Designing ADC-DAC System from Scratch for DE2

Tutorial

System Level Solutions, Inc. (USA)
14100 Murphy Avenue
San Martin, CA 95046
(408) 852 - 0067

http://www.slscorp.com

Version : 0.1.1

Date : July 10, 2007

MPD_SO03_TT1_011

About this Tutorial

Introduction

This tutorial provides all the steps for creating a system for the ADC-DAC board from the scratch with the DE2 board. Also it shows how to create, compile, debug and run a C/C++ program using the Nios II IDE.

Table below shows the Revision history of Designing ADC-DAC System from Scratch for DE2.

Version	Date	Description
0.1.1	July 2007	First Publication of Designing ADC-DAC System from Scratch for DE2 Tutorial.

How to Contact SLS

For the most up-to-date information about SLS products, go to the SLS worldwide website at http://www.slscorp.com. For additional information about SLS products, consult the source shown below.

Information Type	E-mail
Product literature services, SLS literature services, Non-technical customer services, Technical support.	support@slscorp.com

Typographic Conventions

Designing ADC-DAC System from Scratch for DE2, Cyclone II Edition uses the typographic conventions shown as below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	All Headings and Sub Headings Titles in a document are displayed in bold type with initial capital letters; Example: **Introduction, Creating a Quartus II Project.**
Bold Type with Italic Letters	All Definitions, Figure and Table Headings are displayed in Italics. Examples: **Figure 2-1. Create a New System Dialog Box, Figure 2-2. SOPC Builder**
1. **2.**	Numbered steps are used in a list of items, when the sequence of items is important. such as steps listed in procedure.
• ■	Bullets are used in a list of items when the sequence of items is not important.
Courier type	Anything that must be typed exactly as it appears is shown in Courier type. For Example `cpu`
	The hand points to information that requires special attention.
	The caution indicates required information that needs special consideration and understanding and should be read prior to starting or continuing with the procedure or process.
	The warning indicates information that should be read prior to starting or continuing the procedure or processes.
	The feet direct you to more information on a particular topic.

Contents

Figures

1. Introduction

This tutorial walks you through the hardware & software development flow. It shows you how to use SOPC Builder and the Quartus II software to create and use your own Nios II system

This tutorial is basically for users who are new to the Nios II processor as well as users who are new to the concept of using embedded systems in FPGA. This tutorial guides you through the steps necessary to create and compile a ADC-DAC System Design, called **sc_adcdac_refdes_de2**. This simple, single-master ADC-DAC control system consists of a Nios II embedded processor and associated system peripherals as well as interconnections for use with the input & output hardware available on the DE2 board.

This tutorial is divided into the following three sections:

- ***'Designing & Compiling'*** **-** Teaches you how to use SOPC builder to create the ADC-DAC module in block design file (.bdf) and how to compile the ADC-DAC design using the Quartus II Compiler.
- ***'Programming'*** **-** Teaches you how to use the Quartus II Programmer and the USB-Blaster cable to configure the FPGA on DE2 board.
- ***'Running the Software on Your Nios II System'*** **-** Provides the instructions for running software on your Nios II system using the Nios II Integrated Development Environment (IDE).

Hardware & Software Requirements

The user will require following hardware & software

- A PC running with Win 2000/XP OS
- Nios II Embedded Processor
- The Quartus II Software, version 6.1
- DE2 Development Kit

2. Designing & Compiling

To use the instructions in this section, you need to be familiar with the Quartus II software interface-specifically tool bars. Refer to Quartus II help for more information about using the Quartus II software.

Creating a Quartus II Project

Here are the steps to create a new Quartus II project:

1. Open the **Quartus II**.
2. Choose **File>New Project Wizard.**
3. Click **Next**.
4. Select Working Directory of the **Project**, Name of the project as **'sc_adcdac_refdes_de2'** & top-level entity as **'sc_adcdac_refdes_de2'**.
5. Click **Next.**
6. Click **Next.**
7. Select the family as **'Cyclone II'.**
8. We will select the **FPGA** (which is Cyclone II EP2C35F672C6) **for DE2 board**, so under Filters / Speed Grade select **6.** Then under Available devices: Select **'EP2C35F672C6'**. Click **Next**.
9. Click **Next.**
10. Click **Finish.**

Start SOPC builder

SOPC builder is a software tool that allows you to create a fully functioning, ADC-DAC system module. A complete ADC-DAC system module contains a Nios II embedded processor and its associated peripherals.

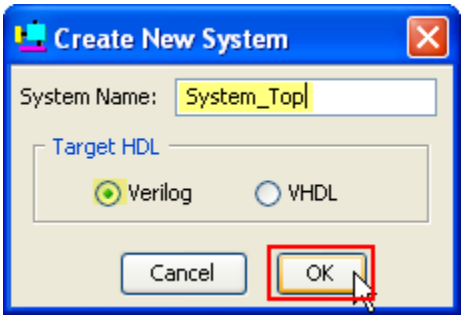
To start SOPC builder, perform the following steps:

1. Open the **Quartus II software**.
2. Choose **SOPC Builder** (**Tools menu**). SOPC Builder displays the Create New System dialog box.

3. Type **'System_Top'**. See Figure 2-1.
4. Specify **Verilog or VHDL** in HDL Language field.

SOPC Builder generates plain text Verilog HDL or VHDL for all of its native components depending on the language you choose.

Figure 2-1. Create New System Dialog Box



5. Click **OK**. The Altera SOPC Builder - System_Top window appears and the System Contents tab is displayed.

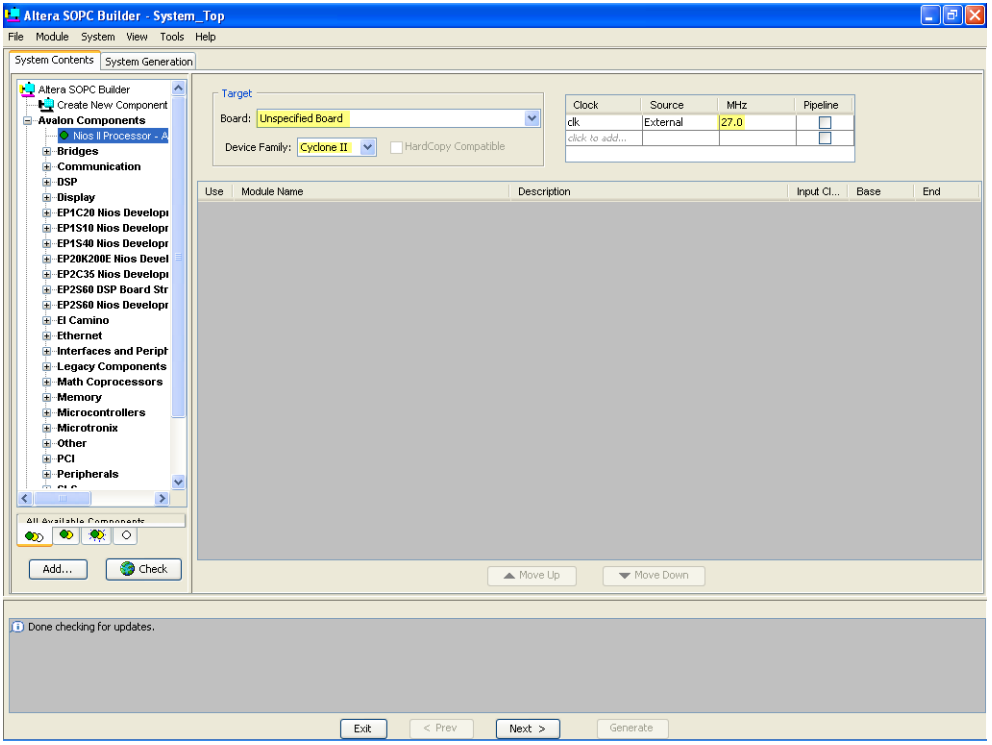
You are now ready to set the speed and add the Nios II CPU and peripherals to your system. The components you will be adding are located in the module pool on the left hand side of the System Content tab. See Figure 2-2.

Specify Target Hardware Settings

The functionality of the SOPC Builder system depends on the hardware on which it will run. Thus, specifying the target board is the first step in creating a system.

- Choose a **board type** in the Target pull-down menu. As DE2 is not yet added in the menu, keep the **'Unspecified Board'**.
- Select the System Clock Frequency as **27Mhz**.

Figure 2-2. SOPC Builder



Adding CPU & Peripherals

This section describes adding following modules to the SOPC Builder.

- Nios II 32-bit CPU
- JTAG UART
- Timer
- Tristate Bridge
- SRAM Memory
- Parallel I/Os for
  - ADC Output Start Bit
  - ADC Output Enable Bit
  - ADC Output Address Latch Enable Bit
  - ADC Output Address Bits
  - ADC Input End of Character Bit

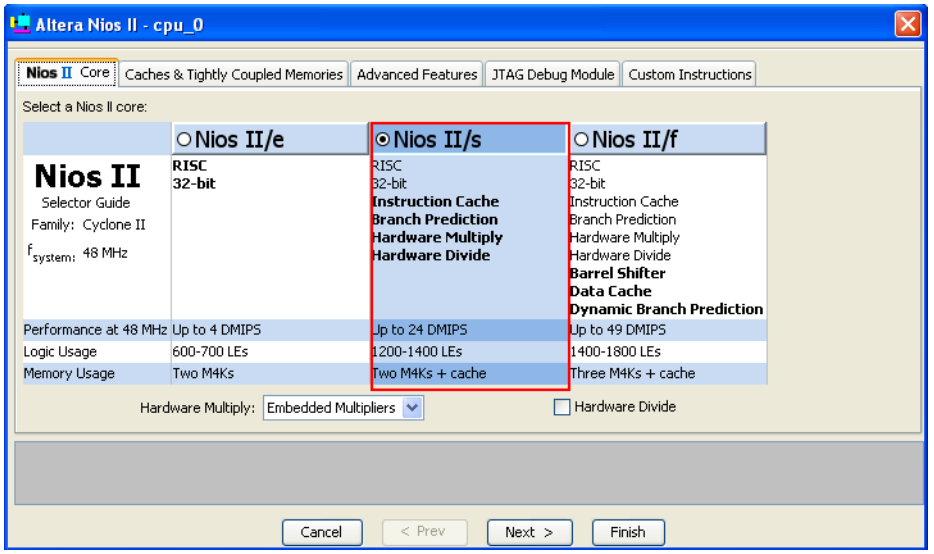
- ADC Input Data Byte
- DAC Output Data Byte
- LEDs
- Pushbutton Switches

Nios II 32-bit CPU

To add the Nios II 32-bit CPU, named **CPU**, perform the following steps:

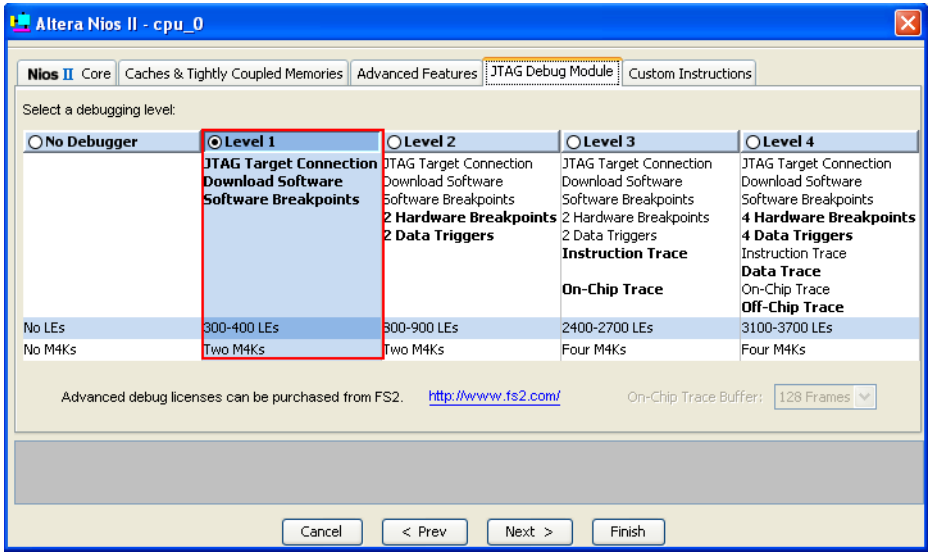
1. Under Avalon Modules, select **Nios II Processor - Altera Corporation**.
2. Click **Add**. The Nios II configuration wizard titled Altera Nios II - cpu_0 displays.
3. Specify the following options in the **Nios II Core tab**.
   - Select the **processor core: Nios II/s** as in Figure 2-3.
   - Instruction Cache Size: **4Kbytes**

Figure 2-3. Nios II Configuration Wizard - Nios II Core Tab



4. Click the **JTAG Debug Module** tab and choose the **selected tab** shown in Figure 2-4.

Figure 2-4. Nios II Configuration Wizard -JTAG Debug Module



5. Clicking **Finish.** You return to the SOPC Builder **System Content** tab and an instance of the CPU named cpu_0 now appears in the table of available components.
6. Right click **cpu_0** and select **Rename**.
7. Type `cpu` and press **Enter**.

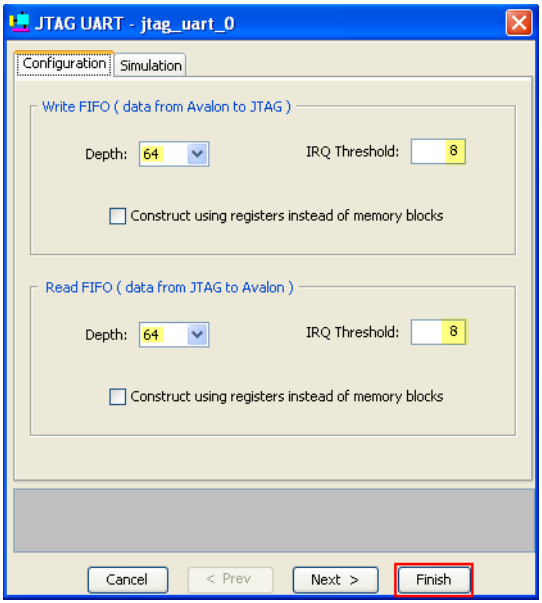
It is recommended to rename the components for avoiding ambiguity in further design process.

JTAG UART

The **JTAG UART** interface component is added to reduce the number of connections necessary to 'talk' to the Nios II. To add it

1. Select **Communication > JTAG UART** and click **Add**.. The JTAG UART - jtag_uart_0 wizard displays as shown in Figure 2-5.

Figure 2-5. JTAG UART Configuration Wizard



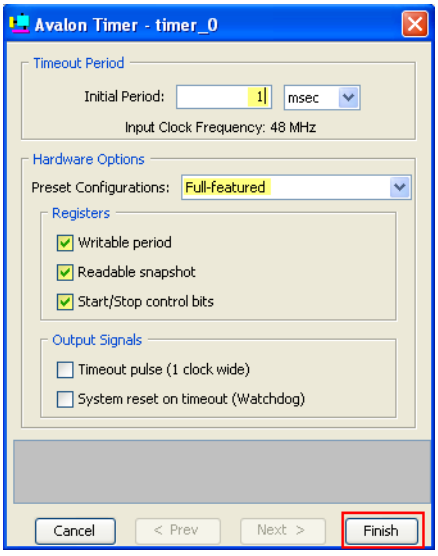
2. Accept the default options by clicking **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the JTAG UART named jtag_uart_0 now appears in the table of available components.
3. Right click **jtag_uart_0** and select **Rename**.
4. Type `jtag_uart` and press **Enter**.

Timer

The Timer is necessary for some of the default device drivers provided in the HAL system library, for example, the JTAG UART. To add the timer perform the following steps:

1. Choose **Other > Interval Timer** and click **Add.**
2. Leave the default settings in the Avalon Timer - **timer_0 window**. Figure 2-6.

Figure 2-6. Timer Configuration Wizard



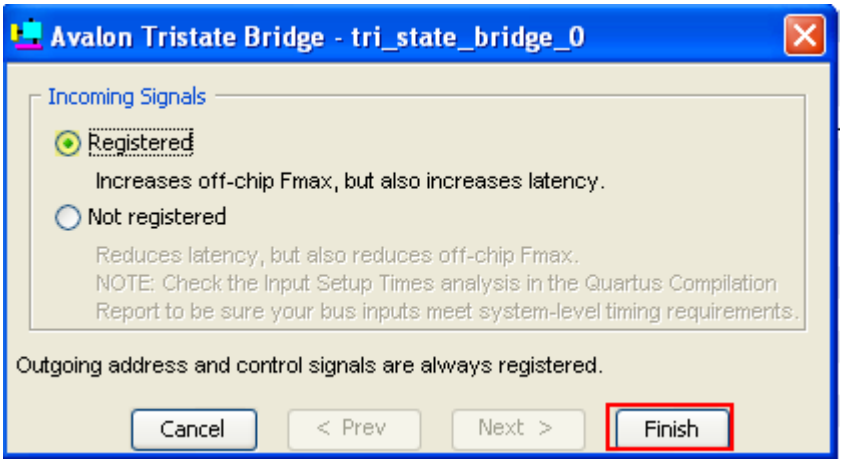
3. Click **Finish**. You return to the SOPC Builder **System Contents** tab and an instance of the Timer named timer_0 now appears in the table of available components.
4. Right click **timer_0** and select **Rename**.
5. Type `timer` and press **Enter**.

Tristate Bridge

For the ADC-DAC system to communicate with Tristate memory external to the FPGA on the DE2, you must add a bridge between the Avalon bus and external memory controller . To add this:

1. Select **Bridges > Avalon Tri-State Bridge** and click **Add.** The **Avalon Tri-State Bridge - tri_state_bridge_0** wizard displays. See Figure 2-7.

Figure 2-7. Avalon Tristate Bridge Configuration Wizard



2. Click **Finish**. You return to the SOPC Builder **System Content** tab and an instance of the Tri_state_bridge named tri_state_bridge_0 now appears in the table of available components.
3. Right click **tri_state_bridge_0** and select **Rename**.
4. Type `tri_state_bridge` and press **Enter**.

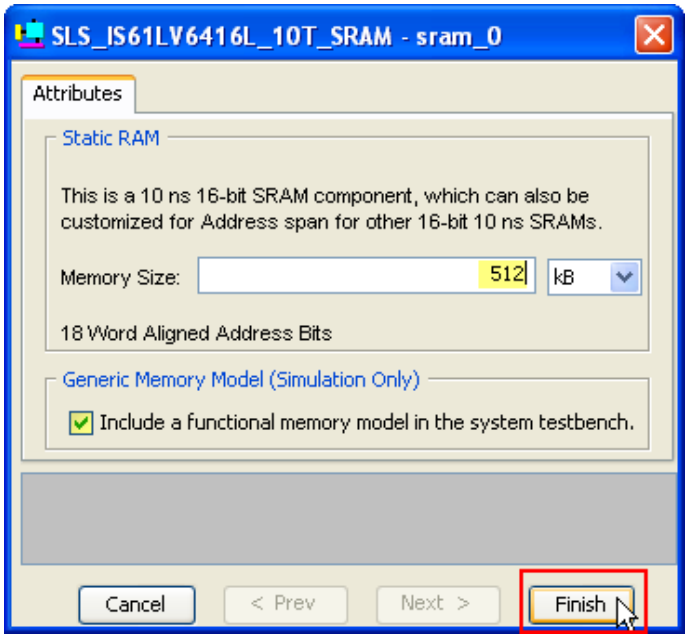
SRAM Memory

Depending on which hardware you are using user has to select the external SRAM or any memory. We shall be using SRAM avialable on DE2 board.

SLS has developed a generic SRAM controller which is provided in the reference design directory.
Therefore, before following the steps mentioned below, copy the contents of the folder ***<../SLS_ADC-DAC_Board\ADC-DAC_Reference_Designs/Components>*** to ***<Quartus installation path>/quartus/sopc_builder/components/*** directory. Go to SOPC builder **File** menu and click **Refresh Component List.**
To add SRAM Memory Controller perform the following steps:

1. Select **Memory** > **SLS_IS61LV6416L_10T_SRAM-sram_0** and click **Add.**
2. The SRAM wizard displays. See Figure 2-8.
3. Set Memory Size = **512 kB**

Figure 2-8. SRAM Memory Configuration Wizard



4. Click **Finish**. You return to the SOPC Builder **System Content** tab and an instance of the SRAM named sram_0 now appears in the table of available components.
5. Right click **sram_0** and select **Rename**.
6. Type `sram` and press **Enter**.

Parallel I/O for ADC Output Start Bit

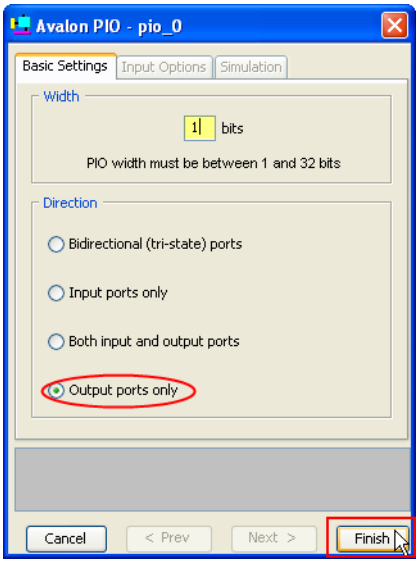
To provide an interface for ADC Output Start Bit on DE2 Board, add the PIO by performing the following steps:

1. Select **Other** > **PIO** (Parallel IO) and click **Add.**
2. Specify the Options. See Figure 2-9.

- Width =**1 bits**
- Direction = **Output ports only**.

3. Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
4. Right click **pio_0** and select **Rename**.
5. Type `adc_start` and press **Enter**.

Figure 2-9. PIO Configuration Wizard For ADC Output Start Bit



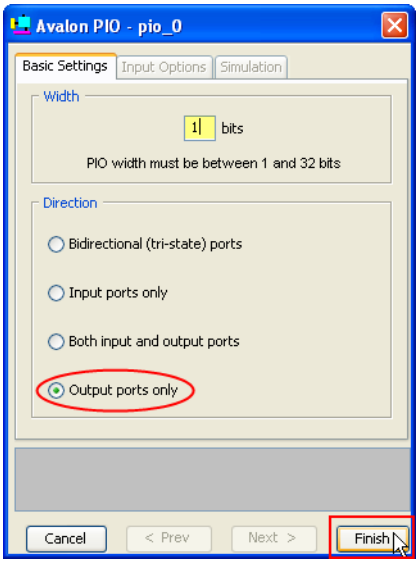
Parallel I/O for ADC Output Enable Bit

To provide an interface for ADC Output Enable Bit on DE2 Board, add the PIO by performing the following steps:

1. Select **Other** > **PIO** (Parallel IO) and click **Add.**
2. Specify the Options. See Figure 2-10.
   - Width =**1 bits**
   - Direction = **Output ports only**.

3. Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
4. Right click **pio_0** and select **Rename**.
5. Type `adc_oe` and press **Enter**.

Figure 2-10. PIO Configuration Wizard For ADC Output Enable Bit



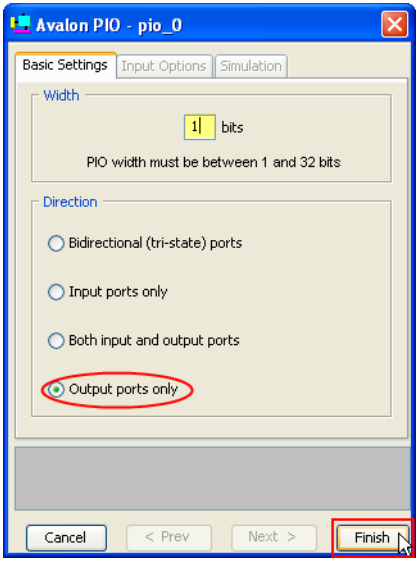
Parallel I/O for ADC Output Address Latch Enable Bit

To provide an interface for ADC Output Address Latch Enable Bit on DE2 Board, add the PIO by performing the following steps:

1. Select **Other** > **PIO** (Parallel IO) and click **Add.**
2. Specify the Options. See Figure 2-11.
   - Width **=1 bits**
   - Direction = **Output ports only**.
3. Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
4. Right click **pio_0** and select **Rename**.

5. Type `adc_ale` and press **Enter**.

Figure 2-11.PIO Configuration Wizard for ADC Output Address Latch Enable Bit

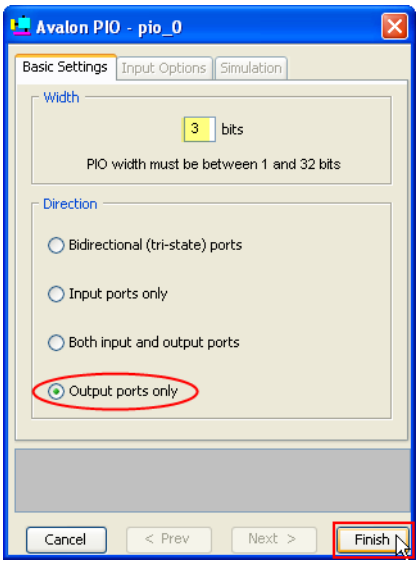


Parallel I/O for ADC Output Address Bits

To provide an interface for ADC Output Address Bits on DE2 Board, add the PIO by performing the following steps:

1. Select **Other** > **PIO** (Parallel IO) and click **Add.**
2. Specify the Options. See Figure 2-12.
   - Width =**3 bits**
   - Direction = **Output ports only**.
3. Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
4. Right click **pio_0** and select **Rename**.
5. Type `adc_add` and press **Enter**.

Figure 2-12. PIO Configuration Wizard For ADC Output Address Bits

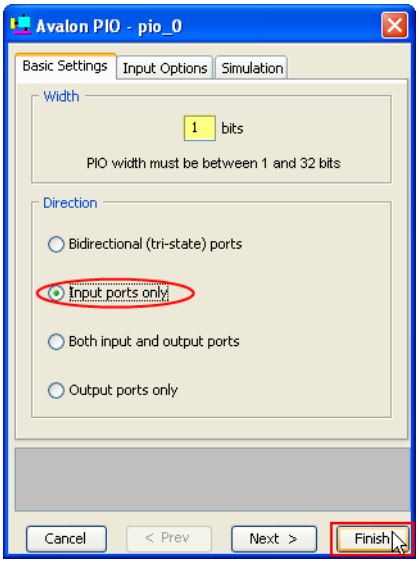


Parallel I/O for ADC Input End Of Character Bit

To provide an interface for ADC Input End of Character Bit on DE2 Board, add the PIO by performing the following steps:

1. Select **Other** > **PIO** (Parallel IO) and click **Add.**
2. Specify the Options. See Figure 2-13.
   - Width =**1 bits**
   - Direction = **Input ports only**.
   - Accept **Input Options** and **Simulation** as default.
3. Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
4. Right click **pio_0** and select **Rename**.
5. Type `adc_eoc` and press **Enter**.

Figure 2-13.PIO Configuration Wizard For ADC Input End Of Character Bit

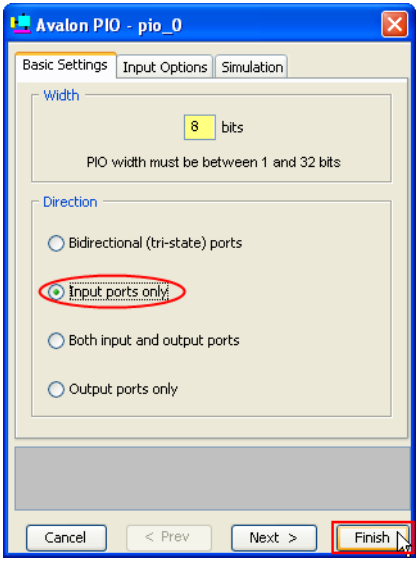


Parallel I/O for ADC Input Data Byte

To provide an interface for ADC Input Data Byte on DE2 Board, add the PIO by performing the following steps:

1. Select **Other** > **PIO** (Parallel IO) and click **Add.**
2. Specify the Options. See Figure 2-14.
   - Width =**8 bits**
   - Direction = **Input ports only**.
   - Keep the default settings under **Input Options** and **Simulation** as default.
   - Click **Finish**
3. You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
4. Right click **pio_0** and select **Rename**.
5. Type `adc_data` and press **Enter**.

Figure 2-14. PIO Configuration Wizard For ADC Input Data Byte

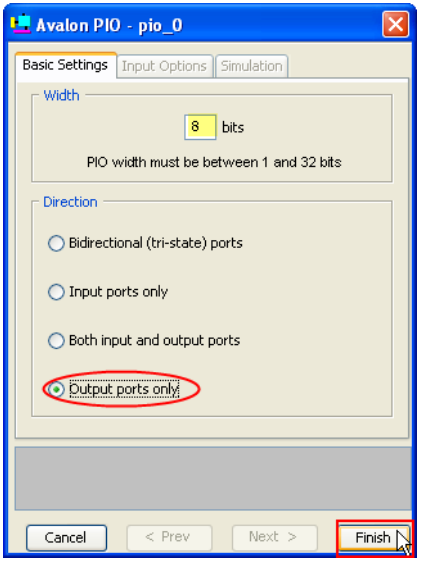


Parallel I/O for DAC Output Data Byte

To provide an interface for DAC Output Data Byte on DE2 Board, add the PIO by performing the following steps:

1. Select **Other** > **PIO** (Parallel IO) and click **Add.**
2. Specify the Options. See Figure 2-15.
   - Width **=8 bits**
   - Direction = **Output ports only**.
   - Click **Finish.**
3. You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
4. Right click **pio_0** and select **Rename**.
5. Type `dac_data` and press **Enter**.

Figure 2-15. PIO Configuration Wizard For DAC Output Data Byte

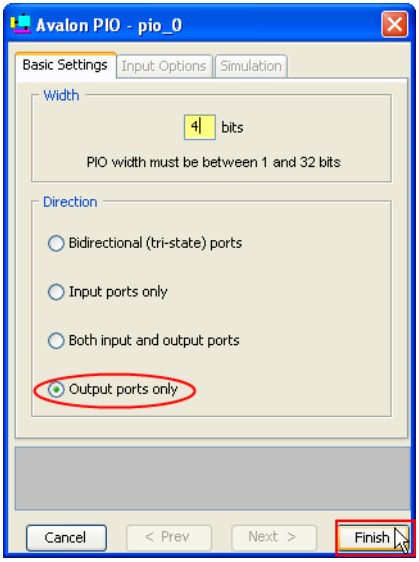


Parallel I/O for LEDs

To provide an interface for led's on DE2 Board, add the PIO by performing the following steps:

1. Select **Other** > **PIO** (Parallel IO) and click **Add.**
2. Specify the Options. See Figure 2-16.
   - Width =**4 bits**
   - Direction = **Output ports only**.
   - Click **Finish.**
3. You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
4. Right click **pio_0** and select **Rename**.
5. Type `led_pio` and press **Enter**.

Figure 2-16. PIO Configuration Wizard For LEDs

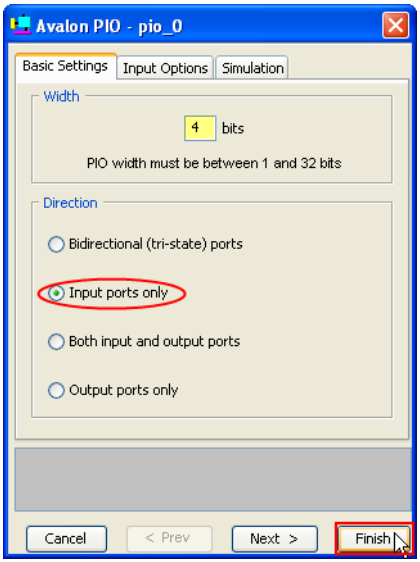


Parallel I/O for Pushbutton Switches

To provide an interface for Pushbutton Switches on DE2 Board, add the PIO by performing the following steps:

1. Select **Other** > **PIO** (Parallel IO) and click **Add.**
2. Specify the Options. See Figure 2-17.
   - Width =**4 bits**
   - Direction = **Input ports only**.

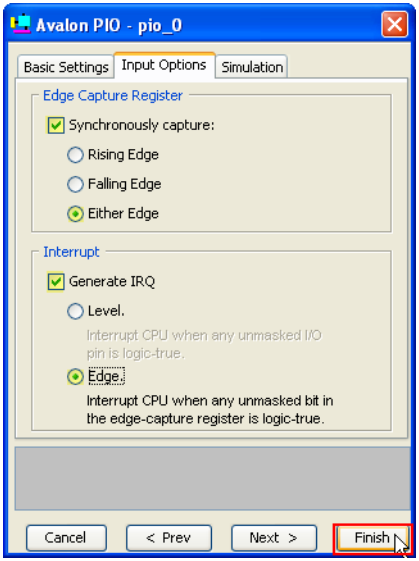
Figure 2-17.PIO Configuration Wizard For Pushbutton Switches-Basic Settings



3. Click **Input Options Tab**
   - Under Edge Capture Register, Check **Synchronously Capture**. Under Synchronously capture, select option **Either Edge**. See Figure 2-18.
   - Under Interrupt, check **Generate IRQ**. Under Generate IRQ, select **Edge option**. See Figure 2-18.
4. Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
5. Right click **pio_0** and select **Rename**.

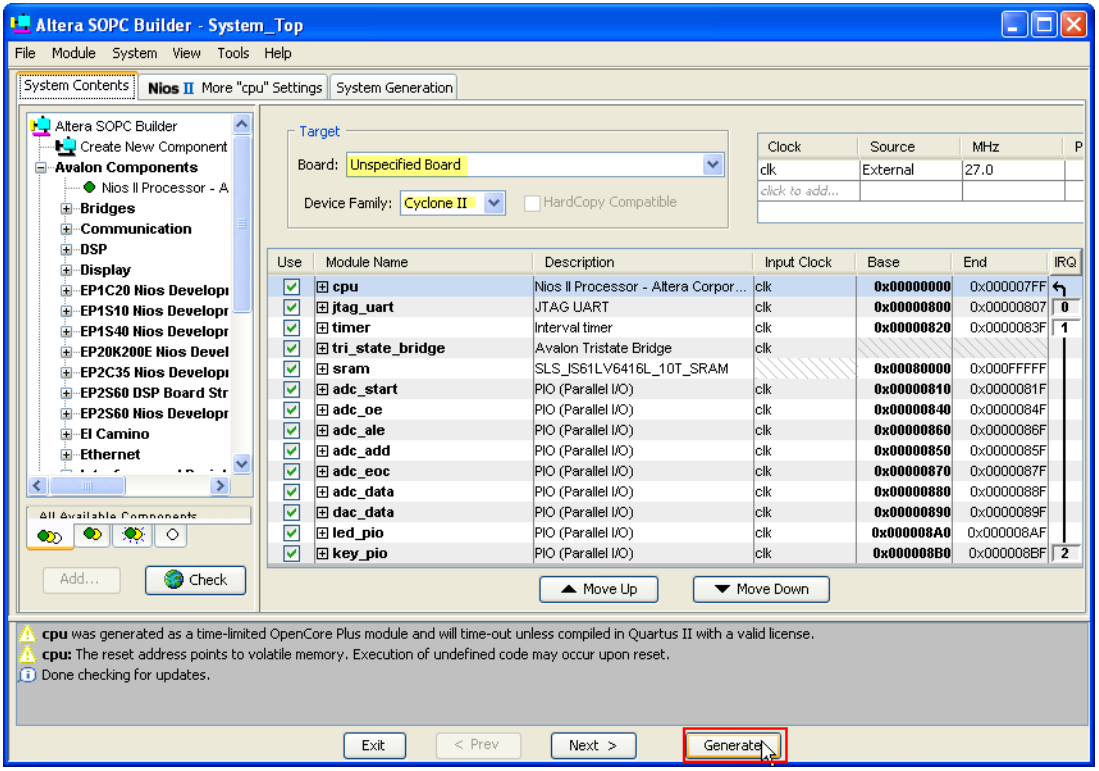
6. Type `key_pio` and press **Enter**.

Figure 2-18.PIO Configuration Wizard For Pushbutton Switches-Input Options Settings



After adding all components, the complete SOPC builder system looks like Figure 2-19.

Figure 2-19.SOPC Builder After Adding All Components

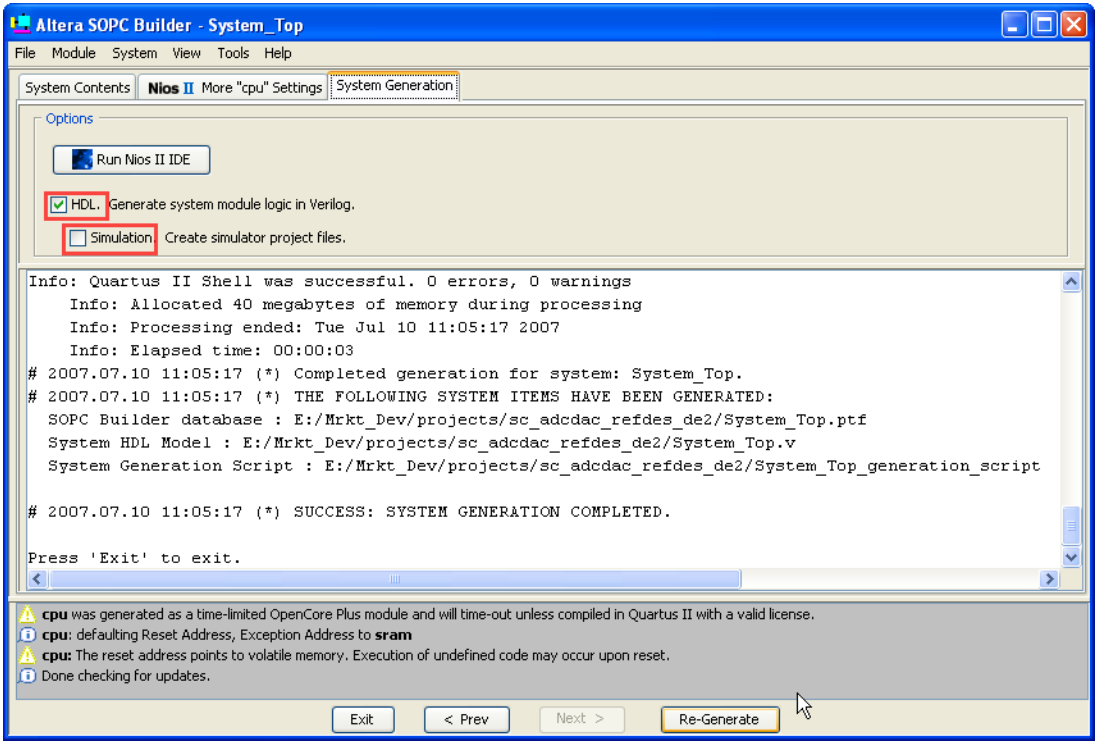


Generating the System

To generate the design logic, perform the following steps.

1. Click the **System Generation** tab.
2. Specify the following settings from the Options window. Figure 2-20.
   - HDL: **Check this box**
   - Simulation: **Check this box** if you have Modelsim installed and would like to simulate the design.

Figure 2-20.Generating the System



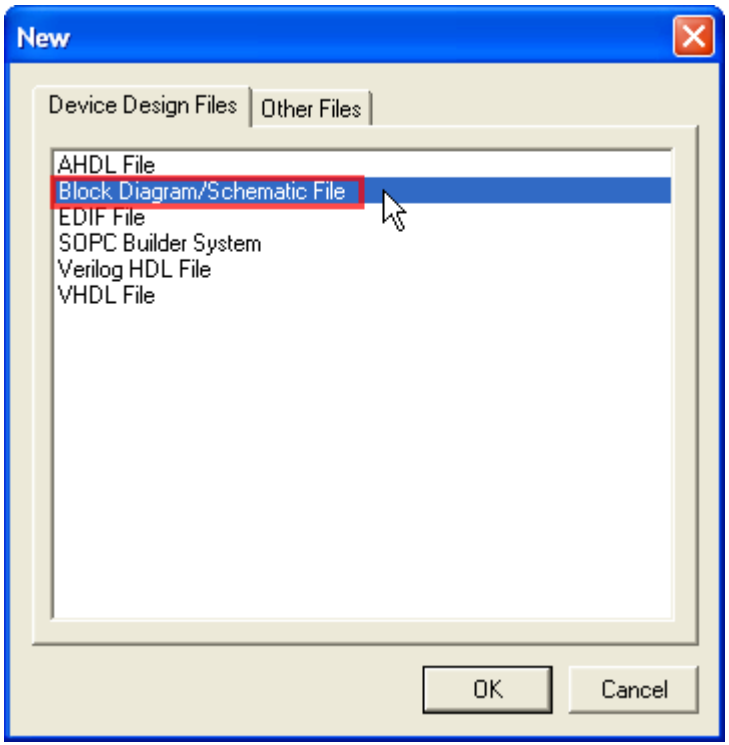
3. Click **Generate**. See Figure 2-20.
4. When generation is complete, the **SYSTEM GENERATION COMPLETED** message displays. “DO NOT EXIT SOPC BUILDER AT THIS POINT.” We will return to this window prior to testing the system with software.

Adding the Quartus II Symbol to the BDF

During generation, SOPC Builder creates a symbol of the **System_Top**, for using in Quartus II. To **add the symbol** perform the following steps:

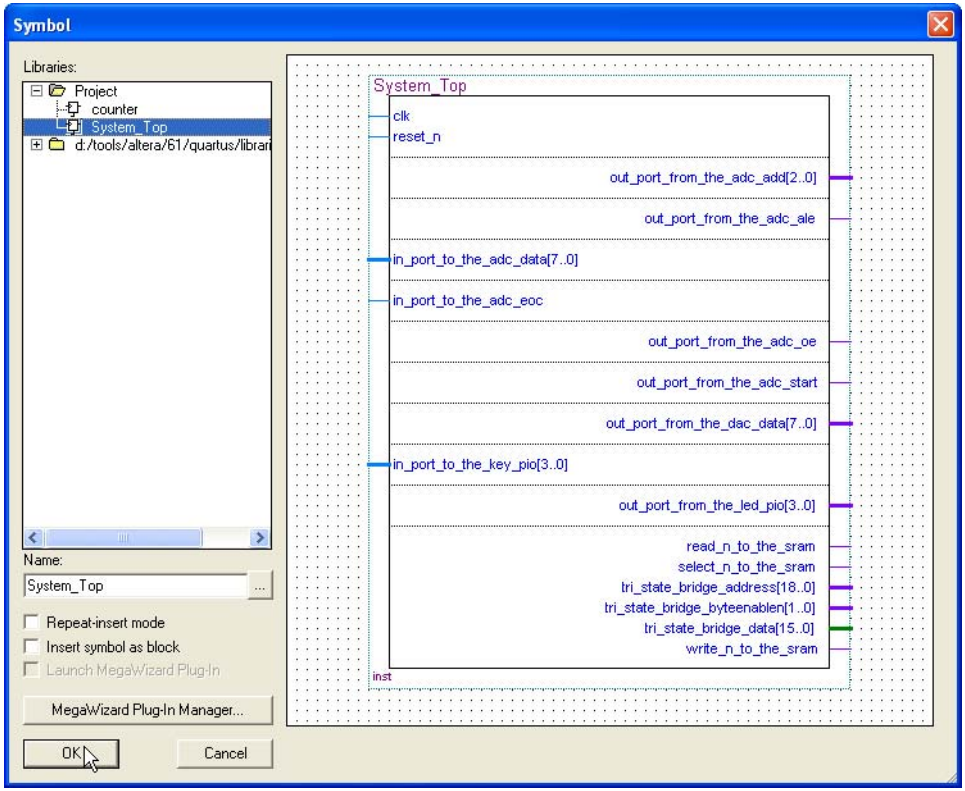
1. Select **File (menu) > New**.
2. Under Device Design Files, select **Block Diagram/Schematic File**. See Figure 2-21.

Figure 2-21.New Block Design File



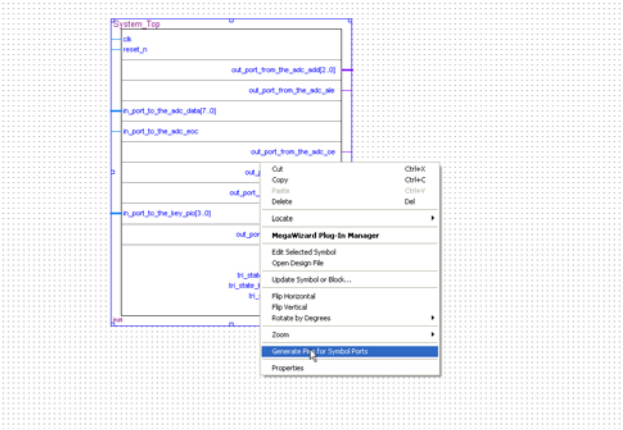
3. Click **OK**. You are return to the **Quartus II software** and double click anywhere inside the **BDF window**. The Symbol dialog box appears. See Figure 2-22.

Figure 2-22.BDF System_Top Symbol



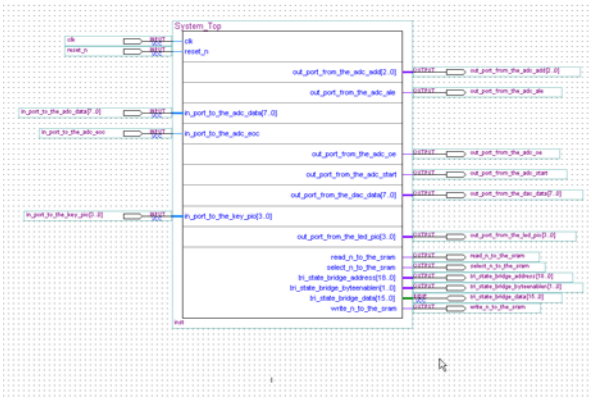
4. From Libraries, expand the **Project directory** by clicking the + sign nearby.
5. Click **System_Top**. A large symbol will appear representing the **ADCDAC system module** you just created.
6. Click **OK**. The Symbol dialog box closes and an outline of the **System_Top** symbol is attached to the pointer.
7. Place the **symbol** in the Block Diagram file by clicking the left mouse button.
8. You can generate the input and output node by selecting the symbol and right click on it and select **Generate Pins for Symbol Ports.** See Figure 2-23.

Figure 2-23.Generate Pins



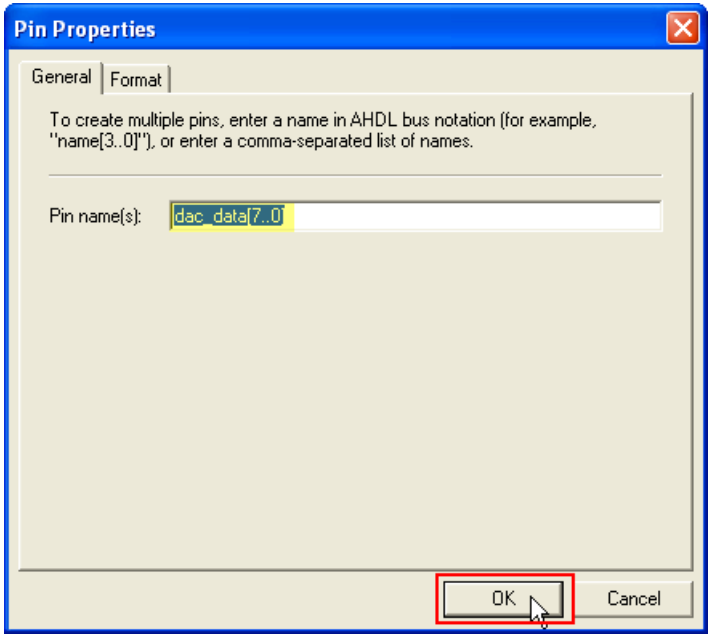
9. Input, output and bidirectional pins for the symbol will be automatically generated and connected accordingly. See Figure 2-24.

Figure 2-24.Autogenerated Pins for Symbol



10. The autogenerated pin names can be changed. Right click the **pin name** and select **Properties.** The Pin Property dialog box appearsas shown in Figure 2-25.

Figure 2-25. Pin Properties



11. You can give all the name according to Table 2-1 . The names are given for ease of reference.

Table 2-1. Pin Names in Final BDF

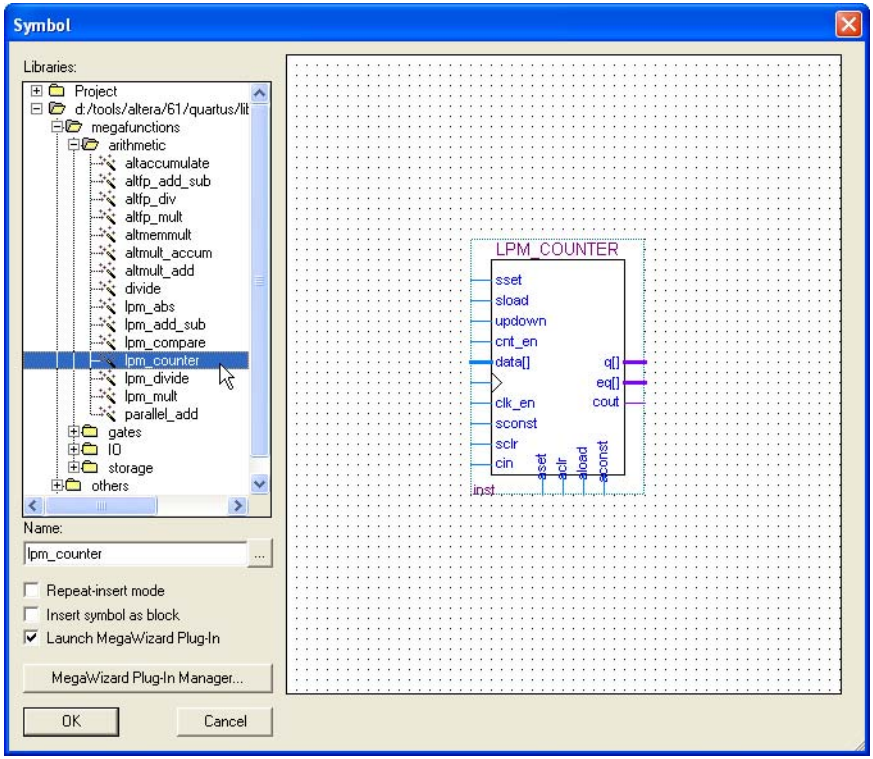
Pin Name of the Symbol	Custom Pin Names
clk	clk
reset_n	reset_n
in_port_to_the_adc_data[7..0]	adc_data[7..0]
in_port_to_the_adc_eoc	adc_eoc
in_port_to_the_key_pio[3..0]	key[3..0]
out_port_from_the_adc_add[2..0]	adc_add[2..0]
out_port_from_the_adc_ale	adc_ale
out_port_from_the_adc_oe	adc_oe
out_port_from_the_adc_start	adc_start
out_port_from_the_dac_data[7..0]	dac_data[7..0]

out_port_from_the_led_pio[3..0]	ledg[3..0]
read_n_to_the_sram	sram_oe_n
select_n_to_the_sram	sram_ce_n
tri_state_bridge_address[18..0]	sram_add[18..0]
tri_state_byteenablen[1..0]	sram_be_n[1..0]
tri_state_bridge_data[15..0]	sram_data[15..0]
write_n_to_the_sram	sram_we_n

12. Add the counter by following the steps below:

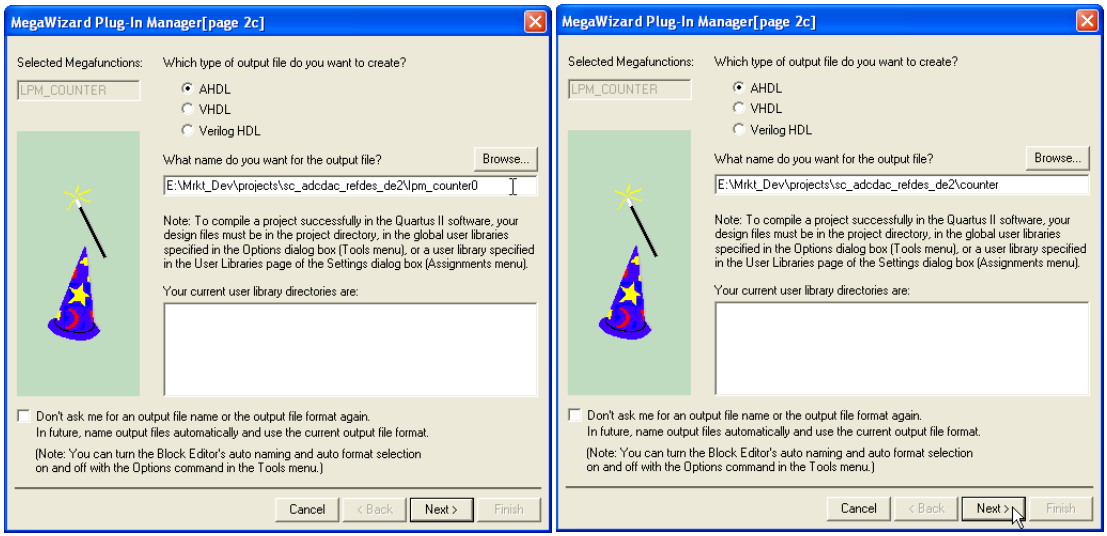
- Double click anywhere inside the **BDF** window.
- It will pops up the **Symbol** window.
- Select ***<dirve path>:/altera/../megafunctions/arithmatic/lpm_Counter***. See Figure 2-26. Click **OK**

Figure 2-26.Symbol Window



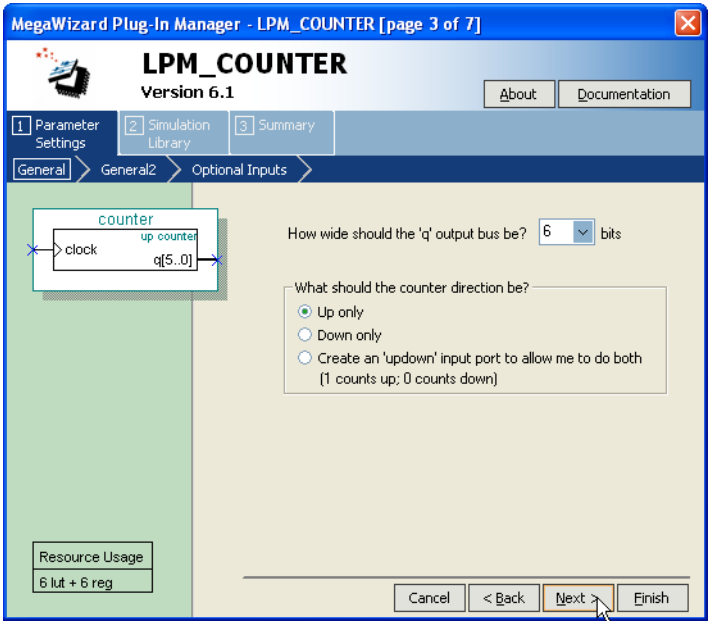
- It will pops up the MegaWizard Plug-In Manager window. See Figure 2-27.
- Give the name **counter** in the Output file text box. See Figure 2-27.

Figure 2-27.MegaWizard Plug-In Manager [page 2C]



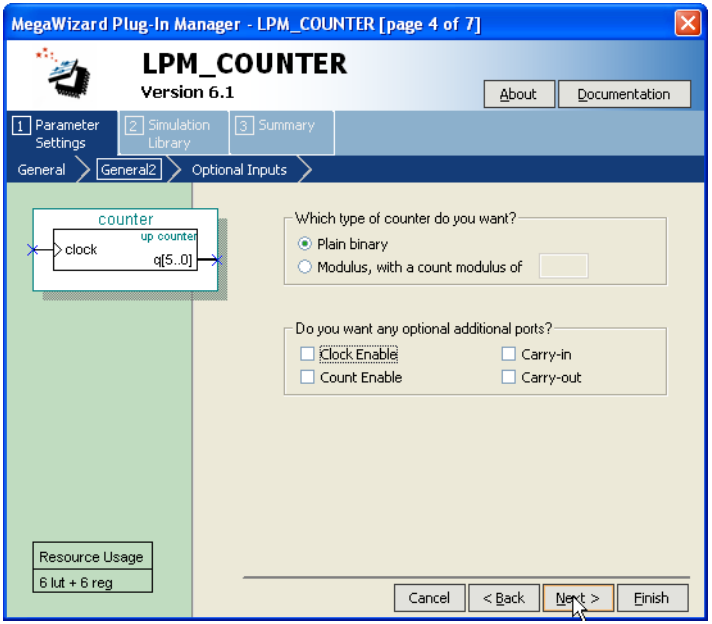
- Click **Next**. You will go to the next page of MegaWizard.
- Select the output bus width as **6** bits. See Figure 2-28.

Figure 2-28.MegaWizard Plug-In Manager [page 3]



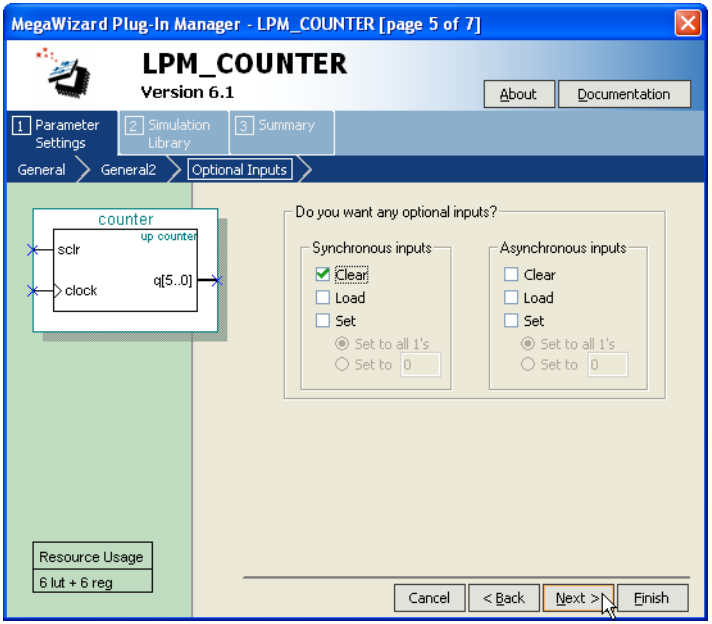
- Click **Next**. You will see the page 4 of MegaWizard.
- Leave the default settings as shown in Figure 2-29.

Figure 2-29.MegaWizard Plug-In Manager [page 4]



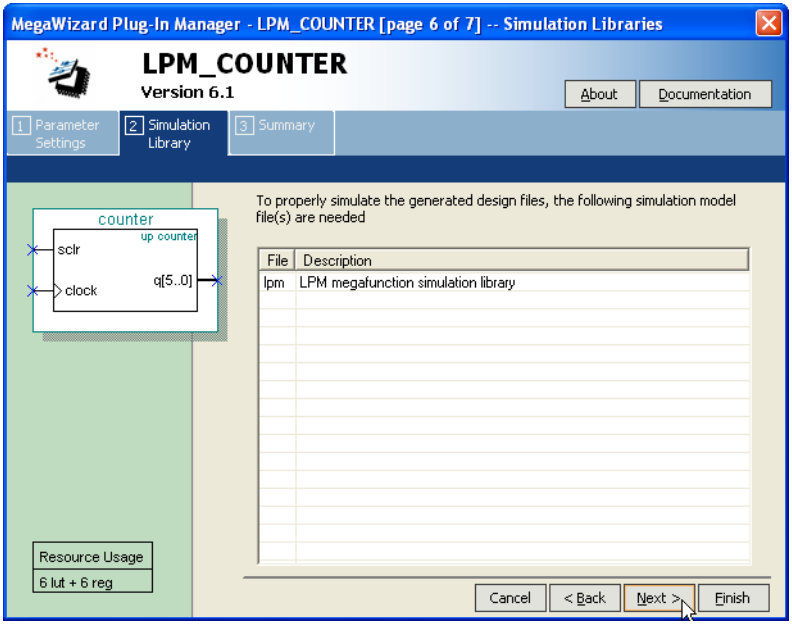
- Click **Next**.

Figure 2-30.MegaWizard Plug-In Manager [page 5]



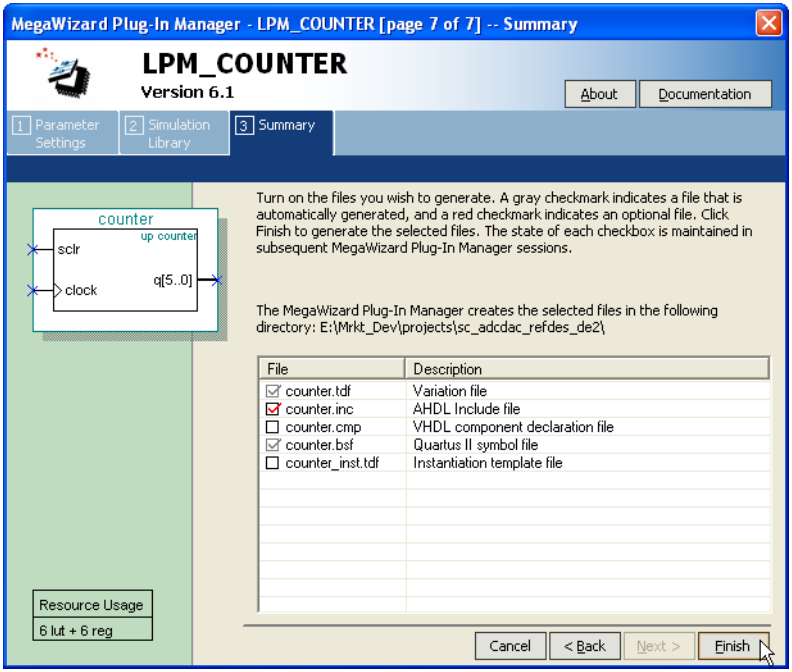
- Under Synchronous inputs, Check the **Clear box**. See Figure 2-30.
- Click **Next**.
- You will see page 6 of MegaWizard. SeeFigure 2-31.

Figure 2-31.MegaWizard Plug-In Manager [page 6]



- Click **Next**.
- You will see page 7 of MegaWizard. See Figure 2-32.

Figure 2-32.MegaWizard Plug-In Manager [page 7]

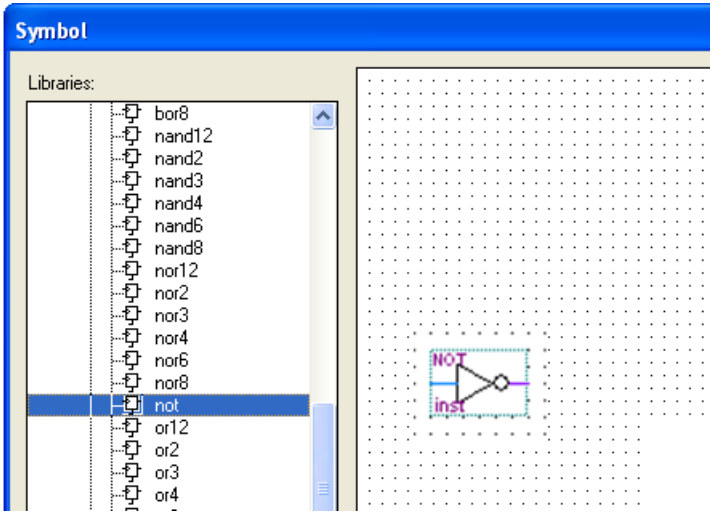


- Click **Finish**.
- You will return to BDF window
- Place the **Counter symbol** in the Block Diagram file by clicking the left mouse button

13. Now the **clk** input of counter is directly connected to the **clk** input of the System_Top. The **sclr** input of the counter is connected to the **reset** input of the System_Top using **NOT gate**.

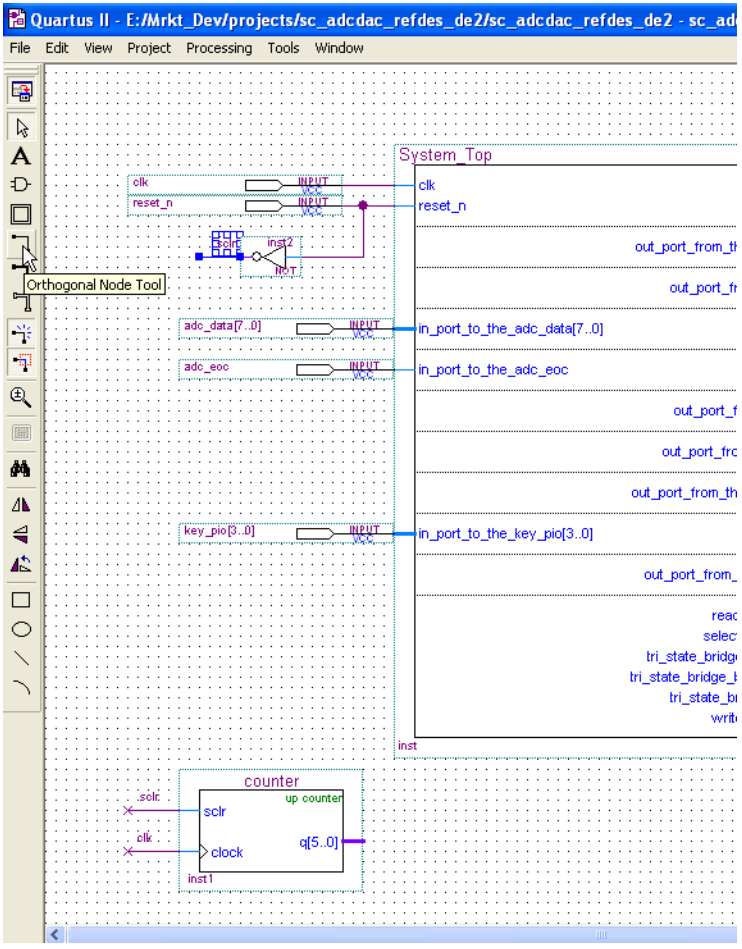
14. For selection of the NOT gate double click inside the block diagram and select ***<drive path>:/.../library/primitives/logic/not***. See Figure 2-33.

Figure 2-33. NOT Gate Selection



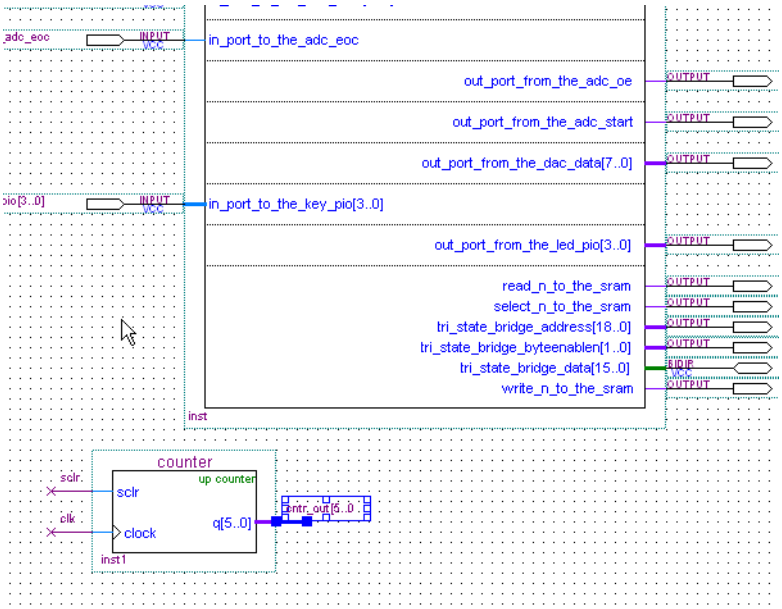
15. Then by using orthogonal node tool, you can connect the input of NOT gate to the reset input of the System_Top and the output of the NOT gate to the sclr input of the counter. See Figure 2-34.

Figure 2-34. NOT Gate Connection



16. Now you have to expand the line as shown in Figure 2-35. Give the name **cntr_out[5..0]**.

Figure 2-35.Counter Output

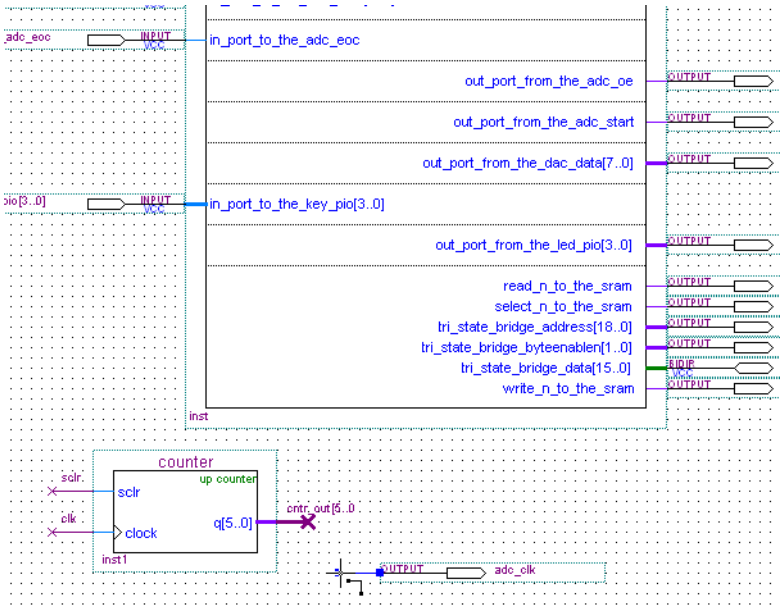


17. Now for the ADC clock output, add output symbol by selecting it from the library ***<drive path>:/.../library/primitives/pin/output.***

18. Give the output pin name as **adc_clk**.

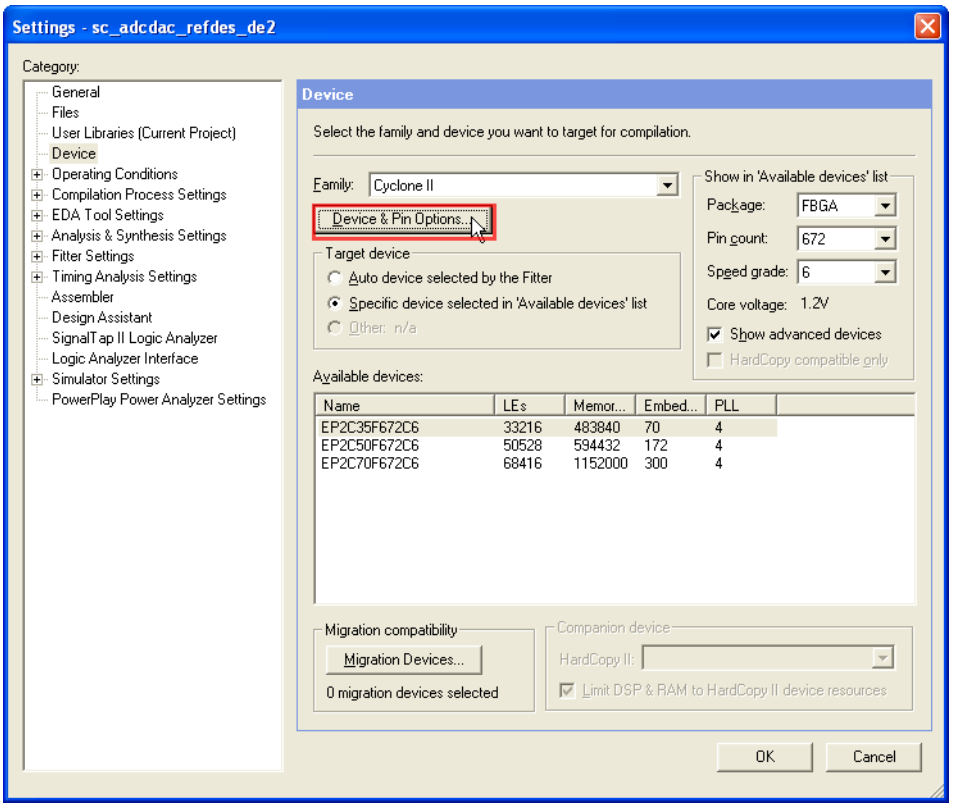
19. Do not connect it with the output of the counter strech its line as shown in Figure 2-36. Give the name as **cntr_out[5]** and press **Enter**.

Figure 2-36.Counter Output Connection



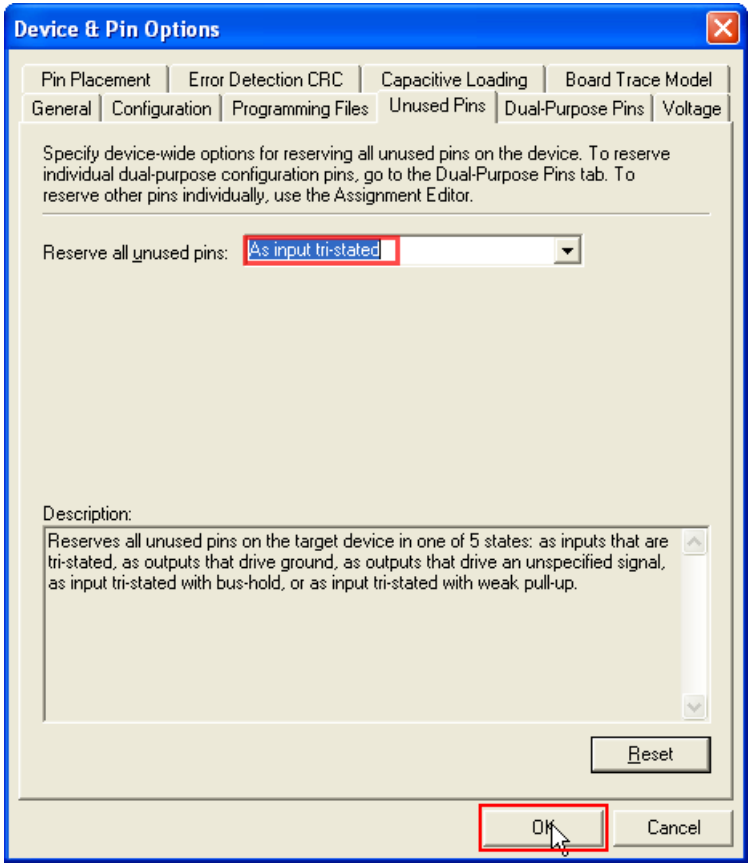
20. After assignment of the IO pins, select **Assignment>Settings.** The Settings Dialog Box pops up as shown in Figure 2-37.
21. In Device dialog box click the button **Device & Pin Options.**
22. Device and Pin options dialog box pops up as shown in Figure 2-38.
23. Click the **Unused pins** Tab. Select **As Input tri-stated** in Reserve all unused pins combo box.
24. Click **OK.** You will return to **Device Settings** window.
25. Click **OK.**
26. Now select the **Processing** from the menu bar and click on **Start**.and select the **Start Analysis And Synthesis.**
27. For assignment open the **Assignments** from menu bar and select the **Assignment Editor**. Select the category as pin and then you can see the list of all pins used in the project. If you do not see the pins then go to **View** menu and select **Show All Known Pin Names**.

Figure 2-37.Settings



28. Now you have to go to the **sc_adcdac_refdes_de2.csv** file at ***SLS_ADC-DAC_Board/ADC-DAC_Reference_Designs*** **folder** and select the pins and copy it. Open the Quartus II software and paste it in the assignment editor.

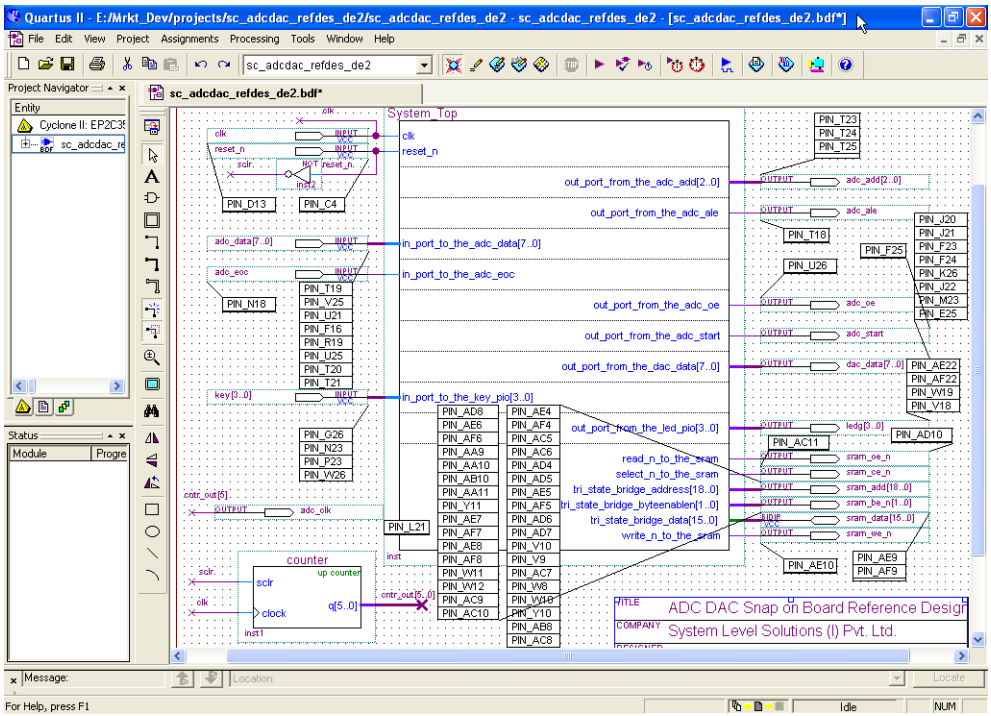
Figure 2-38. Device & Pin Options



29. After adding Inputs/Output, **Inouts and pin assignments** the final BDF looks like Figure 2-39. Please refer the provided Reference Design.

30. Choose **File>Save**.

Figure 2-39.Final BDF



Note: to Figure 2-39.

(1) In this tutorial we are using only one memory (SRAM). In order to disable SDRAM and FLASH memories, the pins are at Pull up status.

(2) You can assign the pin number using the ***SLS_ADC-DAC_Board/ADC-DAC_Reference_Designs/ sc_adcdac_refdes_de2.csv*** file.

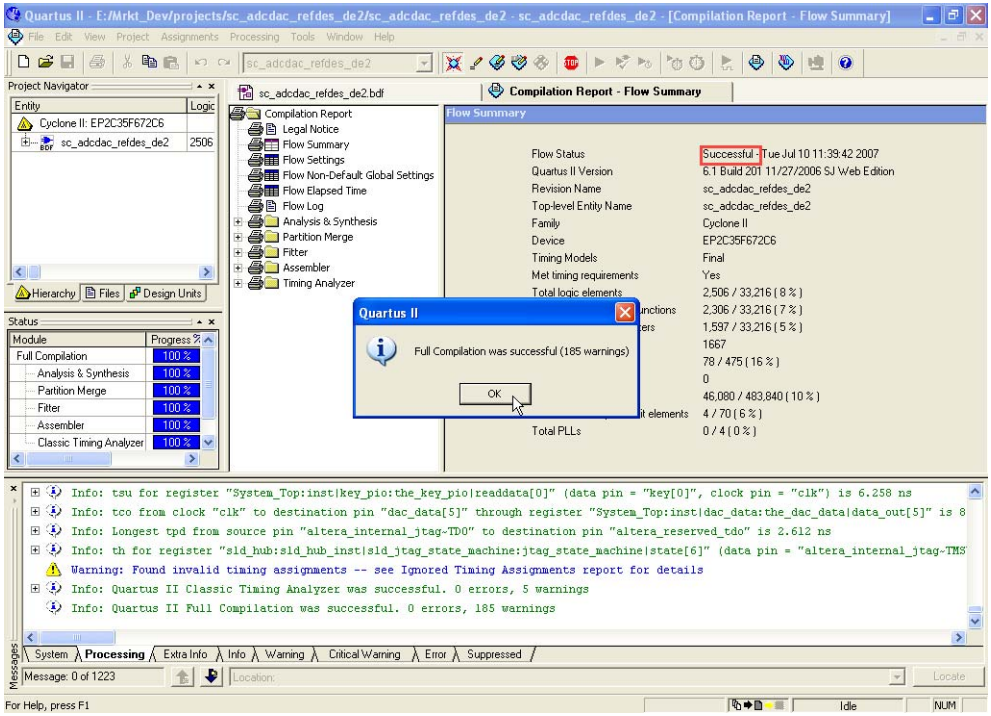
Compiling the Design

During compilation, the Compiler locates and processes all design, project files, generates messages and reports related to the current compilation, creates the **SRAM object file** (.sof) as well as any **optional programming files**.

To compile the **sc_adcdac_refdes_de2** design, follow these steps:

1. Choose **Start Compilation (Processing menu)**, or click the **Start Compilation toolbar button**. If you get a message asking if you want to save the changes you made the BDF file, choose **Yes**.
2. When compilation completes, you can view the results in the **ADC-DAC Compilation Report window**. See Figure 2-40.

Figure 2-40.Compilation Report Window



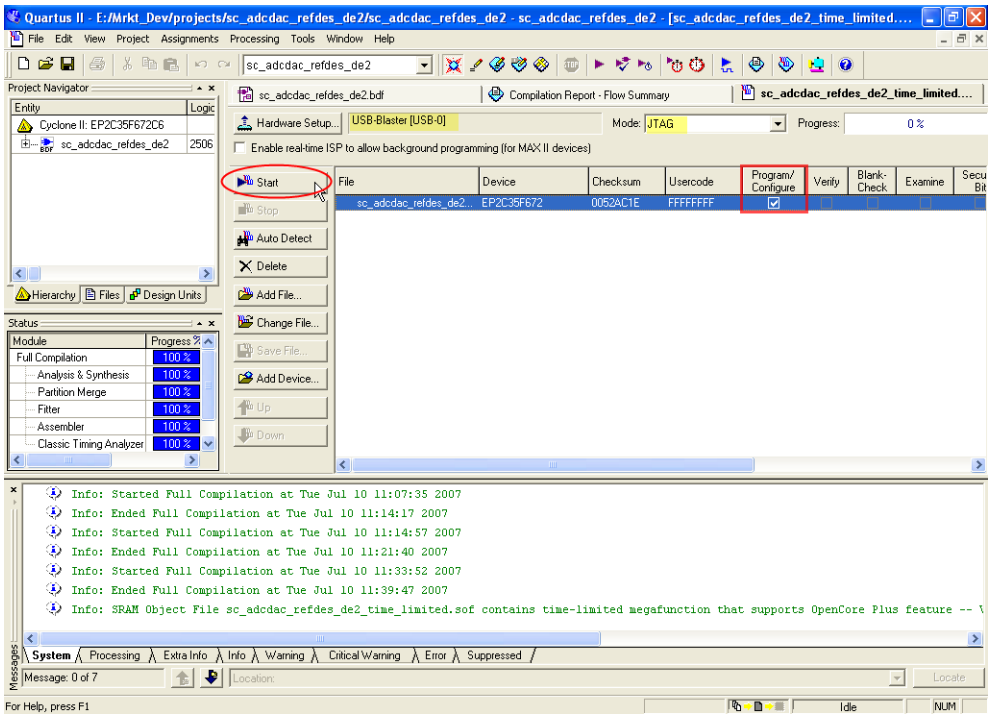
3. Programming

After successful compilation, the Quartus II Compiler generates one or more programming files that the Programmer uses to program or configure a device.

To program your design, perform the following steps.

1. Choose **Programmer (Tools menu)**. The Programmer window opens.
2. In the Mode list of the programmer window, make sure **JTAG** is selected.
3. Click **Hardware Setup..** to configure the programming hardware. The Hardware Setup dialog box appears.
4. From the Hardware column, select **USB Blaster .**
5. Click **Close** to exit the Hardware Setup window.
6. In the Programmer window, turn on **Program/Configure**. See Figure 3-1.
7. Click **Start**.

Figure 3-1. Programming Window



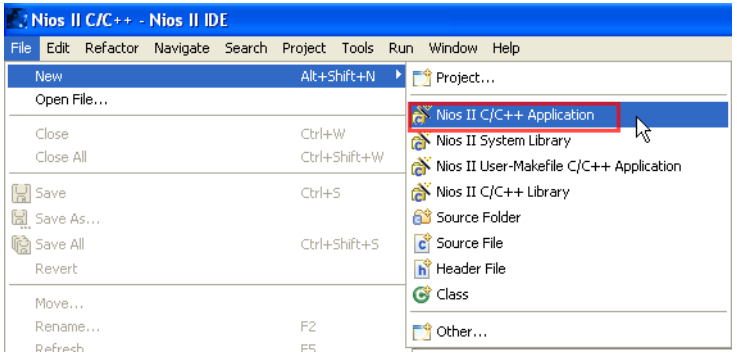
4. Running the Software in Nios II IDE

Creating the Project

We will be using Nios II Integrated Development Environment (IDE) to run our software on top of Nios II System. To start the Nios II IDE from the Quartus II software, perform the following steps:

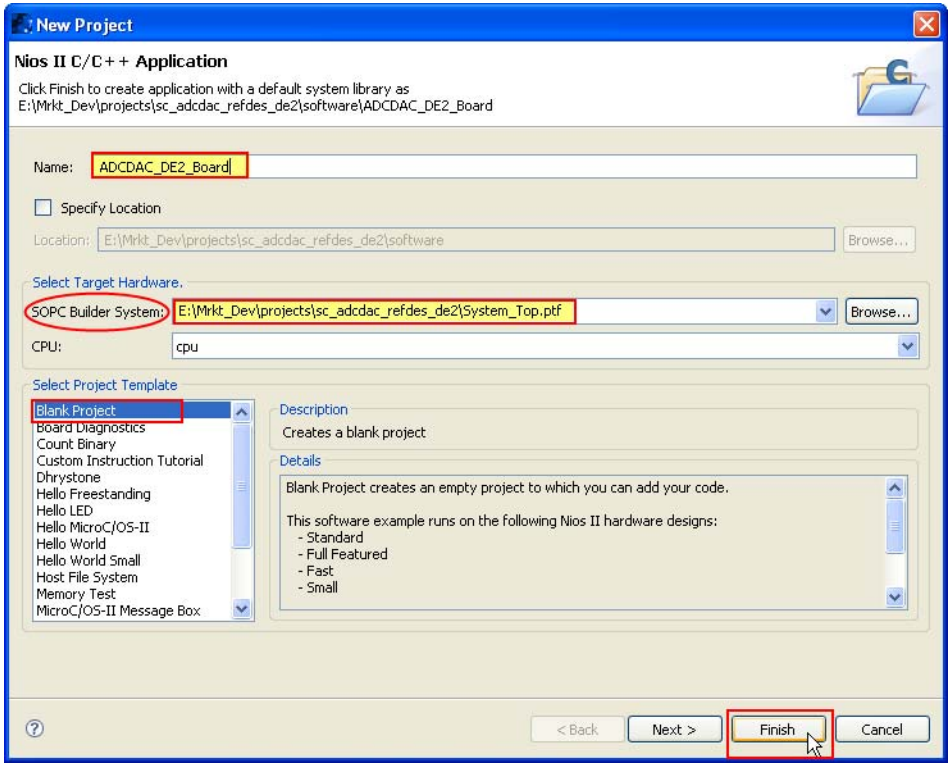
1. Click the **System Generation** tab of the SOPC Builder, then click the **Run Nios II IDE** button**.**
2. From the opening window, choose **File>New>Nios II C/C++ Application**. See Figure 4-1.
3. You will see the New Project Wizard Dialog Box as shown in Figure 4-2.

Figure 4-1. New Project



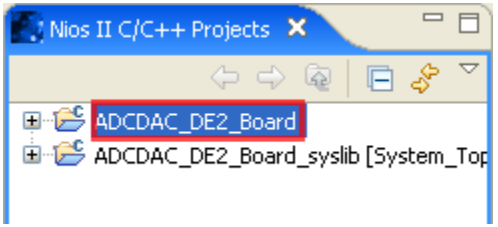
4. The user has been asked to select Blank Project, since SLS has provided a reference C file to access the ADC-DAC on DE2. To import this C file into this project do the following steps.
   - Select the **“Blank Project”**
   - Give a **Name** to the project as **ADCDAC_DE2_Board**.
   - Select the SOPC builder system by clicking on **Browse** and selecting the **System_Top.ptf** file in the reference design. See Figure 4-2.

Figure 4-2. Creating new Nios II C/C++ Application



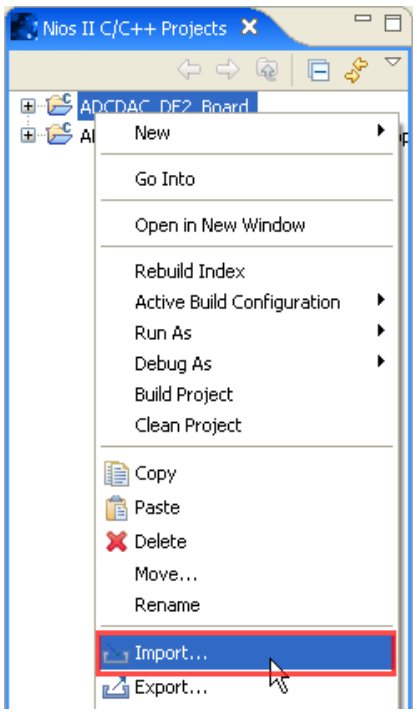
5. Click **Next**
6. Click **Finish.**
7. You will observe your project created under the **Nios C/C++ Projects Tab.** See Figure 4-3.

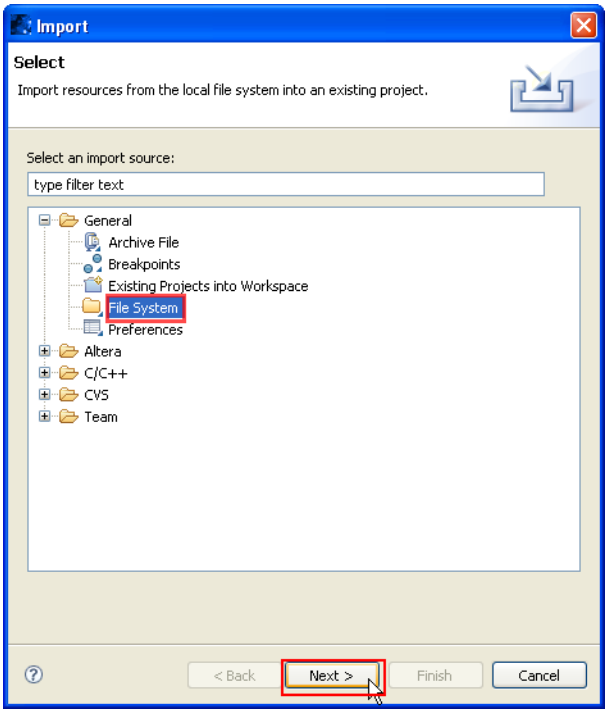
Figure 4-3. Listing of Your Project under Nios II C/C++ Project Tab



8. Select the **Project (ADCDAC_DE2_Board).** Right Click and hit **Import.** See Figure 4-4.

Figure 4-4. Importing File System

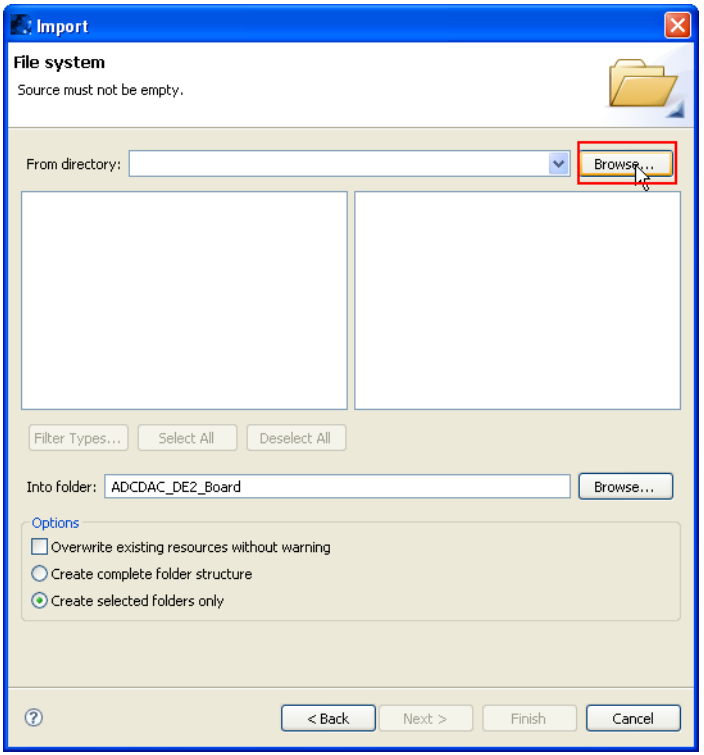




9. You will now be asked for the details of the file to be imported.
   - You have to select the file system as shown in Figure 4-4.

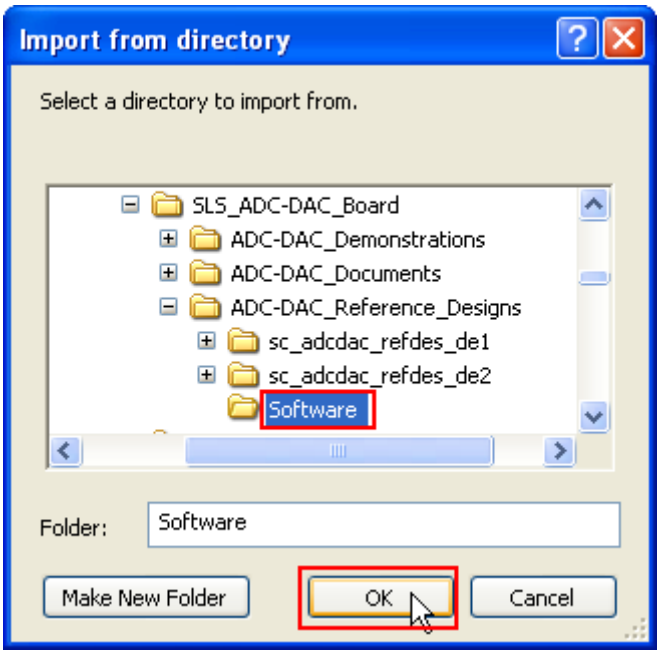
- Click **Next.** You will see the import dialog box as shown in Figure 4-5.

Figure 4-5. Import Dialog Box



- Browse to ***SLS_ADC-DAC_BoardADCDAC_Reference_Designs/software.*** See Figure 4-6.
- Click **OK.** You will return back to Import Wizard.

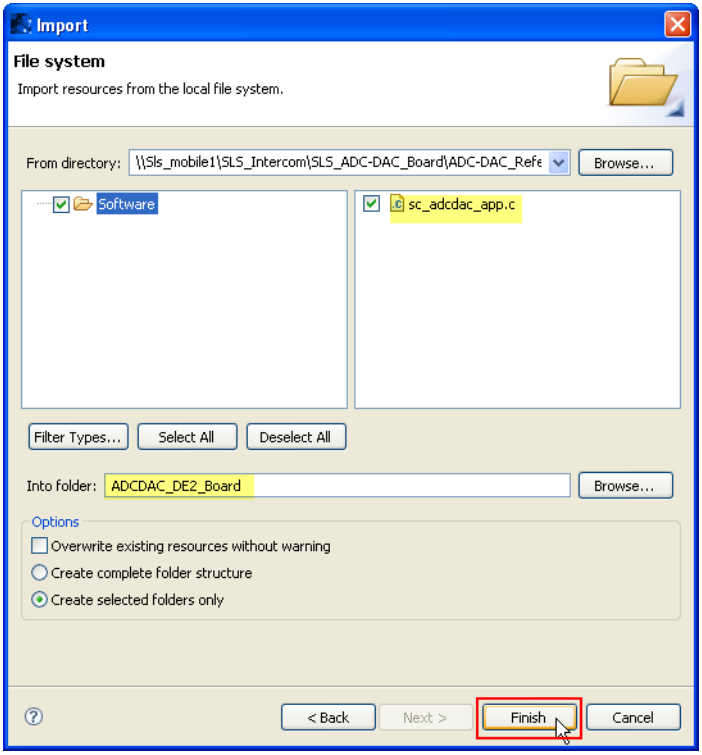
Figure 4-6. Asking for Details of the files to be imported



10. Select the **sc_adcdac_app.c** file in the template by checking the box. See Figure 4-7.

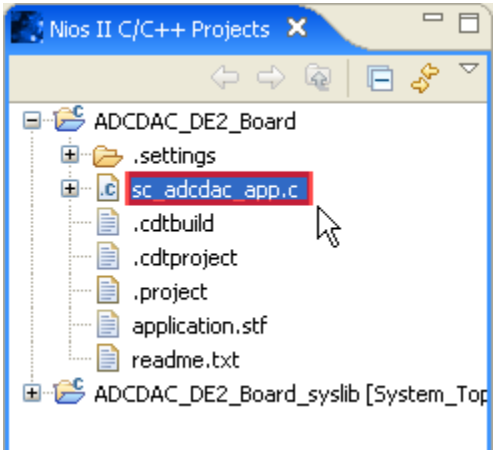
11. Click **Finish.**

Figure 4-7. Asking for Details of the files to be imported



12. Observe the inserted **sc_adcdac_app.c** file template in the application list. See Figure 4-8.

Figure 4-8. Inserted C file template in the application list



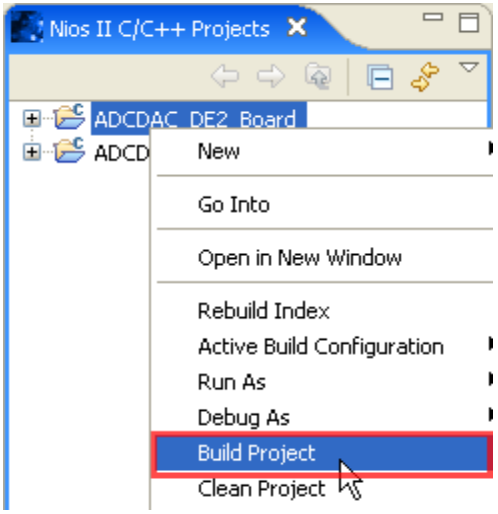
You can browse through the contents of the file by double clicking the file Name in the list.

Building the Project

After successful creating yourproject, its now time to build your project. To build your project follow steps below:

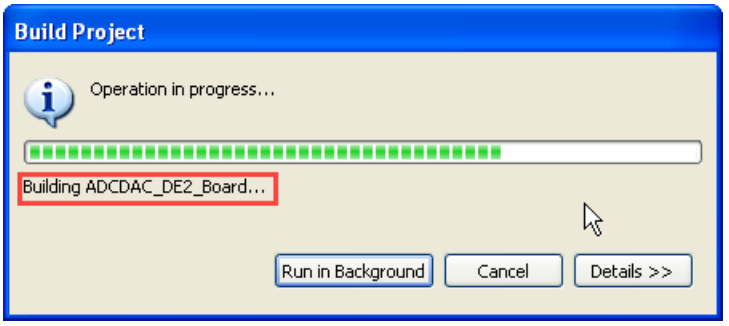
1. Right click the **Name of the Project** and select **"Build Project"** to build your project. See Figure 4-9.

Figure 4-9. Build Project



2. You will observe the window. See Figure 4-10.

Figure 4-10. Build Process

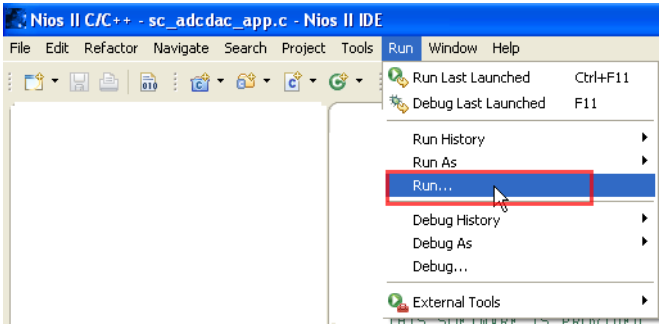


Running the Project

After building the project successfully, its now time to run the project on the hardware. There are two ways to run your project.

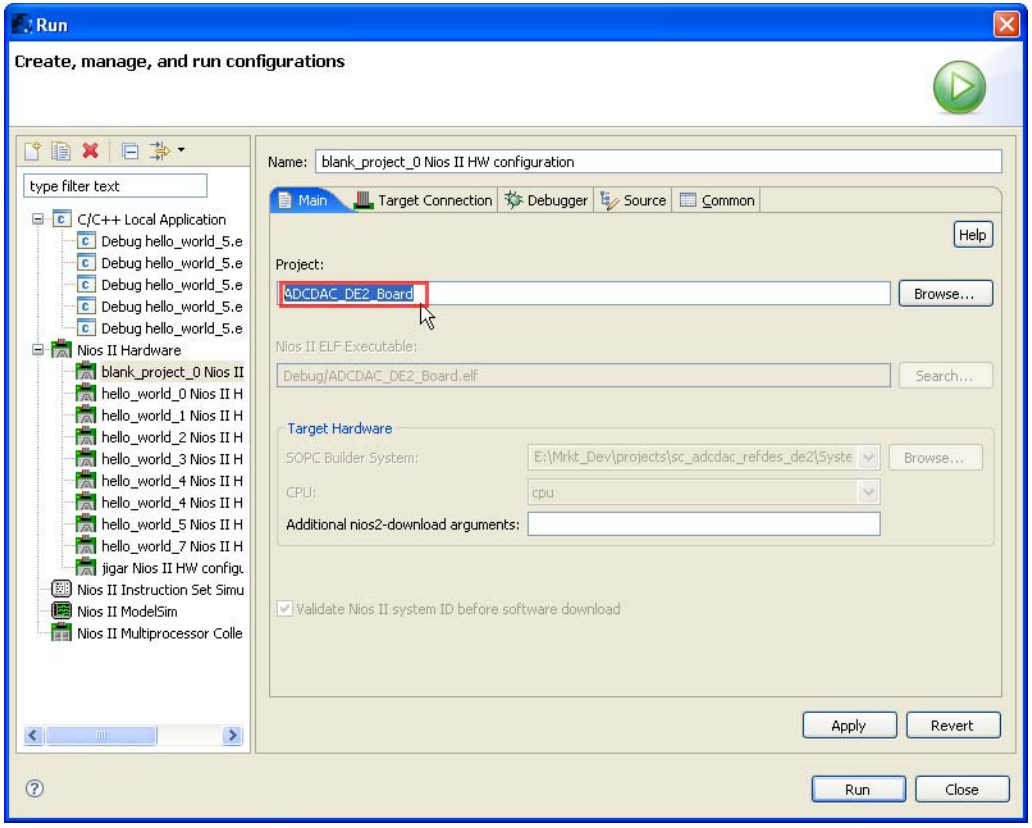
1. Select **Run** > **Run..** as shown in Figure 4-11.

Figure 4-11. Run Option



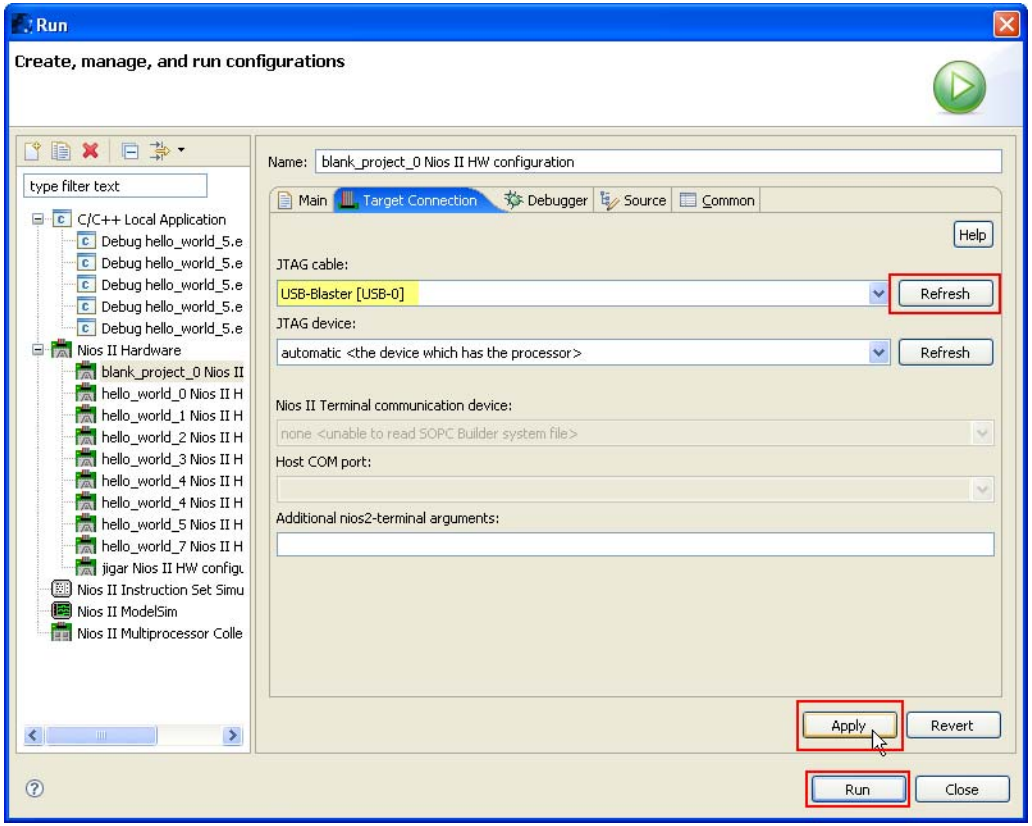
2. It will pops up Run Dialog Box as shown in Figure 4-12.

Figure 4-12.Run Dialog Box



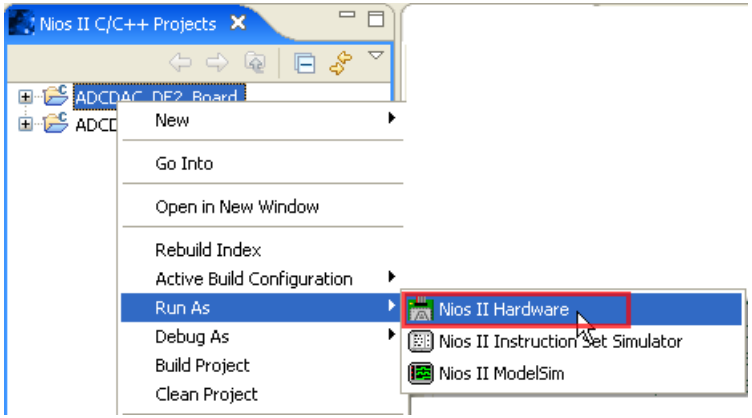
3. Under Project, select the **Nios II C/C++ Porject** by browsing your ADC-DAC Project.
4. Click on **Target Connection Tab**. You will see the Figure 4-13.

Figure 4-13.Run Dialog- Target Connection Settings



5. Select the target Hardware **USB-Blaster** [USB-0] under JTAG Cable. Click **Refresh.**
6. Click **Apply.**
7. Click **Run.**
8. Alternatively you can run the project by Right click the **Project** and select **Run As > Nios II Hardware.** Figure 4-14.

Figure 4-14. Run As-Nios II Hardware Selection

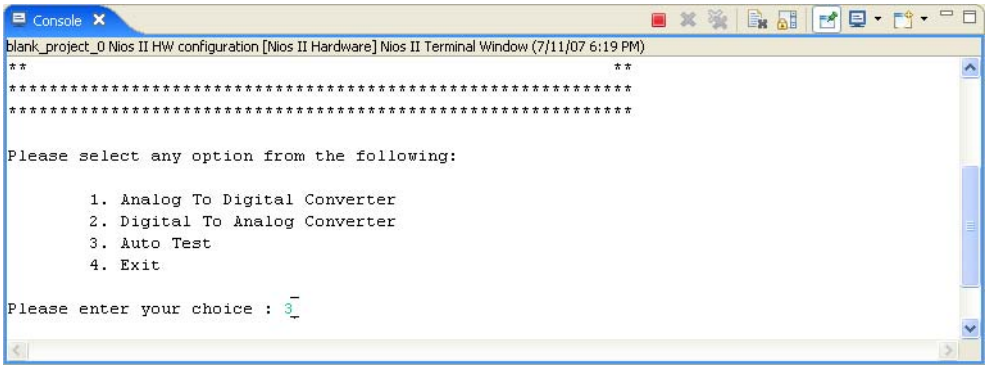


9. You will observe the console window as shown in Figure 4-15.

10. The Console Window asks for the selection of choice from the following options:

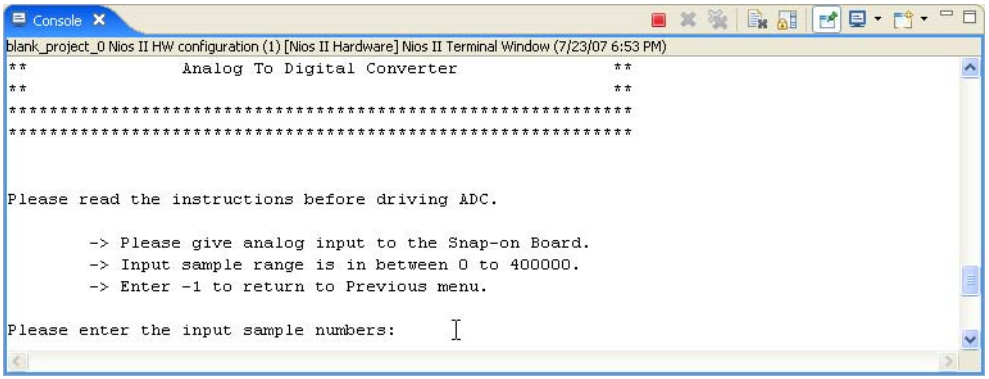
- 1. Analog To Digital Converter
- 2. Digital To Analog Converter
- 3. Auto Test
- 4. Exit
- Please enter your choice :

Figure 4-15.IDE Console Output



11. If you select **Option 1** and press enter, following message will be displayed. See Figure 4-16.

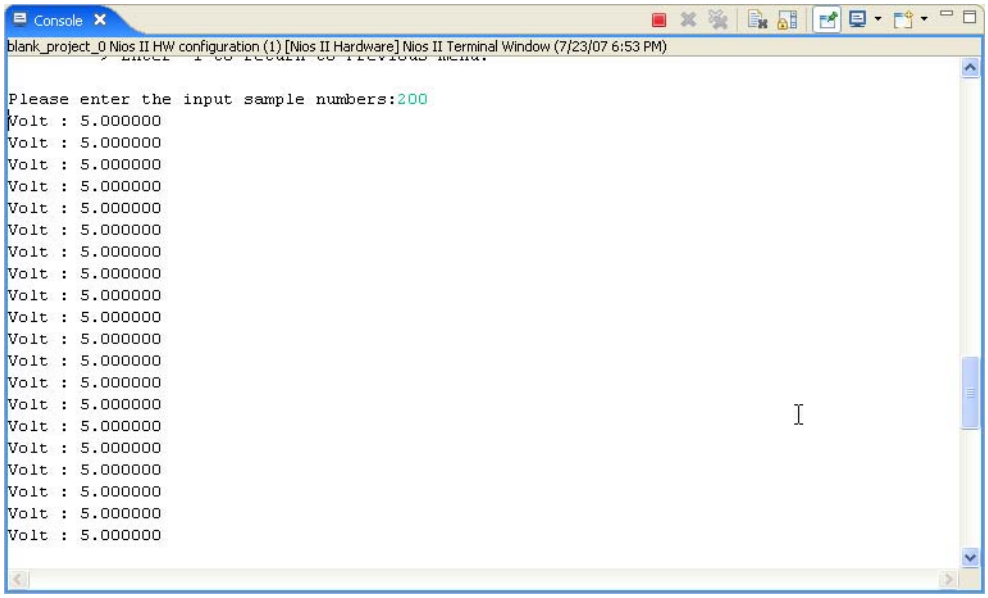
Figure 4-16.Analog to Digital Selection Console Window



- Give the analog input at pin IN0 and connect GND_ADC to ground.
- Enter analog input sample number and press **Enter**. You shall see the sample values displayed.
- Following figure shows the output window for 5V DC analog input at IN0 and 200 samples as analog input sample number. See Figure 4-17.

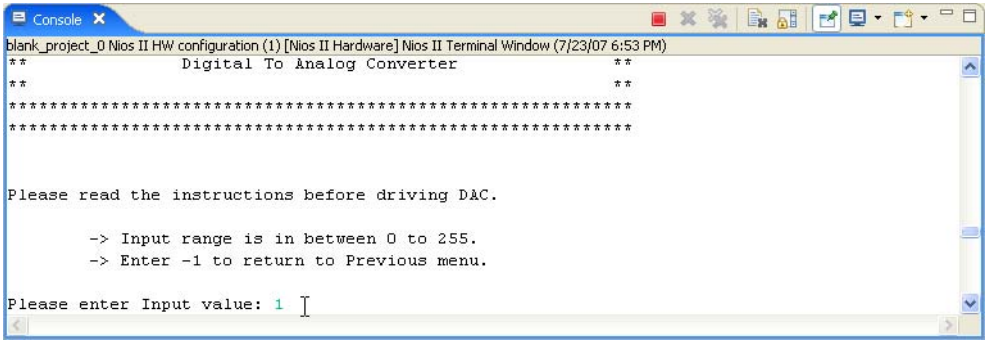
Here each reading is displayed after 10 samples. It shows the average value of 10 sample. Therefore only 20 readings are displayed here.

Figure 4-17.Analog to Digital Output Console Window



12. If you select **Option 2** and press enter, it will display the following message. See Figure 4-18.

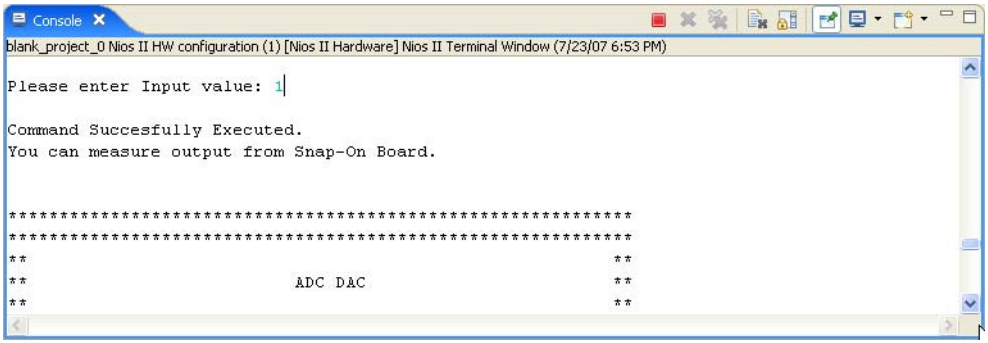
Figure 4-18.Digital to Analog Selection Console Window



- Give the digital input signal range between 0 to 255 and press **Enter**

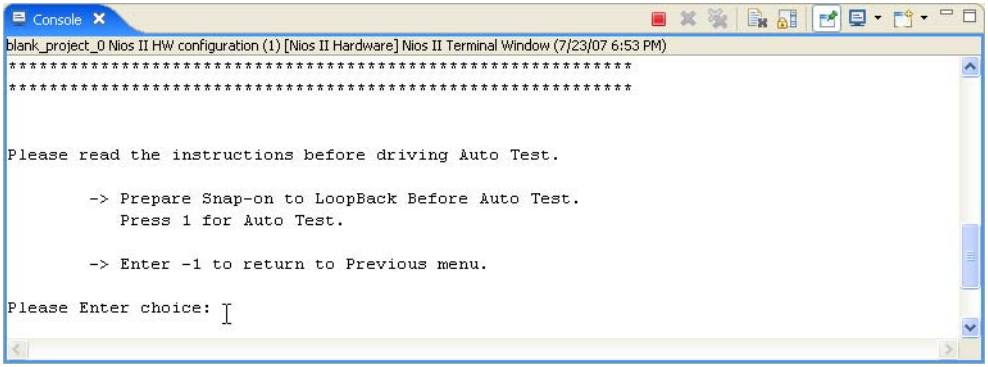
- The analog output can be measured at IOUT_n pin and you can see the display as shown in Figure 4-19.

Figure 4-19.Digital to Analog Output Console Window



13. If you select **Option 3** and press enter, following message will be displayed. See Figure 4-20.

Figure 4-20.Auto Test Selection Console Window

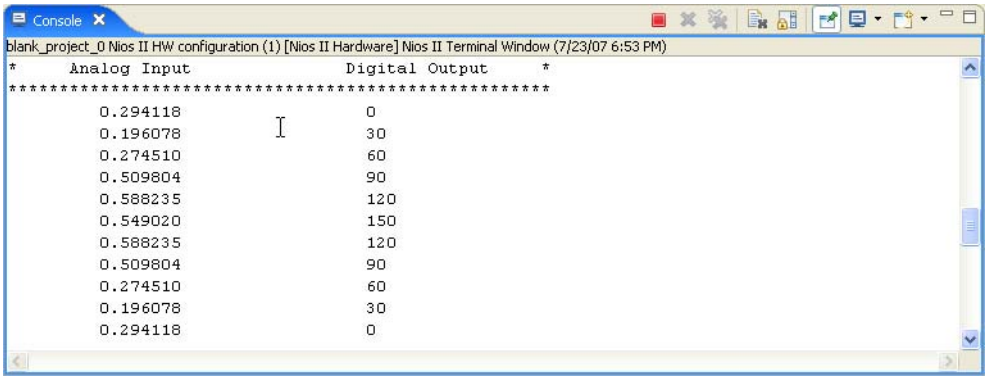


This is an **AUTO TEST** that requires a loop back connection. Therefore, make the connections as mentioned below.

- Connect IOUT_n i.e. output of the DAC to the IN0 i.e. input of the ADC. forming a loop.
- Connect the GND_ADC to ground

- After successful connection, you will be able to see the following window. See Figure 4-21.

Figure 4-21.Auto Test Output Console Window



14. If you select **Option 4**, it will exit form the procedure of conversion.

For details on the hardware for ADC-DAC Board refer to the **ADC-DAC Board Reference Manual.**