

Tutorial

FPGA Tutorial for Lego Ecar

By Karan Savant

This tutorial provides comprehensive information that will help you understand how to create a FPGA based SOPC system implementing on your FPGA development board and run software upon it.

1.1 Required Features

The Nios II processor core is a soft-core central processing unit that you could program onto an Altera field programmable gate array (FPGA). This tutorial illustrates you to the basic flow covering hardware creation and software building. You are assumed to have the latest Quartus II and NIOS II EDS software installed and quite familiar with the operation of Windows OS. If you use a different Quartus II and NIOS II EDS version, there will have some small difference during the operation. You are also be assumed to possess a DE2-115 development board (other kinds of dev.

Board based on Altera FPGA chip also supported).

The example NIOS II standard hardware system provides the following necessary components:

- * Nios II processor core, that's where the software will be executed
- * On-chip memory to store and run the software
- * JTAG link for communication between the host computer and target hardware (typically using a USB-Blaster cable)
- * LED peripheral I/O (PIO), be used as indicators

1.2 Creation of Hardware Design

This section describes the flow of how to create a hardware system including SOPC feature.

1. Launch Quartus II then select **File->New Project Wizard**, start to create a new project. See Figure 1-1 and Figure 1-2.

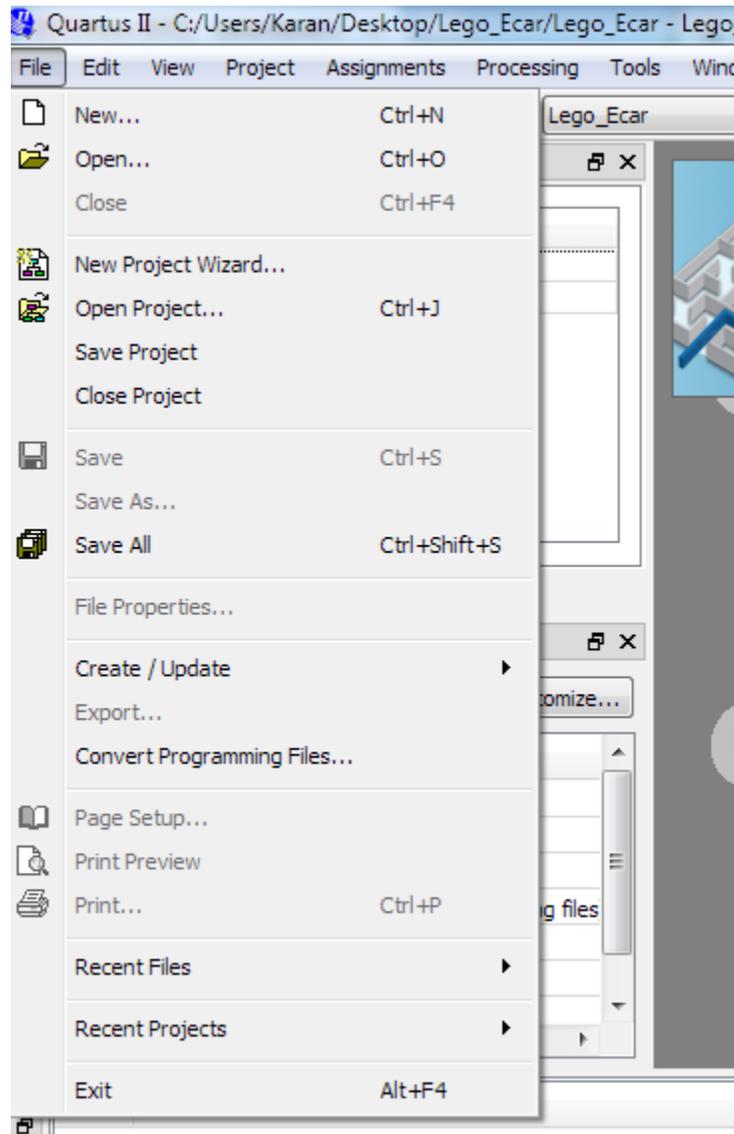


Figure 1-1 Start to Create a New Project

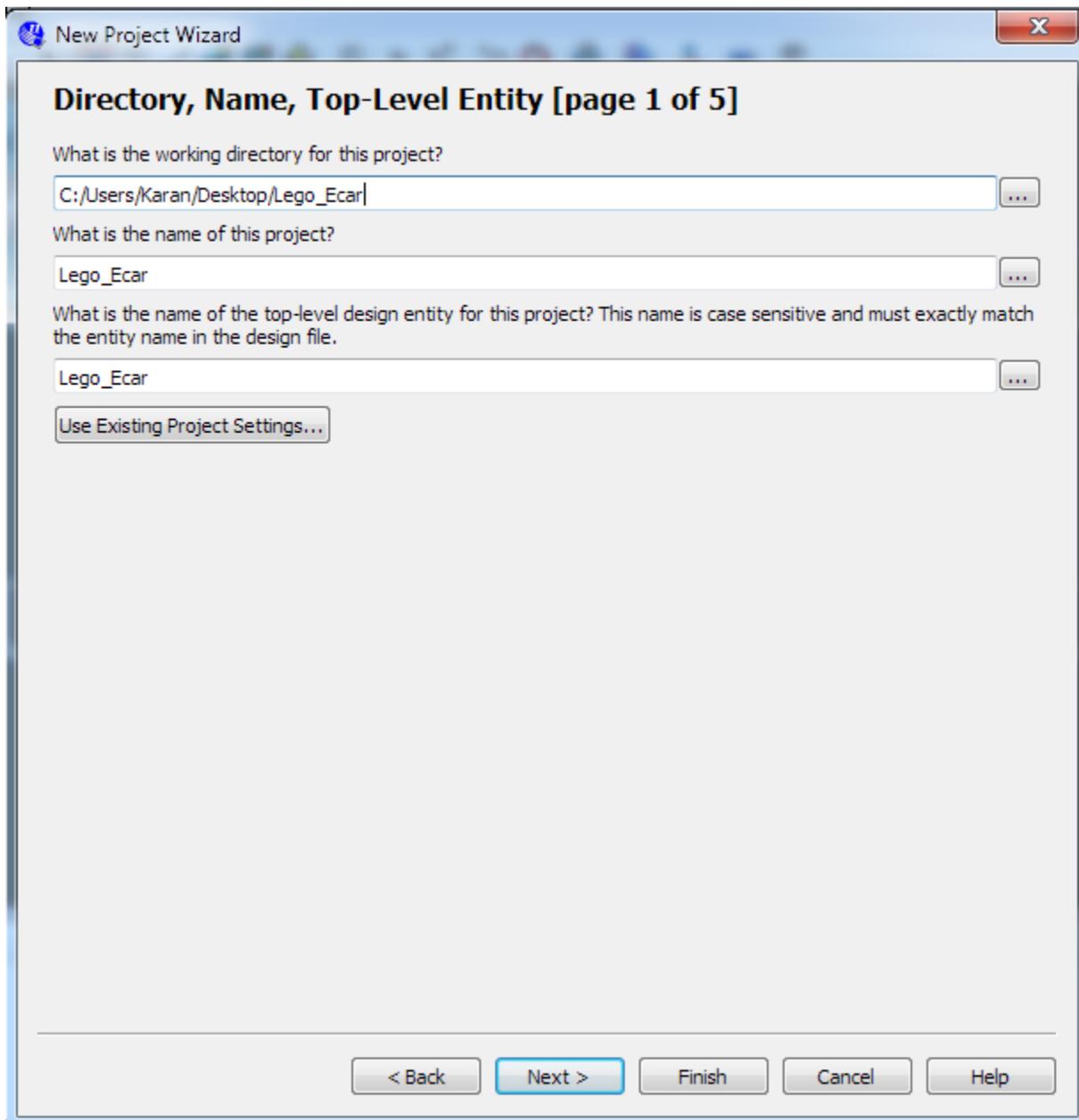
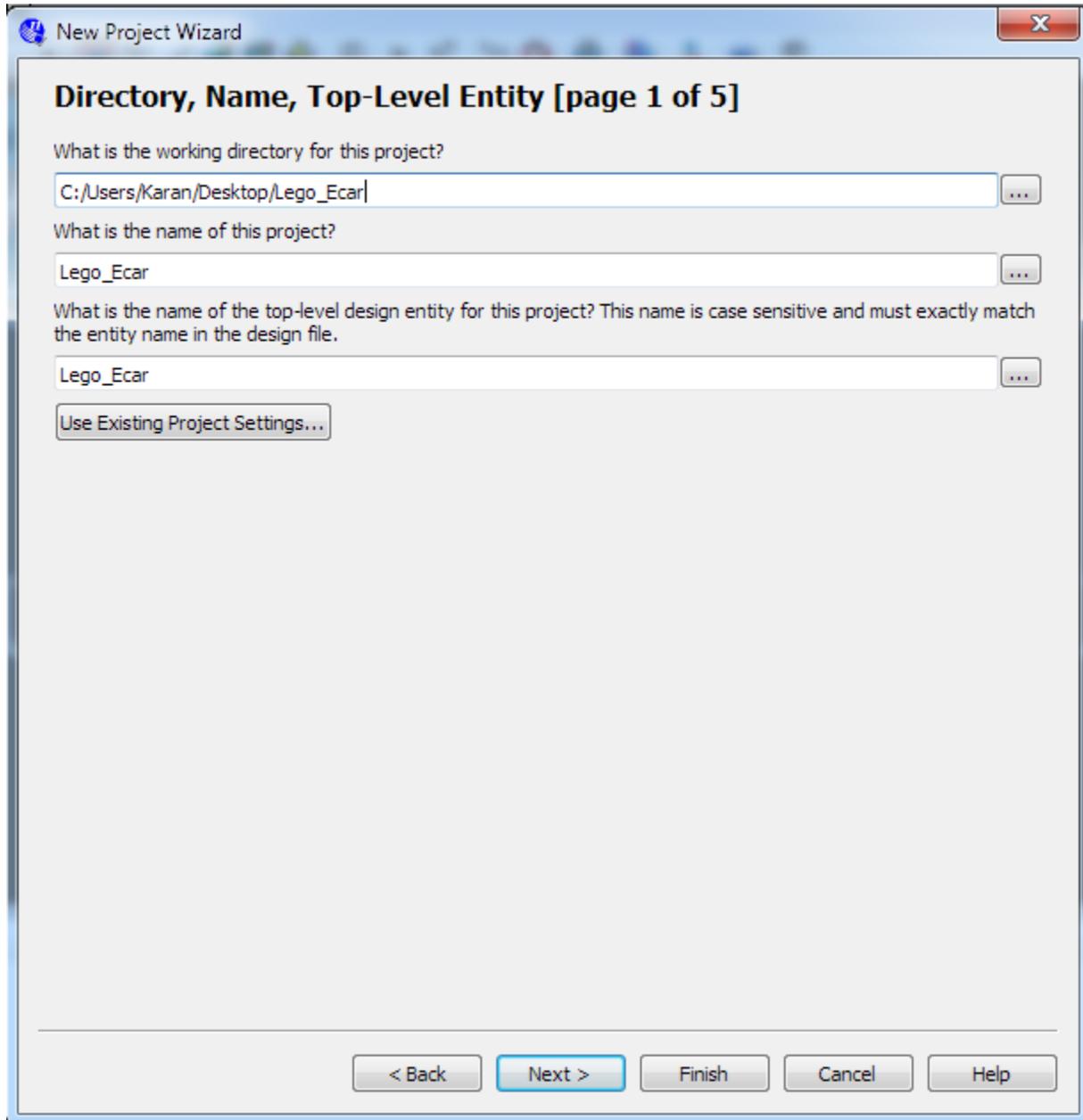


Figure 1-2 New Project Wizard

2. Choose a working directory for this project, type project name and top-level entity name as shown in Figure 1-3. Then click **Next**, you will see a window as shown in Figure 1-4.



New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

 ...

What is the name of this project?

 ...

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

 ...

< Back Next > Finish Cancel Help

Figure 1-3 Input the working directory, the name of project, top-level design entity

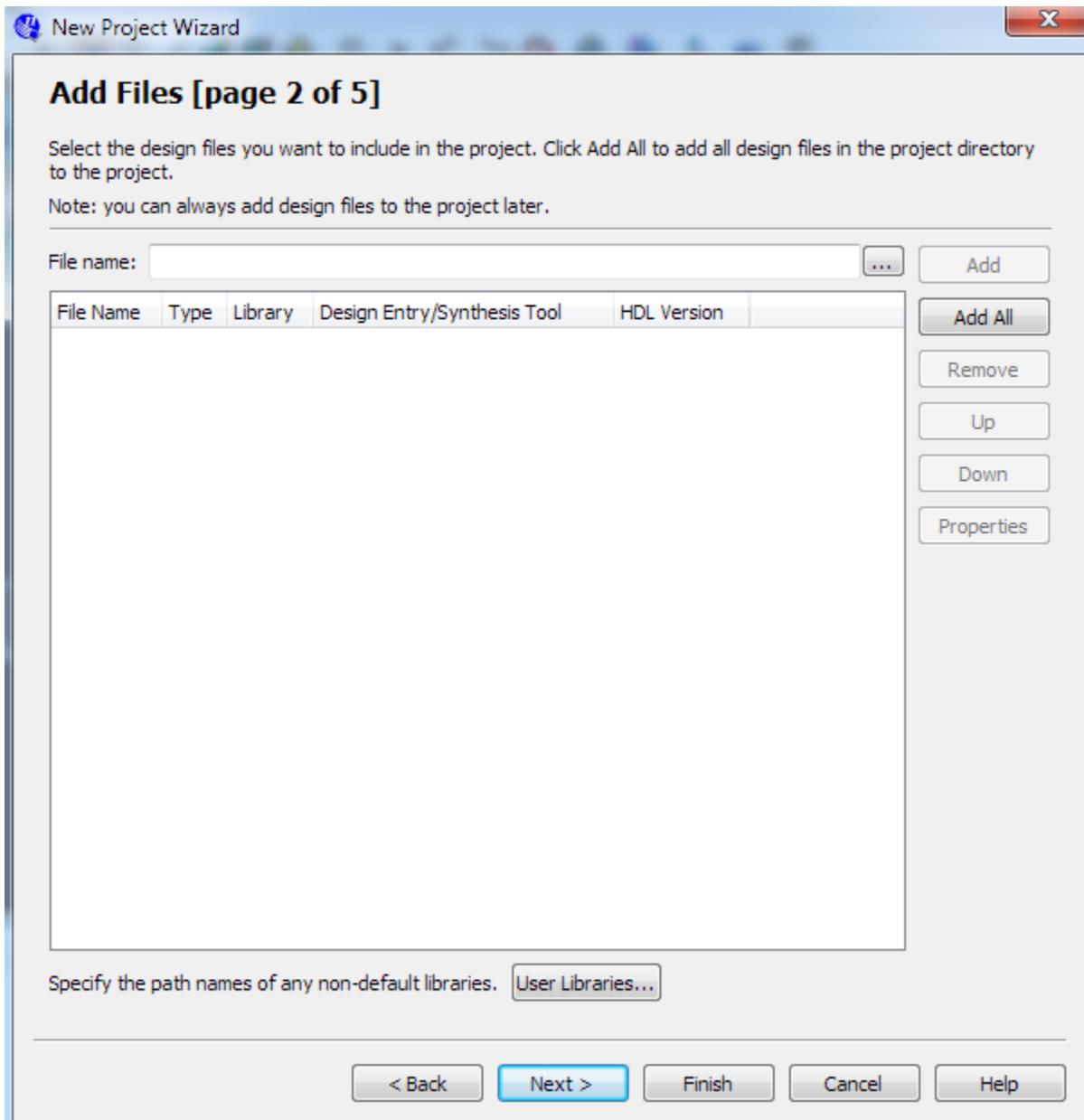


Figure 1-4 New Project Wizard: Add Files [page 2 of 5]

3. Click **Next** to next window. We choose device family and device settings. You should choose settings the same as the Figure 1-5. Then click **Next** to next window as shown in Figure 1-6.

New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family

Family: Cyclone IV E

Devices: All

Target device

Auto device selected by the Fitter

Specific device selected in 'Available devices' list

Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Speed grade: Any

Show advanced devices

HardCopy compatible only 

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits
EP4CE115F23I8L	1.0V	114480	281	3981312
EP4CE115F29C7	1.2V	114480	529	3981312
EP4CE115F29C8	1.2V	114480	529	3981312
EP4CE115F29C8L	1.0V	114480	529	3981312
EP4CE115F29C9L	1.0V	114480	529	3981312
EP4CE115F29I7	1.2V	114480	529	3981312
EP4CE115F29I8L	1.0V	114480	529	3981312

Companion device 

HardCopy:

Limit DSP & RAM to HardCopy device resources

< Back Next > Finish Cancel Help

Figure 1-5 New Project Wizard: Family & Device Settings [page 3 of 5]

4. Click **Next** and will see a window as shown in Figure 1-7. Figure 1-7 is a summary about our new project. Click **Finish** to finish new project. Figure 1-8 show a new complete project.

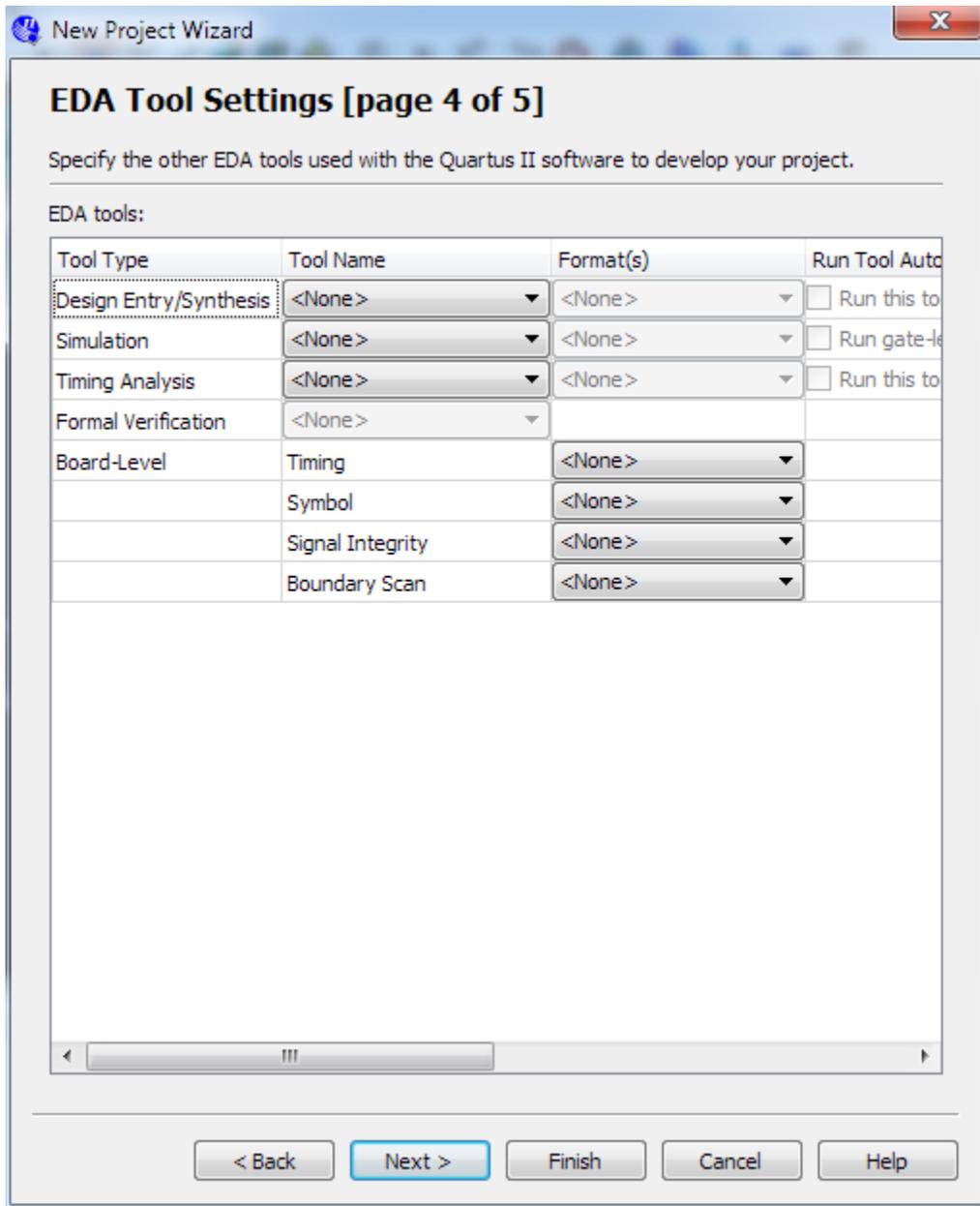


Figure 1-6 New Project Wizard: EDA Tool Settings [page 4 of 5]

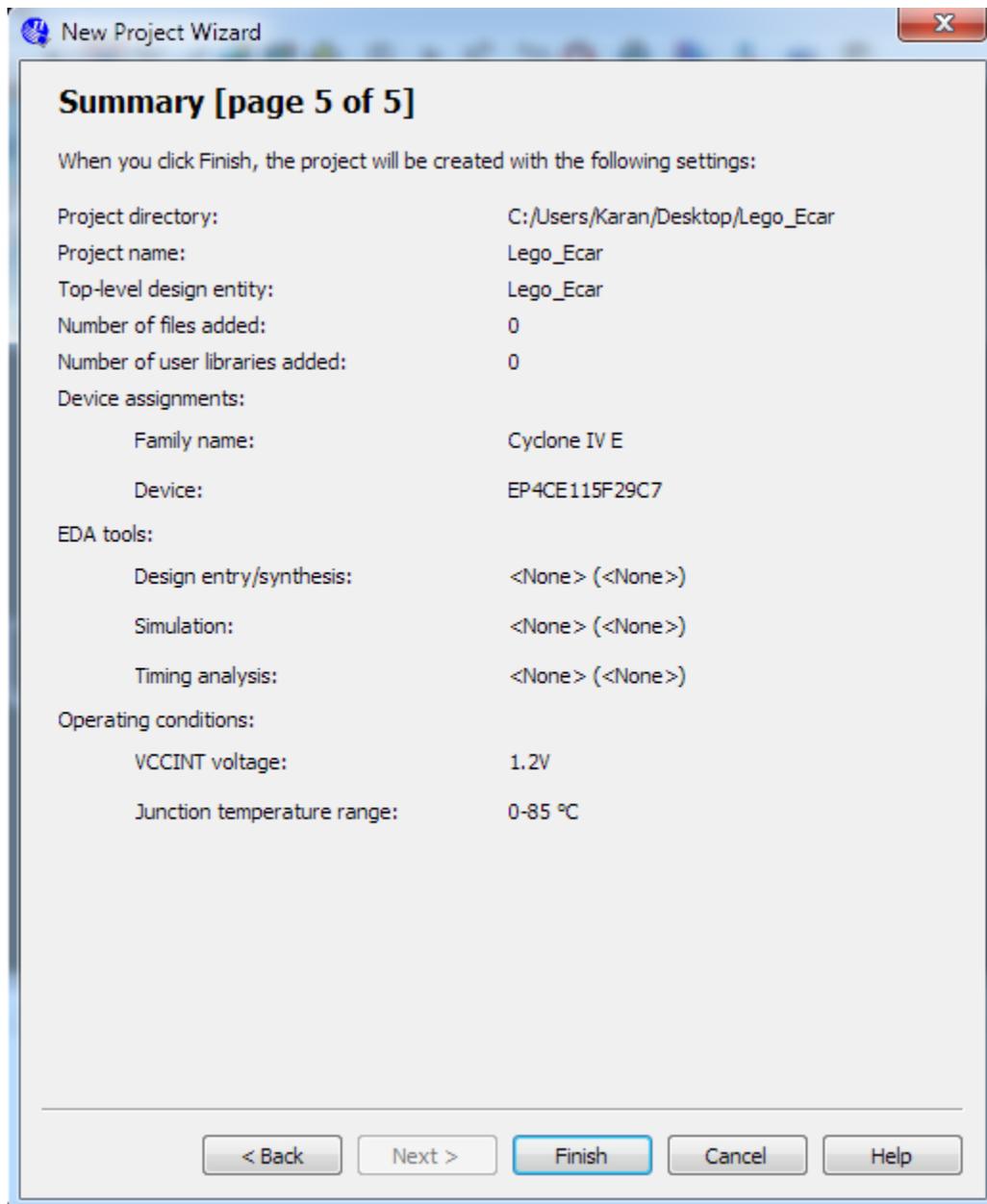


Figure 1-7 New Project Wizard: Summary [page 5 of 5]

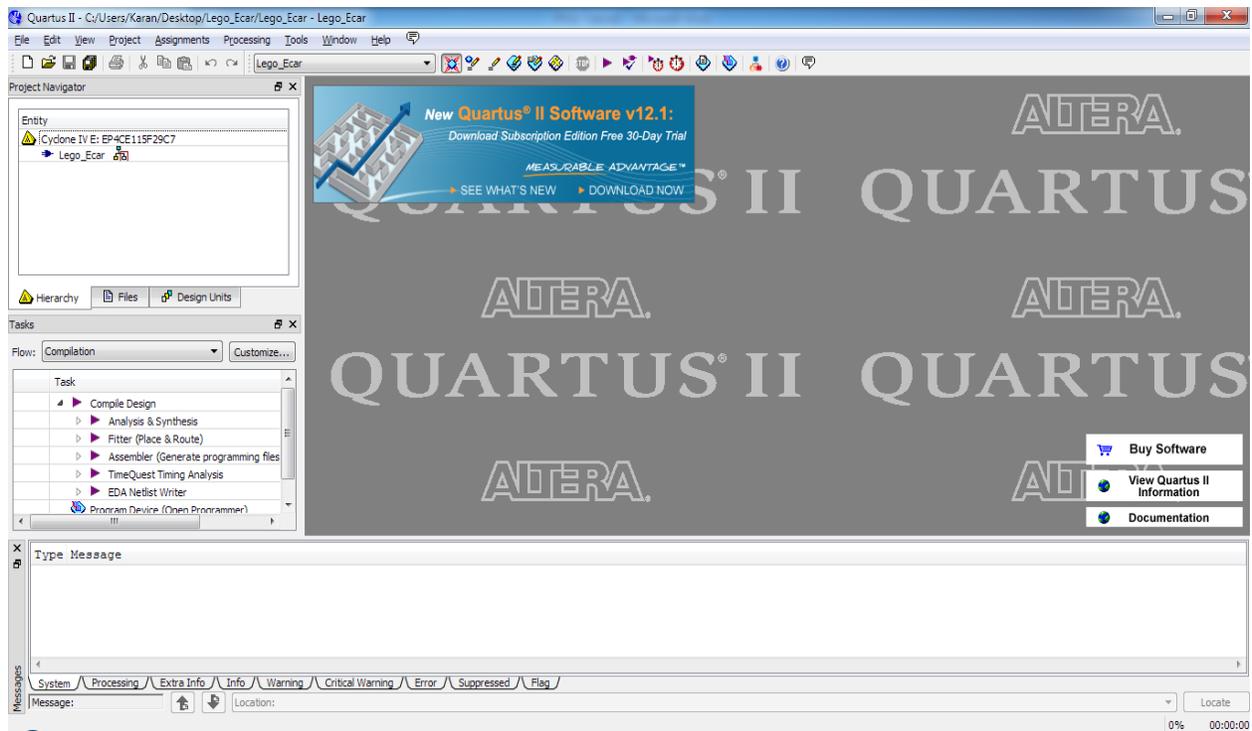


Figure 1-8 A New Complete Project

5. Choose **Tools > SOPC Builder** to open new SOPC system wizard .See Figure 1-9 and Figure 1-10.

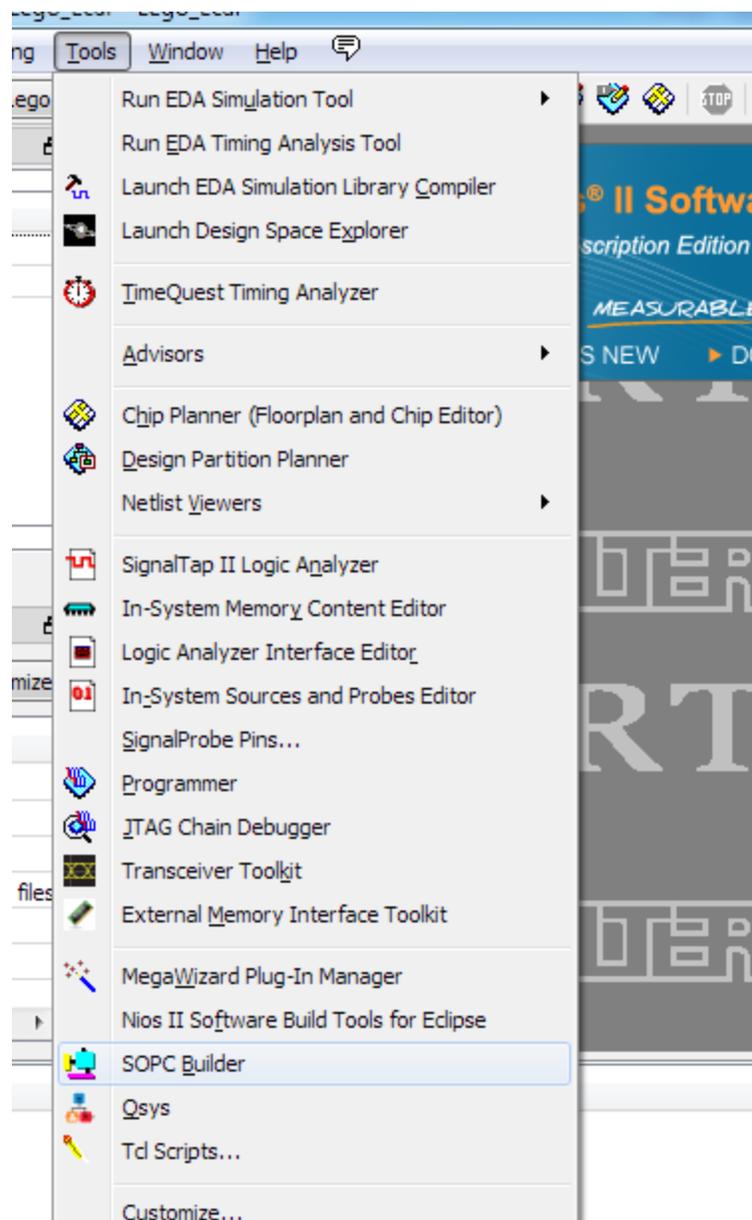


Figure 1-9 SOPC Builder Menu

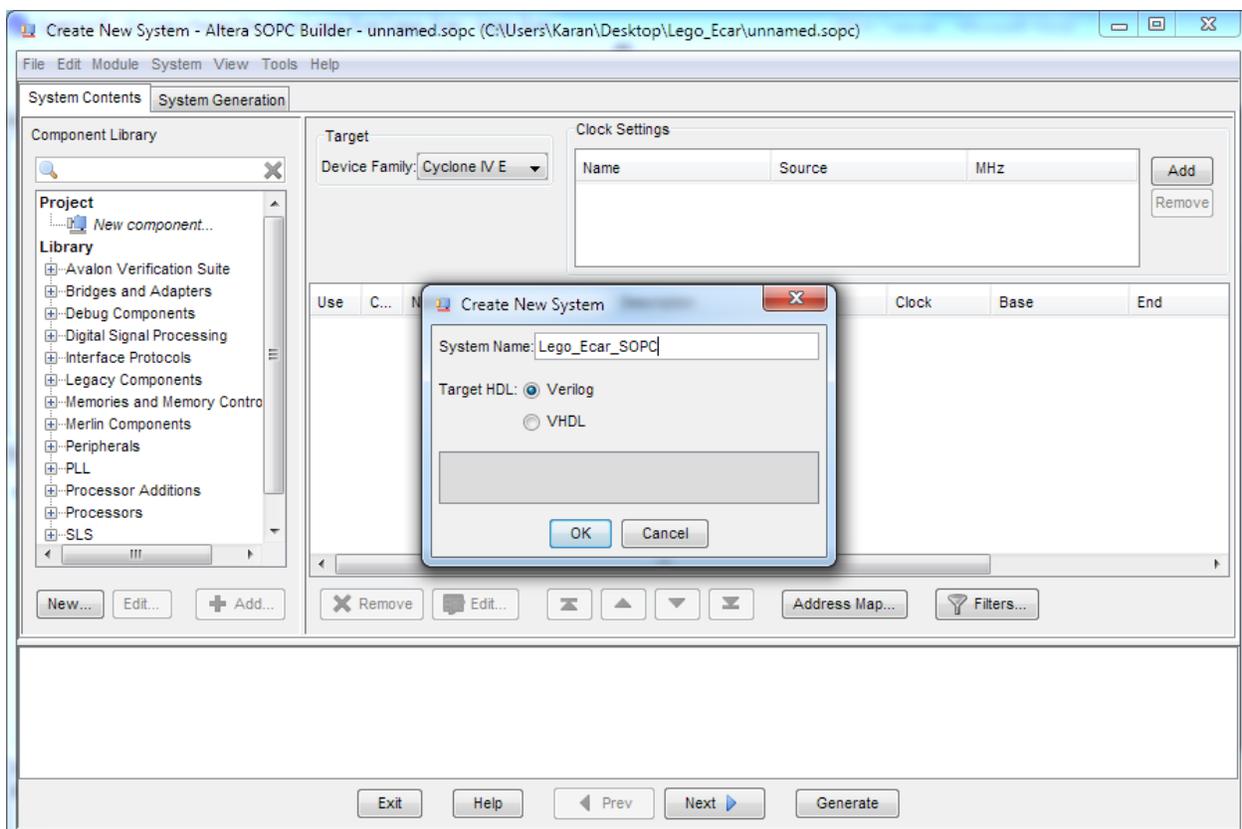
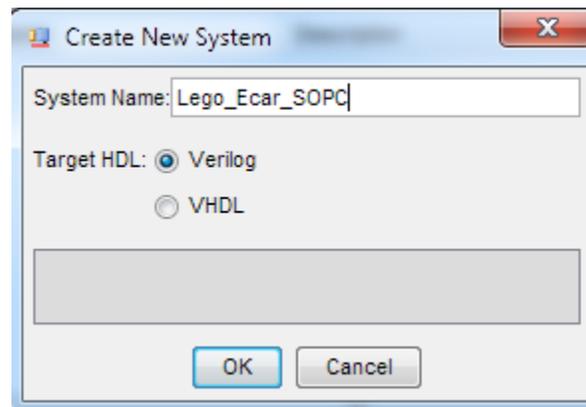
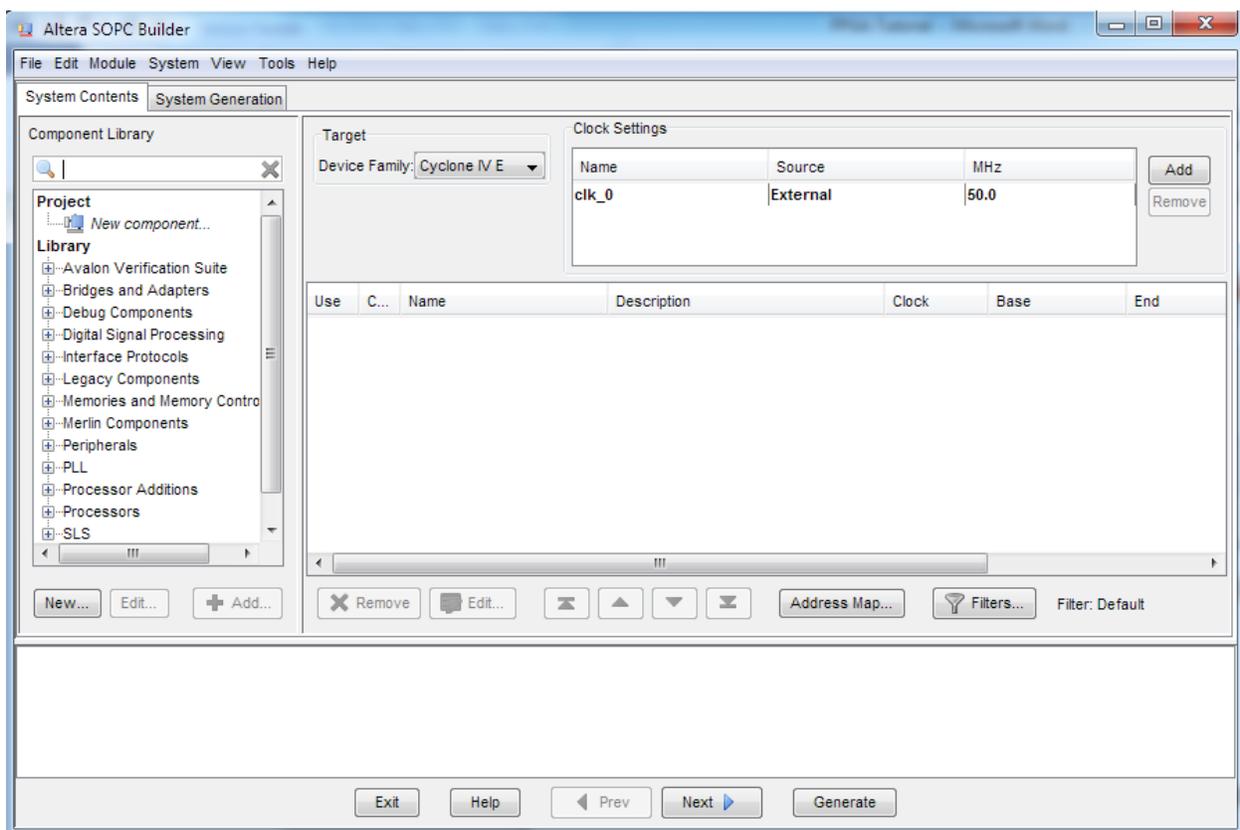


Figure 1-10 Create New SOPC System [0]

6. Rename **System Name** as shown in Figure 1-11. Click **OK** and you will see a window as shown in Figure 1-12.

**Figure 1-11 Create New System [1]****Figure 1-12 Create New System [2]**

7. Click the **Name** of the **Clock Settings** table, rename `clk_0` to `clk_50`. Press Enter to complete the update. See Figure 1-13.

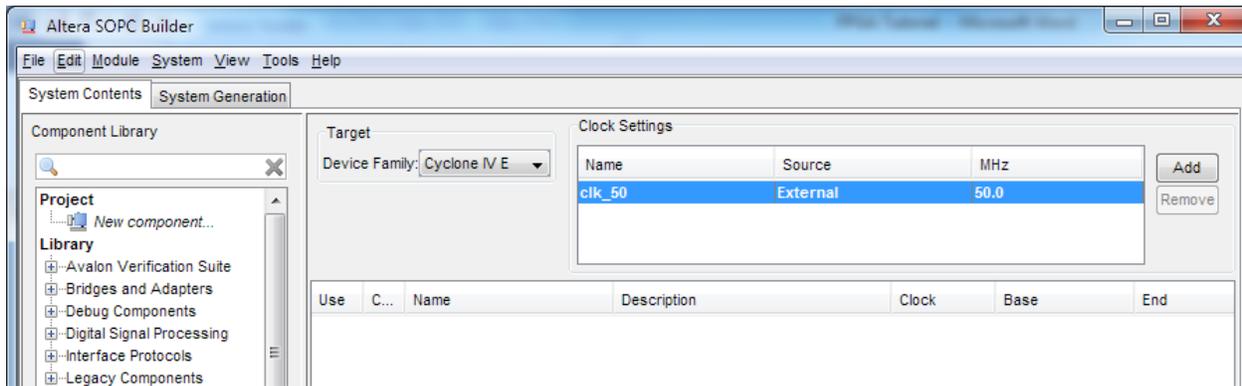


Figure 1-13 Rename Clock Name

8. Choose **Library > Processors > Nios II Processor** to open wizard of adding cpu component. See Figure 1-14 and Figure 1-15.

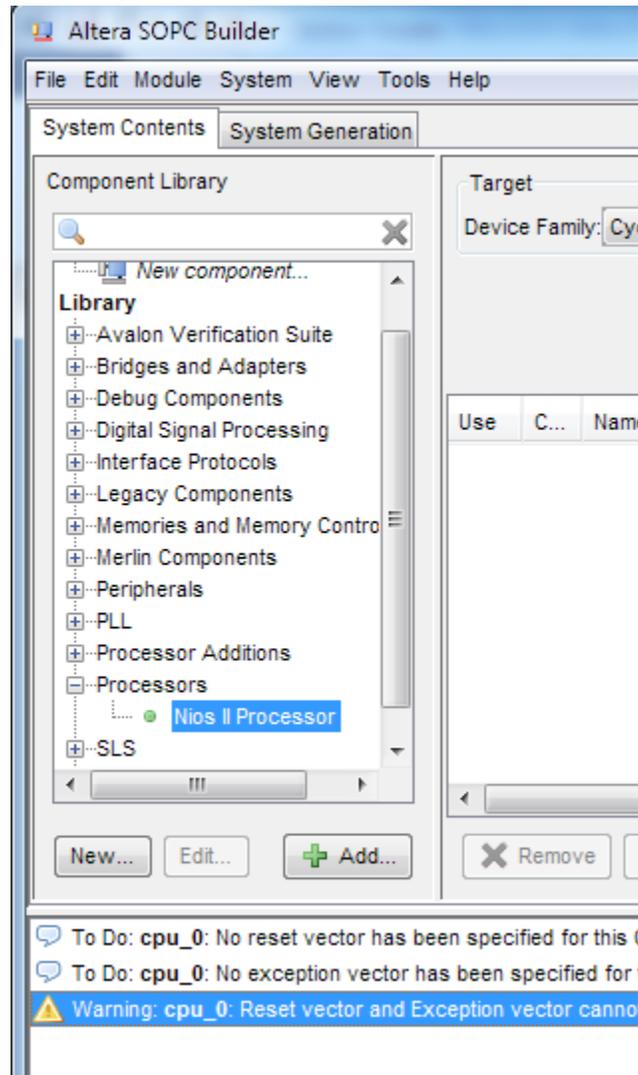


Figure 1-14 Add Nios II Processor

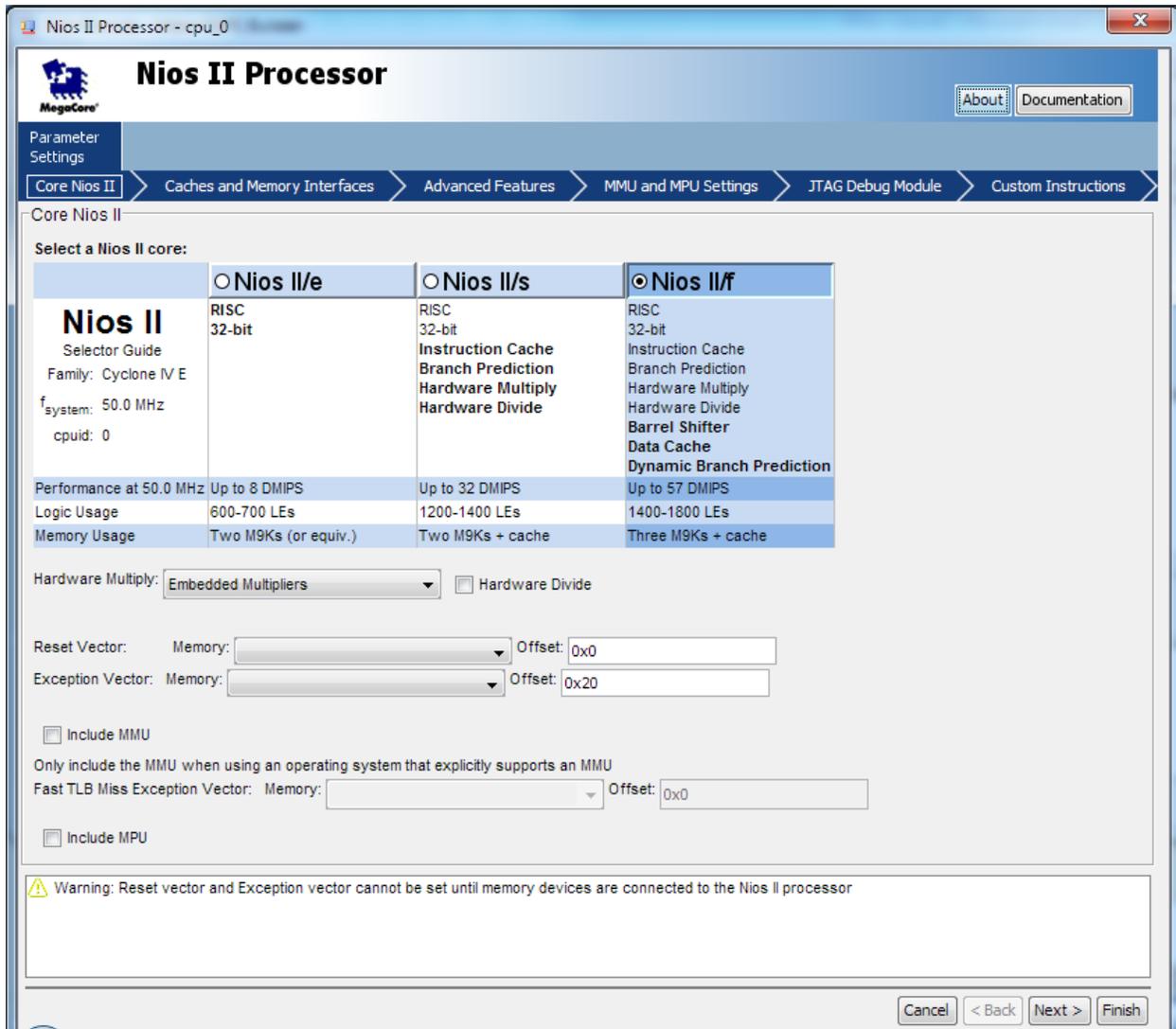


Figure 1-15 Nios II Processor

9. Click Finish to return to main window as shown in Figure 1-16.

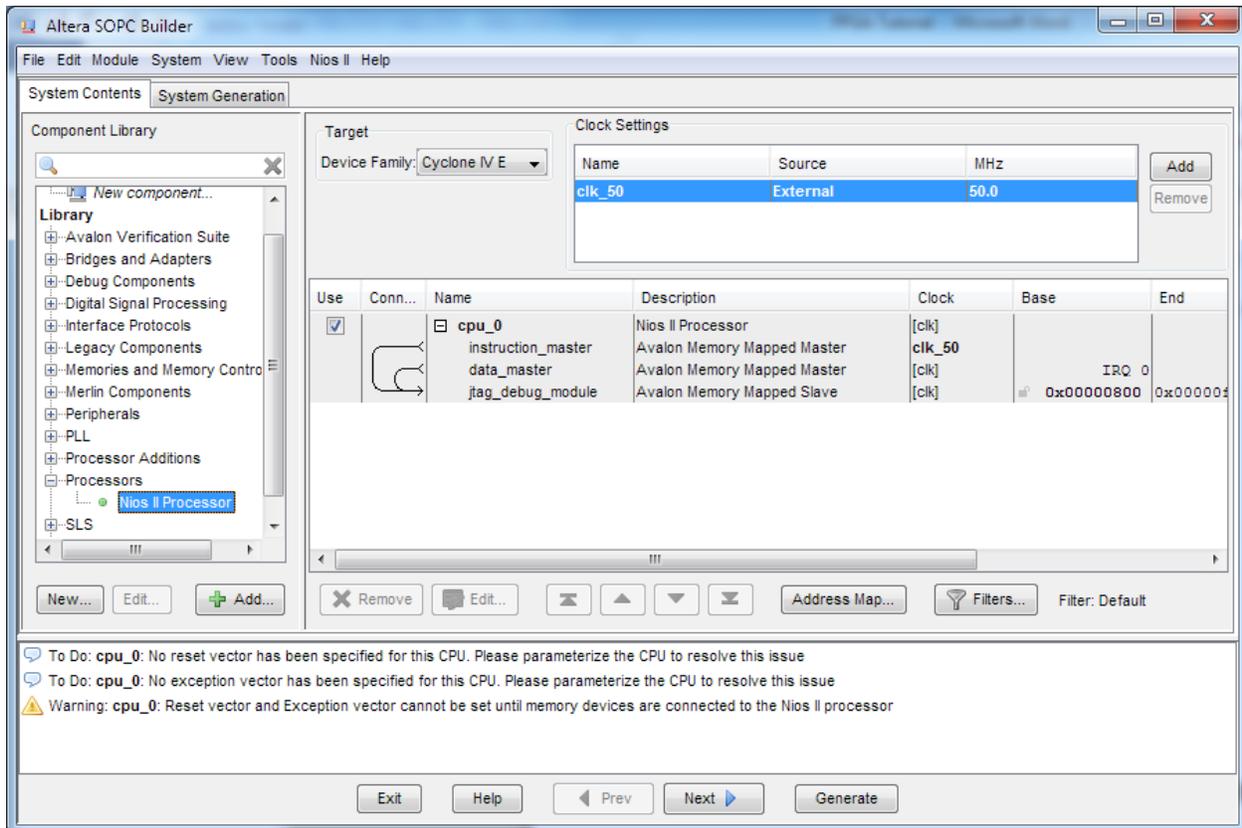


Figure 1-16 Add Nios II CPU completely

10. Choose **cpu_0** and right-click then choose **rename**, after this, you can update **cpu_0** to **cpu**. See Figure 1-17 and Figure 1-18.

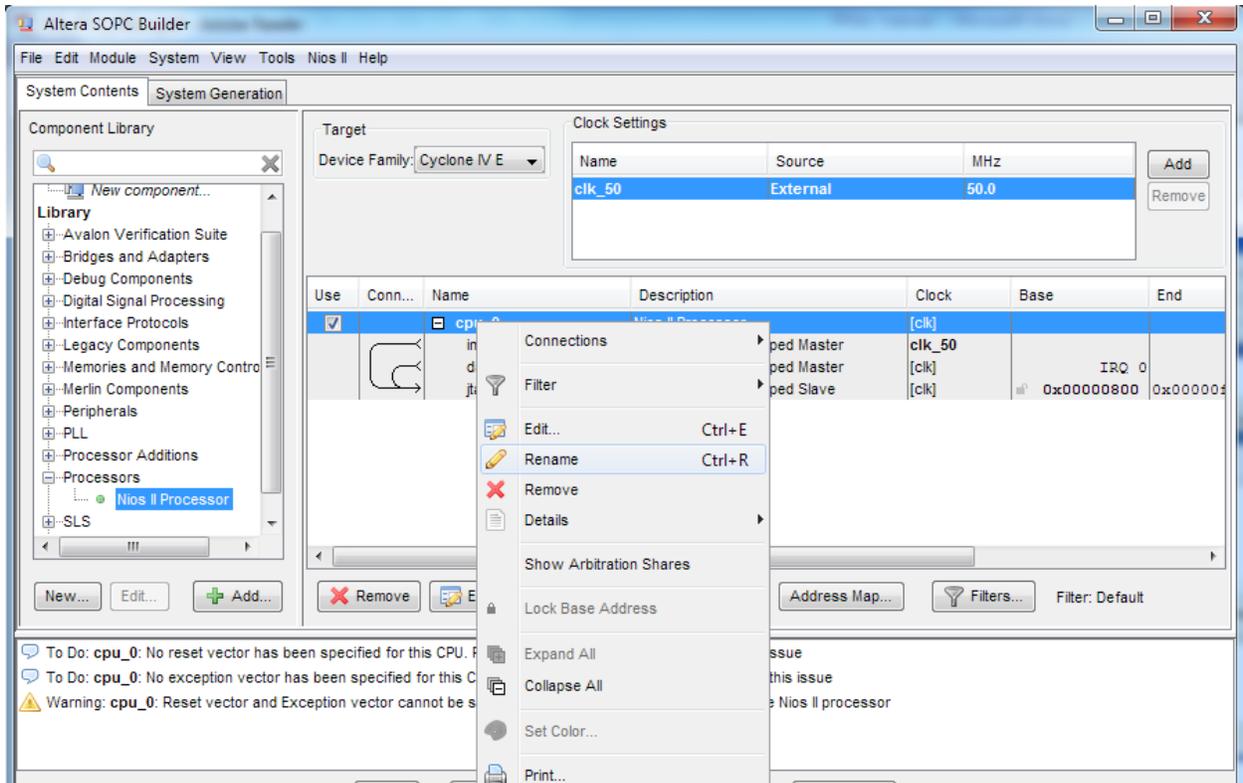


Figure 1-17 Rename CPU name (1)

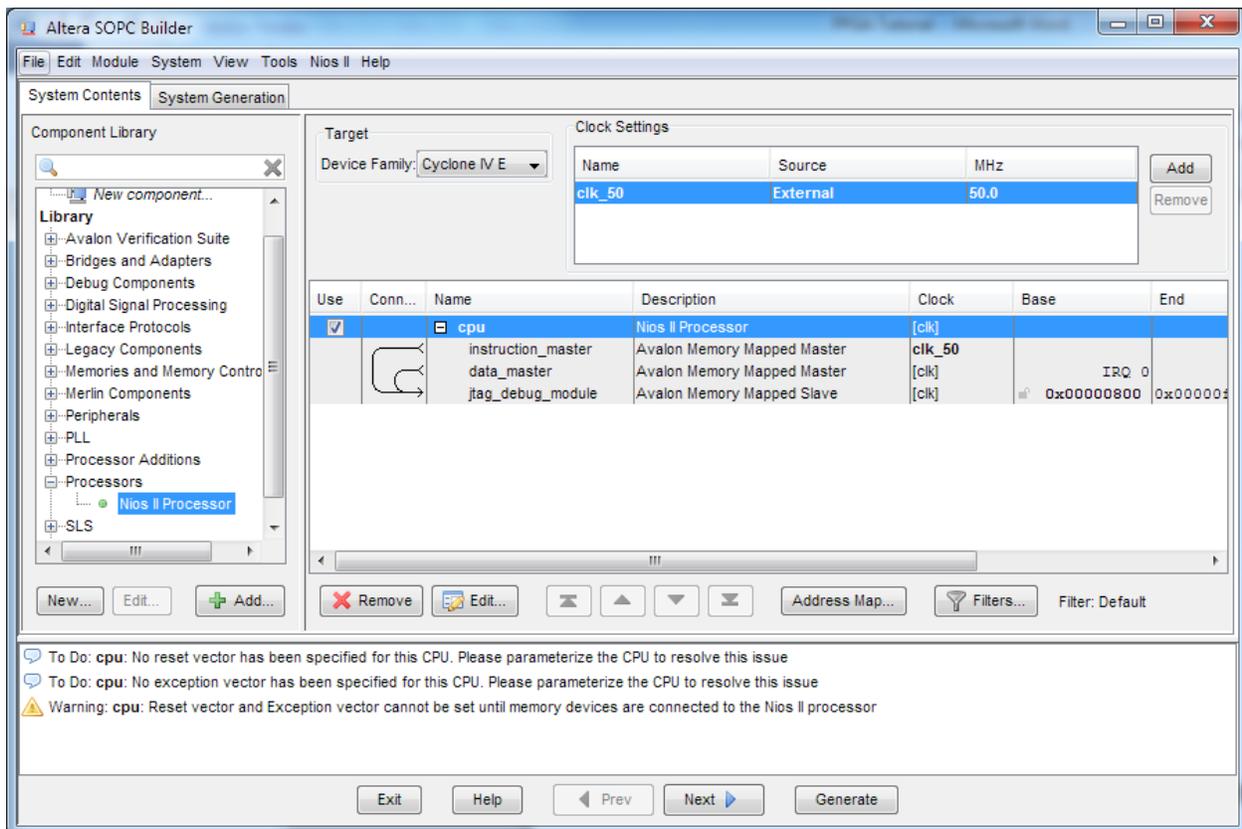


Figure 1-18 Rename CPU Name (2)

11. Choose **Library > Interface Protocols > Serial > JTAG UART** to open wizard of adding JTAG UART. See Figure 1-19 and Figure 1-20.

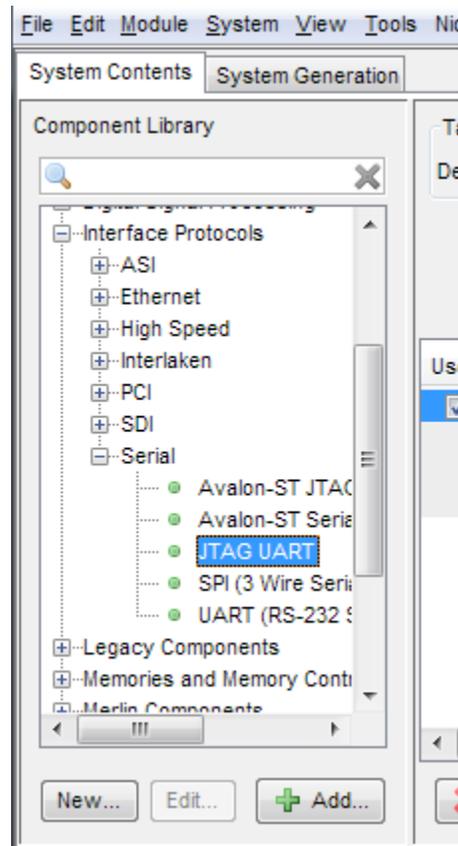


Figure 1-19 Add JTAG UART (1)

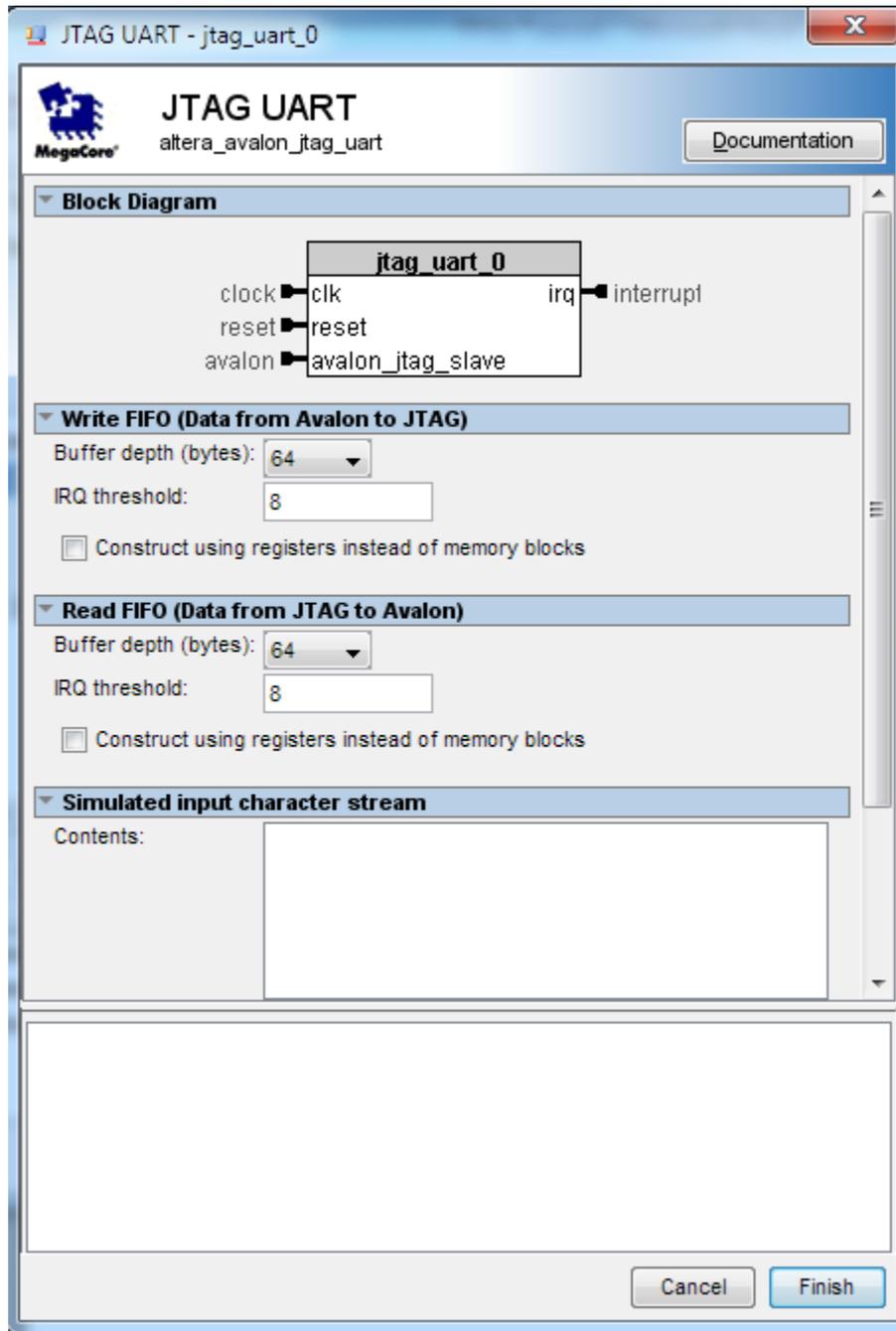


Figure 1-20 JTAG UART

12. Click Finish to close the wizard and return to the window as shown in Figure 1-21.

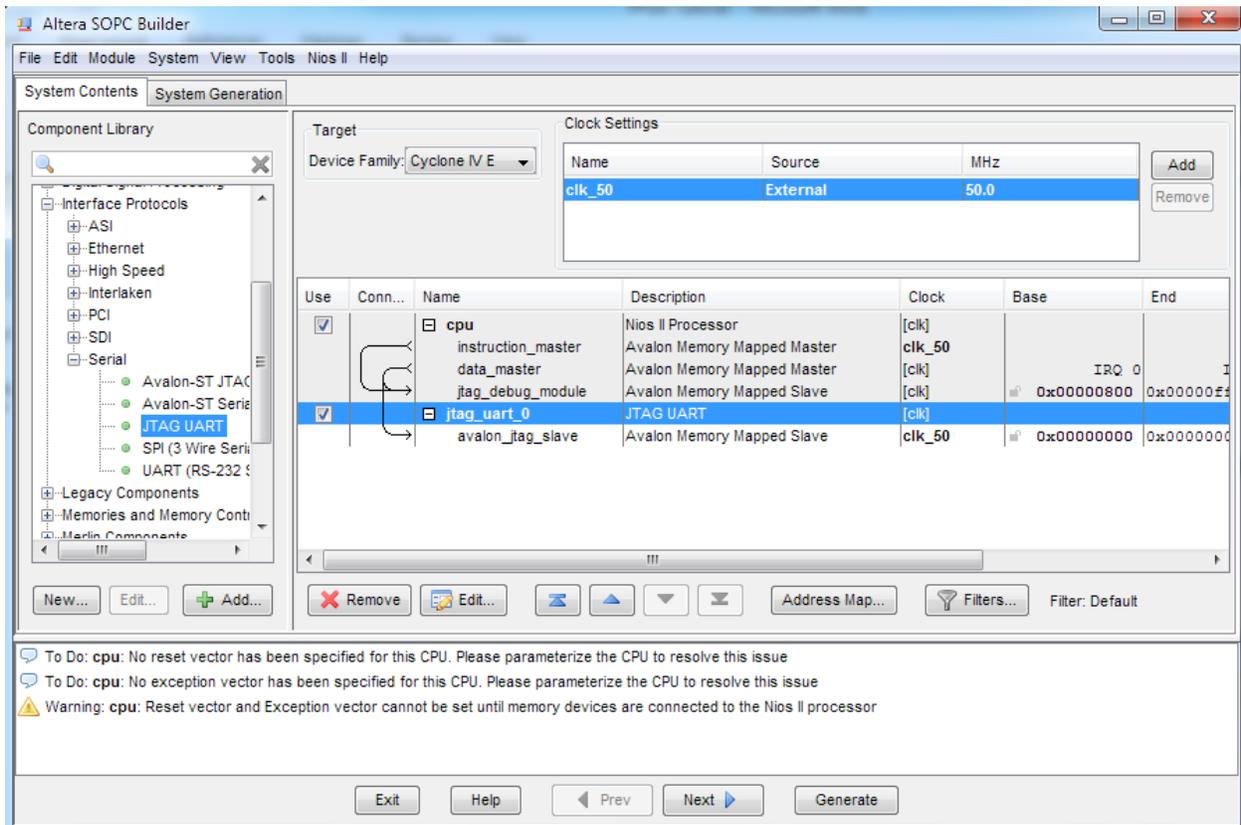


Figure 1-21 JTAG UART

13. Choose jtag_uart_0 and rename it to jtag_uart as shown in Figure 1-22.

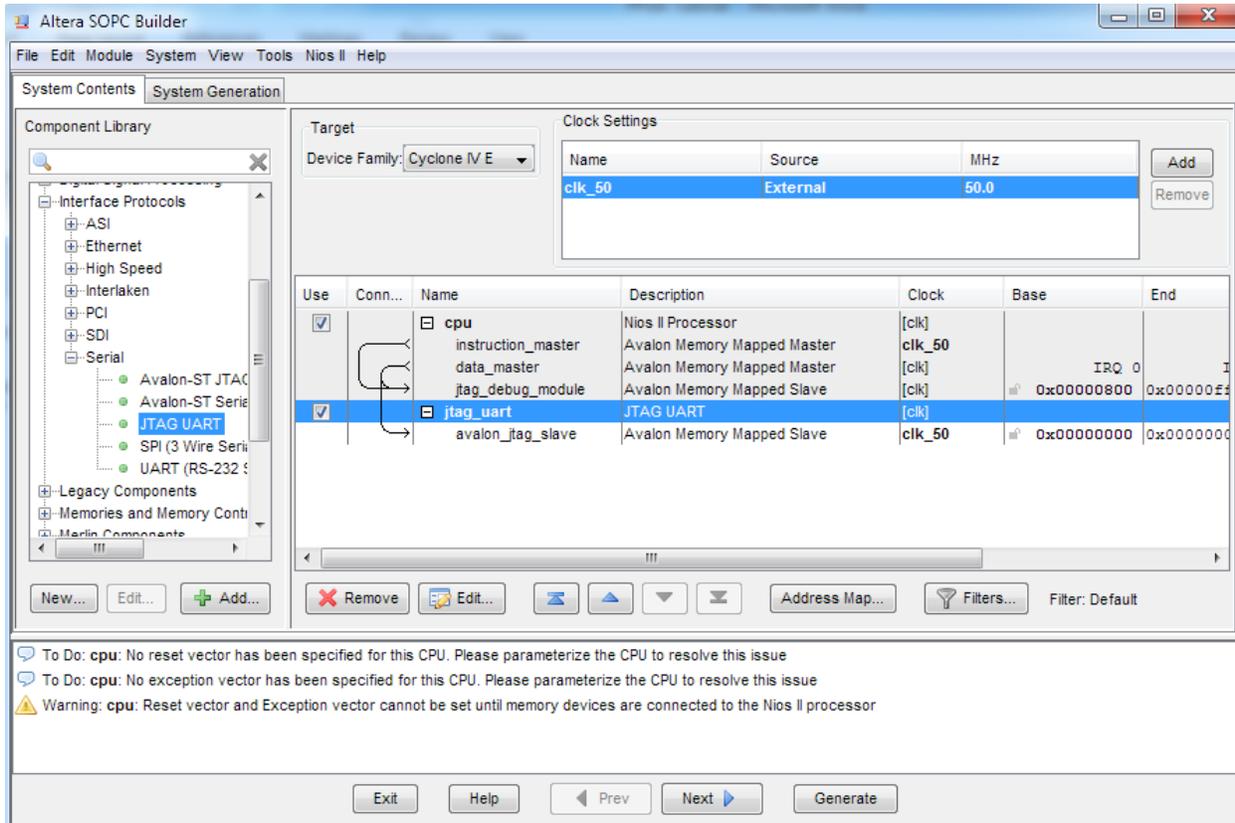


Figure 1-22 Rename JTAG UART

15. Choose **Library > Memories and Memory Controllers > On-Chip > On-Chip Memory (RAM or ROM)** to open wizard of adding On-Chip memory. See Figure 1-23 and Figure 1-24.

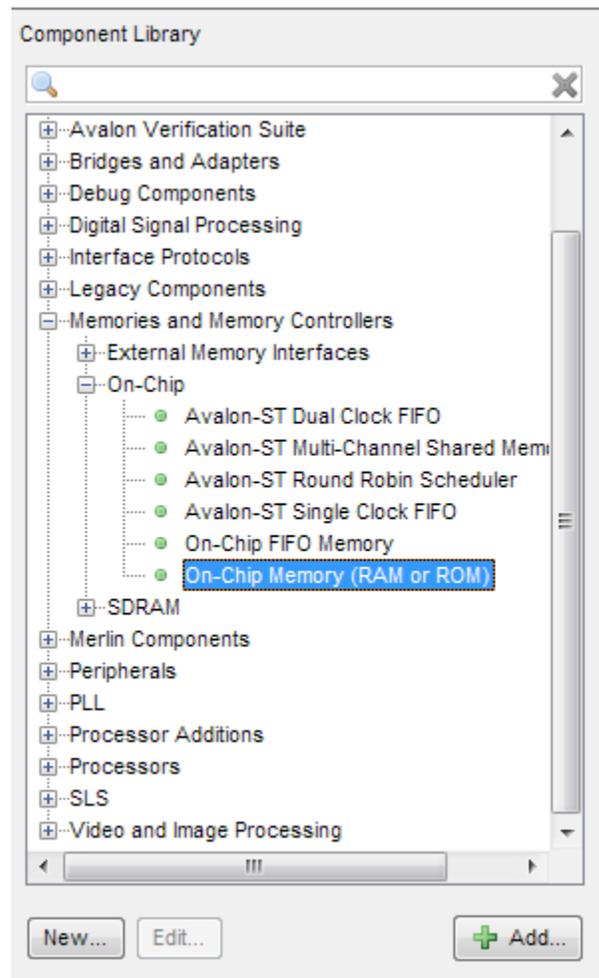


Figure 1-23 Add On-Chip Memory

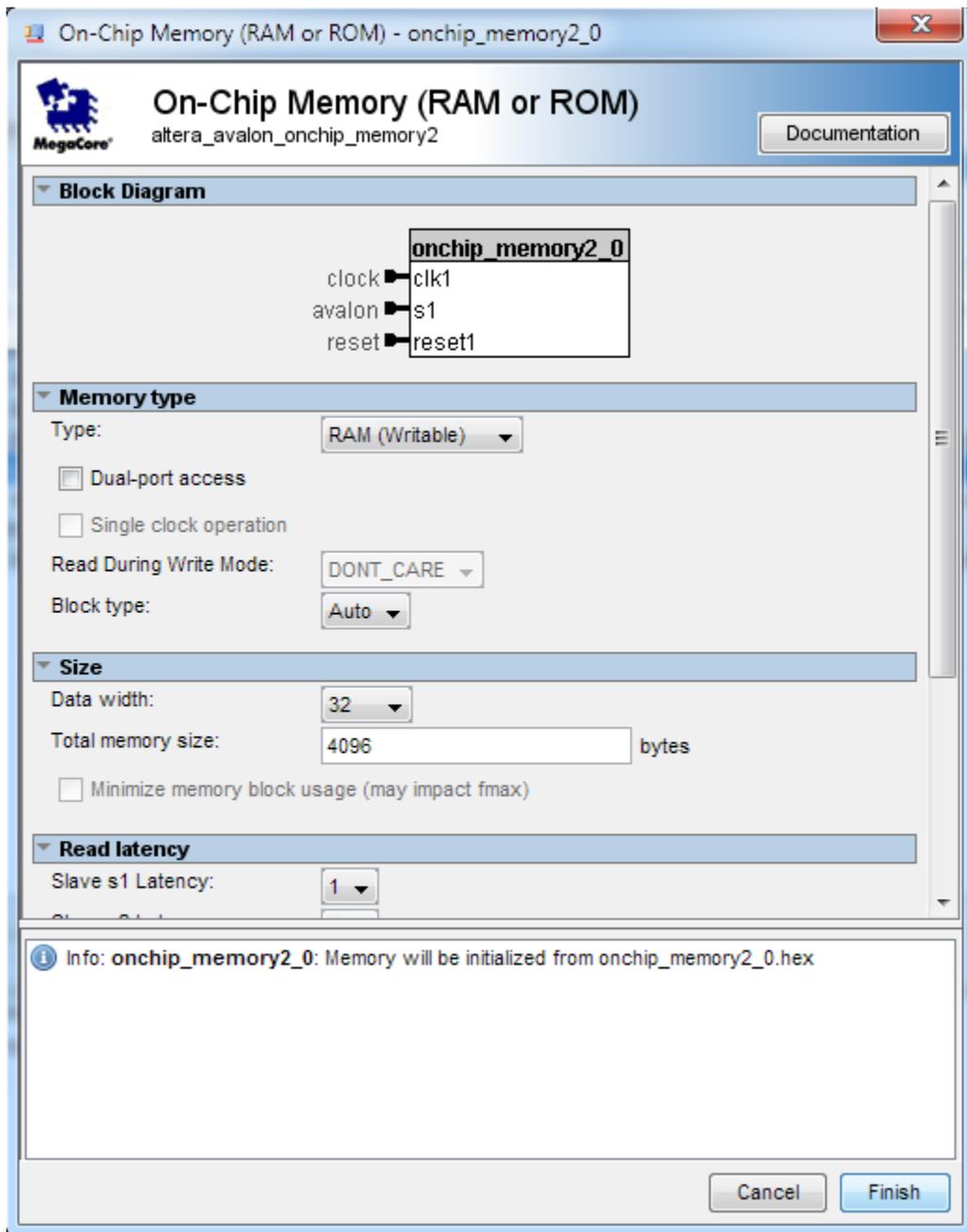


Figure 1-24 On-Chip Memory Box

16. Modify **Total memory size** to **204800** as shown in Figure 1-25. Click **Finish** to return to the window as in Figure 1-26.

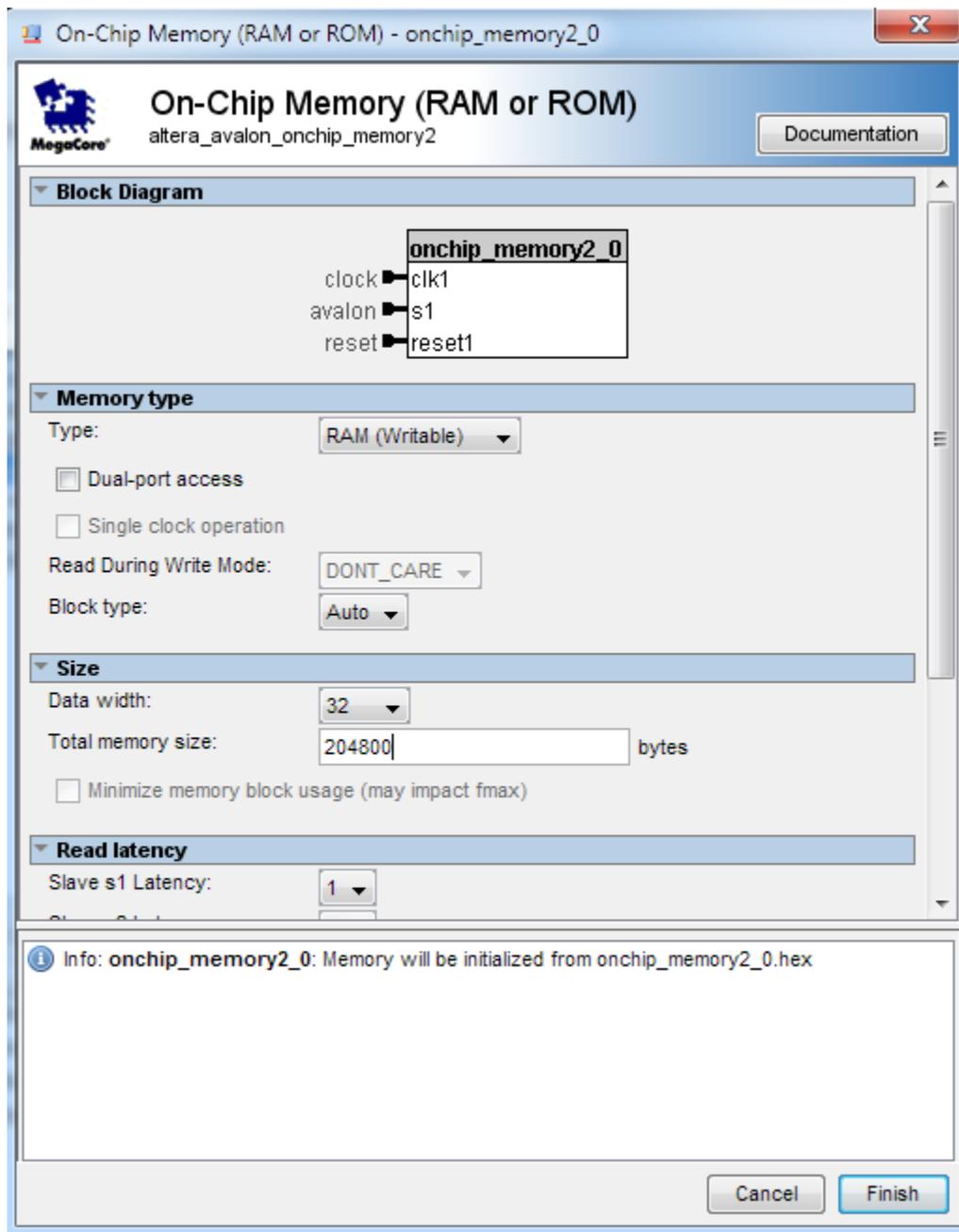


Figure 1-25 Update Total memory size

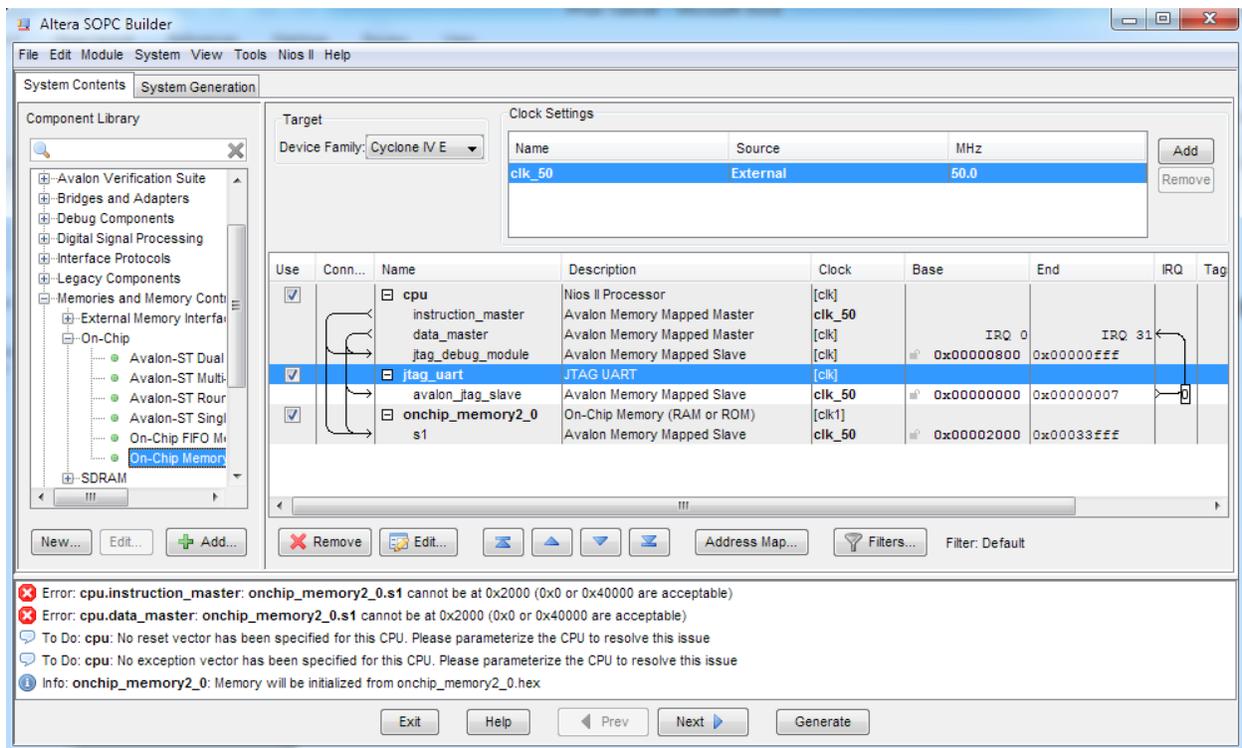


Figure 1-26 Add On-Chip memory Completely

17. Rename **onchip_memory2_0** to **onchip_memory2** as shown in Figure 1-27.

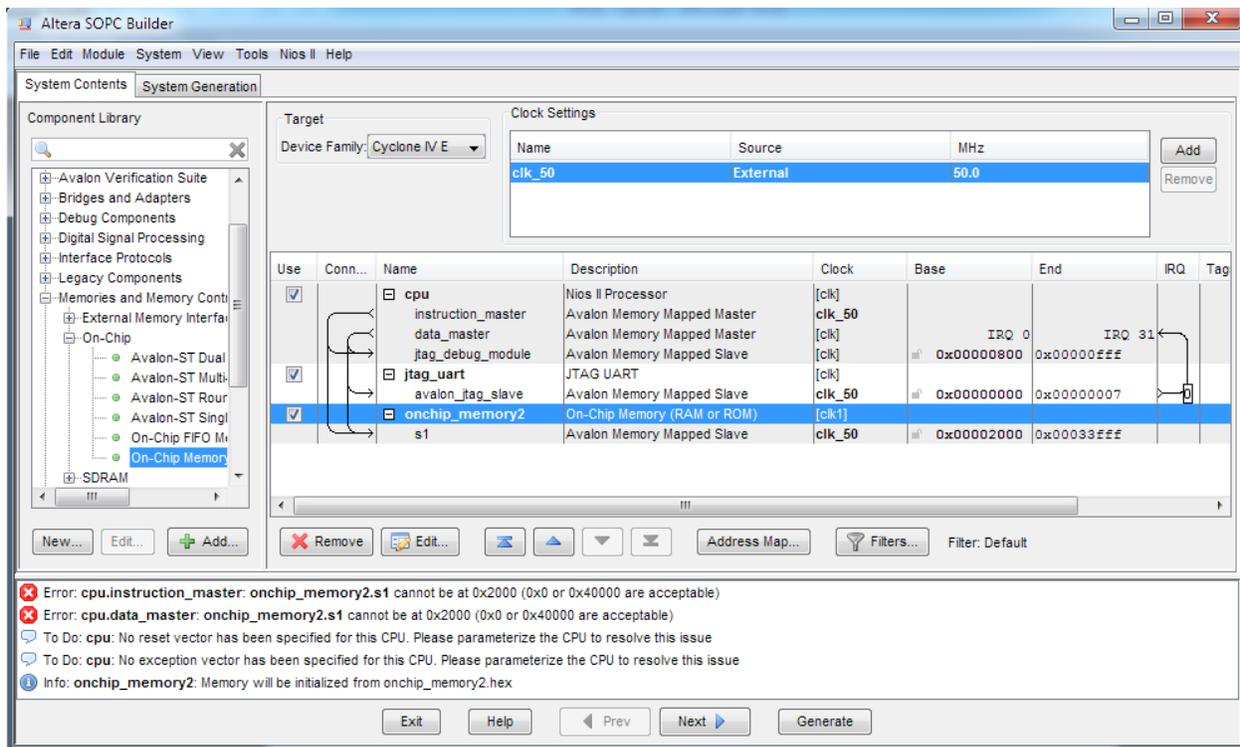


Figure 1-27 Rename On-Chip memory

18. Click **cpu** in the component list on the right part to edit the component. Update **Reset vector** and **Exception Vector** as shown in Figure 1-28. Then click **Finish** to return to the window as shown Figure 1-29.

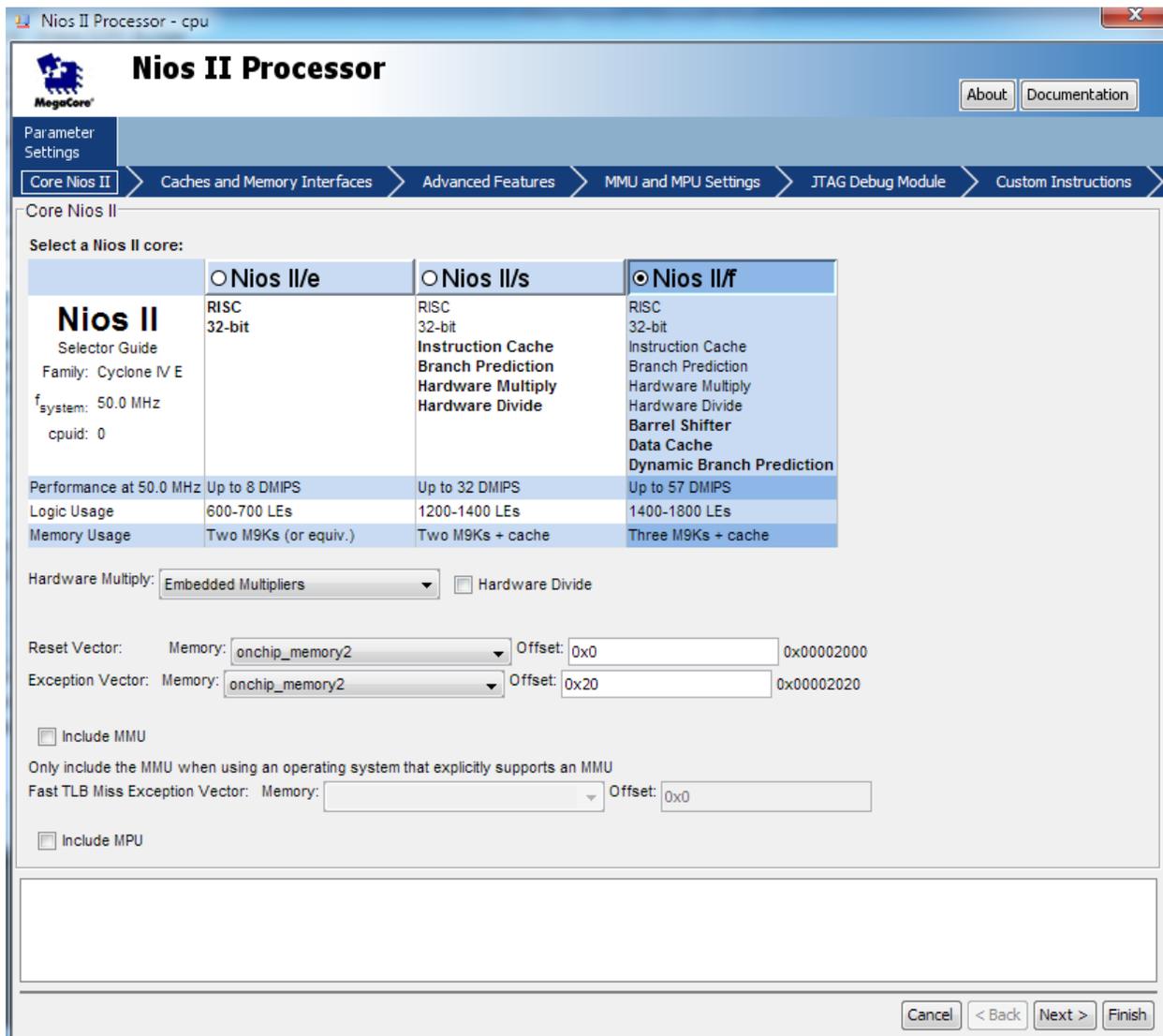


Figure 1-28 Update CPU settings

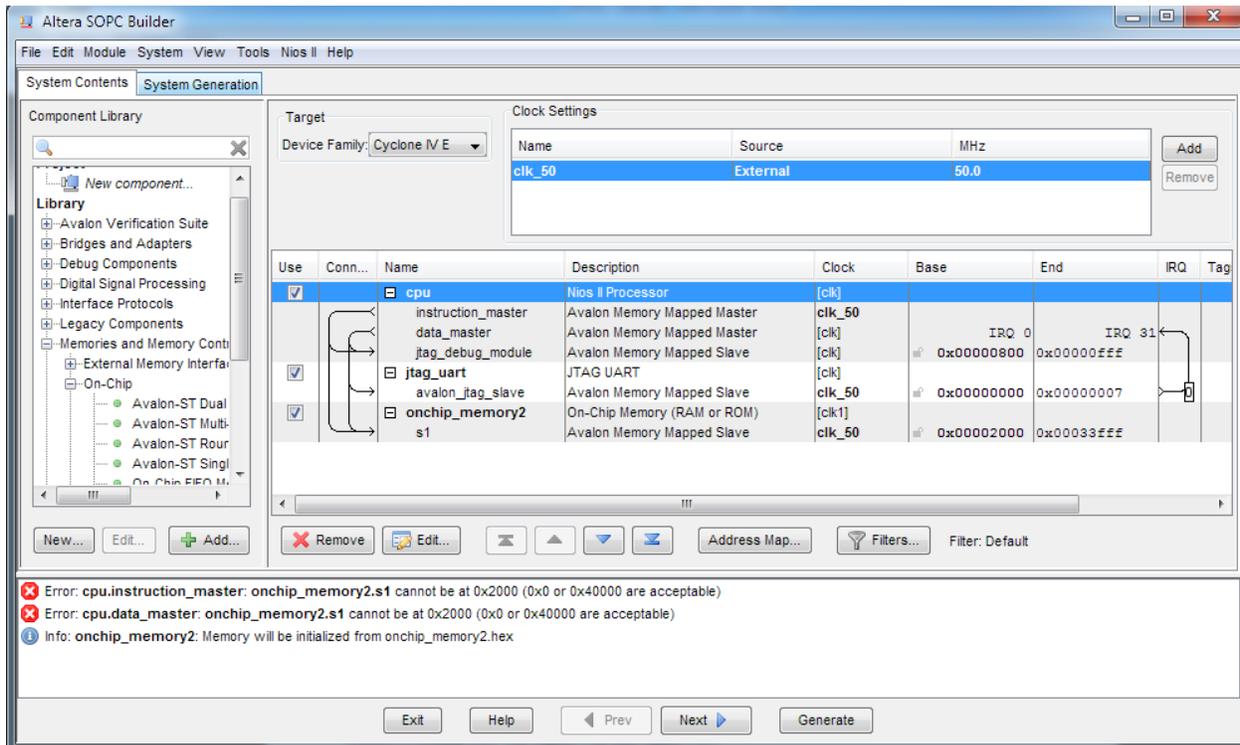


Figure 1-29 Update CPU settings Completely

19. Choose **Library > Peripherals > Microcontroller Peripherals >PIO (Parallel I/O)** to open wizard of adding PIO. See Figure 1-30 and Figure 1-31.

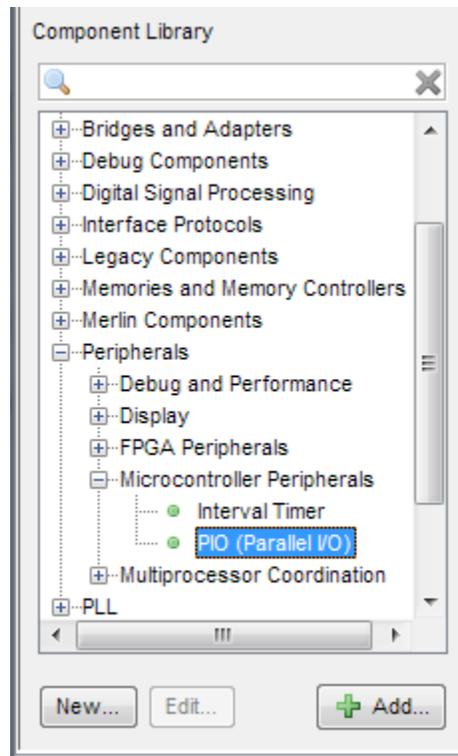


Figure 1-30 Add PIO

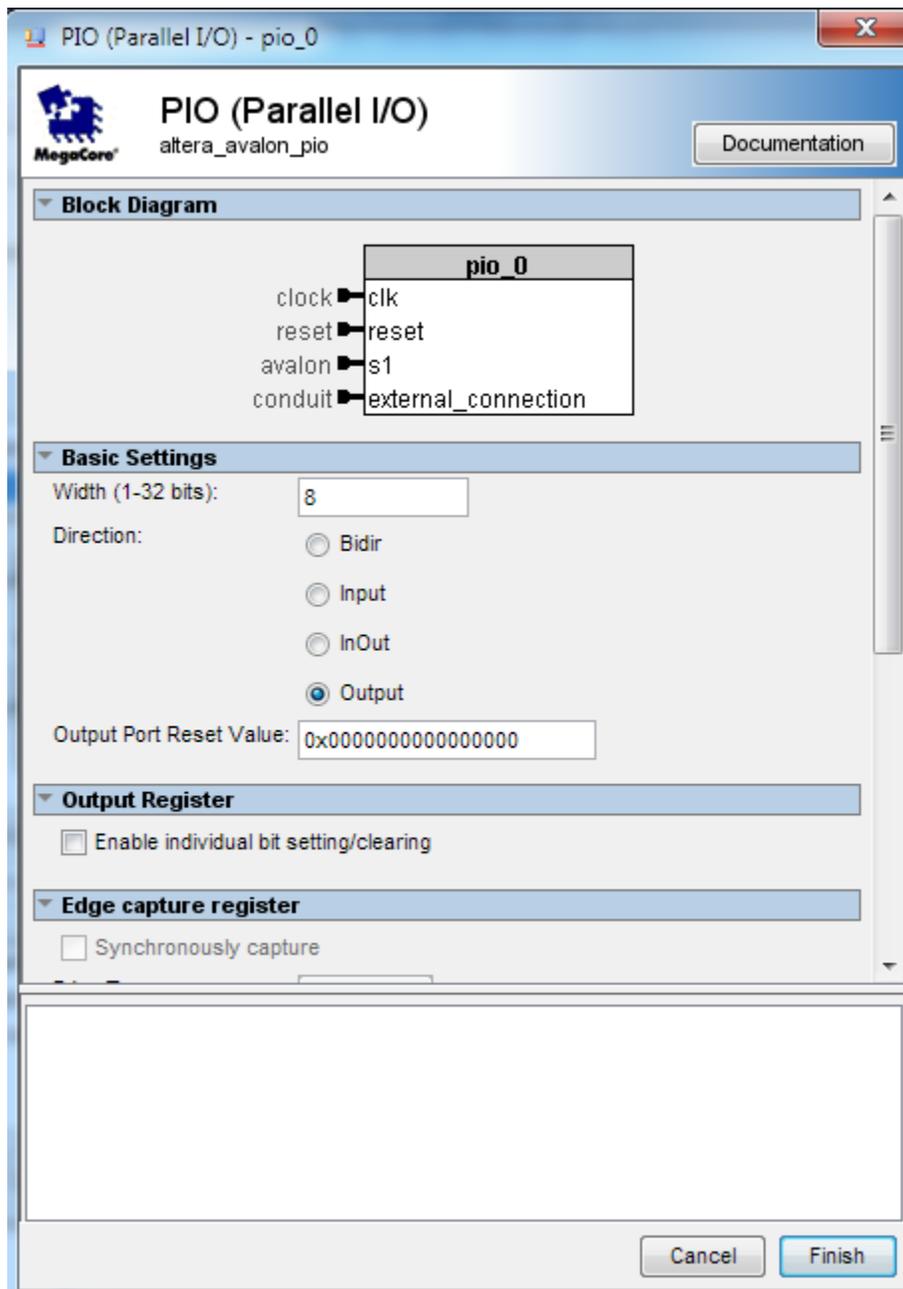


Figure 1-31 Add PIO

20. Click **Finish** to close PIO box and return to the window as shown in Figure 1-32.

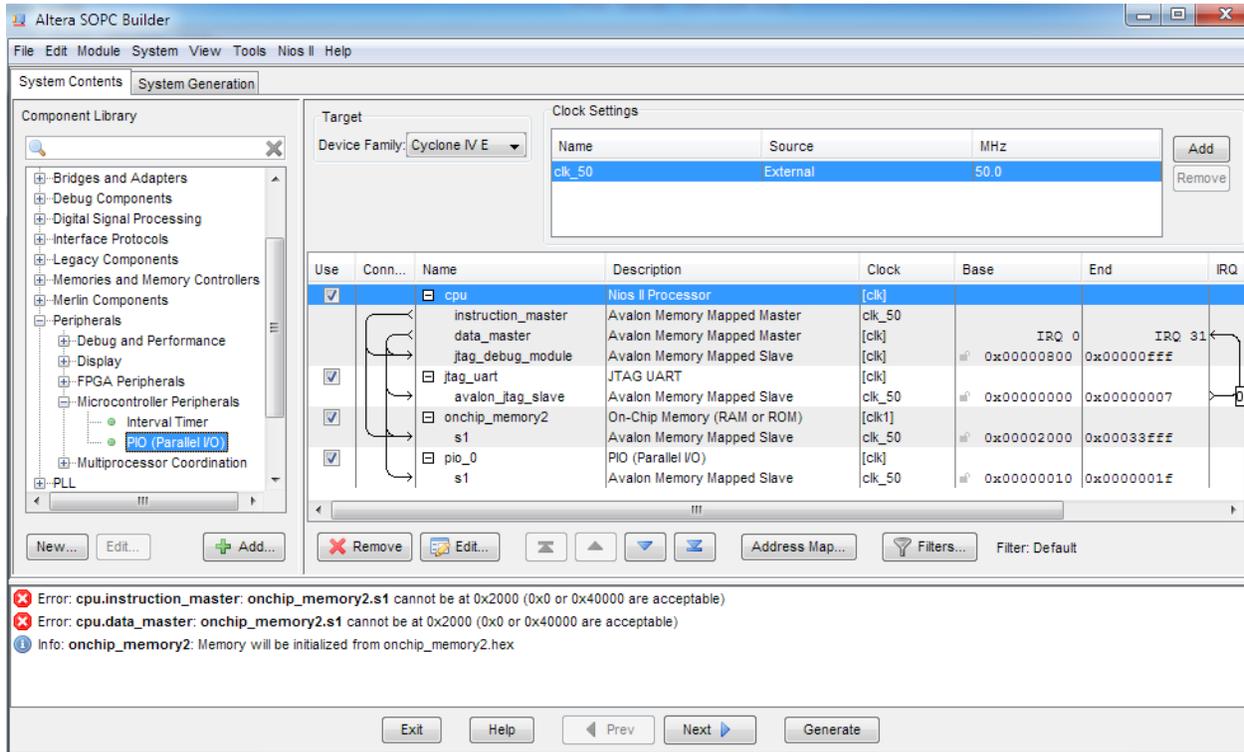


Figure 1-32 PIO

21. Rename **pio_0** to **pio_led** as shown in Figure 1-33.

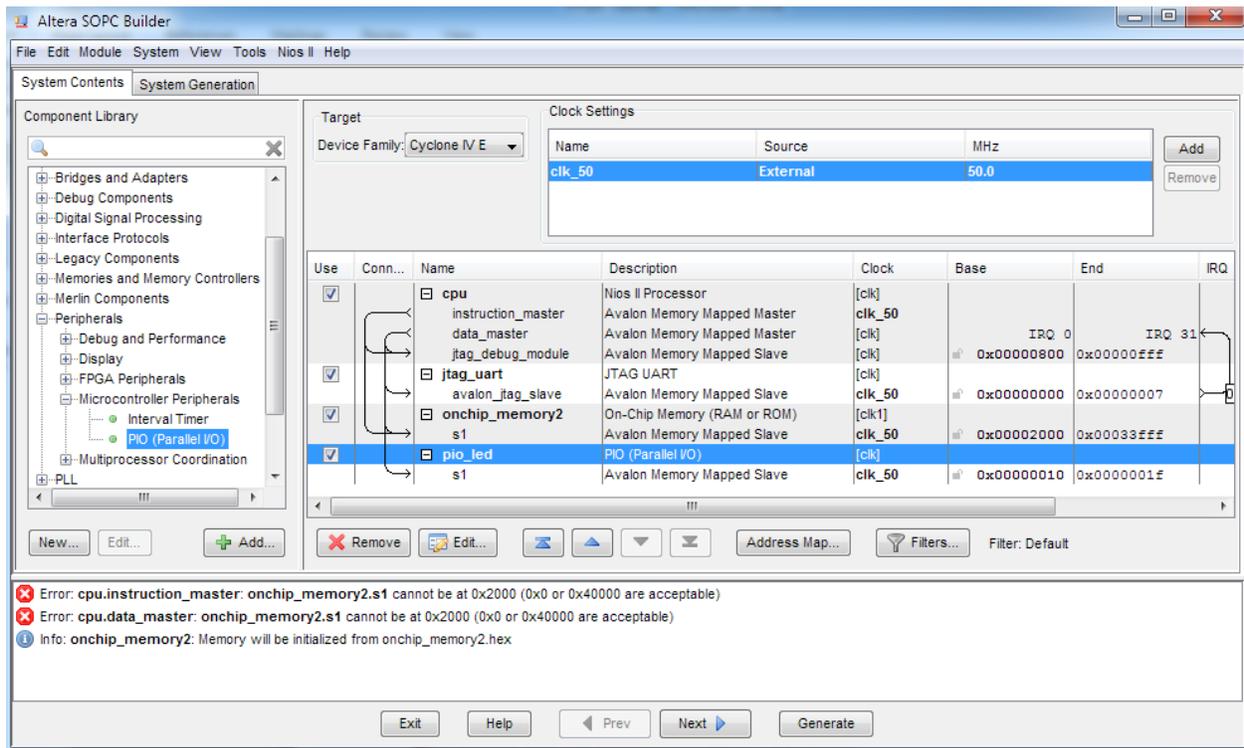
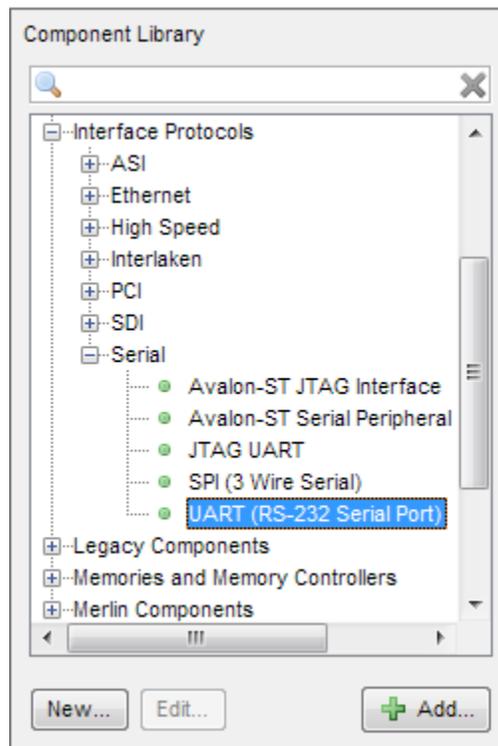
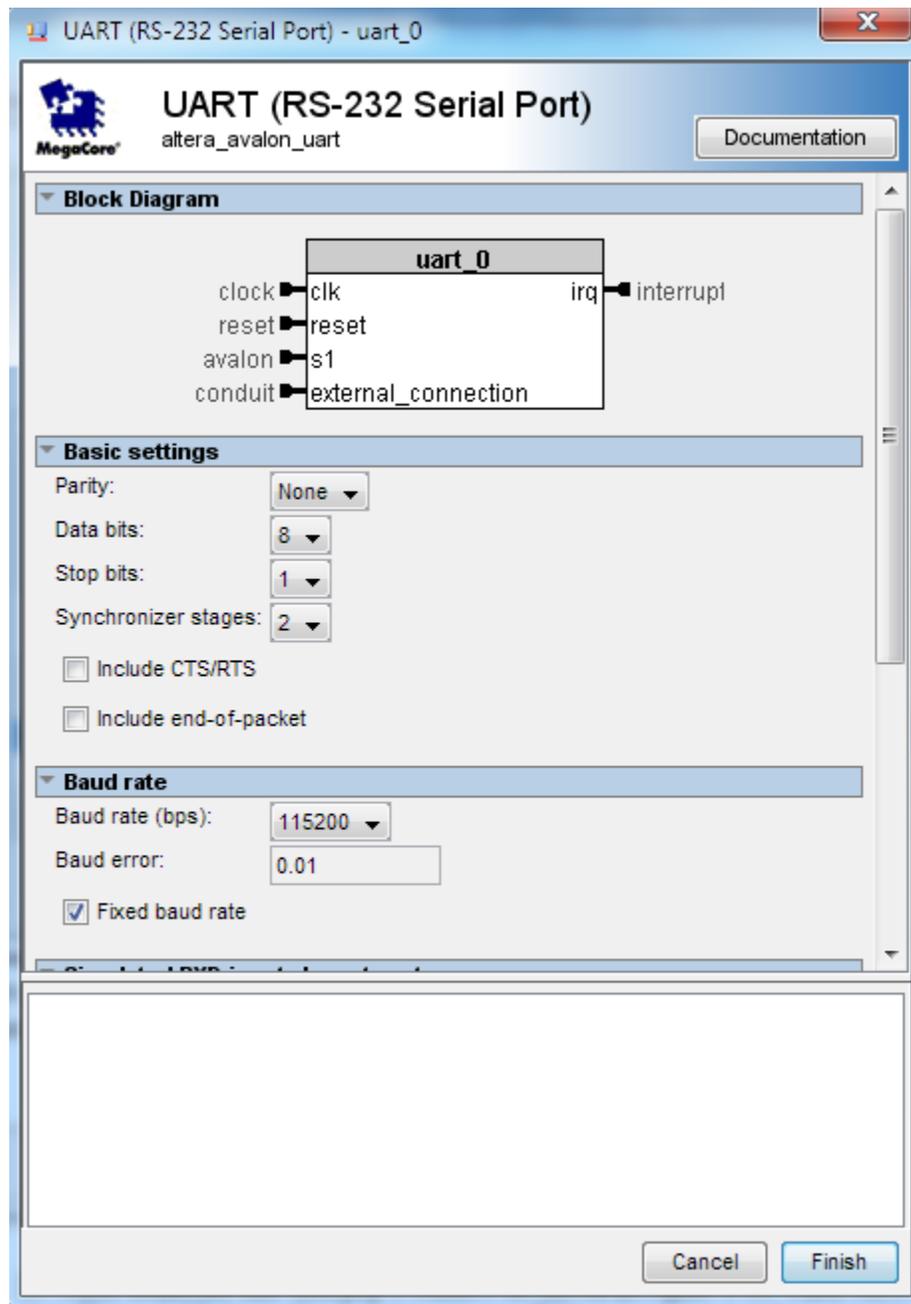


Figure 1-33 Rename PIO

22. Choose **Library > Interface Protocols > Serial > UART** to open wizard of adding PIO. See Figure 1-34 and Figure 1-35.

**Figure 1-34 Add UART**

**Figure 1-35 Add UART**

23. Click **Finish** to close UART box and return to the window as shown in Figure 1-36.

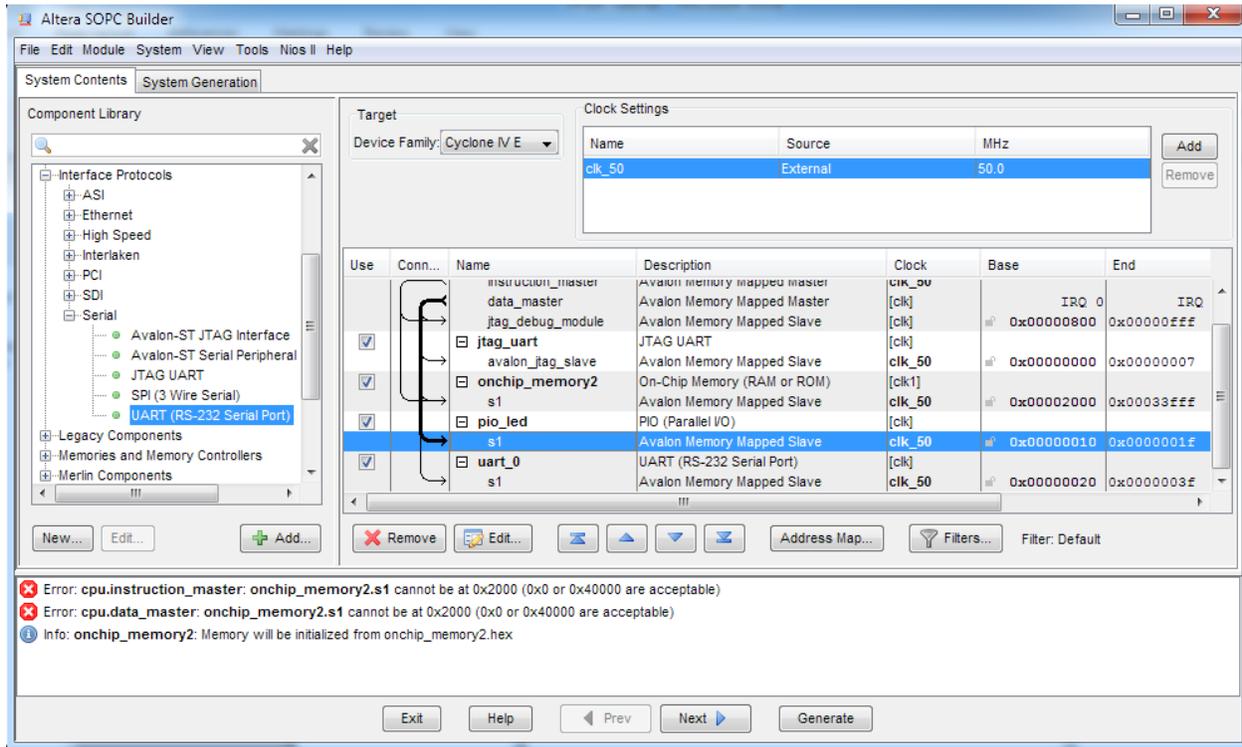


Figure 1-36 UART

24. Rename `uart_0` to `uart_wifi` as shown in Figure 1-37.

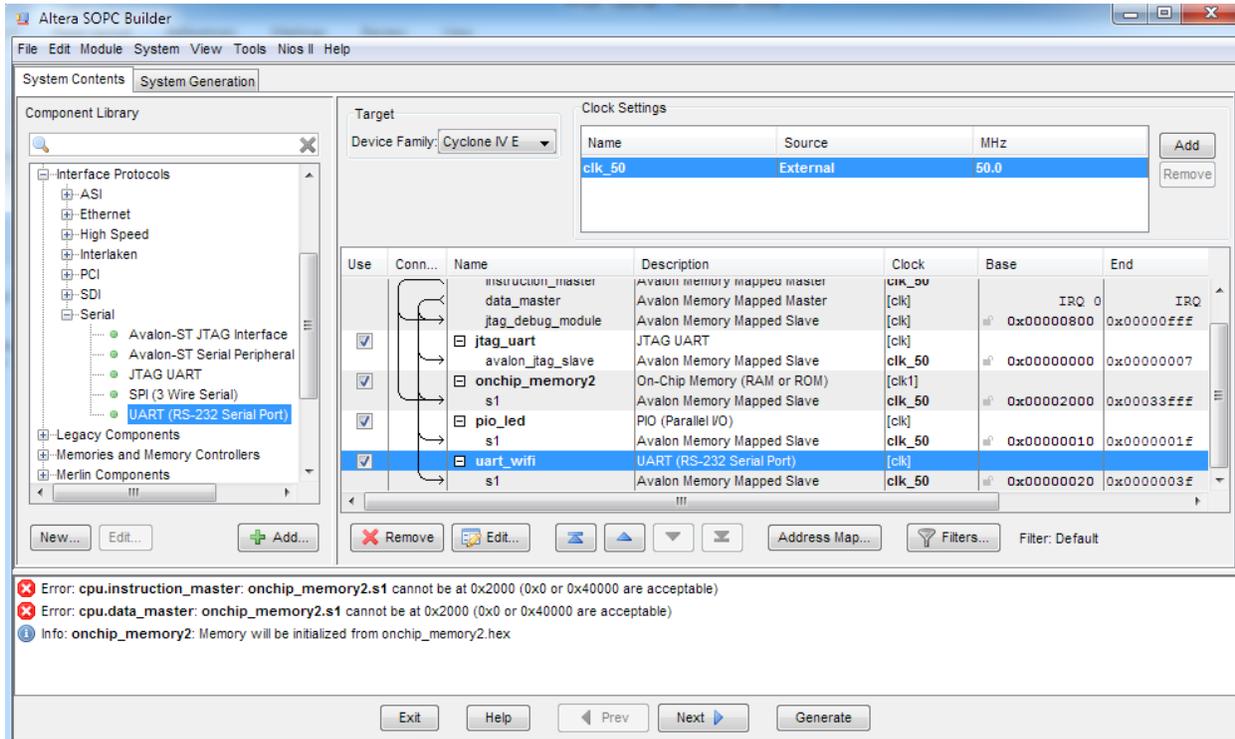


Figure 1-37 Rename UART

25. Choose the **Add** button to add a new component in our case **pwm_gen**.

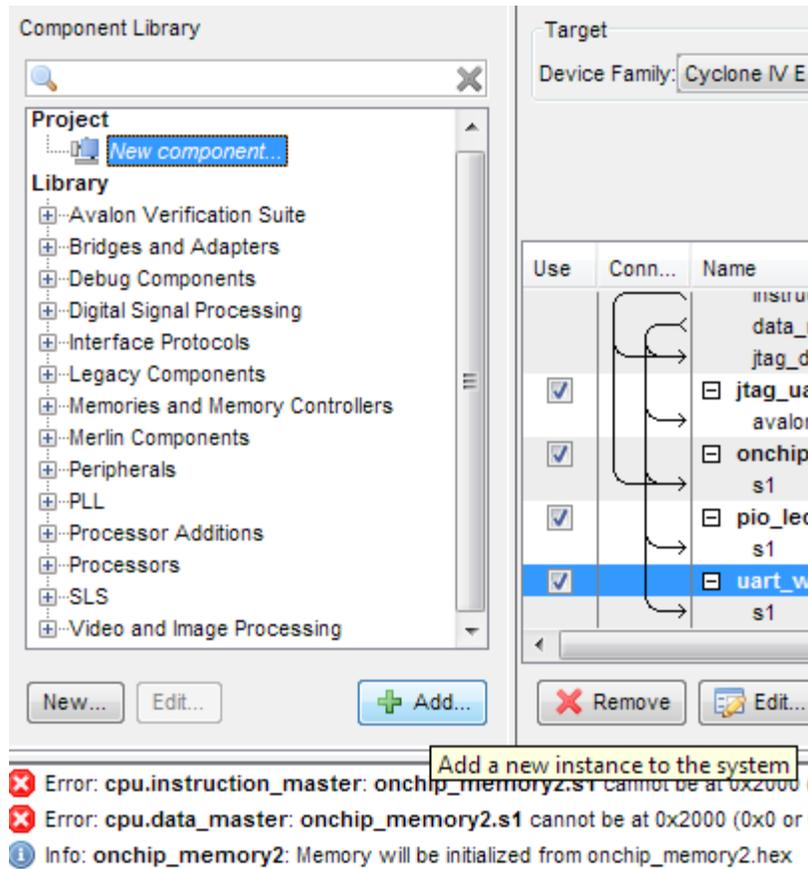


Figure 1-38 Add pwm_gen

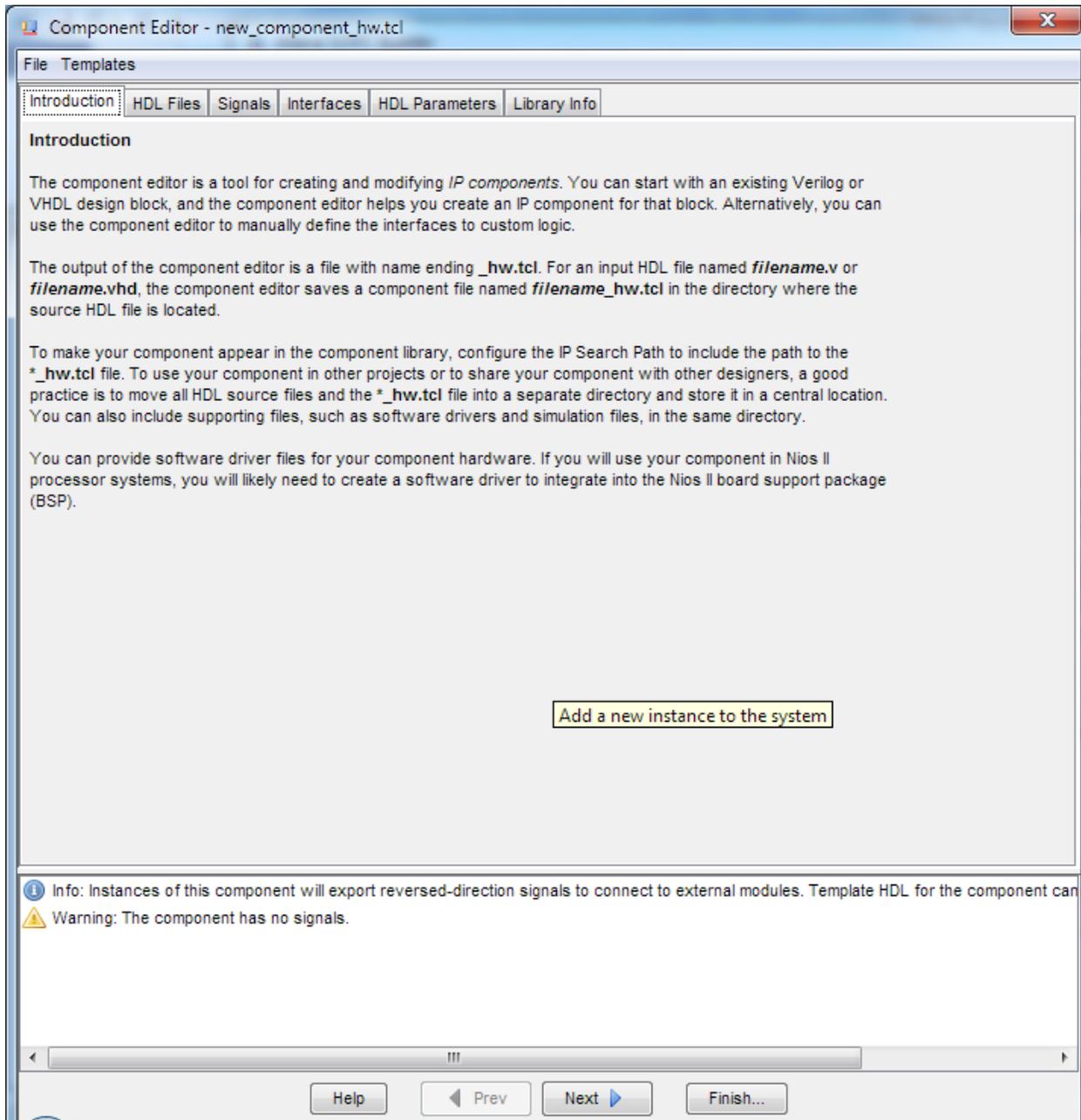


Figure 1-39 Component Editor: Introduction [page 1 of 6]

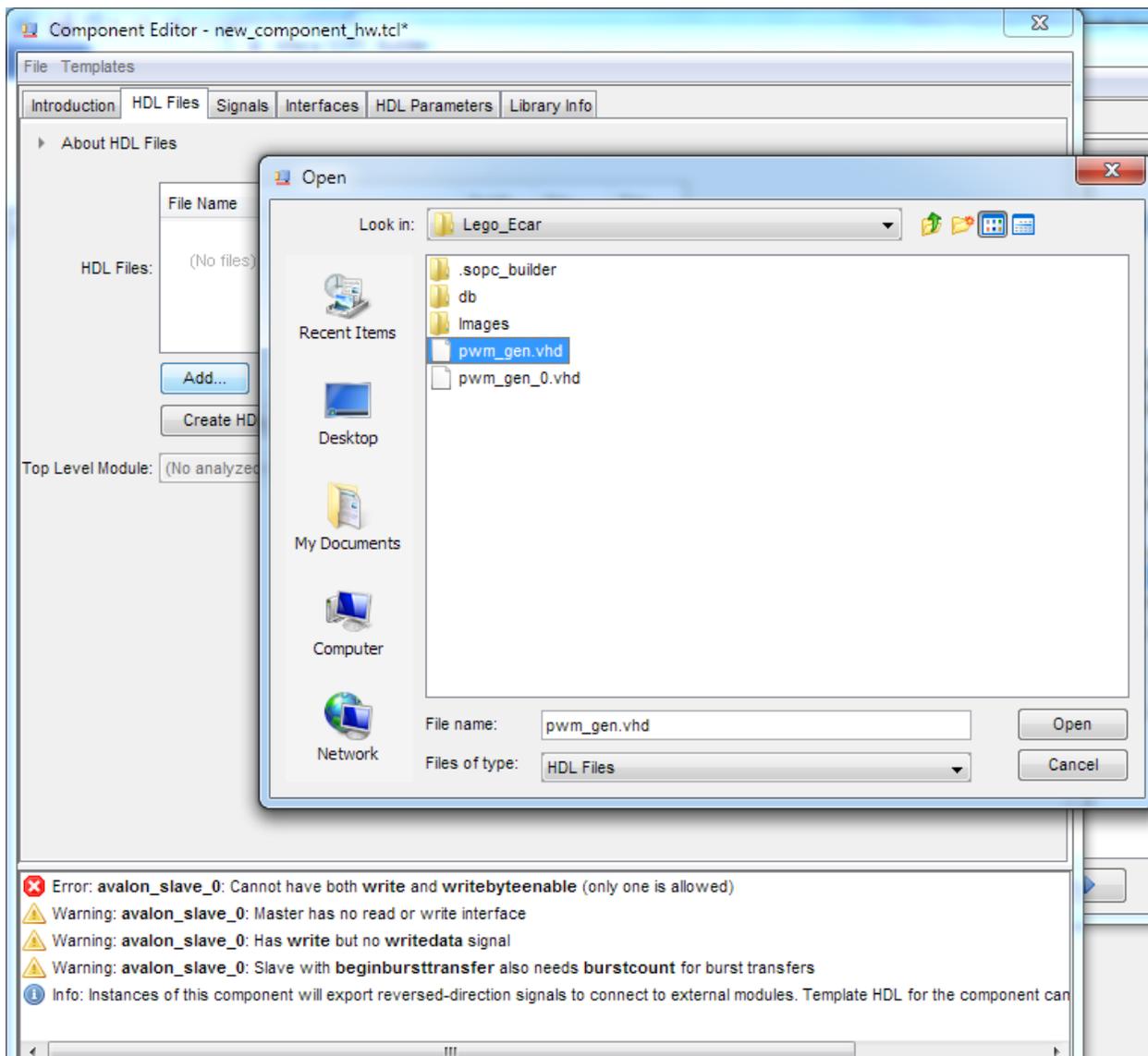


Figure 1-40 Component Editor: HDL Files [page 2 of 6]

26. Choose the **Add** button to add a hdl file in our case **pwm_gen.vhd**
pwm_gen.vhd : vhdl module which implements the function of pwm generation. Its interface to Nios II is Avalon compatible.

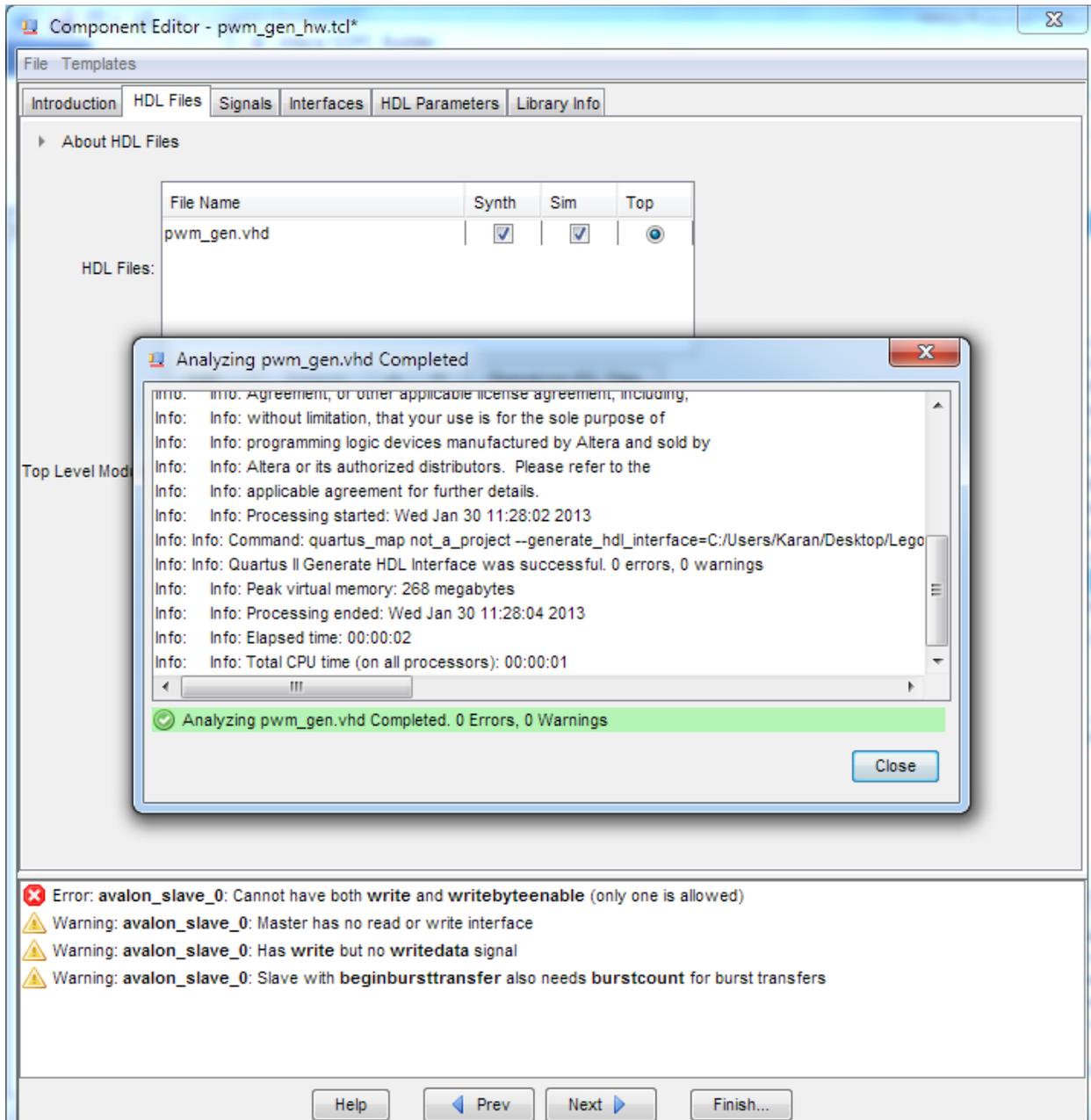


Figure 1-41 Component Editor: HDL Files [page 2 of 6]

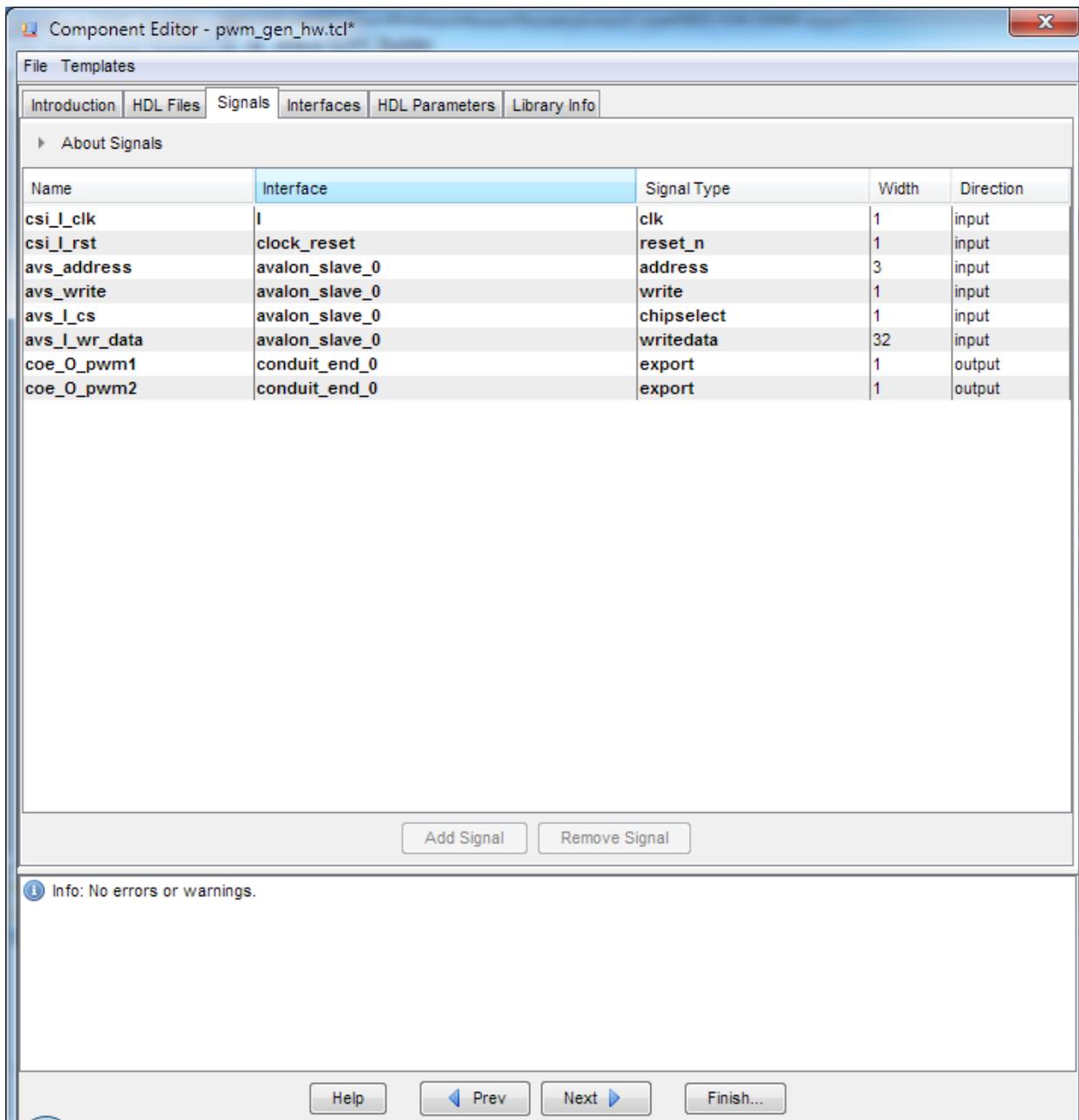


Figure 1-42 Component Editor: Signals [page 3 of 6]

27. Please check the signal type as it might not be correct initially. Change it to make it exactly as shown in the above figure.

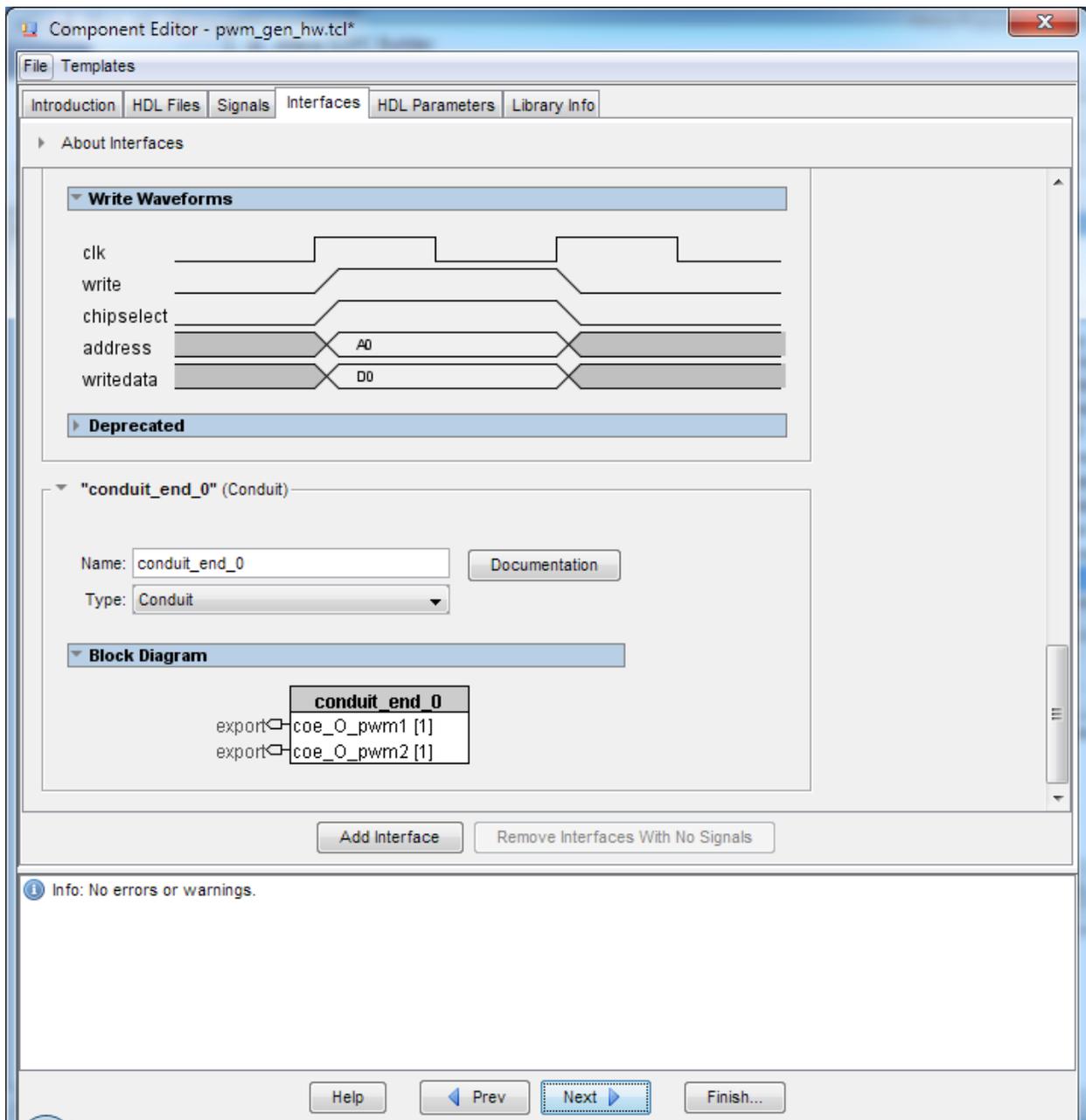
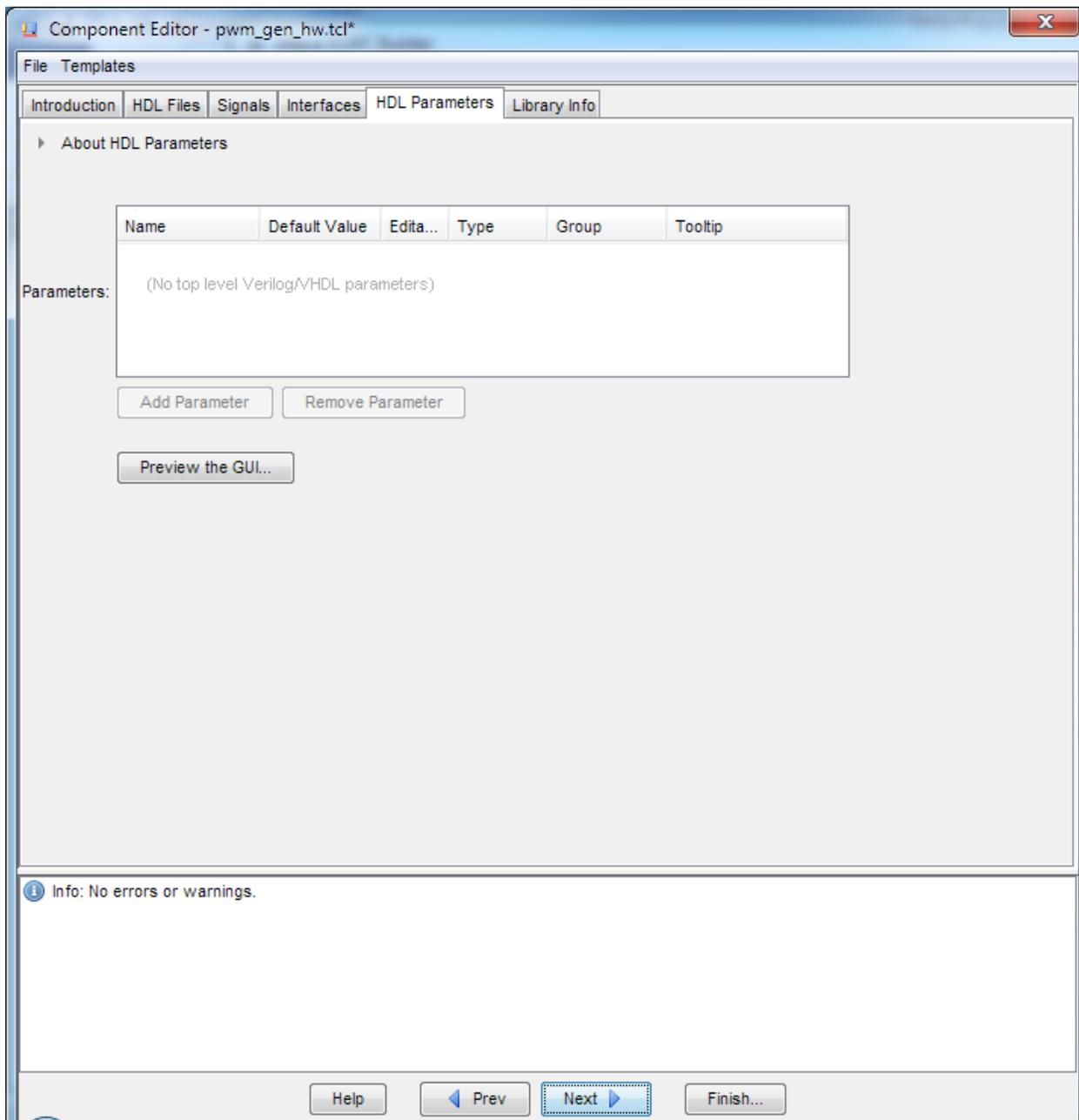


Figure 1-43 Component Editor: Interfaces [page 4 of 6]

**Figure 1-44 Component Editor: HDL Parameters [page 5 of 6]**

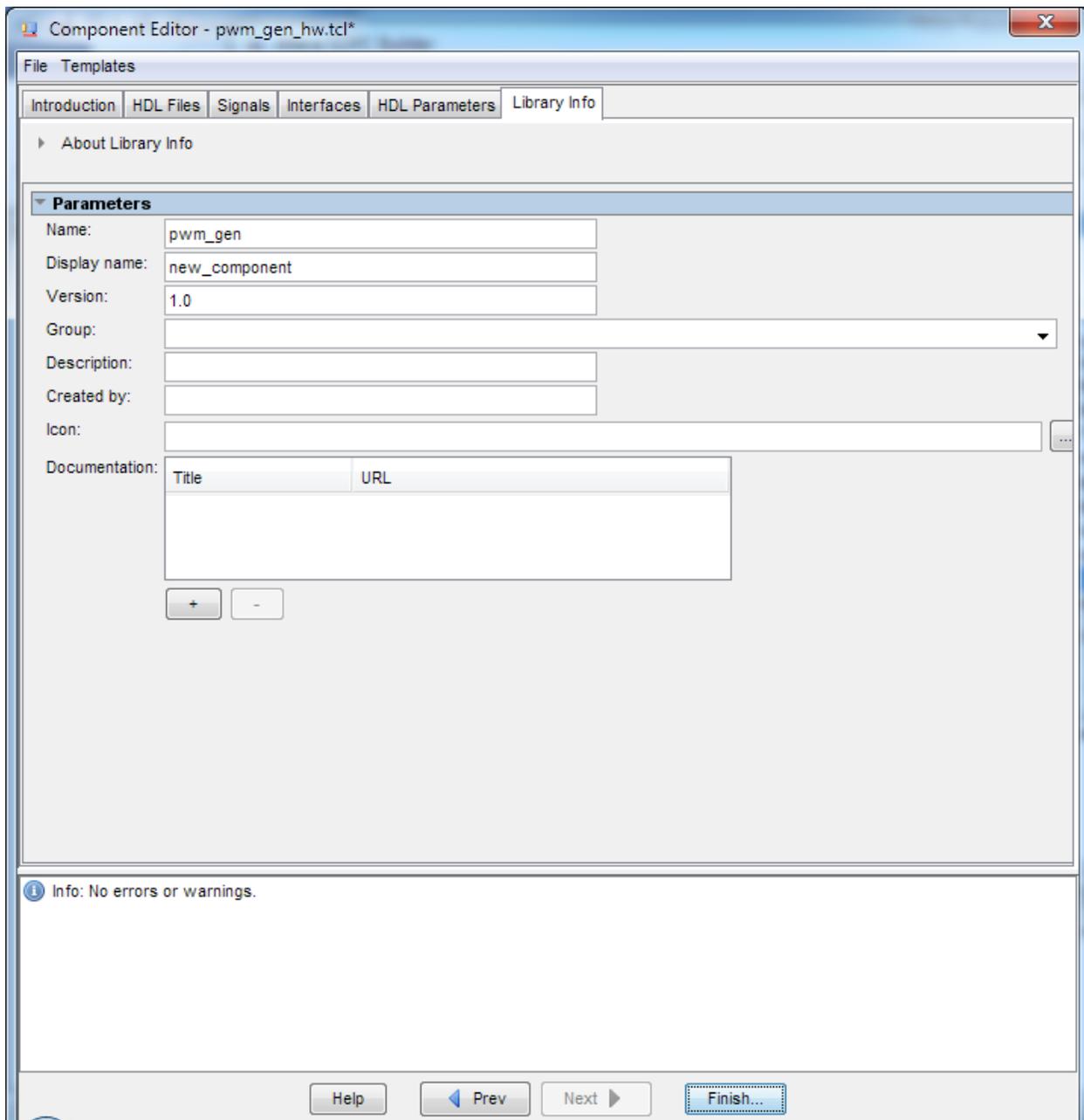


Figure 1-45 Component Editor: Library Info [page 6 of 6]

28. Click **Finish** and you will find the **pwm_gen** component in the Library as shown below.

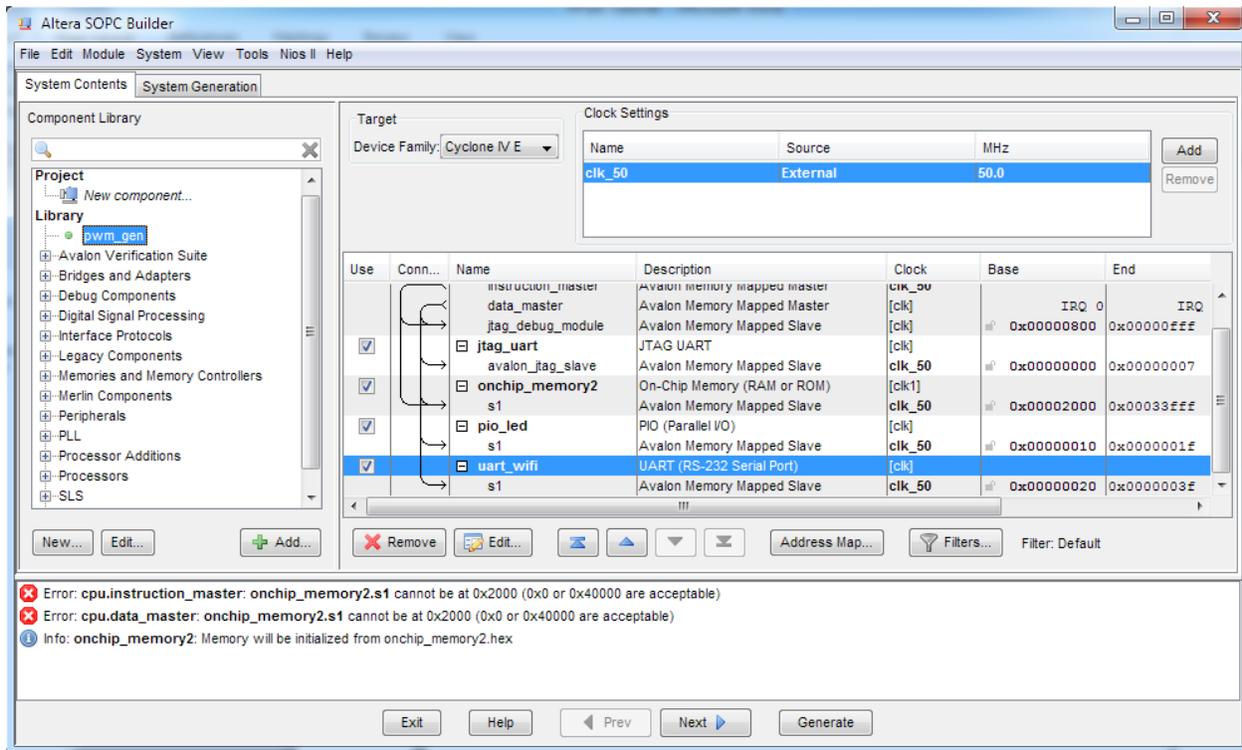


Figure 1-46 Add pwm_gen

29. Click **Finish** to close pwm_gen box and return to the window as shown in Figure 1-47.

30. Choose **System > Auto-Assign Base Addresses** as shown in Figure 1-48. After that, you will find that there is no error in the message window as shown in Figure 1-49.

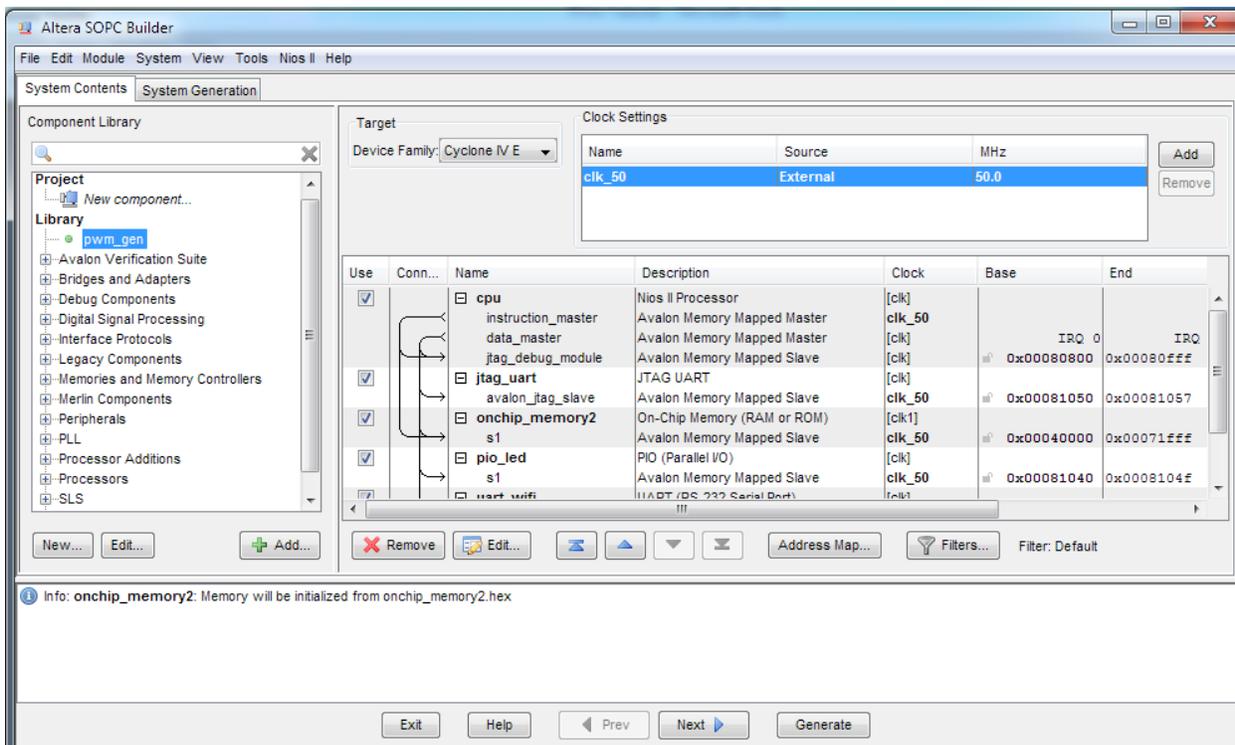
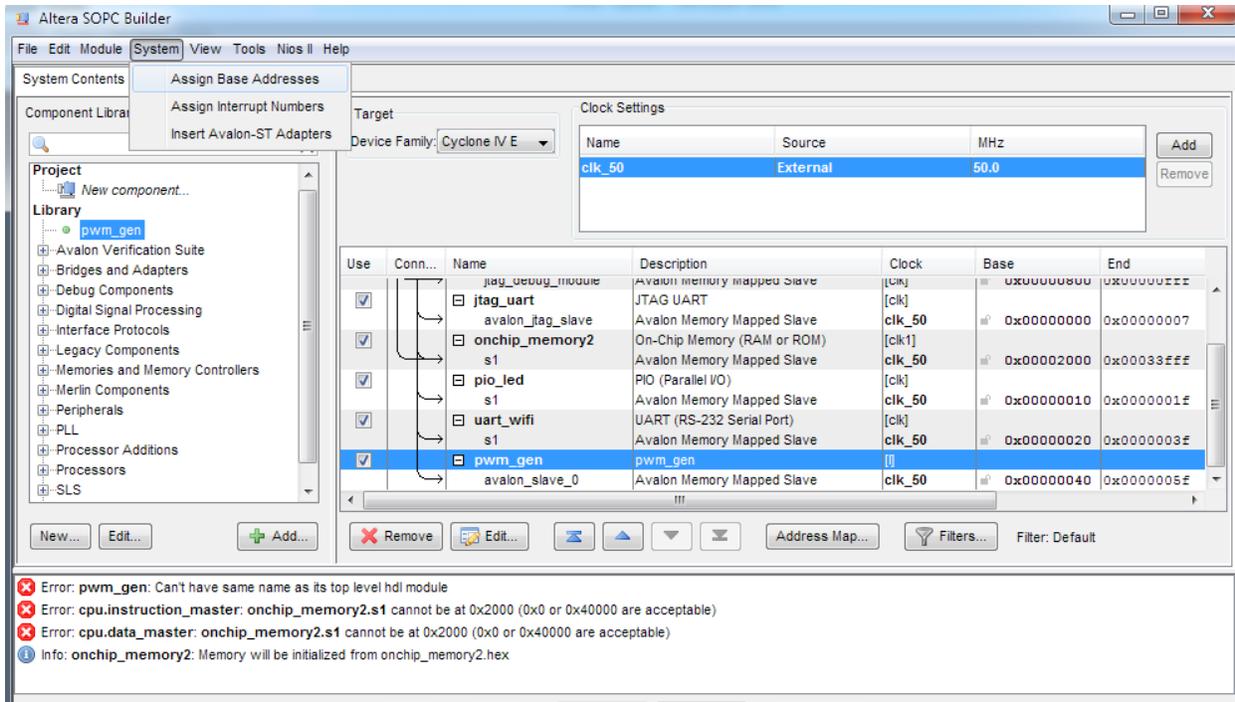


Figure 1-49 No errors or warnings

31. Click **Generate** and then pop a window as shown in Figure 1-50. Click **Save** and the generation start. If there is no error in the generation, the window will show successful as shown in Figure 1-51.

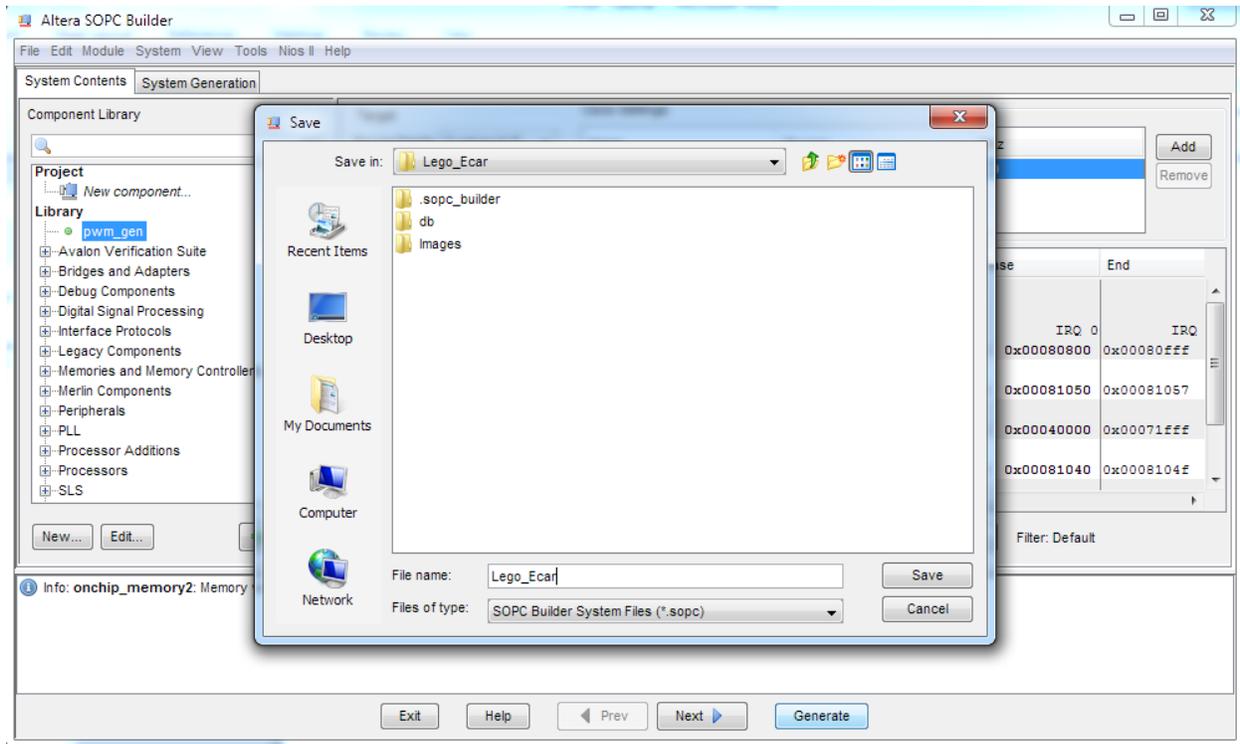


Figure 1-50 Generate SOPC

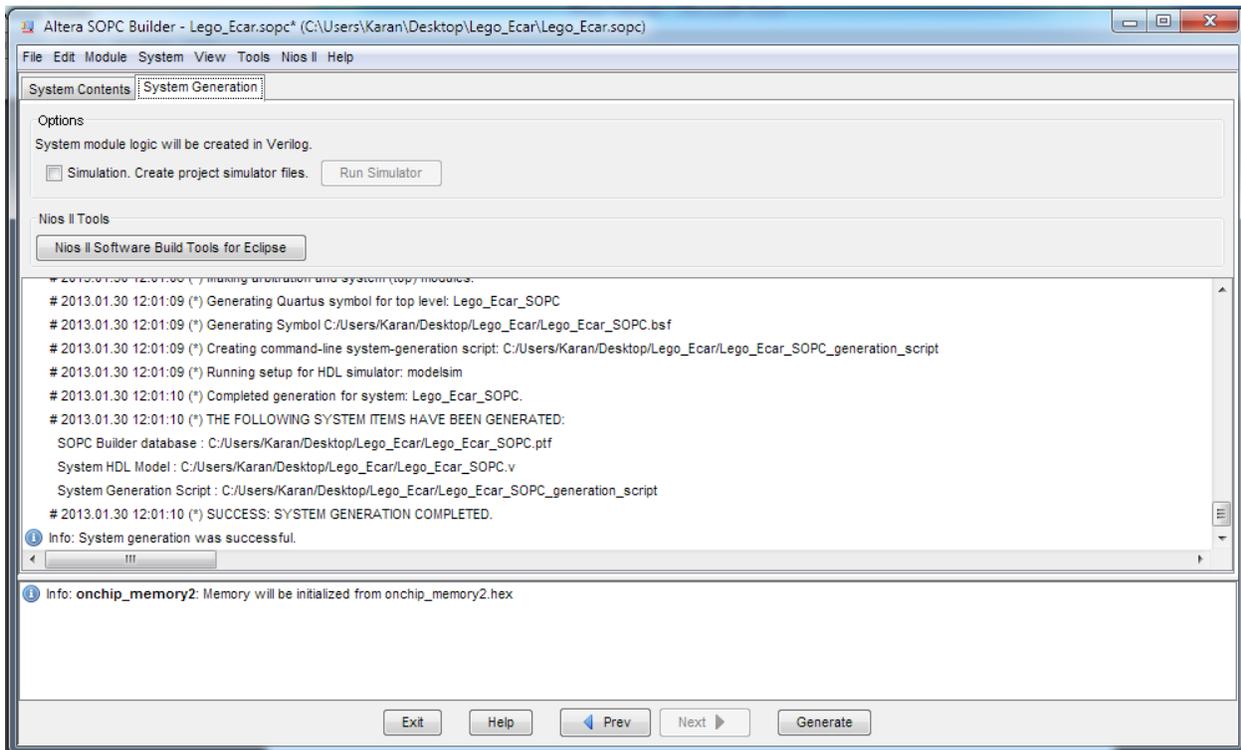


Figure 1-51 Generate SOPC Complete

32. Click **Exit** to exit the SOPC Builder and return to the window as shown in Figure 1-52.

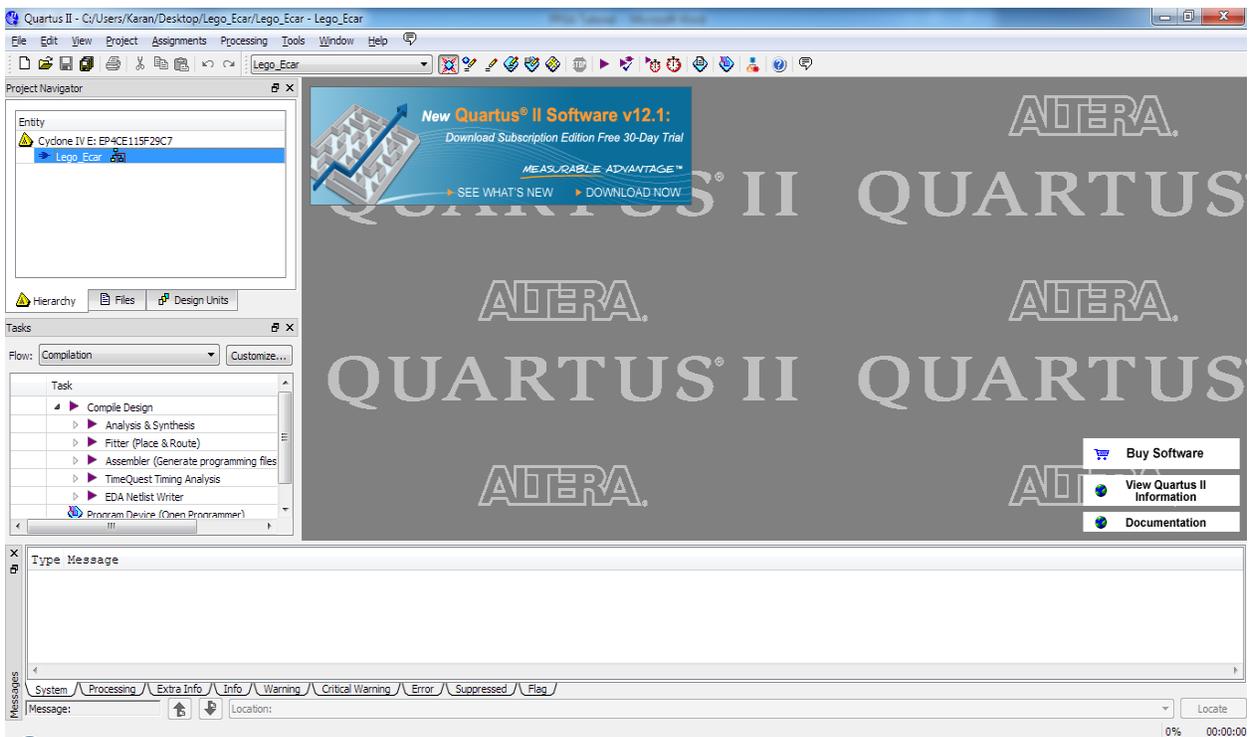


Figure 1-52

33. Choose **File > New** to open new files wizard. See Figure 1-53 and Figure 1-54.

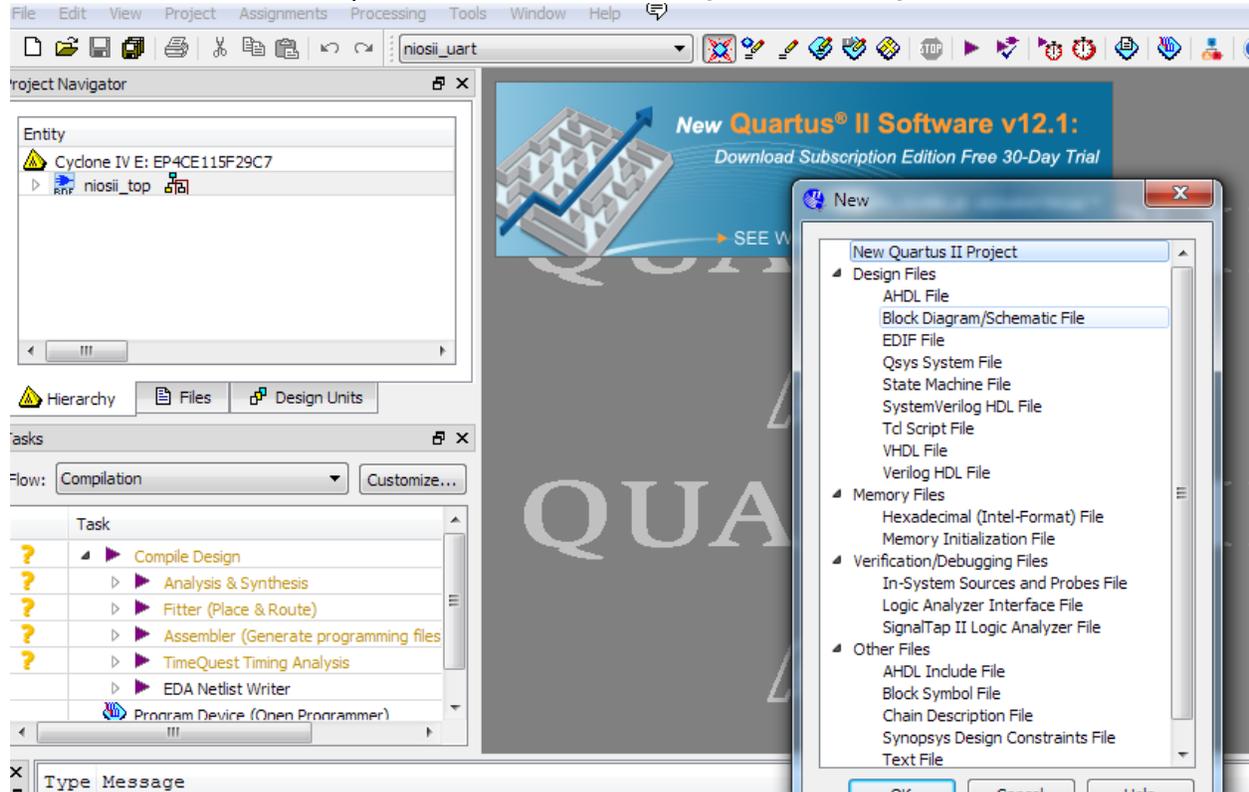


Figure 1-53 New Block Diagram File

34. Choose **New Block Diagram File** and click **OK** to return to the window as shown in Figure 1-54. Figure 1-54 show a blank Block Diagram File

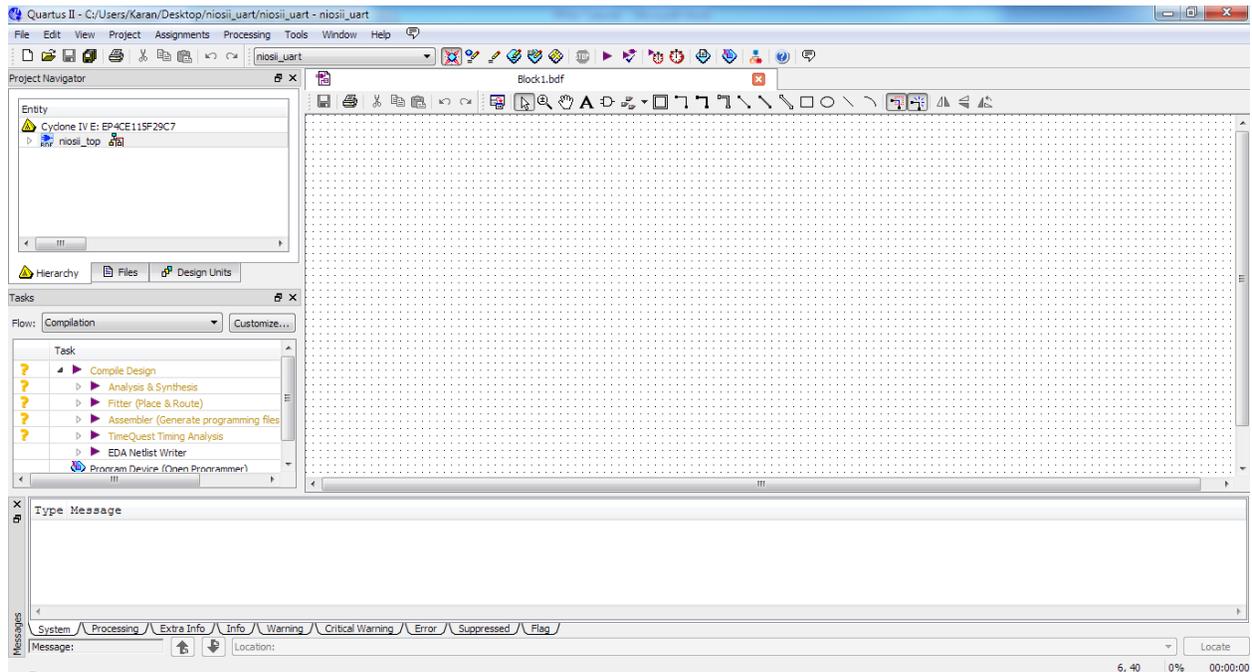


Figure 1-54 A blank Block Diagram File

35. Double click on the empty block diagram file to add the Symbols. Add the **Lego_Ecar_SOPC** from the Project folder.

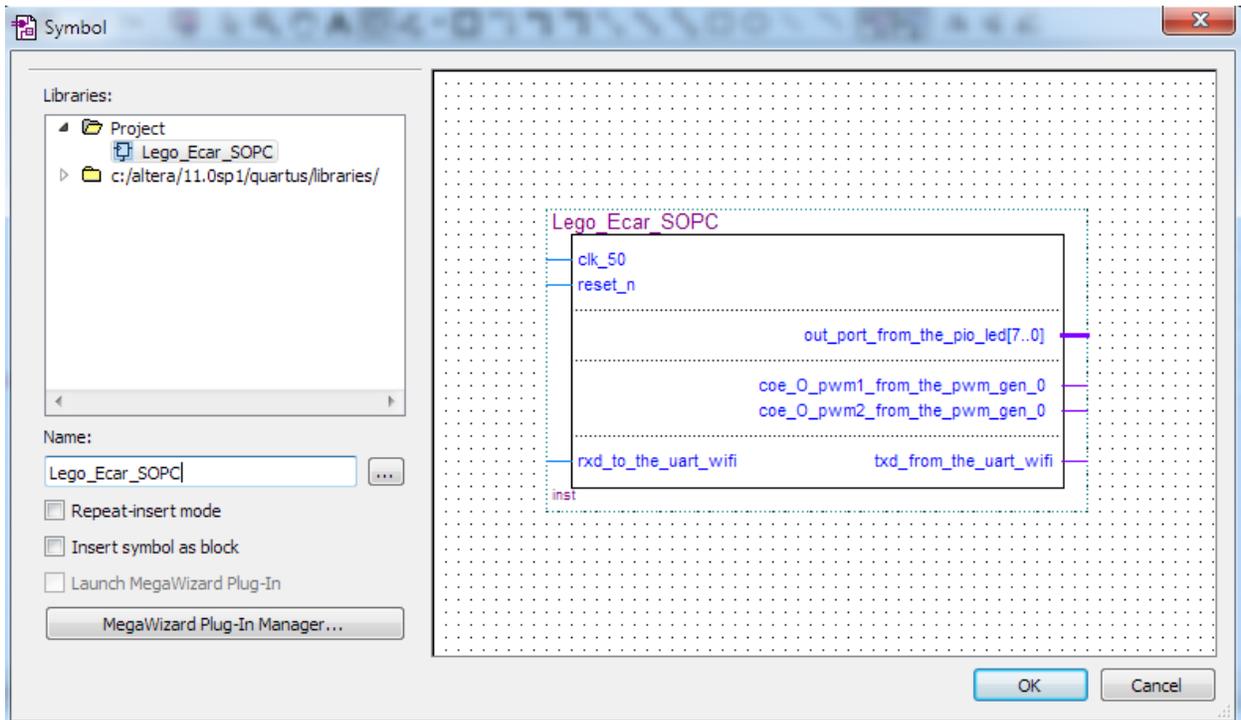


Figure 1-55

36. Add other symbols to make the following block diagram

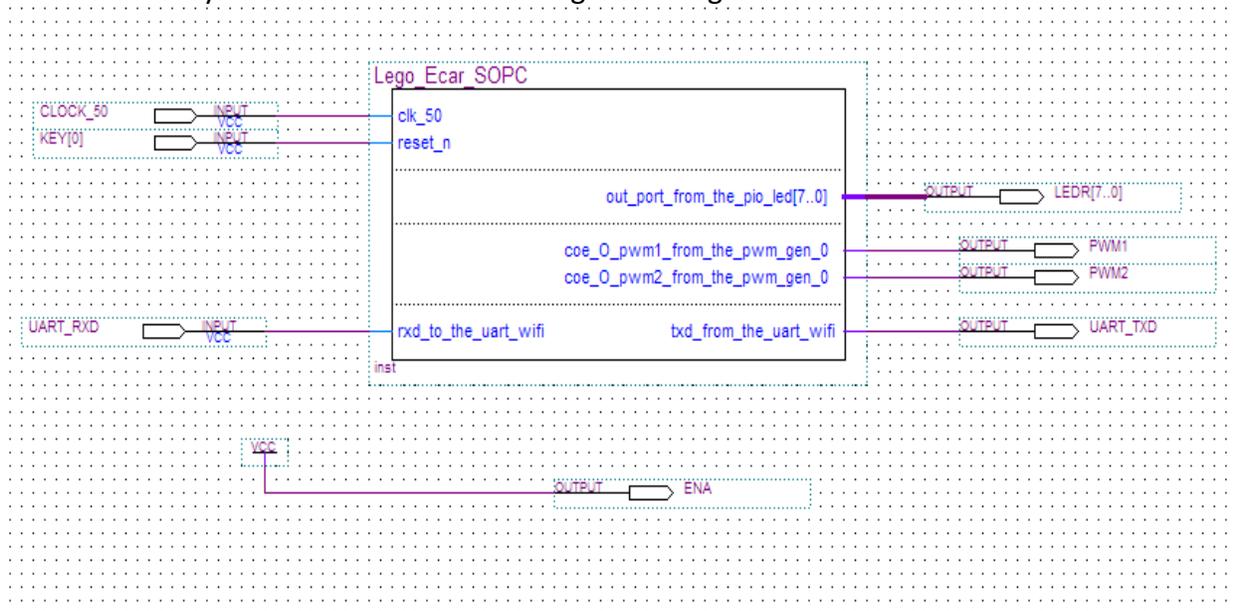


Figure 1-56

37. Choose **Save** icon in the tool bar. There will appear a window as shown in Figure 1-57. Click **Save**.

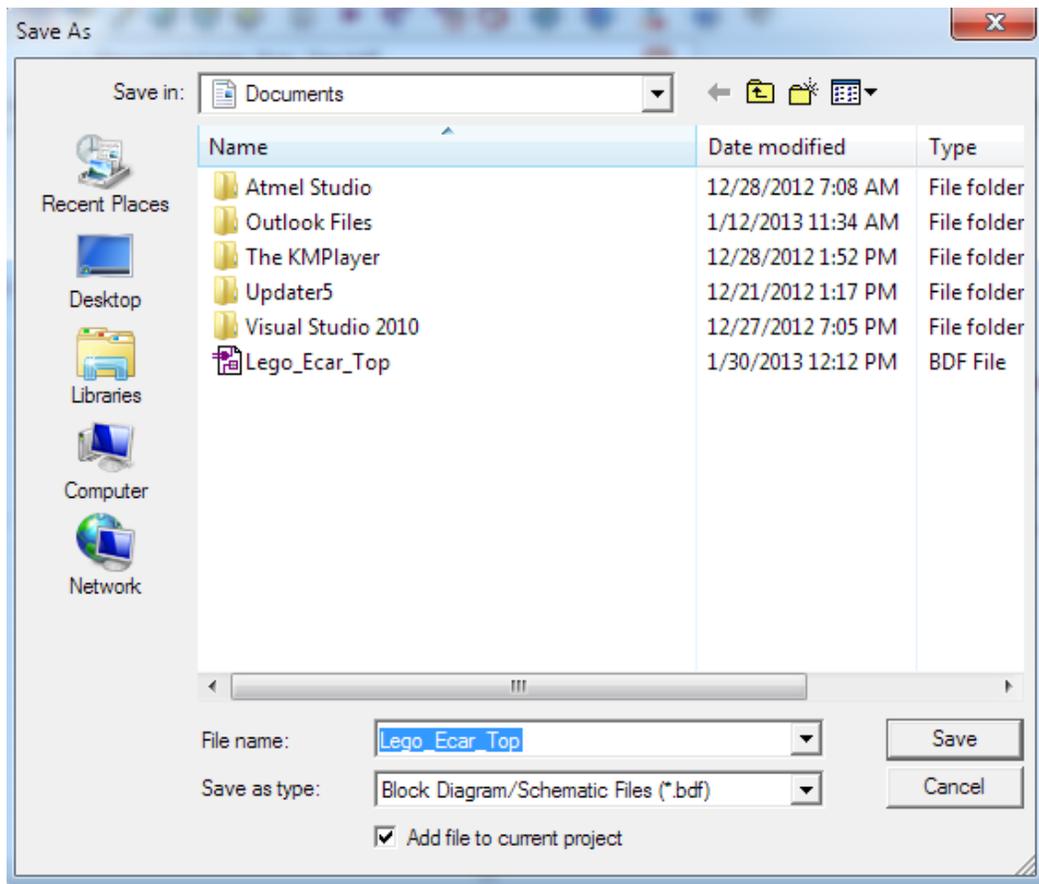


Figure 1-57 Save Bdf File

38. Choose **Processing > Start Compilation**. Figure 1-58 shows the compilation process. Note: You must select **Lego_Ecar_Top.bdf** as the top level entity. This can be done by right clicking and selecting the same.

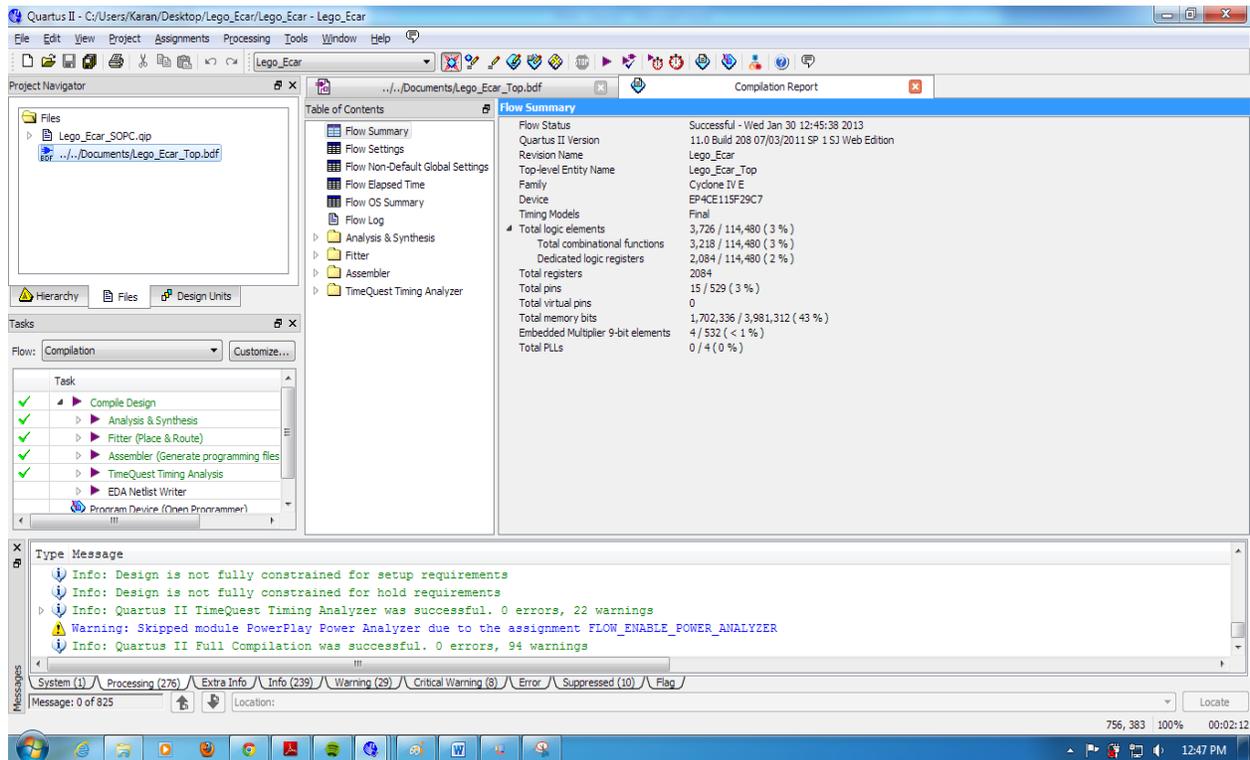


Figure 1-58: Compilation

Note: In the compilation, if there is the error which shows “Error: The core supply voltage of ‘1.0v’ is illegal for the currently selected part.”, you should modify the text “set_global_assignment –name NOMINAL_CORE_SUPPLY_VOLTAGE 1.0V” to “set_global_assignment –name NOMINAL_CORE_SUPPLY_VOLTAGE 1.2V”

39. A window that shows successful compilation will appear as shown in Figure 1-59.

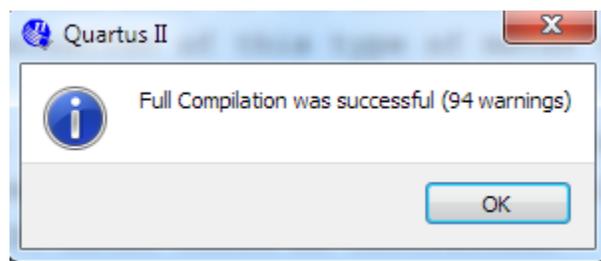


Figure 1-59: Successful

40. Choose **Assignments > Pin Planner** to open pin planner as shown in Figure 1-60. Figure 1-61 show blank pins.

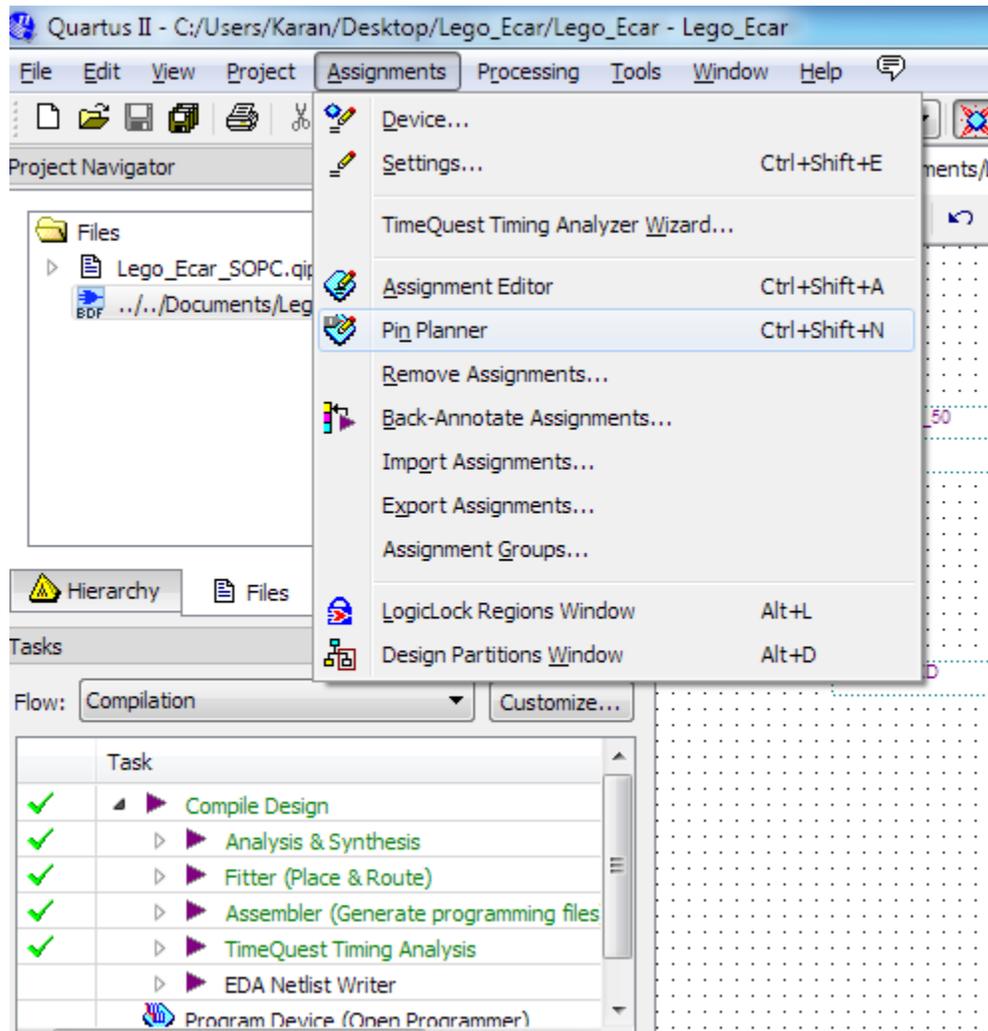


Figure 1-60 Pins menu

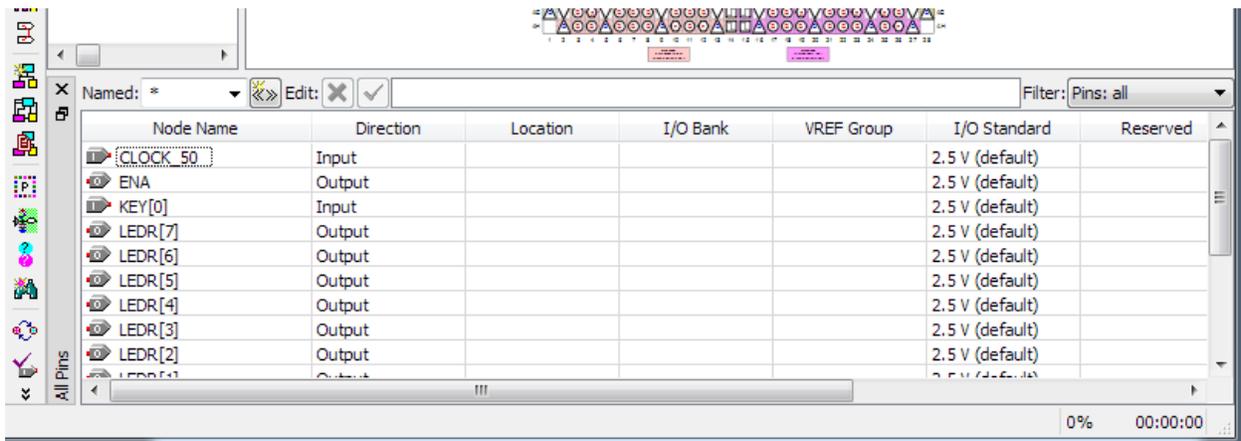


Figure 1-61 Blank Pins

41. Input Location value as shown in Figure 1-62.

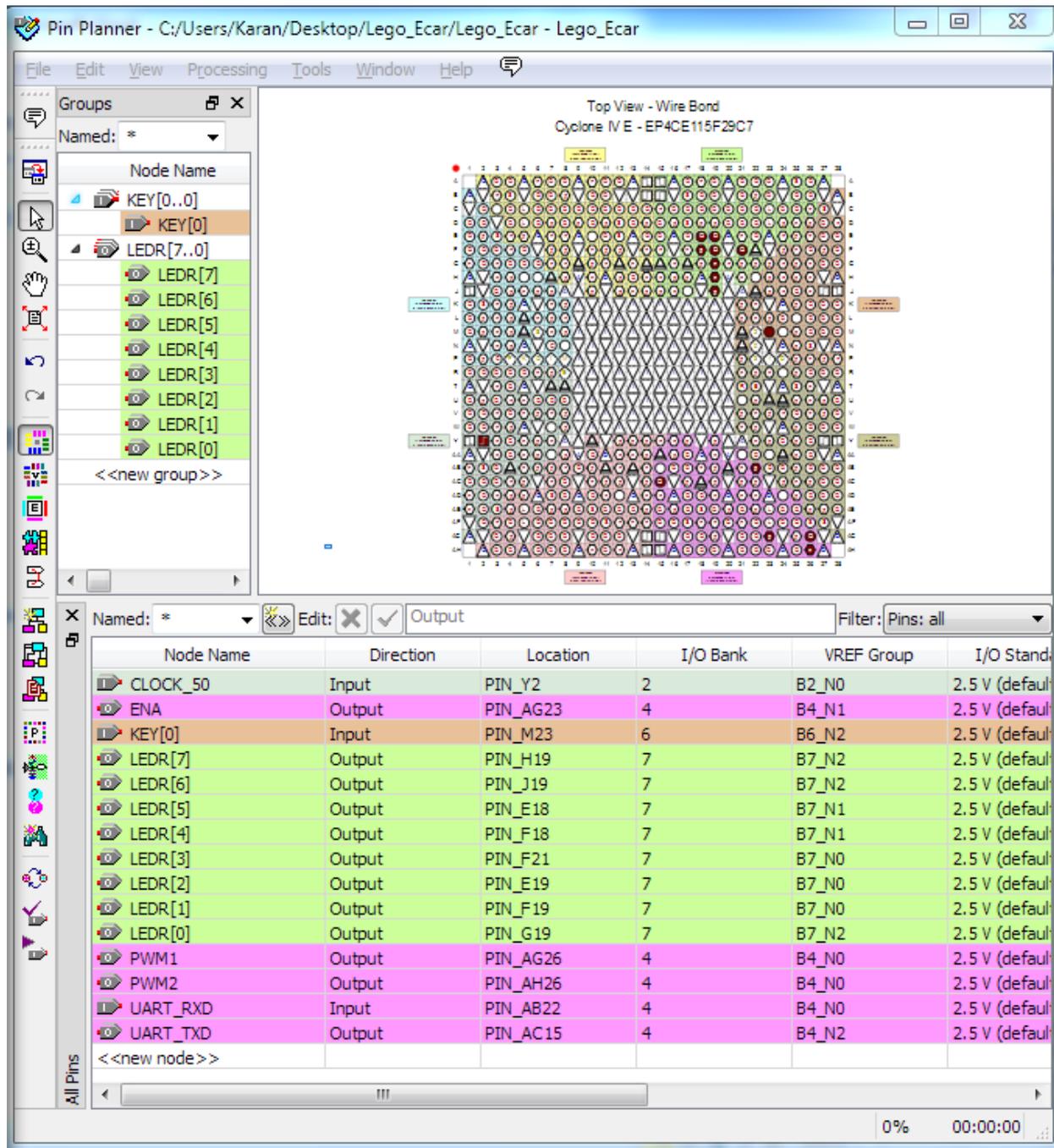


Figure 1-62

42. Close the **pin planner**. Restart compilation the project.

1.3 Download Hardware Design to Target FPGA

This section describes how to download the configuration file to the board.

Download the FPGA configuration file (i.e. the SRAM Object File (.sof) that contains the NIOS II standard system) to the board by performing the following steps:

1. Connect the board to the host computer via the USB download cable.
2. Apply power to the board.
3. Start the NIOS II IDE.
4. After the welcome page appears, click **Workbench**.
5. Choose **Tools->Quartus II Programmer**.
6. Click **Auto Detect**. The device on your development board should be detected automatically.
7. Click the top row to highlight it.
8. Click **Change File**.
9. Browse to the myfirst_niosii project directory.
10. Select the programming file (myfirst_niosii.sof) for your board.
11. Click **OK**.
12. Click **Hardware Setup** in the top, left corner of the Quartus II programmer window. The Hardware Setup dialog box appears.
13. Select USB-Blaster from the **Currently selected hardware** drop-down list box.

Note: If the appropriate download cable does not appear in the list, you must first install drivers for the cable. Refer to Quartus II Help for information on how to install the driver. See Figure 1-63.

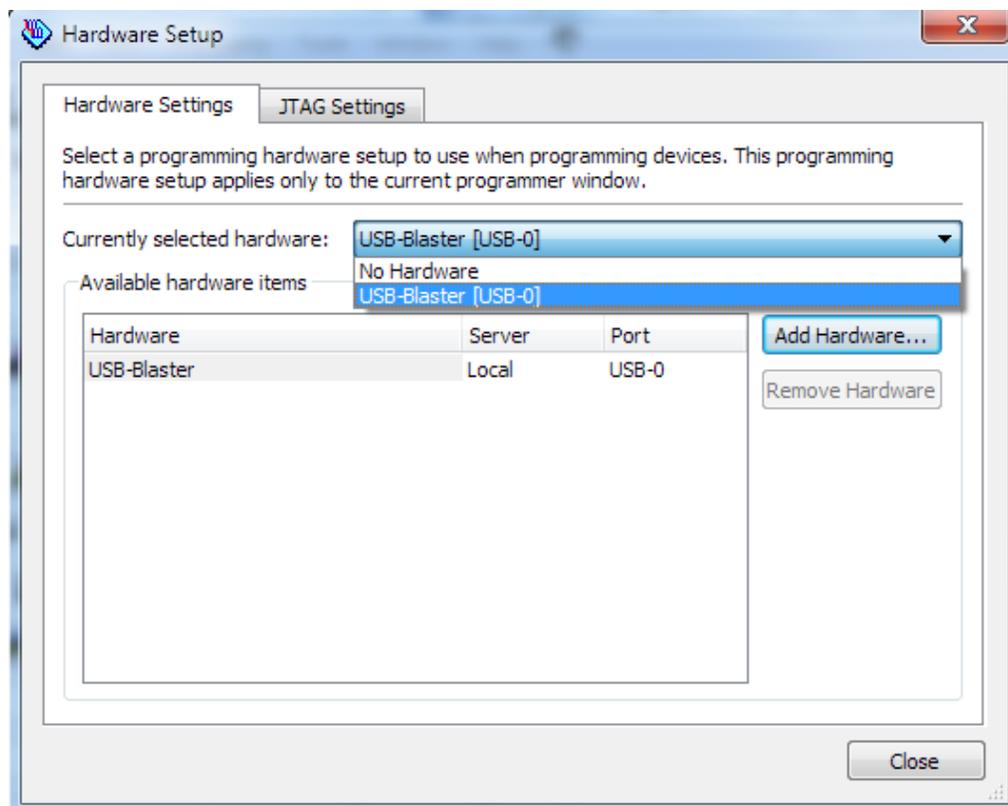
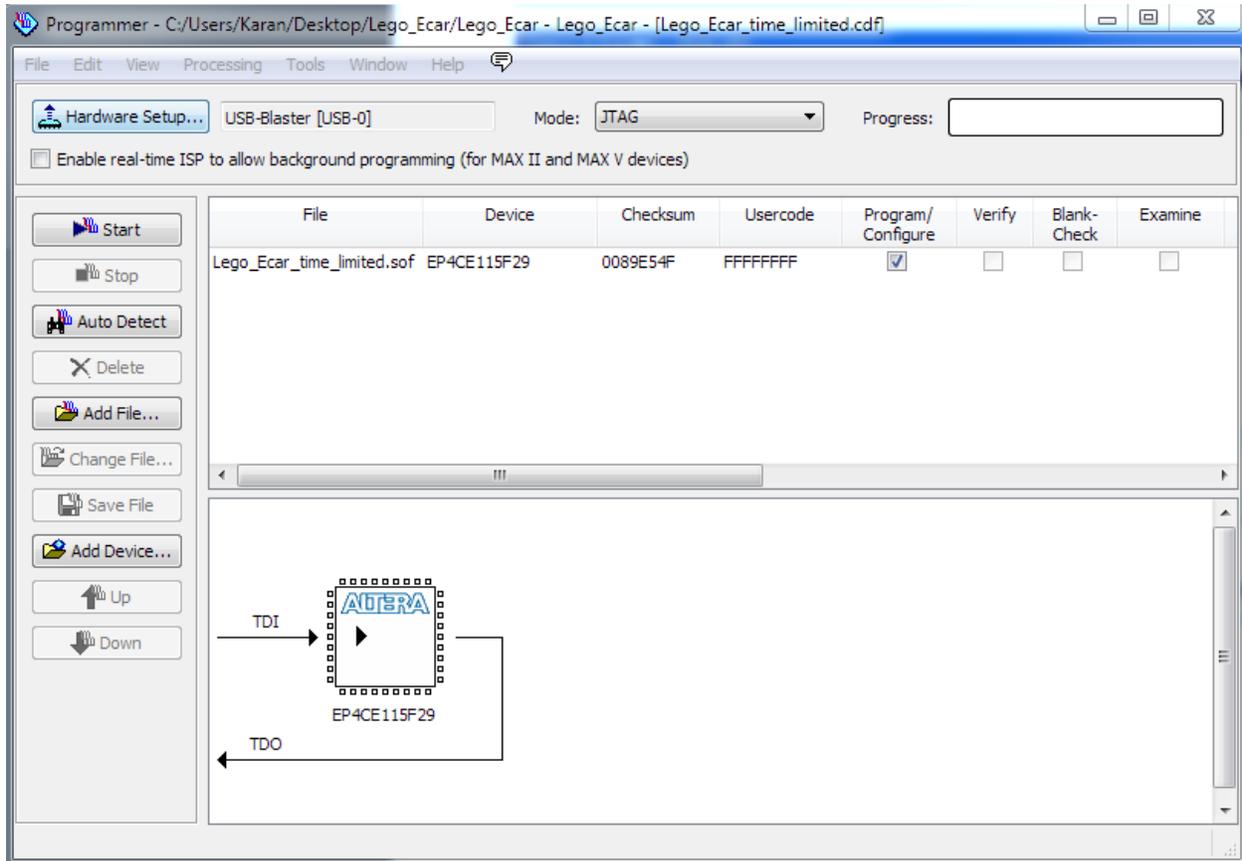


Figure 1-63 Hardware Setup Window

14. Click **Close**.
15. Turn on the **Program/Configure** option for the programming file. (See Figure 1-64 for an example).
16. Click **Start**.

**Figure 1-64 Quartus II Programmer**

The Progress meter sweeps to 100% after the configuration finished. When configuration is complete, the FPGA is configured with the Nios II system, but it does not yet have a C program in memory to execute.

Chapter 2 NIOS II IDE Build Flow

This Chapter covers build flow of Nios II C coded software program. The Nios II IDE build flow is an easy-to-use graphical user interface (GUI) that automates build and makefile management. The Nios II IDE integrates a text editor, debugger, the Nios II flash programmer, the Quartus II Programmer, and the Nios II C-to-Hardware (C2H) compiler GUI. The included example software application templates make it easy for new software programmers to get started quickly. In this section you will use the Nios II IDE to compile a simple C language example software program to run on the Nios II standard system configured onto the FPGA on your development board. You will create a new software project, build it, and run it on the target hardware. You will also edit the project, re-build it, and set up a debug session.

2.1 Create the hello_world Example Project

In this section you will create a new NIOS II C/C++ application project based on an installed example. To begin, perform the following steps in the NIOS II IDE:

1. Return to the NIOS II IDE.

Note: you can close the Quartus II Programmer or leave it open in the background if you want to reload the processor system onto your development board quickly.

2. Choose **File > Switch Workspace** to switch workspace. See Figure 2-1 and Figure 2-2.

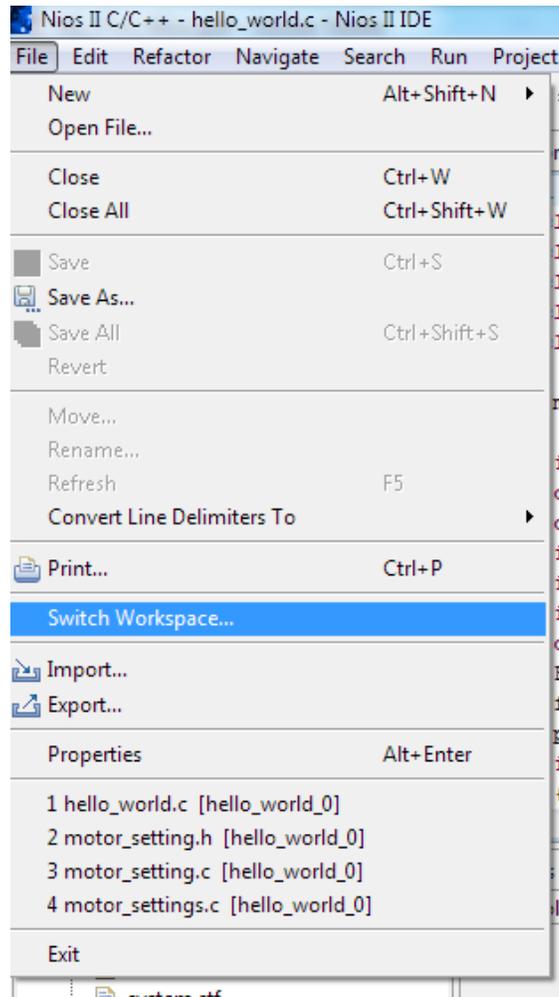


Figure 2-1 Switch Workspace (1)

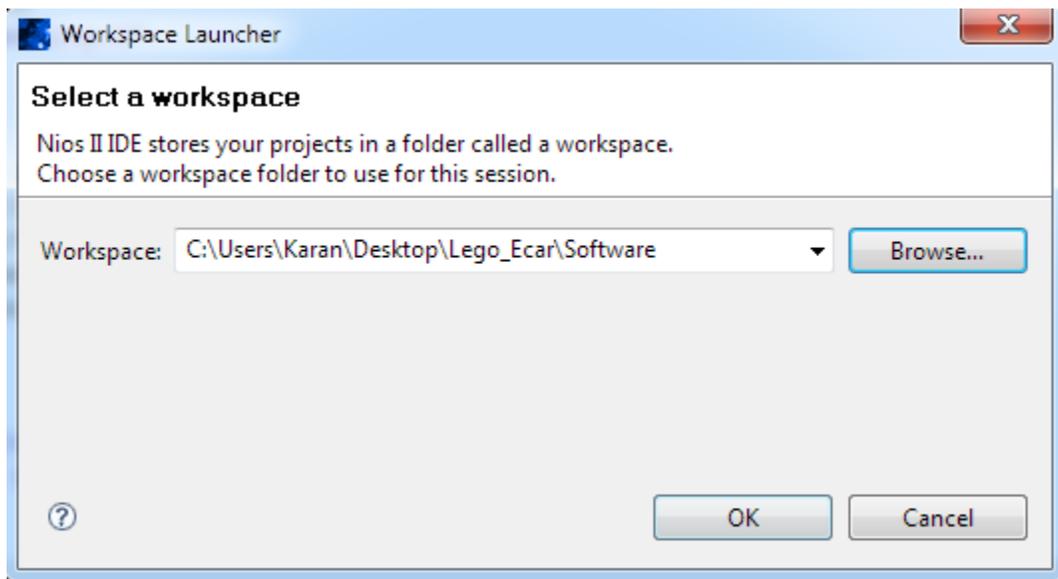


Figure 2-2 Switch Workspace (2)

3. Choose **File->New->NIOS II C/C++ Application** to open the New Project Wizard.

4. In the New Project wizard, make sure the following things:

Select the **Hello World** project template.

- Give the project a name. (**hello_world_0** is default name)
- Select the target hardware system PTF file that locates in where the previously created hardware project resides as shown in Figure 2-3.

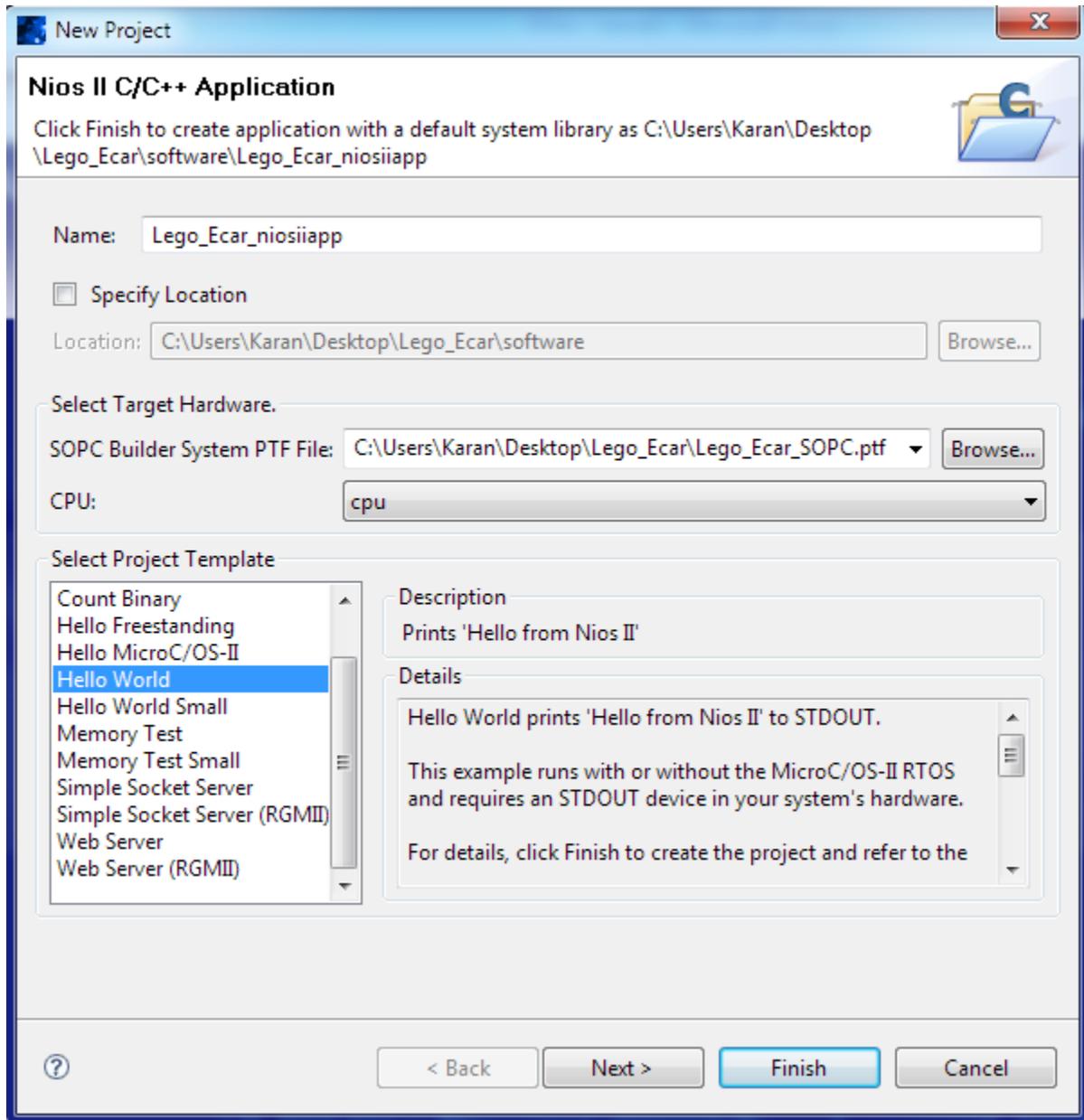


Figure 2-3 Nios II IDE New Project Wizard

5. Click **Finish**. The Nios II IDE creates the **Lego_Ecar_niosiiapp** project and returns to the Nios II C/C++ project perspective. See Figure 2-4.

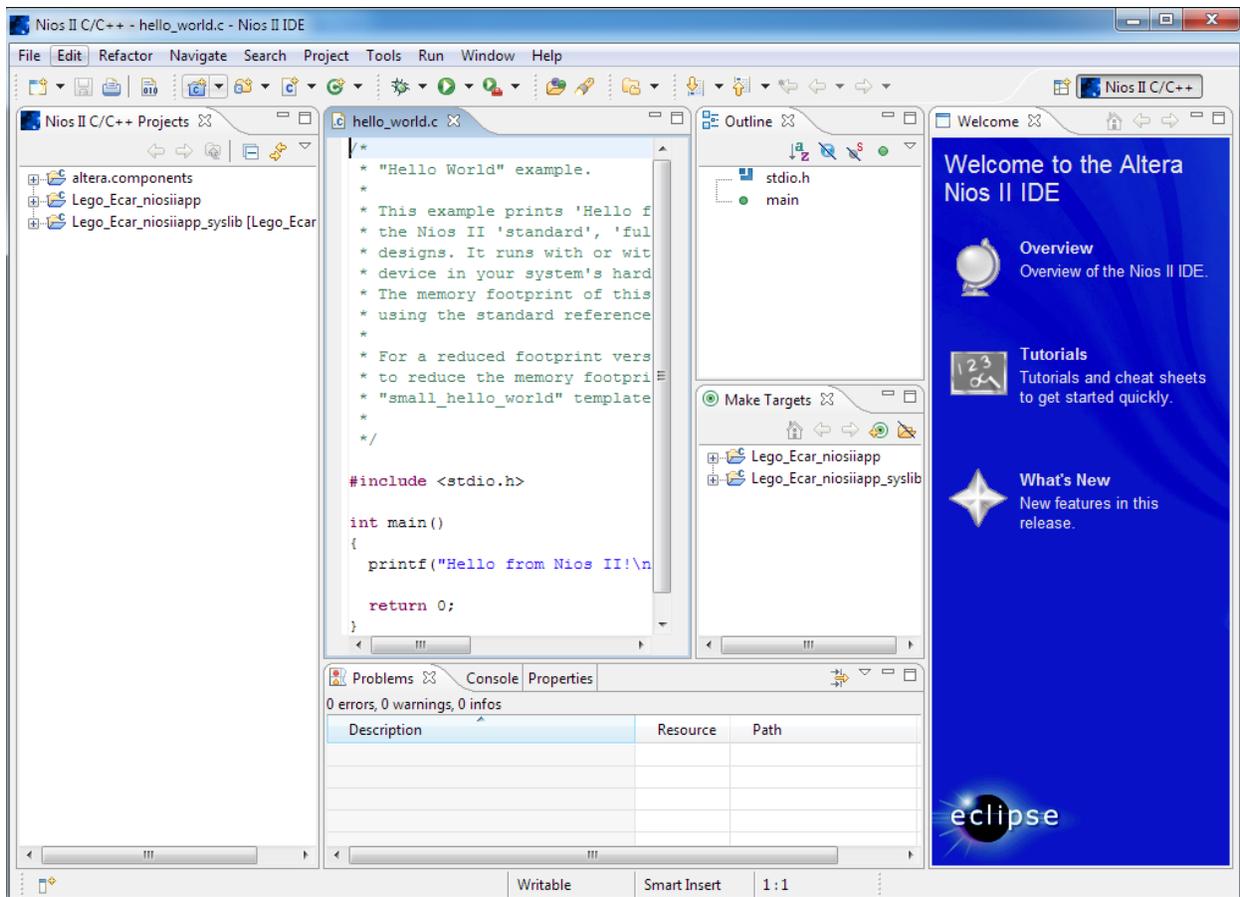


Figure 2-4 Nios II IDE C++ Project Perspective for hello_world_0

When you create a new project, the NIOS II IDE creates two new projects in the NIOS II C/C++ Projects tab:

- **Lego_Ecar_niosiiapp** is your C/C++ application project. This project contains the source and header files for your application.

- **Lego_Ecar_niosiiapp_syslib** is a system library that encapsulates the details of the Nios II system hardware.

Note: When you build the system library for the first time the NIOS II IDE automatically generates files useful for software development, including:

- Installed IP device drivers, including SOPC component device drivers for the NIOS II hardware system
- Newlib C library, which is a richly featured C library for the NIOS II processor.
- NIOS software packages which includes NIOS II hardware abstraction layer, NicheStack TCP/IP

Network stack, NIOS II host file system, NIOS II read-only zip file system and Micrium's μ C/OS-II real time operating system(RTOS).

- **system.h**, which is a header file that encapsulates your hardware system.
- **alt_sys_init.c**, which is an initialization file that initializes the devices in the system.
- **Lego_Ecar_niosiiapp.elf**, which is an executable and linked format file for the application located in **Lego_Ecar_niosiiapp** folder under Debug.

2.2 Build and Run the Program

In this section you will build and run the program to execute the compiled code.

To build the program, right-click the **Lego_Ecar_niosiiapp** project in the Nios II C/C++ Projects tab and

choose **Build Project**. The **Build Project** dialog box appears and the IDE begins compiling the project. When compilation completes, a message 'Build complete' will appear in the Console tab.

The compilation time varies depending on your system. See Figure 2-5 for an example.

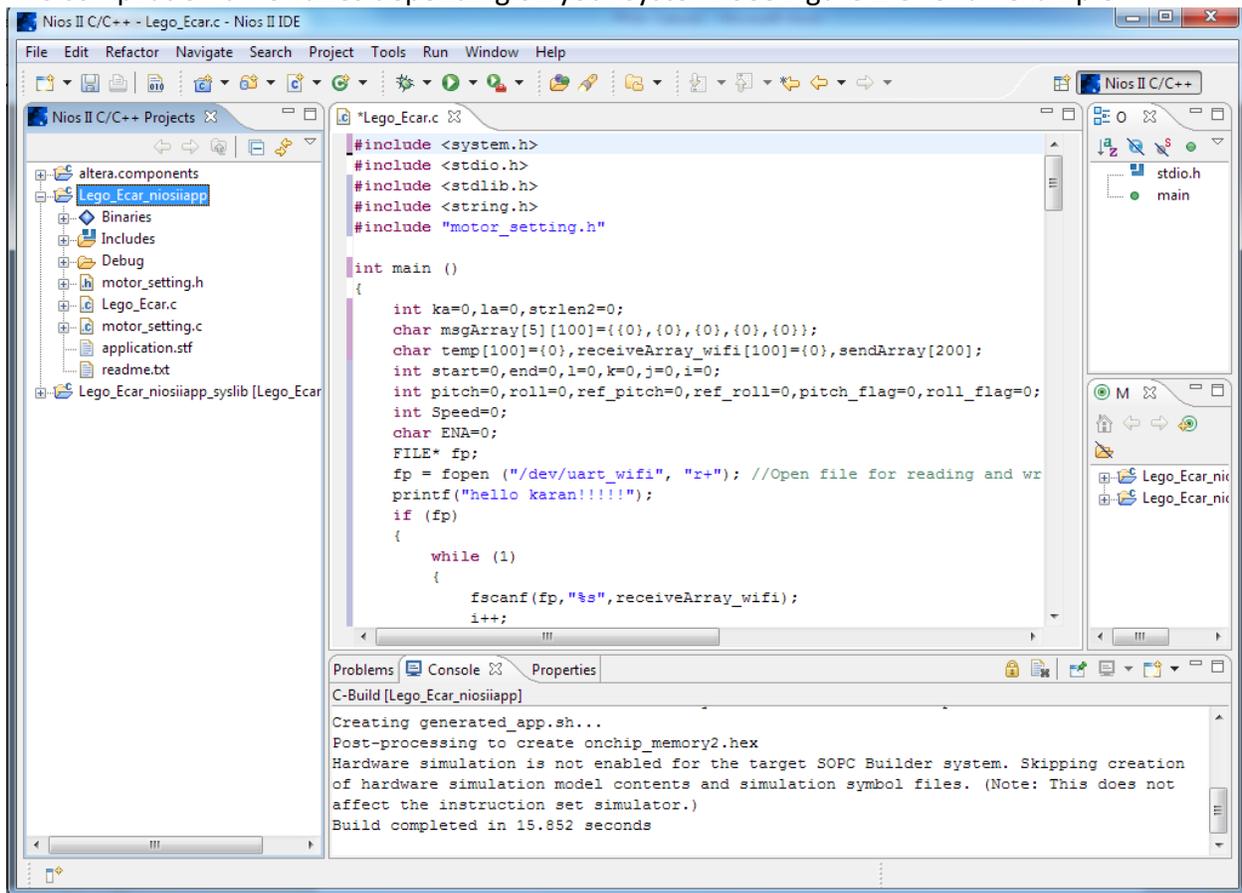


Figure 2-5 Nios II IDE Build Completed

After compilation complete, right-click the **hello_world_0** project, choose **Run As**, and choose **NIOS II Hardware**. The IDE begins to download the program to the target FPGA development board and begins execution. When the target hardware begins executing the program, the message '**Hello from Nios II!**' appears in the NIOS II IDE Console tab.

