BÀI 1: GIỚI THIỆU

1. Mục tiêu

Thông qua bài này, sinh viên cần nắm rõ:

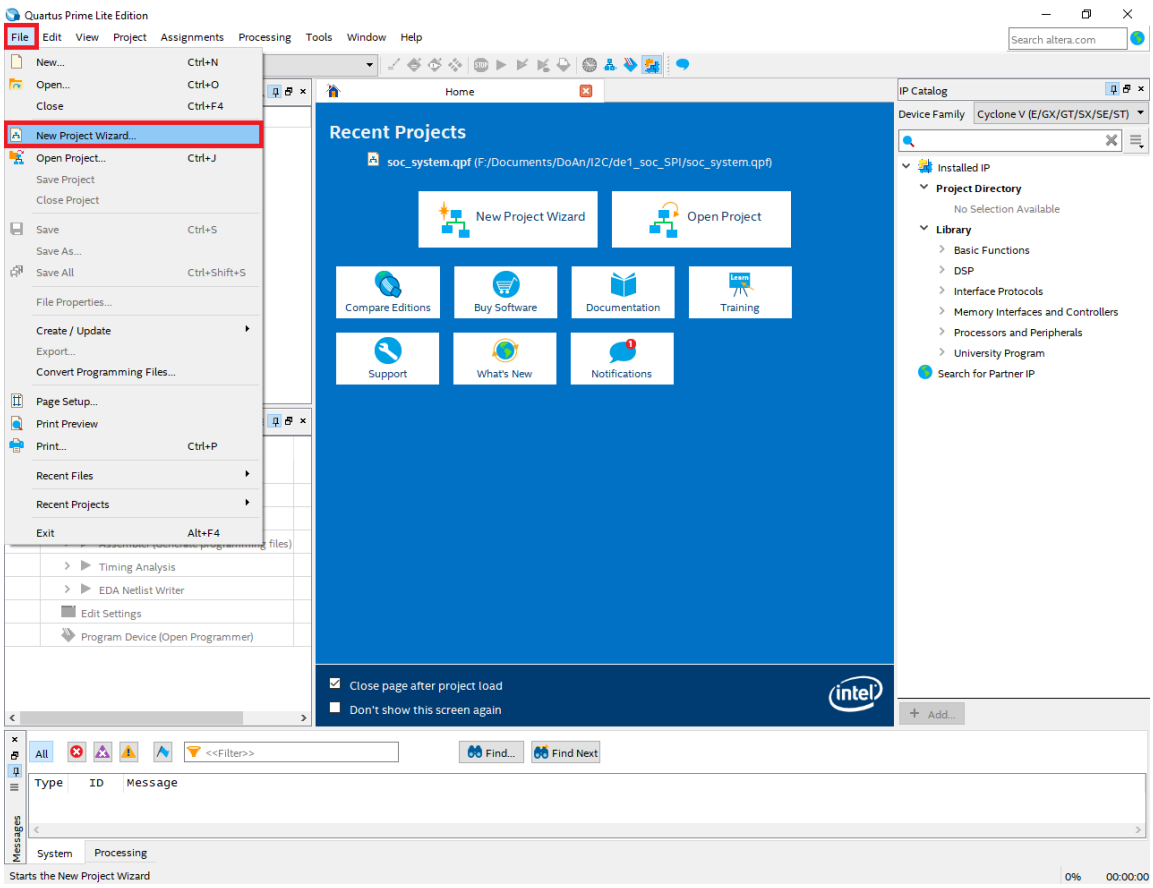
- Quy trình thiết kế hệ thống SoC gồm cả phần cứng và phần mềm.
- Cách sử dụng công cụ Quartus Prime và Platform Designer để tạo phần cứng máy tính, tổng hợp và nạp xuống FPGA.
- Cách sử dụng công cụ NIOS II để tổng hợp và nạp phần mềm xuống máy tính.

2. Nội dung thực hành

2.1. Hệ thống phần cứng

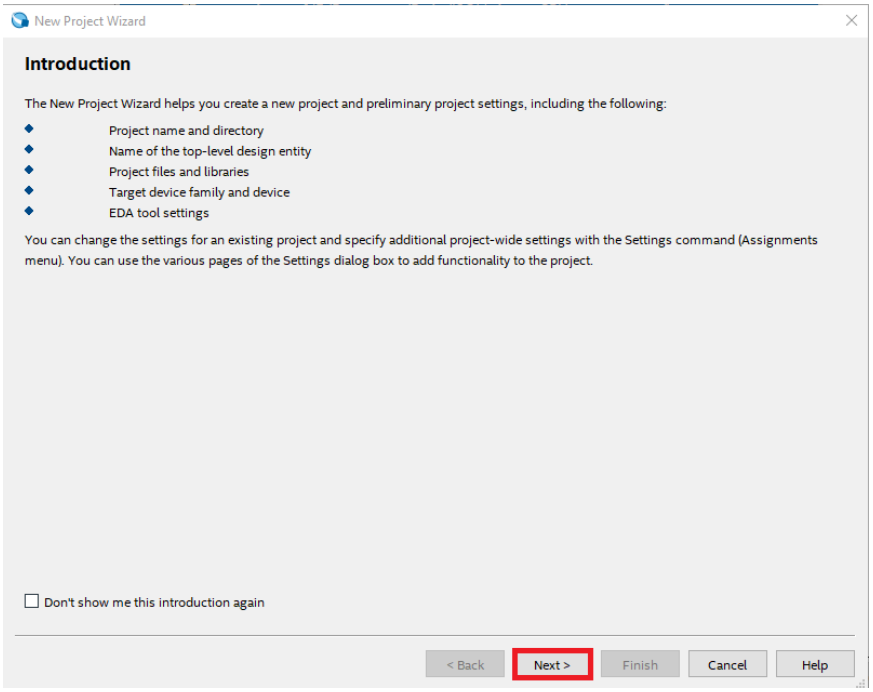
2.1.1. Tạo project Quartus

- Tạo thư mục ở ổ đĩa bất kì và đặt tên là **Bai1**.
- Tại cửa sổ giao diện của Quartus Prime phiên bản 18.1, hình 1, chọn **File → New Project Wizard…**



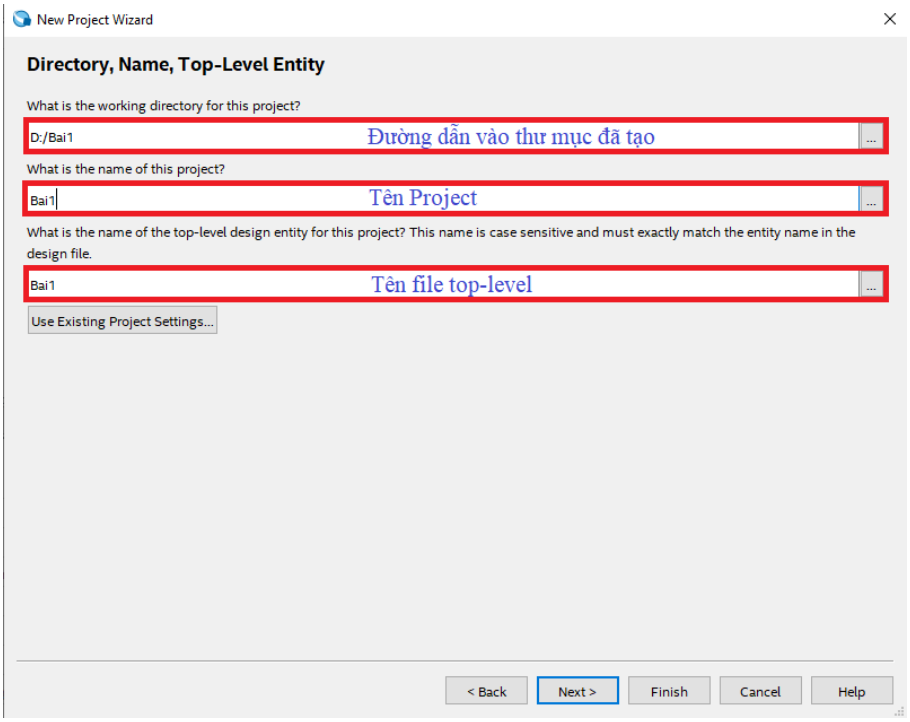
Hình 1. Giao diện phần mềm Quartus Prime.

- Sau khi chọn như hình 1, một cửa sổ New Project Wizard hiện lên, tiếp tục chọn **Next** như hình 2.



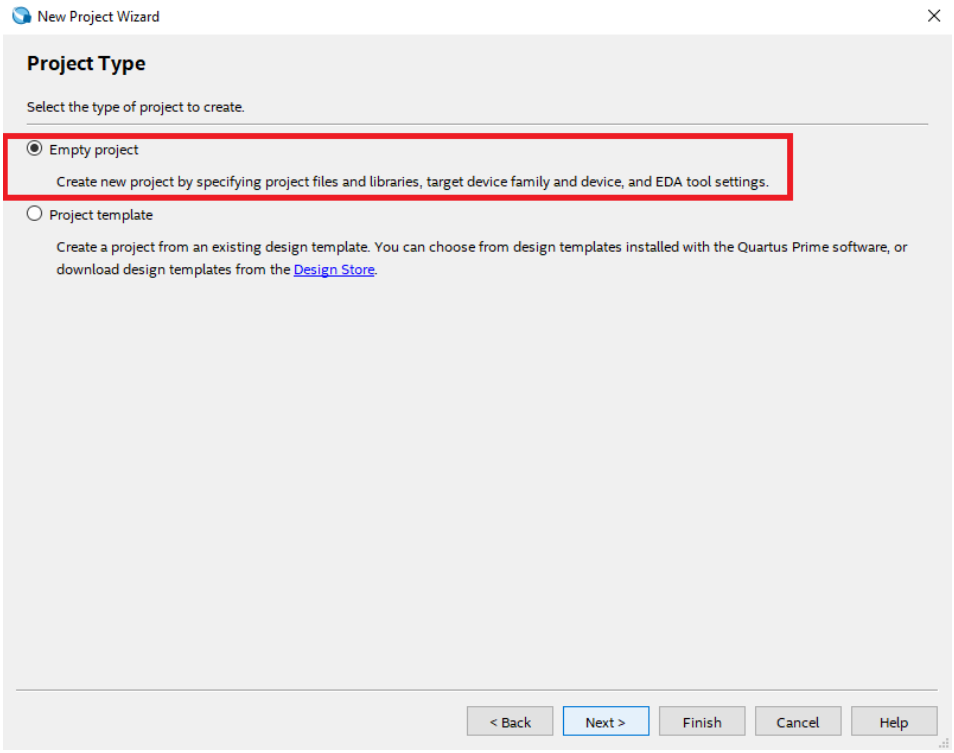
Hình 2. Tab Introduction.

- Ở tab **Directory, Name , Top-Level Entity**, chọn đường dẫn lưu project vào thư mục đã tạo trước đó và đặt tên như hình 3.



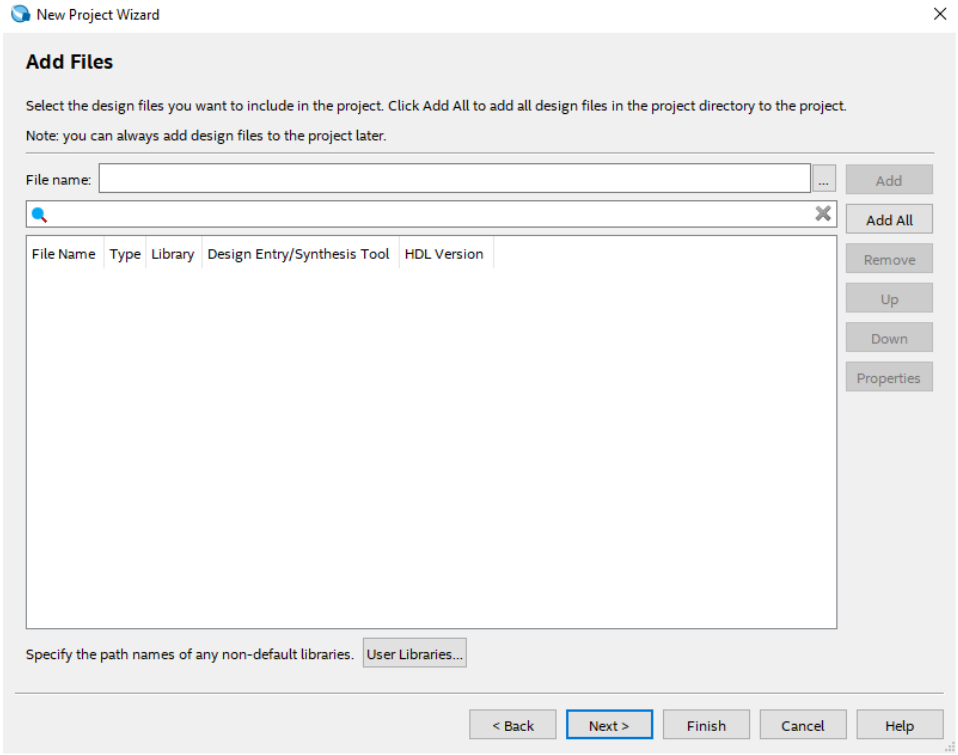
Hình 3. Tab Directory, Name, Top-Level Entity.

- Ở tab **Project Type**, chọn Empty project như hình 4 rồi nhấn **Next**.



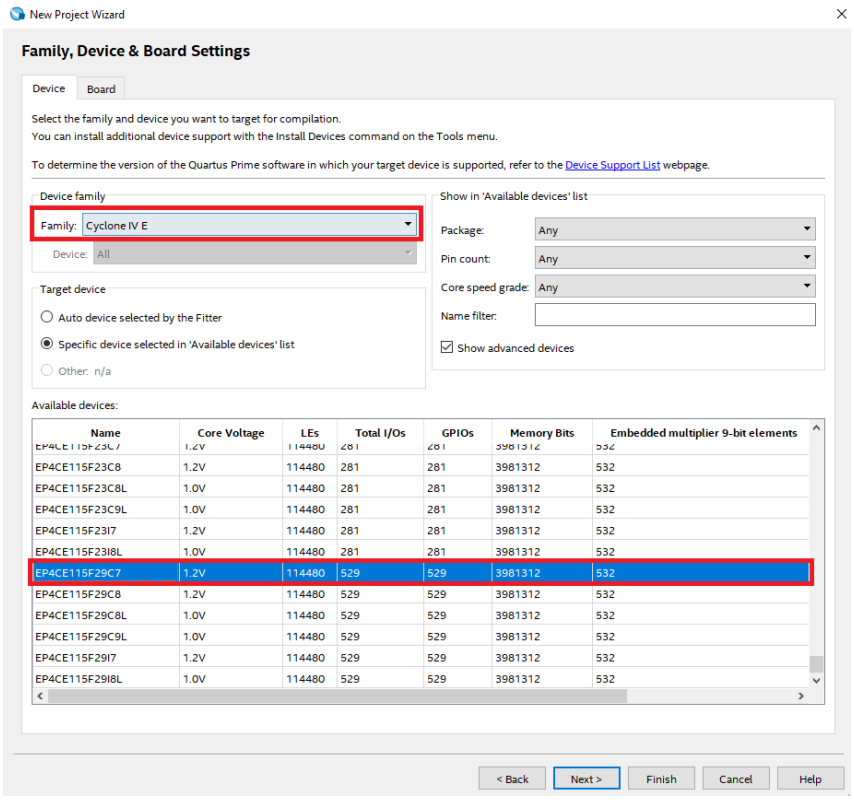
Hình 4. Tab Project Type.

- Ở tab **Add Files**, chọn **Next** như hình 5.



Hình 5. Tab Add Files.

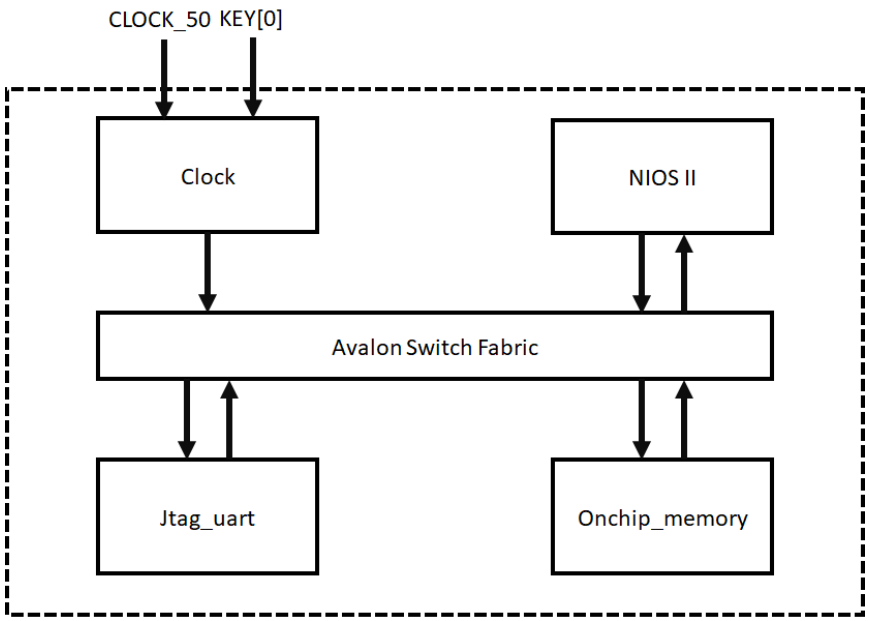
- Ở tab **Family, Device và Board Settings**, chọn Family là ***Cyclone IV E*** và device name là ***EP4E11529C7*** như hình 6. Cuối cùng nhấn Finish.



Hình 6. Tab Family, Device & Board Settings.

2.1.2. Xây dựng hệ thống phần cứng sử dụng Platform Designer.

a. Tổng quan hệ thống

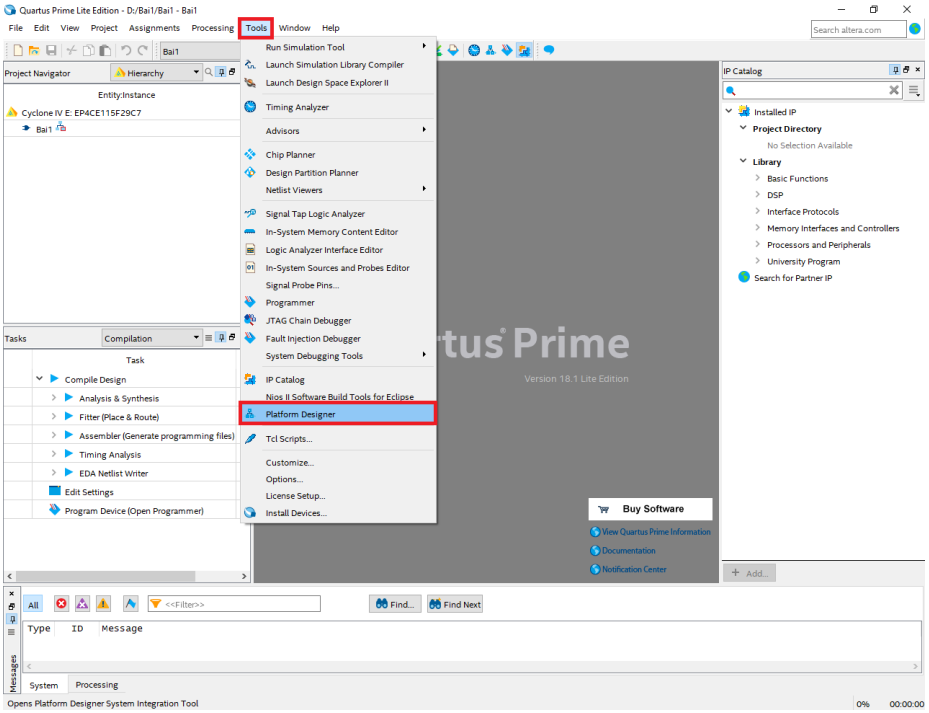


Hình 7. Tổng quan hệ thống.

- Hệ thống phần cứng xây dựng là một máy tính đơn giản, với các thành phần được thể hiện trong hình 7. Trong đó thành phần quan trọng nhất là NIOS II đóng vai trò là CPU và Onchip_memory là bộ nhớ để lưu trữ dữ liệu của hệ thống.

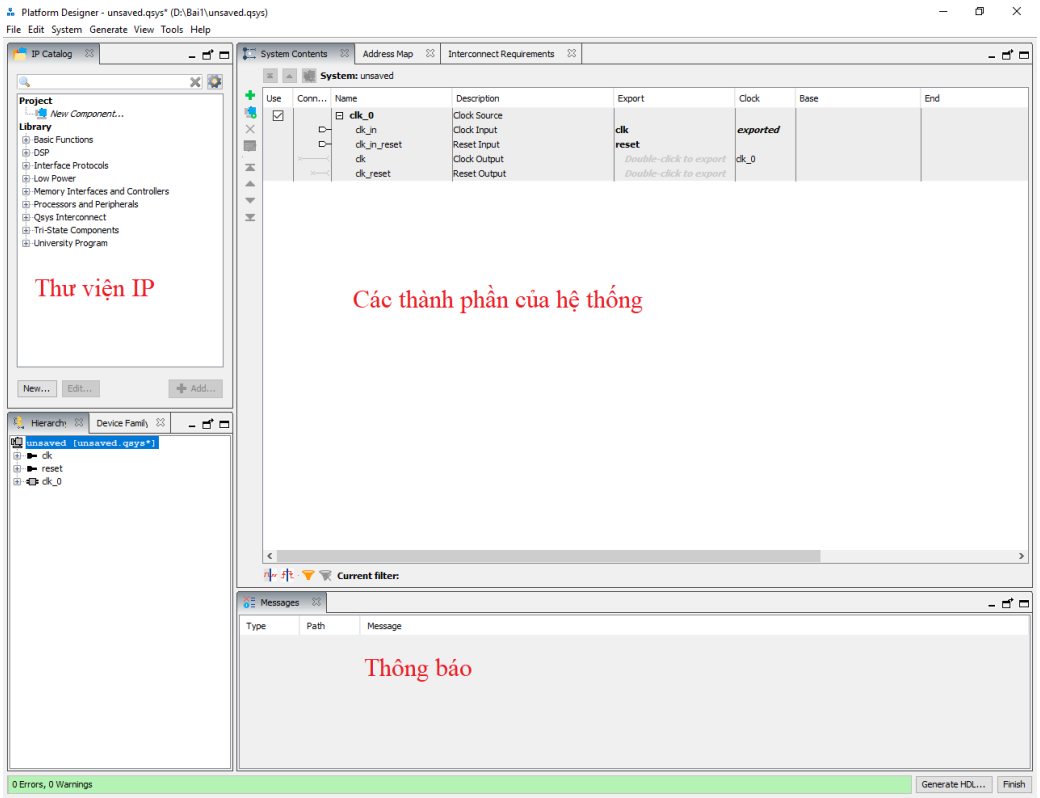
b. Tiến hành xây dựng hệ thống trên Platform Designer

- Ở cửa số 3 của phần mềm Quartus Prime, chọn **Tools → Platform Designer** như hình 8.



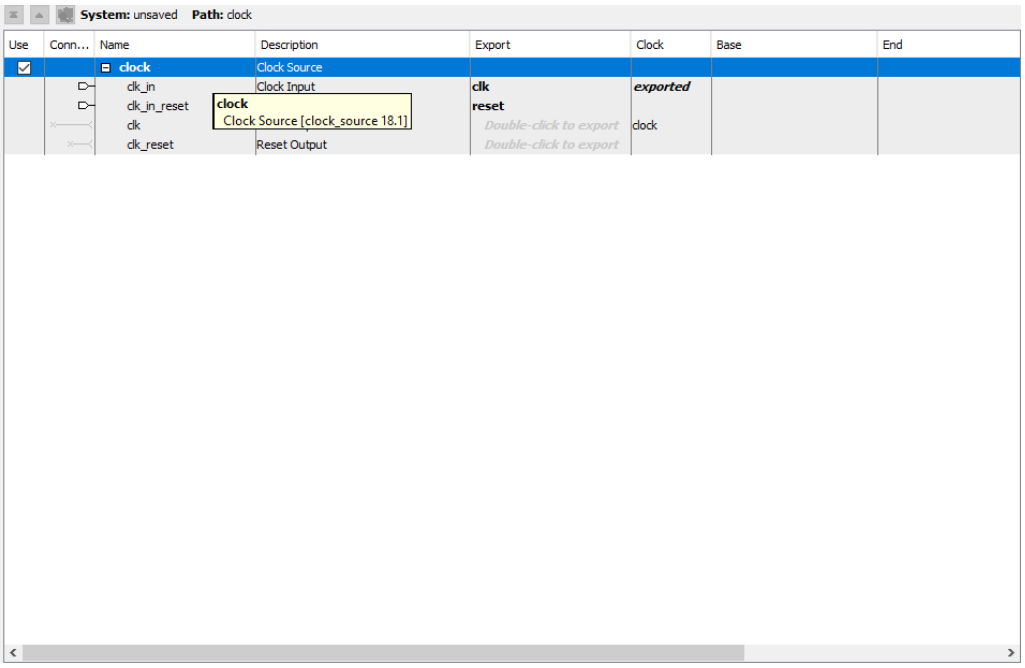
Hình 8. Mở Platform Designer.

- Sau khi mở phần mềm Platform Designer, sẽ có được cửa sổ như hình 9.



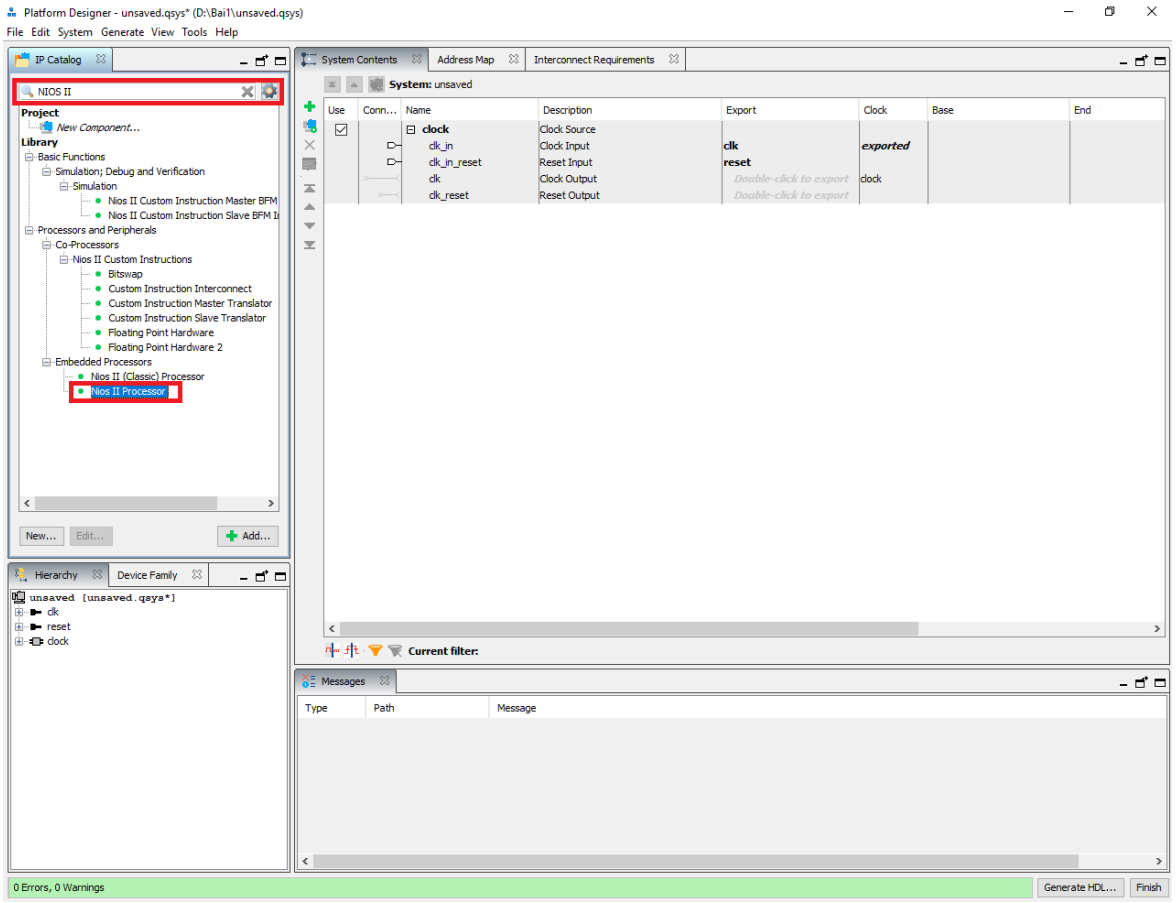
Hình 9. Giao diện phần mềm Platform Designer.

- Trong đó, ở tab System contents đã có thành phần Clock trong hệ thống, tiến hành đổ tên clk_0 thành Clock. (Bấm chuột phải vào IP clk_0 và chọn Rename). Sau khi đổi tên thành công, sẽ được như hình 10.



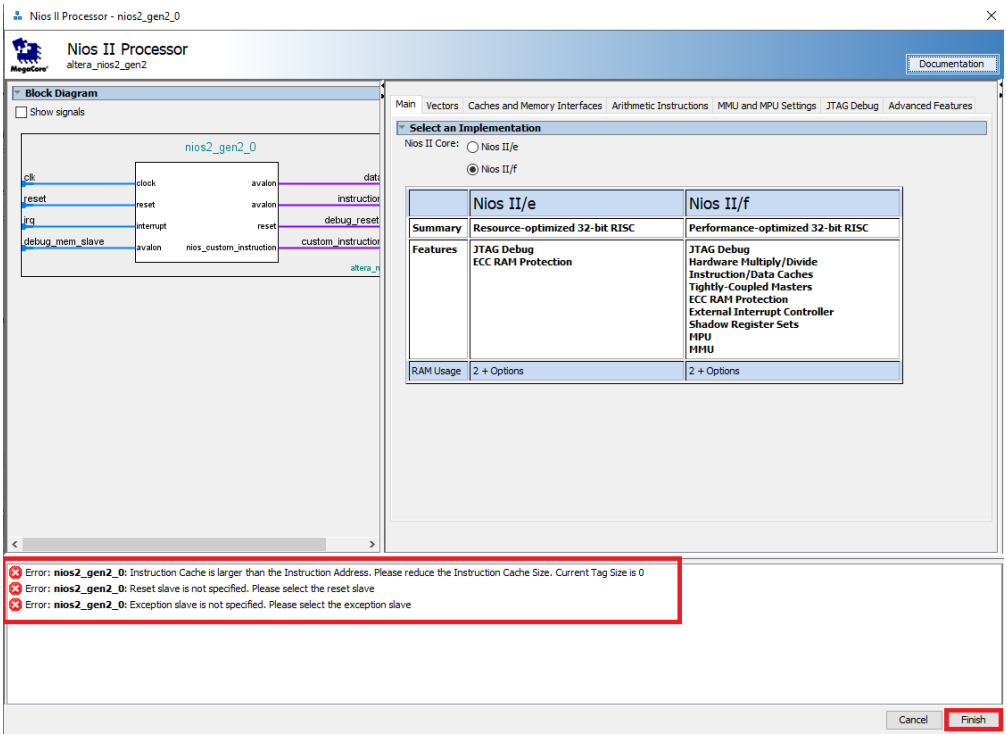
Hình 10. Tab System Contents.

- Tiếp theo, thêm CPU NIOS II bằng cách gõ NIOS II vào khung tìm kiếm và chọn như hình 11. Thêm vào hệ thống bằng cách bấm đúp chuột trái.



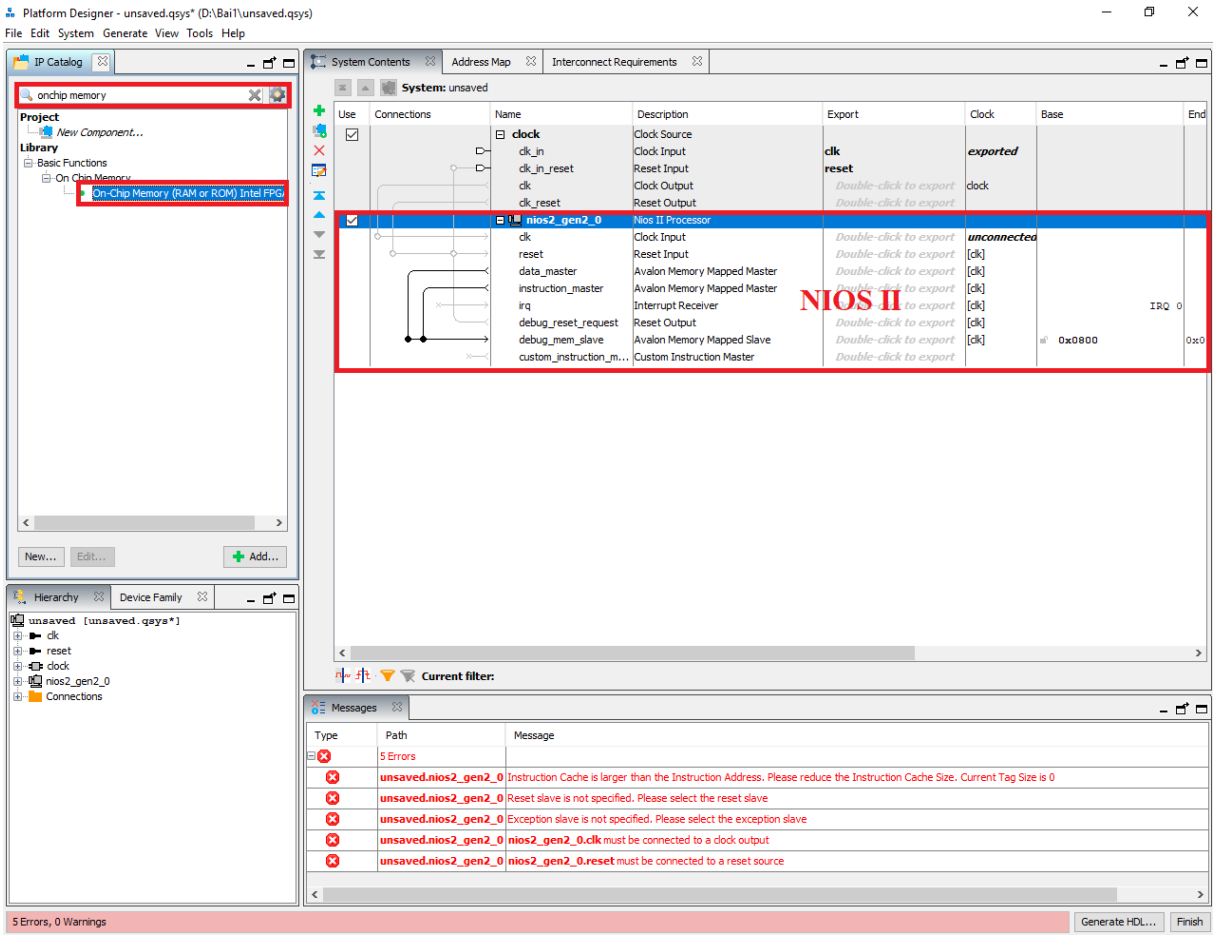
Hình 11. Thêm Nios II.

- Sau khi thêm vào hệ thống, sẽ có một hợp thoại mở ra như hình 12. Trong hình 12 sẽ hiện lỗi, nhưng sinh viên không quan tâm và bấm Finish, việc sửa lỗi sẽ thực hiện sau.



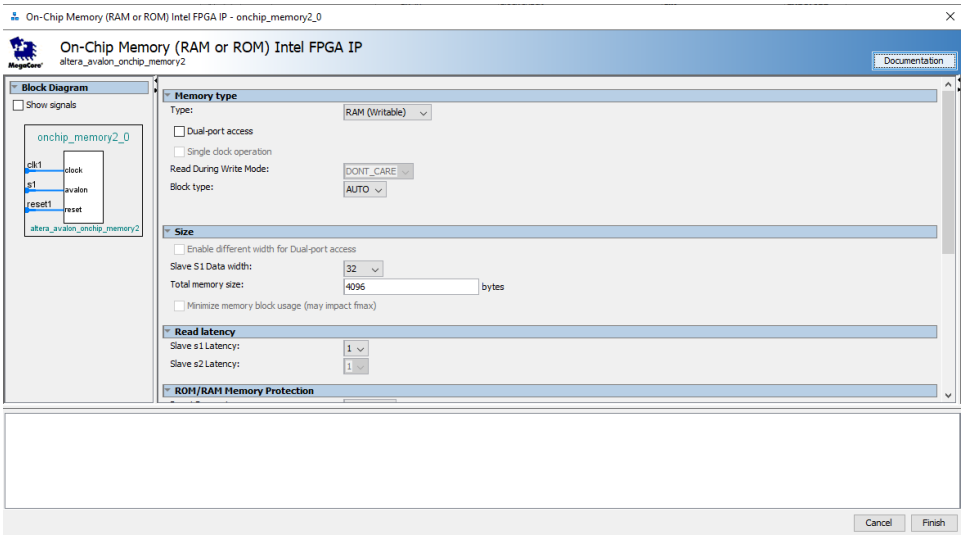
Hình 12. Cửa sổ cấu hình Nios II.

- Tiếp theo, thêm Onchip_memory bằng cách gõ onchip memory vào ô tìm kiếm và chọn như hình 13. Ngoài ra, ở hình 13, bên tab **System Contents** sẽ thấy sự xuất hiện của NIOS II đã thêm trước đó.



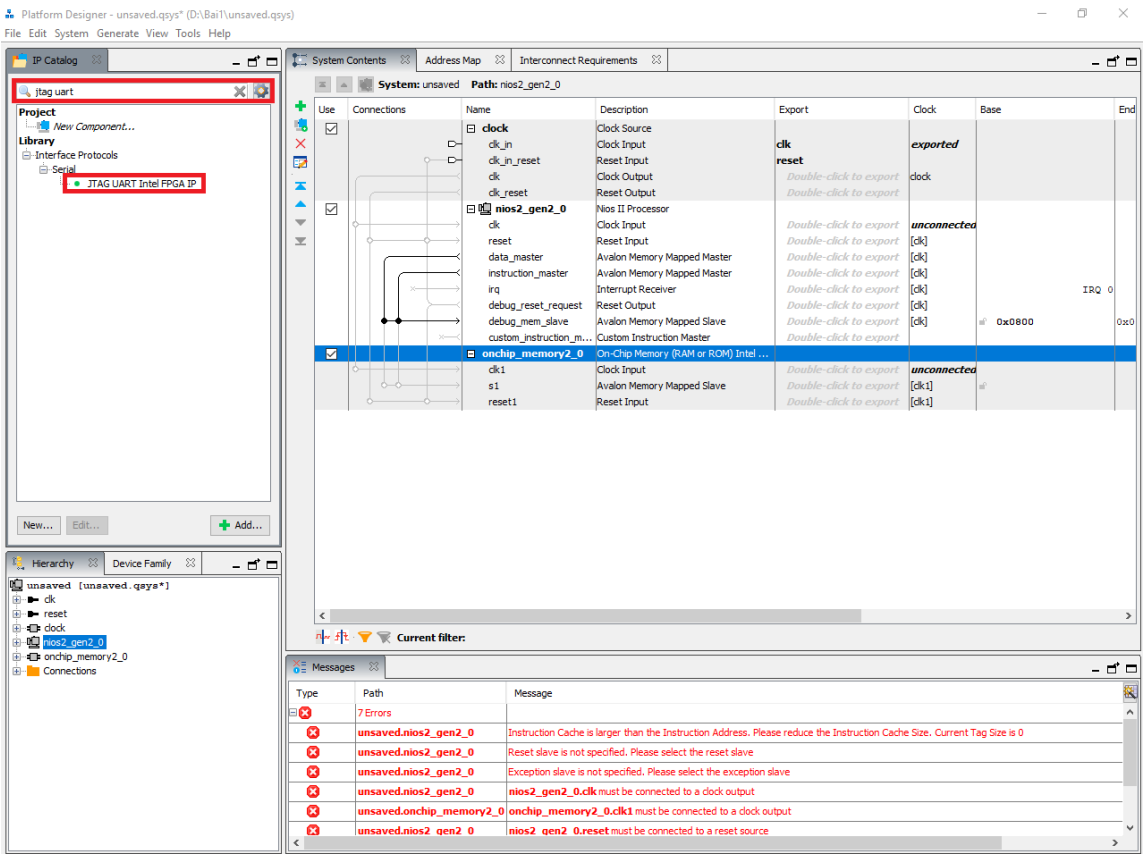
Hình 13. Thêm Onchip-Memory.

- Sau khi thêm onchip memory, cửa sổ hiện ra như hình 14.

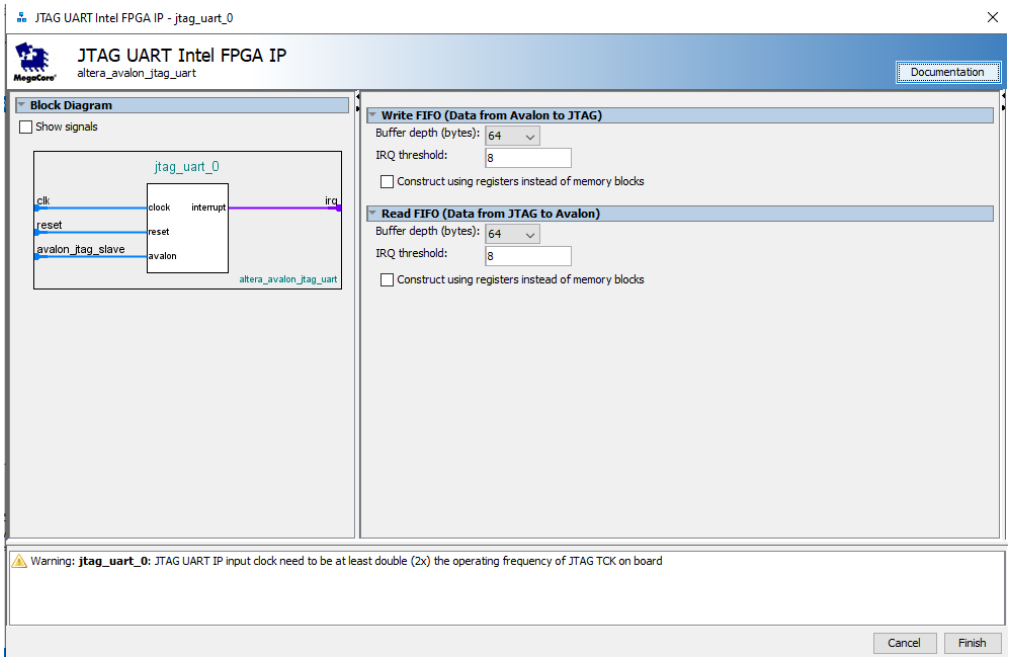


Hình 14. Cửa sổ cấu hình On-Chip Memory.

- Tương tự, thêm thành phần jtag-uart như hình 15 và cửa sổ của mô đun jtag-uart như hình 15.

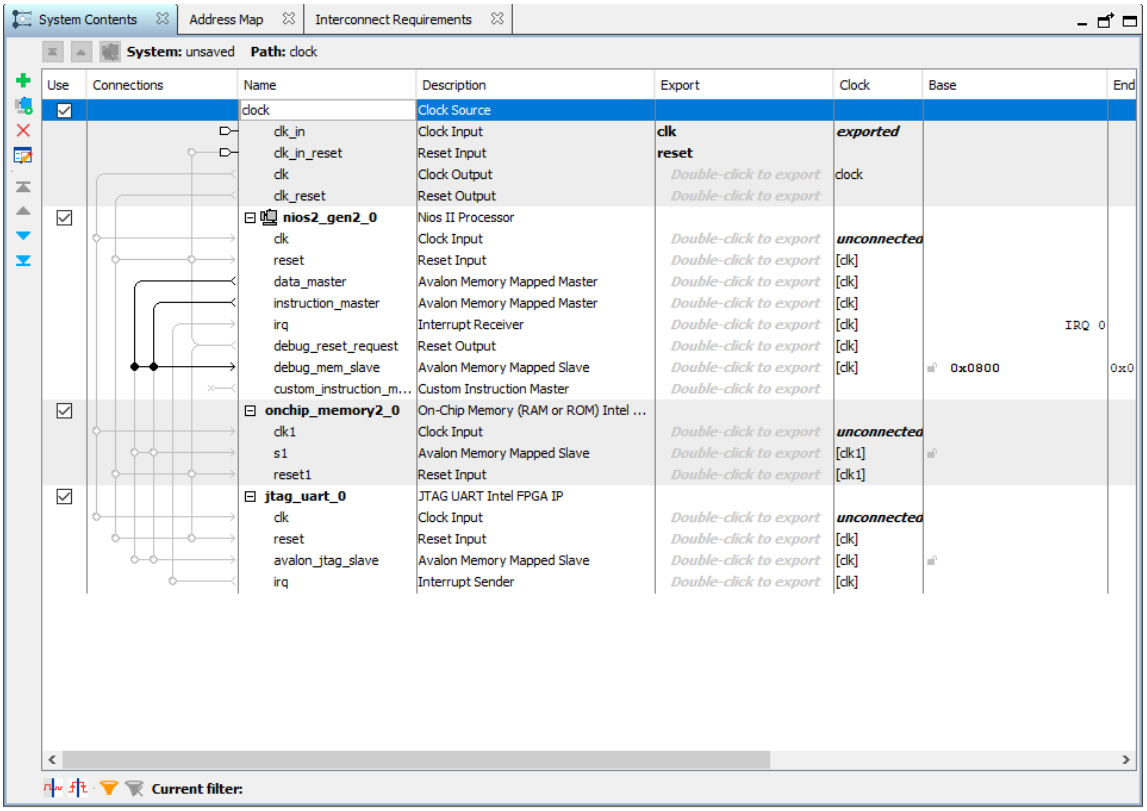


Hình 15. Thêm Jtag-uart.



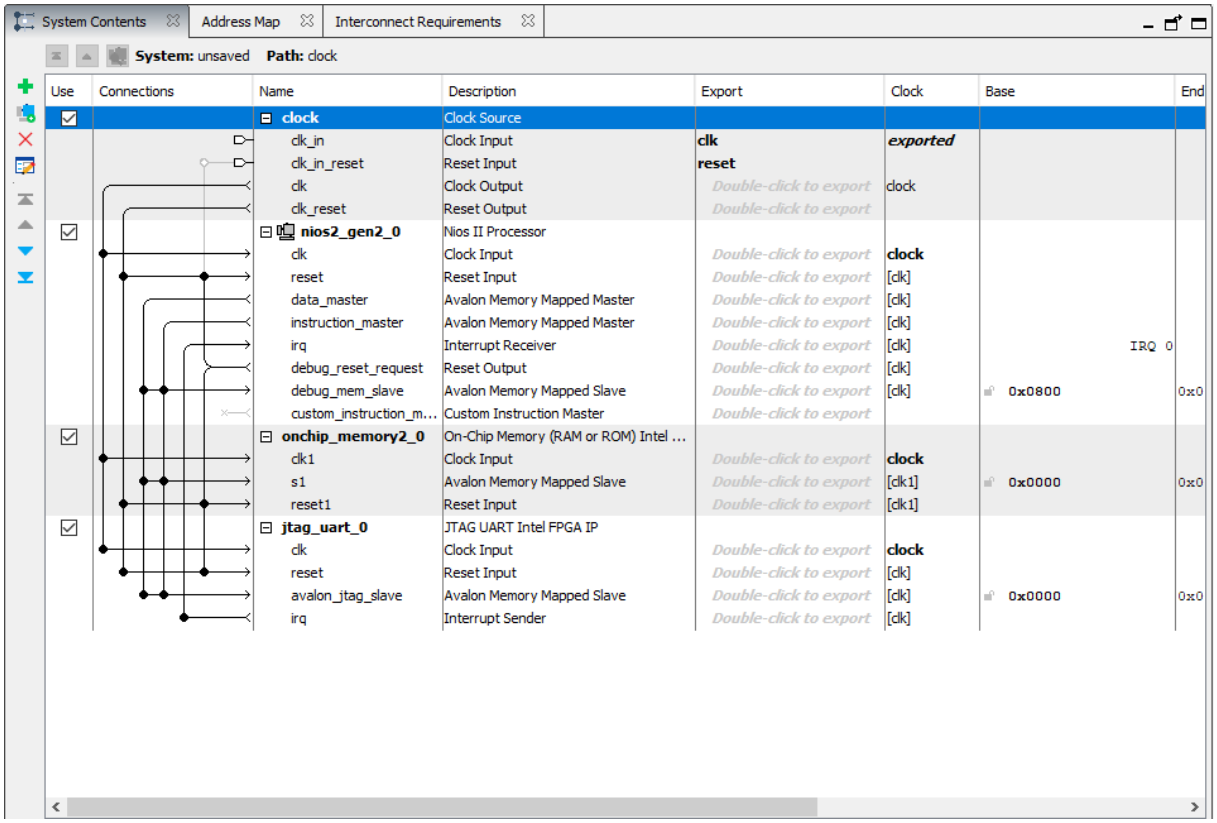
Hình 16. Cửa sổ cấu hình Jtag-uart.

- Sau khi thêm tất cả các mô đun theo yêu cầu, sẽ được hệ thống như hình 17, đây là hệ thống chưa có kết nối.



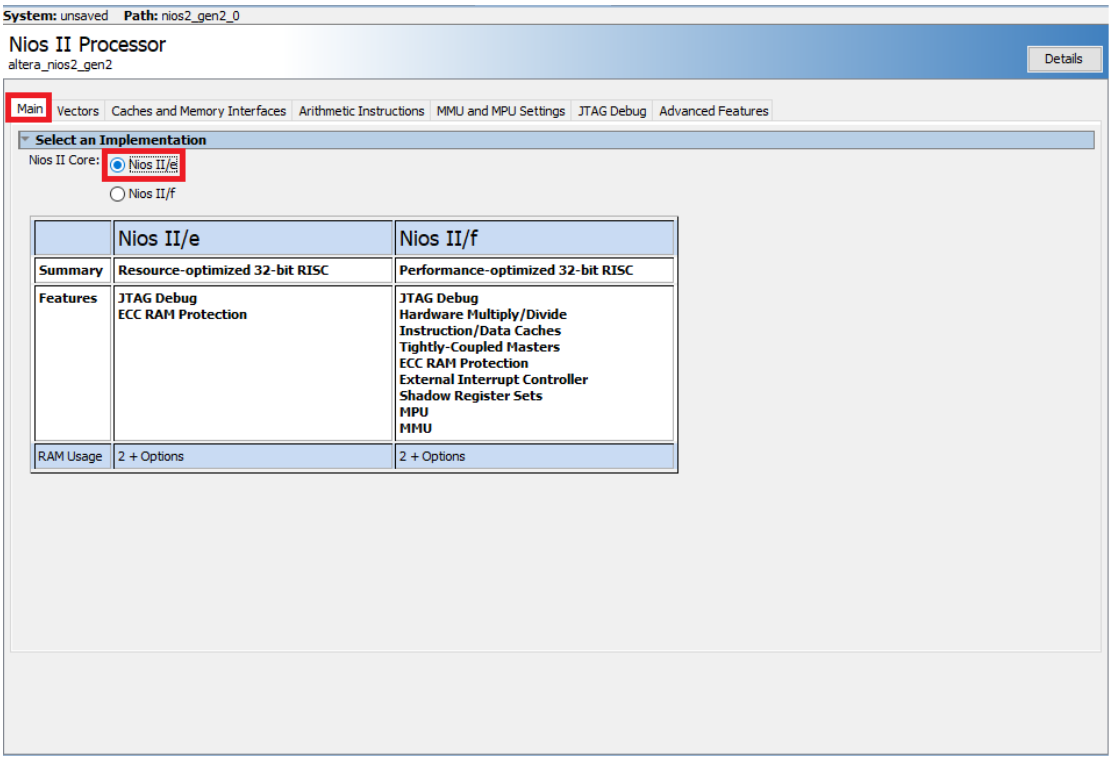
Hình 17. Hệ thống chưa kết nối các tín hiệu.

- Sau đó, tiến hành kết nối cho các thành phần như hình 18.



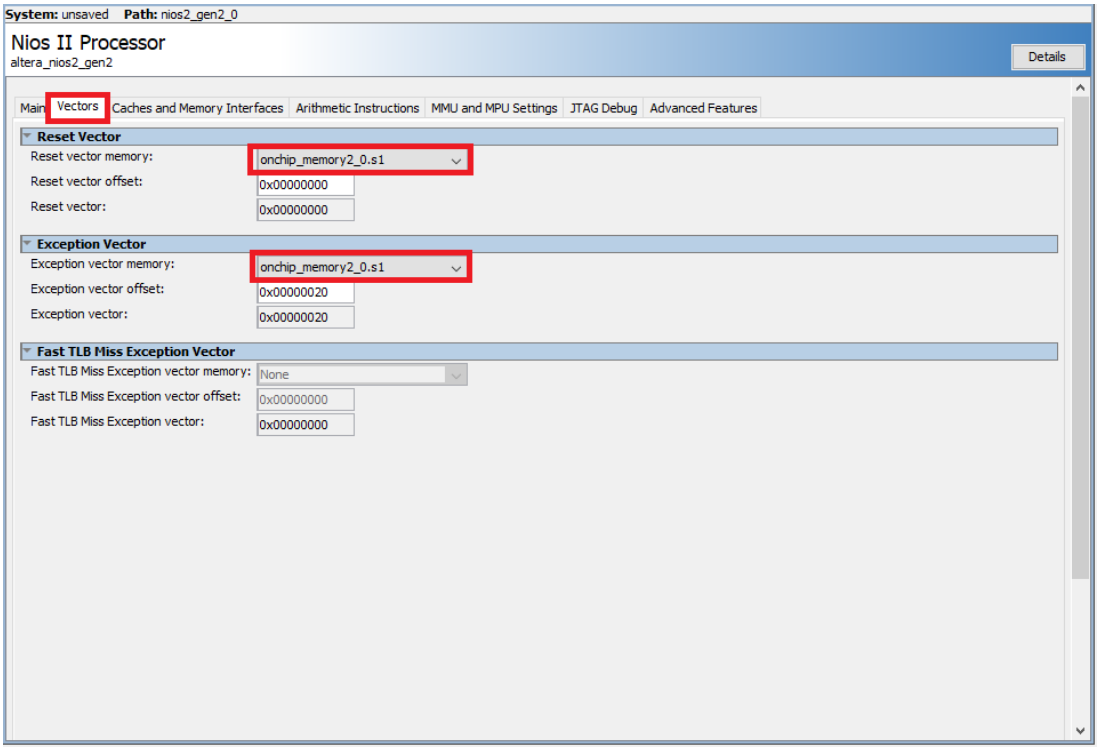
Hình 18. Hệ thống đã kết nối các tín hiệu.

- Tiếp theo tiến hành cấu hình cho từng thành phần. Đầu tiên, cấu hình cho NIOS II bằng cách bấm đúp chuột trái vào mô đun nios_gen2_0 và cấu hình ở cửa sổ Parameters. Ở tab Main, chọn Nios II e như hình 19.



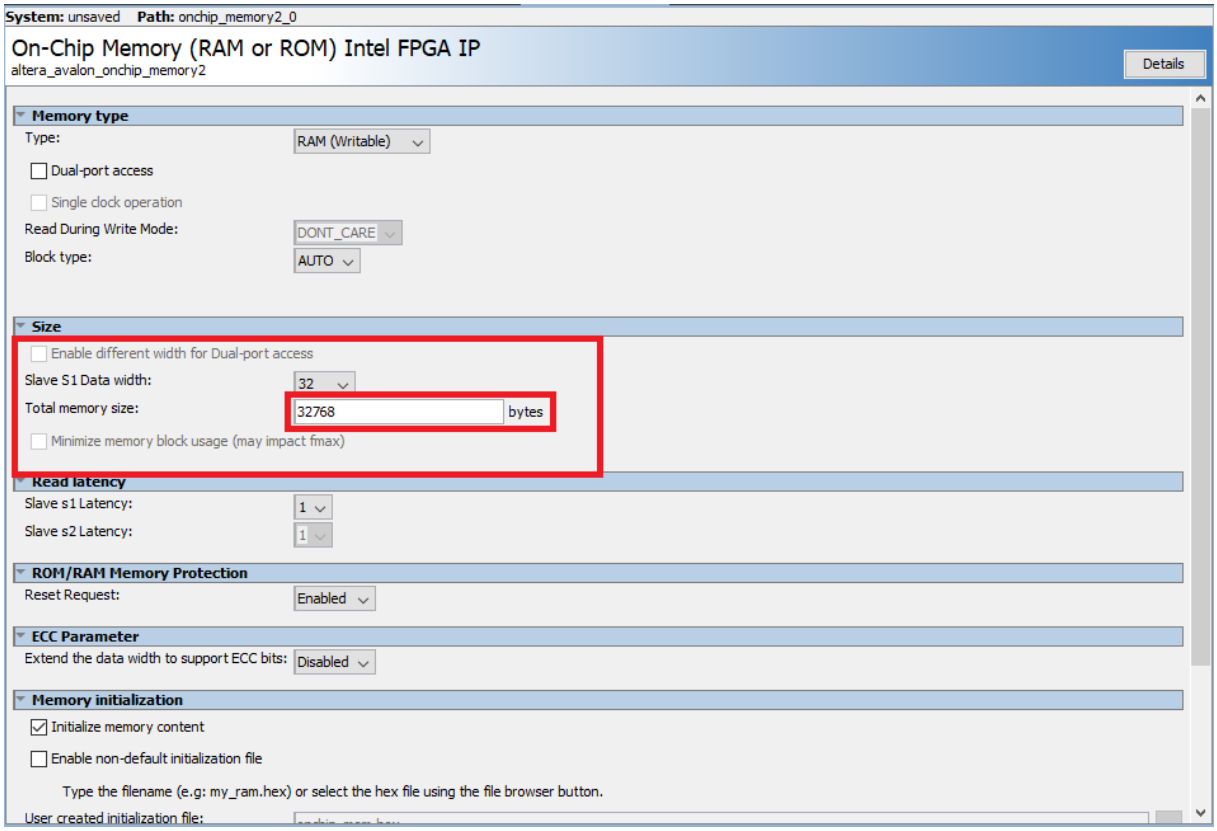
Hình 19. Cửa sổ cấu hình Nios II, tab Main.

- Tiếp đến, chuyển sang tab Vectors và cấu hình như hình 20.



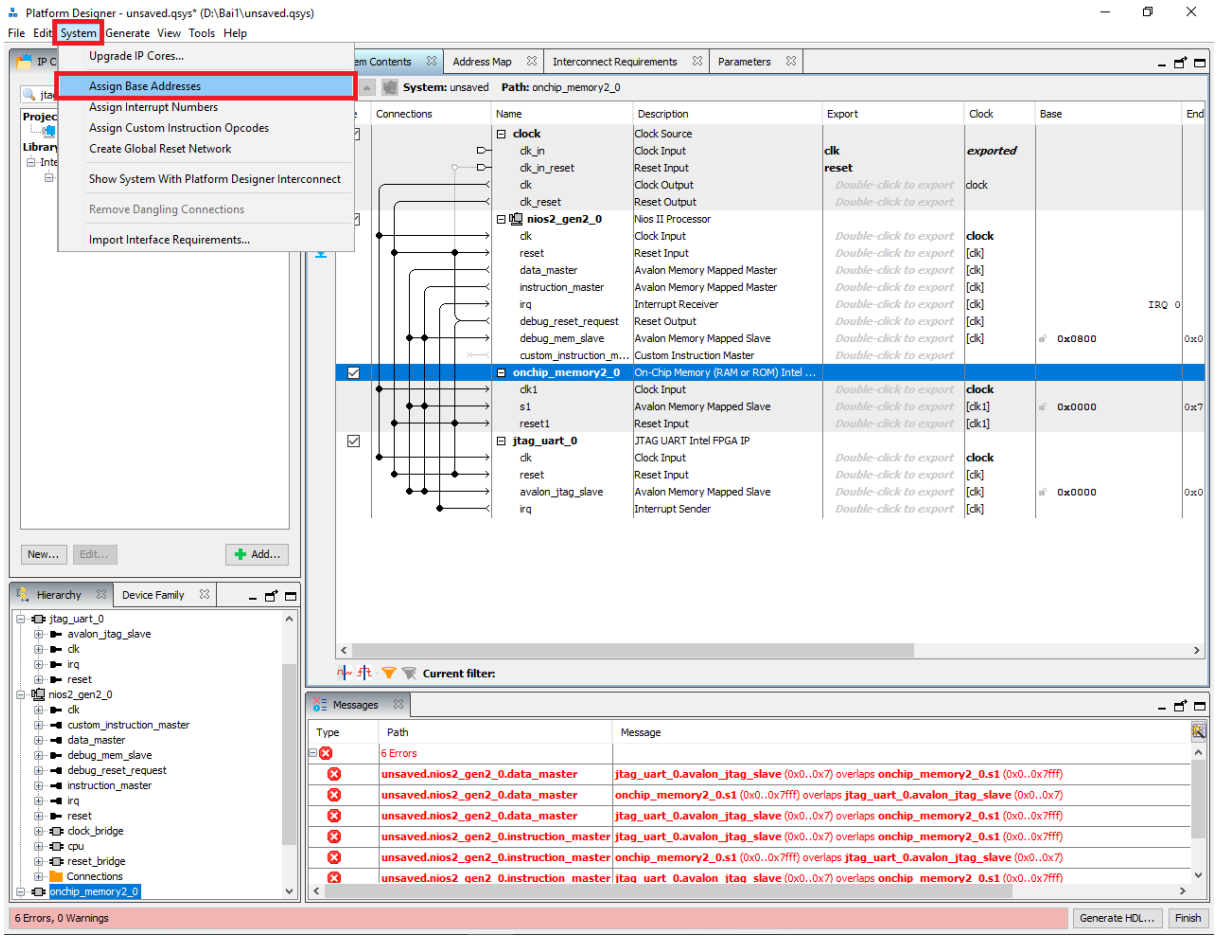
Hình 20. Cửa sổ cấu hình Nios II, tab Vectors.

- Tiếp tục là cấu hình Onchip_memory2_0. Tương tự, bấm đúp chuột trái vào onchip_memory2_0 và cấu hình kích thước bộ nhớ là **32768 bytes** như hình 21.



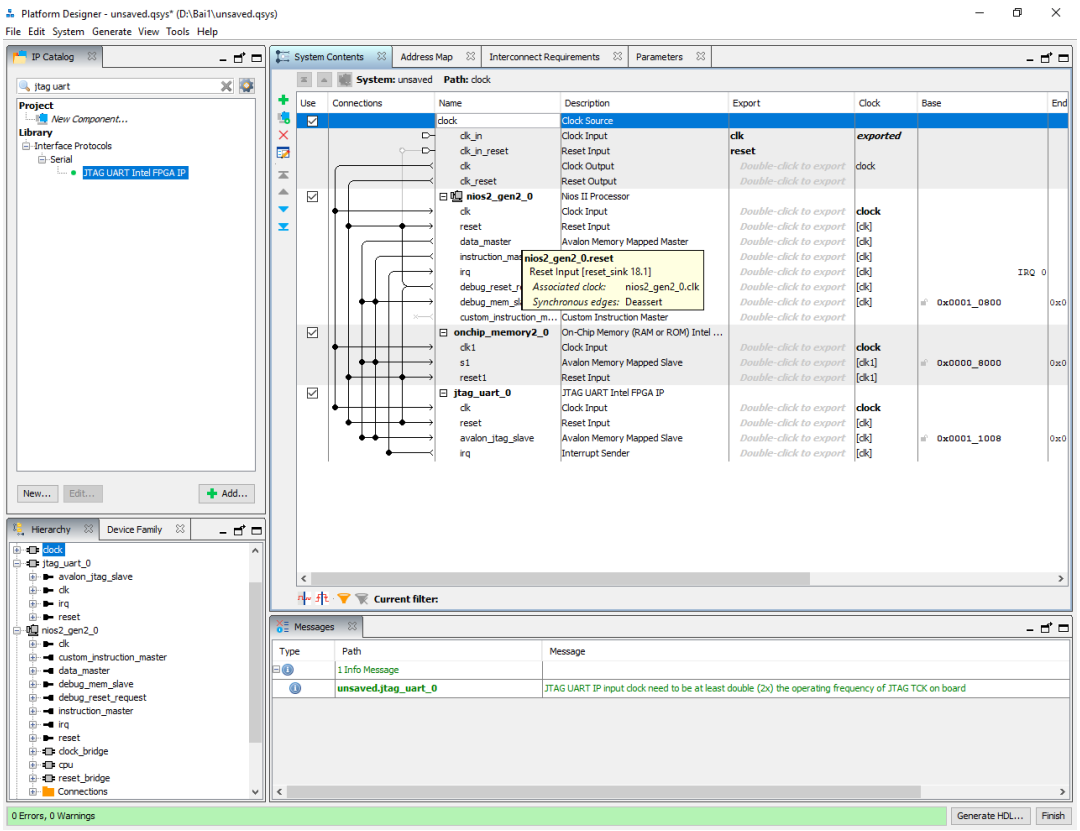
Hình 21. Cửa sổ cấu hình On-Chip Memory.

- Cuối cùng, gán địa chỉ lại cho các thành phần trong hệ thống. Chọn **System → Assign Based Addresses**.



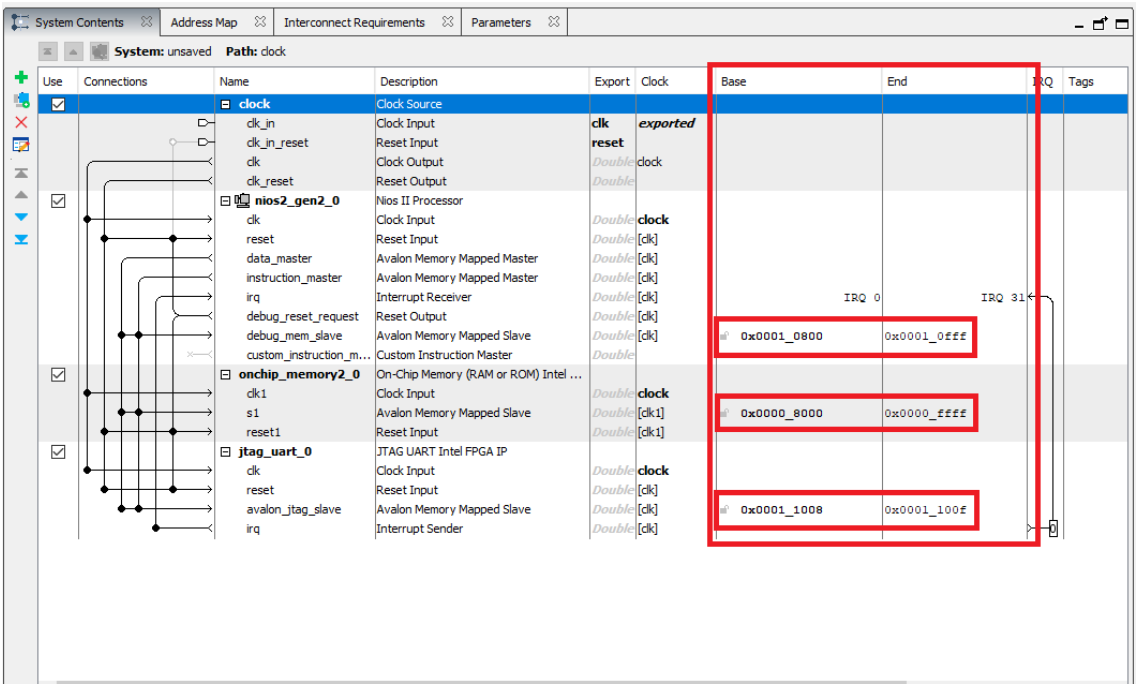
Hình 22. Hệ thống chưa gán địa chỉ.

- Sau khi gán địa chỉ tự động cho hệ thống, sẽ được như hình 23 với không còn lỗi.



Hình 23. Hệ thống sau khi gán địa chỉ.

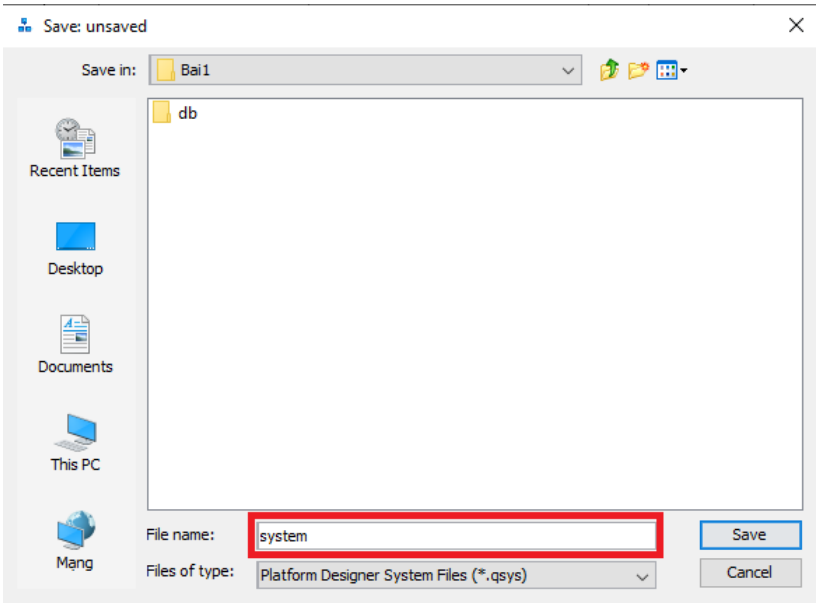
- Chú ý, hệ thống có được các địa chỉ như hình bên 24.



Hình 24. Địa chỉ của các thành phần.

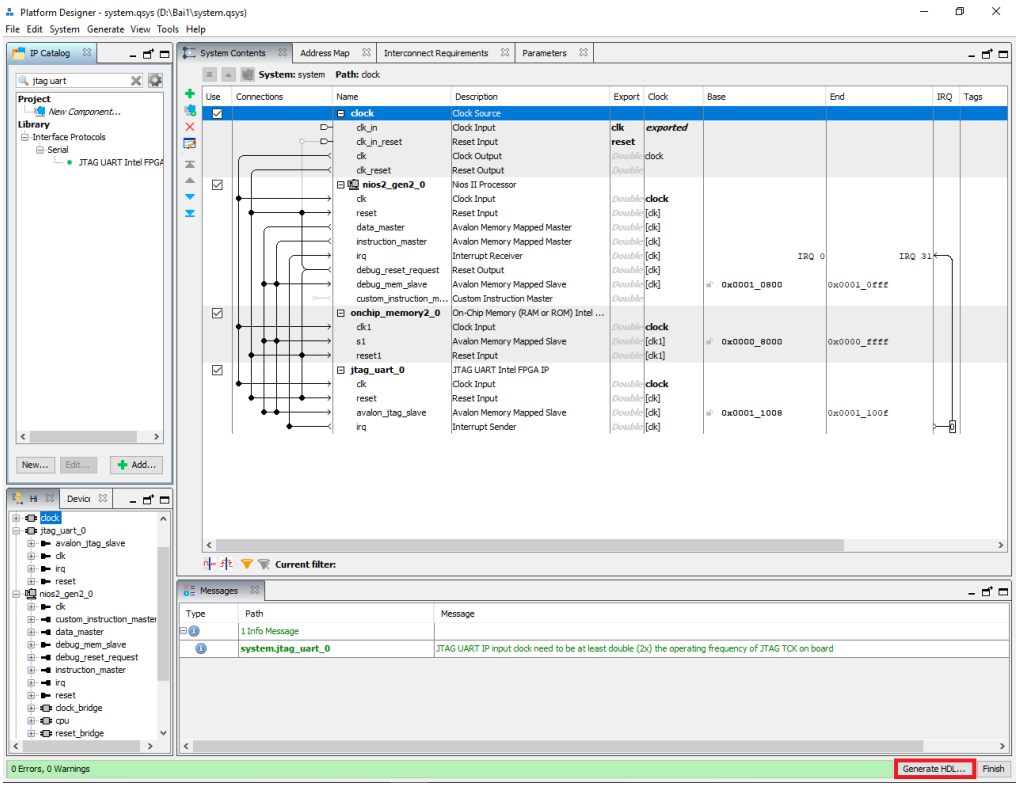
** Lưu ý: Các địa chỉ này sẽ không cố định và có thể bài làm của sinh viên sẽ không giống với bài hướng dẫn.

- Cuối cùng, lưu hệ thống với tên là system. Chọn **File → Save** (hoặc **Ctrl + S**) và chọn như hình 25.



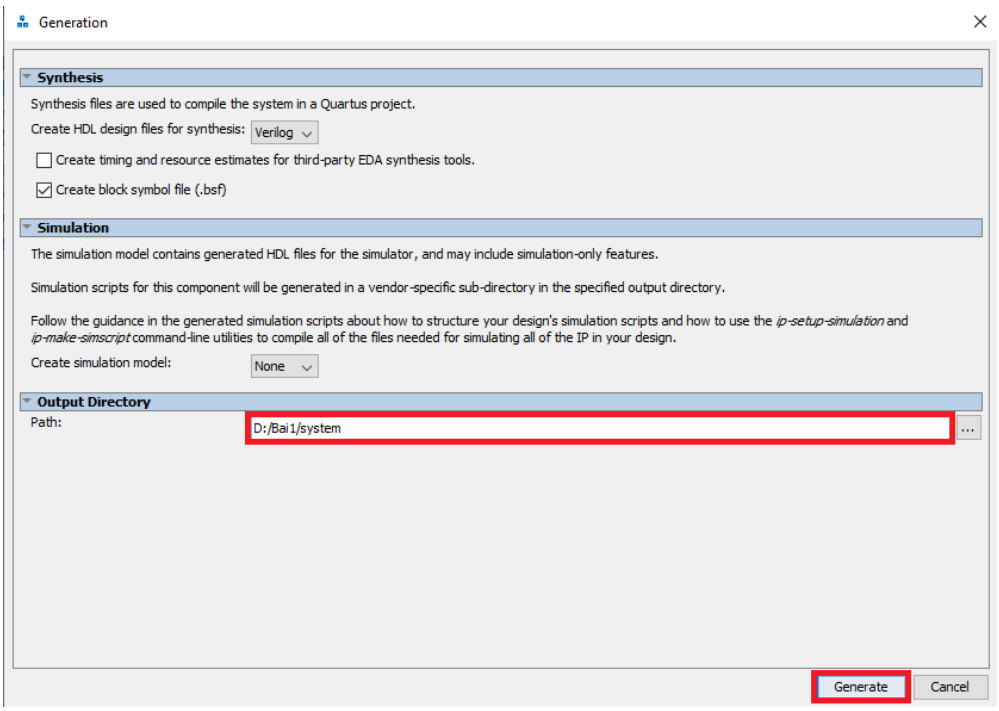
Hình 25. Lưu hệ thống.

- Sau khi lưu hoàn tất, tạo ra hệ thống phần cứng bằng cách chọn **Generate HDL** như hình 26.



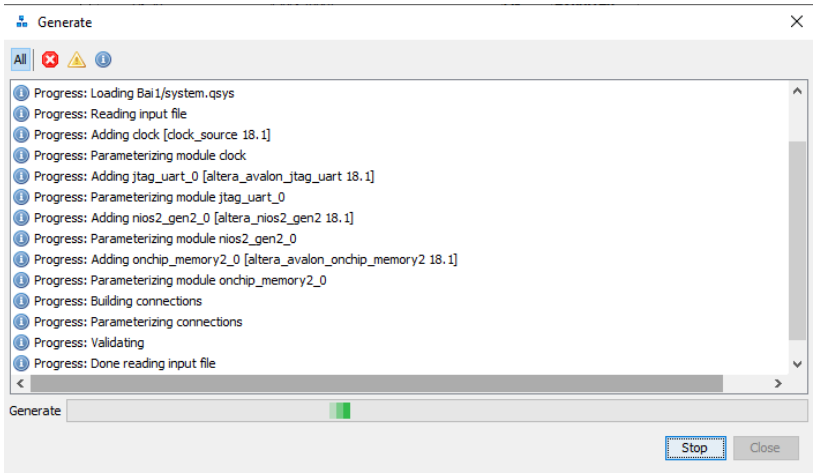
Hình 26. Generate hệ thống.

Giữ nguyên đường dẫn như hình 27 và chọn **Generate**.



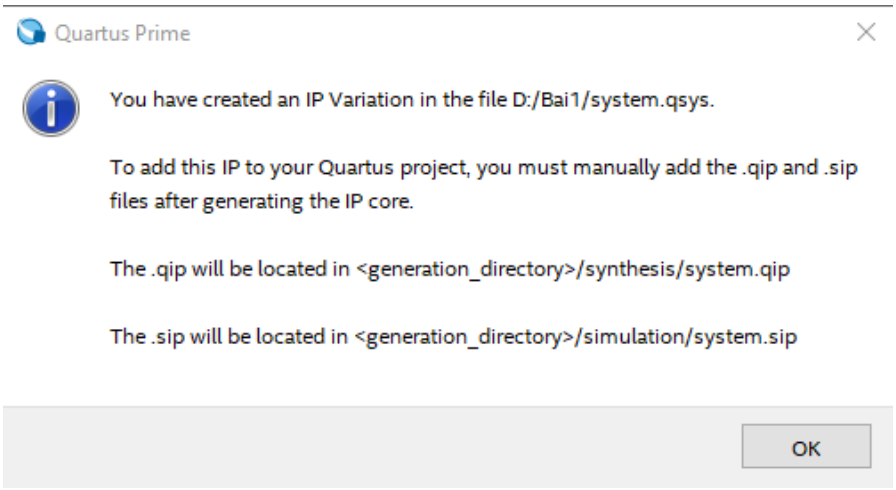
Hình 27. Generate hệ thống.

- Khi chọn **Generate**, sẽ có hộp thoại hiện ra thông báo quá trình generate, hình 28.



Hình 28. Cửa sổ thông báo quá trình generate.

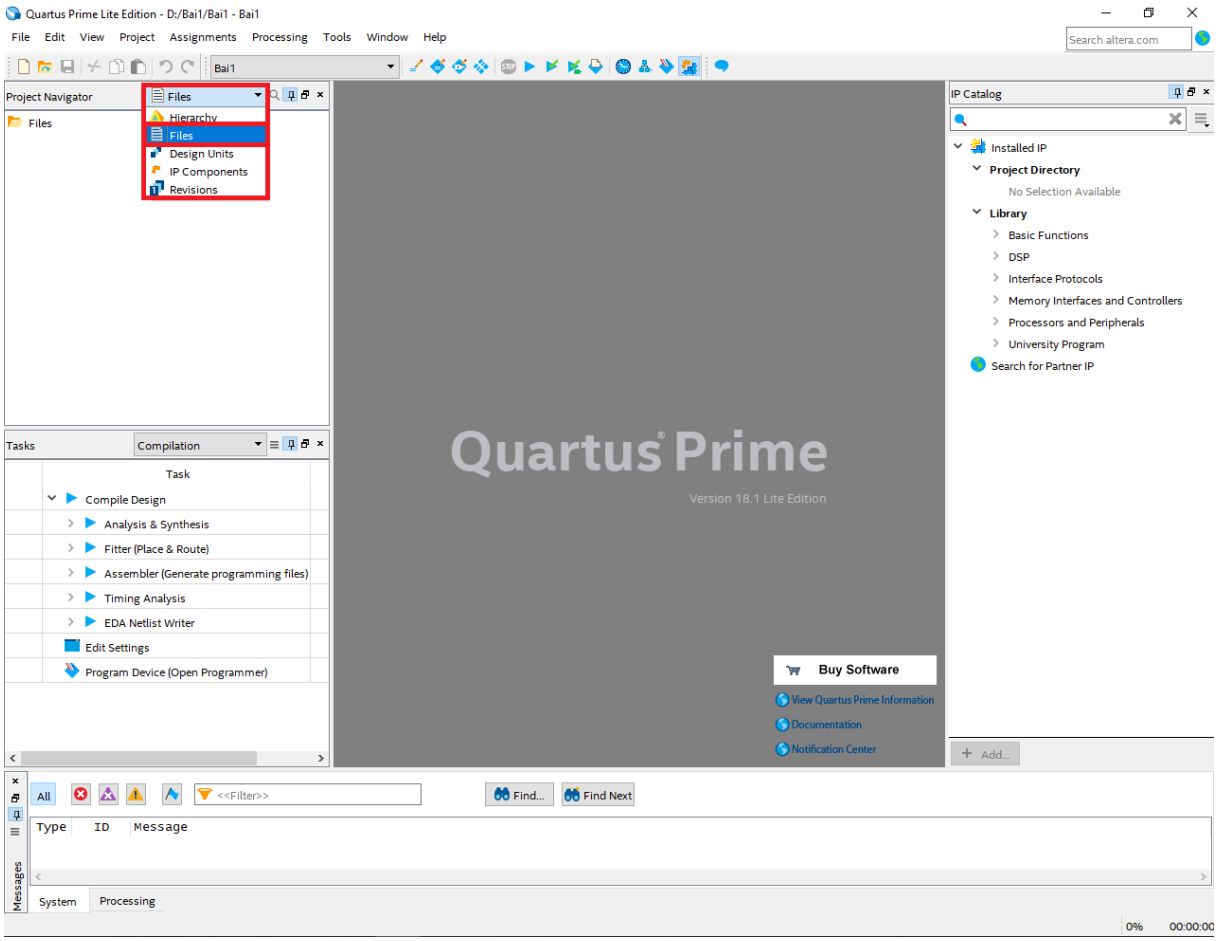
- Sau khi generate hoàn tất, chọn Finish. Sau khi chọn **Finish**, 1 hộp thoại sẽ hiện ra và thông báo đường dẫn và cách thêm hệ thống vừa tạo vào Quartus Prime, hình 29.



Hình 29. Thông báo đường dẫn và cách thêm mô đun.

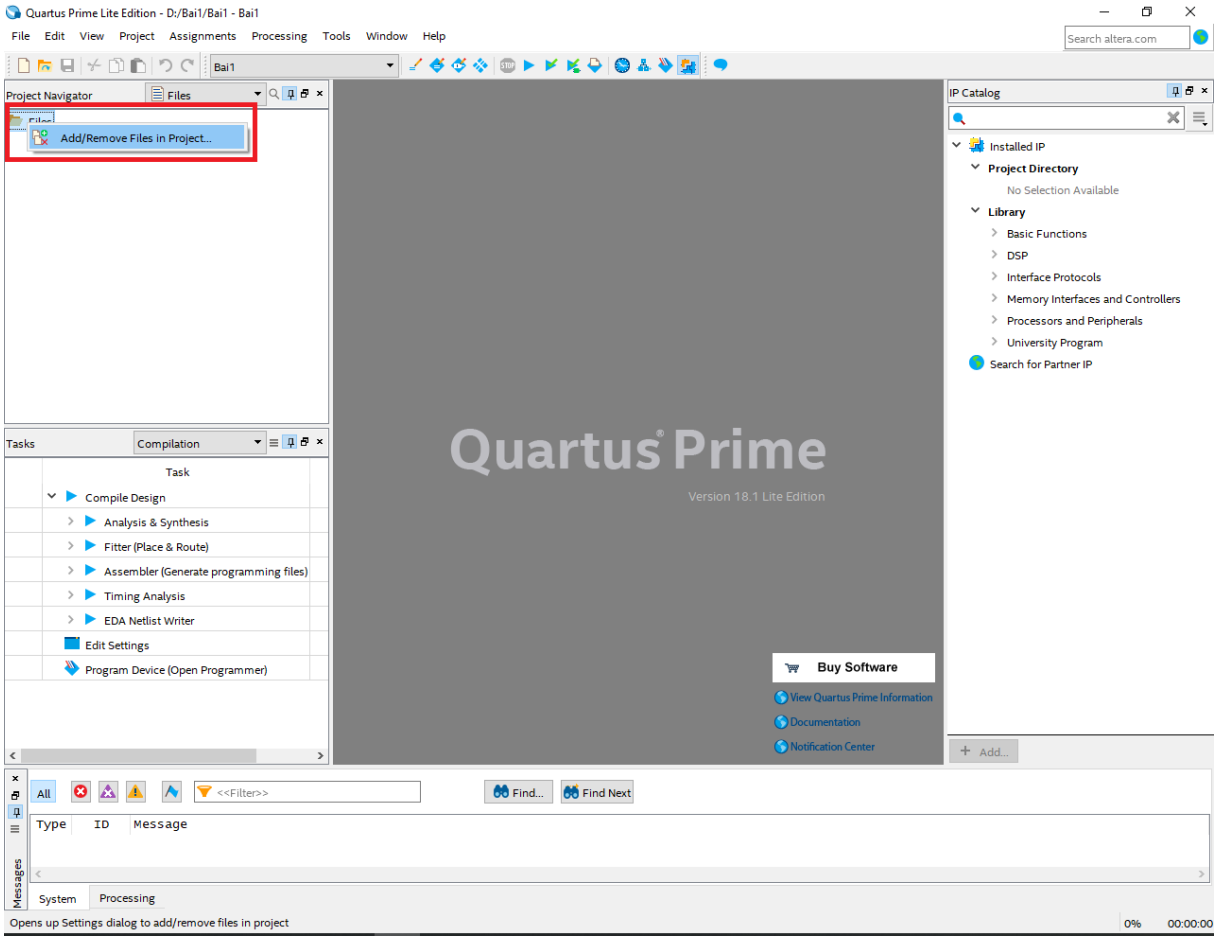
2.1.3. Tích hợp hệ thống vào Quartus Prime

- Tại cửa sổ Quartus Prime, chọn như hình 30 để chuyển từ tab **Hierachy** sang tab **Files**.



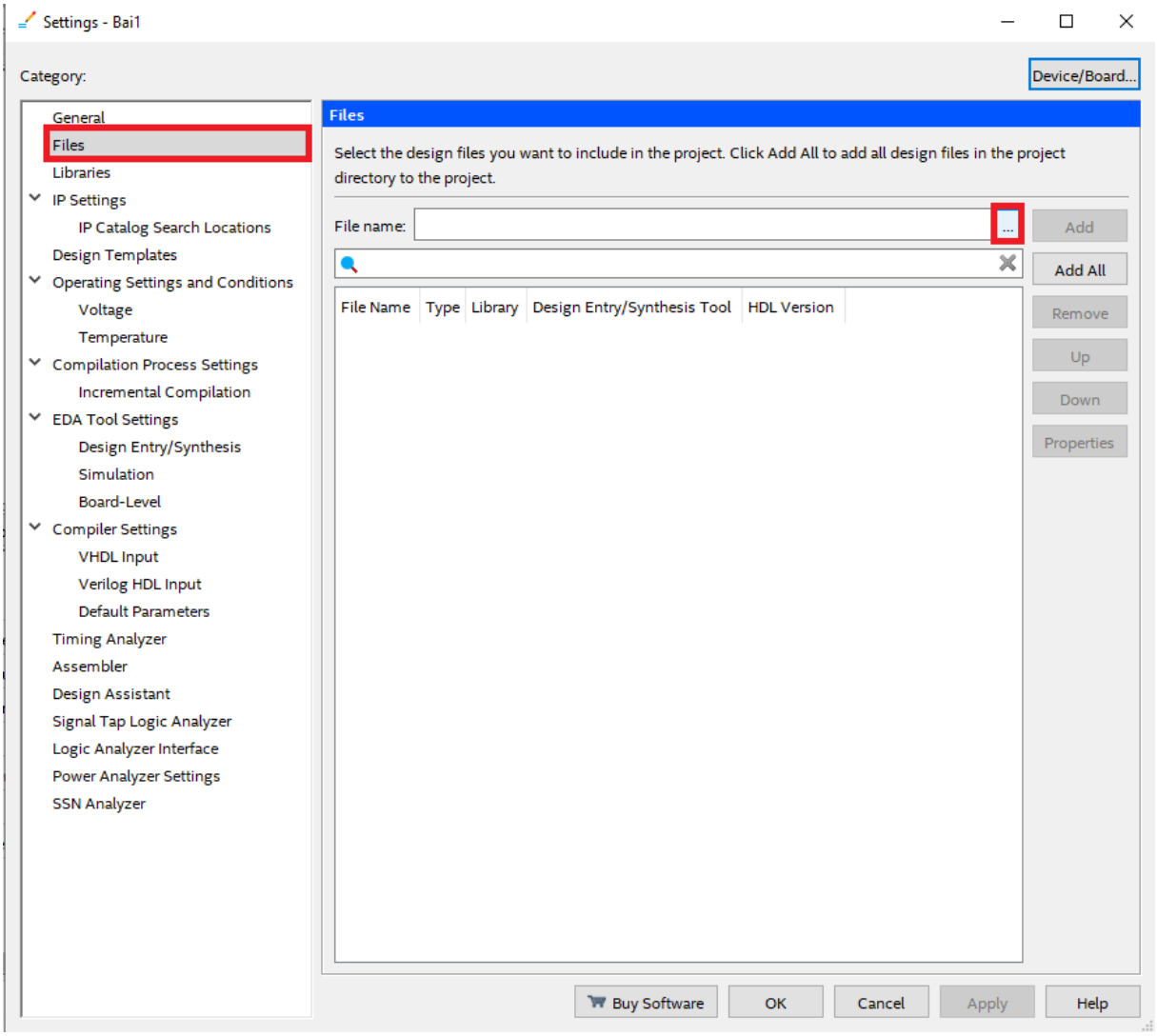
Hình 30. Chuyển sáng tab Files ở giao diện Quartus Prime.

- Ở tab Files, chọn chuột phải vào Files và chọn **Add/Remove Files in Project** như hình 31.



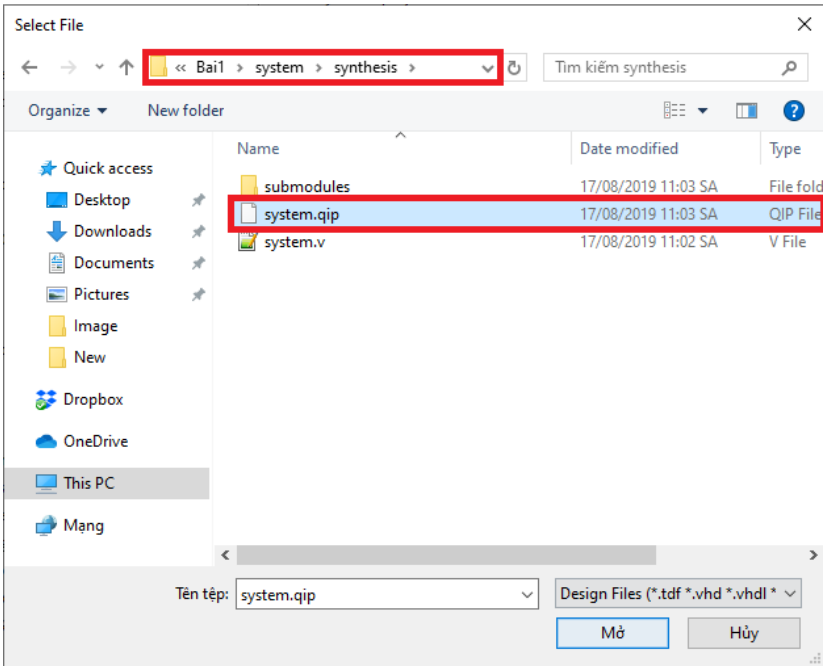
Hình 31. Thêm system.qip vào project.

- Ở cửa sổ Setting, chọn nút … như hình 32 để dẫn đường dẫn file **system.qip** vào project.



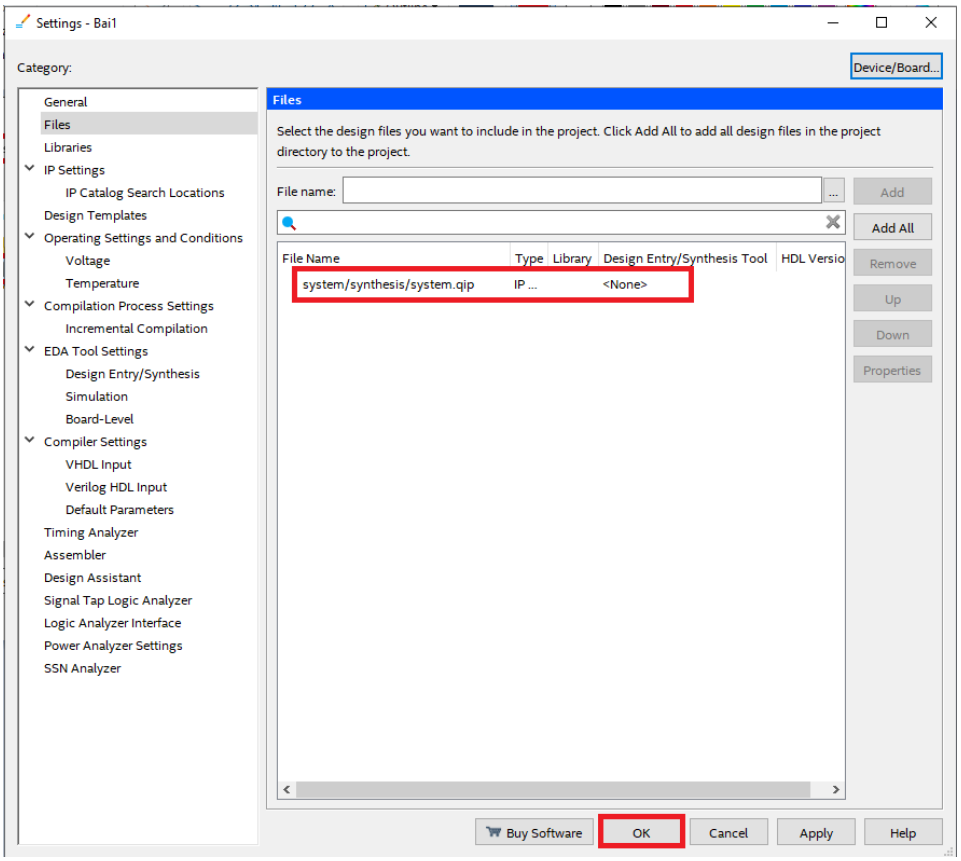
Hình 32. Cửa sổ Setting.

- File system.qip được tạo ra ở thư mục có đường dẫn: ***/system/synthesis***. Tiến hành thêm file **system.qip** như hình 33.



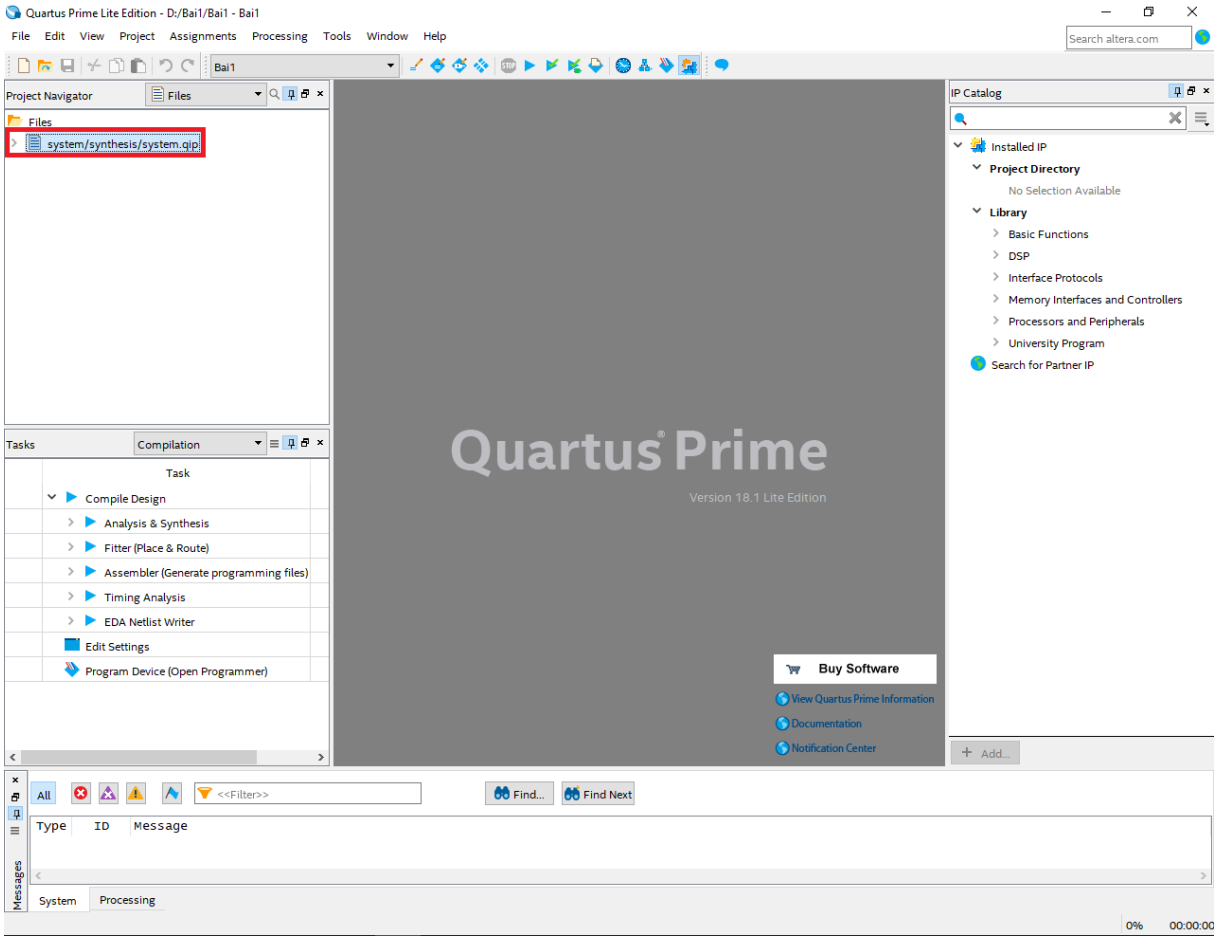
Hình 33. Cửa sổ Select File.

- Khi thêm ở giống như hình 33, sẽ trở về cửa sổ Setting và sẽ xuất hiện file system.qip như hình 34. Sau đó, chọn OK để hoàn tất.



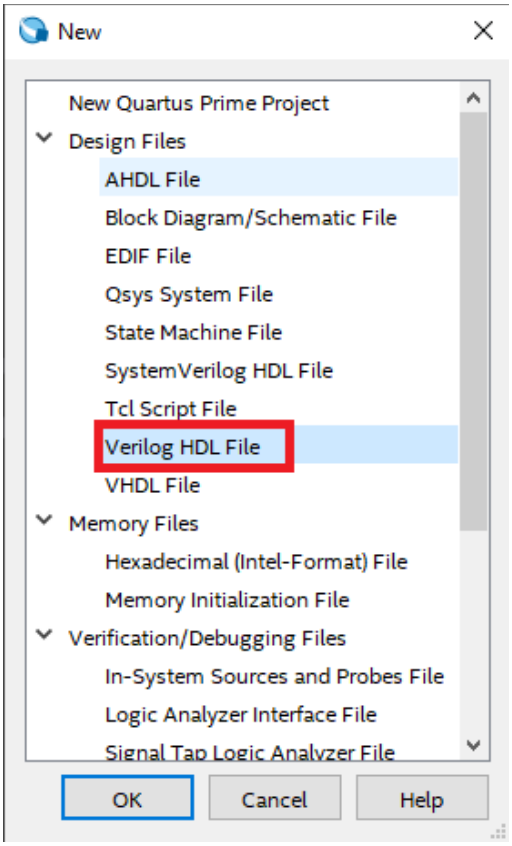
Hình 34. Kết quả sau khi thêm file.

- Sau khi chọn OK, sẽ trở về giao diện của phần mềm Quartus Prime, file cần thêm đã xuất hiện như hình 35.



Hình 35. Kết quả sau khi thêm file ở giao diện Quartus Prime.

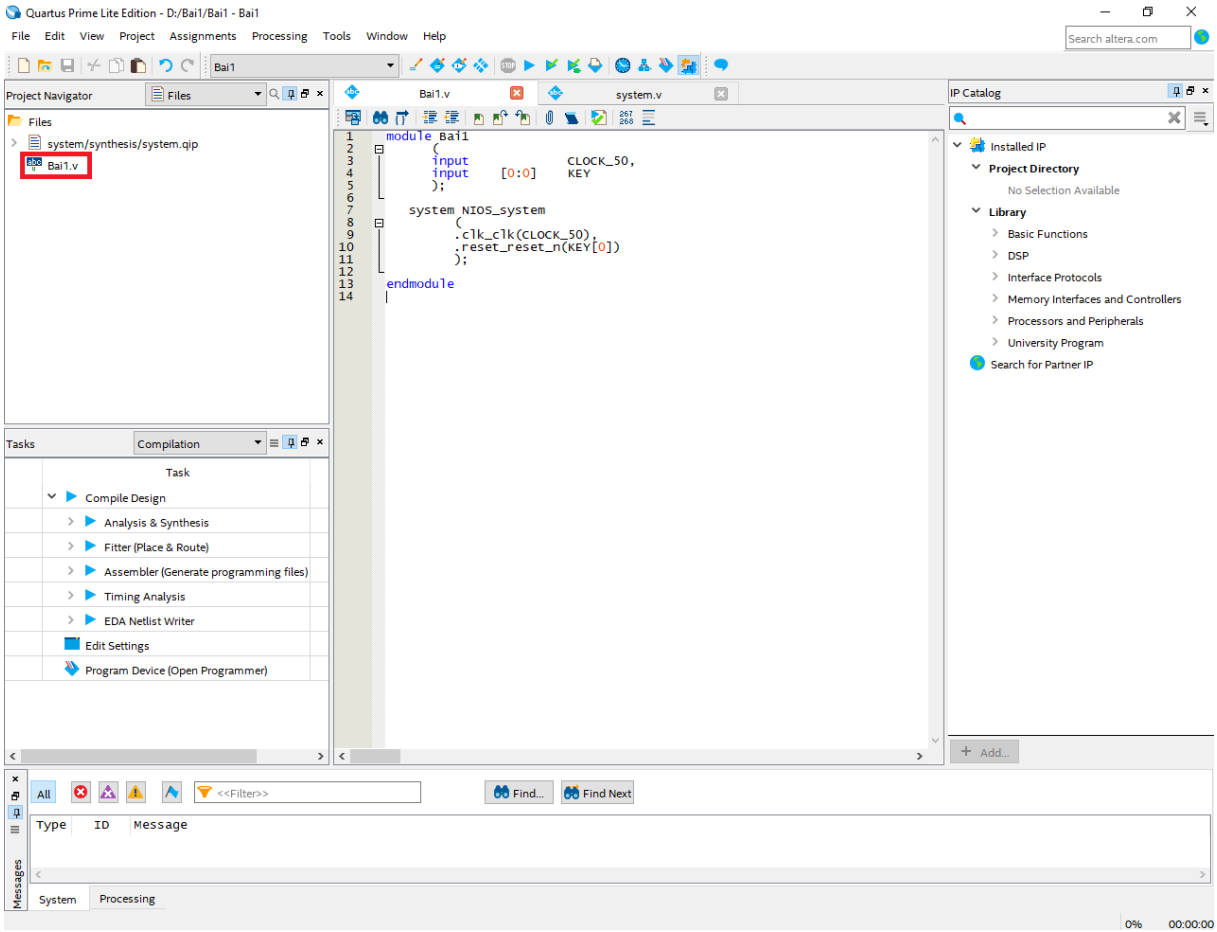
- Tiếp theo, tạo file top-level cho project. Chọn **File → New** (hoặc **Ctrl + N**), sẽ có cửa sổ hiện ra như hình 36. Chọn file mới là Verilog HDL File như hình 36.



Hình 36. Cửa sổ New.

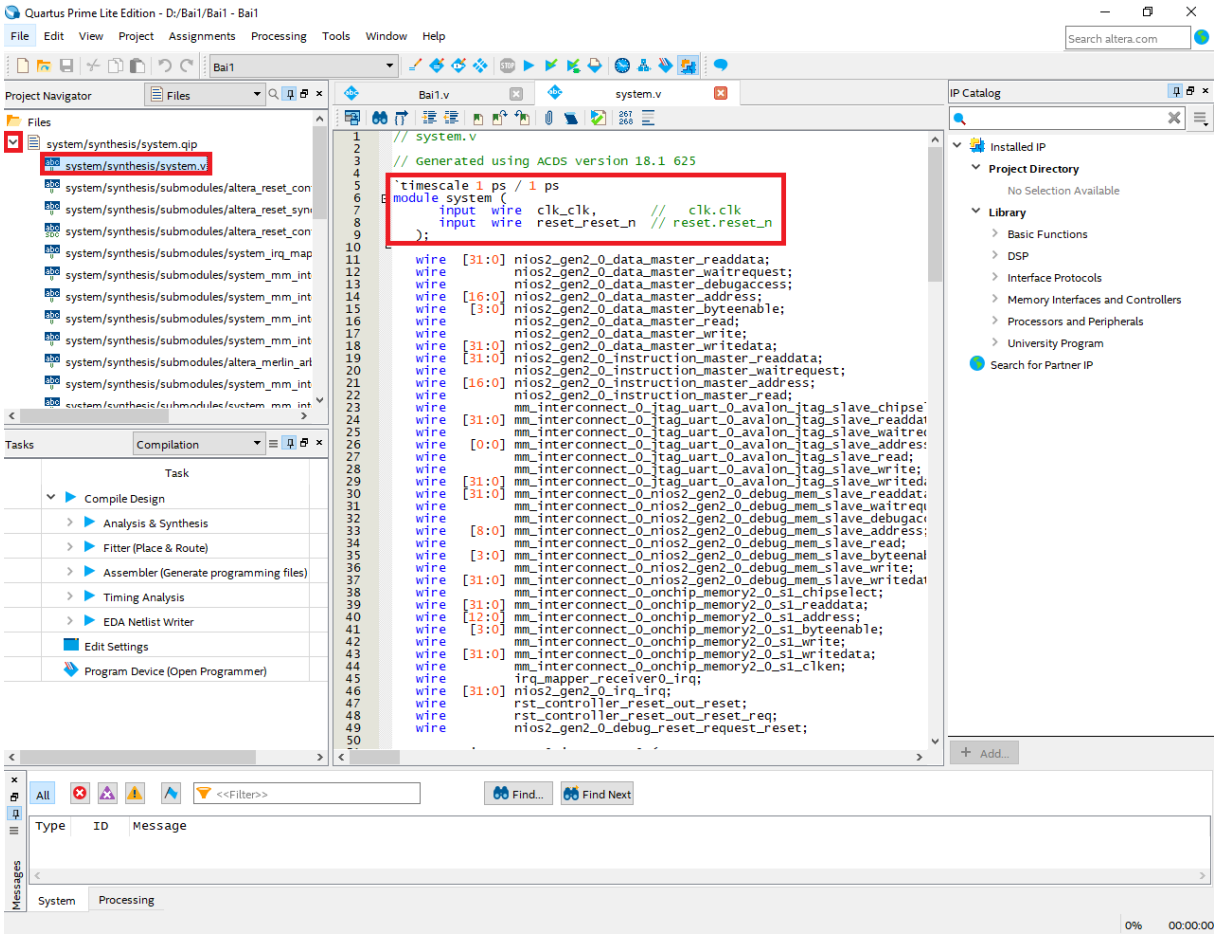
- Gõ đoạn code bên dưới và lưu vào project như hình 37.

```
module Bai1
        (
        input                   CLOCK_50,
        input       [0:0]       KEY
        );

    system NIOS_system
            (
            .clk_clk(CLOCK_50),
            .reset_reset_n(KEY[0])
            );

endmodule
```



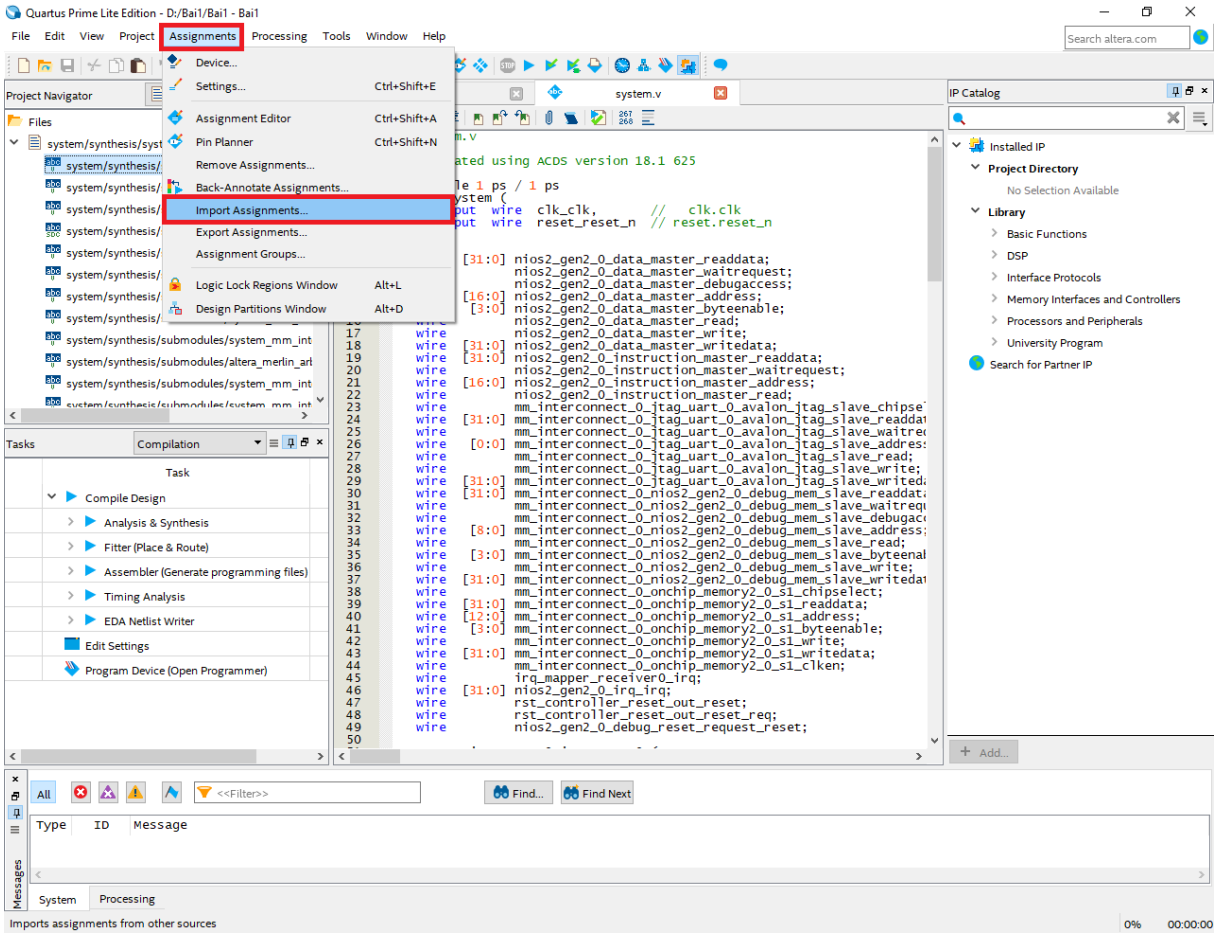
Hình 37. Giao diện Quartus Prime sau khi thêm file Bai1.v

- Để biết được các kết nối vào ra của mô đun system, mở file **system.v**. Hình 38 thể hiện code và các kết nối vào ra của mô đun system.



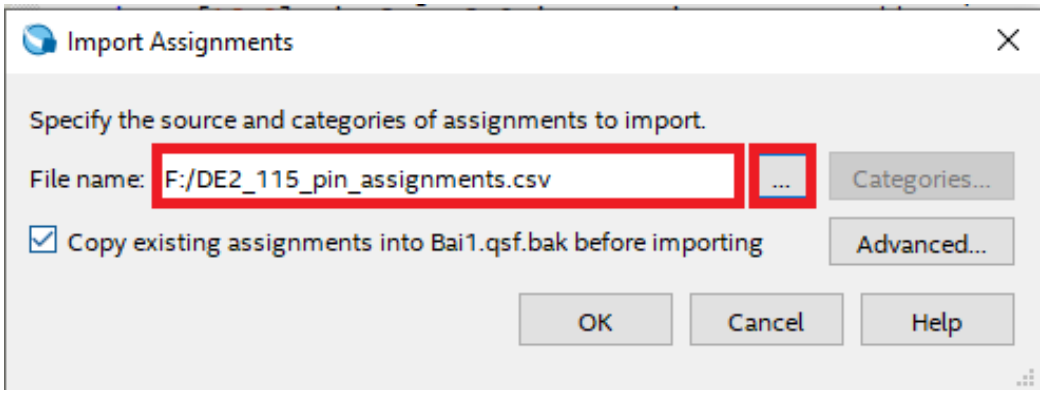
Hình 38. Các chân tín hiệu của mô đun system.

- Sau đó, tiến hành gán pin cho các tín hiệu, chọn **Assignment → Import asignments**… như hình 39.



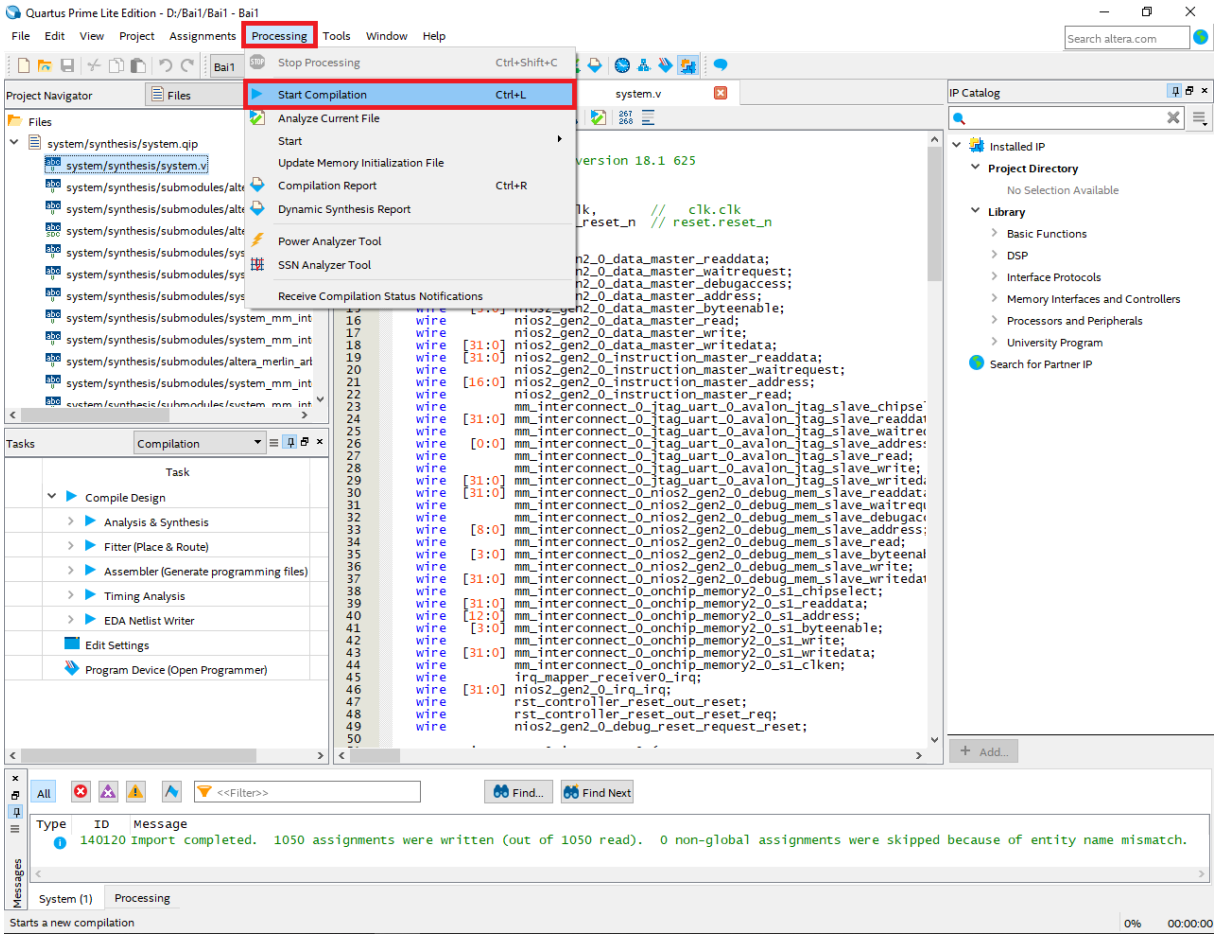
Hình 39. Gán chân cho thiết bị.

- Chọn nút … và dẫn đường dẫn file **DE2_115_pin_assignments.cvs** vào rồi chọn **OK** như hình 40.



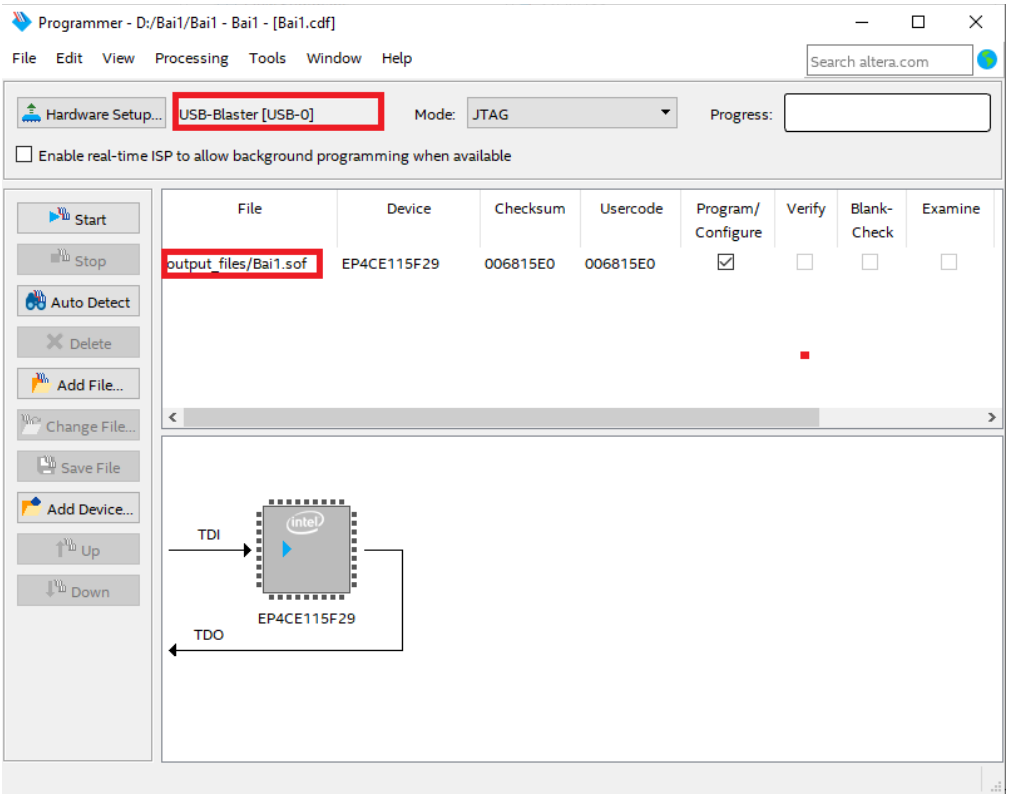
Hình 40. Cửa sổ Import Assignments.

- Cuối cùng, tổng hợp phần cứng bằng cách chọn **Processing → Start Compilation** (hoặc **Ctrl + L**), hình 41.



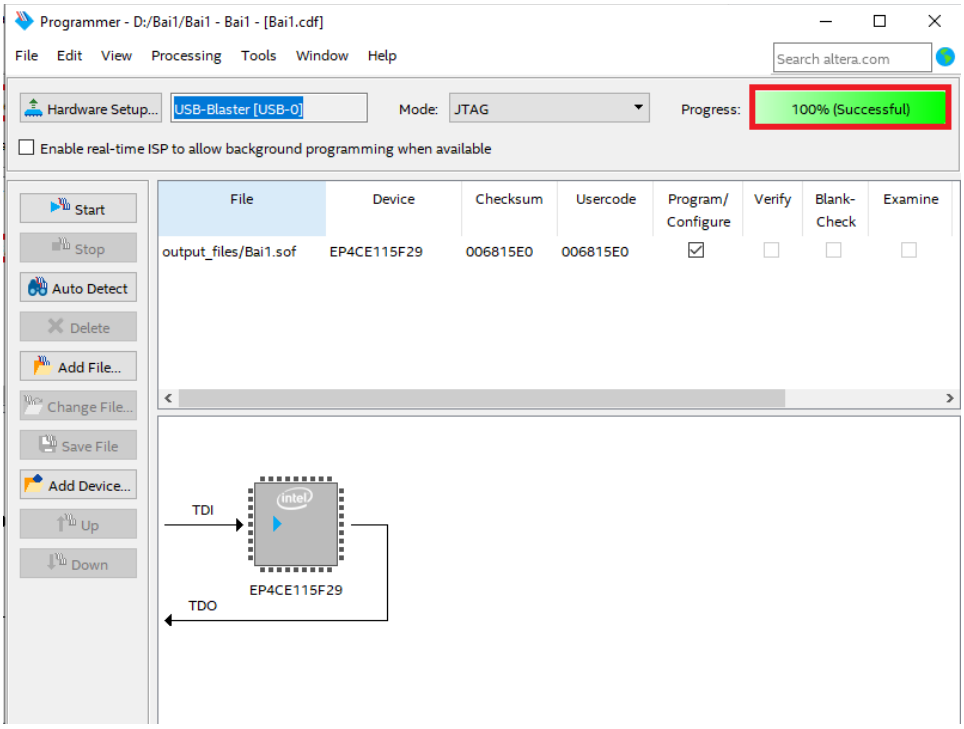
Hình 41. Tiến hành biên dịch.

- Sau khi biên dịch hoàn tất, nạp phần cứng xuống board DE2-115, chọn **Tools → Programmer**. Một cửa sổ hiện ra, chọn như hình 42. Yêu cầu là phải có **USB-Blaster [USB-0]** và **Bai1.sof** như trong hình 42.



Hình 42. Cửa sổ Programmer.

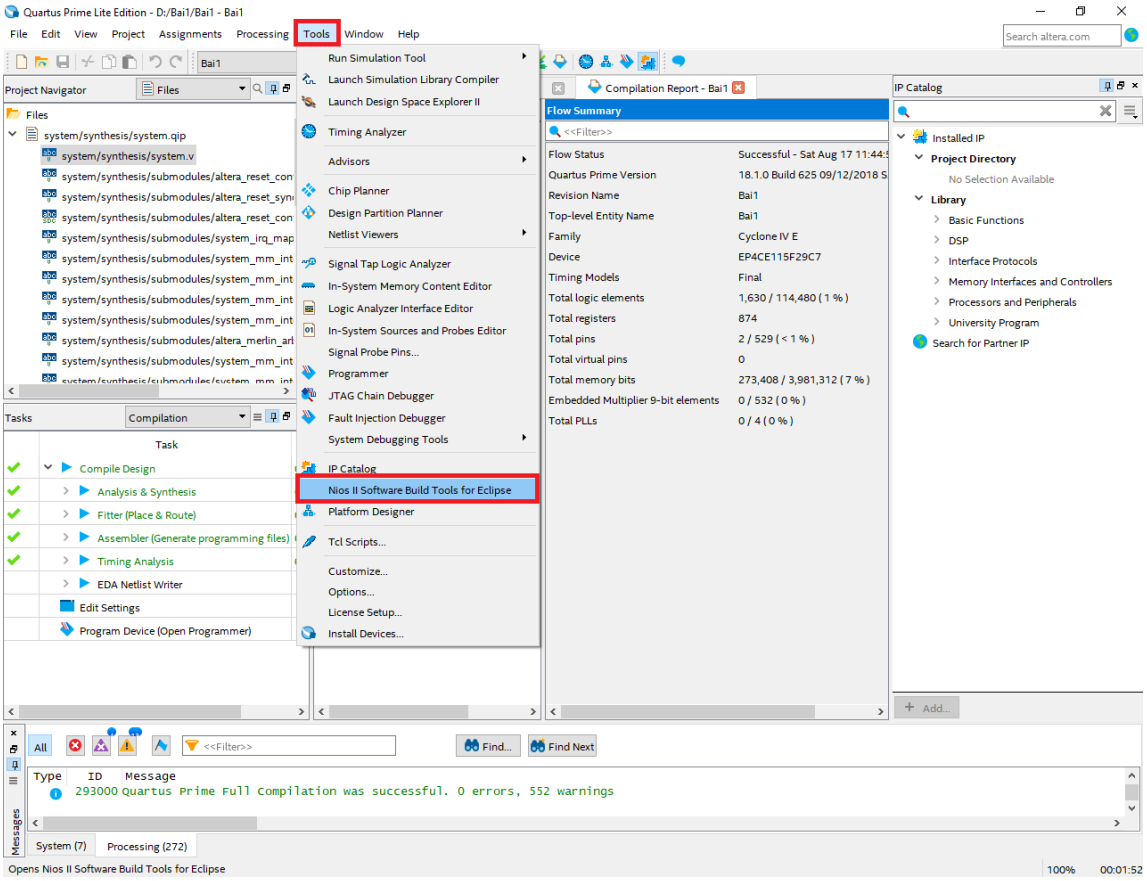
- Sau đó, chọn Start để nạp phần cứng xuống board. Nếu nạp thành công thì các LED sẽ tắt hết. Quá trình nạp thành công được thông báo như hình 43.



Hình 43. Kết quả nạp phần cứng xuống board.

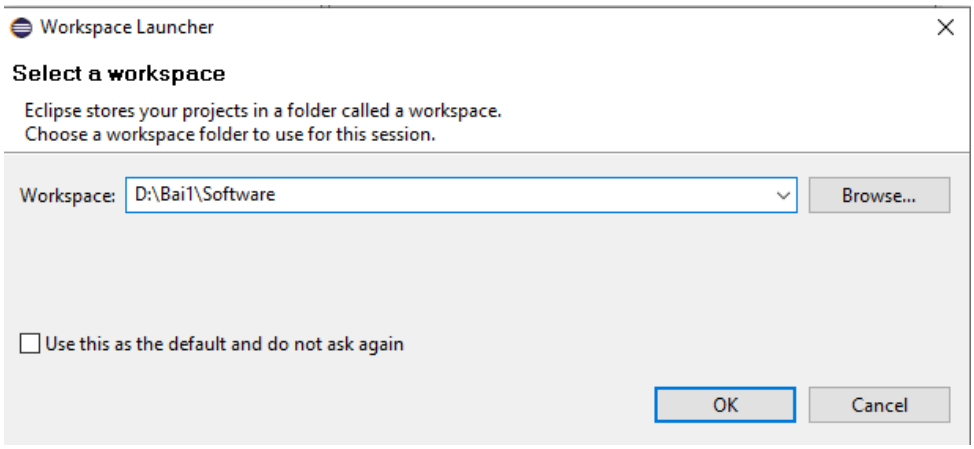
2.2. Xây dựng phần mềm

- Khởi động phần mềm NIOS II bằng cách chọn **Tools → NIOS II Software Build Tools for Eclipse**, hình 44.



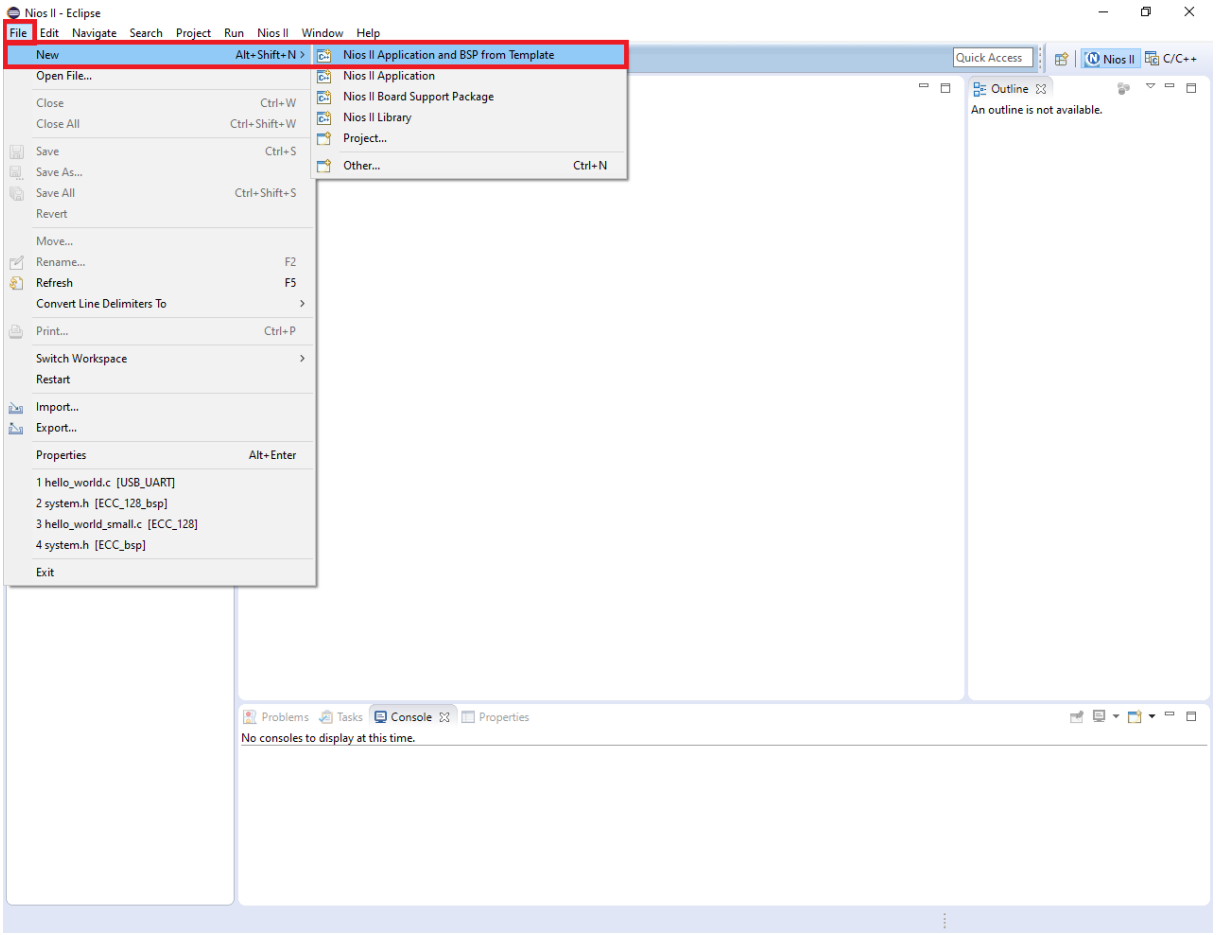
Hình 44. Mở phần mềm Nios II ...

- Sau khi mở xong Elipse, tạo workspake như hình 45, lưu ý không tạo đường dẫn có khoảng trắng.



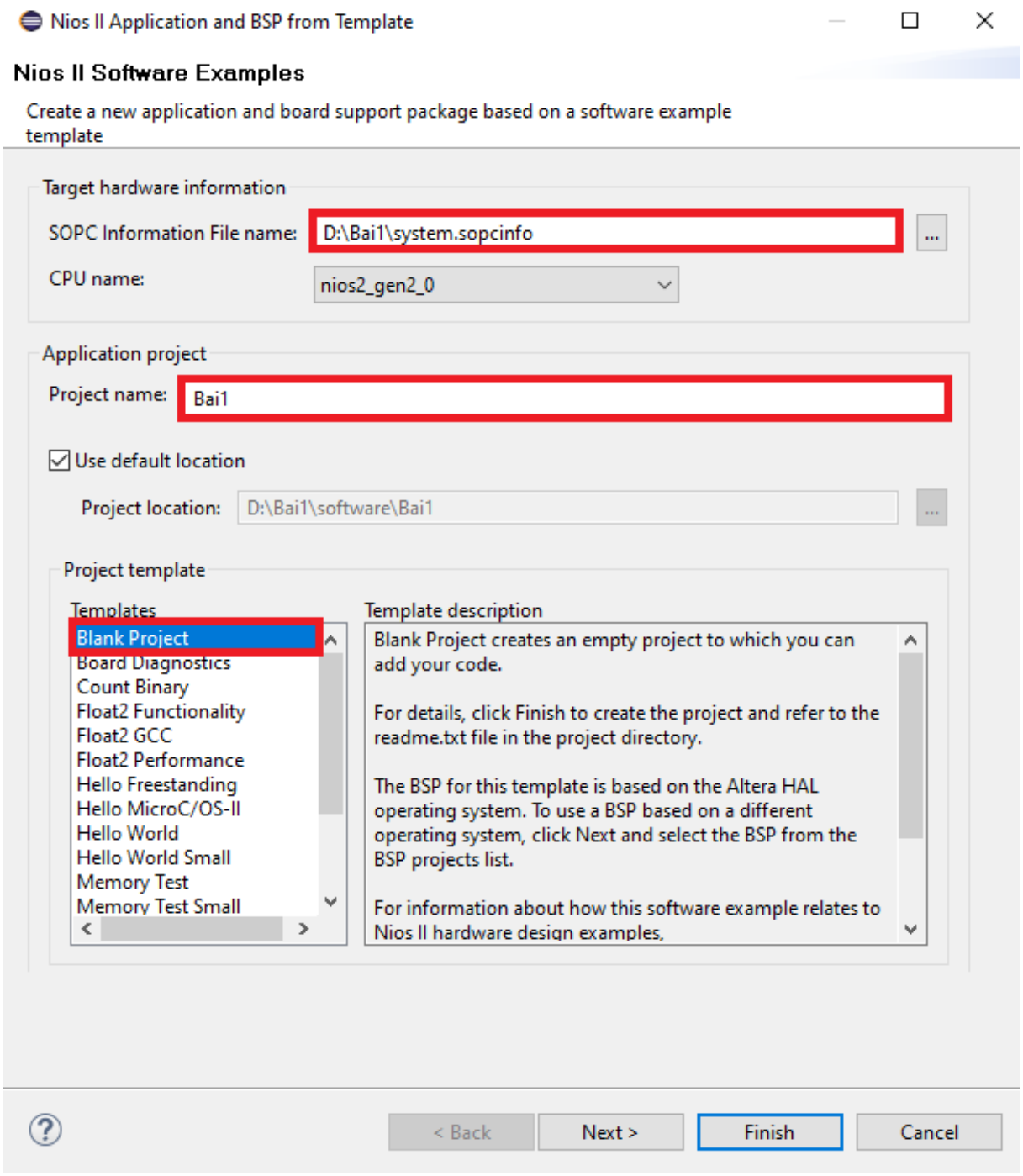
Hình 45. Tạo workspace.

Tiến hành tạo project như hình 46.



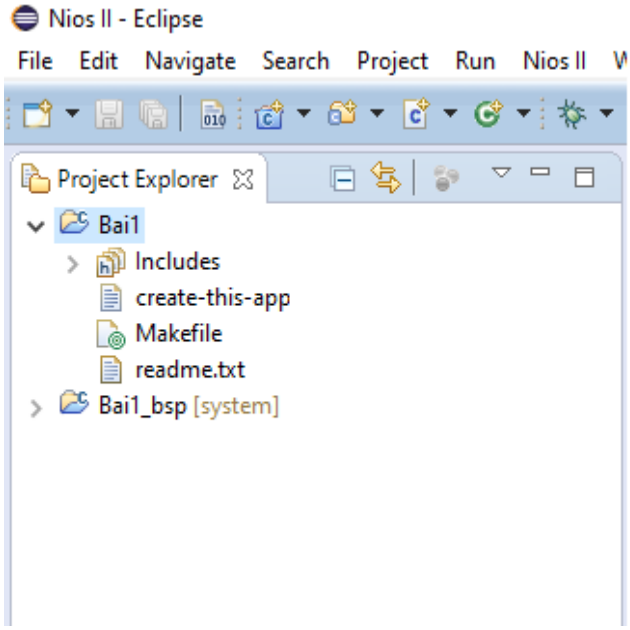
Hình 46. Tạo project.

- Ở cửa sổ Nios II Software Examples, thiết lập như hình 47 rồi chọn **Finish**.



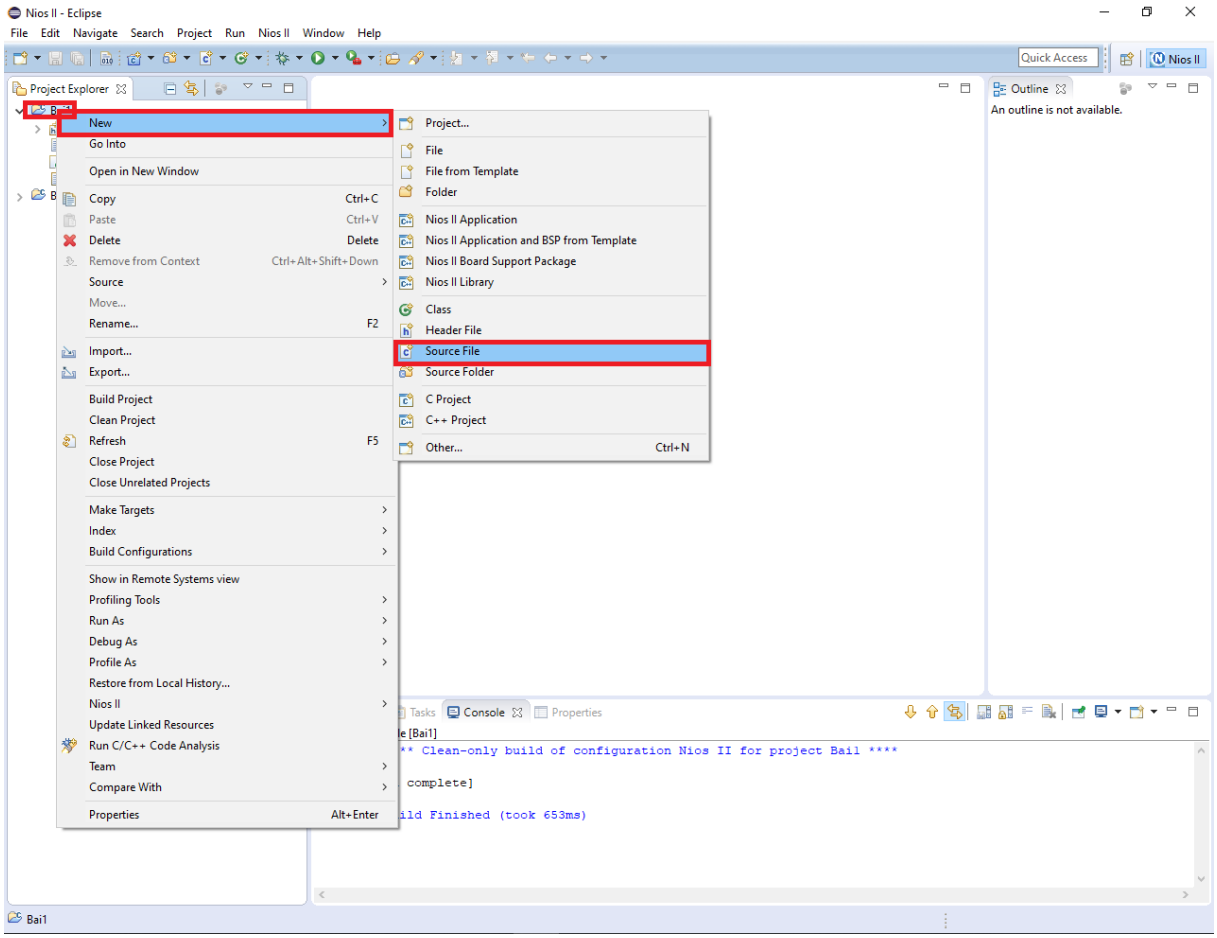
Hình 47. Cửa sổ Nios II Software Examples.

- Sau khi tạo thành công, sẽ được như hình 48.



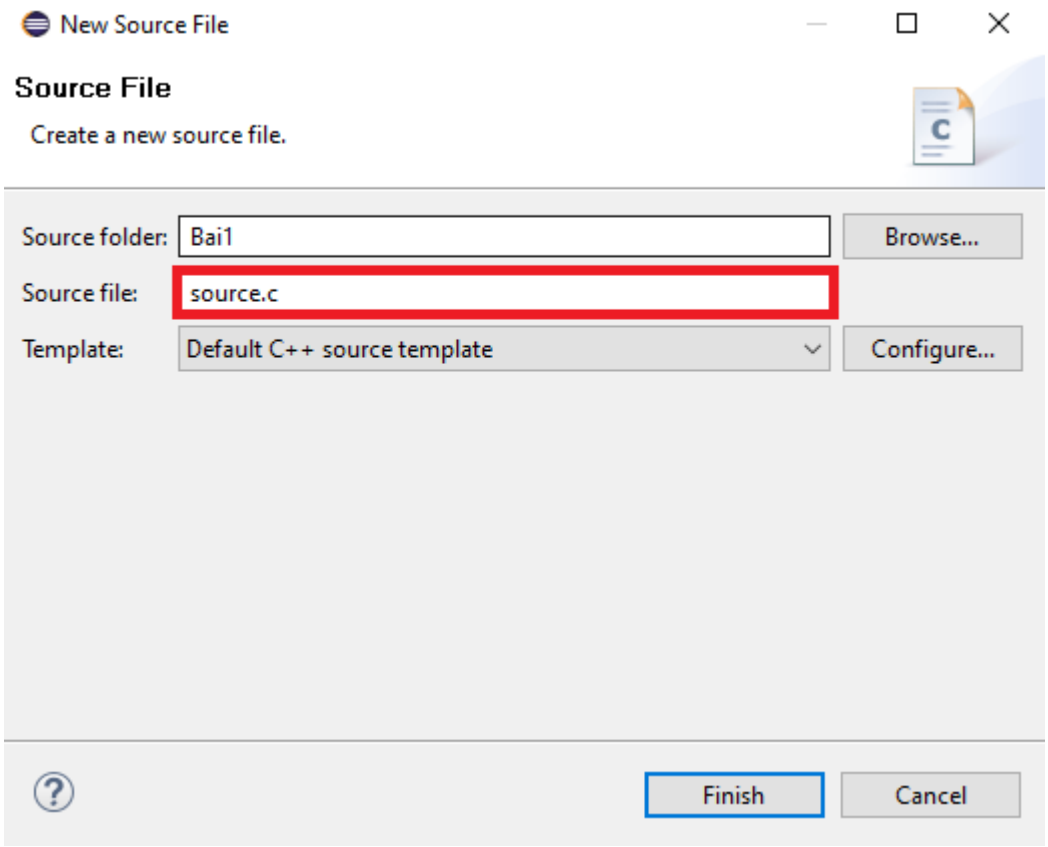
Hình 48. Kết quả sau khi tạo project.

- Tiến hành thêm file source.c như hình 49.



Hình 49. Thêm file souce.c

- Đặt tên là file là source.c như hình 50.

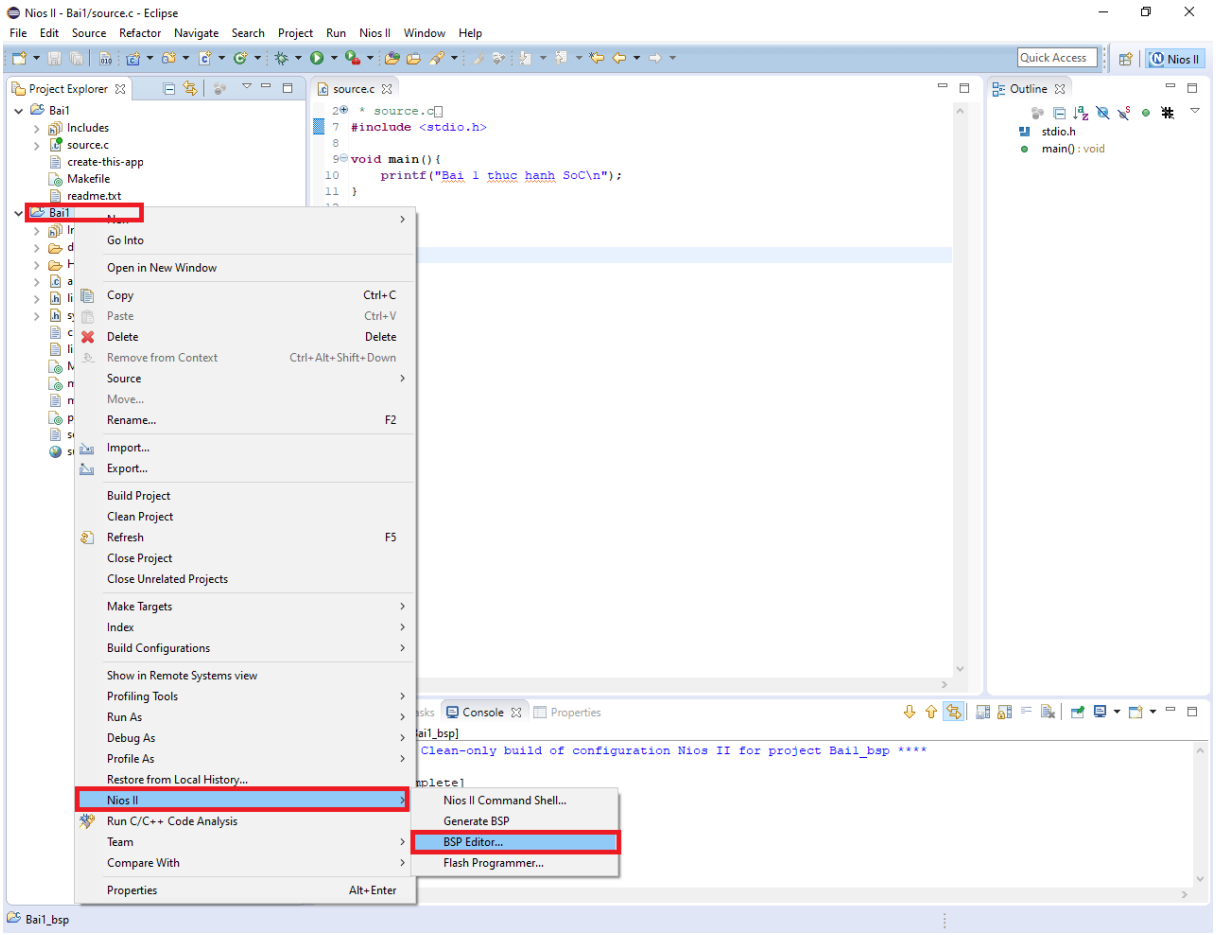


Hình 50. Đặt tên file là sourc.c.

- Tiến hành gõ đoàn code sau vào trong file **souce.c** lưu lại.

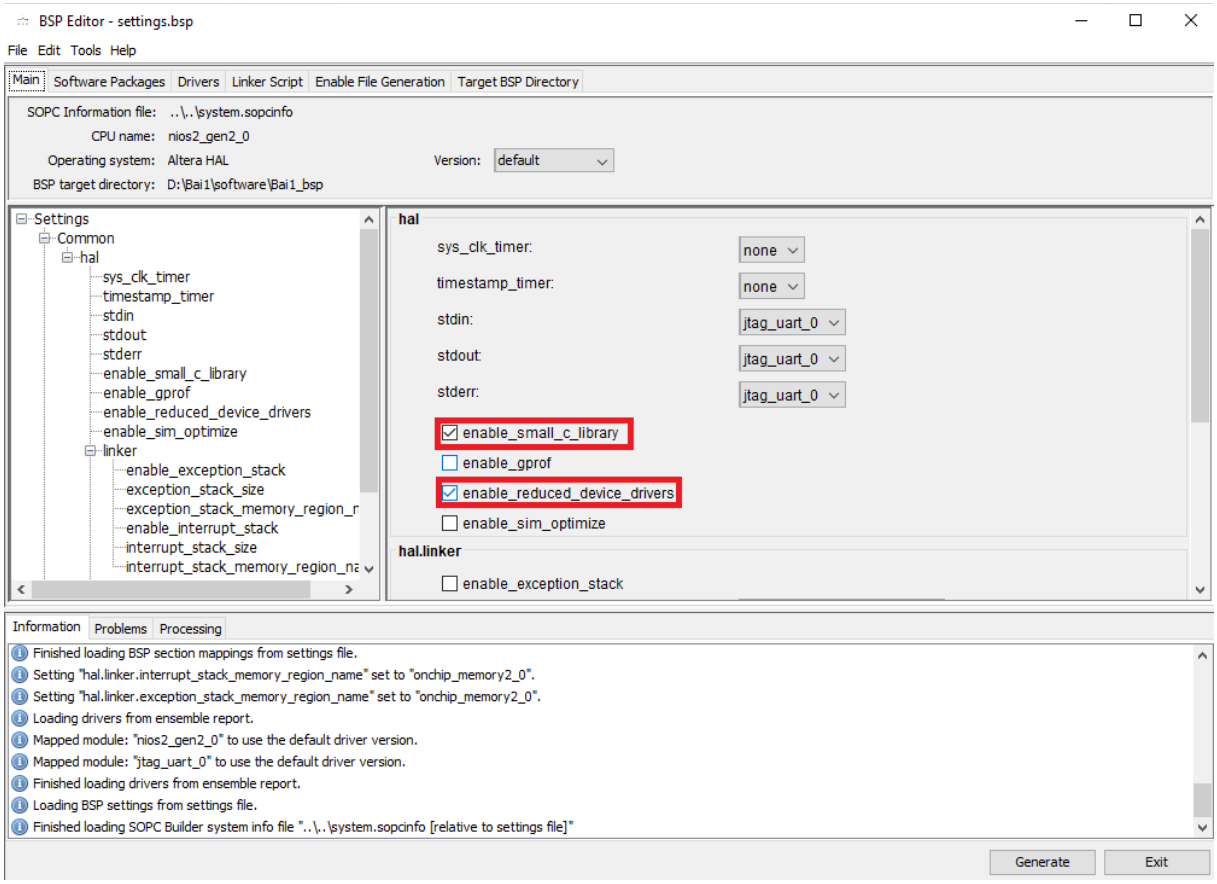
```
#include <stdio.h>

void main(){
      printf("Bai 1 thuc hanh SoC\n");
}
```

- Cấu hình lại bsp, chọn chuột phải vào Bai1_bsp, chọn **Nios II → BSP Editor**… như hình 51.



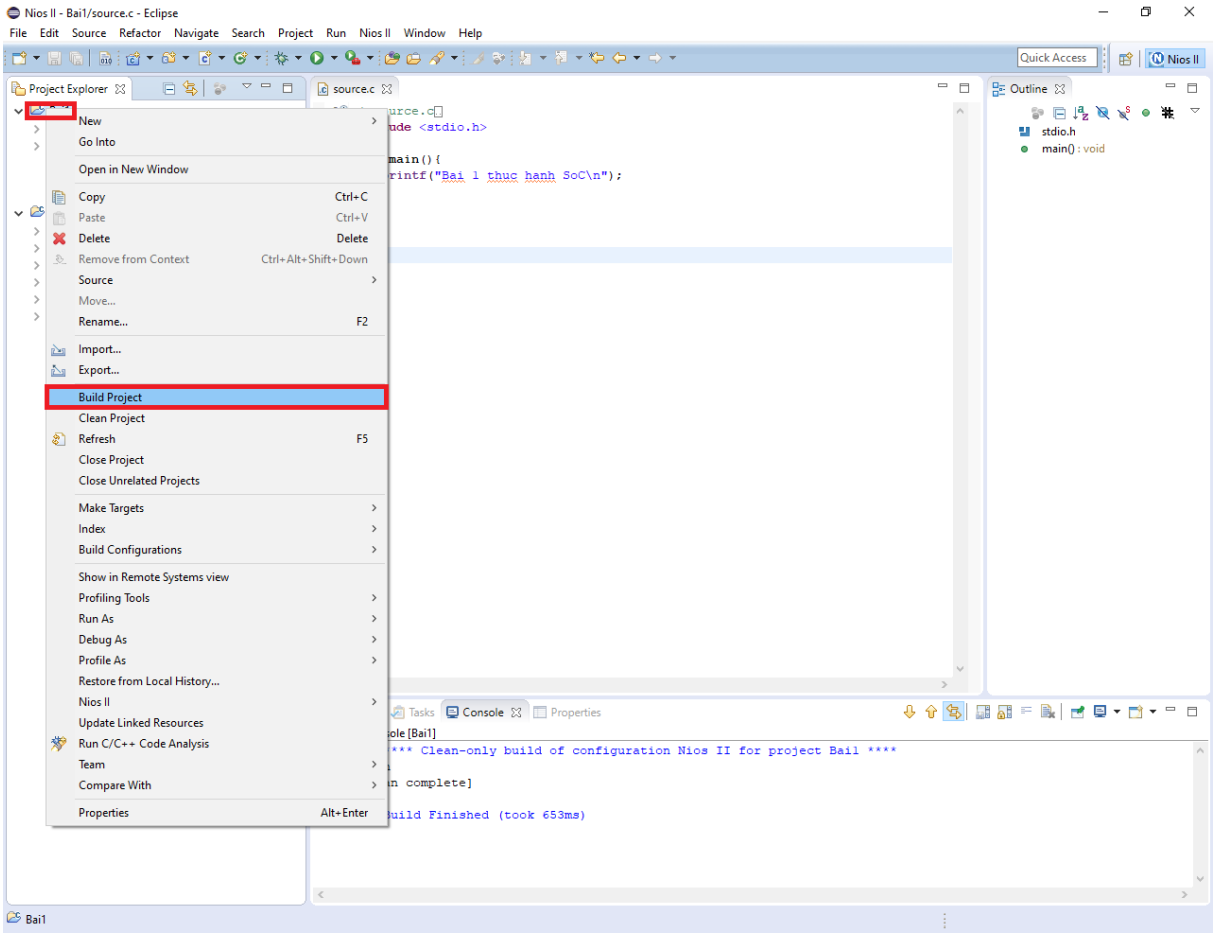
Hình 51. Cấu hình BSP.

- Chọn như hình 52 sau đó chọn **Generate** rồi chọn **Exit**.



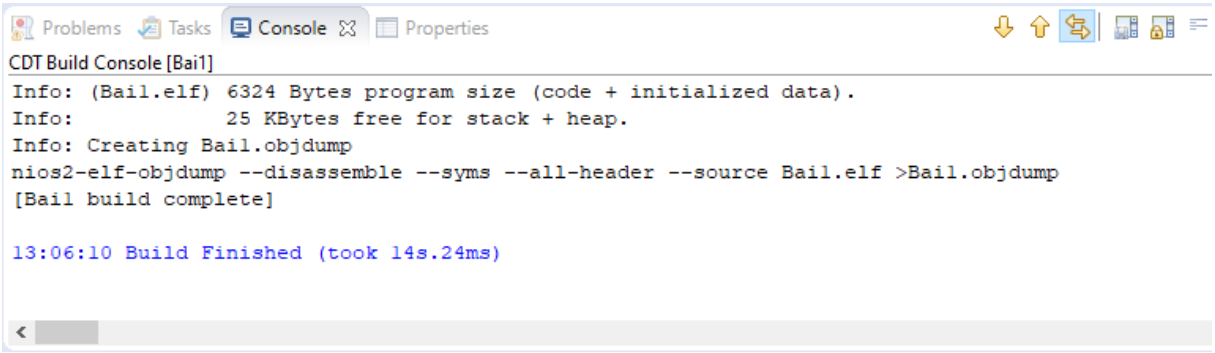
Hình 52. Cấu hình BSP.

- Tiến hành Build project, chọn chuột phải vào **Bai1 → Build Project** như hình 53.



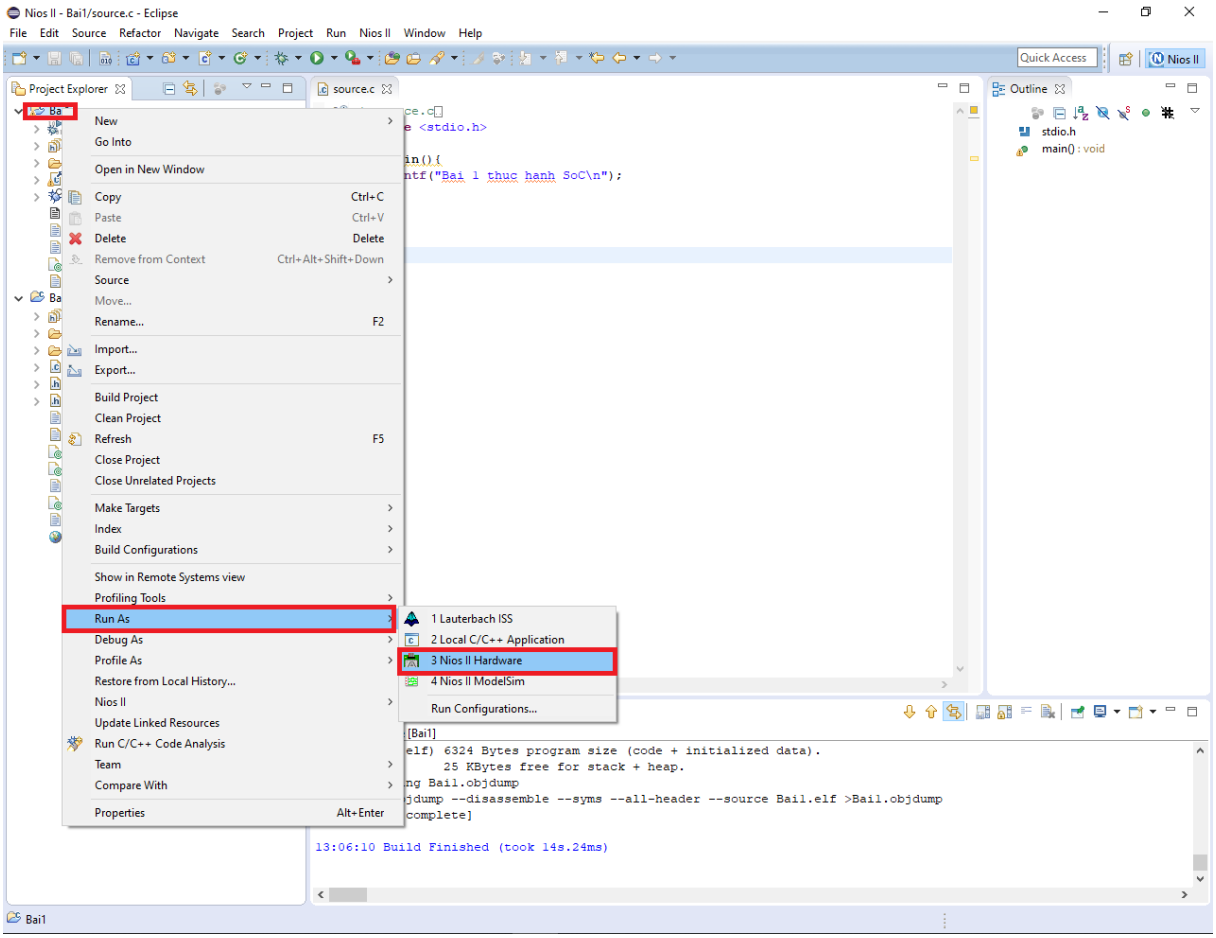
Hình 53. Build project.

- Khi kết quả không có lỗi như hình 54, tiến hành nạp phần mềm xuống board như hình 55.



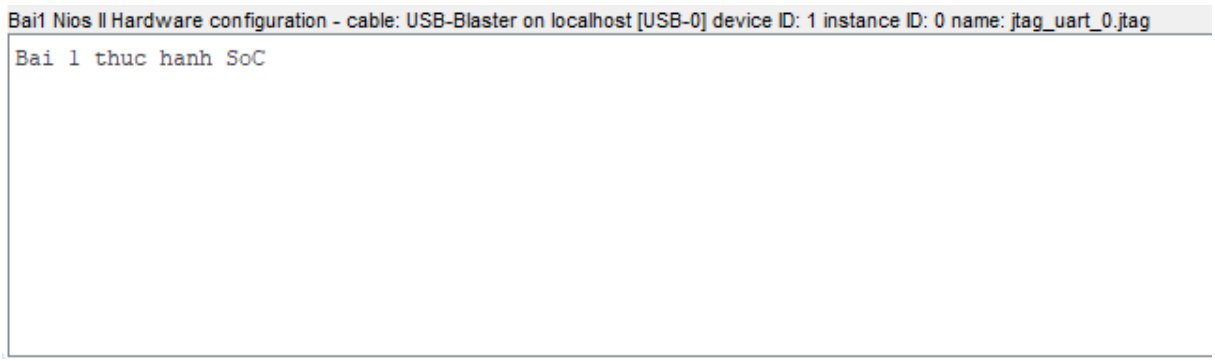
Hình 54. Kết quả build project.

- Nạp phần mềm xuống board bằng cách chọn chuột phải ở **Bai1 → Run as → Nios II Hardware**, hình 55.



Hình 55. Nạp phần mềm xuống board.

- Kết quả của sau khi nạp phần mềm được thể hiện như hình 56.



Hình 56. Kết quả chạy trên board hiển thị trên console.

BÀI TẬP CHUẨN BỊ Ở NHÀ

Bài 1. Cho biết thệt hống SoC là gì?

Bài 2. Dựa vào tài liệu tham khảo [2], cho biết chức năng của mô đun Jtag-uart.

BÁO CÁO THỰC HÀNH

Bài 1. Tìm hiểu cách Debug và cho biết ý nghĩa của và phím tắt của các biểu tượng sau trong cửa sổ debug của NIOS II:

Bài 2: Tiến hành debug đoạn code sau:

```
#include <stdio.h>

void main(){
	int a;
	int *b;
	int c;
	a  = 3;
	b = &a;
	*b = 5;
	c = a + 4;
}
```

TÀI LIỆU THAM KHẢO

[1] DE2_115_User_Manual.

[2] Embedded Peripherals IP User Guide.

[3] Nios II Processor Reference.

[4] Avalon Interface Specification.