CÀI ĐẶT VÀ MÔ PHỎNG

VI XỬ LÝ RISC-V TÙY CHỈNH

TRÊN MODELSIM VÀ VIVADO

Cấu trúc mã nguồn gồm 5 thư mục:

* *core*: Thư mục mã nguồn VHDL của vi xử lý, và các testbench để thực hiện mô phỏng
* *data:* Thư mục chứa ảnh đầu vào, chuyển ảnh đầu vào và bộ lọc thành file bộ nhớ dữ liệu, và mã python nhân chập để đối chiếu.
* *instructions:* Thư mụcfile mã nguồn assembly, các công cụ chuyển đổi từ assembly sang mã máy và chuyển thành file bộ nhớ lệnh.
* *modelsim*: là thư mục chứa project của modelsim, trong đó lưu file bộ nhớ lệnh, bộ nhớ dữ liệu vào và ra
* *vivado*: là thư mục chứa project của Vivado, trong đó lưu file *.tcl* để thiết lập project.

# Chuẩn bị dữ liệu

Quá trinh chuẩn bị dưa liệu bao gồm các việc như: chuyển đổi các ma trận dạng 3x3, 5x5, 7x7 hay ảnh đầu vào 28x28 thành dạng chuỗi mã hex để đưa vào file bộ nhớ lệnh. Các dữ liệu về ảnh và bộ lọc thực hiện nhân chập được lưu trong thư mục *data*.

Bắt đầu bằng việc chuẩn bị ảnh và đưa vào trong thư mục data. Sau đó thực hiện chạy chương trình Python trong cửa sổ lệnh thực hiện nhân chập để lấy kết qảu đối chiếu và xuất ra file bộ nhớ dữ liệu .mem:

* cd <đường dẫn tới thư mục data> (ví dụ: f:\riscv-xcnv1\data)
* python .\conv\_scipy.py -i <ảnh đầu vào> -o <ảnh đầu ra để đối chiếu> -m <file bộ nhớ dữ liệu>

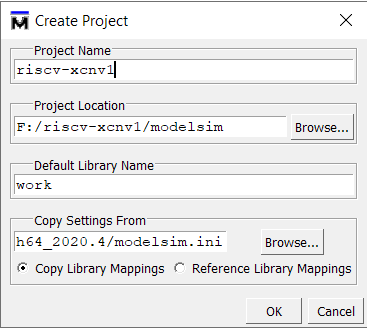
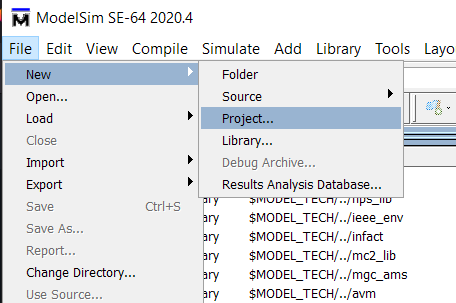
Chương trình thực hiện nhân chập lưu trong thư mục *instructions.* Trong đó file assembly có định dạng .s, chạy chương trình Python chuyển đổi sang file bộ nhớ lệnh .mem:

* cd <đường dẫn tới thư mục instructions> (ví dụ: f:\riscv-xcnv1\data)
* python .\convert.py -i <file assembly> -o <file bộ nhơ lệnh>

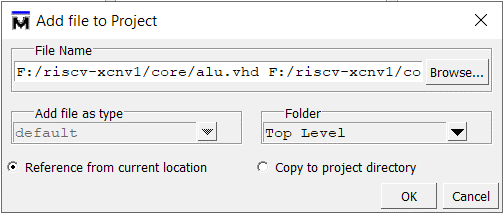
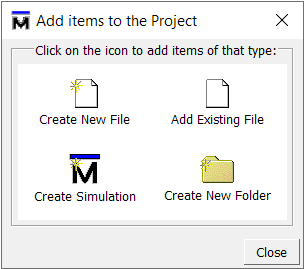
Sau khi xong, file bộ nhớ dữ liệu và bộ nhớ lệnh cần chuyển đến thư mục *modelsim* để thực hiện mô phỏng

# Mô phỏng trên phần mềm ModelSim

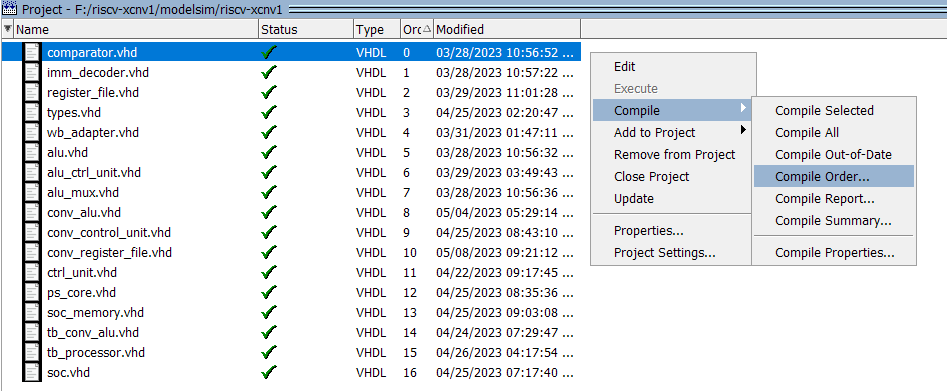
* Sao chép file data.mem và inst.mem vừa tạo trong thư mục *assembly & python* vào thư mục *modelsim.*
* Khởi động phần mềm ModelSim. Để thực hiện mô phỏng cần tạo project mới, chọn Files > New > Project… Sau đó đặt tên và đường dẫn của project, ở đây là đường dẫn tới thư mục *modelsim*.



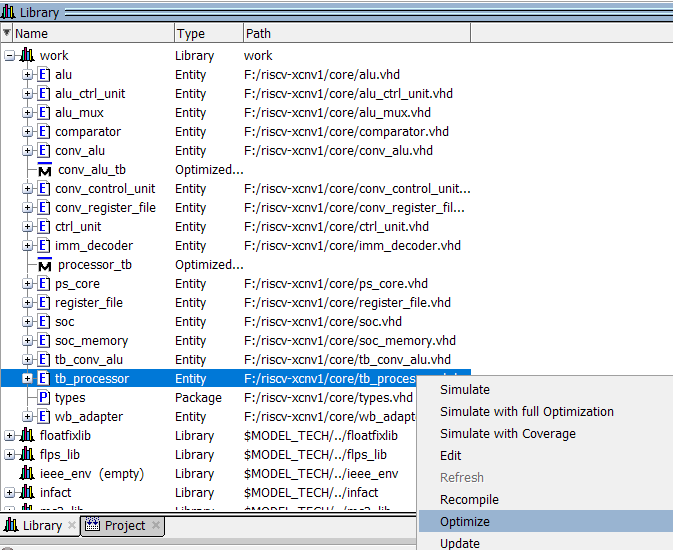
* Chọn thêm tệp đã có (“Add Existing File”) như hình dưới, nhấn “Browse” và chọn toàn bộ file VHDL trong thư mục *core*.



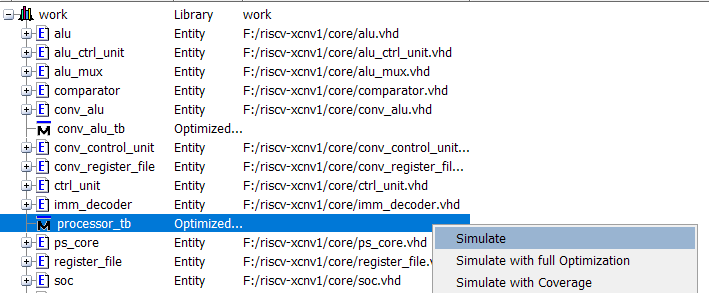
* Thực hiện biên dịch để kiểm tra tệp, có thể chọn biên dịch toàn bộ hoặc biên dịch theo thứ tự. Các file có dấu tích xanh là đã biên dịch thành công và không có lỗi.

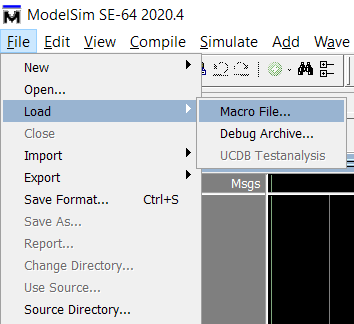
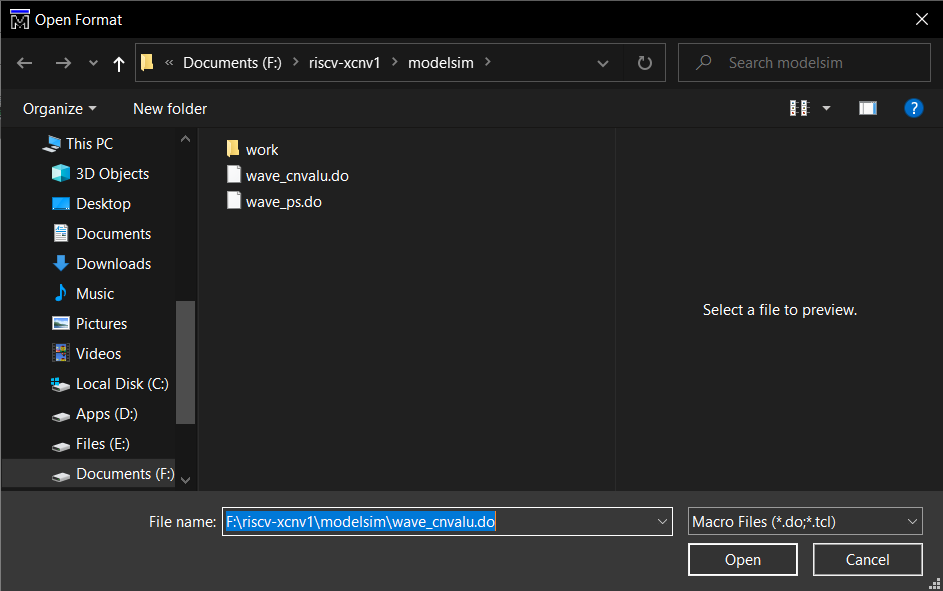


* Chọn tab “Library”, mở thư viện work, trong đó là các file sau khi biên dịch thành các khối, chọn flie *tb\_processor* và *tb\_conv\_alu* và chọn “Optimize” để tối ưu để xuất ra phần mô phỏng như dưới đây

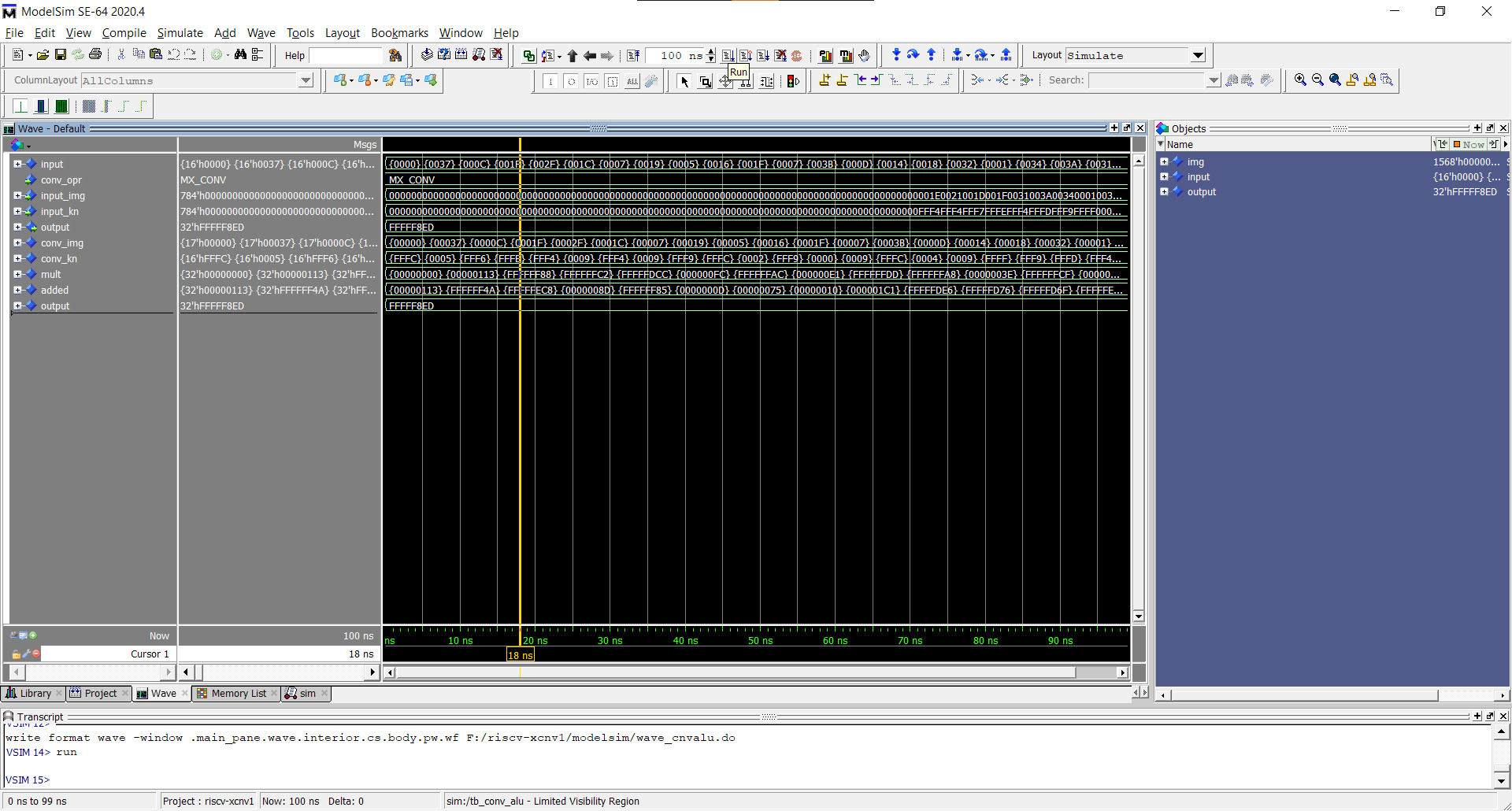


* Thực hiện chạy phần mô phỏng bằng việc chọn “Simulate”, thêm danh sách và dạng tín hiệu vào cửa sổ quan sát Wave bằng cách chọn File > Load > Macro File… > wave\_ps.do (hoặc wave\_cnv\_alu.do, tùy theo thành phần cần mô phỏng kiểm tra).



* Chuyển sang tab “Wave” và chọn khoảng thời gian mô phỏng trong ô được đánh dấu như dưới đây, và thực hiện chạy bằng việc nhấn nút “Run” bên cạnh đó.



* Đánh giá kết quả dựa trên màn hình hiển thị. Ngoài ra kết quả dữ liệu bộ nhớ còn được thể hiện trong tab Memory List.

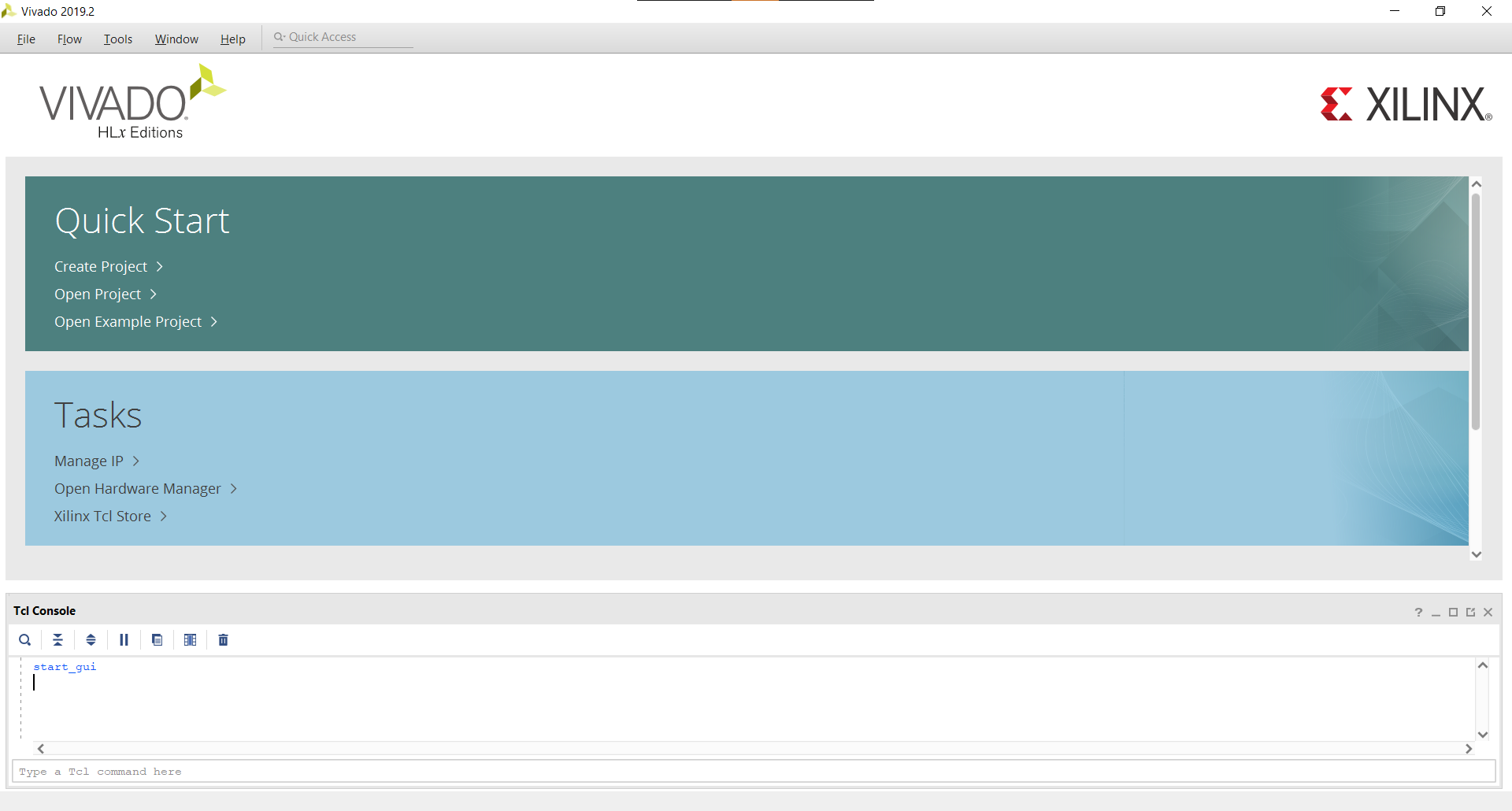
Sau khi hoàn thành, file dữ liệu kết quả nhân chập được xuất ra trong thư mục modelsim, có thể chuyển sang thư mục data và thực hiện chuyển đổi file bộ nhớ thành ảnh:

* python .\hextoimg.py -i <file kết quả bô nhớ> -o <ảnh đầu ra> -s <kích thước ảnh đầu ra>

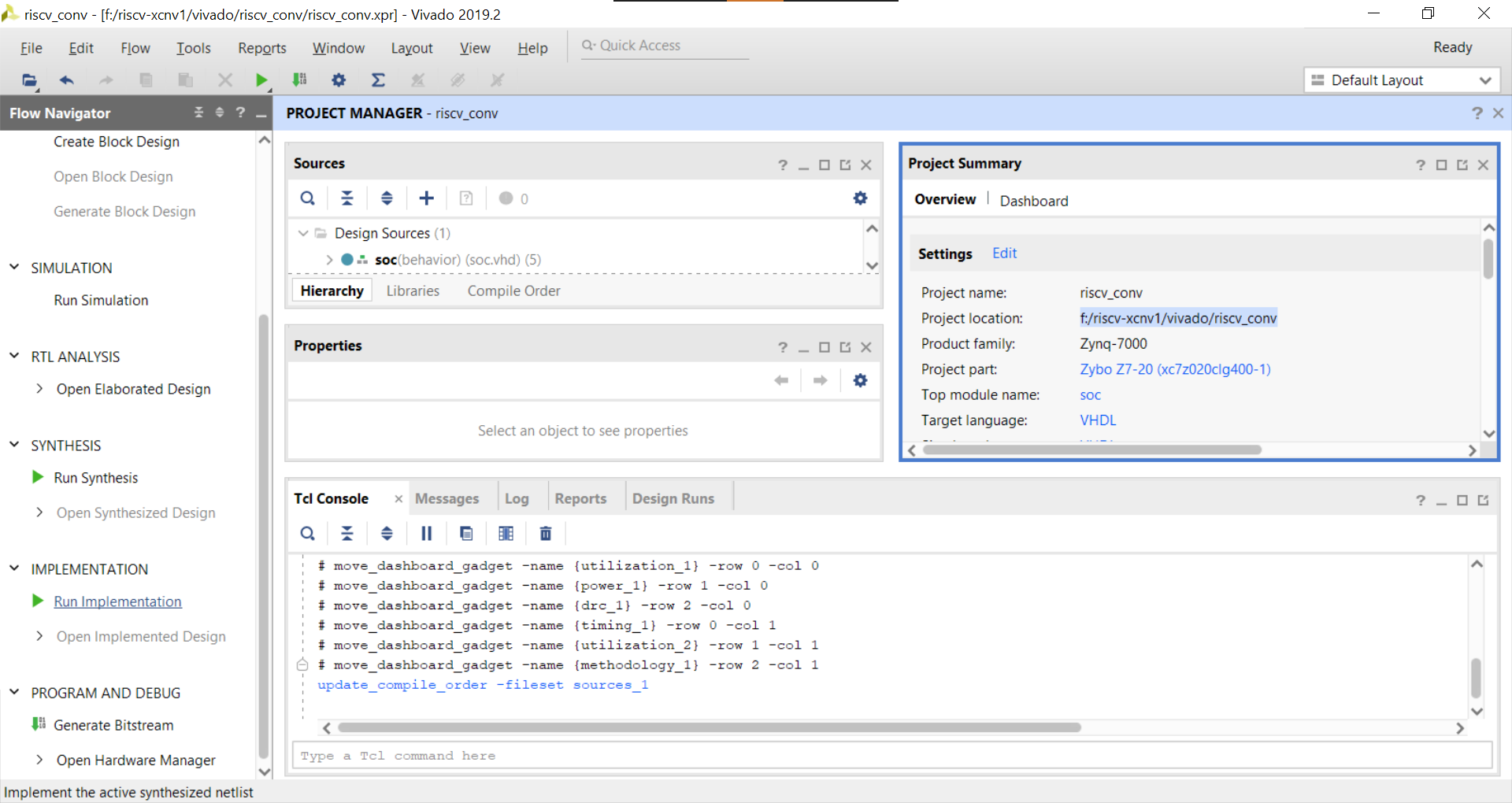
# Cài đặt trên Vivado

Sau khi mô phỏng và thử nghiệm trên ModelSim, tiến hành cài đặt trên Vivado

* Bước 1: Khởi chạy chương trình Vivado, và mở “TCL Console” ở phía dưới phải của cửa sổ



* Bước 2: Thực thi 2 dòng lệnh dưới để tạo project chứa các file cần thiết để cài đặt.
* cd <đường dẫn tới thư mục project Vivado> (ví dụ: f:/riscv-xcnv1/vivado/)
* source ./riscv\_conv\_proj.tcl
* Bước 3: Sau khi hoàn thành tạo project, giao diện thu được như hình dưới. Thực hiện “Run Synthesis” > “Run Implementation” để tổng hợp phần cứng và thực hiện cài đặt.



* Bước 4: Mở thiết kế sau cài đặt (“Open Implemented Design”), sau đó nghiệm thu kết quả và đánh giá.

