Signed-Multiplier-with-CLA]
$ find -name "*.v" | sed "s|.|/||"
./AND_GATE.v
./CarryLookahead4bitsSumComponent.v
./CarryLookaheadAdder4bits.v
./CarryLookahead1bit.v
./CarryLookaheadDandGcreate.v
./Eight-bits-Exact-Signed-Multiplier-with-CLA.v
./CarryLookaheadAdder12bits.v
./FirstRow.v
./LastRowOfGroupB.v
./FullAdder1bit.v
./Fulladder1bit.v
./GroupA_8x4.v
./GroupB_8x4.v
./Halfadder1bit.v
./NAND_GATE.v
./PartialProductUnit.v
./PartialProductUnit_N.v
./RowOfPartialProductUnit.v
./TestCLAI2bits.v
./TestCL4bits.v
./TestCL4abits.v
./TheMultiplier.v
./TheMultiplier.v
nxxfus@nxxfus-X409FA [14:52:43] [Eight-bits-Exact-Signed-Multiplier-with-CLA]
$ find -name "*.v" | sed "s|.|/||"
AND_GATE.v
CarryLookahead4bitsSumComponent.v
CarryLookaheadAdder12bits.v
CarryLookaheadDandGcreate.v
CarryLookaheadDandGcreate.v
Eight-bits-Exact-Signed-Multiplier-with-CLA.v
CarryLookaheadAdder12bits.v
FirstRow.v
LastRowOfGroupB.v
FullAdder1bit.v
Fulladder1bit.v
GroupA_8x4.v
GroupB_8x4.v
Halfadder1bit.v
NAND_GATE.v
PartialProductUnit.v
PartialProductUnit_N.v
RowOfPartialProductUnit.v
TestCLAI2bits.v
TestCL4abits.v
TestMultiplier.v
TheMultiplier.v
TheMultiplier.v
nxxfus@nxxfus-X409FA [14:52:46] [Eight-bits-Exact-Signed-Multiplier-with-CLA]
$
```

Hình . Loại bỏ ./ bằng lệnh sed.

Sau cùng, cần chọn lọc một lần nữa danh sách các file Verilog thực sự nằm trong thiết kế và thêm và mẫu TCL ở phần 1.3.

3.1.2. Xây dựng TCL script để tổng hợp mạch nhân với CLA

```
1 # IMPORTANT: Every directory in this script must be absolute paths and end with a '/'
2 # E.g.: '/home/user/dir' -----> '/home/user/dir/synth_output.v'
3 #       '/home/user/dir' -----> '/home/user/dir/synth_output.v'
4
5 set working_dir "/home/nxxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/"
6 set output_dir "/home/nxxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/"
7 set lib_path "/home/nxxxfus/Downloads/NangateOpenCellLibrary_PDKv1_3_v2010_12/Front_End/Liberty/NLDM/NangateOpenCellLibrary_typical.lib"
8 set synth output "synth mul_1.v"
9 set mapped output "mapped mul_1.v"
10 set top module "TheMultiplier"
11 set top_file "TheMultiplier"
12 set顶层设计 files {
13     AND_GATE.v
14     CarryLookahead4bitsSumComponent.v
15     CarryLookaheadAdder12bits.v
16     CarryLookaheadAdder4bits.v
17     CarryLookaheadDandGcreate.v
18     CarryLookaheadDandGcreate.v
19     FirstRow.v
20     FullAdder1bit.v
21     GroupA_8x4.v
22     GroupB_8x4.v
23     Halfadder1bit.v
24     LastRowOfGroupB.v
25     NAND_GATE.v
26     PartialProductUnit.v
27     PartialProductUnit_N.v
28     RowOfPartialProductUnit.v
29     TestCLAI2bits.v
30     TestCL4abits.v
31     TestMultiplier.v
32     TheMultiplier.v
33 }
34
35 foreach file $verilog_file {
36     yosys read_verilog $working_dir$file
37 }
38 yosys hierarchy -check -top $top_module
39 yosys synth -top $top_module -flatten
40 yosys write_verilog -noattr $output_dir$synth_output
41 yosys read liberty -lib $lib_path
42 yosys opt
43 yosys dfflibmap -liberty $lib_path
44 yosys abc -liberty $lib_path
45 yosys techmap -map +/techmap.v
46 yosys check
47 yosys write verilog -noattr $output_dir$mapped_output
48
```

Hình . TCL Script để tổng hợp mạch nhân với CLA.

3.1.3. Chạy TCL Script để tổng hợp mạch nhân

```
ngxxfus@ngxxfus-X409FA [15:12:56] [Eight-bits-Exact-Signed-Multiplier-with-CLA]
$ yosys -c ./synth.tcl
...
| yosys -- Yosys Open Synthesis Suite
| Copyright (C) 2012 - 2025 Claire Xenia Wolf <claire@yosyshq.com>
| Distributed under an ISC-like license, type "license" to see terms
\

Yosys 0.52+45 (git sha1 19845be85, g++ 13.3.0-gubuntu2-24.04 -fPIC -O3)

1. Executing Verilog-2005 frontend: /home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/AND_GATE.v
Parsing Verilog input from '/home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/AND_GATE.v' to AST representation.
Generating RTLIL representation for module `AND_GATE'.
Successfully finished Verilog frontend.

2. Executing Verilog-2005 frontend: /home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookAhead4bitsSumComponent.v
Parsing Verilog input from '/home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookAhead4bitsSumComponent.v' to AST representation.
Generating RTLIL representation for module `CarryLookAhead4bitsSumComponent'.
Successfully finished Verilog frontend.

3. Executing Verilog-2005 frontend: /home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookaheadAdder12bits.v
Parsing Verilog input from '/home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookaheadAdder12bits.v' to AST representation.
Generating RTLIL representation for module `CarryLookaheadAdder12bits'.
Successfully finished Verilog frontend.

4. Executing Verilog-2005 frontend: /home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookaheadAdder4bits.v
Parsing Verilog input from '/home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookaheadAdder4bits.v' to AST representation.
Generating RTLIL representation for module `CarryLookaheadAdder4bits'.
Successfully finished Verilog frontend.

5. Executing Verilog-2005 frontend: /home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookaheadLogicCircuit.v
Parsing Verilog input from '/home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookaheadLogicCircuit.v' to AST representation.
Generating RTLIL representation for module `CarryLookaheadLogicCircuit'.
Successfully finished Verilog frontend.

6. Executing Verilog-2005 frontend: /home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookaheadPandCreate.v
Parsing Verilog input from '/home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/CarryLookaheadPandCreate.v' to AST representation.
Generating RTLIL representation for module `CarryLookaheadPandCreate'.
Successfully finished Verilog frontend.

7. Executing Verilog-2005 frontend: /home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/FirstRow.v
Parsing Verilog input from '/home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/FirstRow.v' to AST representation.
Generating RTLIL representation for module `FirstRow'.
Successfully finished Verilog frontend.

8. Executing Verilog-2005 frontend: /home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/FullAdder1bit.v
Parsing Verilog input from '/home/ngxxfus/Downloads/Eight-bits-Exact-Signed-Multiplier-with-CLA/FullAdder1bit.v' to AST representation.
Generating RTLIL representation for module `FullAdder1bit'.
Successfully finished Verilog frontend.

9. Executing Verilog-2005 frontend: /home/ngxxfus/Downloads/Fight-bits-Exact-Signed-Multiplier-with-CLA/GroupA_8x4.v
```

a. Quá trình đọc các file Verilog có trong danh sách,

```
Used module: \GroupA_8x4
Removing unused module '\TestMultiplier'.
Removing unused module '\TestCLA4bits'.
Removing unused module '\TestCLA2bits'.
Removing unused module '\NAND_GATE'.
Removing unused module '\AND_GATE'.
Removed 5 unused modules.

22. Executing SYNTH pass.

22.1. Executing HIERARCHY pass (managing design hierarchy).

22.1.1. Analyzing design hierarchy..
Top module: \TheMultiplier
Used module: \CarryLookaheadAdder12bits
Used module: \CarryLookaheadAdder4bits
Used module: \CarryLookahead4bitsSumComponent
Used module: \CarryLookaheadLogicCircuit
Used module: \CarryLookaheadPandCreate
Used module: \GroupB_8x4
Used module: \LastRow0forGroupB
Used module: \Halfadder1bit
Used module: \PartialProductUnit
Used module: \FullAdder1bit
Used module: \PartialProductUnit_N
Used module: \Row0ofPartialProductUnit
Used module: \FirstRow
Used module: \GroupA_8x4

22.1.2. Analyzing design hierarchy..
Top module: \TheMultiplier
Used module: \CarryLookaheadAdder12bits
Used module: \CarryLookaheadAdder4bits
Used module: \CarryLookahead4bitsSumComponent
Used module: \CarryLookaheadLogicCircuit
Used module: \CarryLookaheadPandCreate
Used module: \GroupB_8x4
Used module: \LastRow0forGroupB
Used module: \Halfadder1bit
Used module: \PartialProductUnit
Used module: \FullAdder1bit
Used module: \PartialProductUnit_N
Used module: \Row0ofPartialProductUnit
Used module: \FirstRow
Used module: \GroupA_8x4
Removed 0 unused modules.

22.2. Executing PROC pass (convert processes to netlists).
22.2.1. Executing PROC CLEAN pass (remove empty switches from decision trees).
(Cleaned up 0 empty switches)
```

b. Kiểm tra cấu trúc phân cấp của thiết kế

```
27.1.2. Re-integrating ABC results.
ABC RESULTS: AND2_X1 cells: 4
ABC RESULTS: AND3_X1 cells: 14
ABC RESULTS: AO121_X1 cells: 1
ABC RESULTS: AO122_X1 cells: 37
ABC RESULTS: AO1221_X1 cells: 1
ABC RESULTS: AO122_X1 cells: 5
ABC RESULTS: INV_X1 cells: 8
ABC RESULTS: MUX2_X1 cells: 1
ABC RESULTS: NAND2_X1 cells: 58
ABC RESULTS: NAND3_X1 cells: 9
ABC RESULTS: NOR2_X1 cells: 36
ABC RESULTS: NOR3_X1 cells: 2
ABC RESULTS: OA121_X1 cells: 19
ABC RESULTS: OA122_X1 cells: 2
ABC RESULTS: OR2_X1 cells: 7
ABC RESULTS: OR3_X1 cells: 5
ABC RESULTS: XNOR2_X1 cells: 41
ABC RESULTS: XOR2_X1 cells: 41
ABC RESULTS: internal signals: 359
ABC RESULTS: input signals: 16
ABC RESULTS: output signals: 16
Removing temp directory.
```

c. Liệt kê các cell tiêu chuẩn được sử dụng trong thiết kế

```
28.2. Continuing TECHMAP pass.
No more expansions possible.
<suppressed >-75 debug messages>
29. Executing CHECK pass (checking for obvious problems).
Checking module TheMultiplier...
Found and reported 0 problems.
30. Executing Verilog backend.

30.1. Executing BMUXMAP pass.

30.2. Executing DEMUXMAP pass.
Dumping module `TheMultiplier'.

End of script. Logfile hash: 97b4ae2365, CPU: user 0.42s system 0.03s, MEM: 39.50 MB peak
Yosys 0.52+45 (git sha1 19845be85, g++ 13.3.0-gubuntu2-24.04 -fPIC -O3)
Time spent: 38% 2x abc (0 sec), 14% 16x opt_expr (0 sec), ...
```

d. Giai đoạn cuối của quá trình tổng hợp, xuất file và in thông báo về hiệu suất.

Hình . (a, b, c, d) Quá trình tổng hợp mạch nhân với CLA.

Kiểm tra các file đã tổng hợp:

```
ngxxfus@ngxxfus-X409FA [15:21:34] [Eight-bits-Exact-Signed-Multiplier-with-CLA]
$ find -name "*.v"
./mapped mul.v
./synth mul.v
ngxxfus@ngxxfus-X409FA [15:21:42] [Eight-bits-Exact-Signed-Multiplier-with-CLA]
```

Hình . Liệt kê các file đã tổng hợp từ Yosys.

3.2. Tổng hợp bộ nhân SCDM

Quá trình tổng hợp bộ nhân SCDM tương tự như quá trình tổng hợp bộ nhân sử dụng CLA.

3.2.1. Chuẩn bị

Trước khi sử dụng Yosys để tổng hợp, cần chuẩn bị các thông tin sau:

- + Tên top-module: Điều này rất quan trọng, vì nếu không có sẽ không tổng hợp được.
- + Danh sách các file Verilog có trong thiết kế. Yosys yêu cầu nhập tất cả các file .v liên quan.
- + Thư viện cell tiêu chuẩn, yosys có thể không cần thư viện cell tiêu chuẩn để tổng hợp, nhưng sẽ cần cell tiêu chuẩn để ánh xạ sau này, thuận tiện cho việc mô phỏng.

Có thể lấy cá danh sách file Verilog thông qua lệnh find của CentOS, hình bên dưới mô tả cách sử dụng lệnh find để liệt kê tất cả các file Verilog có trong thư mục hiện tại (dấu * trong chuỗi so khớp cho biết rằng bất cứ tên nào kết thúc bằng .v đều khớp):

```
find -name "*.v"
```

```
ngxxfus@ngxxfus-X409FA [15:30:42] [SCDM_73]
$ ped
/home/ngxxfus/Downloads/SCDM_73
ngxxfus@ngxxfus-X409FA [15:30:43] [SCDM_73]
$ find -name "*.v"
./CarryLookahead4bitSSCMComponent.v
./CarryLookaheadAdder12bits.v
./CarryLookaheadAdder4bits.v
./CarryLookaheadLogicCircuit.v
./CarryLookaheadPadAndGCreate.v
./FirstRow.v
./HalfAdder1bit.v
./GA7.v
./GB5.v
./HalfAdder1bit.v
./PartialProductUnit0_N.v
./LastRowForGroupA.v
./LastRowForGroupB.v
./mapped SCDM_73.v
./netgen/synthesis/SCDM8_73_synthesis.v
./PartialProductUnit0.v
./PartialProductUnit1.v
./PartialProductUnit2.v
./PartialProductUnit3.v
./PartialProductUnit4.v
./PartialProductUnit5.v
./PartialProductUnit6.v
./PartialProductUnit7.v
./PartialProductUnit8.v
./RowAndRow20OfGroupA.v
./RowAndRow20OfGroupB.v
./SCDM8_73.v
./synth SCDM8_73.v
./TestsSCDM8_73.v
./TestsSCDM8_75.v
ngxxfus@ngxxfus-X409FA [15:30:46] [SCDM_73]
```

Hình . Sử dụng lệnh find để liệt kê tất cả các file Verilog.

```
ngxxfus@ngxxfus-X409FA [15:31:00] [SCDM_73]
$ pwd
/home/ngxxfus/Downloads/SCDM_73
ngxxfus@ngxxfus-X409FA [15:31:05] [SCDM_73]
$ find -name "*.v" | sed -e 's/\.\///'
CarryLookahead4bitSSCMComponent.v
CarryLookaheadAdder12bits.v
CarryLookaheadAdder4bits.v
CarryLookaheadLogicCircuit.v
CarryLookaheadPadAndGCreate.v
FirstRow.v
FullAdder1bit.v
GA7.v
GB5.v
HalfAdder1bit.v
PartialProductUnit0_N.v
LastRowForGroupA.v
LastRowForGroupB.v
mapped SCDM_75.v
netgen/synthesis/SCDM8_73_synthesis.v
PartialProductUnit0.v
PartialProductUnit1.v
PartialProductUnit11_N.v
PartialProductUnit2.v
PartialProductUnit2_N.v
PartialProductUnit3.v
RowAndRow20OfGroupA.v
RowAndRow20OfGroupB.v
SCDM8_75.v
synth SCDM8_75.v
TestsSCDM8_73.v
TestsSCDM8_75.v
ngxxfus@ngxxfus-X409FA [15:31:09] [SCDM_73]
```

Hình . Loại bỏ ./ bằng lệnh sed.

Sau cùng, cần chọn lọc một lần nữa danh sách các file Verilog thực sự nằm trong thiết kế và thêm vào mẫu TCL ở phần 1.3.

3.1.2. Xây dựng TCL script để tổng hợp mạch nhân SCDM

```
# IMPORTANT: Every directory in this script must be absolute paths and end with a '/'
# E.g:
#   '/home/user/dir/' ..... '/home/user/dir/synth_output.v'
#   '/home/user/dir' ..... '/home/user/dir/synth_output.v'

set working_dir "/home/ngxxfus/Downloads/SCDM_73"
set output_dir "/home/ngxxfus/Downloads/SCDM_73"
set lib_path "/home/ngxxfus/Downloads/NangateOpenCellLibrary_PDKv1_3_v2010_12/Front_End/Liberty/NLDM/NangateOpenCellLibrary_typical.lib"
set synth_output "synth mul_2.v"
set mapped_output "mapped mul_2.v"
set top_module "SCDM_73"
set verilog_files {
    CarryLookaheadAdder1bitsSumComponent.v
    CarryLookaheadAdder2bits.v
    CarryLookaheadAdder3bits.v
    CarryLookaheadLogicCircuit.v
    CarryLookaheadProductCreate.v
    FirstRow.v
    FullAdder1bit.v
    GA7.v
    GBS.v
    HalfAdder1bit.v
    PartialProductUnit0.v
    LastRowOfGroupA.v
    LastRowOfGroupB.v
    PartialProductUnit0.v
    PartialProductUnit1.v
    PartialProductUnit1_N.v
    PartialProductUnit2.v
    PartialProductUnit2_N.v
    PartialProductUnit3.v
    Row1AndRow20ForGroupA.v
    Row1AndRow20ForGroupB.v
    SCDM_73.v
}
foreach file $verilog_files {
    yosys read_verilog $working_dir$file
}
yosys hierarchy -check -top $top_module
yosys synthesize -top module flatten
yosys write_verilog -noattr $output_dir$synth_output
yosys read liberty -lib $lib_path
yosys opt
yosys dfflibmap -liberty $lib_path
yosys abc -liberty $lib_path
yosys techmap +techmap.v
yosys check
yosys write_verilog -noattr $output_dir$mapped_output
```

Hình . TCL Script để tổng hợp mạch nhân SCDM.

3.1.3. Chạy TCL Script để tổng hợp mạch nhân SCDM

```
24.25.1. Analyzing design hierarchy..
Top module: `SCDM8_73

24.25.2. Analyzing design hierarchy..
Top module: `SCDM8_73
Removed 0 unused modules.

24.26. Printing statistics.

== SCDM8_73 ===

Number of wires:          470
Number of wire bits:      740
Number of public wires:   205
Number of public wire bits: 475
Number of ports:          3
Number of port bits:      32
Number of memories:       0
Number of memory bits:    0
Number of processes:      0
Number of cells:          37
$ ANDNOT_                71
$ AND_                     59
$ NAND_                   16
$ NOR_                     15
$ NOT_                     2
$ ORNOT_                  4
$ OR_                      27
$ XNOR_                   20
$ XOR_                     67
$scopeinfo                 96
```

a. In thông tin về thiết kế (chưa ánh xạ)

```
29.1.2. Re-integrating ABC results.
ABC RESULTS: AND2_X1 cells:     8
ABC RESULTS: AND3_X1 cells:     7
ABC RESULTS: AND4_X1 cells:     5
ABC RESULTS: AOI211_X1 cells:   1
ABC RESULTS: AOI211_X1 cells:   26
ABC RESULTS: AOI221_X1 cells:   1
ABC RESULTS: AOI222_X1 cells:   5
ABC RESULTS: IOR1_X1 cells:    4
ABC RESULTS: NAND2_X1 cells:   56
ABC RESULTS: NAND3_X1 cells:   5
ABC RESULTS: NAND4_X1 cells:   2
ABC RESULTS: NOR2_X1 cells:    17
ABC RESULTS: NOR3_X1 cells:    3
ABC RESULTS: NOR4_X1 cells:    1
ABC RESULTS: OAII11_X1 cells:   1
ABC RESULTS: OAII21_X1 cells:   16
ABC RESULTS: OAII22_X1 cells:   3
ABC RESULTS: OR2_X1 cells:     3
ABC RESULTS: OR3_X1 cells:     4
ABC RESULTS: XOR2_X1 cells:    35
ABC RESULTS: XOR2_X1 cells:    38
ABC RESULTS: internal signals: 285
ABC RESULTS: input signals:    16
ABC RESULTS: output signals:   16

Removing temp directory.
```

b. In thông tin về thiết kế (đã ánh xạ)

```

30. Executing TECHMAP pass (map to technology primitives).
30.1. Executing Verilog-2000 frontend: /usr/local/bin/.../share/yosys/techmap.v
Parsing Verilog input from '/usr/local/bin/.../share/yosys/techmap.v' to AST representation.
Generating RTILL representation for module '\_90_simplemap_bool_ops'.
Generating RTILL representation for module '\_90_simplemap_reduce_ops'.
Generating RTILL representation for module '\_90_simplemap_logic_ops'.
Generating RTILL representation for module '\_90_simplemap_compare_ops'.
Generating RTILL representation for module '\_90_simplemap_various'.
Generating RTILL representation for module '\_90_simplemap_registers'.
Generating RTILL representation for module '\_90_shift_ops_shl_shrl_shsr'.
Generating RTILL representation for module '\_90_shift_shiftx'.
Generating RTILL representation for module '\_90_fa'.
Generating RTILL representation for module '\_90_lcm_gcd_kung'.
Generating RTILL representation for module '\_90_macc'.
Generating RTILL representation for module '\_90_alignacc'.
Generating RTILL representation for module '\_90_div_mod_u'.
Generating RTILL representation for module '\_90_div_mod_trunc'.
Generating RTILL representation for module '\_90_div'.
Generating RTILL representation for module '\_90_mod'.
Generating RTILL representation for module '\_90_mod_floor'.
Generating RTILL representation for module '\_90_modceil'.
Generating RTILL representation for module '\_90_pow'.
Generating RTILL representation for module '\_90_pnrm'.
Generating RTILL representation for module '\_90_pnrmu'.
Generating RTILL representation for module '\_90_lut'.
Successfully finished Verilog frontend.

30.2. Continuing TECHMAP pass.
No more expansions possible.
<suppressed -75 debug messages>

31. Executing CHECK pass (checking for obvious problems).
Checking module SCDM_73...
Found and reported 0 problems.

32. Executing Verilog backend.

32.1. Executing BMUXMAP pass.

32.2. Executing DEMUXMAP pass.
Dumping module '\$SCDM_73'.

End of script. Logfile hash: 46bbbe6a0a, CPU: user 0.38s system 0.01s, MEM: 39.54 MB peak
Yosys 0.52+45 (git sha1 19845be05, g++ 13.3.0-6ubuntu2-24.04 -fPIC -O3)
Time spent: 40% 2x abc (0 sec), 14% 16x opt_expr (0 sec), ...

```

c. Giai đoạn cuối của quá trình tổng hợp, xuất file và in thông báo về hiệu suất.

Hình . (a, b, c) Quá trình tổng hợp mạch nhân SCDM.

Kiểm tra các file đã tổng hợp:

```

Ngxxfus@Ngxxfus-X409FA [15:30:30] [SCDM_73]
$ find -name "\*mul\_*"
./mapped mul_2.v
./synth mul_2.v
Ngxxfus@Ngxxfus-X409FA [15:40:59] [SCDM_73]
$ 

```

Hình . Liệt kê các file đã tổng hợp từ Yosys.

4. So sánh công suất, độ trễ hai bộ nhân 8-bit

4.1. Điều kiện mô phỏng

4.1.1. Thông số môi trường

- Nhiệt độ: 27°C
- Model: FreePDK45/ncsu_basekit/hspice_nom
- Thư viện cell tiêu chuẩn: NangateOpenCellLibrary

4.1.2. Thông số nguồn DC

- $V_{DD} = V_{DD} = 1V$

4.1.3. Thông số nguồn AC (nguồn xung)

Các thông số cơ sở:

- Tần số cơ sở: $F_{base} = F = 50MHz$
- Thời gian cạnh lên / xuống: $t_{falling} = t_{raising} = 0.0025/F$
(Giống nhau ở các nguồn cấp xung)

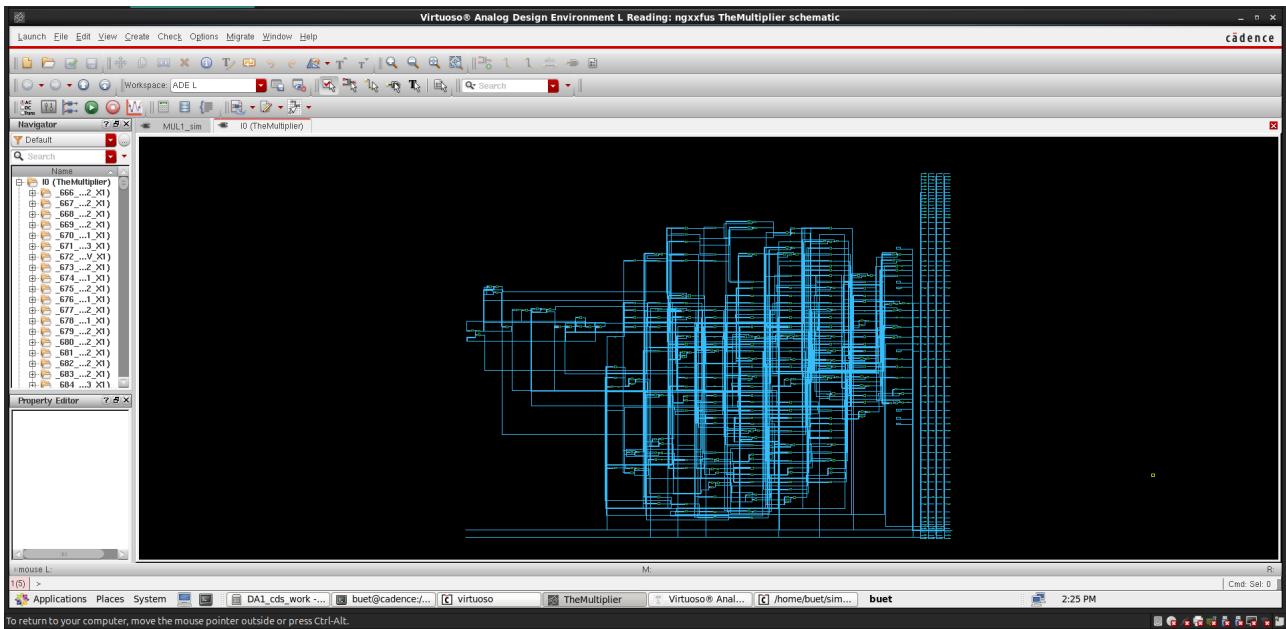
Bảng 4.1.3.1. Giá trị chu kỳ ứng với các bit của hai đầu vào **a** và **b**:

period	LSB 1-bit	2-bit	3-bit	4-bit	5-bit	6-bit	7-bit	MSB 8-bit
a	1/F	2/F	3/F	4/F	5/F	6/F	7/F	8/F
b	9/F	10/F	11/F	12/F	13/F	14/F	15/F	16/F

4.2. Mô phỏng mạch nhân sử dụng CLA

4.2.1. Sơ đồ nguyên lý mạch nhân sử dụng CLA

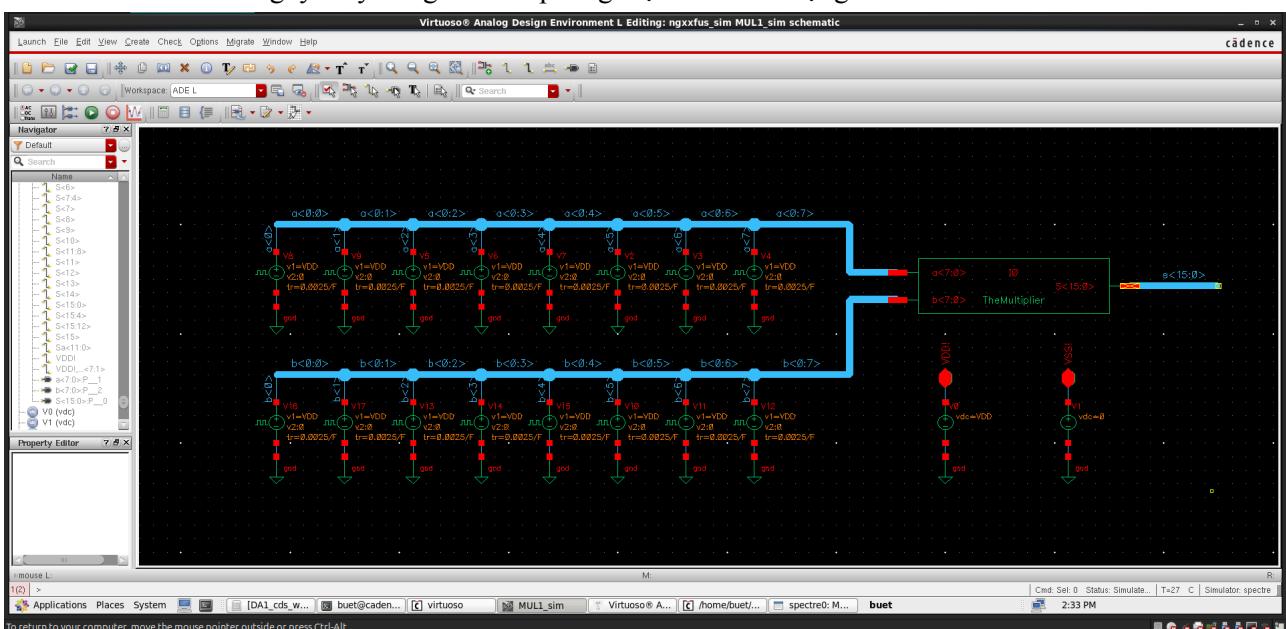
Sau khi tổng hợp bằng Yosys, thực hiện import tập tin Verilog chứa bộ nhân CLA đã tổng hợp vào Cadence Virtuoso. Hình bên dưới mô tả sơ đồ nguyên lý mạch nhân sau khi được import vào Cadence Virtuoso.



Hình . Sơ đồ nguyên lý mạch nhân sử dụng CLA.

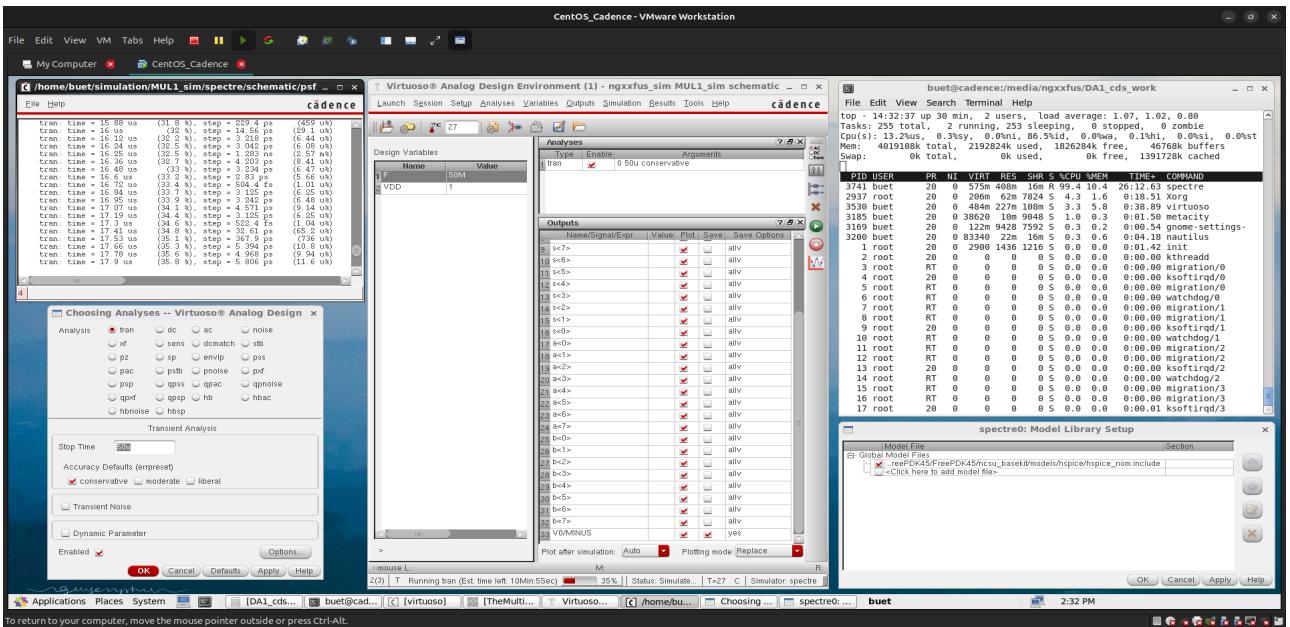
4.2.2. Sơ đồ nguyên lý dùng cho mô phỏng mạch nhân sử dụng CLA

Nhập mạch nhân và kết nối với nguồn xung, nguồn DC (VDD!, VSS!) để thực hiện mô phỏng. Hình bên dưới mô tả sơ đồ nguyên lý dùng cho mô phỏng mạch nhân sử dụng CLA.



Hình . Sơ đồ nguyên lý cho mô phỏng mạch nhân với CLA.

Hình bên dưới mô tả thiết lập mô trường mô phỏng ADE với khoảng thời gian mô phỏng là 50us trên Cadence Virtuoso.



Hình . Các thiết lập mô phỏng mô trường ADE của Candence Virtuoso.

4.2.3. Quá trình mô phỏng

Quá trình mô phỏng bị tạm dừng ở micro giây thứ 40 vì không đủ dung lượng mô phỏng. Hình bên dưới mô tả quá trình mô phỏng của trình mô phỏng mạch điện analog - spectre. Bao gồm các cảnh báo và thông số hệ thống.

```
Time for EDB Visiting: CPU = 17.998 ms, elapsed = 18.6069 ms.
Time accumulated: CPU = 397.939 ms, elapsed = 461.438 ms.
Peak resident memory used = 38.9 Mbytes.

Notice from spectre during topology check.
Only one connection to the following 284 nodes:
  S1<3>
  S1<2>
  S1<1>
  IO.Sa<11>
  IO.Sa<10>
  Further occurrences of this notice will be suppressed.
No DC path from node `IO.CLA\CLAO\P<1>' to ground, Gmin installed to provide path.
No DC path from node `IO.CLA\CLAO\P<2>' to ground, Gmin installed to provide path.
No DC path from node `IO.CLA\CLAO\P<3>' to ground, Gmin installed to provide path.
No DC path from node `IO.Sa<4>' to ground, Gmin installed to provide path.
No DC path from node `IO.Sa<5>' to ground, Gmin installed to provide path.
  Further occurrences of this notice will be suppressed.

Circuit inventory:
  nodes 1302
  bsim4 1918
  iprobe 306
  vsource 18

Notice from spectre.
  14 notices suppressed.

Time for parsing: CPU = 223.965 ms, elapsed = 228.725 ms.
Time accumulated: CPU = 622.904 ms, elapsed = 690.402 ms.
Peak resident memory used = 44.8 Mbytes.

Entering remote command mode using MPSC service (spectre, ipi, v0.0, spectre0_3575_1, ).
```

a.

```

Warning from spectre.
WARNING (SPECTRE-16707): Only tran supports psfkl format, result of other analyses will be in psfbin format.

*****
Transient Analysis `tran': time = (0 s -> 50 us)
*****
DC simulation time: CPU = 860.869 ms, elapsed = 863.116 ms.
Important parameter values:
  start = 0 s
  outputstart = 0 s
  stop = 50 us
  step = 50 ns
  maxstep = 500 ns
  ic = all
  useprevic = no
  skipdcs = no
  rltol = 100e-06
  abstol(V) = 1 uV
  abstol(I) = 1 pA
  temp = 27 C
  tnom = 27 C
  tempeffects = all
  errpreset = conservative
  method = gear2only
  iteratio = 10
  relref = allocal
  cmin = 0 F
  gmin = 1 pS

  tran: time = 133.3 ps   (267 u%), step = 101.8 fs   (204 n%)
  tran: time = 40.06 ns   (80.1 m%), step = 2.855 ps   (5.71 u%)
  tran: time = 90.11 ns   (180 m%), step = 325.4 fs   (651 n%)
  tran: time = 100.2 ns   (200 m%), step = 528.6 fs   (1.06 u%)
  tran: time = 110.2 ns   (220 m%), step = 743.5 fs   (1.49 u%)

```

b.

```

  tran: time = 40.2 us   (80.4 %), step = 872.6 fs   (1.75 u%)
  tran: time = 40.22 us   (80.4 %), step = 638.6 fs   (1.28 u%)
  tran: time = 40.24 us   (80.5 %), step = 1.15 ps   (2.3 u%)
  tran: time = 40.25 us   (80.5 %), step = 387.9 fs   (776 n%)
  tran: time = 40.26 us   (80.5 %), step = 410.7 fs   (821 n%)
  tran: time = 40.27 us   (80.5 %), step = 912.2 fs   (1.82 u%)
  tran: time = 40.28 us   (80.6 %), step = 378.5 ps   (757 u%)
  tran: time = 40.3 us   (80.6 %), step = 504.3 fs   (1.01 u%)
  tran: time = 40.31 us   (80.6 %), step = 514.5 fs   (1.03 u%)
  tran: time = 40.32 us   (80.6 %), step = 405.1 fs   (810 n%)
  tran: time = 40.35 us   (80.7 %), step = 604.2 fs   (1.21 u%)
  tran: time = 40.36 us   (80.7 %), step = 334.1 fs   (668 n%)
  tran: time = 40.37 us   (80.7 %), step = 952 fs   (1.9 u%)
  tran: time = 40.38 us   (80.8 %), step = 467.2 fs   (934 n%)

```

Saving the states into file at t=40.3802 us...

Killed by user.

Aggregate audit (10:40:57 AM, Mon May 26, 2025):
Time used: CPU = 28.9 ks (8h 1m 60s), elapsed = 53.5 ks (14h 52m 10s), util. = 54%.
Time spent in licensing: elapsed = 261 ms.
Peak memory used = 44.8 Mbytes.
Simulation started at: 7:48:47 PM, Sun May 25, 2025, ended at: 10:40:57 AM, Mon May 26, 2025, with elapsed time (wall clock): 53.5 ks (14
spectre completes with 0 errors, 5 warnings, and 6 notices.

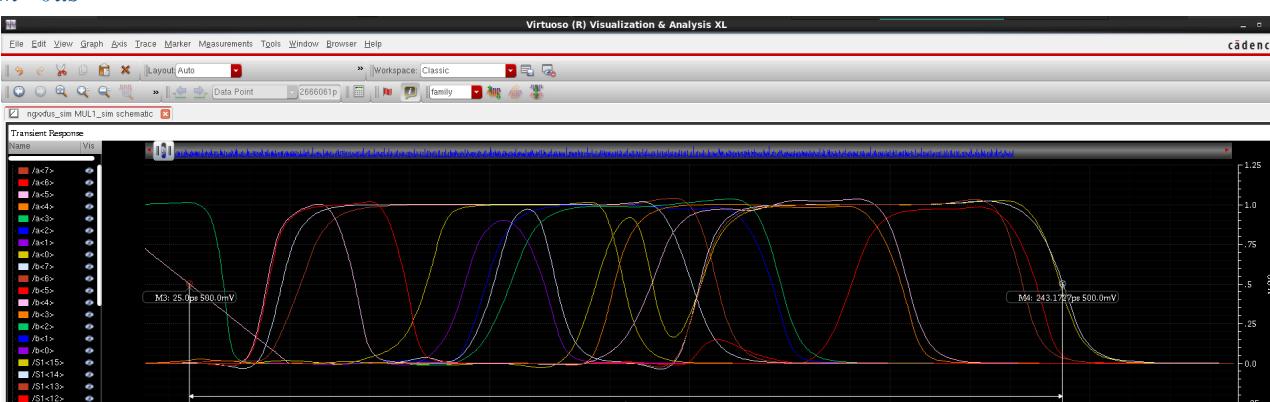
c.

Hình . (a, b, c) Thông tin trạng thái mô phỏng của spectre.

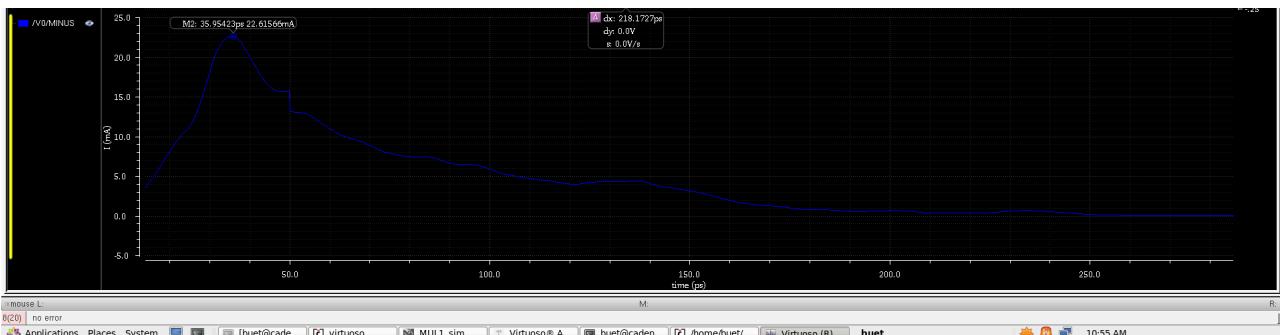
4.2.4. Kết quả mô phỏng mạch nhân với CLA

Tại thời điểm ban đầu ($t=0$), các ngõ vào đồng loạt có giá trị từ '1' chuyển sang '0'. Hình bên dưới mô tả dạng sóng ngõ ra của bộ nhân tại thời điểm đó. Công suất tiêu thụ là 22,6mW (cực đại trong suốt quá trình mô phỏng).

#t=0us



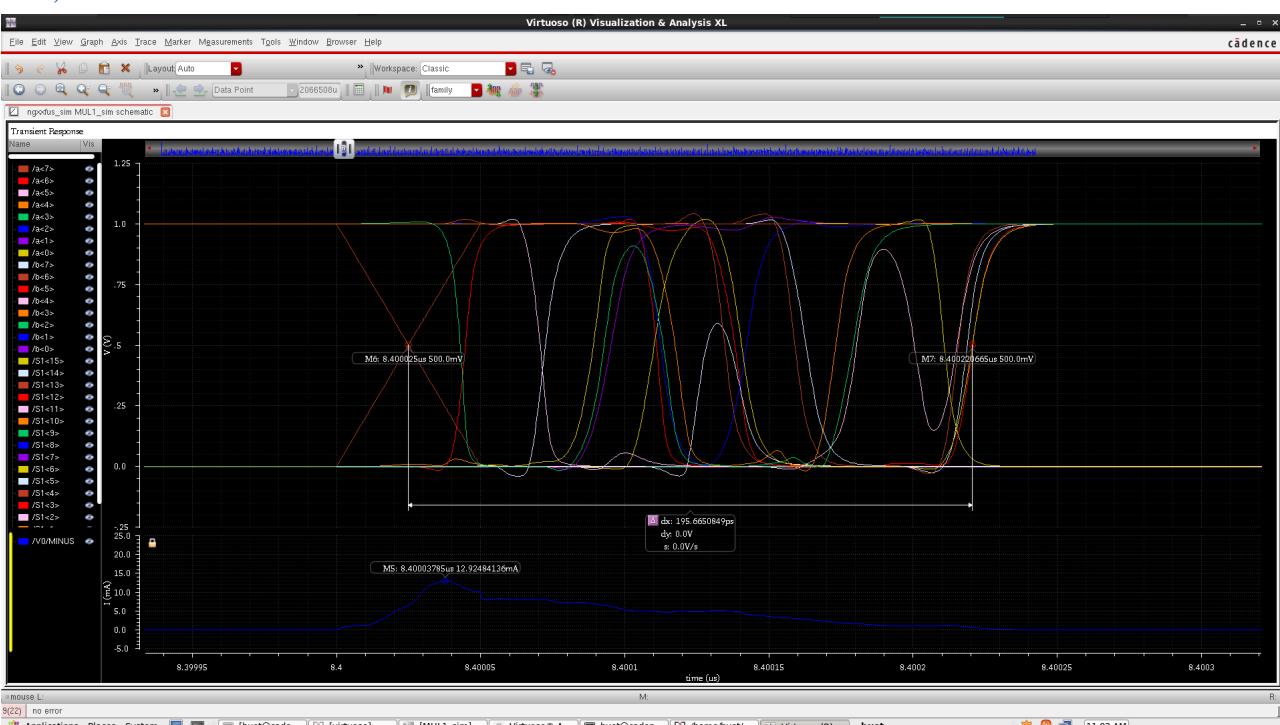
Nhận xét: Xuất hiện các khoảng vượt ngưỡng lớn hơn VDD, giai đoạn này kéo dài 218,2ps kể từ VDD/2 của các tín hiệu ngõ vào đến thời điểm VDD/2 của tín hiệu vượt ngưỡng cuối cùng.



Hình . Dạng sóng ngõ ra dòng điện tiêu tụ của bộ nhân tại thời điểm ban đầu.

Nhận xét: Công suất tiêu thụ lớn nhất trong quá trình mô phỏng, lên đến 22,6mW.

$t=8,400025\mu s$

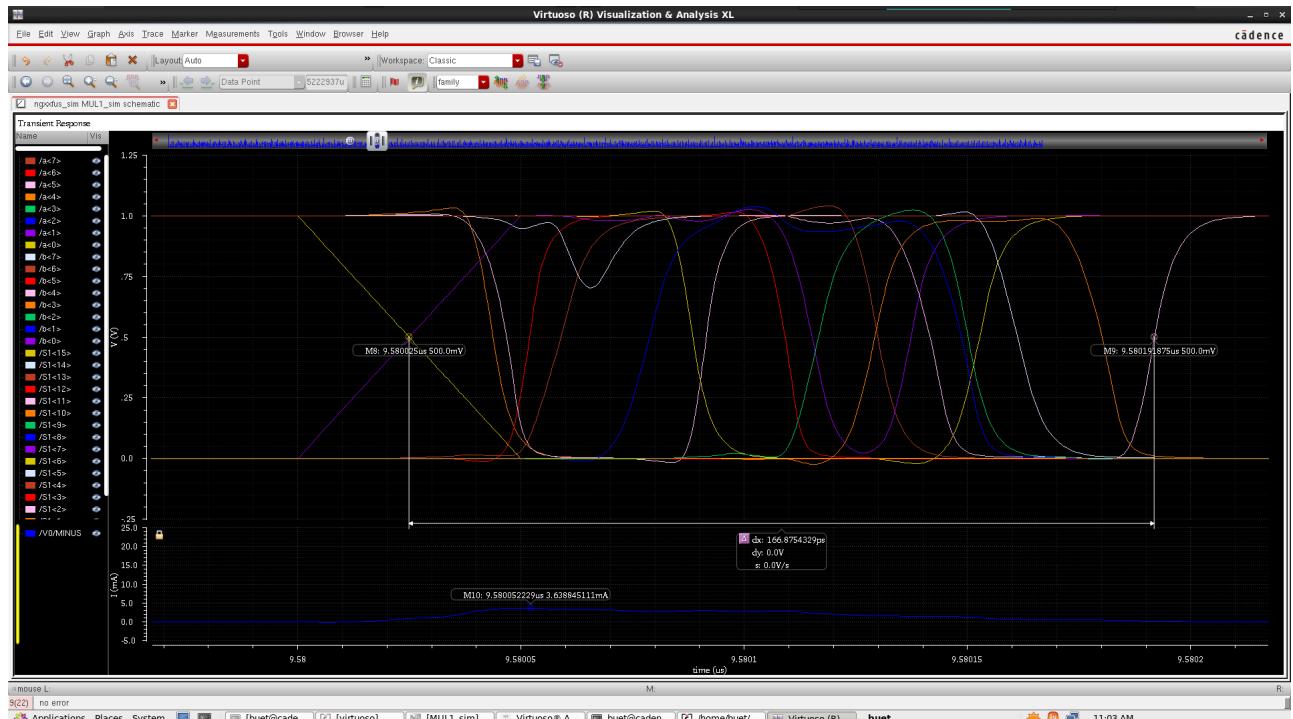


Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm $t=8,400025\mu s$.

Nhận xét:

- Độ trễ lan truyền: 195,7ps
- Công suất cực đại cục bộ: 12.9mW

#t=9,580025us

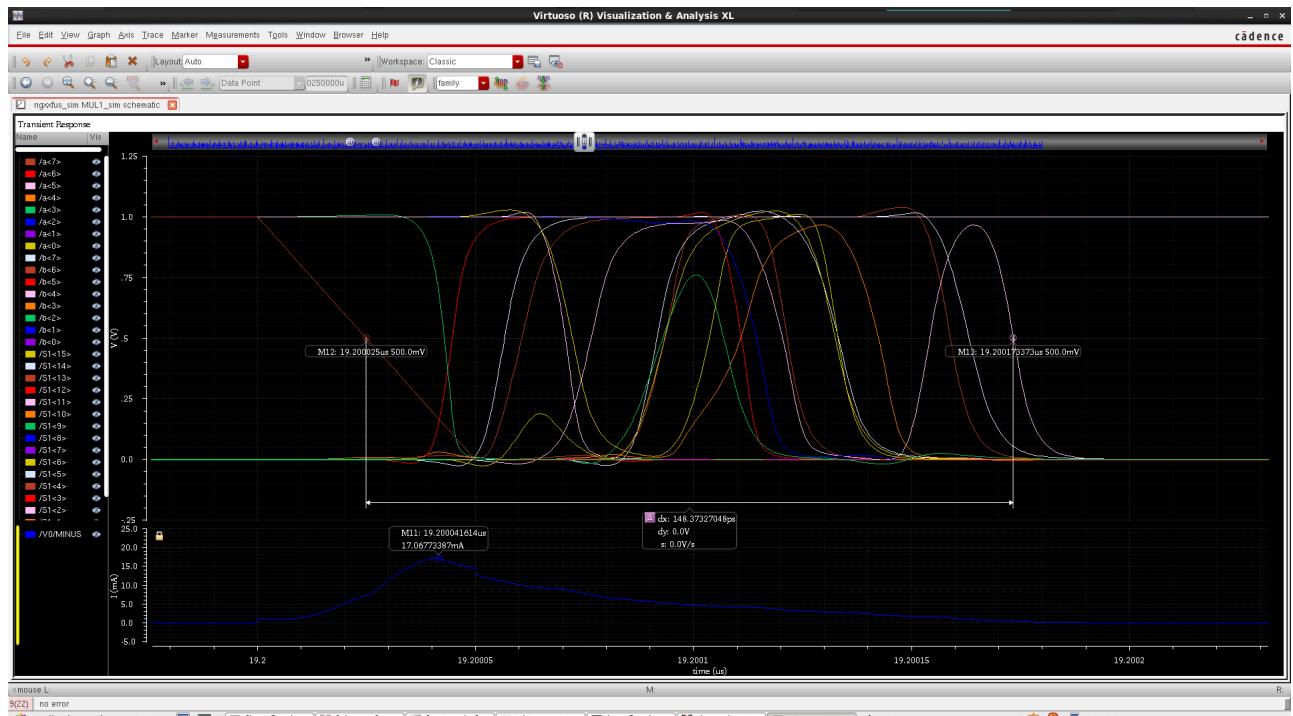


Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm t=9,580025us.

Nhận xét:

- Độ trễ lan truyền: 166.9ps
- Công suất cực đại cục bộ: 3.6mW

#t=19,200025us



Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm t=19,200025us.

Nhận xét:

- Độ trễ lan truyền: 148.4ps
- Công suất cực đại cục bộ: 17.1mW

$t=25,200025\mu s$

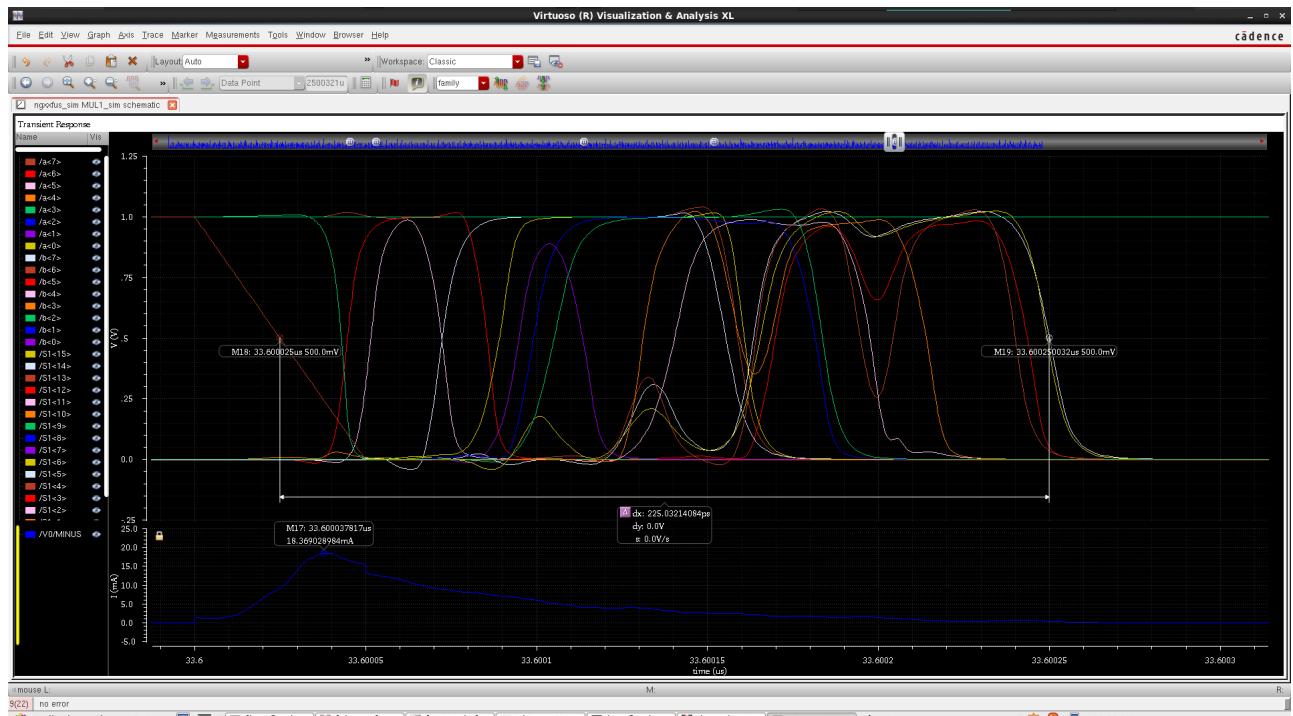


Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm $t=25,200025\mu s$.

Nhận xét:

- Độ trễ lan truyền: 192.0ps
- Công suất cực đại cục bộ: 19.7mW

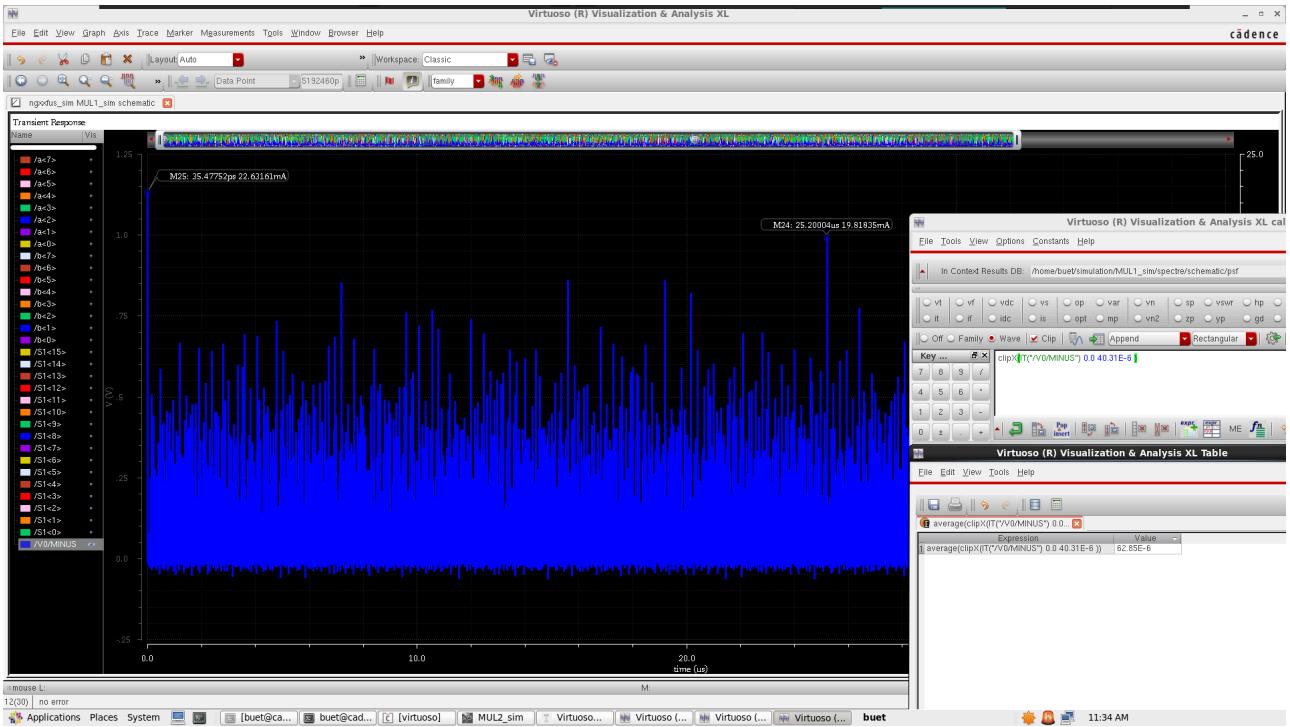
$t=33,600025\mu s$



Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm $t=33,600025\mu s$.

Nhận xét:

- Độ trễ lan truyền: 225.0ps
- Công suất cực đại cục bộ: 18.4mW



Hình . Dạng sóng dòng điện tiêu thụ trong suốt quá trình mô phỏng và công suất trung bình.

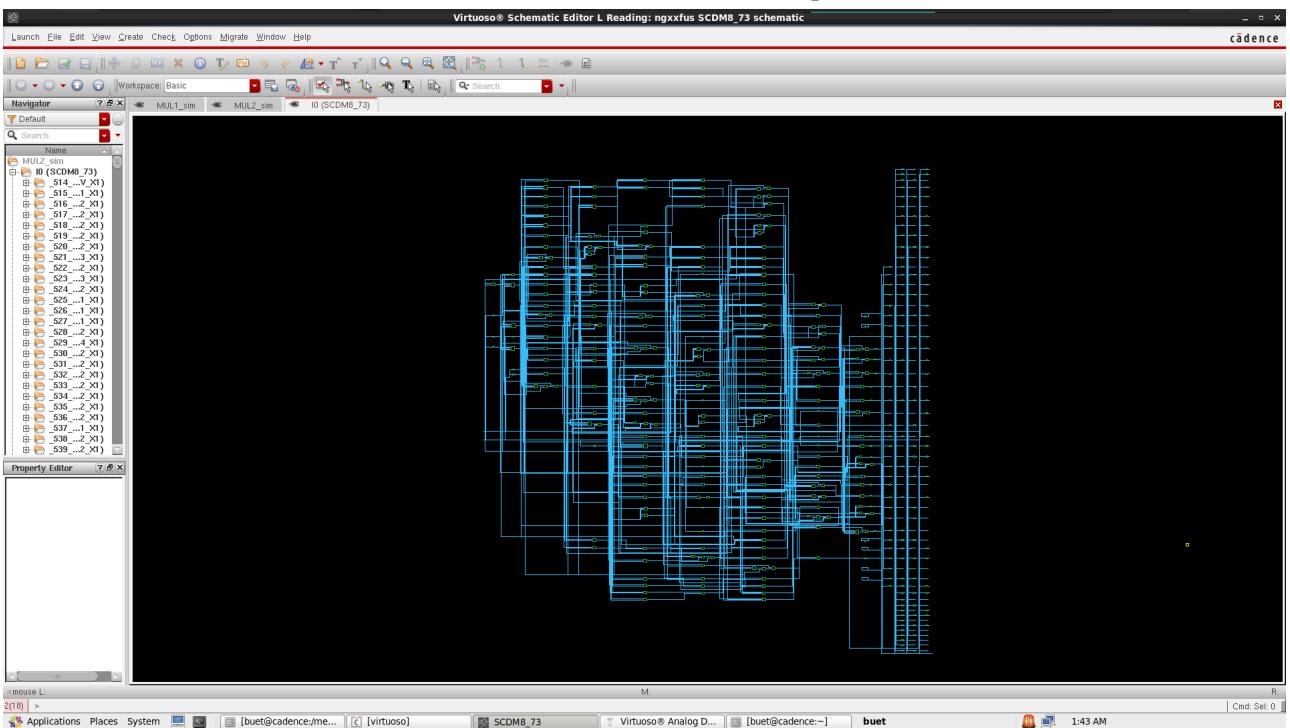
Nhận xét:

- + Công suất cực đại: 22.63mW.
- + Công suất trung bình: 62.85uW.

4.3. Mô phỏng mạch nhân SCDM

4.3.1. Sơ đồ nguyên lý mạch nhân SCDM8_73

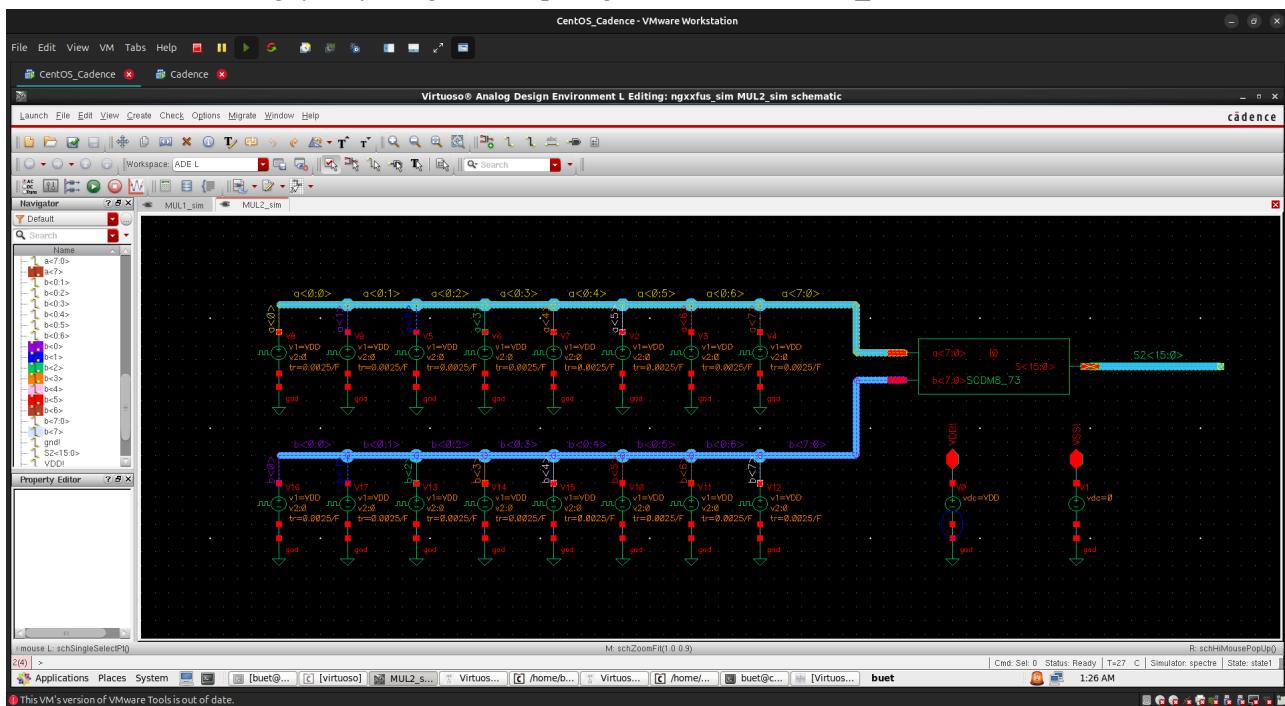
Sau khi tổng hợp bằng Yosys, thực hiện import tập tin Verilog chứa bộ nhân CLA đã tổng hợp vào Cadence Virtuoso. Hình bên dưới mô tả mạch nhân sau khi được import vào Cadence Virtuoso.



Hình . Sơ đồ nguyên lý mạch nhân SCDM8_73.

4.3.2. Sơ đồ nguyên lý dùng cho mô phỏng mạch nhân SCDM8_73

Nhập mạch nhân và kết nối với nguồn xung, nguồn DC (VDD!, VSS!) để thực hiện mô phỏng. Hình bên dưới mô tả sơ đồ nguyên lý dùng cho mô phỏng mạch nhân SCDM8_73.



Hình . Sơ đồ nguyên lý cho mô phỏng của mạch nhân SCDM8_73.

4.2.3. Quá trình mô phỏng

Quá trình mô phỏng bị tạm dừng ở micro giây thứ 40 vì không đủ dung lượng mô phỏng. Hình bên dưới mô tả quá trình mô phỏng của ADE. Bao gồm các cảnh báo và thông số hệ thống.

```
Time for EDB Visiting: CPU = 17.998 ms, elapsed = 18.8909 ms.
Time accumulated: CPU = 440.932 ms, elapsed = 425.053 ms.
Peak resident memory used = 38 Mbytes.

Notice from spectre during topology check.
Only one connection to the following 268 nodes:
S2<3>
S2<2>
S2<1>
IO_Sa<11>
IO_Sa<10>
Further occurrences of this notice will be suppressed.
No DC path from node 'IO_CLA\CLA0,P<1>' to ground, Gmin installed to provide path.
No DC path from node 'IO_CLA\CLA0,P<2>' to ground, Gmin installed to provide path.
No DC path from node 'IO_CLA\CLA0,P<3>' to ground, Gmin installed to provide path.
No DC path from node 'IO_Sa<4>' to ground, Gmin installed to provide path.
No DC path from node 'IO_Sa<5>' to ground, Gmin installed to provide path.
Further occurrences of this notice will be suppressed.

Circuit inventory:
nodes 1110
bsim4 1568
iprobe 290
vssource 18

Notice from spectre.
14 notices suppressed.

Time for parsing: CPU = 205.969 ms, elapsed = 217.671 ms.
Time accumulated: CPU = 647.901 ms, elapsed = 642.982 ms.
Peak resident memory used = 43 Mbytes.

Entering remote command mode using MPSC service (spectre, ipi, v0.0, spectre1_3575_2, ).
```

a.

```

Warning from spectre.
WARNING (SPECTRE-16707): Only tran supports psfxml format, result of other analyses will be in psfbin format.

*****
Transient Analysis 'tran': time = (0 s -> 50 us)
*****
DC simulation time: CPU = 622.905 ms, elapsed = 623.978 ms.
Important parameter values:
start = 0 s
outputstart = 0 s
stop = 50 us
step = 50 ns
maxstep = 500 ns
ic = all
useprevic = no
skipdc = no
reltol = 100e-06
abstol(V) = 1 uV
abstol(I) = 1 pA
temp = 27 C
trom = 27 C
tempeffects = all
errpreset = conservative
method = gear2only
literatio = 10
relref = alloclocal
cmin = 0 F
gmin = 1 pS

tran: time = 10.01 ns (20 m%), step = 2.412 ps (4.82 u%)
tran: time = 70.91 ns (142 m%), step = 158.7 ps (317 u%)
tran: time = 100 ns (200 m%), step = 461 fs (922 n%)
tran: time = 120 ns (240 m%), step = 509.6 fs (1.02 u%)

```

b.

```

tran: time = 41.18 us (82.4 %), step = 1.293 ps (2.59 u%)
tran: time = 41.2 us (82.4 %), step = 834.5 fs (1.67 u%)
tran: time = 41.22 us (82.4 %), step = 549.3 fs (1.1 u%)
tran: time = 41.24 us (82.5 %), step = 499.4 fs (999 n%)
tran: time = 41.25 us (82.5 %), step = 3.125 ps (6.25 u%)
tran: time = 41.27 us (82.5 %), step = 1.064 ns (2.13 m%)
tran: time = 41.29 us (82.6 %), step = 634.8 fs (1.27 u%)

Saving the states into file at t=41.3002 us...
Killed by user.

Aggregate audit (1:17:37 AM, Mon May 26, 2025):
Time used: CPU = 19.6 ks (5h 26m 46s), elapsed = 19.6 ks (5h 26m 49s), util. = 100%.
Time spent in licensing: elapsed = 333 ms.
Peak memory used = 43 Mbytes.
Simulation started at: 7:50:47 PM, Sun May 25, 2025, ended at: 1:17:37 AM, Mon May 26, 2025, with elapsed time (wall clock): 19.6 ks (5h 26m 49s).
spectre completes with 0 errors, 4 warnings, and 6 notices.

```

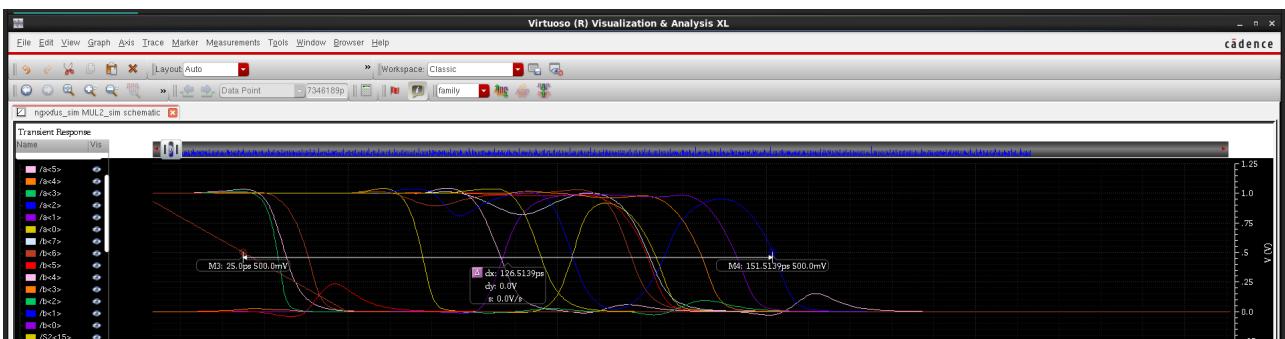
c.

Hình . (a, b, c) Thông tin trạng thái mô phỏng của spectre.

4.2.4. Kết quả mô phỏng mạch nhân SCDM

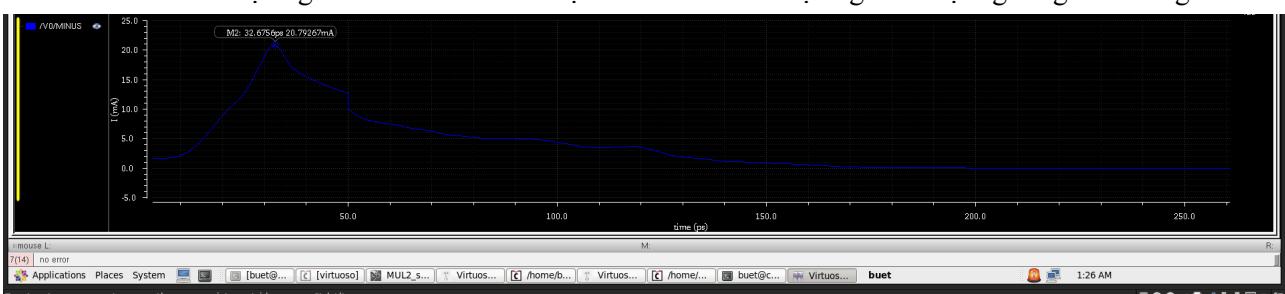
Tại thời điểm ban đầu ($t=0$), các ngõ vào đồng loạt có giá trị từ '1' chuyển sang '0'. Hình bên dưới mô tả dạng sóng ngõ ra của bộ nhân tại thời điểm đó.

$t=0$ us



Hình . Dạng sóng ngõ ra của bộ nhận tại thời điểm ban đầu.

Nhận xét: Xuất hiện các khoảng vượt ngưỡng lớn hơn VDD, giai đoạn này kéo dài 126,5ps kể từ thời điểm VDD/2 của các tín hiệu ngõ vào đến thời điểm đạt VDD/2 của tín hiệu ngõ ra vượt ngưỡng cuối cùng.



Hình . Dạng sóng ngõ ra dòng điện tiêu thụ của bộ nhân tại thời điểm ban đầu.
Nhận xét: Công suất tiêu thụ lớn nhất trong quá trình mô phỏng, lên đến 20,8mW.

#t=8,400025us

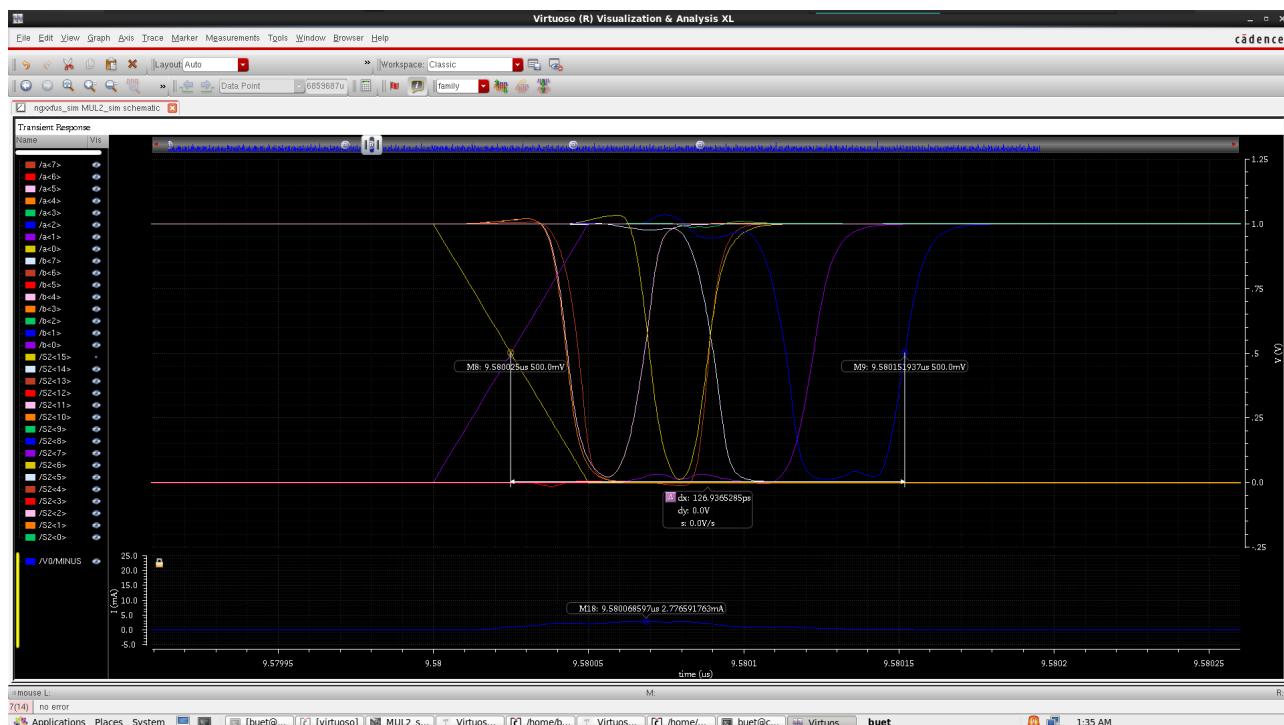


Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm $t=8,400025\mu s$.

Nhận xét:

- Độ trễ lan truyền: 114,3ps
- Công suất cực đại cục bộ: 11.6mW

#t=9,500025us



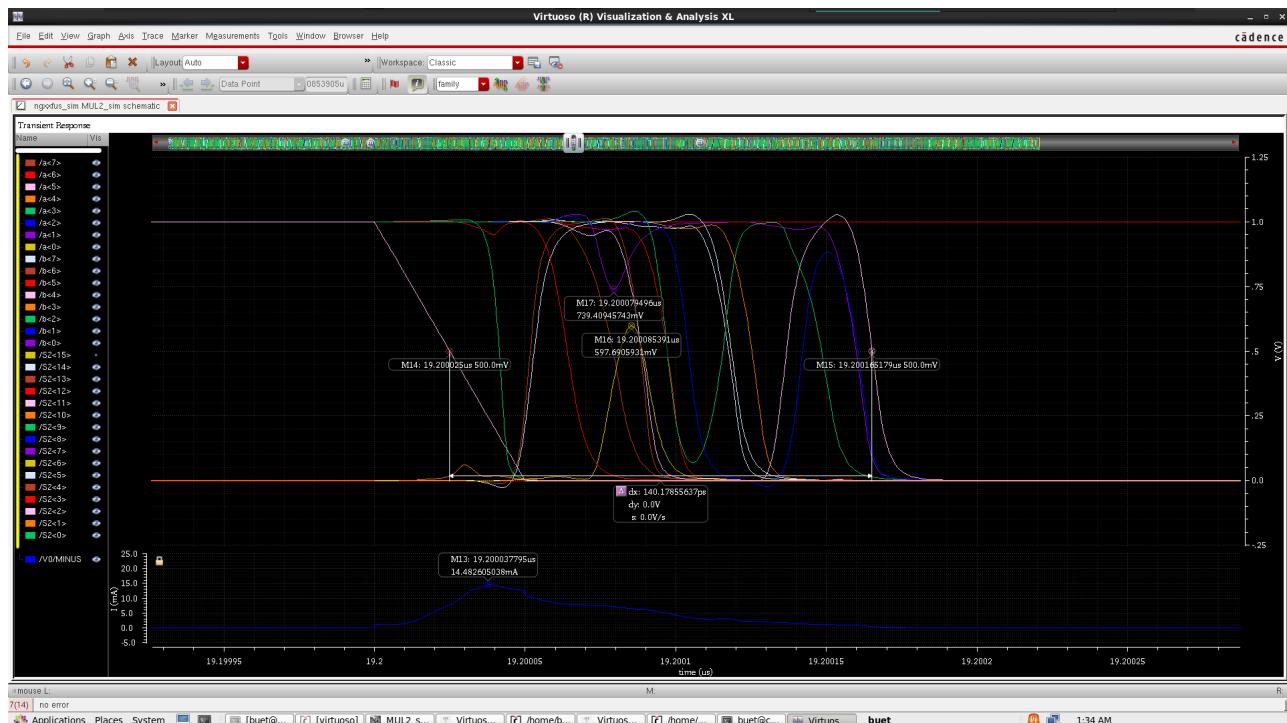
Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm $t=9,500025\mu s$.

Nhận xét:

- Độ trễ lan truyền: 126,9ps

- Công suất cực đại cục bộ: 2,8mW

#t=19,200025us

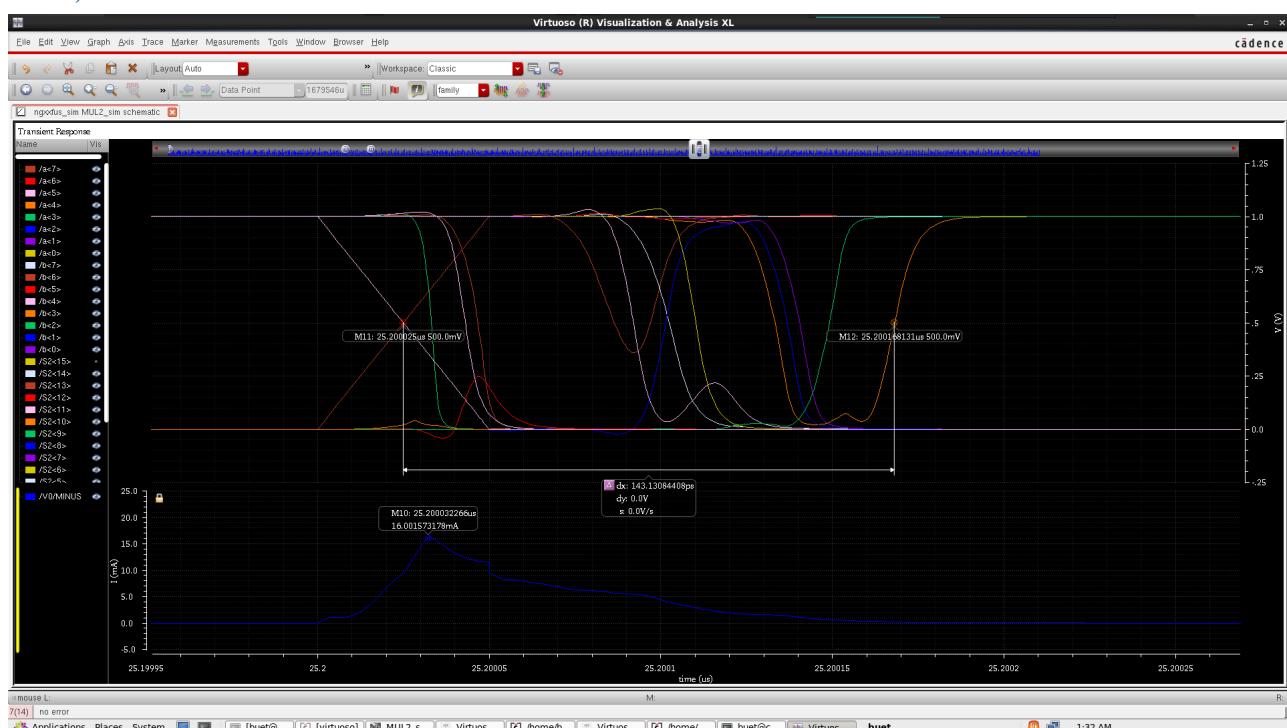


Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm $t=19,200025\mu s$.

Nhận xét:

- Độ trễ lan truyền: 140,2ps
- Công suất cực đại cục bộ: 14,5mW

#t=25,200025us

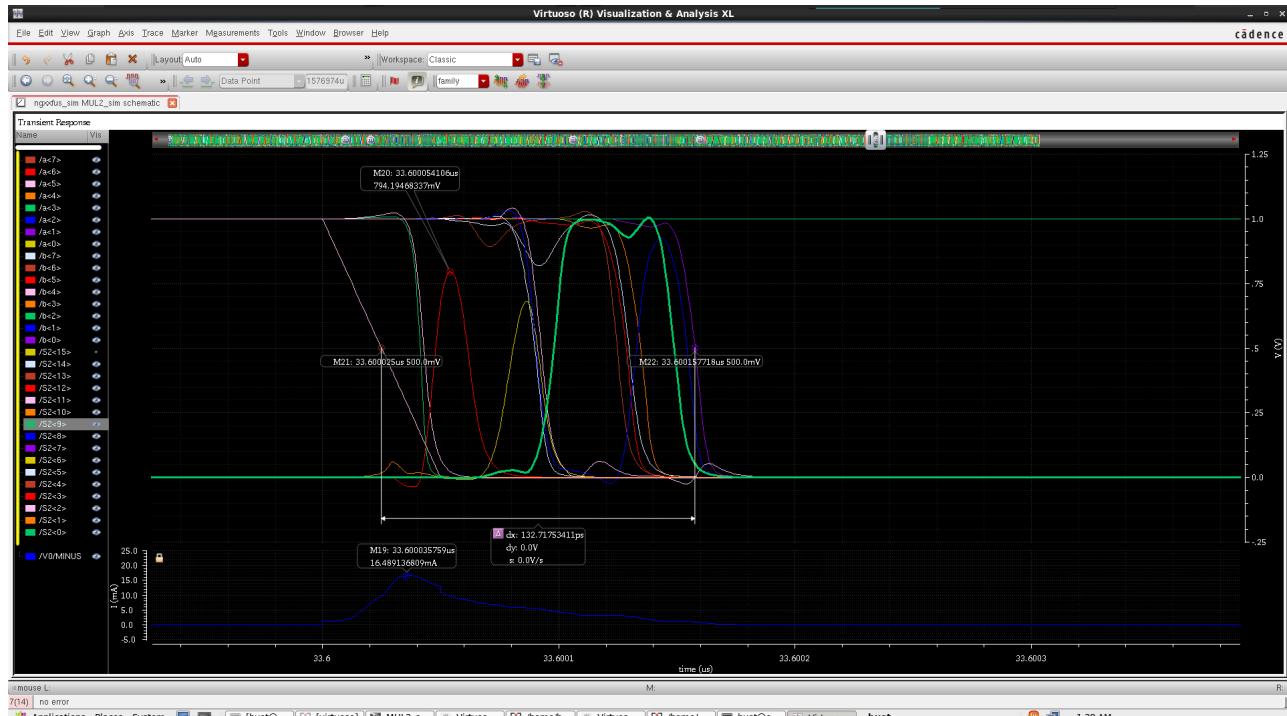


Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm $t=25,200025\mu s$.

Nhận xét:

- Độ trễ lan truyền: 143,1ps
- Công suất cực đại cục bộ: 16,0mW

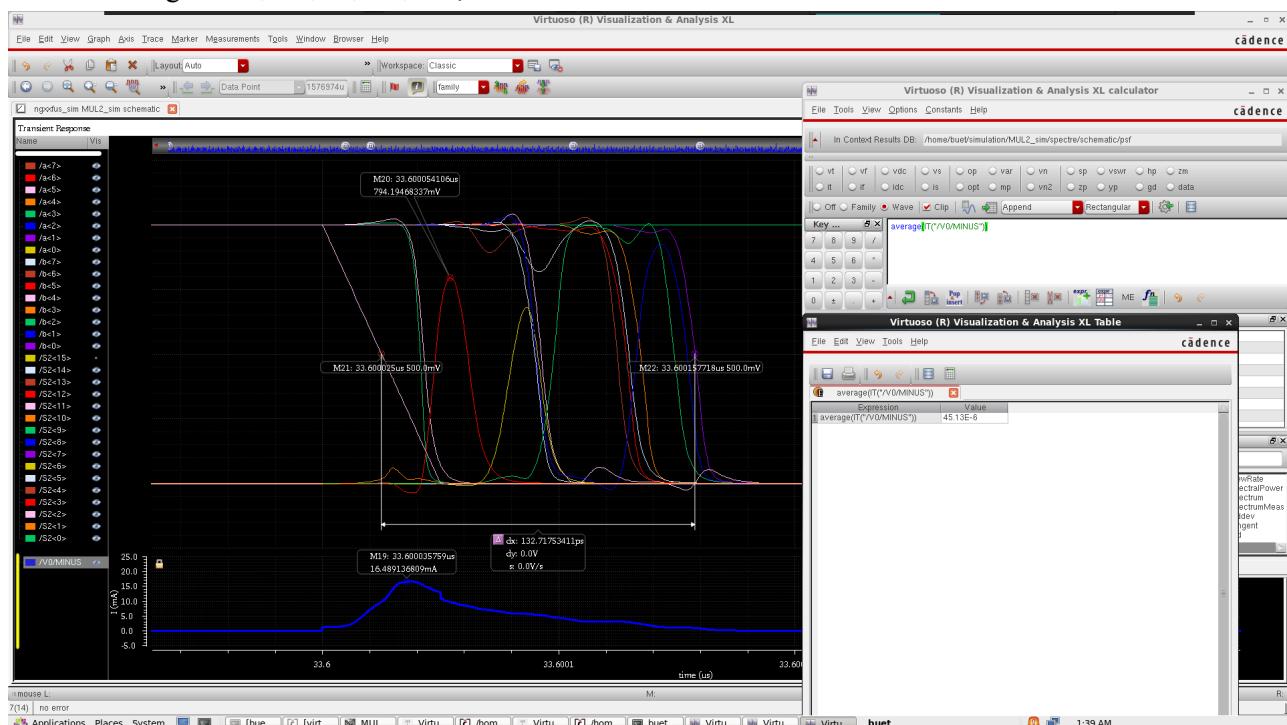
#t=33,600025us



Hình . Dạng sóng ngõ ra và dòng điện tiêu thụ của bộ nhân tại thời điểm $t=33,600025\mu s$.

Nhận xét:

- Độ trễ lan truyền: 132,7ps
- Công suất cực đại cục bộ: 16,5mW



Hình . Dạng sóng dòng điện tiêu thụ trong suốt quá trình mô phỏng và công suất trung bình.

Nhận xét:

- + Công suất cực đại: 20,8mW.
- + Công suất trung bình: 45,13mW.

4.4. So sánh, kết luận

4.4.1. So sánh độ trễ lan truyền

Lập bảng so sánh các giá trị độ trễ lan truyền được lấy mẫu ngẫu nhiên tại các thời điểm khác nhau:

Thời gian (us)	0	8,400025	9,580025	19,200025	25,200025	33,600025	MAX
MUL1 (ps)	218,2	195,7	148,4	148,4	192,0	225,0	225,0
MUL2 (ps)	126,5	114,3	126,9	140,2	143,1	132,7	143,1
Δ (ps)	91,7	81,4	21,5	8,2	48,9	92,3	81,9

Ghi chú:

- MUL1: Bộ nhân sử dụng CLA
- MUL2: Bộ nhân SCDM

Nhận xét:

- Bộ nhân SCDM có độ trễ lan truyền ngắn hơn bộ nhân dùng CLA.

4.4.2. So sánh công suất tiêu thụ

Thực hiện gộp dạng sóng dòng điện tiêu thụ của mạch nhân CLA (màu đỏ) và mạch nhân SCDM (màu xanh). Hình bên dưới mô tả dạng sóng của hai mạch nhân.



Hình . Dạng sóng dòng điện tiêu thụ của mạch nhân CLA (màu đỏ) và mạch nhân SCDM (màu xanh).

Nhận xét:

- Tổng thể mạch nhân CLA (màu đỏ) có công suất tiêu thụ lớn hơn mạch nhân SCDM (màu xanh).

Lập bảng so sánh các giá trị công suất tiêu thụ của mạch nhân CLA và mạch nhân SCDM được lấy mẫu ngẫu nhiên tại các thời điểm khác nhau:

Thời gian (us)	0	8,400025	9,580025	19,200025	25,200025	33,600025	Trung bình (0us-50us)
MUL1 (mW)	22,6	12,9	3,6	17,1	19,7	18,4	$62,85 \times 10^{-3}$
MUL2 (mW)	20,8	11,6	2,8	14,5	16,0	16,5	$45,13 \times 10^{-3}$
Δ (mW)	2,6	1,3	1,2	2,6	3,7	1,9	$17,72 \times 10^{-3}$

Ghi chú:

- MUL1: Bộ nhân sử dụng CLA
- MUL2: Bộ nhân SCDM

Nhận xét:

- Bộ nhân SCDM có công suất tiêu thụ ít hơn bộ nhân dùng CLA.

4.4.3. So sánh thời gian mô phỏng

Thực hiện mô phỏng với các thông số đã thiết lập như ở phần 4.1, lập bảng so sánh về thời gian cần thiết và dung lượng lưu trữ để mô phỏng ~40us.

	Thời gian mô phỏng (HH:MM:SS)	Dung lượng lưu trữ (GB)
Bộ nhân sử dụng CLA	10:40:57	19.1
Bộ nhân SCDM	05:26:46	13.6