

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM KHOA: ĐIỆN – ĐIỆN TỬ**

**MÔN: THỰC TẬP THIẾT KẾ VI MẠCH VLSI**

**------oOo-----**

BÁO CÁO

**BÁO CÁO LAB4 HW/SW CODESIGN TRÊN VIVADO**

|  |  |
| --- | --- |
| **GVHD:** PSG. TS. PHAN VĂN CA |  |
| **SVTH:** | **MSSV:** |
| ĐẶNG HƯNG THỊNH | 22119137 |

Lớp thứ 2 – Tiết 1- 5

**Tp. Hồ Chí Minh, tháng 04 năm 2025**

**ĐIỂM SỐ**

|  |  |  |  |
| --- | --- | --- | --- |
| **TIÊU CHÍ** | **NỘI DUNG** | **TRÌNH BÀY** | **TỔNG** |
| **ĐIỂM** |  |  |  |

**NHẬN XÉT**

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

..................................................................... ...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

*Ký tên*

......................................................................

*PGS. TS. Phan Văn Ca*

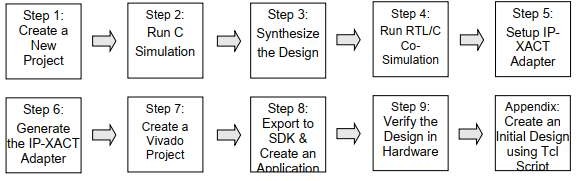
* 1. **Tổng quan lý thuyết và các bước thực hiện**

Thiết kế được sử dụng trong bài báo cáo này bao gồm bộ lọc FIR được dùng để lọc âm có cao độ 4 KHz được thêm vào âm nhạc chất lượng CD. Đặc tính của bộ lọc như sau:

- FS=48000 Hz

* FPASS1=2000 Hz
* FSTOP1=3800 Hz
* FSTOP2=4200 Hz
* FPASS2=6000 Hz
* APASS1=APASS2=1 dB
* ASTOP=60 dB

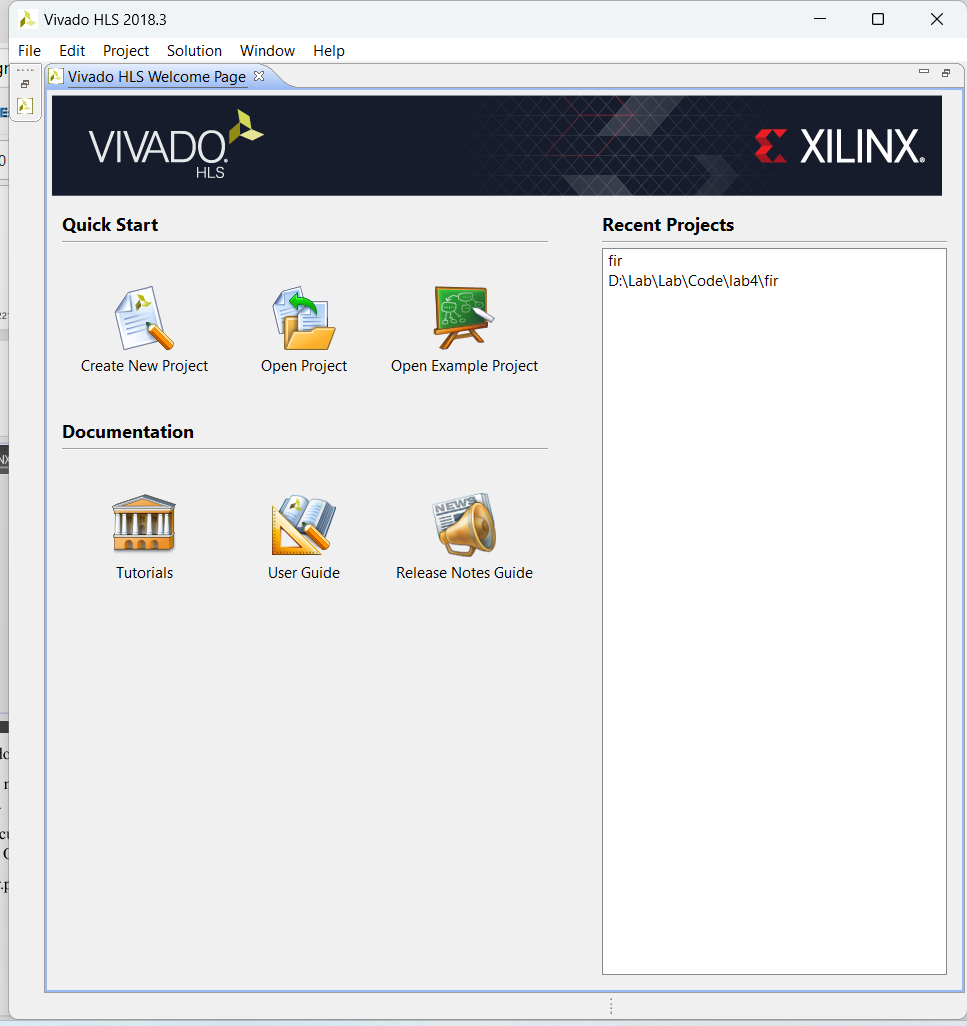
Sơ đồ tổng quan các bước thực hiện project với ZedBoard:



*Hình 1.1 Sơ đồ tổng quan*

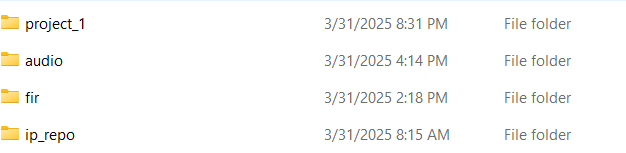
* 1. **Tạo một project mới**

Bước 1: Tạo một project mới trong Vivado HLS



*Hình 1.2. Vivado HLS tạo project*

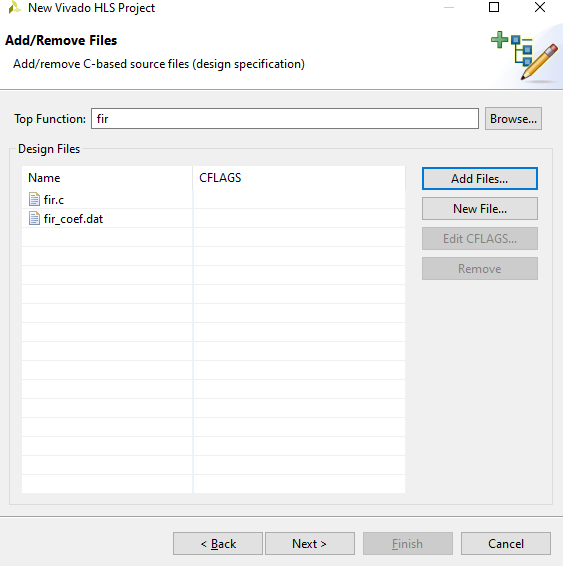
Bước 2: Tạo tên là fir (tùy vào người thiết kế) -> browns đúng vào thư mục \*/lab4



*Hình 1.3. fir trong lab4*

Bước 3: Trong mục Add/Remove Files cho các tệp nguồn, gõ fir là **tên hàm**

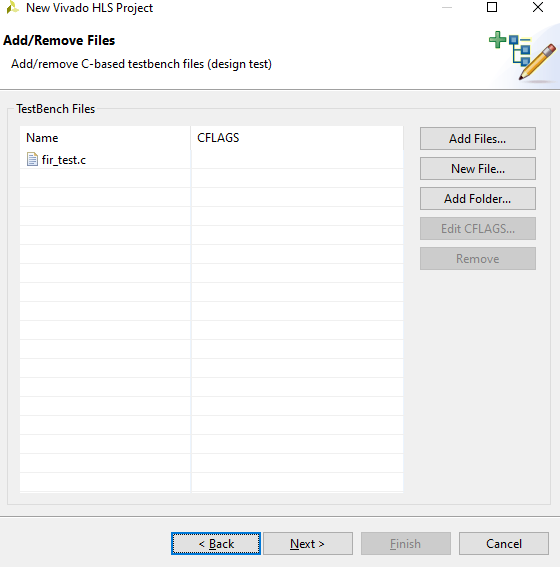
Bước 4: Nhấp vào nút Add Files..., chọn tệp **fir.c** và **fir\_coef.dat** từ thư mục \*/lab4, sau đó nhấp **Open**.



*Hình 1.4. Add/Remove Files cho các Source file*

Bước 5: Nhấp vào Next.

Bước 6: Trong mục Add/Remove Files cho testbench, nhấp vào nút Add Files..., chọn tệp **fir\_test.c** từ thư mục **\*/lab4** và nhấp **Open**.



*Hình 1.5. Add/Remove Files cho Testbench*

Bước 7: Nhấp vào Next.

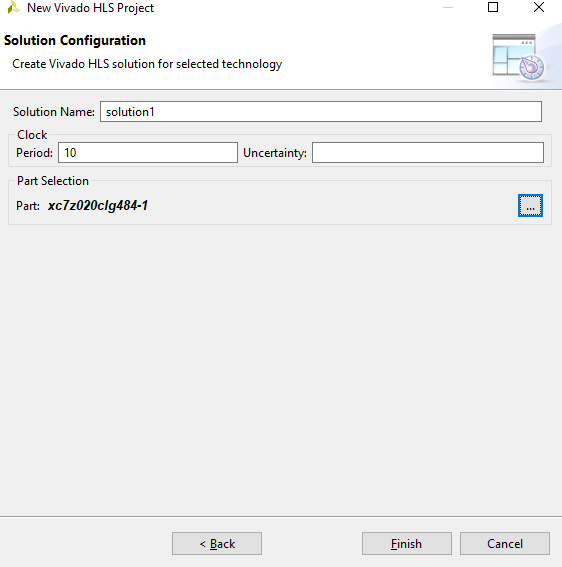
Bước 8: Trong trang Solution Configuration, để trường Solution Name là solution1 và đặt **chu kỳ clock** là 10 cho ZedBoard.

Bước 9: Nhấp vào nút **…** của **Part** và chọn các bộ lọc sau đây, sử dụng tùy chọn Parts Specify, để chọn **xc7z020clg484** sau đó nhấp vào OK:

Family: Zynq

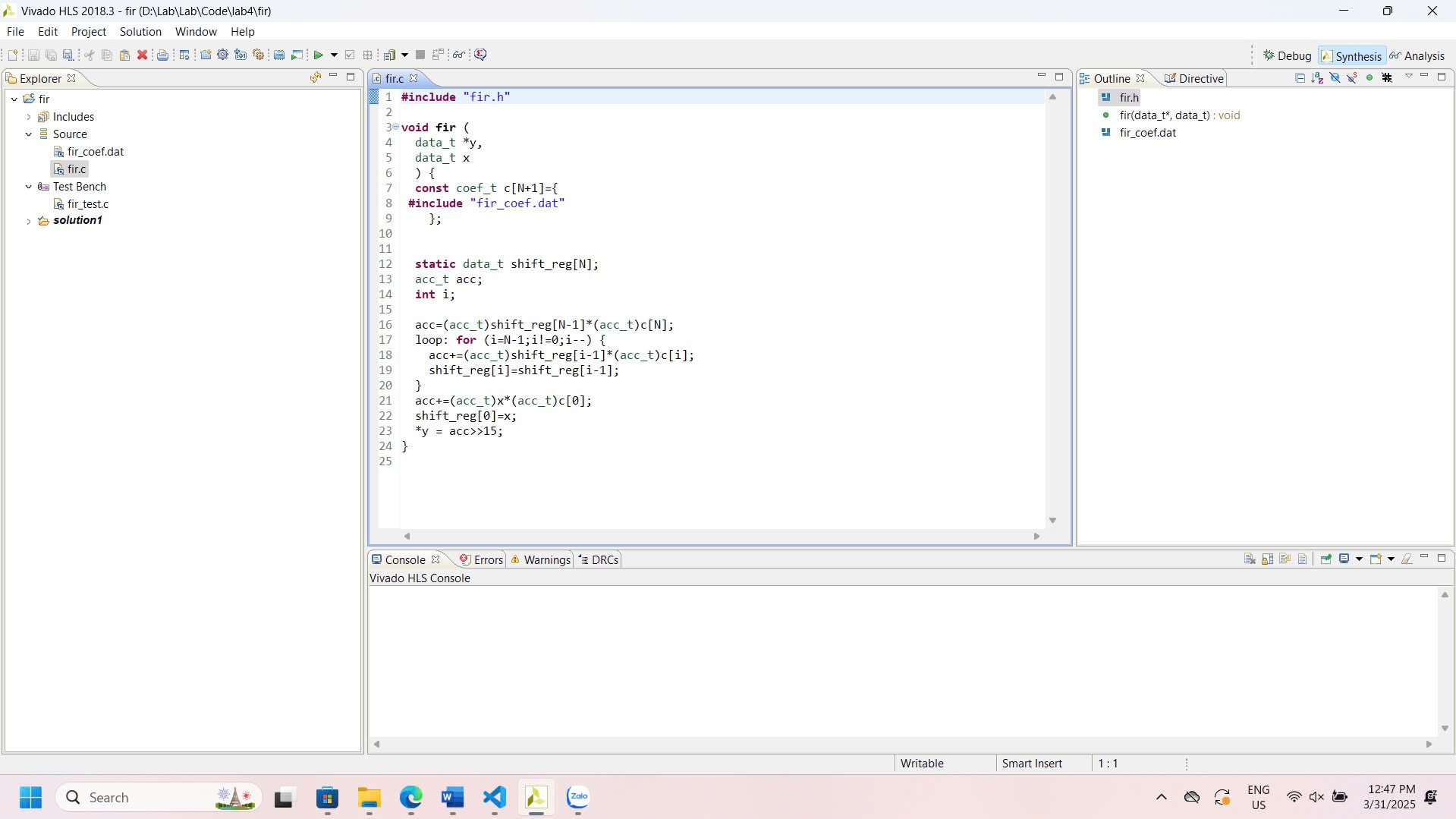
Sub-Family: Zynq

Package: clg484 (ZedBoard) hoặc clg400 (Zybo) Speed Grade: -1



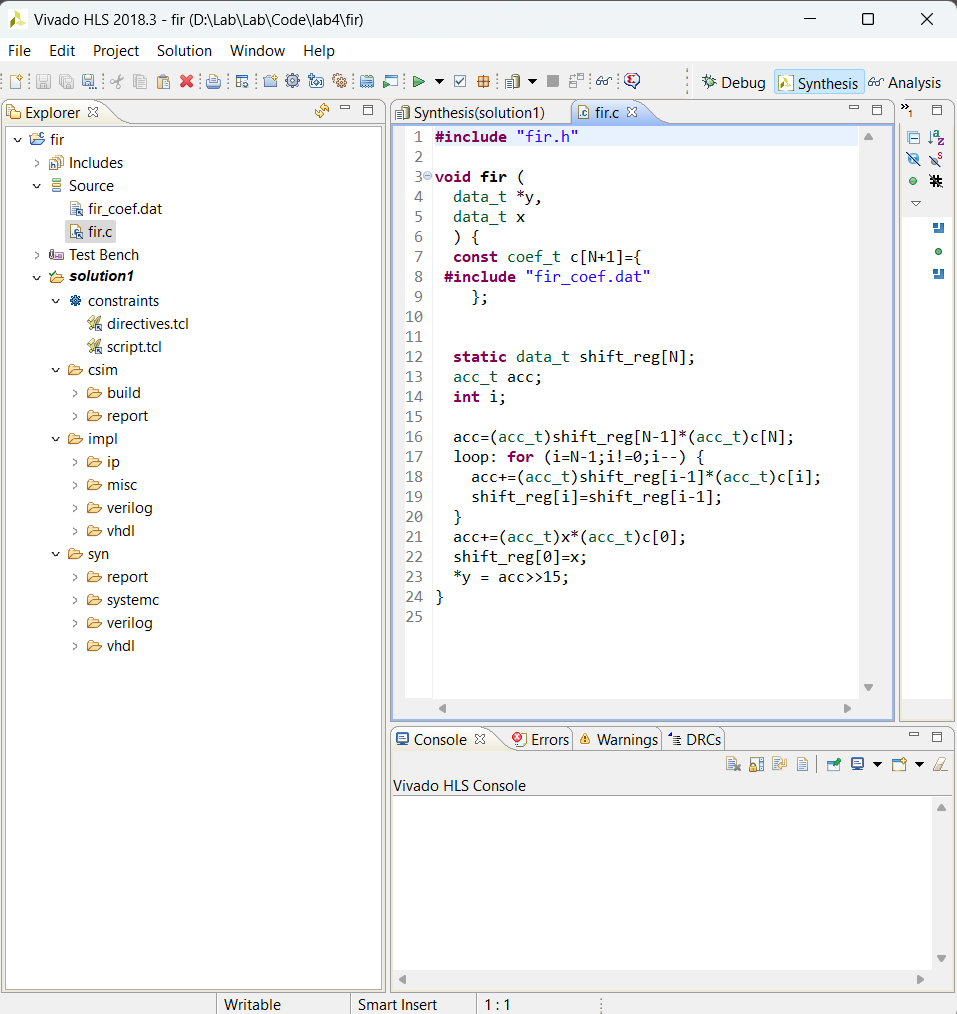
*Hình 1.6. Cấu hình*

Bước 10: Nhấp vào Finish.



*Hình 1.7. Cửa sổ làm việc Vivado HLS*

Bước 11: Nhấp đúp vào tệp **fir.c** trong thư mục nguồn để mở nội dung của nó trong bảng thông tin.



*Hình 1.8. Thiết kế của bộ lọc fir được sử dụng*

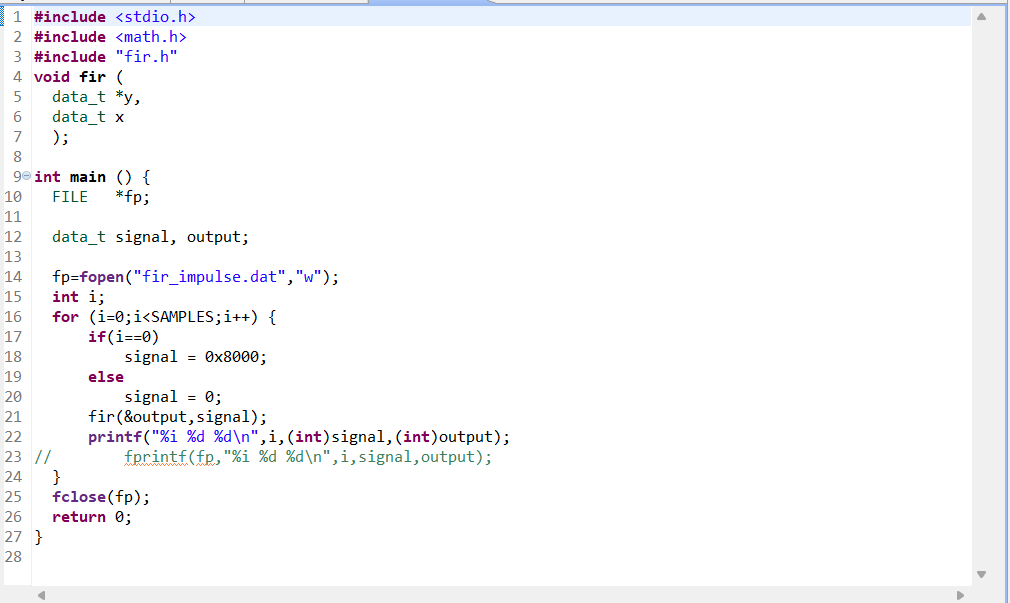
Bộ lọc FIR nhận một mẫu đầu vào **x**, sau đó nhân với các hệ số trong mảng **c** (đọc từ **fir\_coef.dat**) và cộng dồn vào biến **acc**. Kết quả cuối cùng được lưu vào **sample\_out**.

* **x**: Dữ liệu đầu vào (**data\_t**).
* **sample\_out**: Kết quả đầu ra (**data\_t**).
* **c**: Mảng chứa hệ số (**coef\_t**).
* **acc**: Biến tích lũy kết quả (**acc\_t**).

**Công thức:**

y[n]=∑c[i]×x[n−i]y[n]

Bước 12: Nhấp đúp vào tệp fir\_test.c trong thư mục testbench để mở nội dung của nó trong bảng thông tin.



*Hình 1.9. Nội dung file testbench*

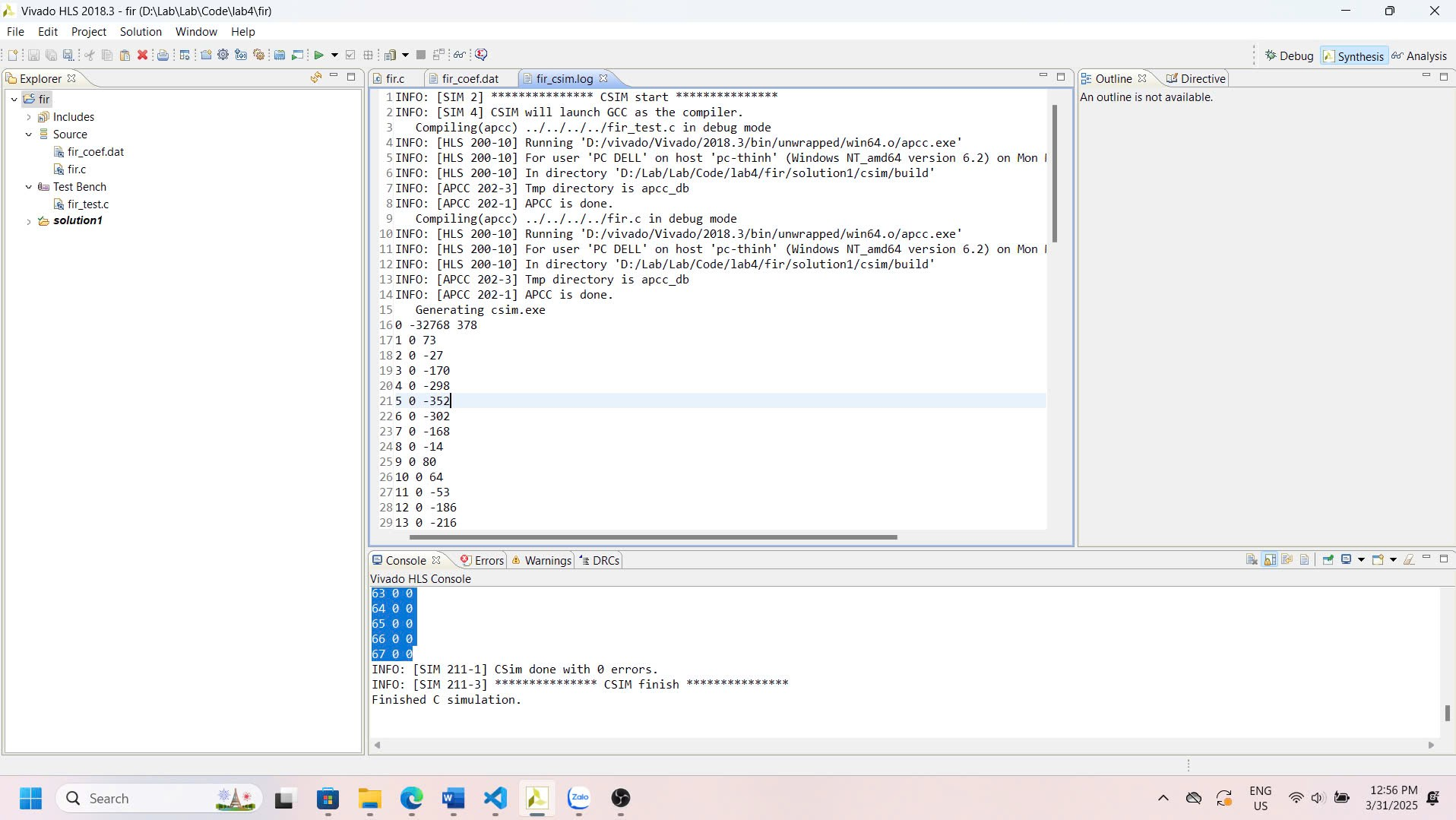
Lưu ý: testbench mở fir\_impulse.dat ở chế độ ghi và gửi một xung (mẫu đầu tiên là 0x8000).

* 1. **Chạy mô phỏng C**

Chạy mô phỏng C để quan sát đầu ra dự kiến.

Chọn Project > Run C Simulation.

Testbench sẽ được biên dịch bằng trình biên dịch apcc và tệp csim.exe sẽ được tạo ra. Sau đó, csim.exe sẽ được thực thi và đầu ra sẽ được hiển thị trong cửa sổ console view.



*Hình 1.10. Kết quả testbench được tạo ra*

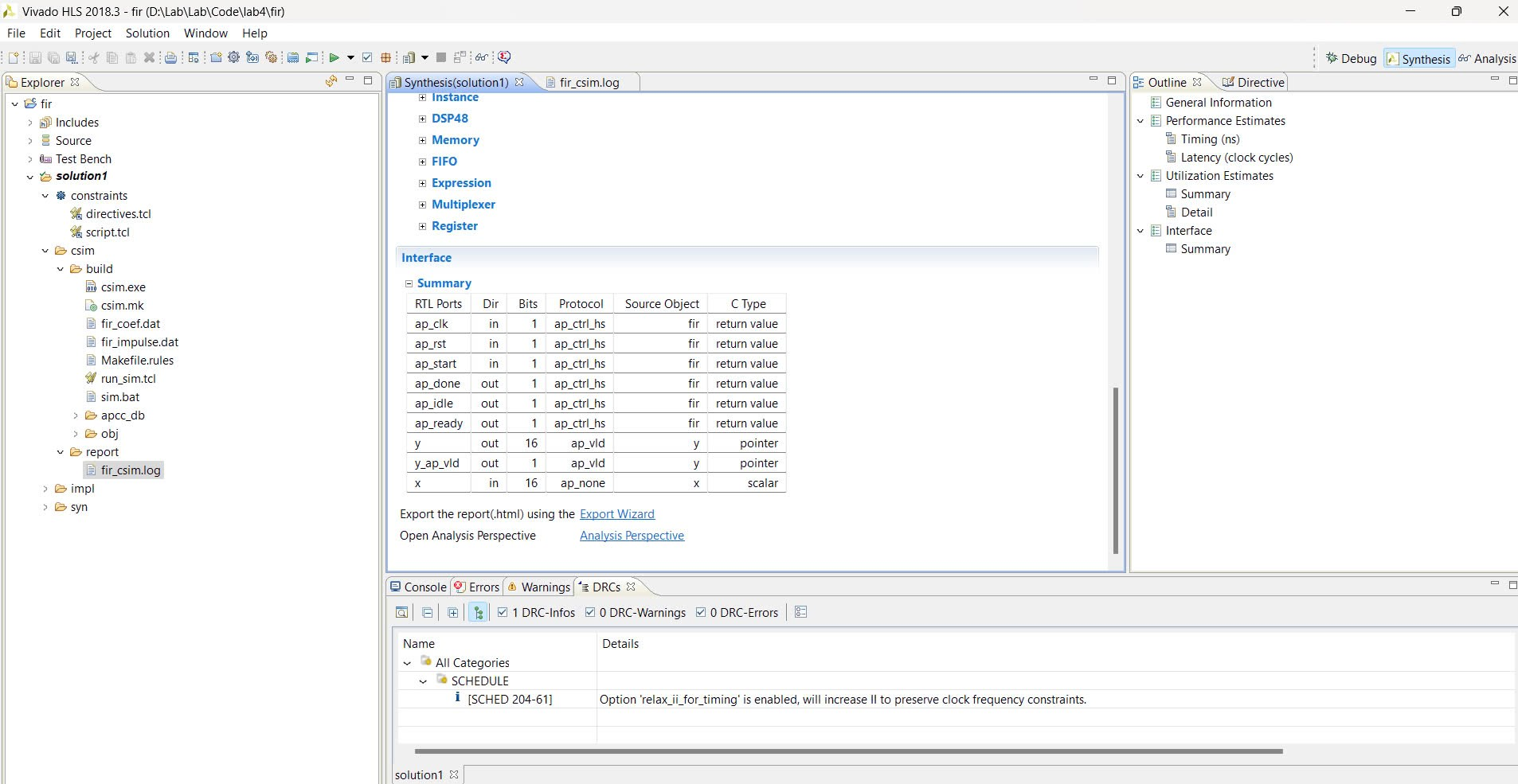
* 1. **Tổng hợp thiết kế**

Tổng hợp thiết kế với các giá trị mặc định. Xem kết quả tổng hợp và trả lời câu hỏi được liệt kê trong phần chi tiết của bước này.

Bước 1: Chọn Solution > Run C Synthesis > Active Solution để bắt đầu quá trình tổng hợp.

Bước 2: Khi tổng hợp hoàn thành, một số tệp báo cáo sẽ trở nên truy cập được và Kết quả Tổng hợp sẽ được hiển thị trong bảng thông tin.

Bước 3: Báo cáo Tổng hợp hiển thị ước tính hiệu suất và tài nguyên cũng như độ trễ ước tính trong thiết kế.



*Hình 1.11. Synthesis Report*

Bước 4: Báo cáo cũng hiển thị các tín hiệu interface cấp cao được tạo ra bởi các công cụ.

Bước 5: Chọn tab Directive và áp dụng chỉ thị PIPELINE vào vòng lặp.

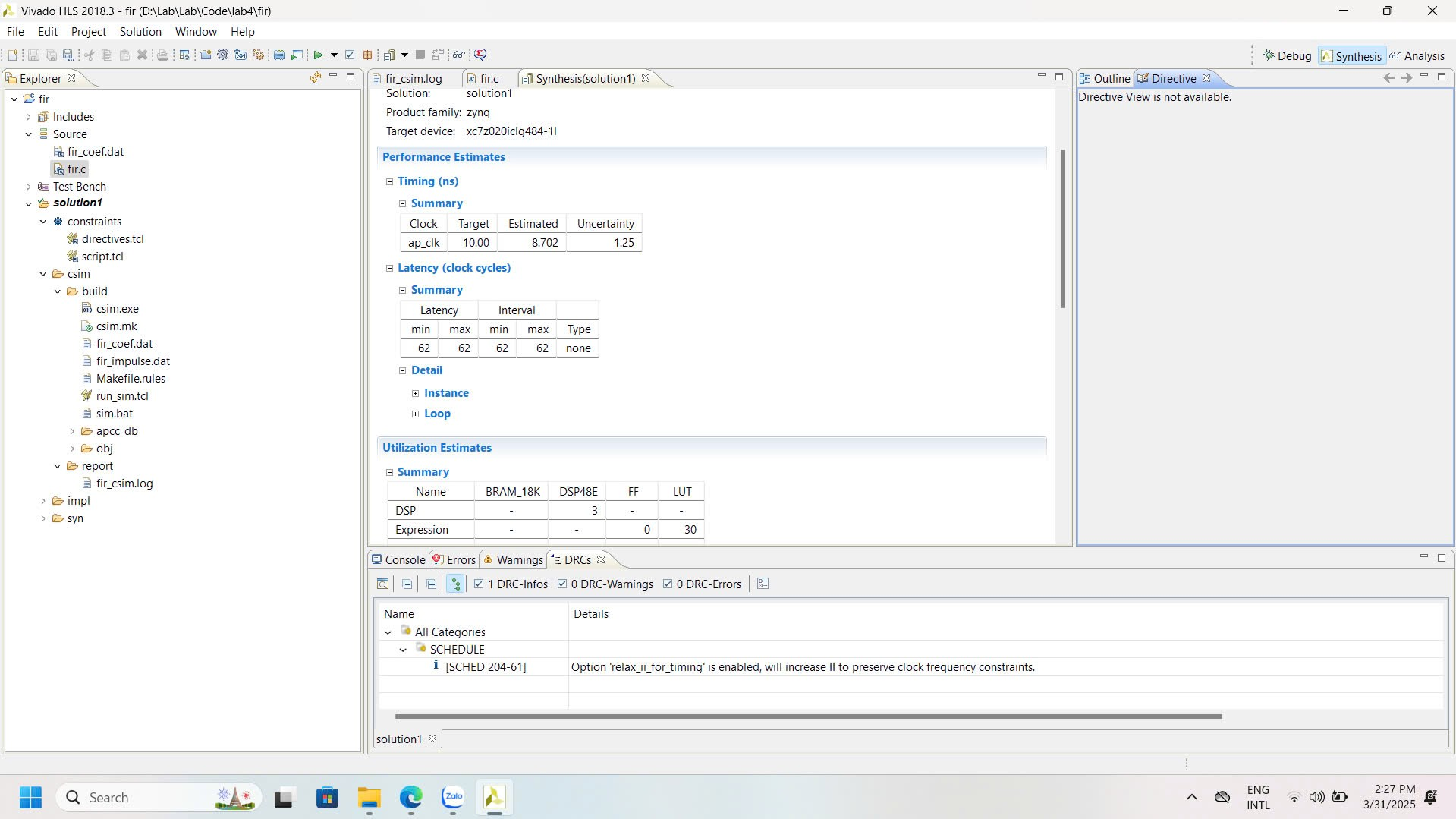


*Hình 1.12. Áp dụng PIPELINE vào vòng lặp*

Bước 6: Chọn Solution > Run C Synthesis > Active Solution để bắt đầu quá trình tổng hợp.

Bước 7: Khi tổng hợp hoàn thành, Kết quả Tổng hợp sẽ được hiển thị trong bảng thông tin.

Bước 8: Lưu ý rằng độ trễ đã giảm xuống 62 hoặc 63 chu kỳ xung. Tiêu thụ DSP48 và BRAM vẫn giữ nguyên; tuy nhiên, tiêu thụ LUT và FF tăng một chút.



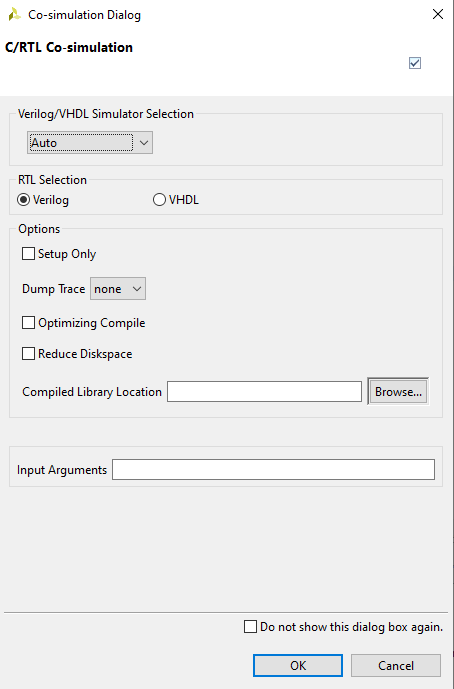
*Hình 1.13. Synthesis Report sau khi thêm PIPELINE vào vòng lặp*

* 1. **Chạy RTL/C Co-simulation**

Chạy RTL/C Co-simulation, chọn Verilog. Xác nhận rằng mô phỏng thành công

Bước 1: Chọn Solution -> Run C/RTL Co-simulation hoặc nhấp vào nút để mở hộp thoại, từ đó bạn có thể chạy các mô phỏng mong muốn.

Một hộp thoại C/RTL Co-simulation sẽ mở ra.



*Hình 1.14. Cửa sổ Co-simulation Dialog*

Bước 2: Chọn tùy chọn Verilog và nhấp OK.

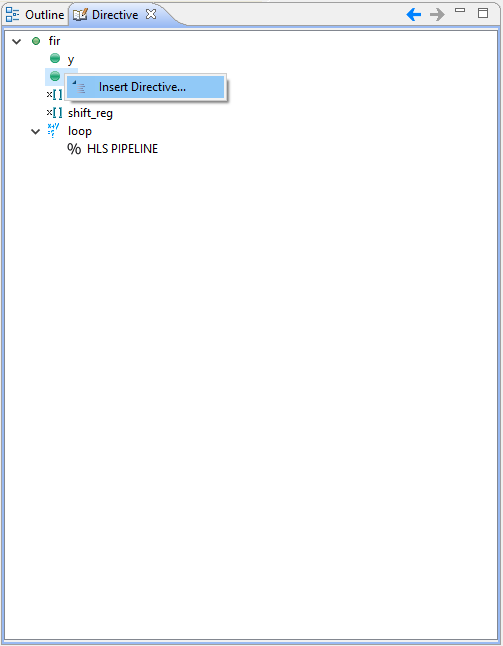
* 1. **Cấu hình Bộ chuyển đổi IP-XACT**

Thêm chỉ thị INTERFACE để tạo các bộ chuyển đổi AXI4LiteS để có thể tạo bộ chuyển đổi IP-XACT trong bước Xuất RTL.

Bước 1: Đảm bảo rằng tệp fir.c đang được mở và có trọng tâm trong cửa sổ thông tin (information view).

Bước 2: Chọn tab Directive (chỉ thị).

Bước 3: Nhấp chuột phải vào x và chọn Insert Directive... (Chèn chỉ thị...).



*Hình 1.15. Hình ảnh thực hiện*

Bước 4: Trong hộp thoại Vivado HLS Directive Editor, chọn INTERFACE bằng cách sử dụng nút thả xuống.

Bước 5: Nhấp vào nút bên cạnh mode (tùy chọn). Chọn s\_axilite. Bước 6: Trong trường bundle (tùy chọn), nhập fir\_io và nhấp OK

Bước 7: Tương tự, áp dụng chỉ thị INTERFACE (bao gồm bundle) vào đầu ra y.

Bước 8: Áp dụng chỉ thị INTERFACE vào module cấp cao fir để bao gồm các tín hiệu ap\_start, ap\_done và ap\_idle là một phần của bộ chuyển đổi bus (tên biến được hiển thị sẽ là return).

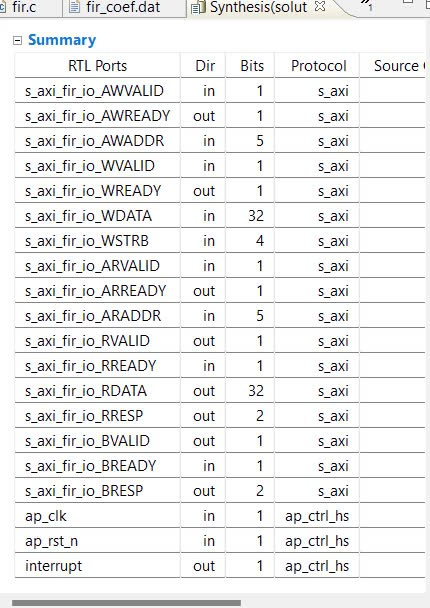
Lưu ý: các bước 3 đến 8 ở trên sẽ tạo ra các bản đồ địa chỉ cho x, y, ap\_start, ap\_valid, ap\_done và ap\_idle, có thể truy cập thông qua phần mềm. Ngoài ra, tín hiệu ap\_start, ap\_valid, ap\_done và ap\_idle có thể được tạo ra như các cổng riêng biệt trên lõi bằng cách không áp dụng chỉ thị RESOURCE vào module cấp cao fir. Các cổng này sau đó phải được kết nối trong hệ thống xử lý bằng cách sử dụng IP GPIO có sẵn.

* 1. **Tạo Bộ chuyển đổi IP-XACT Adapter**

Tổng hợp lại thiết kế vì đã thêm các chỉ thị. Chạy Export RTL để tạo bộ chuyển đổi IP-XACT

Bước 1: Sau khi các chỉ thị đã được thêm, bạn có thể tổng hợp lại thiết kế. Chọn Solution > Run C Synthesis > Active Solution

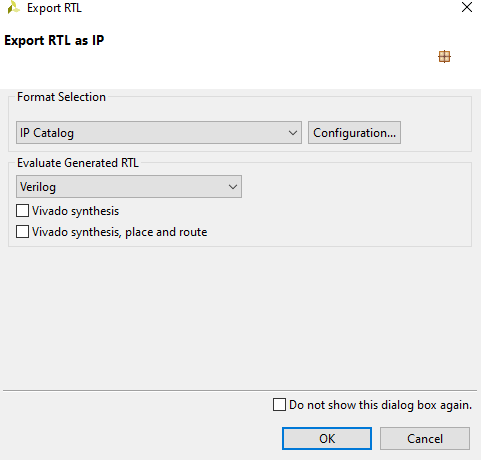
Kiểm tra Tóm tắt giao diện ở cuối báo cáo Tổng hợp để xem giao diện đã được tạo ra.



*Hình 1.16. Interface được tạo sau khi tổng hợp lại thiết kế*

Bước 2: Sau khi thiết kế đã được tổng hợp, chọn Solution > Export RTL để mở hộp thoại, từ đó IP mong muốn có thể được tạo ra.

Hộp thoại Export RTL sẽ mở ra.



*Hình 1.17. Hộp thoại Export RTL*

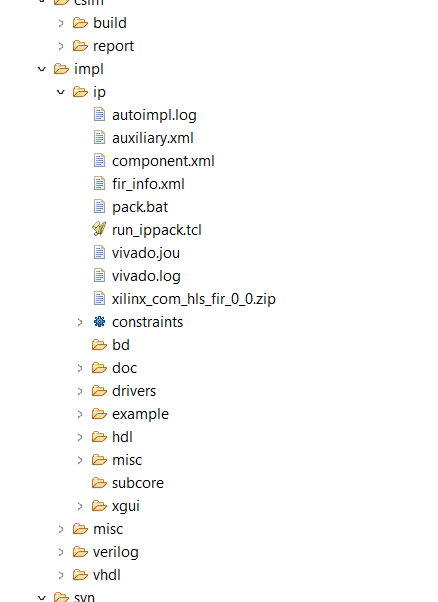
Bước 3: Nhấp OK để tạo bộ chuyển đổi IP-XACT.

Bước 4: Khi quá trình chạy hoàn thành, mở rộng thư mục impl trong chế độ xem Explorer và quan sát các thư mục được tạo ra; ip, misc, verilog và vhdl.



*Hình 1.17. Các thư mục được tạo ra*

Mở rộng thư mục IP và quan sát một số tập tin và thư mục con. Một trong các thư mục con mà chú ý là thư mục "drivers" bao gồm các tập tin header, c, tcl, mdd và makefile. Một tập tin khác đáng quan tâm là tập tin zip, đó là tập tin kho IP có thể được nhập vào một thiết kế IP Integrator.



*Hình 1.18. Thư mục drivers của Adapter*

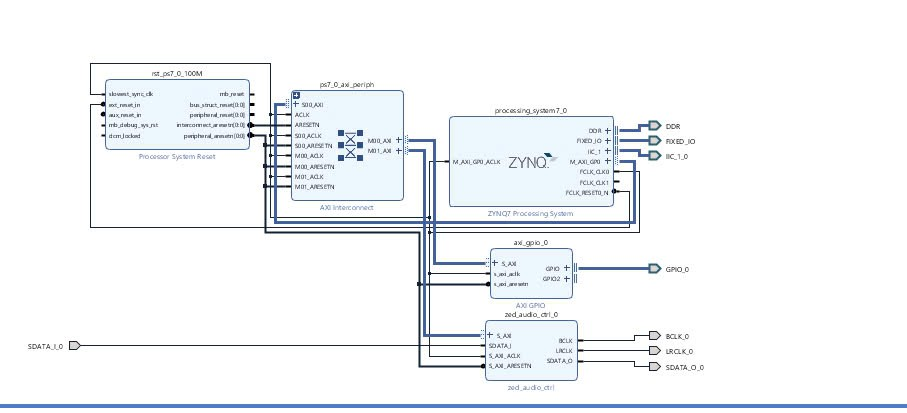
Bước 5: Đóng Vivado HLS bằng cách chọn File > Exit.

* 1. **Tạo Vivado Project**
     1. **Tạo hệ thống ban đầu cho Zedboard**

Ở bước này, ta sử dụng Vivado Tcl Shell và chạy tập lệnh tcl được cung cấp phía dưới để tạo hệ thống ban đầu cho Zedboard.

Bước 1: Mở Vivado Tcl Shell theo đường dẫn Start > All Programs > Xilinx Design Tools > Vivado 2018.3 > Vivado 2018.3 Tcl Shell.

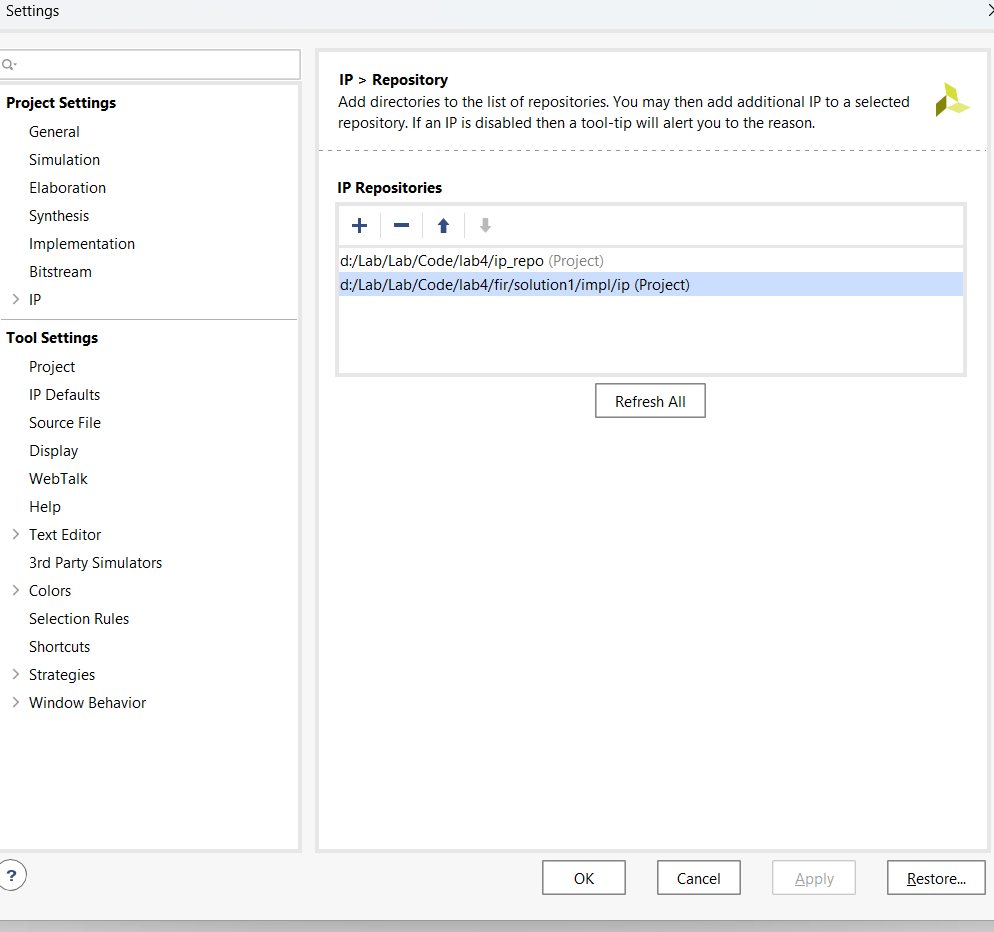
Bước 2: Chạy file script được cung cấp để tạo hệ thống ban đầu có các thiết bị ngoại vi zed\_audio\_ctrl và GPIO bằng lệnh: source zed\_audio\_project\_create.tcl.



*Hình 1.19. Thiết kế khối có zed\_audio\_ctrl và các kết nối được tạo cho ZedBoard*

* + 1. Thêm IP HLS vào danh sách IP

Bước 1: Trong phần Flow Navigator, nhấp vào Project Settings trong Project Manager. Cửa sổ Project Settings sẽ mở ra như hình 2.27.



*Hình 1.20. Cửa sổ Project Settings*

Ở cửa sổ Project Settings, chọn IP và thẻ Repository Manager. Nhấp nút ở IP Repositories, tìm đến đường dẫn: \*\labs\lab4\fir\solution1\impl\ip và nhấp Select.

* + 1. **Khởi tạo core fir\_top**

Bước 1: Nhấp vào biểu tượng Add IP và tìm Fir trong danh sách IP và double- click vào mục Fir để thêm IP vào sơ đồ khối như hình 2.29 (IP được thêm vào có logo HLS cho biết IP này được tạo bởi Vivado HLS).



*Hình 1.21. IP Fir được thêm vào sơ đồ khối*

Bước 2: Chọn IP đã được thêm vào sơ đồ khối và đổi tên thành fir\_left bằng cách chọn vào IP và nhập vào ô Name ở phần Block Properties phía bên trái.

Bước 3: Tượng tự, thêm vào sơ đồ khối một IP Fir khác và đổi tên thành fir\_right.

Bước 4: Nhấp vào Run Connection Automation và chọn All Automation. Và nhấp OK.

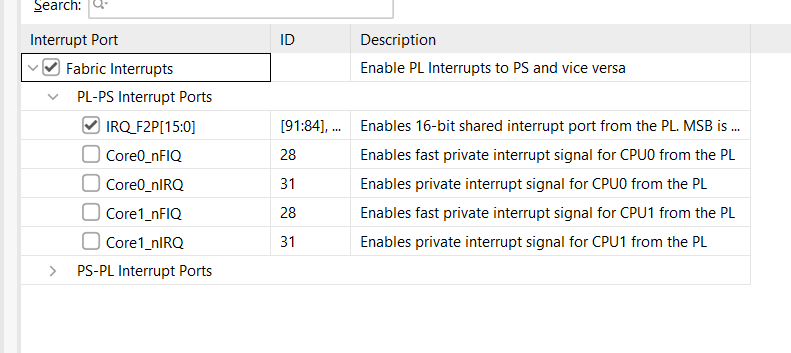
* + 1. **Mở các cổng PS-PL Interrupt**

Bước 1: Double-click vào processing\_system7\_0 để mở cửa sổ Re-customize IP như hình 2.30.



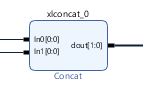
*Hình 1.26. Cửa sổ Re-customize IP*

Bước 2: Chọn Interrupts ở tab bên trái và chọn Fabric Interrupts như hình 2.31.



*Hình 1.27 PL-PS Interrupt Ports*

Bước 4 : Tiếp đến, thêm concat IP vào sơ đồ khối như hình 2.33.

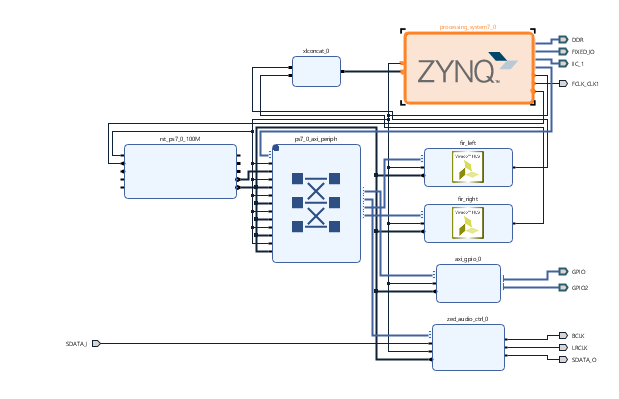


*Hình 1.28 Concat IP*

Bước 5 : Kết nối ngõ ra interrupt của mỗi Fir IP với hai ngõ vào của Concat IP.

Bước 6 : Kết nối ngõ ra của Concat IP với ngõ vào IRQ\_F2P[15:0] vừa tạo ở bước 3 của processing\_system7\_0.

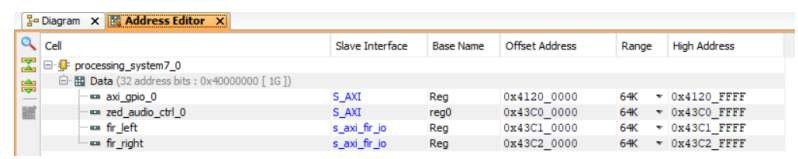
Sau bước 6, sơ đồ khối thiết kế sẽ giống như hình 2.34.



*Hình 1.29. Thiết kế phần cứng hoàn chỉnh dành cho Zedboard*

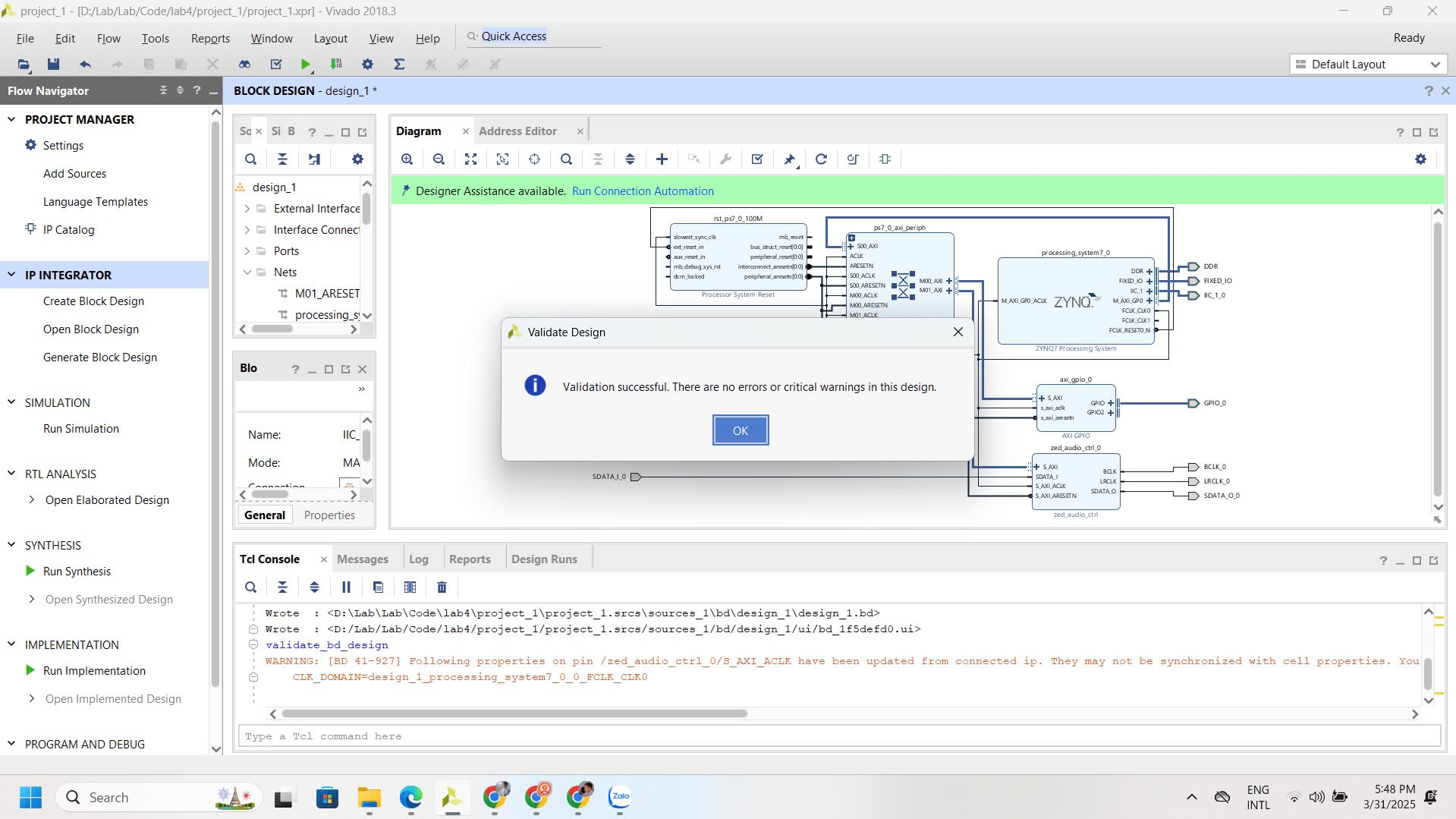
* + 1. **Kiểm tra địa chỉ và đánh giá thiết kế.**

Bước 1: Nhấp vào Address Editor, mở rộng processing\_system7\_0 > Data. Address Map được tạo ra sẽ giống như hình 2.35.

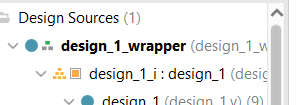


*Hình 1.30 Address Map được tạo ra cho Zedboard*

Bước 2: Chạy Design Validation theo Tools > Validate Design và kiểm tra thiết kế không có lỗi như hình 2.36.



*Hình 1.31 Thông báo đánh giá thiết kế thành công*



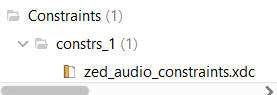
*Hình 1.32. Tạo file HDL wrapper*

Bước 4: Trong phần Flow Navigator, nhấp vào Add Sources trong Project Manager. Chọn Add or Create Constraints trong cửa sổ Add Sources như hình 2.38 và nhấp Next để tiếp tục.



*Hình 1.33. Cửa sổ Add Sources*

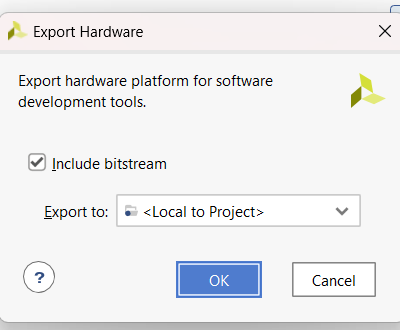
Bước 5: Nhấp vào nút Add và chọn Add Files, tìm đến đường dẫn C:\xup\hls\labs\lab4, chọn file zed\_audio\_constraints.xdc. Chọn tùy chọn Copy constraints files into project như hình 2.39 và nhấp Finish để thêm file vào project.



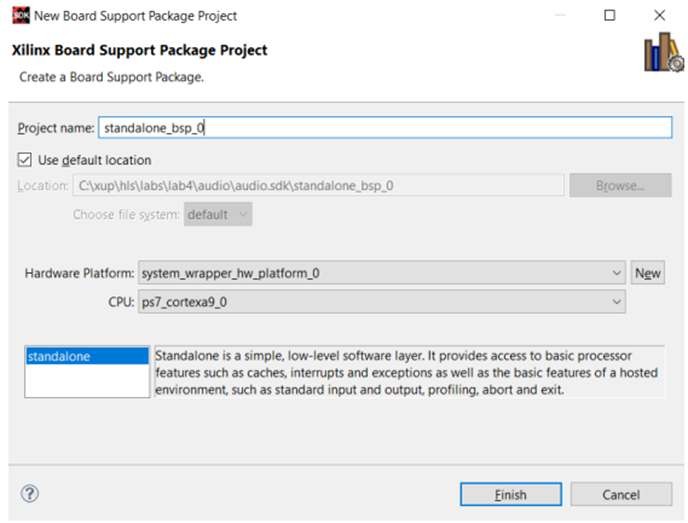
*Hình 1.34. Thêm file constraint vào project*

Bước 6: Trong phần Flow Navigator, nhấp vào Generate Bitstream trong Program and Debug để chạy các quá trình tổng hợp, triển khai và tạo bitstream. Khi quá trình tạo bitstream hoàn tất, sẽ có cửa sổ Bitstream Generation Completed mở ra như hình 2.40, nhấp Cancel để tiếp tục.

* 1. **Xuất thiết kế phần cứng sang SDK và tạo Application Project**
     1. **Xuất thiết kế phần cứng với bitstream đã tạo sang phần mềm SDK**



*Hình 1.35 Cửa sổ Export Hardware*



*1.36. New Board Support Package Project*

Tiếp theo, nhấp OK để chấp nhận các cài đặt mặc định vì ta muốn tạo project nền tảng phần mềm standalone\_bsp\_0 không yêu cầu hỗ trợ thư viện.

* Project Name: TestApp
* Board Support Package: Use Existing (standalone\_bsp\_0)

