

**DOLPHIN TRAINING**

**FPGA Zero to Hero**

***Overview***

***Revision1.0***

**10 March 2021**

**Overview**

**DOLPHIN TRAINING**

**FPGA Zero to Hero**

**Revision**

**10 March 2021**

# TABLE OF CONTENTS

[TABLE OF CONTENTS 2](#_Toc66284583)

[LIST OF FIGURES 3](#_Toc66284584)

[LIST OF TABLES 4](#_Toc66284585)

[REVISION HISTORY 5](#_Toc66284586)

[1. Introduction 6](#_Toc66284587)

[1.1. Cấu tạo của FPGA 6](#_Toc66284588)

[1.1.1. Configurable logic blocks 6](#_Toc66284589)

[1.1.1.1. Locgic cell 7](#_Toc66284590)

[1.1.1.2. LUT-Based Memory 9](#_Toc66284591)

[1.1.2. Dedicated blocks 9](#_Toc66284592)

[1.1.2.1. Memory Block 10](#_Toc66284593)

[1.1.3. IOB Element 11](#_Toc66284594)

[1.1.4. Routing 12](#_Toc66284595)

[1.1.5. Clock 12](#_Toc66284596)

[1.1.6. Dedicated and Special Resources 13](#_Toc66284597)

[2. Programming an FPGA 13](#_Toc66284598)

[2.1. Bước 1 – Cấu trúc thư mục 14](#_Toc66284599)

[2.2. Bước 2 – Packet IP 14](#_Toc66284600)

[2.3. Bước 3 – Tạo block design 24](#_Toc66284601)

[2.4. Bước 4 – Generate Block Design 28](#_Toc66284602)

[2.5. Bước 5 – Create hdl wrapper 28](#_Toc66284603)

[2.6. Bước 6 – Export Harware 29](#_Toc66284604)

[2.7. Bước 7 – SDK 29](#_Toc66284605)

[2.8. Bước 8 – Build testbench & simulation 33](#_Toc66284606)

[2.9. Bước 9 – Synthesis 33](#_Toc66284607)

[2.10. Bước 10 – Implement 33](#_Toc66284608)

[2.11. Bước 11 – Generate bitstream 33](#_Toc66284609)

[2.12. Bước 12 – Export hardware with bitstream 33](#_Toc66284610)

[2.13. Bước 13 – Bật SDK, nạp kit, debug 33](#_Toc66284611)

# LIST OF FIGURES

[Figure 1. Cấu tạo FPGA 6](#_Toc66284612)

[Figure 2: Logic cell architecture 7](#_Toc66284613)

[Figure 3: Carry Chain 8](#_Toc66284614)

[Figure 4: LUT-Based Memory 9](#_Toc66284615)

[Figure 5: Memory block 10](#_Toc66284616)

[Figure 6: IOB Element 11](#_Toc66284617)

[Figure 7: Routing 12](#_Toc66284618)

[Figure 8: 1.1.6. Dedicated and Special Resources 13](#_Toc66284619)

# LIST OF TABLES

**No table of figures entries found.**

# REVISION HISTORY

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Description of Changes** |
| 0.0.1 | 07 December 2020 | Original |

# Introduction

Bài viết ở link dưới đây sẽ trả lời 3 câu hỏi: “FPGA là gì?”, “Sự khác biệt giữa ASIC và FPGA” – 1 video 1h, “Ứng dụng của FPGA”.

[www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html](http://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html)

## Cấu tạo của FPGA

Cấu tạo cơ bản của FPGA.

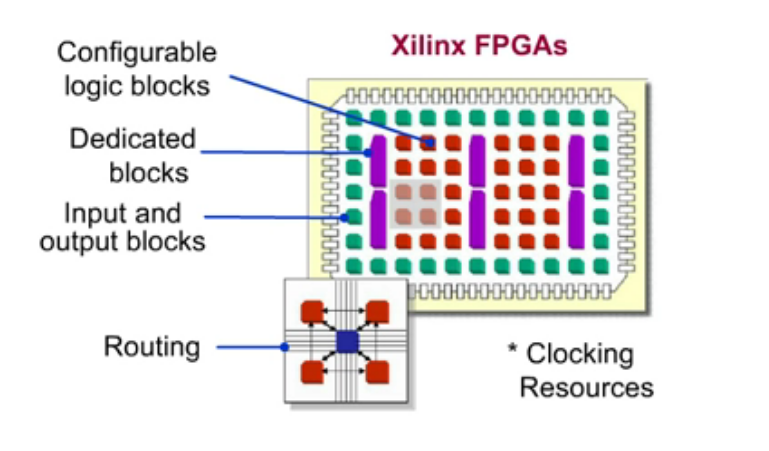


Figure 1. Cấu tạo FPGA

### Configurable logic blocks

Configurable logic blocks thường chứa 1 vài logic cell, cấu tạo của 1 logic cell.

#### Locgic cell

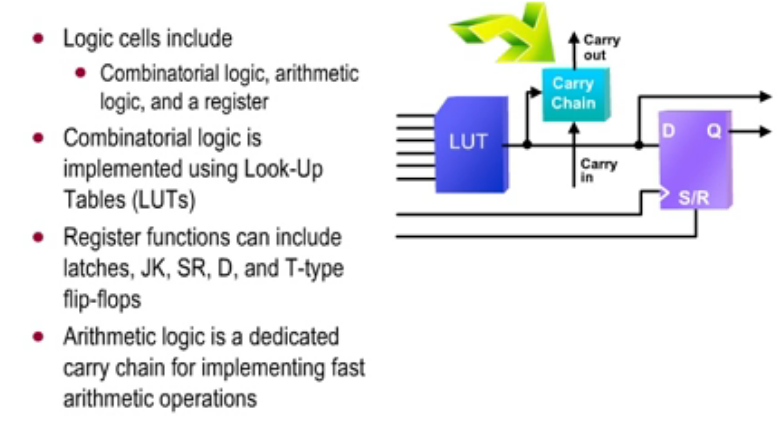


Figure 2: Logic cell architecture

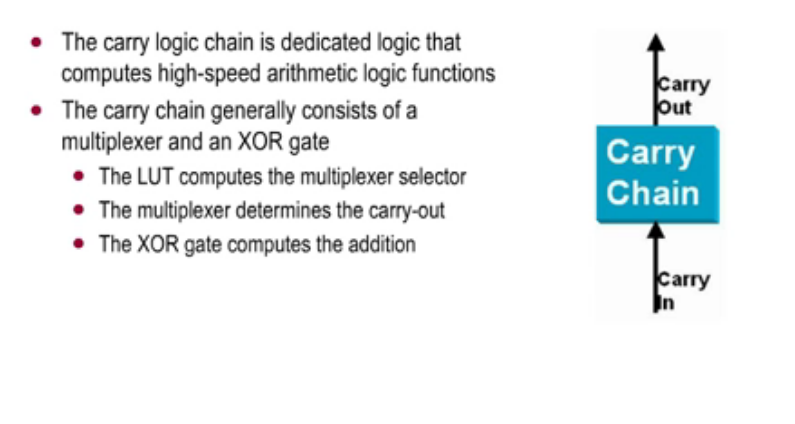


Figure 3: Carry Chain

#### LUT-Based Memory

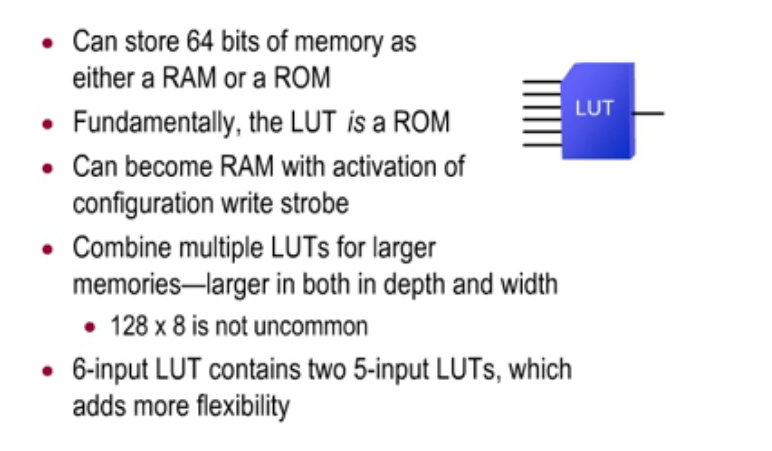


Figure 4: LUT-Based Memory

### Dedicated blocks

Đây là các khối IP chuyên dụng được đặt sẵn trong mạch để tiết kiệm diện tích và tối ưu tốc độ. Có tài liệu gọi tên là Hard blocks

Vd: Bộ nhân, DSP, embedded RAM, PCI, PHY, ….

#### Memory Block

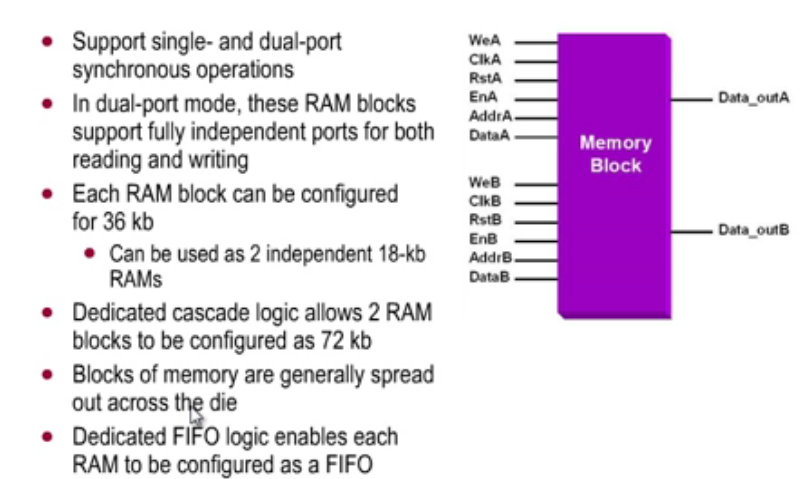


Figure 5: Memory block

### IOB Element

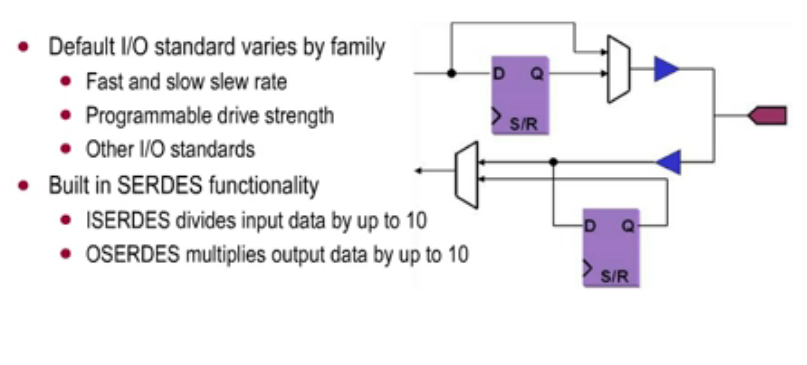


Figure 6: IOB Element

### Routing

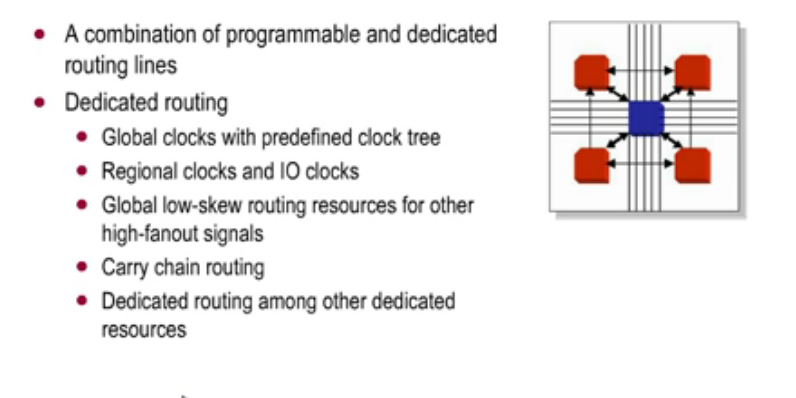
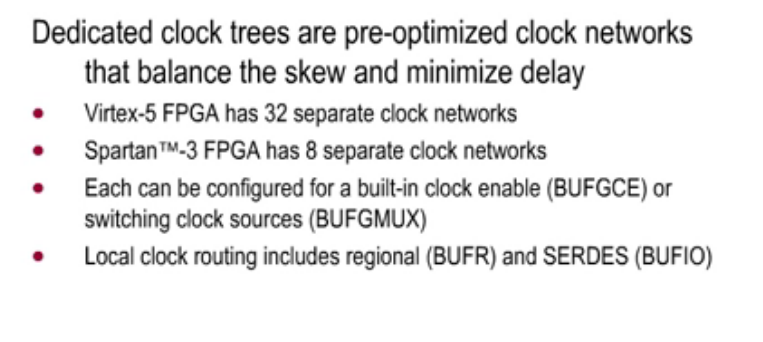


Figure 7: Routing

### Clock



### Dedicated and Special Resources

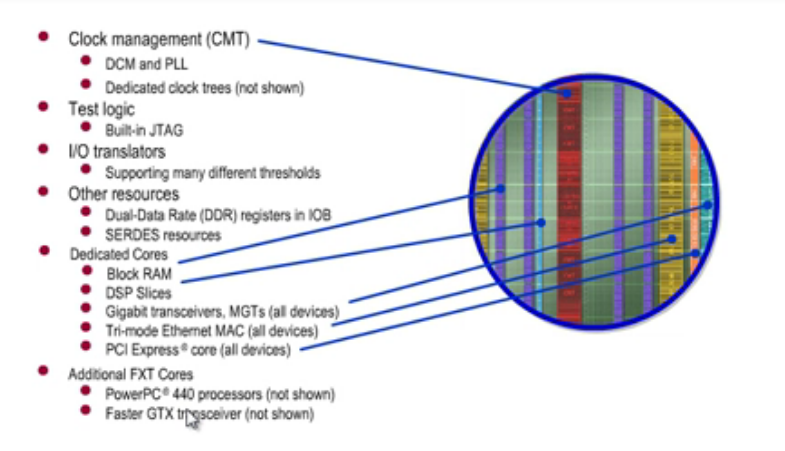


Figure 8: 1.1.6. Dedicated and Special Resources

# Programming an FPGA

Dolphin sử dụng FPGA với mục đích test & demo IP cho khách hàng.

## Bước 1 – Cấu trúc thư mục

Sau khi check out project của IP cần test về, cấu trúc thư mục theo định dạng sau (Lưu ý: các thư mục con của thư mục Fpga sẽ đc tạo khi sử dụng vivado, chỉ cần tạo thư mục Fpga và Fpga/<name\_pj>\_fpga\_lib):

<name\_pj>

* Hdl
* Sim
* Lib
* Syn
* Fpga
  + <name\_pj>\_fpga\_interface // chứa tất cả các interface user tự tạo
  + <name\_pj>\_fpga\_top // chứa top fpga pj, block design….
    - <name\_pj>\_fpga\_top.cache
    - <name\_pj>\_fpga\_top.hw
    - <name\_pj>\_fpga\_top.ip\_user\_files
    - <name\_pj>\_fpga\_top.runs // xóa file này nếu gặp lỗi k mở lại được pj (chứa file đang chạy)
    - <name\_pj>\_fpga\_top.sdk // chứa pj cho Microblaze, elf, platform,..
    - <name\_pj>\_fpga\_top.sim // các file gen ra khi chạy sim
    - <name\_pj>\_fpga\_top.srcs // chứa source add thêm, tb,…
    - <name\_pj>\_fpga\_top.tmp
  + <name\_pj>\_fpga\_<user\_ip1> // chứa pj packet IP cần dùng trong block design
    - Tương tự như top // nếu tạo đúng hướng dẫn file componet.xml sẽ ở đây
  + <name\_pj>\_fpga\_<user\_ip2>
  + <name\_pj>\_fpga\_<user\_ip3>
  + <name\_pj>\_fpga\_<user\_ip4>
  + <name\_pj>\_fpga\_lib // chứa source các IP cần dùng trong block design

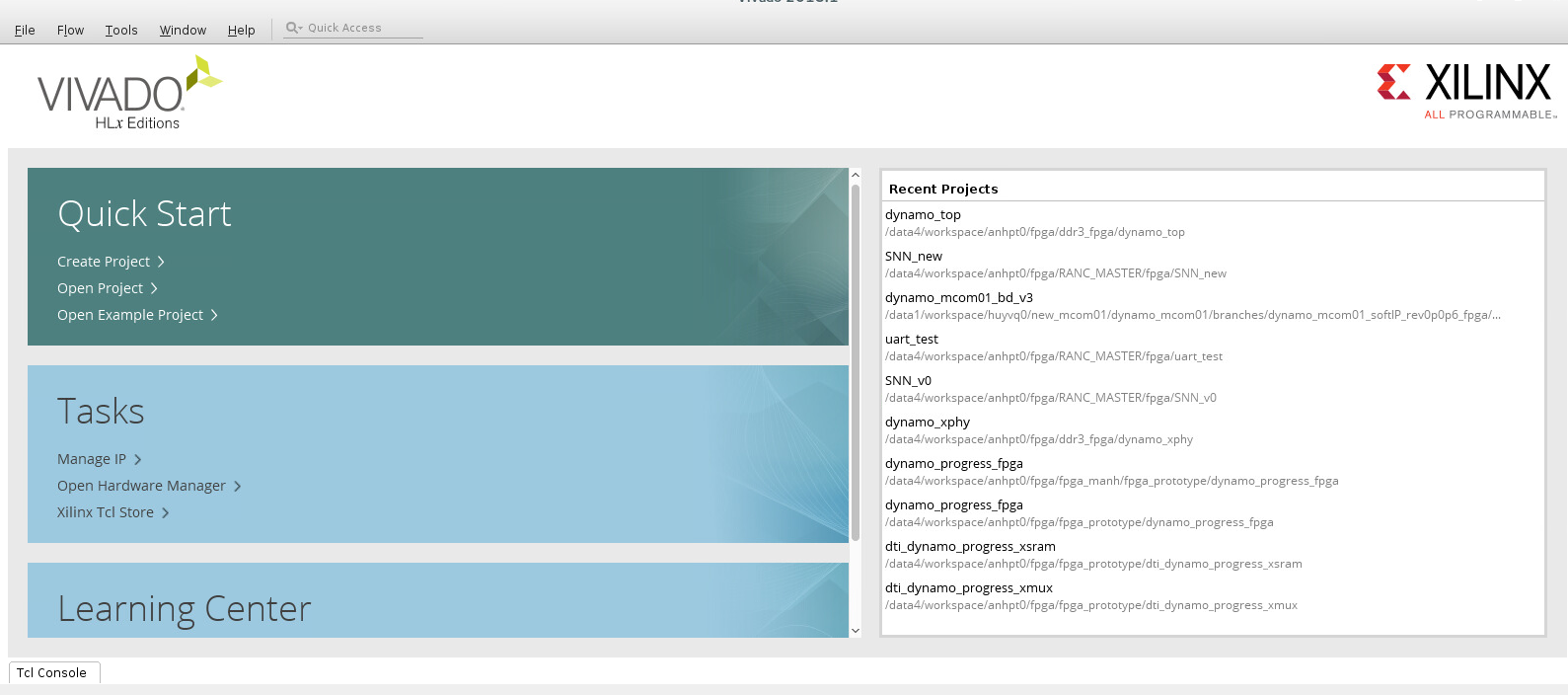
## 

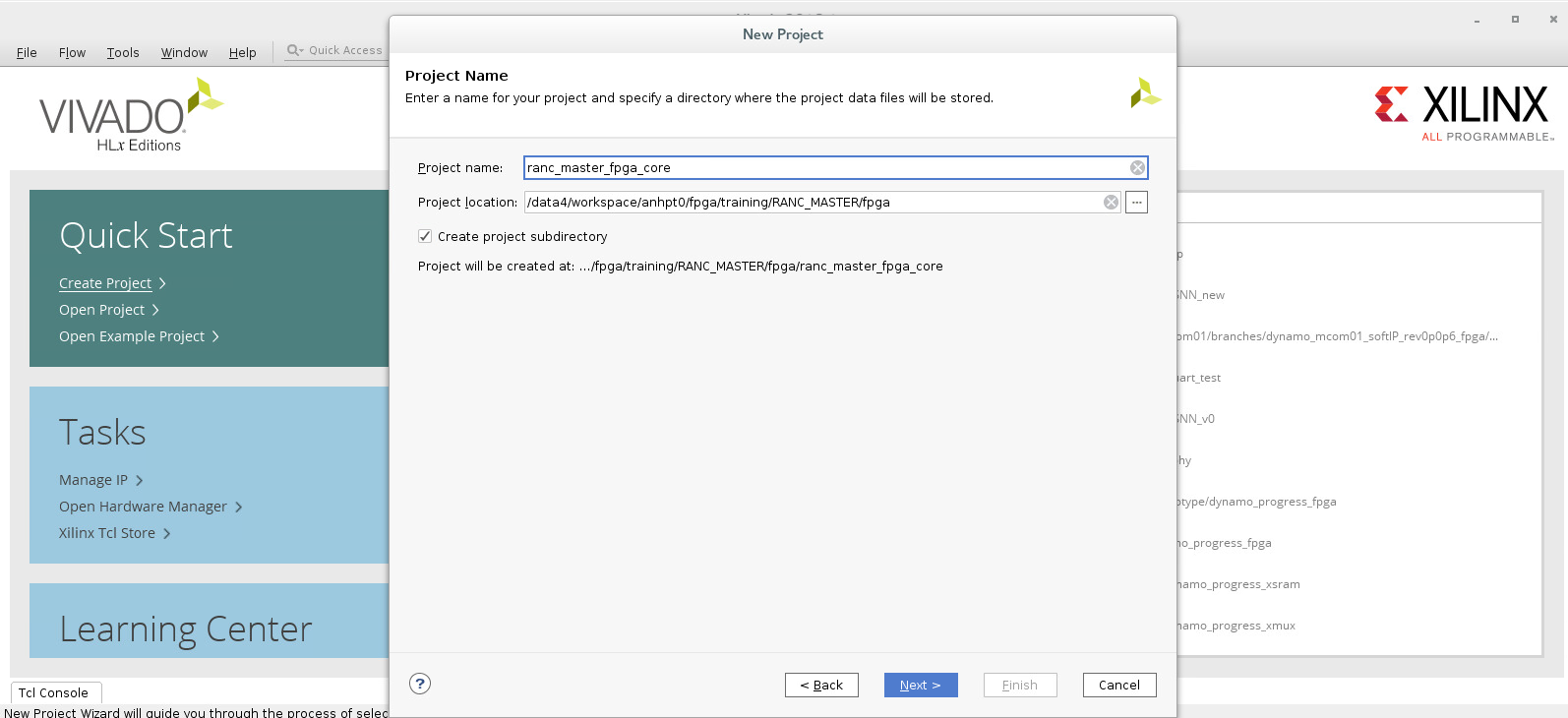
## Bước 2 – Packet IP

Trước khi có thể add các IP user vào block design cần packet các IP đó lưu tên pj <name\_pj>\_fpga\_<user\_ip1>

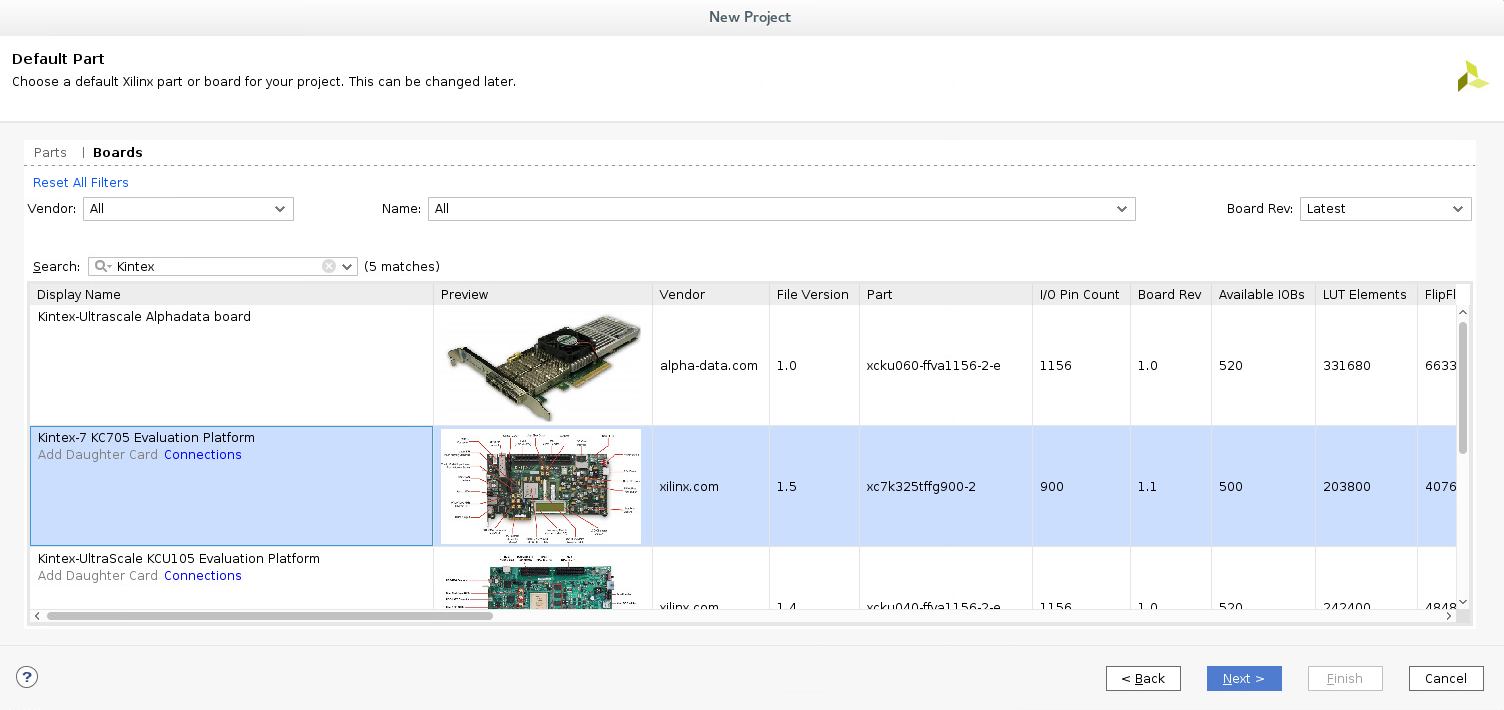
Vào thư mục fpga và bật vivado -> giao diện bắt đầu

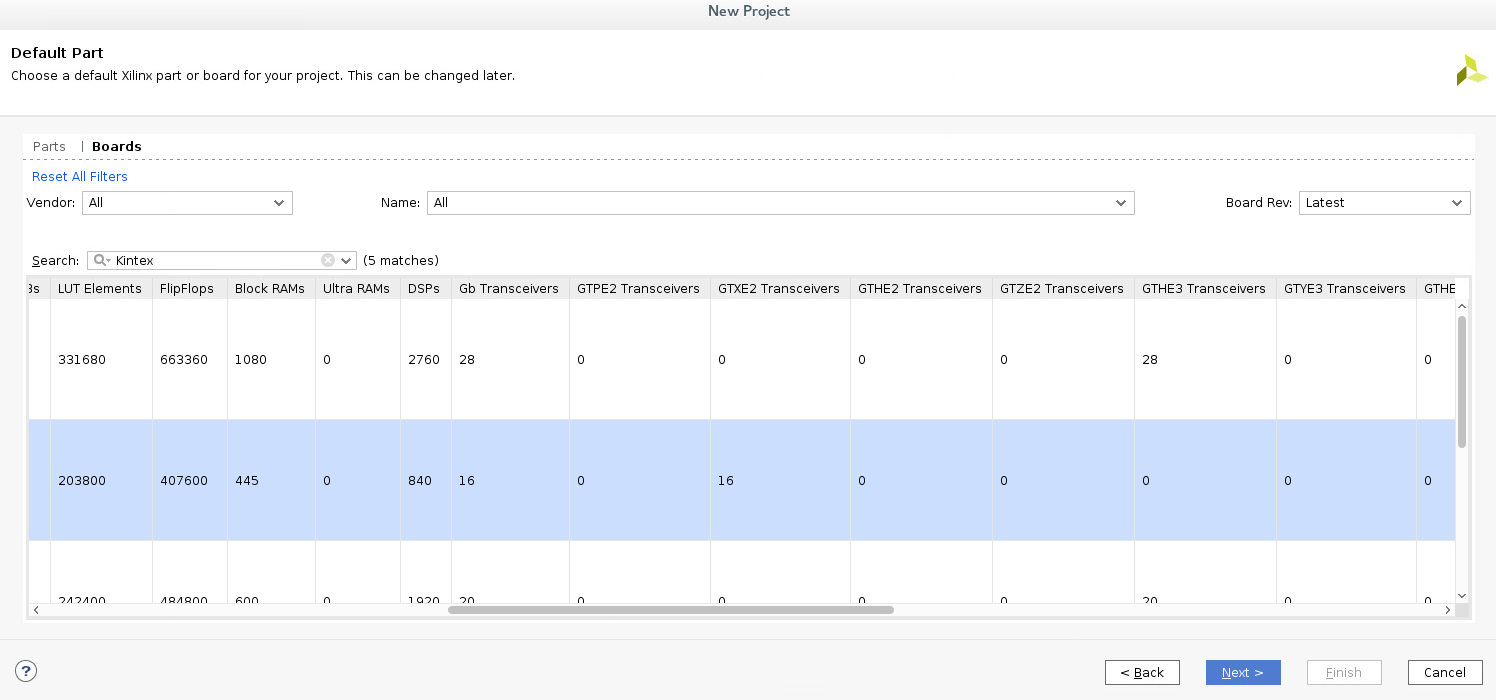
* Xilinx Kintex UltraScale
* Part: xcku095-ffvb2104-2-e
* Microprocessor: Xilinx MicroBlaze

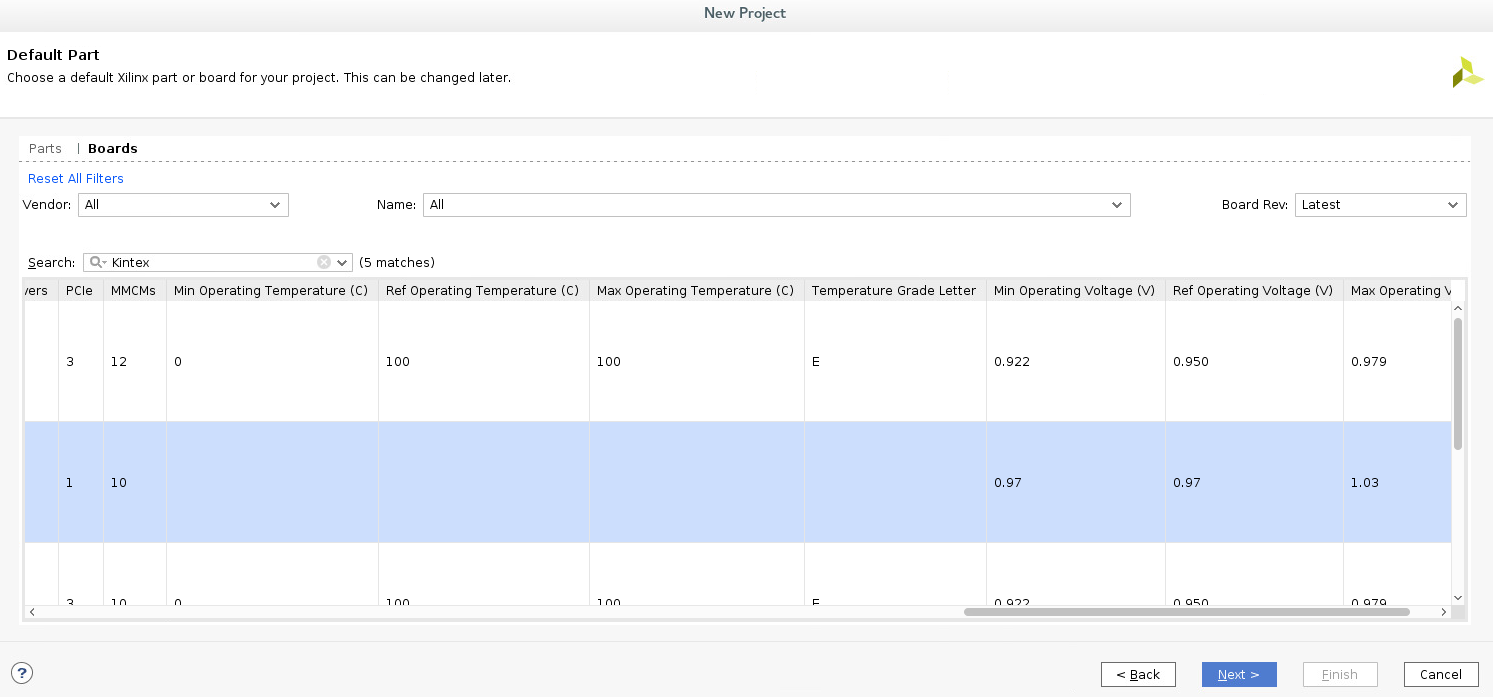




* + - * Part: Tên con FPGA
      * I/O pin count: số lượng chân FPGA
      * Available IOBs: Số lượng I/O buffer







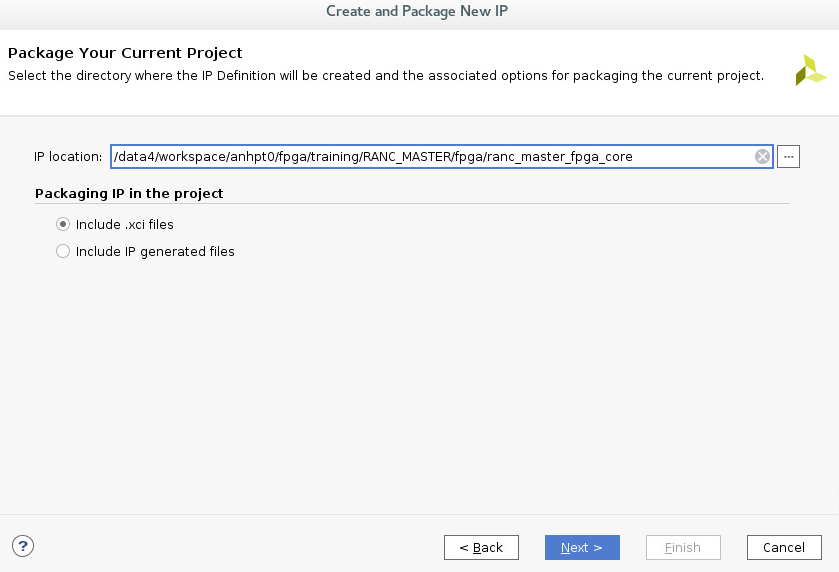
Sau khi finish chọn vào dấu + trong cửa sổ sources để add sources của IP vào



Nhớ chọn file top cho đúng(file top sẽ in đậm) nếu thấy sai chọn vào file top chuột phải set as top.

Bước này lưu ý chọn lại IP loaction vào đúng project vừa tạo, nếu k lúc add vào block design sẽ k tìm thấy IP.

Vào Tools → Create an package new IP

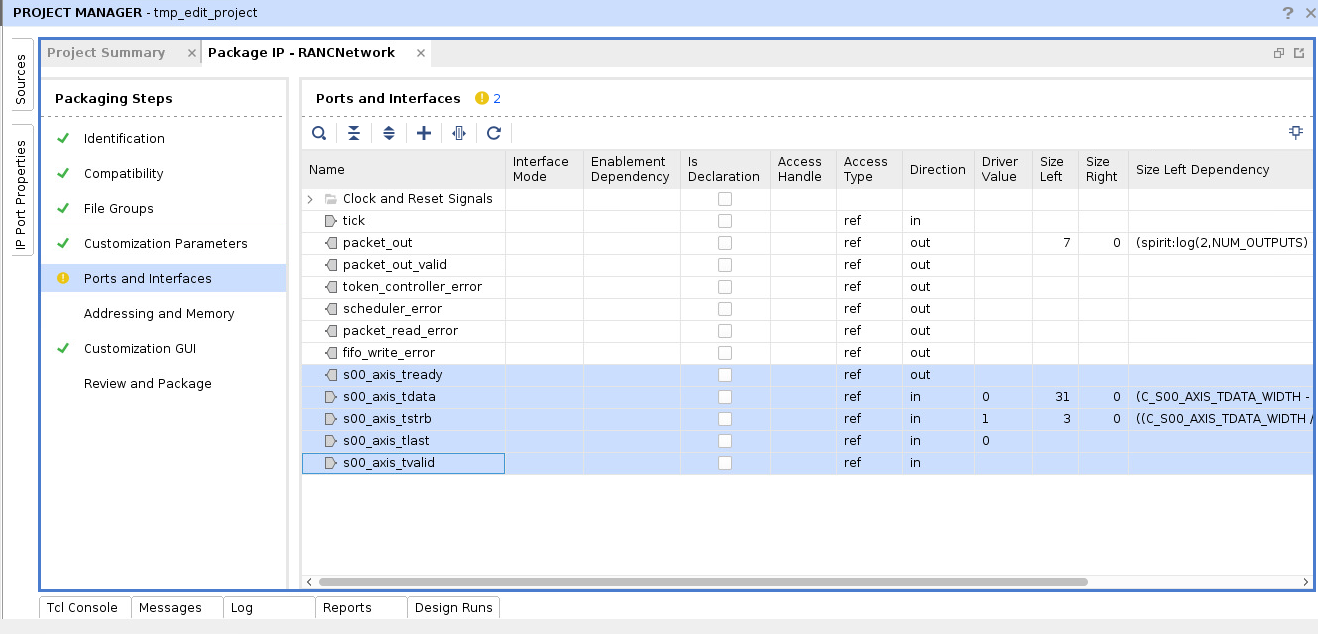


Chọn Tools -> create and packet IP.

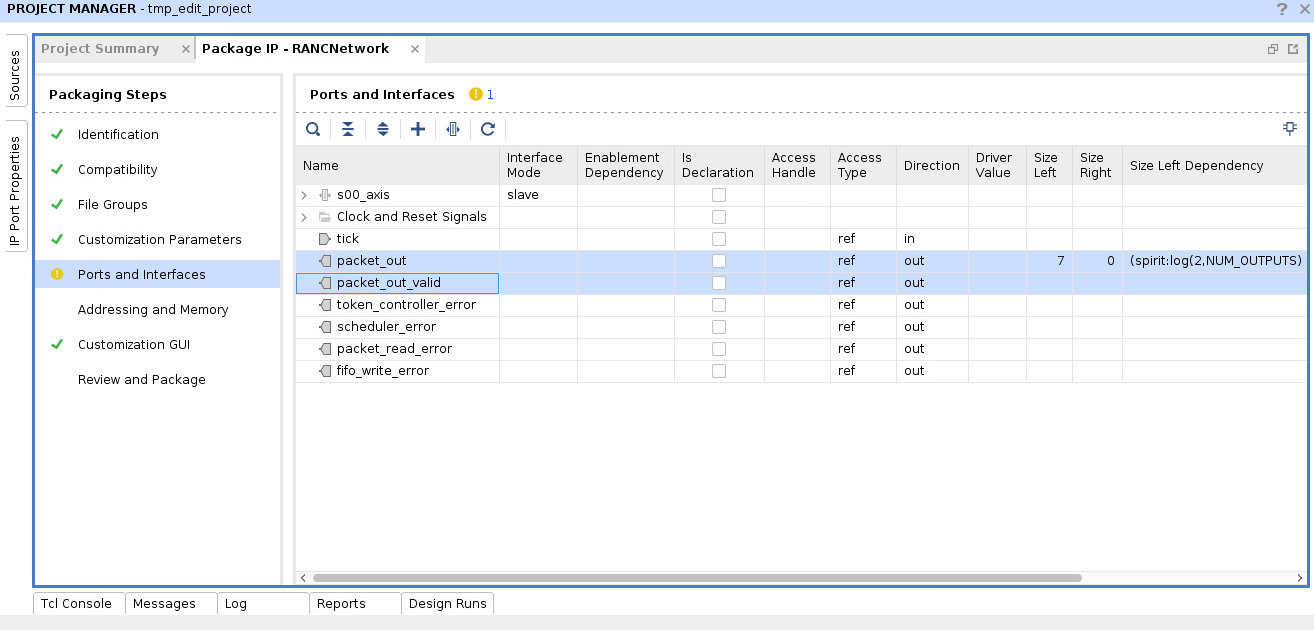
Lưu ý phần Ports and Interfaces. Phần này gom các tín hiệu cùng thuộc 1 loại interface vào với nhau.

Ở đây sẽ có 2 trường hợp:

* Interface phổ biến Xilinx có sẵn vd: axis, axi, apb, …. -> bôi đen, chuoojt phari và chọn auto refer đến tên interface đó -> click ok

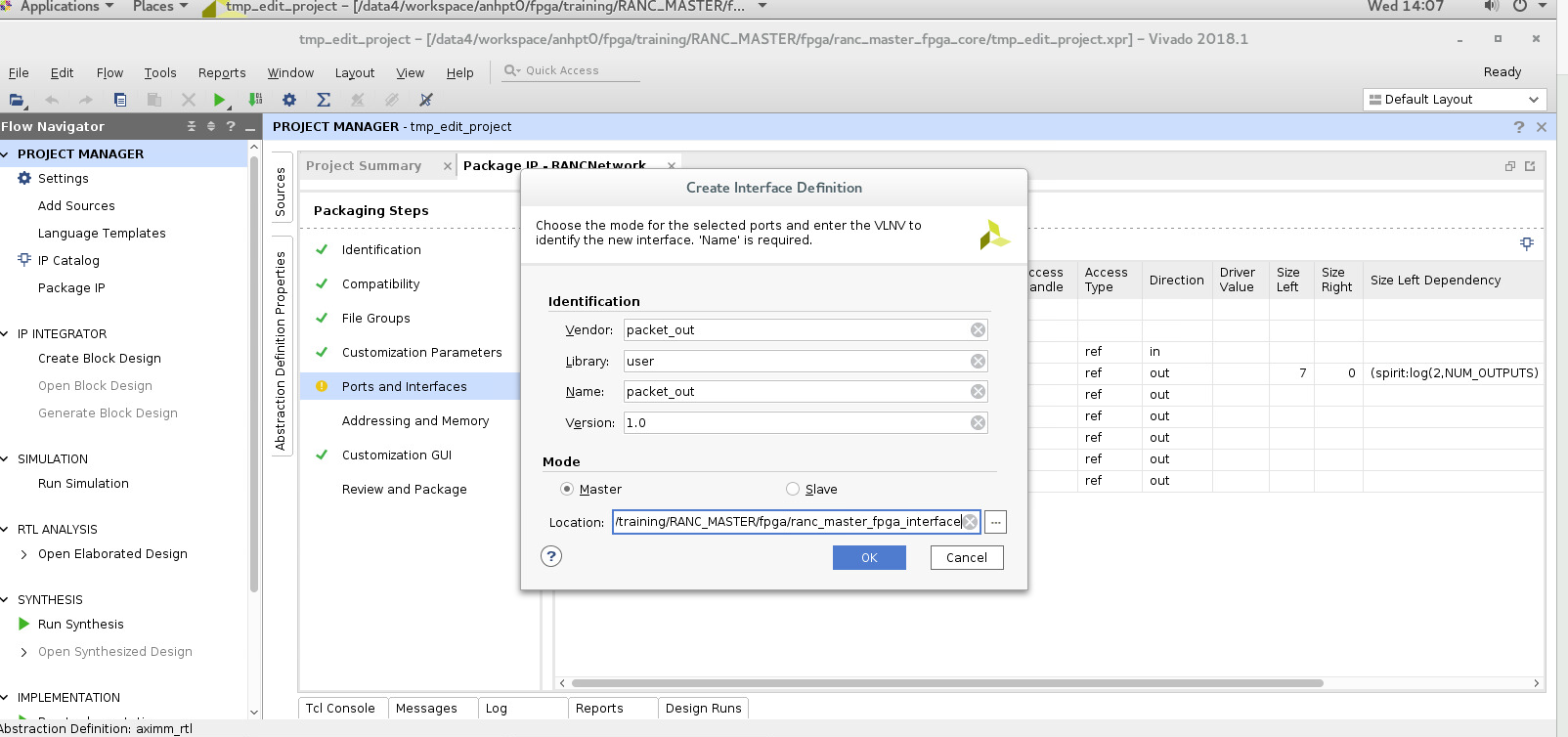


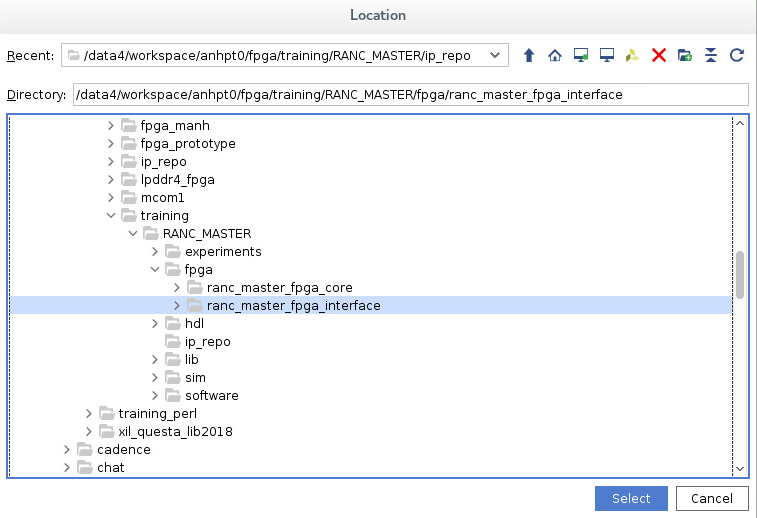
* Trường hợp 2: Interface k có sẵn, người dùng định nghĩa -> bôi đen → Tools -> chọn create interface define…



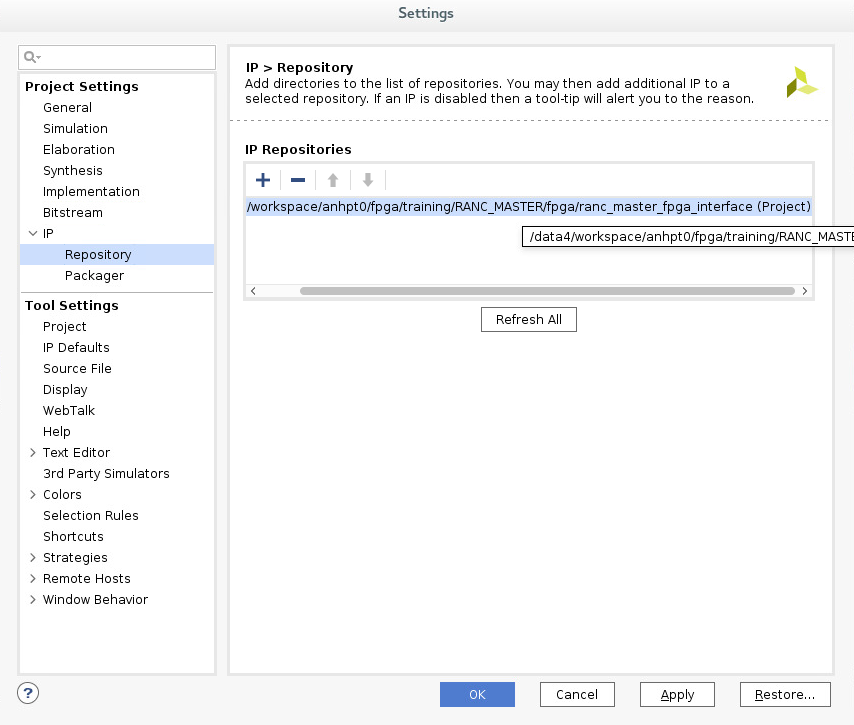
Điền tên Vendor và tên interface

Lưu ý chọn location vào thư mục <name\_pj>\_fpga\_interface như phần 1 có quy định. Điều này giúp khi vẽ block design các interface người dùng tương tác k gây lỗi và khi packet IP khác có cùng interface này k phải làm lại.

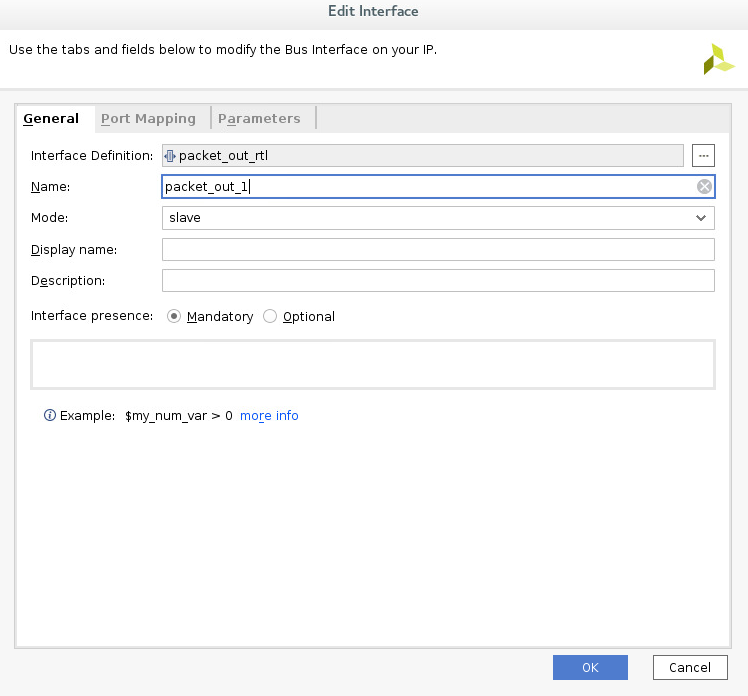




Khi packet 1 IP khác có cùng interface vừa tạo, chọn settting -> IP -> Repository -> dẫn link đến thư mục interface



Khi nhóm tín hiệu ở bước packet chỉ cần chọn lại interface đã tạo bôi đen -> auto refer -> tìm đúng tên -> ok

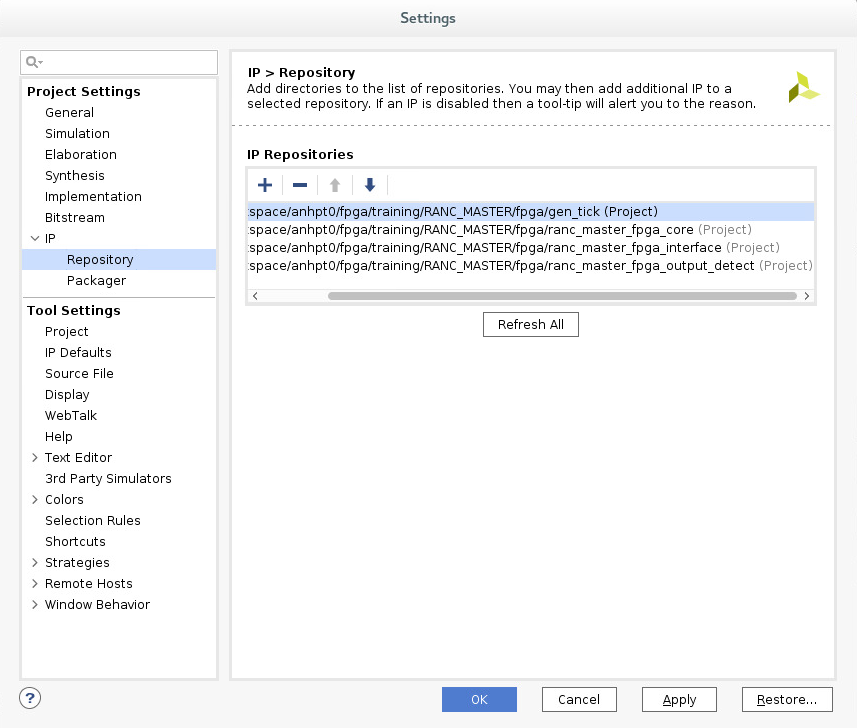


## Bước 3 – Tạo block design

- Tao project dti\_pcm2pdm\_fpga\_top giong nhu fpga\_core nhung ko can add file .sv ma chi can add duong link den cac packet vua tao trong muc Setting -> IP -> Repository

Sau khi packet toàn bộ IP cần thiết tạo pj với tên <name\_pj>\_fpga\_top

Chọn Setting -> add đường dẫn các IP vừa tạo và đường dẫn thư mục interface.



Chon Create Block Design -> OK

Add Microblaze và chọn các thông số.

Add AXI interconnect

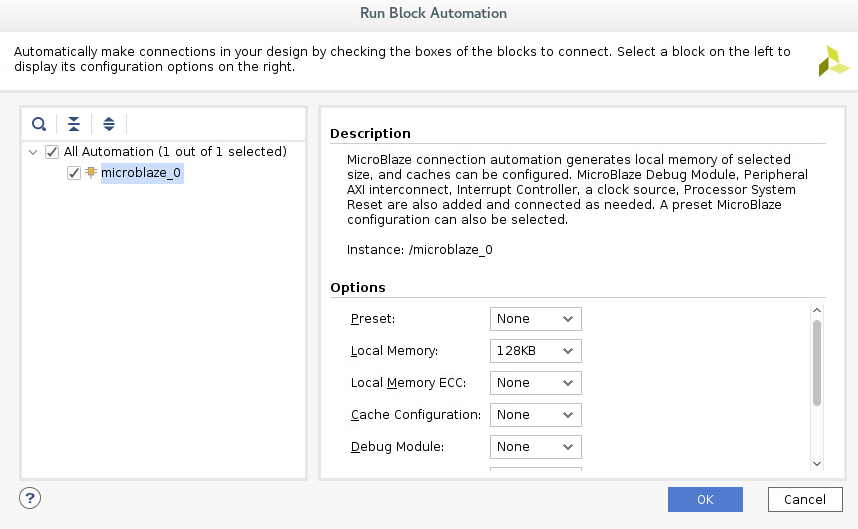
Add BRAM, ctrl\_bram

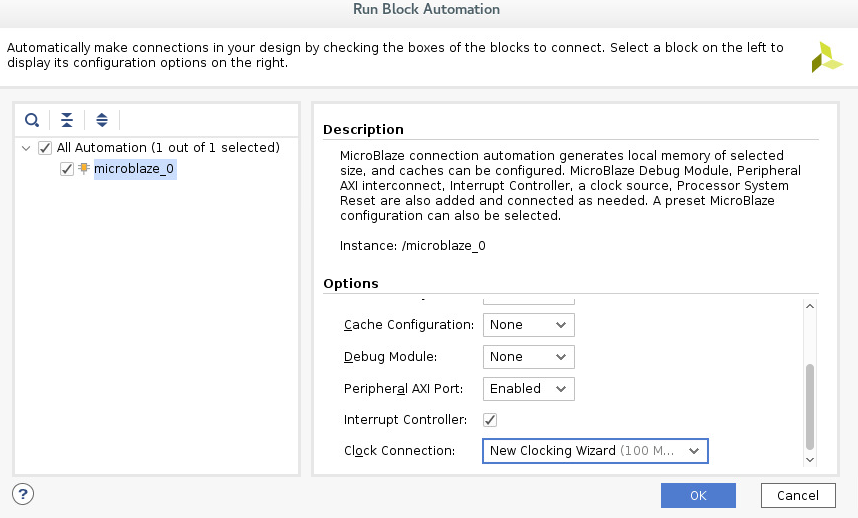
Add IP minh vua tao

Noi cac chan

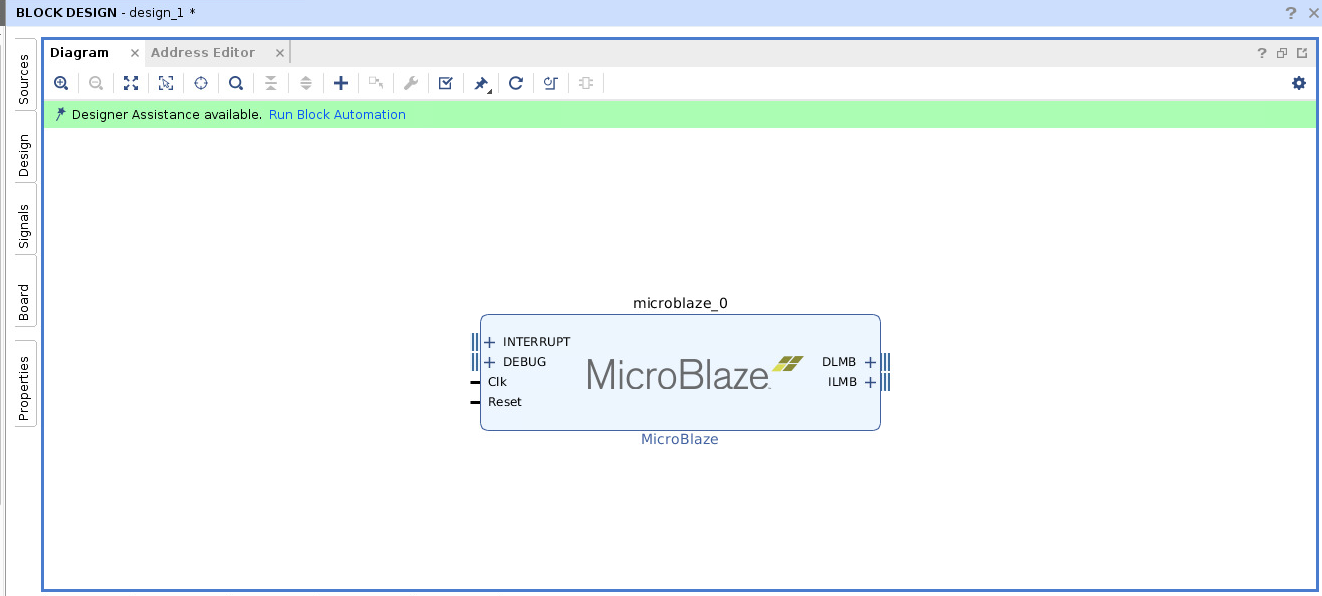
Local Memory: Bộ nhớ cho phần thực thi các lệnh software.

ECC: kiểm tra lỗi + fix lỗi

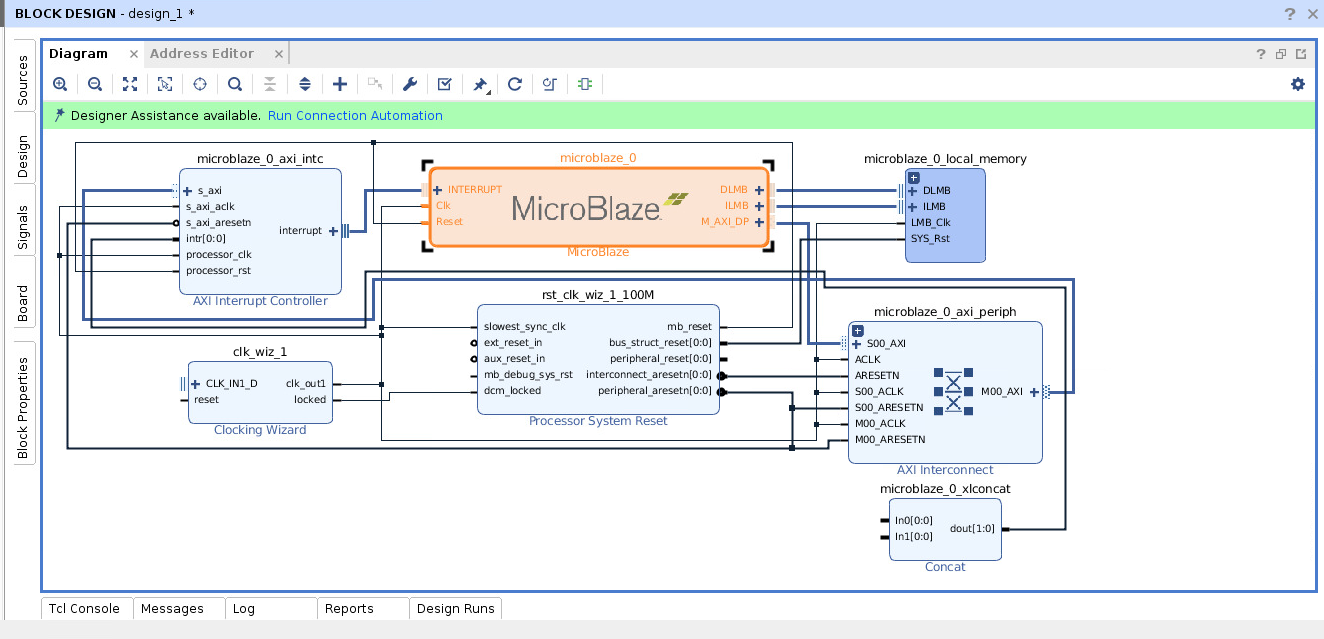




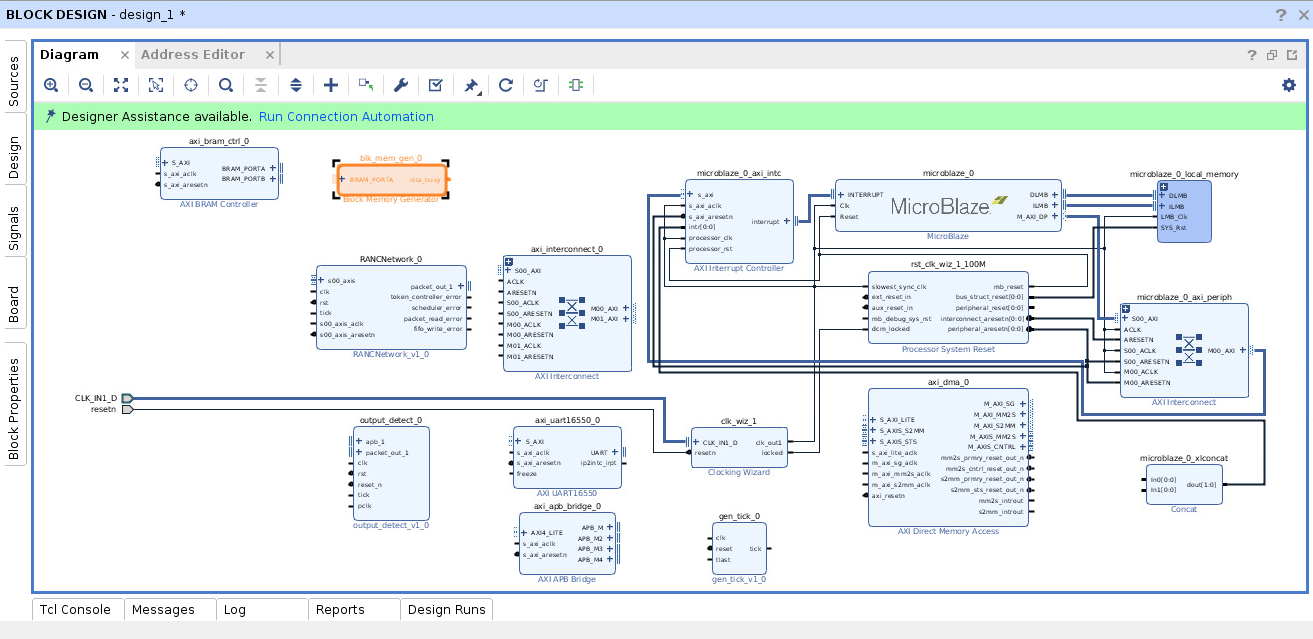
Click ok -> Run block Automation



Các khối cơ bản sẽ tự động thêm vào và nối dây



Add toàn bộ IP cần thiết vào



Nối dây và chọn sang cửa sổ address editor -> update -> validate design.

## Bước 4 – Generate Block Design

Bam chuot phai vao design1.bd -> Generate output products

Bước này sẽ tạo top-level netlish của BD.

Có 3 option:

* Global: Tạo top-level netlish (tổng hợp theo kiểu top-down)
* Out of context per IP (mặc định): Tổng hợp từng IP riêng lẻ -> bỏ qua các IP đã tổng hợp nếu nó k thay đổi.
* Out of context per BD: Sử dụng khi chạy tổng hợp của bên thứ 3.

## Bước 5 – Create hdl wrapper

Chuot phai vao design1.bd -> Create hdl wrapper

Tạo file hdl cho block design. Ở đây có 2 option

* Copy generated ……: Sử dụng khi block design là 1 thành phần của 1 project khác
* Let Vivado ….: Sử dụng khi BD là top

## Bước 6 – Export Harware

Run synthesis o ben trai

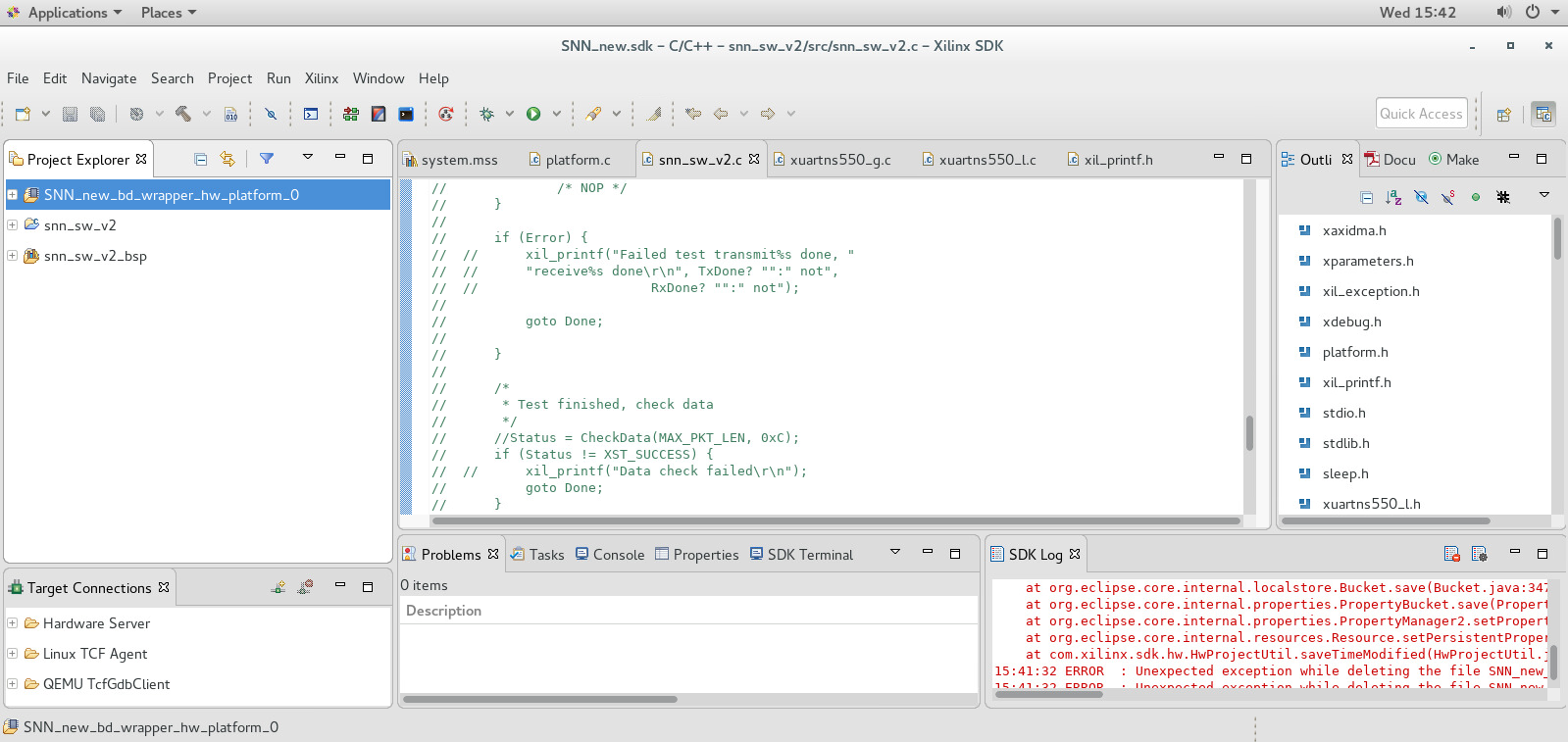
Gen bitstream xong thì vào file -> export hardware

Bước này tạo ra platform: 1 file khai báo phần cứng gồm những gì, đánh địa chỉ ra sao,….

## Bước 7 – SDK

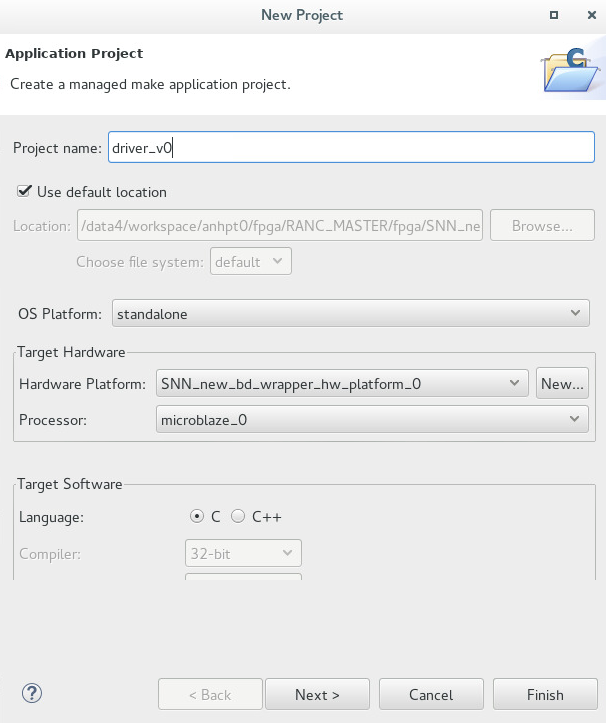
Bước này build 1 application cho MB(Microblaze) dựa trên platform vừa tạo ra từ bước Export Harware.

Tại vivado chọn File -> lauch SDK -> giao diện như dưới ( 2 folder snn\_sw\_v2 và snn\_sw\_v2\_bsp được tạo từ trước, xem như không có 2 folder này)

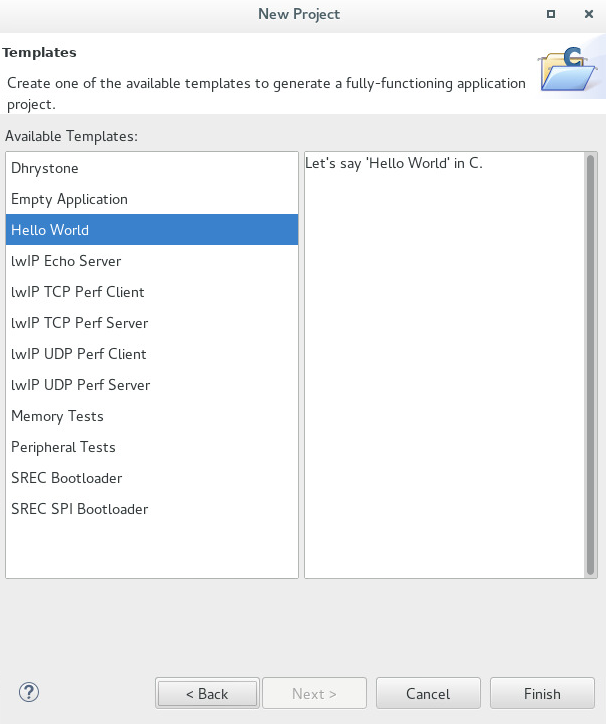


Ta thấy folder SNN\_new\_bd\_wrapeer\_hw\_platform\_0. Folder này chứa các file tạo ra từ bước export hardware.

File -> New -> application pj



Next





## Bước 8 – Build testbench & simulation

## Bước 9 – Synthesis

## Bước 10 – Implement

## Bước 11 – Generate bitstream

## Bước 12 – Export hardware with bitstream

## Bước 13 – Bật SDK, nạp kit, debug

* Xilinx Kintex UltraScale
* Part: xcku095-ffvb2104-2-e
* Microprocessor: Xilinx MicroBlaze