

**BÁO CÁO**  
**TÌM HIỂU, TỔNG HỢP CÁC CÔNG CỤ**  
**PHÂN TÍCH THỜI GIAN TĨNH**  
**(STATIC TIMING ANALYSIS)**



```

ngxxfus@ngxxfus-X409FA [13:51:41] [OpenTimer] [master]
$ tree -L 1
.
├── benchmark
├── CMakeLists.txt
├── doc
├── example
├── image
├── inttest
├── LICENSE
├── licenses
├── main
├── ot
├── README.md
├── shan
├── unittest
└── wiki

12 directories, 3 files

```

Hình . Cây thư mục mã nguồn OpenTimer.

### 1.2.2. Chạy CMake để tạo Makefile

OpenTimer được xây dựng dựa trên CMake (<https://cmake.org/>). CMake là một công cụ giúp sinh Makefile từ tệp CMakeLists.txt. Công cụ này giúp đơn giản hóa việc tạo Makefile vì cú pháp Makefile khá phức tạp. Do đó, để tiếp tục, phải đảm bảo sự có mặt của cmake trong hệ thống. Lệnh bên dưới mô tả quá trình sử dụng cmake để tạo Makefile.

```

mkdir build
cd build
cmake ../

```

"cmake ../" có ý nghĩa rằng CMake để cấu hình dự án ở thư mục cha (../), và lưu kết quả cấu hình (Makefile) vào thư mục hiện tại.

```

ngxxfus@ngxxfus-X409FA [14:05:07] [build] [master]
$ pwd
/home/ngxxfus/OpenTimer/build
ngxxfus@ngxxfus-X409FA [14:05:08] [build] [master]
$ cmake ..
-- CMAKE_ROOT: /usr/share/cmake-3.28
-- The C compiler identification is GNU 13.3.0
-- The CXX compiler identification is GNU 13.3.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- CMAKE_HOST_SYSTEM: Linux-6.11.0-26-generic
-- CMAKE_CXX_COMPILER_VERSION: 13.3.0
-- CMAKE_BUILD_TYPE:
-- CMAKE_SYSTEM_NAME: Linux
-- CMAKE_CXX_COMPILER: /usr/bin/c++
-- CMAKE_CXX_FLAGS: -Wall -O2
-- CMAKE_CXX_FLAGS_DEBUG: -g
-- CMAKE_CXX_FLAGS_RELEASE: -O3 -DNDEBUG
-- CMAKE_CXX_FLAGS_RELWITHDEBINFO: -O2 -g -DNDEBUG
-- CMAKE_CXX_FLAGS_MINSIZEREL: -Os -DNDEBUG
-- CMAKE_EXE_LINKER_FLAGS:
-- CMAKE_INSTALL_PREFIX: /usr/local
-- CMAKE_MODULE_PATH:
-- OT_VERSION: 2.1.0
-- OT_HOME: /home/ngxxfus/OpenTimer
-- OT_BENCHMARK_DIR: /home/ngxxfus/OpenTimer/benchmark
-- OT_UNITTEST_DIR: /home/ngxxfus/OpenTimer/unittest
-- OT_LICENSE: /home/ngxxfus/OpenTimer/LICENSE
-- CMAKE_ARCHIVE_OUTPUT_DIRECTORY: /home/ngxxfus/OpenTimer/lib
-- PYTHON_EXECUTABLE: /usr/bin/python
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Success
-- Found Threads: TRUE
-- Found Tclsh: /bin/tclsh (found version "8.6")
-- Building executables ...
-- OT_LINK_FLAGS: OpenTimer;Threads;Threads
-- TAU15 executable: /home/ngxxfus/OpenTimer/bin/ot-tau15
-- TAU18 executable: /home/ngxxfus/OpenTimer/bin/ot-tau18
-- Utility executable: /home/ngxxfus/OpenTimer/bin/ot-utility
-- OpenTimer shell: /home/ngxxfus/OpenTimer/bin/ot-shell
-- Building unit tests ...
-- Building TAU15 integration tests ...
-- Building Shell integration tests ...
-- Configuring done (0.5s)
-- Generating done (0.0s)
-- Build files have been written to: /home/ngxxfus/OpenTimer/build
ngxxfus@ngxxfus-X409FA [14:05:13] [build] [master]
$

```

Hình . Quá trình chạy cmake để tạo Makefile.

### 1.2.3. Chạy Makefile để biên dịch thư viện và các tập tin thực thi khác

Tại thư mục build, thực hiện biên dịch thư viện và các tập tin thực thi thông qua lệnh **make**. Để tăng tốc quá trình biên dịch, thêm cờ **-j<số luồng CPU>** để Makefile tận dụng tối đa tài nguyên đa luồng của CPU.

```

ngxfus@ngxfus-X409FA [14:09:59] [build] [master ~]
$ make -j8
/usr/bin/cmake -S/home/ngxfus/OpenTimer -B/home/ngxfus/OpenTimer/build --check-build-system CMakeFiles/Makefile.cmake 0
/usr/bin/cmake -E cmake_progress_start /home/ngxfus/OpenTimer/build/CMakeFiles /home/ngxfus/OpenTimer/build/CMakeFiles/progress.marks
make -f CMakeFiles/Makefile2 all
make[1]: Entering directory '/home/ngxfus/OpenTimer/build'
make -f CMakeFiles/OpenTimer.dir/build.make CMakeFiles/OpenTimer.dir/depend
make[2]: Entering directory '/home/ngxfus/OpenTimer/build'
cd /home/ngxfus/OpenTimer/build && /usr/bin/cmake -E cmake_depends "Unix Makefiles" /home/ngxfus/OpenTimer /home/ngxfus/OpenTimer /home/ngxfus/OpenTimer/build /home/ngxfus/OpenTimer/build /home/ngxfus/OpenTimer/build/CMakeFiles/OpenTimer.dir/DependInfo.cmake --color-
make[2]: Leaving directory '/home/ngxfus/OpenTimer/build'
make -f CMakeFiles/OpenTimer.dir/build.make CMakeFiles/OpenTimer.dir/build
make[2]: Entering directory '/home/ngxfus/OpenTimer/build'
[ 1%] Building CXX object CMakeFiles/OpenTimer.dir/ot/timer/pfxt.cpp.o
[ 6%] Building CXX object CMakeFiles/OpenTimer.dir/ot/timer/clock.cpp.o
[ 6%] Building CXX object CMakeFiles/OpenTimer.dir/ot/timer/net.cpp.o
[ 6%] Building CXX object CMakeFiles/OpenTimer.dir/ot/timer/endpoint.cpp.o
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/OpenTimer.dir/ot/timer/pfxt.cpp.o -MF CMakeFiles/OpenTimer.dir/ot/timer/pfxt.cpp.o.d -o CMakeFiles/OpenTimer.dir/ot/timer/pfxt.cpp.o -c /home/ngxfus/OpenTimer/ot/timer/pfxt.cpp
[ 7%] Building CXX object CMakeFiles/OpenTimer.dir/ot/timer/sdc.cpp.o
[ 9%] Building CXX object CMakeFiles/OpenTimer.dir/ot/timer/sfxt.cpp.o
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/OpenTimer.dir/ot/timer/net.cpp.o -MF CMakeFiles/OpenTimer.dir/ot/timer/net.cpp.o.d -o CMakeFiles/OpenTimer.dir/ot/timer/net.cpp.o -c /home/ngxfus/OpenTimer/ot/timer/net.cpp
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/OpenTimer.dir/ot/timer/clock.cpp.o -MF CMakeFiles/OpenTimer.dir/ot/timer/clock.cpp.o.d -o CMakeFiles/OpenTimer.dir/ot/timer/clock.cpp.o -c /home/ngxfus/OpenTimer/ot/timer/clock.cpp
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/OpenTimer.dir/ot/timer/sdc.cpp.o -MF CMakeFiles/OpenTimer.dir/ot/timer/sdc.cpp.o.d -o CMakeFiles/OpenTimer.dir/ot/timer/sdc.cpp.o -c /home/ngxfus/OpenTimer/ot/timer/sdc.cpp
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/OpenTimer.dir/ot/timer/endpoint.cpp.o -MF CMakeFiles/OpenTimer.dir/ot/timer/endpoint.cpp.o.d -o CMakeFiles/OpenTimer.dir/ot/timer/endpoint.cpp.o -c /home/ngxfus/OpenTimer/ot/timer/endpoint.cpp
[ 11%] Building CXX object CMakeFiles/OpenTimer.dir/ot/timer/timer.cpp.o
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/OpenTimer.dir/ot/timer/sfxt.cpp.o -MF CMakeFiles/OpenTimer.dir/ot/timer/sfxt.cpp.o.d -o CMakeFiles/OpenTimer.dir/ot/timer/sfxt.cpp.o -c /home/ngxfus/OpenTimer/ot/timer/sfxt.cpp
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/OpenTimer.dir/ot/timer/timer.cpp.o -MF CMakeFiles/OpenTimer.dir/ot/timer/timer.cpp.o.d -o CMakeFiles/OpenTimer.dir/ot/timer/timer.cpp.o -c /home/ngxfus/OpenTimer/ot/timer/timer.cpp
[ 12%] Building CXX object CMakeFiles/OpenTimer.dir/ot/timer/unit.cpp.o
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/OpenTimer.dir/ot/timer/unit.cpp.o -MF CMakeFiles/OpenTimer.dir/ot/timer/unit.cpp.o.d -o CMakeFiles/OpenTimer.dir/ot/timer/unit.cpp.o -c /home/ngxfus/OpenTimer/ot/timer/unit.cpp

```

a .

```

[ 92%] Building CXX object CMakeFiles/path.dir/unittest/path.cpp.o
[ 93%] Building CXX object CMakeFiles/utility.dir/unittest/utility.cpp.o
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/utility.dir/unittest/utility.cpp.o -MF CMakeFiles/utility.dir/unittest/utility.cpp.o.d -o CMakeFiles/utility.dir/unittest/utility.cpp.o -c /home/ngxfus/OpenTimer/unittest/utility.cpp
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/path.dir/unittest/path.cpp.o -MF CMakeFiles/path.dir/unittest/path.cpp.o.d -o CMakeFiles/path.dir/unittest/path.cpp.o -c /home/ngxfus/OpenTimer/unittest/path.cpp
/usr/bin/c++ -I/home/ngxfus/OpenTimer -Wall -O2 -std=c++17 -MD -MT CMakeFiles/sizer.dir/example/sizer/sizer.cpp.o -MF CMakeFiles/sizer.dir/example/sizer/sizer.cpp.o.d -o CMakeFiles/sizer.dir/example/sizer/sizer.cpp.o -c /home/ngxfus/OpenTimer/example/sizer/sizer.cpp
make[2]: Leaving directory '/home/ngxfus/OpenTimer/build'
make[2]: Leaving directory '/home/ngxfus/OpenTimer/build'
[ 93%] Built target ot-shell
[ 93%] Built target unit
make[2]: Leaving directory '/home/ngxfus/OpenTimer/build'
[ 93%] Built target ot-tau15
[ 93%] Linking CXX executable /home/ngxfus/OpenTimer/bin/ot-utility
/usr/bin/cmake -E cmake_link_script CMakeFiles/ot-utility.dir/link.txt --verbose=1
/usr/bin/c++ -Wall -O2 CMakeFiles/ot-utility.dir/main/utility/utility.cpp.o -o /home/ngxfus/OpenTimer/bin/ot-utility /home/ngxfus/OpenTimer/lib/libOpenTimer.a
make[2]: Leaving directory '/home/ngxfus/OpenTimer/build'
[ 95%] Built target ot-utility
[ 96%] Linking CXX executable /home/ngxfus/OpenTimer/unittest/utility
/usr/bin/cmake -E cmake_link_script CMakeFiles/utility.dir/link.txt --verbose=1
/usr/bin/c++ -Wall -O2 CMakeFiles/utility.dir/unittest/utility.cpp.o -o /home/ngxfus/OpenTimer/unittest/utility /home/ngxfus/OpenTimer/lib/libOpenTimer.a
make[2]: Leaving directory '/home/ngxfus/OpenTimer/build'
[ 96%] Built target utility
[ 98%] Linking CXX executable /home/ngxfus/OpenTimer/example/sizer/sizer
/usr/bin/cmake -E cmake_link_script CMakeFiles/sizer.dir/link.txt --verbose=1
/usr/bin/c++ -Wall -O2 CMakeFiles/sizer.dir/example/sizer/sizer.cpp.o -o /home/ngxfus/OpenTimer/example/sizer/sizer /home/ngxfus/OpenTimer/lib/libOpenTimer.a
make[2]: Leaving directory '/home/ngxfus/OpenTimer/build'
[ 98%] Built target sizer
[100%] Linking CXX executable /home/ngxfus/OpenTimer/unittest/path
/usr/bin/cmake -E cmake_link_script CMakeFiles/path.dir/link.txt --verbose=1
/usr/bin/c++ -Wall -O2 CMakeFiles/path.dir/unittest/path.cpp.o -o /home/ngxfus/OpenTimer/unittest/path /home/ngxfus/OpenTimer/lib/libOpenTimer.a
make[2]: Leaving directory '/home/ngxfus/OpenTimer/build'
[100%] Built target path
make[1]: Leaving directory '/home/ngxfus/OpenTimer/build'
/usr/bin/cmake -E cmake_progress_start /home/ngxfus/OpenTimer/build/CMakeFiles 0
ngxfus@ngxfus-X409FA [14:13:00] [build] [master ~]

```

b.

Hình . (a, b) Quá trình biên dịch thư viện và tập tin thực thi của OpenTimer.

Make kết thúc với trạng thái không lỗi cho thấy quá trình biên dịch thành công và tạo ra các tập tin thực thi trong **opentimer-root-dir/bin/** và thư viện tĩnh (static library) **opentimer-root-dir/lib/libOpenTimer.a** (trong đó opentimer-root-dir là đường dẫn đến thư mục OpenTimer đã tải ở phần 1.2.1, khuyến nghị đặt đường dẫn đến OpenTimer-Root-Dir là ~/OpenTimer). Hình bên dưới mô tả cấu trúc thư mục mã nguồn OpenTimer sau khi biên dịch thông qua **make** và một số thư mục thường xuyên sử dụng (bin/ chứa các file thực thi; lib/ chứa thư viện tĩnh; ot/timer/ chứa các header C++).

```

ngxfus@ngxfus-X409FA [15:24:19] [OpenTimer] [master ~]
$ tree -L 1
|-- benchmark
|-- bin
|-- build
|-- CMakeLists.txt
|-- doc
|-- example
|-- img
|-- inttest
|-- lib
|-- LICENSE
|-- license
|-- main
|-- ot
|-- README.md
|-- sbin
|-- unittest
|-- wiki

15 directories, 3 files
ngxfus@ngxfus-X409FA [15:24:22] [OpenTimer] [master ~]
$ ls -l bin/
ot-shell  ot-tau15  ot-tau18  ot-utility
ngxfus@ngxfus-X409FA [15:24:27] [OpenTimer] [master ~]
$ ls -l lib/
libOpenTimer.a
ngxfus@ngxfus-X409FA [15:24:32] [OpenTimer] [master ~]
$ ls -l ot/timer/
src.cpp  otLib.cpp  clock.hpp  cprp.hpp  endpoint.cpp  gate.cpp  net.cpp  path.cpp  pfxt.cpp  pin.cpp  scc.cpp  sdc.cpp  sfxt.hpp  test.cpp  timer.cpp  unit.cpp
src.hpp  clock.cpp  cprp.cpp  dump.cpp  endpoint.hpp  gate.hpp  net.hpp  path.hpp  pfxt.hpp  pin.hpp  scc.hpp  sfxt.cpp  spef.cpp  test.hpp  timer.hpp  verilog.cpp

```

Hình . Cấu trúc thư mục mã nguồn OpenTimer sau khi biên dịch.

## 1.2.4. Chạy bài test đi kèm

Sau khi quá biên dịch thư viện và tập tin thực thi của OpenTimer hoàn thành, thực hiện bài test đi kèm của OpenTimer thông qua **make** (make test). OpenTimer sử dụng Doctest cho các bài kiểm tra đơn vị và TAU15 cho các bài kiểm tra tích hợp/kiểm tra hồi quy. Các bộ benchmark này được tạo ra bởi một công cụ định thời chuẩn công nghiệp và đang được sử dụng bởi nhiều nhà nghiên cứu EDA. hình bên dưới mô tả quá trình chạy test.

```
ngxfus@ngxfus-X489FA [15:08:03] [build] [master ~]
$ make test -j8
Running tests...
/usr/bin/ctest --force-new-ctest-process
Test project /home/ngxfus/OpenTimer/build
  Start 1: ut.utility
1/64 Test #1: ut.utility ..... Passed    0.00 sec
  Start 2: ut.path
2/64 Test #2: ut.path ..... Passed    0.00 sec
  Start 3: it.tau15.c3 slack
3/64 Test #3: it.tau15.c3 slack ..... Passed    0.07 sec
  Start 4: it.tau15.c3 path
4/64 Test #4: it.tau15.c3 path ..... Passed    0.06 sec
  Start 5: it.tau15.c17
5/64 Test #5: it.tau15.c17 ..... Passed    0.08 sec
  Start 6: it.tau15.c17 slack
6/64 Test #6: it.tau15.c17 slack ..... Passed    0.06 sec
  Start 7: it.tau15.c432
7/64 Test #7: it.tau15.c432 ..... Passed    0.07 sec
  Start 8: it.tau15.c499
8/64 Test #8: it.tau15.c499 ..... Passed    0.07 sec
  Start 9: it.tau15.c880
9/64 Test #9: it.tau15.c880 ..... Passed    0.09 sec
  Start 10: it.tau15.c1355
10/64 Test #10: it.tau15.c1355 ..... Passed    0.07 sec
  Start 11: it.tau15.c1908
11/64 Test #11: it.tau15.c1908 ..... Passed    0.07 sec
  Start 12: it.tau15.c2670
12/64 Test #12: it.tau15.c2670 ..... Passed    0.09 sec
  Start 13: it.tau15.c3540
13/64 Test #13: it.tau15.c3540 ..... Passed    0.09 sec
  Start 14: it.tau15.c5315
14/64 Test #14: it.tau15.c5315 ..... Passed    0.13 sec
  Start 15: it.tau15.c6288
15/64 Test #15: it.tau15.c6288 ..... Passed    0.15 sec
  Start 16: it.tau15.c7352
```

a.

```
  Start 49: it.shell.c7552
49/64 Test #49: it.shell.c7552 ..... Passed    0.61 sec
  Start 50: it.shell.c7552 slack
50/64 Test #50: it.shell.c7552 slack ..... Passed    0.70 sec
  Start 51: it.shell.simple
51/64 Test #51: it.shell.simple ..... Passed    0.09 sec
  Start 52: it.shell.s27
52/64 Test #52: it.shell.s27 ..... Passed    0.14 sec
  Start 53: it.shell.s344
53/64 Test #53: it.shell.s344 ..... Passed    0.13 sec
  Start 54: it.shell.s349
54/64 Test #54: it.shell.s349 ..... Passed    0.13 sec
  Start 55: it.shell.s386
55/64 Test #55: it.shell.s386 ..... Passed    0.11 sec
  Start 56: it.shell.s400
56/64 Test #56: it.shell.s400 ..... Passed    0.39 sec
  Start 57: it.shell.s510
57/64 Test #57: it.shell.s510 ..... Passed    0.15 sec
  Start 58: it.shell.s526
58/64 Test #58: it.shell.s526 ..... Passed    0.10 sec
  Start 59: it.shell.s1196
59/64 Test #59: it.shell.s1196 ..... Passed    0.17 sec
  Start 60: it.shell.s1494
60/64 Test #60: it.shell.s1494 ..... Passed    0.17 sec
  Start 61: it.shell.ac97 ctrl
61/64 Test #61: it.shell.ac97 ctrl ..... Passed    0.91 sec
  Start 62: it.shell.aes core
62/64 Test #62: it.shell.aes core ..... Passed    2.57 sec
  Start 63: it.shell.des perf
63/64 Test #63: it.shell.des perf ..... Passed    5.20 sec
  Start 64: it.shell.vga lcd
64/64 Test #64: it.shell.vga lcd ..... Passed   11.19 sec

100% tests passed, 0 tests failed out of 64

Total Test time (real) =  58.66 sec
ngxfus@ngxfus-X489FA [15:12:46] [build] [master ~]
```

b.

Hình . Quá trình chạy test đi kèm của OpenTimer.

1.2.5. Thêm thư mục thực thi vào đường dẫn PATH

Việc thêm thư mục chứa tập tin thực thi (trong đó có ot-shell) vào đường dẫn hệ thống làm các tập tin thực thi này có thể được gọi từ bất cứ đâu. Đoạn mã bên dưới mô tả cách mà các tập tin thực thi này được thêm vào đường dẫn hệ thống.

```
export PATH=$PATH:/home/ngxfus/OpenTimer/bin
```

1.3. Chuẩn bị cho STA

1.3.1. Mô-đun MY\_FUNCTION

Mô-đun dùng để STA được mô tả bằng Verilog bên dưới sẽ được tổng hợp bởi YoSys và ánh xạ đến thư viện NangateOpenCellLibrary\_typical.lib và sẽ được đánh giá STA trong các bài kiểm thử ở các phần kế tiếp.

```
my_function.v

module MY_FUNCTION (
    input B, C, A,
    output Y
);
    reg internal_out, _temp_var;

    always @( posedge B )
    begin
        if ( A == 0 )
            internal_out <= 0;
```

```

else if (C == 1)begin
    internal_out <= (B|(~_temp_var))^B;
    _temp_var <= internal_out & B;
end else begin
    internal_out <= (_temp_var|(~C))^B;
    _temp_var <= internal_out & A;
end
end
assign Y = internal_out;
endmodule

```

### 1.3.2. Tổng hợp mô-đun MY\_FUNCTION

Module MY\_FUNCTION sẽ không thể nhập (đọc) trực tiếp vào OpenTimer vì OpenTimer chỉ hiểu thiết kế ở mức gate-netlist. Do đó MY\_FUNCTION phải được tổng hợp bởi YoSys. Trong các lần thử không được ghi nhận vào trong báo cáo này, một số thiết kế được tổng hợp bởi YoSys vẫn còn sót lại phép gán **assign** (lưu ý rằng thiết kế vẫn hoạt động đúng khi mô phỏng ADE trong Cadence Virtuoso) nhưng sẽ làm cho OpenTimer không thể đọc được. Việc còn sót lại phép gán liên tục - assign hay không phụ thuộc nhiều vào thư viện cell tiêu chuẩn dùng để ánh xạ. Ví dụ, cũng trong các lần thử không được ghi nhận vào báo cáo này, nếu dùng thư viện cell (các cổng logic cơ bản) tiêu chuẩn 180um - osu018\_stdcells.lib thì thiết kế sau tổng hợp không để lại assign, nhưng nếu dùng thư viện cell tiêu chuẩn NangateOpenCellLibrary\_typical.lib để lại phép gán liên tục assign. Trong trường hợp sau khi tổng hợp gate-netlist thông qua YoSys mà vẫn còn **assign**, đầu tiên cần xem xét lại thiết kế Verilog, nếu vẫn không được thì có thể thử gán thủ công các tín hiệu này thông qua một cổng đệm/dệm đảo. Đoạn mã viết bằng TCL bên dưới dùng để tổng hợp module MY\_FUNCTION.

```

my_function_synthesis.tcl

# 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'

set working_dir "/mnt/sda1/DOC/HCMUTE_DOCs/NH_2024_2025_HK_02/DA1/OpenTimerTest/"
set output_dir "/mnt/sda1/DOC/HCMUTE_DOCs/NH_2024_2025_HK_02/DA1/OpenTimerTest/"
set lib_path "/mnt/sda1/DOC/HCMUTE_DOCs/NH_2024_2025_HK_02/DA1/OpenTimerTest/NangateOpenCellLibrary_typical.lib"

set synth_output "my_function_synth.v"
set mapped_output "my_function_mapped_sta.v"
set top_module "MY_FUNCTION"
set verilog_files {
    my_function.v
}
foreach file $verilog_files {
    yosys read_verilog $working_dir$file
}
yosys hierarchy -check -top $top_module
yosys synth -top $top_module -flatten
yosys write_verilog -noattr $output_dir$synth_output
yosys read_liberty -lib $lib_path
yosys opt -purge
yosys dfflibmap -liberty $lib_path
yosys abc -liberty $lib_path
yosys techmap -map +/techmap.v
yosys check
yosys opt_clean
yosys opt -purge
yosys clean
yosys write_verilog -noattr $output_dir$mapped_output

```

Sau khi tổng hợp, MY\_FUNCTION ở mức gate-netlist, thu được đoạn mã Verilog bên dưới:

my_function_mapped_sta.v	my_function_mapped_sta.v (tiếp tục)	my_function_mapped_sta.v (tiếp tục)
<pre>/* Generated by Yosys 0.52+45 (git sha1      19845be85,      g++ 13.3.0-6ubuntu2~24.04 -fPIC -O3) */  module MY_FUNCTION(B, C, A, Y); wire _00_; wire _01_; wire _02_; wire _03_; wire _04_; wire _05_; wire _06_; wire _07_; input A; wire A; input B; wire B; input C; wire C; wire _temp_var; output Y; wire Y;</pre>	<pre>INV_X1_08_( .A(C), .ZN(_04_) ); INV_X1_09_( .A(_temp_var), .ZN(_05_) ); INV_X1_10_( .A(A), .ZN(_02_) ); OAI211_X1_11_( .A(A), .B(Y), .C1(_04_), .C2(B), .ZN(_03_) ); OAI21_X1_12_( .A(_03_), .B1(A), .B2(_05_), .ZN(_00_) );</pre>	<pre>AOI211_X1_13_( .A(B), .B(_02_), .C1(C), .C2(_temp_var), .ZN(_01_) ); DFF_X1_14_( .CK(B), .D(_00_), .Q(_temp_var), .QN(_07_) ); DFF_X1_15_( .CK(B), .D(_01_), .Q(Y), .QN(_06_) ); endmodule</pre>

### 1.3.3. Ràng buộc thời gian - SDC

Để thực hiện đánh giá STA, cần ít nhất thêm một tập tin định nghĩa các ràng buộc (constraint) - SDC cho ngõ ra, ngõ vào của thiết kế. Đoạn mã bên dưới mô tả file SDC cần thiết cho ngõ vào (bao gồm xung CLK) và ngõ ra.

my_function_mapped_timing.sdc
<pre># Create clock 50 time-unit create_clock -period 50 -name B [get_ports B]  # Set rasing constraint for `B` set_input_delay 0 -min -rise [get_ports B] -clock B set_input_delay 20 -max -rise [get_ports B] -clock B # Set falling constraint for `B` set_input_delay 0 -min -fall [get_ports B] -clock B set_input_delay 25 -max -fall [get_ports B] -clock B  # Set rasing constraint for `A` set_input_delay 0 -min -rise [get_ports A] -clock B set_input_delay 20 -max -rise [get_ports A] -clock B # Set falling constraint for `A` set_input_delay 0 -min -fall [get_ports A] -clock B set_input_delay 25 -max -fall [get_ports A] -clock B</pre>

```

# Set rasing constraint for `C`
set_input_delay 0 -min -rise [get_ports C] -clock B
set_input_delay 20 -max -rise [get_ports C] -clock B
# Set falling constraint for `C`
set_input_delay 0 -min -fall [get_ports C] -clock B
set_input_delay 25 -max -fall [get_ports C] -clock B

# Set transistion constraint for `B`
set_input_transition 10 -min -rise [get_ports B] -clock B
set_input_transition 15 -min -fall [get_ports B] -clock B
set_input_transition 20 -max -rise [get_ports B] -clock B
set_input_transition 25 -max -fall [get_ports B] -clock B

# Set transistion constraint for `A`
set_input_transition 30 -min -rise [get_ports A] -clock B
set_input_transition 30 -min -fall [get_ports A] -clock B
set_input_transition 40 -max -rise [get_ports A] -clock B
set_input_transition 40 -max -fall [get_ports A] -clock B

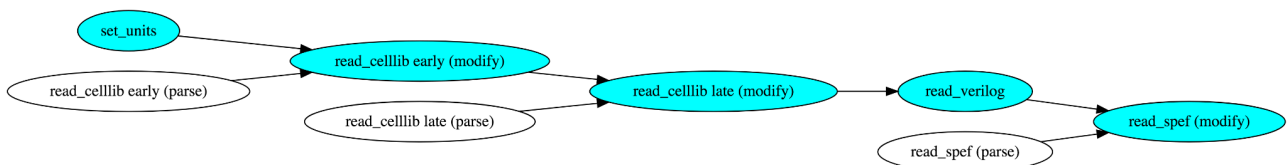
# Set transistion constraint for `ss`
set_input_transition 10 -min -rise [get_ports C] -clock B
set_input_transition 15 -min -fall [get_ports C] -clock B
set_input_transition 10 -max -rise [get_ports C] -clock B
set_input_transition 15 -max -fall [get_ports C] -clock B

set_load -pin_load 4 [get_ports Y]
set_output_delay -10 -min -rise [get_ports Y] -clock B
set_output_delay -10 -min -fall [get_ports Y] -clock B
set_output_delay 30 -max -rise [get_ports Y] -clock B
set_output_delay 30 -max -fall [get_ports Y] -clock B

```

#### 1.4. Sử dụng OpenTimer - ot-shell

OpenTimer cung cấp ot-shell cho phép đọc tập tin Verilog, đọc thư viện, đọc ràng buộc SDC và cung cấp STA thông qua lệnh report\_timing, ... thông qua giao diện dòng lệnh (CLI). Cốt lõi, OpenTimer sử dụng một biểu đồ phả hệ (lineage graph) để ghi lại chuỗi các thao tác gọi builder (builder: các hàm chuẩn bị dữ liệu đầu vào cho việc phân tích timing). Hình bên dưới mô tả một biểu đồ phả hệ (lineage graph).



Hình . Biểu đồ phả hệ (lineage graph).

Bảng . Các lệnh thường dùng trong ot-shell.

Lệnh	Loại	Tham số	Giải thích	Ví dụ
read_celllib	builder	[-min   -max] file.lib	Đọc thư viện cell cho phân tích set-up/hold.	read_celllib mylib.lib
read_verilog	builder	file.v	Đọc gate-netlist Verilog	read_verilog mynetlist.v



read_spef	builder	file.spef	Đọc các tụ ký sinh trong định dạng SPEF	read_spef myparasitics.spef
read_sdc	builder	file.sdc	Đọc tệp Synopsys Design Constraint	read_sdc myrule.sdc
update_timing	action	KHÔNG CÓ	Cập nhật thời gian	update_timing
report_timing	action	[-num_paths k]	Báo cáo các critical path.	report_timing -num_paths 10
report_tns	action	KHÔNG CÓ	Báo cáo tổng độ trễ âm (total negative slack)	report_tns
report_wns	action	KHÔNG CÓ	Báo cáo độ trễ âm tồi tệ nhất (worst negative slack)	report_wns
dump_graph	accessor	[-o file]	Xuất đồ thị thời gian sang định dạng DOT	dump_graph
dump_timer	accessor	[-o file]	Xuất thống kê thiết kế	dump_timer

Các lệnh khác: <https://github.com/OpenTimer/OpenTimer/blob/master/wiki/home.m>

Đoạn mã bên dưới mô tả quá trình sử dụng ot-shell (bao gồm đọc thư viện, đọc netlist, đọc SDC) để đánh giá STA.

```
read_celllib ./NangateOpenCellLibrary_typical.lib
read_verilog my_function_mapped_sta.v
read_sdc my_function_mapped_timing.sdc
report_timer
```

```
ot> read_celllib ./NangateOpenCellLibrary_typical.lib
ot> read_verilog my_function_mapped_sta.v
ot> read_sdc my_function_mapped_timing.sdc
ot> report_timer
[5:42:00 PM -05-27 16:13:38 obsolete.cpp:67] "report_timer" is obsolete, use "dump_timer"
OpenTimer 2.1.0
Time unit      : 1e-09 s
Capacitance unit : 1e-15 F
Voltage unit    : 1 V
Resistance unit  : 1e+03 Ohm
Current unit    : 0.001 A
Power unit      : 1e-09 W
Voltage         : 1.1
# Pins          : 34
# P0s           : 1
# P1s           : 3
# Gates         : 8
# Nets          : 13
# Arcs          : 69
# SCCs          : 2
# Tests        : 2
# Cells         : 135
ot> report_timing
I 59072 25-05-27 16:13:52 celllib.cpp:34] loading celllib "./NangateOpenCellLibrary_typical.lib"
I 63168 25-05-27 16:13:52 verilog.cpp:14] loading netlist "my_function_mapped_sta.v"
I 54976 25-05-27 16:13:52 sdc.cpp:35] loading sdc "my_function_mapped_timing.sdc" ...
I 59072 25-05-27 16:13:52 celllib.cpp:79] merged with library "NangateOpenCellLibrary" [cells:135]
I 59072 25-05-27 16:13:52 celllib.cpp:79] merged with library "NangateOpenCellLibrary" [cells:135]
W 59072 25-05-27 16:13:52 timer.cpp:561] can't insert PI A (already existed)
W 59072 25-05-27 16:13:52 timer.cpp:561] can't insert PI B (already existed)
W 59072 25-05-27 16:13:52 timer.cpp:561] can't insert PI C (already existed)
W 59072 25-05-27 16:13:52 timer.cpp:602] can't insert P0 Y (already existed)
W 59072 25-05-27 16:13:52 timer.cpp:140] gate_08 already existed
W 59072 25-05-27 16:13:52 timer.cpp:140] gate_09 already existed
W 59072 25-05-27 16:13:52 timer.cpp:140] gate_10 already existed
W 59072 25-05-27 16:13:52 timer.cpp:140] gate_11 already existed
W 59072 25-05-27 16:13:52 timer.cpp:140] gate_12 already existed
W 59072 25-05-27 16:13:52 timer.cpp:140] gate_13 already existed
W 59072 25-05-27 16:13:52 timer.cpp:140] gate_14 already existed
W 59072 25-05-27 16:13:52 timer.cpp:140] gate_15 already existed
I 59072 25-05-27 16:13:52 verilog.cpp:21] added verilog module "MY_FUNCTION" [gates:8]
I 59072 25-05-27 16:13:52 sdc.cpp:21] added 30 sdc commands
Startpoint      : _15_~CK
Endpoint        : Y
Analysis type : min
-----
Type      Delay      Time      Dir      Description
-----
pin       0.000      0.000      0.000 rise   _15_~CK (DFF_X1)
pin       0.262      0.262      70.756 fall  _15_~Q (DFF_X1)
port     0.000      0.262      0.000 fall   Y
arrival   0.262      0.262      data arrival time
port     10.000     10.000      output port delay
required 10.000     10.000      data required time
-----
slack      -9.738      VIOLATED
ot>
```

Hình . Phân tích STA mô-đun MY\_FUNCTION.

Nhận xét: Thiết kế vi phạm ràng buộc về thời gian (set-up time).

## 1.5. Sử dụng OpenTimer - C++API

Ngoài ot-shell, OpenTimer cung cấp C++ API thông qua lớp **Timer**, là điểm bắt đầu để tích hợp STA vào dự án. Tất cả các phương thức công khai đều an toàn trong môi trường đa luồng nhờ cơ chế lineage. API được chia thành ba nhóm: builder để cấu hình, action để thực thi và accessor để truy xuất kết quả.

### 1.5.1. Chương trình C++ dùng để phân tích STA

Lớp Timer được định nghĩa bên trong tập tin header **opentimer-root-dir/ot/timer/timer.hpp**. Trong đó, opentimer-root-dir là ~/OpenTimer/. Ngoài ra không gian tên đi kèm với các C++API của OpenTimer là **ot**. Đoạn mã bên dưới mô tả cách sử dụng C++API của OpenTimer để phân tích STA:

my\_function\_mapped\_sta.cpp

```
#include <ot/timer/timer.hpp>
#include <filesystem>
#include <iostream>
namespace fs = std::filesystem;
int main(int argc, char *argv[])
{
    ot::Timer timer;
    fs::path lib_dir_path = "./";
    fs::path lib_file = "NangateOpenCellLibrary_typical.lib";
    fs::path lib_full_path = lib_dir_path / lib_file;
    fs::path verilog_file_path = "my_function_mapped_sta.v";
    fs::path sdc_file_path = "my_function_mapped_timing.sdc";

    timer.read_celllib(lib_full_path, ot::MIN)
        .read_celllib(lib_full_path, ot::MAX)
        .read_verilog(verilog_file_path)
        .read_sdc(sdc_file_path);

    auto paths = timer.report_timing(5);
    if (paths.size())
        for (size_t i = 0; i < paths.size(); ++i) {
            std::cout << "#critical-path-" << i << "#####\n";
            std::cout << paths[i] << '\n';
        }
    else
        std::cout << "\nNo critical path found!\n";

    std::cout << "\n#timer.dump_at(std::cout)#####\n";
    timer.dump_at(std::cout);

    std::cout << "\n#timer.dump_power(std::cout)#####\n";
    timer.dump_power(std::cout);

    std::cout << "\n#timer.dump_graph(std::cout)#####\n";
    timer.dump_graph(std::cout);
    return 0;
}
```

### 1.5.2. Biên dịch C++

Vì thư viện cung cấp bởi OpenTimer không phải là thư viện chuẩn nên cần phải thêm thư viện tĩnh (cho linker) và các tập tin header khi biên dịch. Đoạn mã Makefile bên dưới giúp biên dịch, thực thi và xóa các tập tin thực thi.

Makefile

```

SRC = my_function_mapped_sta.cpp
OUT = my_function_mapped_sta
CXX = g++
CXXFLAGS = -std=c++17 -Wall -I/home/ngxxfus/OpenTimer/
LDFLAGS = -L/home/ngxxfus/OpenTimer/lib -l:libOpenTimer.a
LDLIBS =
all: $(OUT)
$(OUT): $(SRC)
    $(CXX) $(CXXFLAGS) $(SRC) -o $(OUT) $(LDFLAGS) $(LDLIBS)
exec:
    ./$(OUT)
clean:
    rm -f $(OUT)

```

Sau đó biên dịch và chạy chương trình C++ bằng lệnh make. Đoạn mã bên dưới mô tả quá trình biên dịch và thực thi chương trình C++.

```
make clean all exec
```

### 1.5.3. Kết quả đánh giá STA

Sau khi chạy lệnh make, kết quả thu được được mô tả qua các hình bên dưới.

```

ngxxfus@ngxxfus-X809FA [17:19:30] [OpenTimerTest] [master ~]
$ make clean all exec
rm -f my_function_mapped_sta
g++ -std=c++17 -Wall -I/home/ngxxfus/OpenTimer/ my_function_mapped_sta.cpp -o my_function_mapped_sta -L/home/ngxxfus/OpenTimer/lib -l:libOpenTimer.a
./my_function_mapped_sta
I 50880 25-05-27 17:19:51 celllib.cpp:34] loading celllib "./NangateOpenCellLibrary_typical.lib"
I 54976 25-05-27 17:19:51 celllib.cpp:34] loading celllib "./NangateOpenCellLibrary_typical.lib"
I 46784 25-05-27 17:19:51 verilog.cpp:14] loading netlist "my_function_mapped_sta.v"
I 63168 25-05-27 17:19:51 sdc.cpp:35] loading sdc "my_function_mapped_timing.sdc" ...
I 50880 25-05-27 17:19:52 unit.cpp:244] use celllib time unit 1e-09 s
I 50880 25-05-27 17:19:52 unit.cpp:258] use celllib capacitance unit 1e-15 F
I 50880 25-05-27 17:19:52 unit.cpp:272] use celllib current unit 0.001 A
I 50880 25-05-27 17:19:52 unit.cpp:286] use celllib voltage unit 1 V
I 50880 25-05-27 17:19:52 unit.cpp:300] use celllib resistance unit 1000 Ohm
I 50880 25-05-27 17:19:52 unit.cpp:314] use celllib power unit 1e-09 W
I 50880 25-05-27 17:19:52 celllib.cpp:66] added min celllib "NangateOpenCellLibrary" [cells:135]
I 54976 25-05-27 17:19:52 celllib.cpp:66] added max celllib "NangateOpenCellLibrary" [cells:135]
I 54976 25-05-27 17:19:52 verilog.cpp:21] added verilog module "MY_FUNCTION" [gates:8]
I 54976 25-05-27 17:19:52 sdc.cpp:21] added 30 sdc commands
#critical-path-0#####
Startpoint   : _15 :CK
Endpoint     : Y
Analysis type : min
-----
      Type      Delay      Time      Dir      Description
-----
      pin        0.000      0.000      0.000      rise   _15 :CK (DFF_X1)
      pin        0.262      0.262      70.756      fall   _15 :Q (DFF_X1)
      port        0.000      0.262      0.000      fall   Y
      arrival     0.262
      data arrival time
-----
      port       10.000     10.000
      output port delay
      required    10.000
      data required time
-----
      slack      -9.738
      VIOLATED
#critical-path-1#####
Startpoint   : _15 :CK
Endpoint     : Y
Analysis type : min
-----
      Type      Delay      Time      Dir      Description
-----
      pin        0.000      0.000      0.000      rise   _15 :CK (DFF_X1)
      pin        0.363      0.363      70.174      rise   _15 :Q (DFF_X1)
      port        0.000      0.363      0.000      rise   Y
      arrival     0.363
      data arrival time
-----
      port       10.000     10.000
      output port delay
      required    10.000
      data required time
-----
      slack      -9.637
      VIOLATED

```

a.

```
#critical-path-2#####
Startpoint : _15_:CK
Endpoint : Y
Analysis type : max
-----
Type      Delay      Time      Dir      Description
-----
pin       20.000      20.000      0.000 rise   _15_:CK (DFF_X1)
pin       0.614       20.614     136.323 rise   _15_:Q (DFF_X1)
port      0.000       20.614      0.000 rise   Y
arrival   0.000       20.614      data arrival time
-----
port      20.000      20.000      output port delay
required  20.000      20.000      data required time
-----
slack     -0.614       VIOLATED

#critical-path-3#####
Startpoint : _15_:CK
Endpoint : Y
Analysis type : max
-----
Type      Delay      Time      Dir      Description
-----
pin       20.000      20.000      0.000 rise   _15_:CK (DFF_X1)
pin       0.419       20.419     137.691 fall   _15_:Q (DFF_X1)
port      0.000       20.419      0.000 fall   Y
arrival   0.000       20.419      data arrival time
-----
port      20.000      20.000      output port delay
required  20.000      20.000      data required time
-----
slack     -0.419       VIOLATED

#timer.dump at(std::cout)#####
Arrival time [pins:34]
-----
E/R      E/F      L/R      L/F      Pin
-----
n/a      n/a      n/a      n/a      _15_:
-1.579   -0.964   30.118   30.923   _15_:D
0.000    0.000   20.000   25.000   _15_:CK
0.363    0.262   20.614   20.419   _15_:Q
0.233    0.315   20.390   20.554   _15_:QN
0.000    0.000   20.000   25.000   _11_:C2
0.000    0.000   20.000   25.000   _11_:A
0.000    0.000   20.000   25.000   _12_:B1
4.797    -2.780   31.389   16.287   _13_:B
n/a      n/a      n/a      n/a      _14_:
0.359    0.258   20.611   20.415   _09_:A
0.000    0.000   20.000   25.000   _08_:A
0.000    0.000   20.000   25.000   _09_:C
0.269    0.367   20.426   20.620   _09_:ZN
2.394    -0.921   27.394   19.079   _08_:ZN
0.000    0.000   20.000   25.000   _10_:A
0.363    0.262   20.614   20.419   Y
0.000    0.000   20.000   25.000   B
0.363    0.262   20.614   20.419   _11_:B
-1.579   -0.964   30.118   30.923   _13_:ZN
0.000    0.000   20.000   25.000   _14_:CK
0.000    0.000   20.000   25.000   A
2.394    -0.921   27.394   19.079   _11_:C1
0.269    0.367   20.426   20.620   _12_:B2
0.294    -2.310   33.839   31.892   _12_:ZN
4.797    -2.780   31.389   16.287   _10_:ZN
-0.589   0.275    32.076   27.393   _11_:ZN
-0.589   0.275    32.076   27.393   _12_:A
0.000    0.000   20.000   25.000   _13_:C1
0.000    0.000   20.000   25.000   _13_:A
0.233    0.315   20.390   20.554   _14_:QN
0.359    0.258   20.611   20.415   _14_:Q
0.359    0.258   20.611   20.415   _13_:C2
0.294    -2.310   33.839   31.892   _14_:D
-----
```

c.

```
#timer.dump at(std::cout)#####
Arrival time [pins:34]
-----
E/R      E/F      L/R      L/F      Pin
-----
n/a      n/a      n/a      n/a      _15_:
-1.579   -0.964   30.118   30.923   _15_:D
0.000    0.000   20.000   25.000   _15_:CK
0.363    0.262   20.614   20.419   _15_:Q
0.233    0.315   20.390   20.554   _15_:QN
0.000    0.000   20.000   25.000   _11_:C2
0.000    0.000   20.000   25.000   _11_:A
0.000    0.000   20.000   25.000   _12_:B1
4.797    -2.780   31.389   16.287   _13_:B
n/a      n/a      n/a      n/a      _14_:
0.359    0.258   20.611   20.415   _09_:A
0.000    0.000   20.000   25.000   _08_:A
0.000    0.000   20.000   25.000   _09_:C
0.269    0.367   20.426   20.620   _09_:ZN
2.394    -0.921   27.394   19.079   _08_:ZN
0.000    0.000   20.000   25.000   _10_:A
0.363    0.262   20.614   20.419   Y
0.000    0.000   20.000   25.000   B
0.363    0.262   20.614   20.419   _11_:B
-1.579   -0.964   30.118   30.923   _13_:ZN
0.000    0.000   20.000   25.000   _14_:CK
0.000    0.000   20.000   25.000   A
2.394    -0.921   27.394   19.079   _11_:C1
0.269    0.367   20.426   20.620   _12_:B2
0.294    -2.310   33.839   31.892   _12_:ZN
4.797    -2.780   31.389   16.287   _10_:ZN
-0.589   0.275    32.076   27.393   _11_:ZN
-0.589   0.275    32.076   27.393   _12_:A
0.000    0.000   20.000   25.000   _13_:C1
0.000    0.000   20.000   25.000   _13_:A
0.233    0.315   20.390   20.554   _14_:QN
0.359    0.258   20.611   20.415   _14_:Q
0.359    0.258   20.611   20.415   _13_:C2
0.294    -2.310   33.839   31.892   _14_:D
-----
```

d.

#timer.dump_power(std::cout)#####				#timer.dump_graph(std::cout)#####							
Switch	Internal	Pin		digraph TimingGraph {							
0.000	0.000	15	:Pin	"15 :";				"15 :0" -> "11 :B";			
1.101	0.000	15	:D	"15 :D";				"15 :0" -> "Y";			
0.903	119.295	15	:CK	"15 :CK";				"15 :D" -> "15 :D";			
0.000	0.000	15	:Q	"15 :Q";				"15 :CK" -> "15 :D";			
0.000	0.000	15	:QN	"15 :QN";				"15 :D" -> "15 :CK";			
1.540	208.581	11	:C2	"11 :C2";				"15 :CK" -> "15 :CK";			
1.573	525.240	11	:A	"11 :A";				"15 :CK" -> "15 :CK";			
1.561	497.037	12	:B1	"12 :B1";				"15 :CK" -> "15 :QN";			
1.564	42.548	13	:B	"13 :B";				"15 :D" -> "15 :D";			
0.000	0.000	14	:	"14 :";				"15 :CK" -> "15 :0";			
1.625	0.952	09	:A	"09 :A";				"15 :QN" -> "15 :QN";			
1.625	245.699	08	:A	"08 :A";				"B" -> "14 :CK";			
0.000	0.000	C	:	"C";				"12 :ZN" -> "14 :D";			
0.000	0.000	09	:ZN	"09 :ZN";				"14 :0" -> "13 :C2";			
0.000	0.000	08	:ZN	"08 :ZN";				"14 :0" -> "09 :A";			
1.625	705.291	10	:A	"10 :A";				"14 :D" -> "14 :D";			
4.000	0.000	Y	:	"Y";				"14 :CK" -> "14 :D";			
0.000	0.000	B	:	"B";				"14 :D" -> "14 :CK";			
1.578	2.686	11	:B	"11 :B";				"14 :CK" -> "14 :CK";			
0.000	0.000	13	:ZN	"13 :ZN";				"14 :CK" -> "14 :QN";			
0.903	118.257	14	:CK	"14 :CK";				"14 :D" -> "14 :D";			
0.000	0.000	A	:	"A";				"14 :CK" -> "14 :QN";			
1.519	22.363	11	:C1	"11 :C1";				"B" -> "13 :A";			
1.565	1.974	12	:B2	"12 :B2";				"C" -> "13 :C1";			
0.000	0.000	12	:ZN	"12 :ZN";				"10 :ZN" -> "13 :B";			
0.000	0.000	10	:ZN	"10 :ZN";				"13 :C2" -> "13 :ZN";			
0.000	0.000	11	:ZN	"11 :ZN";				"13 :C1" -> "13 :ZN";			
1.595	44.263	12	:A	"12 :A";				"13 :B" -> "13 :ZN";			
1.529	135.790	13	:C1	"13 :C1";				"13 :A" -> "13 :ZN";			
1.591	169.890	13	:A	"13 :A";				"13 :C2" -> "13 :ZN";			
0.000	0.000	14	:QN	"14 :QN";				"13 :C1" -> "13 :ZN";			
0.000	0.000	14	:Q	"14 :Q";				"13 :B" -> "13 :ZN";			
1.526	1.929	13	:C2	"13 :C2";				"13 :A" -> "13 :ZN";			
1.101	0.000	14	:D	"14 :D";				"15 :0" -> "11 :B";			
30.024	2841.795	total		"13 :ZN" -> "15 :D";							

e.

f.

g.

```

"11 :ZN" -> "12 :A";
"A" -> "12 :B1";
"09 :ZN" -> "12 :B2";
"12 :B2" -> "12 :ZN";
"12 :B1" -> "12 :ZN";
"12 :A" -> "12 :ZN";
"12 :B2" -> "12 :ZN";
"12 :B1" -> "12 :ZN";
"12 :A" -> "12 :ZN";
"A" -> "11 :A";
"08 :ZN" -> "11 :C1";
"B" -> "11 :C2";
"11 :C2" -> "11 :ZN";
"11 :C1" -> "11 :ZN";
"11 :B" -> "11 :ZN";
"11 :A" -> "11 :ZN";
"11 :C2" -> "11 :ZN";
"11 :C1" -> "11 :ZN";
"11 :B" -> "11 :ZN";
"11 :A" -> "11 :ZN";
"A" -> "10 :A";
"10 :A" -> "10 :ZN";
"10 :A" -> "10 :ZN";
"09 :A" -> "09 :ZN";
"09 :A" -> "09 :ZN";
"C" -> "08 :A";
"08 :A" -> "08 :ZN";
"08 :A" -> "08 :ZN";
}
ngxxfus@ngxxfus-X489FA [17:19:52] [OpenTimerTest] [master +]

```

h.

Hình . (a, b, c, d, e, f, g, h,) Kết quả phân tích STA sử dụng C++API.

Nhận xét:

- Tốp 05 critical-path từ CK->Y đều vi phạm ràng buộc thời gian (set-up time).
- Công suất tiêu thụ:  $30.02 \text{ nW} + 2841.8 \text{ nW} \approx 2.87 \text{ } \mu\text{W}$

2. OpenSTA

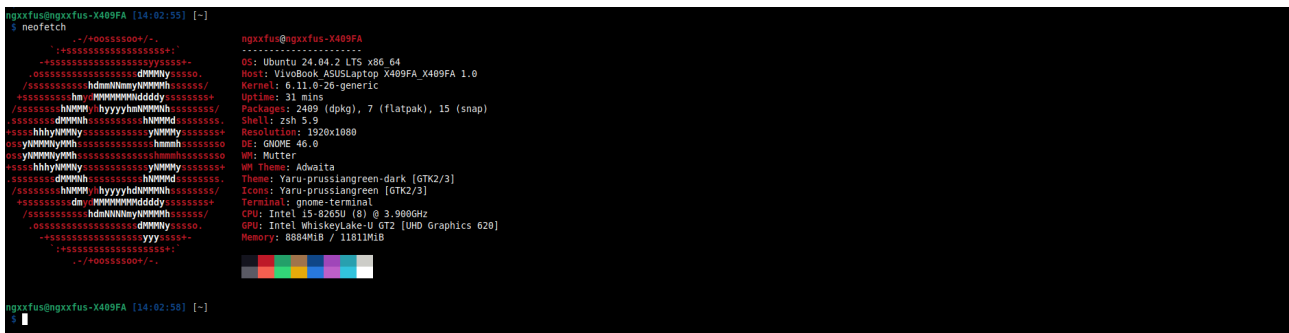
2.1. Giới thiệu về OpenSTA

OpenSTA là một công cụ mã nguồn mở thực hiện phân tích thời gian tĩnh (Static Timing Analysis - STA) ở cấp độ gate-level dành cho các thiết kế vi mạch kỹ thuật số. Được phát triển trong dự án OpenROAD và viết bằng C++, OpenSTA cung cấp khả năng phân tích thời gian chính xác và hiệu quả, hỗ trợ các tệp SDC (Synopsys Design Constraints) tiêu chuẩn để định nghĩa ràng buộc thời gian.

Với thiết kế tối ưu cho tích hợp vào các luồng thiết kế EDA, OpenSTA nổi bật nhờ tính linh hoạt, khả năng xử lý các thiết kế phức tạp và hỗ trợ tốt cho các công cụ thiết kế mã nguồn mở. So với OpenTimer, OpenSTA được sử dụng rộng rãi nhờ khả năng tương thích cao với các định dạng công nghiệp và dễ dàng tích hợp vào quy trình thiết kế tự động hóa, phù hợp cho các dự án yêu cầu độ tin cậy và tính chuẩn hóa cao.

2.2. Cài đặt OpenSTA

Môi trường cài đặt:



Hình . Thông tin môi trường cài đặt thông qua neofetch.

2.2.1. Cài đặt công cụ và các thư viện phụ thuộc

Bảng . và . mô tả các phụ thuộc và phiên bản phù hợp dựa theo tài liệu của OpenSTA (<https://github.com/The-OpenROAD-Project/OpenSTA>).

Công cụ xây dựng:

Phụ thuộc	cmake	clang	gcc	tcl	swig	bison	flex
Ubuntu	3.24.2	--	11.4.0	8.6	4.1.0	3.8.2	2.6.4

Thư viện phụ thuộc:

Phụ thuộc	eigen	cudd	tclreadline	zLib
Ubuntu	3.4.0	3.0.0	2.3.8	1.2.5

2.2.1.1. eigen

```
sudo apt install libeigen3-dev
```

2.2.1.2. tclreadline

```
sudo apt install tcl-tclreadline
```

2.2.1.3. CUDD

2.2.1.3.1. Tải tập tin nén CUDD

Sử dụng wget để tải tập tin [cudd-3.0.0.tar.gz](#) vào thư mục hiện tại.

```
wget https://raw.githubusercontent.com/davidkebo/cudd/refs/heads/main/cudd_versions/cudd-3.0.0.tar.gz

ngxxfus@ngxxfus-X489FA [22:00:03] [Downloads]
$ wget https://raw.githubusercontent.com/davidkebo/cudd/refs/heads/main/cudd_versions/cudd-3.0.0.tar.gz
--2025-05-27 22:00:09-- https://raw.githubusercontent.com/davidkebo/cudd/refs/heads/main/cudd_versions/cudd-3.0.0.tar.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.111.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1175302 (1.1M) [application/octet-stream]
Saving to: 'cudd-3.0.0.tar.gz'

cudd-3.0.0.tar.gz                               100%[=====] 1.12M --.-KB/s   in 0.1s

2025-05-27 22:00:10 (9.10 MB/s) - 'cudd-3.0.0.tar.gz' saved [1175302/1175302]

ngxxfus@ngxxfus-X489FA [22:00:10] [Downloads]
```

Hình . Quá trình tải cudd-3.0.0.tar.gz thông qua wget.

### 2.2.1.3.2. Giải nén CUDD

Sử dụng lệnh tar với các cờ -zxvf để giải nén [cudd-3.0.0.tar.gz](https://raw.githubusercontent.com/davidkebo/cudd/refs/heads/main/cudd_versions/cudd-3.0.0.tar.gz), sau đó di chuyển đến thư mục đã giải nén.

```
tar -xvf cudd-3.0.0.tar.gz
cd cudd-3.0.0

ngxxfus@ngxxfus-X489FA [22:00:10] [Downloads]
$ tar -xvf cudd-3.0.0.tar.gz
cd cudd-3.0.0

cudd-3.0.0/
cudd-3.0.0/README
cudd-3.0.0/doc/
cudd-3.0.0/doc/phase.pdf
cudd-3.0.0/doc/Included.am
cudd-3.0.0/doc/cudd.tex.in
cudd-3.0.0/obj/
cudd-3.0.0/obj/st.c
cudd-3.0.0/obj/testst.c
cudd-3.0.0/obj/Included.am
cudd-3.0.0/obj/test-st-test.in
cudd-3.0.0/obj/testst.c
```

Hình . Quá trình giải nén cudd-3.0.0.tar.gz.

### 2.2.1.3.3. Xây dựng các thiết lập

Các thiết lập các thiết lập phù hợp để xây dựng cudd-3.0.0 thông qua ./configure

```
./configure

ngxxfus@ngxxfus-X489FA [22:01:33] [cudd-3.0.0]
$ ./configure
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install: setting permissions for '/home/ngxxfus/Downloads/cudd-3.0.0/conftest.dir/conftest.one': Operation not permitted
/usr/bin/install: setting permissions for '/home/ngxxfus/Downloads/cudd-3.0.0/conftest.dir/conftest.two': Operation not permitted
/bin/install: setting permissions for '/home/ngxxfus/Downloads/cudd-3.0.0/conftest.dir/conftest.one': Operation not permitted
/bin/install: setting permissions for '/home/ngxxfus/Downloads/cudd-3.0.0/conftest.dir/conftest.two': Operation not permitted
build-aux/install-sh -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for gcc... gcc
checking whether the C compiler works... yes

a.

config.status: creating dddmp/obj/test7.sh
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
.....
Configuration summary for cudd 3.0.0
Build system : x86_64-unknown-linux-gnu
Host system : x86_64-unknown-linux-gnu
Prefix : '/usr/local'
Compilers : 'gcc -Wall -Wextra -g -O3'
           'g++ -Wall -Wextra -std=c++0x -g -O3'
Shared library : no
dddmp enabled : no
obj enabled : no
.....
ngxxfus@ngxxfus-X489FA [22:01:42] [cudd-3.0.0]
$
```

b.

Hình . Quá trình thiết lập môi trường để xây dựng cudd-3.0.0.

### 2.2.1.3.4. Xây dựng CUDD

Xây dựng CUDD thông qua make (với cờ -j8). Sau khi quá trình xây dựng (build) CUDD hoàn tất, lưu lại đường dẫn đến thư mục cudd. Trong trường hợp này đường dẫn đến cudd là: /home/ngxxfus/cudd-3.0.0

```
make -j8

ngxxfus@ngxxfus-X489FA [22:01:42] [cudd-3.0.0]
$ make -j8
make all-am
make[1]: Entering directory '/mnt/sda1/Linux Downloads/cudd-3.0.0'
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddAbs.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddAbs.Tpo -c -o cudd/cudd_libcudd_la-cuddAddAbs.lo `test -f 'cudd/cuddAddAbs.c' || echo './'`cudd/cuddAddAbs.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddApply.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddApply.Tpo -c -o cudd/cudd_libcudd_la-cuddAddApply.lo `test -f 'cudd/cuddAddApply.c' || echo './'`cudd/cuddAddApply.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddFind.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddFind.Tpo -c -o cudd/cudd_libcudd_la-cuddAddFind.lo `test -f 'cudd/cuddAddFind.c' || echo './'`cudd/cuddAddFind.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddInv.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddInv.Tpo -c -o cudd/cudd_libcudd_la-cuddAddInv.lo `test -f 'cudd/cuddAddInv.c' || echo './'`cudd/cuddAddInv.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddIte.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddIte.Tpo -c -o cudd/cudd_libcudd_la-cuddAddIte.lo `test -f 'cudd/cuddAddIte.c' || echo './'`cudd/cuddAddIte.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddNeg.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddNeg.Tpo -c -o cudd/cudd_libcudd_la-cuddAddNeg.lo `test -f 'cudd/cuddAddNeg.c' || echo './'`cudd/cuddAddNeg.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddWalsh.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddWalsh.Tpo -c -o cudd/cudd_libcudd_la-cuddAddWalsh.lo `test -f 'cudd/cuddAddWalsh.c' || echo './'`cudd/cuddAddWalsh.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAbs.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAbs.Tpo -c -o cudd/cudd_libcudd_la-cuddAbs.lo `test -f 'cudd/cuddAbs.c' || echo './'`cudd/cuddAbs.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddAbs.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddAbs.Tpo -c -o cudd/cudd_libcudd_la-cuddAddAbs.lo `test -f 'cudd/cuddAddAbs.c' || echo './'`cudd/cuddAddAbs.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddApply.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddApply.Tpo -c -o cudd/cudd_libcudd_la-cuddAddApply.lo `test -f 'cudd/cuddAddApply.c' || echo './'`cudd/cuddAddApply.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddFind.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddFind.Tpo -c -o cudd/cudd_libcudd_la-cuddAddFind.lo `test -f 'cudd/cuddAddFind.c' || echo './'`cudd/cuddAddFind.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddInv.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddInv.Tpo -c -o cudd/cudd_libcudd_la-cuddAddInv.lo `test -f 'cudd/cuddAddInv.c' || echo './'`cudd/cuddAddInv.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddIte.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddIte.Tpo -c -o cudd/cudd_libcudd_la-cuddAddIte.lo `test -f 'cudd/cuddAddIte.c' || echo './'`cudd/cuddAddIte.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddNeg.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddNeg.Tpo -c -o cudd/cudd_libcudd_la-cuddAddNeg.lo `test -f 'cudd/cuddAddNeg.c' || echo './'`cudd/cuddAddNeg.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAddWalsh.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAddWalsh.Tpo -c -o cudd/cudd_libcudd_la-cuddAddWalsh.lo `test -f 'cudd/cuddAddWalsh.c' || echo './'`cudd/cuddAddWalsh.c
/bin/bash ./libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I/cudd -I./st -I./epd -I./mtr -I./util -Wall -Wextra -g -O3 -MT cudd/cudd_libcudd_la-cuddAbs.lo -MD -MP -MF cudd/.deps/cudd_libcudd_la-cuddAbs.Tpo -c -o cudd/cudd_libcudd_la-cuddAbs.lo `test -f 'cudd/cuddAbs.c' || echo './'`cudd/cuddAbs.c
make[1]: Leaving directory '/mnt/sda1/Linux Downloads/cudd-3.0.0'
```

Hình . Quá trình build cudd-3.0.0.

### 2.2.2. Tải OpenSTA từ kho GitHub

Sử dụng git clone để tải toàn bộ mã nguồn của OpenSTA về ~ (/home/ngxxfus/). Lệnh bên dưới giúp tải OpenSTA từ kho GitHub về thư mục hiện tại.

git clone --recursive https://github.com/parallaxsw/OpenSTA.git  
cd OpenSTA

```
ngxxfus@ngxxfus-X489FA [22:28:06] [-]  
$ git clone --recursive https://github.com/parallaxsw/OpenSTA.git  
Cloning into 'OpenSTA'...  
remote: Enumerating objects: 17644, done.  
remote: Counting objects: 100% (2849/2849), done.  
remote: Compressing objects: 100% (277/277), done.  
remote: Total 17644 (delta 2663), reused 2587 (delta 2568), pack-reused 14795 (from 2)  
Receiving objects: 100% (17644/17644), 48.66 MiB | 16.20 MiB/s, done.  
Resolving deltas: 100% (14291/14291), done.  
ngxxfus@ngxxfus-X489FA [22:29:13] [-]  
$ cd OpenSTA  
ngxxfus@ngxxfus-X489FA [22:29:17] [OpenSTA] [master]
```

Hình . Quá trình tải OpenSTA.

### 2.2.3. Tạo Makefile để biên dịch OpenSTA

Sử dụng cmake để tạo Makefile cho bước xây dựng và cài đặt OpenSTA. Bên trong thư mục OpenSTA, tạo thư mục build/ - là nơi chứa Makefile sẽ được sinh ra bởi cmake. Sau đó gọi cmake.

cmake -DCUDD\_DIR=/home/ngxxfus/Downloads/cudd-3.0.0 ..

```
ngxxfus@ngxxfus-X489FA [22:29:17] [OpenSTA] [master]  
$ mkdir -p ./build/  
ngxxfus@ngxxfus-X489FA [22:32:04] [OpenSTA] [master]  
$ cd build  
ngxxfus@ngxxfus-X489FA [22:32:05] [build] [master]  
$ cmake -DCUDD_DIR=/home/ngxxfus/Downloads/cudd-3.0.0 ..  
-- The CXX compiler identification is GNU 13.3.0  
-- Detecting CXX compiler ABI info  
-- Detecting CXX compiler ABI info - done  
-- Check for working CXX compiler: /usr/bin/c++ - skipped  
-- Detecting CXX compile features  
-- Detecting CXX compile features - done  
-- STA version: 2.7.0  
-- STA git sha: b2bc9951600c1a6e27fd53eed03899126634216  
-- System name: Linux  
-- Compiler: GNU 13.3.0  
-- Build type: RELEASE  
-- Build CXX_FLAGS: -O3 -DDEBUG  
-- Install prefix: /usr/local  
-- Found FLEX: /usr/bin/flex (found version "2.6.4")  
-- Found BISON: /usr/bin/bison (found version "3.8.2")  
-- TCL library: /usr/lib/x86_64-linux-gnu/libtcl.so  
-- TCL header: /usr/include/tcl/tcl.h  
-- TCL readline library: /usr/lib/x86_64-linux-gnu/libtclreadline.so  
-- TCL readline header: /usr/include/x86_64-linux-gnu/tclreadline.h  
-- Found ZLIB: /usr/lib/x86_64-linux-gnu/libz.so (found version "1.3")  
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Success  
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Success  
-- Found Threads: TRUE  
-- CUDD library: /home/ngxxfus/Downloads/cudd-3.0.0/cudd/.libs/libcudd.a  
-- CUDD header: /home/ngxxfus/Downloads/cudd-3.0.0/cudd/cudd.h  
-- SSTA: 0  
-- Found SWIG: /usr/bin/swig4.0 (found suitable version "4.2.0", minimum required is "3.8")  
-- STA library: /home/ngxxfus/OpenSTA/build/libOpenSTA.a  
-- STA executable: /home/ngxxfus/OpenSTA/build/sta  
-- Configuring done (0.3s)  
-- Generating done (0.0s)  
-- Build files have been written to: /home/ngxxfus/OpenSTA/build  
ngxxfus@ngxxfus-X489FA [22:32:27] [build] [master]
```

Hình . Quá trình sinh Makefile.

### 2.2.4. Biên dịch tập tin thực thi và thư viện OpenSTA

Sau khi sinh Makefile, sử dụng lệnh make để biên dịch tập tin thực thi và thư viện của OpenSTA.

make -j8

```
ngxxfus@ngxxfus-X489FA [22:32:27] [build] [master]  
$ make -j8  
[ 2%] [BISON][LibertyParse] Building parser with bison 3.8.2  
[ 2%] [BISON][VerilogParse] Building parser with bison 3.8.2  
[ 2%] [FLEX][LibertyLex] Building scanner with flex 2.6.4  
[ 2%] [BISON][LibExprParse] Building parser with bison 3.8.2  
[ 2%] [FLEX][LibExprLex] Building scanner with flex 2.6.4  
[ 3%] [BISON][SaiParse] Building parser with bison 3.8.2  
[ 3%] [FLEX][SaiLex] Building scanner with flex 2.6.4  
[ 4%] Swig compile app/StaApp.i for tcl  
[ 5%] [FLEX][SdfLex] Building scanner with flex 2.6.4  
[ 6%] [BISON][SdfParse] Building parser with bison 3.8.2  
[ 6%] [FLEX][SdfLex] Building scanner with flex 2.6.4  
[ 7%] [BISON][SdfParse] Building parser with bison 3.8.2  
[ 7%] Generating StaTclUnitVar.cc  
[ 8%] [FLEX][VerilogLex] Building scanner with flex 2.6.4  
[ 9%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/ArnoldDelayCalc.cc.o  
[ 9%] Building CXX object CMakeFiles/OpenSTA.dir/app/StaMain.cc.o  
[ 9%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/ArnoldWaveforms.cc.o  
[10%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/ArnoldReduce.cc.o  
[11%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/ArnoldDelayCalc.cc.o  
[12%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/ArnoldAnalysisPT.cc.o  
[12%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/CcseffDelayCalc.cc.o  
[12%] Built target sta_swig_swig_compilation  
[12%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/DelayCalc.cc.o  
[13%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/DelayCalcBase.cc.o  
[13%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/DmpCeff.cc.o  
[14%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/DmpDelayCalc.cc.o  
[14%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/FanInRoot.cc.o  
[15%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/GraphDelayCalc.cc.o  
[16%] Building CXX object CMakeFiles/OpenSTA.dir/dcalc/LumpedCapDelayCalc.cc.o
```

a.

```
[ 96%] Building CXX object CMakeFiles/OpenSTA.dir/VerilogLex.cc.o  
[ 96%] Building CXX object CMakeFiles/OpenSTA.dir/VerilogParse.cc.o  
[ 97%] Building CXX object CMakeFiles/OpenSTA.dir/SaiLex.cc.o  
[ 97%] Building CXX object CMakeFiles/OpenSTA.dir/SaiParse.cc.o  
[ 98%] Linking CXX static library libOpenSTA.a  
[ 98%] Built target OpenSTA  
[ 98%] Building CXX object CMakeFiles/sta_swig.dir/StaAppTclWrap.cc.o  
[ 99%] Linking CXX static library sta_swig.a  
[ 99%] Built target sta_swig  
[ 99%] Building CXX object CMakeFiles/sta.dir/app/Main.cc.o  
[100%] Linking CXX executable sta  
[100%] Built target sta
```

b.

Hình . (a, b) Quá trình biên dịch tập tin thực thi và thư viện của OpenSTA.



### 2.2.5. Cài đặt OpenSTA vào hệ thống

Sau khi biên dịch tập tin thực thi và thư viện của OpenSTA, tiếp sử dụng make install để cài đặt OpenSTA vào hệ thống.

```
sudo make -j8 install
```

```
ngxfus@ngxfus-X489FA [22:35:45] [build] [master]
$ sudo make -j8 install
[sudo] password for ngxfus:
[ 1%] Built target sta_swig_compilation
[ 98%] Built target OpenSTA
[ 99%] Built target sta_swig
[100%] Built target sta
Install the project...
-- Install configuration: "RELEASE"
-- Installing: /usr/local/bin/sta
-- Installing: /usr/local/lib/libOpenSTA.a
-- Installing: /usr/local/include/sta
```

a.

```
-- Installing: /usr/local/include/sta/ParseBus.hh
-- Installing: /usr/local/include/sta/VertexVisitor.hh
-- Installing: /usr/local/include/sta/ClockInsertion.hh
-- Installing: /usr/local/include/sta/VerilogReader.hh
-- Installing: /usr/local/include/sta/SCCcdComment.hh
-- Installing: /usr/local/include/sta/InternalPower.hh
-- Installing: /usr/local/include/sta/VertexId.hh
-- Installing: /usr/local/include/sta/Debug.hh
-- Installing: /usr/local/include/sta/ObjectId.hh
-- Installing: /usr/local/include/sta/PathAnalysisPt.hh
Up-to-date: /usr/local/include/sta
-- Installing: /usr/local/include/sta/StaConfig.hh
```

b.

Hình . (a, b) Quá trình cài đặt OpenSTA vào hệ thống.

### 2.3. Chuẩn bị cho STA

Mạch tổ hợp thực hiện phép nhân 8-bit được tái sử dụng với hai module được mô tả bên dưới. Hai thiết kế này đề sẽ được tổng hợp bởi YoSys với cùng thư viện cell tiêu chuẩn và điều được đánh giá STA. Tất cả quá trình đều sử dụng chung các script TCL, chỉ đổi mỗi tên mô-đun/top-mô-đun.

Mạch tổ hợp:

```
module mul_2(EN, A, B, S);
    input EN;
    input [7:0] A, B;
    output [15:0] S;

    wire [15:0] _S;
    SCDM8_73 scdm8_73(.a(A), .b(B), .S(_S));

    assign S = _S;
endmodule
```

Mạch tuần tự với tính hiệu RST:

```
module mul_n(CLK, RST, A, B, S);
    input EN;
    input [7:0] A, B;
    output reg [15:0] S;

    wire [15:0] _S;
    SCDM8_73 scdm8_73(.a(A), .b(B), .S(_S));

    always @(posedge CLK) begin
        if( RST ) begin
            S <= _S;
        end else begin
            S <= 0;
        end
    end
endmodule
```

Tương ứng với hai mô-đun, hai đoạn mã bên dưới mô tả SDC cho hai mô-đun.

Mạch tổ hợp:

```
set delay 1.0
set_input_delay $delay [get_ports {A[*] B[*]}]
set_output_delay $delay [get_ports {S[*]}]
set_input_transition 0.1 [get_ports {A[*] B[*]}]
```

Mạch tuần tự với tín hiệu RST:

```
set period 5
create_clock -period $period [get_ports CLK]
set clk_period_factor 0.2
set delay [expr $period * $clk_period_factor]
set_input_delay $delay -clock CLK [get_ports {A[*] B[*] RST}]
set_output_delay $delay -clock CLK [all_outputs]
set_input_transition 0.1 [all_inputs]
set_load 0.05 [all_outputs]
```

## 2.4. Phân tích STA cho mạch tổ hợp

### 2.4.1. Tổng hợp mul\_2 bằng YoSys

Đoạn mã TCL bên dưới thực hiện việc tổng hợp và ánh xạ đến thư viện cell tiêu chuẩn thông qua YoSys.

```
# 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`

set working_dir
"/mnt/sda1/DOC/HCMUTE_DOCs/NH_2024_2025_HK_02/DA1/OpenTimerTest/"
set output_dir "/mnt/sda1/DOC/HCMUTE_DOCs/NH_2024_2025_HK_02/DA1/OpenTimerTest/"
set lib_path
"/mnt/sda1/DOC/HCMUTE_DOCs/NH_2024_2025_HK_02/DA1/OpenTimerTest/NangateOpenCellL
ibrary_typical.lib"

set synth_output "mul_2_synth.v"
set mapped_output "mul_2_mapped.v"
set top_module "mul_2"
set verilog_files {
    SCDM_73/CarryLookAhead4bitsSumComponent.v
    SCDM_73/CarryLookaheadAdder12bits.v
    SCDM_73/CarryLookaheadAdder4bits.v
    SCDM_73/CarryLookaheadLogicCircuit.v
    SCDM_73/CarryLookaheadPandGcreate.v
    SCDM_73/FirstRow.v
    SCDM_73/FullAdder1bit.v
    SCDM_73/GA7.v
    SCDM_73/GB5.v
    SCDM_73/HalfAdder1bit.v
    SCDM_73/LastRowOfGroupA.v
```

```

SCDM_73/PartialProductUnit1_N.v
SCDM_73/Row1AndRow2OfGroupB.v
SCDM_73/LastRowOfGroupB.v
SCDM_73/PartialProductUnit0.v
SCDM_73/PartialProductUnit0_N.v
SCDM_73/PartialProductUnit1.v
SCDM_73/PartialProductUnit2.v
SCDM_73/PartialProductUnit2_N.v
SCDM_73/PartialProductUnit3.v
SCDM_73/Row1AndRow2OfGroupA.v
SCDM_73/SCDM8_75.v
mul_2_top.v
}
foreach file $verilog_files {
    yosys read_verilog $working_dir$file
}
yosys hierarchy -check -top $top_module
yosys synth -top $top_module -flatten
yosys write_verilog -noattr $output_dir$synth_output
yosys read_liberty -lib $lib_path
yosys opt -purge
yosys dfflibmap -liberty $lib_path
yosys abc -liberty $lib_path
yosys techmap -map +/techmap.v
yosys check
yosys opt_clean
yosys opt -purge
yosys clean
yosys write_verilog -noattr $output_dir$mapped_output

```

#### 2.4.2. Đánh giá độ trễ lan truyền

Mô-đun mul\_n sau khi được tổng hợp và ánh xạ đến cell tiêu chuẩn thì được đánh giá STA bởi OpenSTA. Đoạn mã TCL thực hiện việc nhập như viện (best-case, typical, worst-case), nhập thiết kế Verilog, ràng buộc (SDC), chỉ định top-module và cuối cùng là đánh giá STA. Để dễ dàng hơn trong việc đọc, các giá trị màu sắc được thêm vào mã TCL; Tùy vào Shell/Terminal mà sẽ có xuất hiện màu sắc hay không. Cờ **-unconstrained** được thêm vào các lệnh report (báo cáo) để yêu cầu OpenSTA hãy đưa vào báo cáo cả những đường (path) không bị ràng buộc bởi bất kỳ ràng buộc thời gian nào (tức là unconstrained paths).

```

set RED      "\033\[91> "
set GREEN    "\033\[92m"
set YELLOW   "\033\[1m\033\[93m> "
set BLUE     "\033\[94m> "
set RESET    "\033\[0m"

set LIB_WC NangateOpenCellLibrary_worst_low.lib
set LIB_TYP NangateOpenCellLibrary_typical.lib
set LIB_BC NangateOpenCellLibrary_fast.lib
set VERILOG_FILE mul_2_mapped.v
set SDC_FILE mul_2.sdc
set TOP_MODULE mul_2
puts "${YELLOW}define_corners wc typ bc"
define_corners wc typ bc

```

```
puts "${YELLOW}read_liberty -corner wc      ${LIB_WC}${RESET}"
read_liberty -corner wc      ${LIB_WC}
puts "${YELLOW}read_liberty -corner typ    ${LIB_TYP}${RESET}"
read_liberty -corner typ    ${LIB_TYP}
puts "${YELLOW}read_liberty -corner bc     ${LIB_BC}${RESET}"
read_liberty -corner bc     ${LIB_BC}
puts "${YELLOW}read_verilog                ${VERILOG_FILE}${RESET}"
read_verilog                ${VERILOG_FILE}
puts "${YELLOW}link_design                 ${TOP_MODULE}${RESET}"
link_design                 ${TOP_MODULE}
puts "${YELLOW}read_sdc                   ${SDC_FILE}${RESET}"
read_sdc                   ${SDC_FILE}
puts "${YELLOW}report_checks -path_delay min_max -unconstrained${RESET}"
report_checks -path_delay min_max -unconstrained
puts "${YELLOW}report_checks -corner typ -unconstrained${RESET}"
report_checks -corner typ -unconstrained
puts "${YELLOW}report_checks -corner wc -from B[3] -to S[9] -unconstrained${RESET}"
report_checks -corner wc -from B[3] -to S[9] -unconstrained
```

Hình bên dưới mô tả quá trình nhập thư viện (best-case, typical, worst-case), nhập thiết kế Verilog, ràng buộc (SDC), chỉ định top-module.

```
ngxxfus@ngxxfus-X409FA [02:43:56] [OpenTimeTest] [master *]
$ sta ./mul_2_OpenSTA_MinMaxDelay.tcl
OpenSTA 2.7.0 b2bc995160 Copyright (c) 2025, Parallax Software, Inc.
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>

This is free software, and you are free to change and redistribute it
under certain conditions; type 'show copying' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show warranty'.
> define_corners wc typ bc
> read_liberty -corner wc      NangateOpenCellLibrary_worst_low.lib
> read_liberty -corner typ    NangateOpenCellLibrary_typical.lib
Warning: NangateOpenCellLibrary_typical.lib line 37, library NangateOpenCellLibrary already exists.
> read_liberty -corner bc     NangateOpenCellLibrary_fast.lib
Warning: NangateOpenCellLibrary_fast.lib line 37, library NangateOpenCellLibrary already exists.
> read_verilog                mul_2_mapped.v
> link_design                 mul_2
> read_sdc                   mul_2.sdc
```

Hình . Quá trình đọc thư viện và thiết kế.

### 2.4.3. Kết quả đánh giá độ trễ lan truyền

```
> report_checks -path_delay min_max -unconstrained
```

```
Startpoint: B[4] (input port)
Endpoint: S[4] (output port)
Path Group: unconstrained
Path Type: min
Corner: bc
```

Delay	Time	Description
-----		
1.00	1.00	^ input external delay
0.00	1.00	^ B[4] (in)
0.01	1.01	v _251_/ZN (AOI21_X1)
0.01	1.02	^ _252_/ZN (NOR2_X1)
0.00	1.02	^ S[4] (out)
	1.02	data arrival time

(Path is unconstrained)

```
Startpoint: B[5] (input port)
Endpoint: S[14] (output port)
Path Group: unconstrained
Path Type: max
Corner: typ
```

a.

Delay	Time	Description
-----		
1.00	1.00	v input external delay
0.00	1.00	v B[5] (in)
0.08	1.08	v _352_/ZN (AND4_X1)
0.05	1.13	^ _355_/ZN (OAI22_X1)
0.03	1.16	v _358_/ZN (AOI22_X1)
0.09	1.26	v _359_/ZN (OR3_X1)
0.04	1.29	v _361_/ZN (AND3_X1)
0.03	1.32	^ _364_/ZN (NOR2_X1)
0.06	1.38	^ _365_/Z (XOR2_X1)
0.06	1.44	^ _378_/Z (XOR2_X1)
0.02	1.46	v _380_/ZN (AOI21_X1)
0.05	1.51	^ _408_/ZN (OAI21_X1)
0.03	1.54	v _430_/ZN (AOI21_X1)
0.04	1.58	^ _227_/ZN (NOR3_X1)
0.04	1.62	^ _234_/ZN (XNOR2_X1)
0.00	1.62	^ S[14] (out)
	1.62	data arrival time

(Path is unconstrained)

b.

```
> report_checks -corner typ -unconstrained
Startpoint: B[5] (input port)
Endpoint: S[14] (output port)
Path Group: unconstrained
Path Type: max
Corner: typ
```

Delay	Time	Description
1.00	1.00	v input external delay
0.00	1.00	v B[5] (in)
0.08	1.08	v _352_/ZN (AND4_X1)
0.05	1.13	^ _355_/ZN (OAI22_X1)
0.03	1.16	v _358_/ZN (AOI22_X1)
0.09	1.26	v _359_/ZN (OR3_X1)
0.04	1.29	v _361_/ZN (AND3_X1)
0.03	1.32	^ _364_/ZN (NOR2_X1)
0.06	1.38	^ _365_/Z (XOR2_X1)
0.06	1.44	^ _378_/Z (XOR2_X1)
0.02	1.46	v _380_/ZN (AOI21_X1)
0.05	1.51	^ _408_/ZN (OAI21_X1)
0.03	1.54	v _430_/ZN (AOI21_X1)
0.04	1.58	^ _227_/ZN (NOR3_X1)
0.04	1.62	^ _234_/ZN (XNOR2_X1)
0.00	1.62	^ S[14] (out)
	1.62	data arrival time

(Path is unconstrained)

c.

```
> report_checks -corner wc -from B[3] -to S[9] -unconstrained
Startpoint: B[3] (input port)
Endpoint: S[9] (output port)
Path Group: unconstrained
Path Type: max
Corner: wc
```

Delay	Time	Description
1.00	1.00	v input external delay
0.00	1.00	v B[3] (in)
0.05	1.05	^ _265_/ZN (NAND2_X1)
0.05	1.10	^ _267_/Z (XOR2_X1)
0.04	1.14	^ _271_/ZN (XNOR2_X1)
0.05	1.19	^ _273_/Z (XOR2_X1)
0.02	1.22	v _275_/ZN (AOI21_X1)
0.04	1.25	^ _292_/ZN (OAI21_X1)
0.03	1.28	v _316_/ZN (AOI21_X1)
0.04	1.31	^ _347_/ZN (OAI21_X1)
0.04	1.35	^ _379_/Z (XOR2_X1)
0.00	1.35	^ S[9] (out)
	1.35	data arrival time

(Path is unconstrained)

d.

Hình . (a, b, c, d) Kết quả phân tích STA về độ trễ.

## 2.4.4. Đánh giá công suất

Khác với mạch tuần tự, mạch tổ hợp không có tín hiệu CLK vì vậy không thể dùng **-activity** để thiết lập tuần suất chuyển mạch, thay vào đó dùng **-density** để thiết lập mật độ.

```
set RED      "\033\[91> "
set GREEN    "\033\[92m"
set YELLOW   "\033\[1m\033\[93m> "
set BLUE     "\033\[94m> "
set RESET    "\033\[0m"

set LIB_WC NangateOpenCellLibrary_worst_low.lib
set LIB_TYP NangateOpenCellLibrary_typical.lib
set LIB_BC NangateOpenCellLibrary_fast.lib
set VERILOG_FILE mul_2_mapped.v
set SDC_FILE mul_2.sdc
set TOP_MODULE mul_2

puts "${YELLOW}read_liberty ${LIB_TYP}${RESET}"
read_liberty ${LIB_TYP}
puts "${YELLOW}read_verilog ${VERILOG_FILE}${RESET}"
read_verilog ${VERILOG_FILE}
puts "${YELLOW}link_design ${TOP_MODULE}${RESET}"
link_design ${TOP_MODULE}
puts "${YELLOW}read_sdc ${SDC_FILE}${RESET}"
read_sdc ${SDC_FILE}
puts "${YELLOW}set_power_activity -input -density 0.5${RESET}"
set_power_activity -input -density 0.5
puts "${YELLOW}report_power${RESET}"
report_power
```

## 2.4.5. Kết quả đánh giá công suất

```
ngxxfus@ngxxfus-X409FA [03:14:12] [OpenTimerTest] [master *]
$ sta ./mul_2 OpenSTA Power.tcl
OpenSTA 2.7.0 b2bc995160 Copyright (c) 2025, Parallax Software, Inc.
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>

This is free software, and you are free to change and redistribute it
under certain conditions; type 'show copying' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show_warranty'.
> read_liberty NangateOpenCellLibrary_typical.lib
> read_verilog mul_2_mapped.v
> link_design mul_2
> read_sdc mul_2.sdc
> set_power_activity -input -density 0.5
> report_power
Group              Internal Power  Switching Power  Leakage Power  Total Power (Watts)
-----
Sequential          0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.0%
Combinational       1.14e-03    7.04e-04    6.27e-06    1.85e-03   100.0%
Clock               0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.0%
Macro               0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.0%
Pad                 0.00e+00    0.00e+00    0.00e+00    0.00e+00    0.0%
-----
Total               1.14e-03    7.04e-04    6.27e-06    1.85e-03   100.0%
                    61.6%    38.1%    0.3%
```

Hình . Kết quả phân tích công suất của OpenSTA.

## 2.5. Phân tích STA cho mạch tuần tự

### 2.5.1. Tổng hợp mul\_n bằng YoSys

```
Giống 2.4.1; Chỉ đổi mul_2 thành mul_n.
[...]
set mapped_output "mul_2_mapped.v"
set top_module "mul_n"
set verilog_files {
    SCDM_73/CarryLookAhead4bitsSumComponent.v
}
[...]
```

### 2.5.2. Đánh giá độ trễ lan truyền

```
set RED      "\033\[91> "
set GREEN    "\033\[92m"
set YELLOW   "\033\[1m\033\[93m> "
set BLUE     "\033\[94m> "
set RESET    "\033\[0m"

set LIB_WC NangateOpenCellLibrary_worst_low.lib
set LIB_TYP NangateOpenCellLibrary_typical.lib
set LIB_BC NangateOpenCellLibrary_fast.lib
set VERILOG_FILE mul_2_mapped.v
set SDC_FILE mul_2.sdc
set TOP_MODULE mul_2
puts "${YELLOW}define_corners wc typ bc"
define_corners wc typ bc
puts "${YELLOW}read_liberty -corner wc      ${LIB_WC}${RESET}"
read_liberty -corner wc      ${LIB_WC}
puts "${YELLOW}read_liberty -corner typ      ${LIB_TYP}${RESET}"
read_liberty -corner typ      ${LIB_TYP}
puts "${YELLOW}read_liberty -corner bc      ${LIB_BC}${RESET}"
read_liberty -corner bc      ${LIB_BC}
puts "${YELLOW}read_verilog                    ${VERILOG_FILE}${RESET}"
read_verilog                    ${VERILOG_FILE}
puts "${YELLOW}link_design                      ${TOP_MODULE}${RESET}"
```

```

link_design                                ${TOP_MODULE}
puts "${YELLOW}read_sdc                    ${SDC_FILE} ${RESET}"
read_sdc                                    ${SDC_FILE}
puts "${YELLOW}report_checks -path_delay min_max${RESET}"
report_checks -path_delay min_max
puts "${YELLOW}report_checks -corner typ${RESET}"
report_checks -corner typ
puts "${YELLOW}report_checks -corner wc -from B[3] -to S[9]${RESET}"
report_checks -corner wc -from B[3] -to S[9]

```

### 2.5.3. Kết quả đánh giá độ trễ lan truyền

```

ngxxfus@ngxxfus-X409FA [07:53:04] [OpenTimerTest] [master *]
$ sta ./mul_2_OpenSTA_MinMaxDelay.tcl
OpenSTA 2.7.0 b2bc995160 Copyright (c) 2025, Parallax Software, Inc.
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>

This is free software, and you are free to change and redistribute it
under certain conditions; type 'show copying' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show warranty'.
> define_corners wc typ bc
> read_liberty -corner wc NangateOpenCellLibrary_worst_low.lib
> read_liberty -corner typ NangateOpenCellLibrary_typical.lib
Warning: NangateOpenCellLibrary_typical.lib line 37, library NangateOpenCellLibrary
> read_liberty -corner bc NangateOpenCellLibrary_fast.lib
Warning: NangateOpenCellLibrary_fast.lib line 37, library NangateOpenCellLibrary
> read_verilog mul_2_mapped.v
> link_design mul_n
> read_sdc mul_2.sdc
> report_checks -path_delay min_max
Startpoint: RST (input port clocked by CLK)
Endpoint: 339 (rising edge-triggered flip-flop clocked by CLK)
Path Group: CLK
Path Type: min
Corner: bc

```

Delay	Time	Description
0.00	0.00	clock CLK (rise edge)
0.00	0.00	clock network delay (ideal)
1.00	1.00	input external delay
0.00	1.00	^ RST (in)
0.00	1.00	v 335 /ZN (INV X1)
0.00	1.00	v 339 /D (DFF X1)
1.00	1.00	data arrival time
0.00	0.00	clock CLK (rise edge)
0.00	0.00	clock network delay (ideal)
0.00	0.00	clock reconvergence pessimism
0.00	0.00	^ 339 /CK (DFF X1)
0.00	0.00	library hold time
0.00	0.00	data required time
0.00	0.00	data required time
-1.00	0.00	data arrival time
0.99	0.00	slack (MET)

```

Startpoint: A[3] (input port clocked by CLK)
Endpoint: 350 (rising edge-triggered flip-flop clocked by CLK)
Path Group: CLK
Path Type: max
Corner: typ

```

Delay	Time	Description
0.00	0.00	clock CLK (rise edge)
0.00	0.00	clock network delay (ideal)
1.00	1.00	v input external delay
0.00	1.00	v A[3] (in)
0.08	1.08	v 214 /ZN (AND3 X1)
0.08	1.16	v 218 /ZN (OR3 X1)
0.04	1.19	v 220 /ZN (AND3 X1)
0.07	1.27	v 223 /ZN (OR3 X1)
0.03	1.29	^ 225 /ZN (NAND3 X1)
0.02	1.32	v 251 /ZN (AOI21 X1)
0.03	1.35	^ 269 /ZN (NOR2 X1)
0.06	1.41	^ 278 /Z (XOR2 X1)
0.07	1.48	^ 286 /Z (XOR2 X1)
0.03	1.50	v 290 /ZN (AOI21 X1)
0.06	1.56	^ 306 /ZN (AOI21 X1)
0.03	1.59	v 327 /ZN (NAND3 X1)
0.03	1.62	^ 328 /ZN (NAND2 X1)
0.02	1.64	v 334 /ZN (AOI22 X1)
0.00	1.64	v 350 /D (DFF X1)
1.64	1.64	data arrival time
5.00	5.00	clock CLK (rise edge)
0.00	5.00	clock network delay (ideal)
0.00	5.00	clock reconvergence pessimism
5.00	5.00	^ 350 /CK (DFF X1)
-0.04	4.96	library setup time
4.96	4.96	data required time
4.96	4.96	data required time
-1.64	4.96	data arrival time
3.31	3.31	slack (MET)

a. Kiểm tra hold-time.

b. Kiểm tra setup-time.

Nhận xét:

- Slack dương, nên không vi phạm hold, và có dư 0.99ns an toàn.
- Slack rất lớn (3.41 ns) → dư thời gian truyền

```
> report_checks -corner typ
Startpoint: A[3] (input port clocked by CLK)
Endpoint: _350 (rising edge-triggered flip-flop clocked by CLK)
Path Group: CLK
Path Type: max
Corner: typ
```

Delay	Time	Description
0.00	0.00	clock CLK (rise edge)
0.00	0.00	clock network delay (ideal)
1.00	1.00	v input external delay
0.00	1.00	^ A[3] (in)
0.08	1.08	v _214_/ZN (AND3_X1)
0.08	1.16	v _218_/ZN (OR3_X1)
0.04	1.19	v _220_/ZN (AND3_X1)
0.07	1.27	v _223_/ZN (OR3_X1)
0.03	1.29	^ _225_/ZN (NAND3_X1)
0.02	1.32	v _251_/ZN (AOI21_X1)
0.03	1.35	^ _269_/ZN (NOR2_X1)
0.06	1.41	^ _278_/Z (XOR2_X1)
0.07	1.48	^ _286_/Z (XOR2_X1)
0.03	1.50	v _290_/ZN (AOI21_X1)
0.06	1.56	^ _306_/ZN (AOI21_X1)
0.03	1.59	v _327_/ZN (NAND3_X1)
0.03	1.62	^ _328_/ZN (NAND2_X1)
0.02	1.64	v _334_/ZN (AOI22_X1)
0.00	1.64	v _350_/D (DFF_X1)
	1.64	data arrival time
5.00	5.00	clock CLK (rise edge)
0.00	5.00	clock network delay (ideal)
0.00	5.00	clock reconvergence pessimism
	5.00	^ _350_/CK (DFF_X1)
-0.04	4.96	library setup time
	4.96	data required time
	4.96	data required time
	-1.64	data arrival time
3.31		slack (MET)

a. Kiểm tra set-time trong trường hợp hoạt động bình thường.

```
> report_checks -corner wc -from RST -to _350_
Startpoint: A[0] (input port clocked by CLK)
Endpoint: _350 (rising edge-triggered flip-flop clocked by CLK)
Path Group: CLK
Path Type: max
Corner: wc
```

Delay	Time	Description
0.00	0.00	clock CLK (rise edge)
0.00	0.00	clock network delay (ideal)
1.00	1.00	^ input external delay
0.00	1.00	^ A[0] (in)
0.05	1.05	v _205_/ZN (NAND2_X1)
0.07	1.12	v _227_/Z (XOR2_X1)
0.03	1.15	^ _230_/ZN (AOI21_X1)
0.03	1.18	v _255_/ZN (AOI21_X1)
0.06	1.24	^ _258_/ZN (AOI21_X1)
0.02	1.26	v _280_/ZN (NOR2_X1)
0.04	1.30	^ _282_/ZN (XNOR2_X1)
0.05	1.35	^ _284_/ZN (XNOR2_X1)
0.06	1.40	^ _286_/Z (XOR2_X1)
0.03	1.43	v _290_/ZN (AOI21_X1)
0.05	1.48	^ _306_/ZN (AOI21_X1)
0.03	1.51	v _327_/ZN (NAND3_X1)
0.02	1.53	^ _328_/ZN (NAND2_X1)
0.02	1.55	v _334_/ZN (AOI22_X1)
0.00	1.55	v _350_/D (DFF_X1)
	1.55	data arrival time
5.00	5.00	clock CLK (rise edge)
0.00	5.00	clock network delay (ideal)
0.00	5.00	clock reconvergence pessimism
	5.00	^ _350_/CK (DFF_X1)
-0.03	4.97	library setup time
	4.97	data required time
	4.97	data required time
	-1.55	data arrival time
3.41		slack (MET)

b. Kiểm tra setup-time với đường dẫn được chỉ định.

Nhận xét:

- Cả hai đặt timing.

## 2.5.4. Đánh giá công suất

```
set RED      "\033\[91> "
set GREEN    "\033\[92m"
set YELLOW   "\033\[1m\033\[93m> "
set BLUE     "\033\[94m> "
set RESET    "\033\[0m"

set LIB_WC NangateOpenCellLibrary_worst_low.lib
set LIB_TYP NangateOpenCellLibrary_typical.lib
set LIB_BC NangateOpenCellLibrary_fast.lib
set VERILOG_FILE mul_2_mapped.v
set SDC_FILE mul_2.sdc
set TOP_MODULE mul_n

puts "${YELLOW}read_liberty ${LIB_TYP}${RESET}"
read_liberty ${LIB_TYP}
puts "${YELLOW}read_verilog ${VERILOG_FILE}${RESET}"
read_verilog ${VERILOG_FILE}
puts "${YELLOW}link_design ${TOP_MODULE}${RESET}"
link_design ${TOP_MODULE}
puts "${YELLOW}read_sdc ${SDC_FILE}${RESET}"
read_sdc ${SDC_FILE}
puts "${YELLOW}set_power_activity -input -activity 0.5${RESET}"
set_power_activity -input -activity 0.5
puts "${YELLOW}set_power_activity -input_port RST -activity 0.2${RESET}"
set_power_activity -input_port RST -activity 0.2
puts "${YELLOW}report_power${RESET}"
report_power
```



2.5.5. Kết quả đánh giá công suất

```
ngxxfus@ngxxfus-X409FA [03:46:50] [OpenTimerTest] [master *]
$ sta ./mul_2_OpenSTA_Power.tcl
OpenSTA 2.7.0 b2bc995160 Copyright (c) 2025, Parallax Software, Inc.
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>

This is free software, and you are free to change and redistribute it
under certain conditions; type `show_copying' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type `show_warranty'.
> read_liberty NangateOpenCellLibrary_typical.lib
> read_verilog mul_2_mapped.v
> link_design mul_n
> read_sdc mul_2.sdc
> set_power_activity -input -activity 0.5
> set_power_activity -input_port RST -activity 0.2
> report_power
Group              Internal Power    Switching Power    Leakage Power    Total Power (Watts)
-----
Sequential          3.83e-05      6.93e-08      1.26e-06      3.96e-05      16.6%
Combinational        1.07e-04      8.74e-05      4.16e-06      1.99e-04      83.4%
Clock                0.00e+00      0.00e+00      0.00e+00      0.00e+00      0.0%
Macro               0.00e+00      0.00e+00      0.00e+00      0.00e+00      0.0%
Pad                  0.00e+00      0.00e+00      0.00e+00      0.00e+00      0.0%
-----
Total                1.45e-04      8.75e-05      5.42e-06      2.38e-04      100.0%
                        61.0%       36.7%       2.3%
```

Nhận xét:

- + Công suất tiêu thụ ước tính của các FF và Latch chiếm 16,6%, trong khi mạch tổ hợp chiếm đa số, vì đây là mạch nhân nhiều bit.
- + Leakage chiếm tỷ lệ rất nhỏ, cho thấy thư viện cell tiêu chuẩn dùng hiệu quả.

2.6. OpenSTA và OpenTimer

Bảng . So sánh nhanh OpenSTA và OpenTimer:

	OpenSTA	OpenTimer
C++API	Không	Có
Mã TCL	Có	Không
Shell	Có	Có
Công suất	Có	Không
Tốc độ	Tốt, nhưng không tối ưu cho timing với thiết kế lớn	Rất nhanh, phù hợp cho thiết kế lớn (>1M gates)
Đa góc	Có	Không cung cấp sẵn, có thể viết thông qua API C++
Khả năng mở rộng	Không	Có thể tích hợp vào công cụ khác
Báo cáo timing cho đường dẫn cụ thể	Có	Không

Tổng quan, OpenSTA dễ dàng hơn cho việc sử dụng trực tiếp, trong khi đó OpenTimer phù hợp hơn cho việc tích hợp và phát triển tiếp nhờ C++API. OpenSTA cung cấp bản báo cáo công suất chi tiết hơn OpenTimer và cung cấp các bản báo cáo dựa trên việc chọn điểm bắt đầu (start-point) và điểm kết thúc (end-point).

### 3. Tham khảo

- [ 1]   OpenTimer, <https://github.com/OpenTimer/OpenTimer>
- [2]   SDC Commands, [https://docs.verilogtorouting.org/en/latest/vpr/sdc\\_commands/](https://docs.verilogtorouting.org/en/latest/vpr/sdc_commands/)
- [3]   The-OpenROAD-Project/OpenSTA, <https://github.com/The-OpenROAD-Project/OpenSTA>
- [4]   Synopsys Design Constraints (SDC) Basics, <https://www.vlsi-expert.com/2011/02/synopsys-design-constraints-sdc-basics.html>
- [5]   NangateOpenCellLib, [https://raw.githubusercontent.com/OpenTimer/OpenTimer/refs/heads/master/example/sizer/NangateOpenCellLibrary\\_typical.lib](https://raw.githubusercontent.com/OpenTimer/OpenTimer/refs/heads/master/example/sizer/NangateOpenCellLibrary_typical.lib)
- [6]   set\_input\_delay (SDC), [http://ebook.pldworld.com/\\_Semiconductors/Actel/Libero\\_v70\\_fusion\\_web\\_help/libero.htm#set\\_input\\_delay\\_\(sdc\\_input\\_delay\\_constraint\).htm](http://ebook.pldworld.com/_Semiconductors/Actel/Libero_v70_fusion_web_help/libero.htm#set_input_delay_(sdc_input_delay_constraint).htm)