One Technology Way · P.O. Box 9106 · Norwood, MA 02062-9106 · Tel: 781.329.4700 · Fax: 781.461.3113 · www.analog.com

BeMicro FPGA Project for AD7367 with Nios driver

Supported Devices

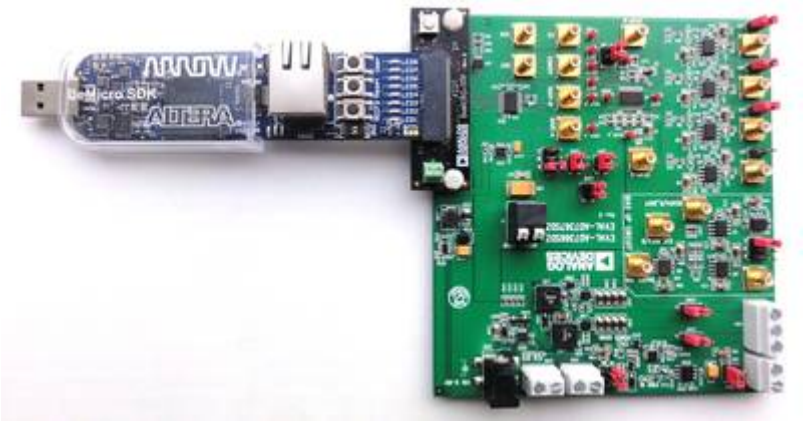
- AD7367

Evaluation Boards

- EVAL-AD7367SDZ

Overview

This lab presents the steps to setup an environment for using the **EVAL-AD7367SDZ** evaluation board together with the **BeMicro SDK** USB stick and the Nios II Embedded Development Suite (EDS). Below is presented a picture of the EVAL-AD7367SDZ Evaluation Board with the BeMicro SDK Platform.



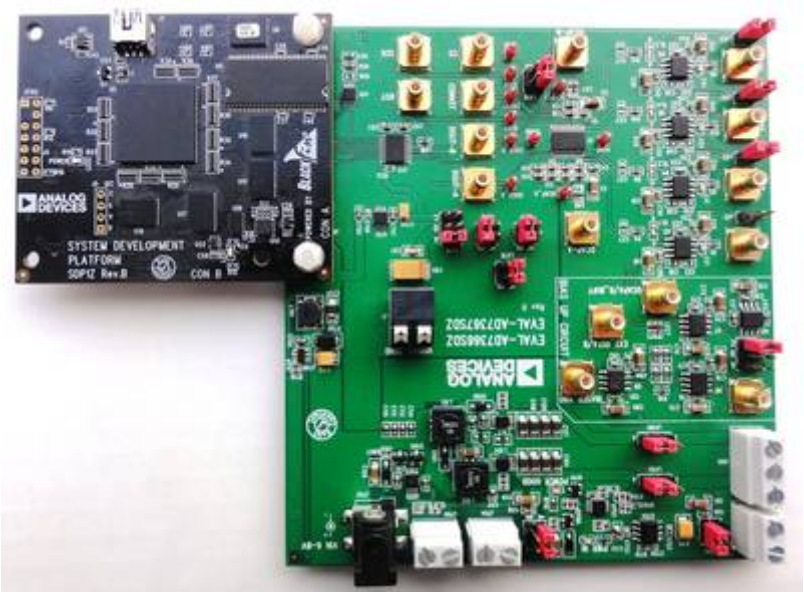
For component evaluation and performance purposes, as opposed to quick prototyping, the user is directed to use the part evaluation setup. This consists of:

- 1. A controller board like the SDP-B (EVAL-SDP-CS1Z)
- 2. The component SDP compatible product evaluation board
- 3. Corresponding PC software (shipped with the product evaluation board)

The SDP-B controller board is part of Analog Devices System Demonstration Platform (SDP). It provides a high speed USB 2.0 connection from the PC to the component evaluation board. The PC runs the evaluation software. Each evaluation board, which is an SDP compatible daughter board, includes the necessary installation file required for performance testing.

Note: it is expected that the analog performance on the two platforms may differ.

Below is presented a picture of **SDP-B** Controller Board with the **EVAL-AD7367SDZ** Evaluation Board.



The **EVAL-AD7367SDZ** evaluation board is a member of a growing number of boards available for the **SDP**. It was designed to help customers evaluate performance or quickly prototype new **AD7367** circuits and reduce design time.

The **AD7367** is a dual 14-bit, high speed, low power, successive approximation analog-to-digital converter that feature throughput rates up to 1 MSPS. The device contains two ADCs, each preceded by a 2-channel multiplexer, and a low noise, wide bandwidth track-and-hold amplifier. The AD7367 is fabricated on the Analog Devices, Inc., industrial CMOS process (iCMOS), which is a technology platform combining the advantages of low and high voltage CMOS. The iCMOS process allows the AD7367 to accept high voltage bipolar signals in addition to reducing power consumption and package size. The AD7367 can accept true bipolar analog input signals in the ±10 V range, ±5 V range, and 0 V to 10 V range.

More information

- AD7367 Product Info - pricing, samples, datasheet
- EVAL-AD7367SDZ evaluation board user guide
- BeMicro SDK
- Nios II Embedded Development Suite (EDS)

Getting Started

The first objective is to ensure that you have all of the items needed and to install the software tools so that you are ready to create and run the evaluation project.

Hardware Items

Below is presented the list of required hardware items:

- Arrow Electronics BeMicro SDK FPGA-based MCU Evaluation Board
- BeMicro SDK/SDP Interposer adapter board
- EVAL-AD7367SDZ evaluation board
- Intel Pentium III or compatible Windows PC, running at 866MHz or faster, with a minimum of 512MB of system memory

Software Tools

Below is presented the list of required software tools:

- Quartus II Web Edition design software v12.0sp2
- Nios II EDS v12.0sp2

The **Quartus II** design software and the **Nios II EDS** is available via the Altera Complete Design Suite DVD or by downloading from the web.

Downloads

- Lab Design Files

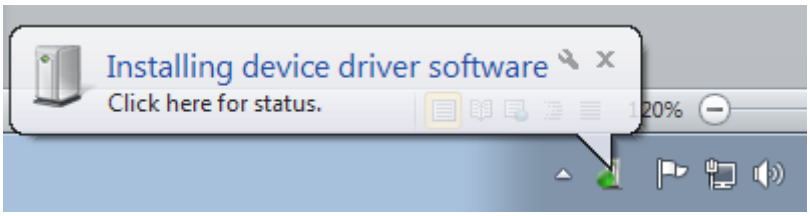
Extract the Lab Files

Create a folder called "***ADIEvalBoardLab***" on your PC and extract the ***ad7367_evalboardlab.zip*** archive to this folder. Make sure that there are ***NO SPACES*** in the directory path. After extracting the archive the following folders should be present in the ***ADIEvalBoardLab*** folder: ***FPGA***, ***Hdl Software***, ***DataCapture***, ***NiosCpu***.

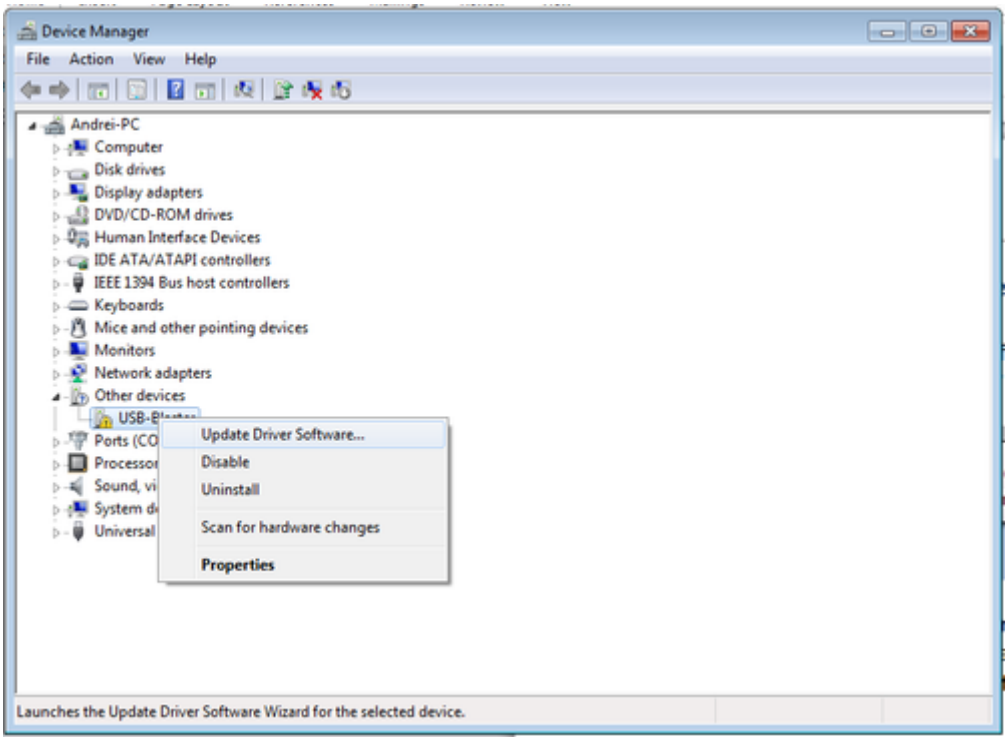
Folder	Description
FPGA	Contains all the files necessary to program the BeMicro FPGA board in order to run the evaluation project. By executing the script *program_fpga.bat* the FPGA will be programmed with the evaluation project. New NIOS II applications can be created using the files from this folder. The *ip* subfolder contains the AD7367 NIOS II peripheral's source code.
Hdl	Contains the source files for the AD7367 HDL driver: - The *doc* subfolder contains a brief documentation for the driver. - The *src* subfolder contains the HDL source files. - The *tb* folder contains the sources of the driver's testbench.
NiosCpu	Contains the Quartus evaluation project source files . The *ip* subfolder contains the AD7367 Nios2 peripheral source code.
Software	Contains the source files of the Nios2 SBT evaluation project.
DataCapture	Contains the script files used for data acquisition

Install the USB-Blaster Device Driver

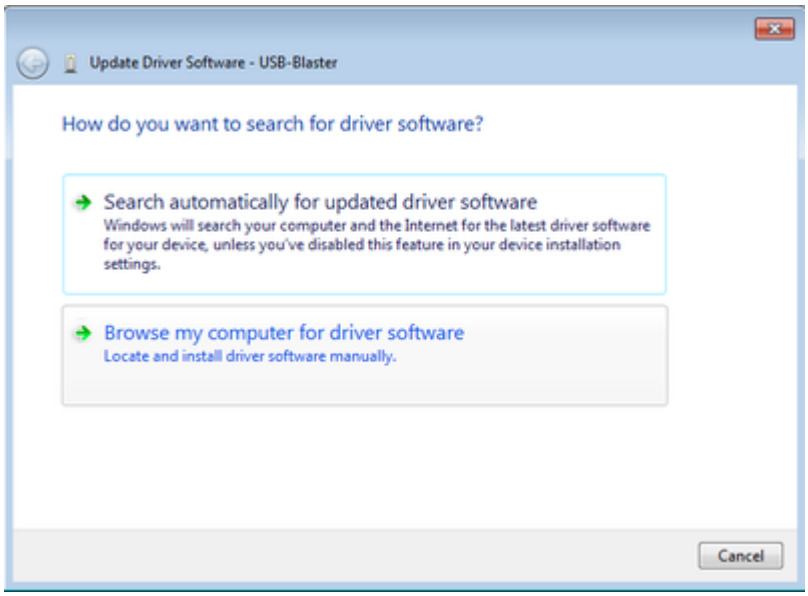
After the **Quartus II** and **Nios II** software packages are installed, you can plug the BeMicro SDK board into your USB port. Your Windows PC will find the new hardware and try to install the driver.

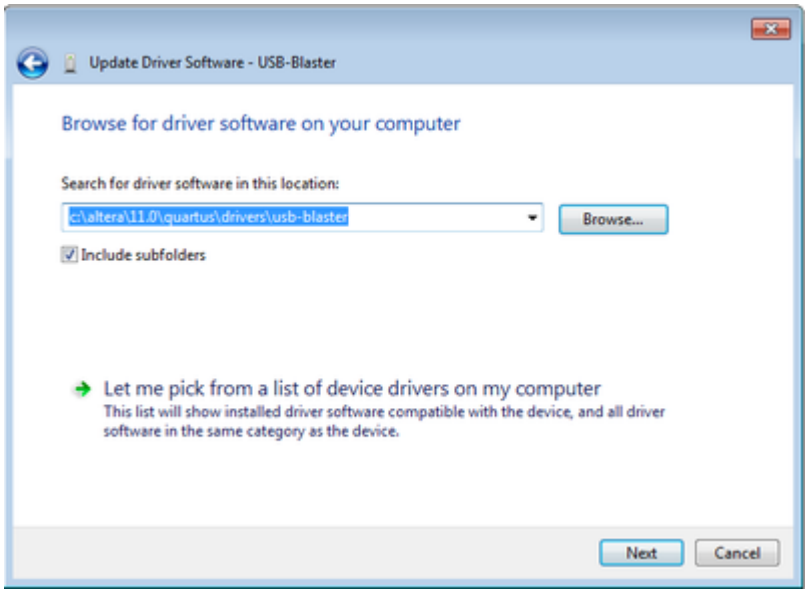


Since Windows cannot locate the driver for the device the automatic installation will fail and the driver has to be installed manually. In the *Device Manager* right click on the **USB-Blaster** device and select ***Update Driver Software***.

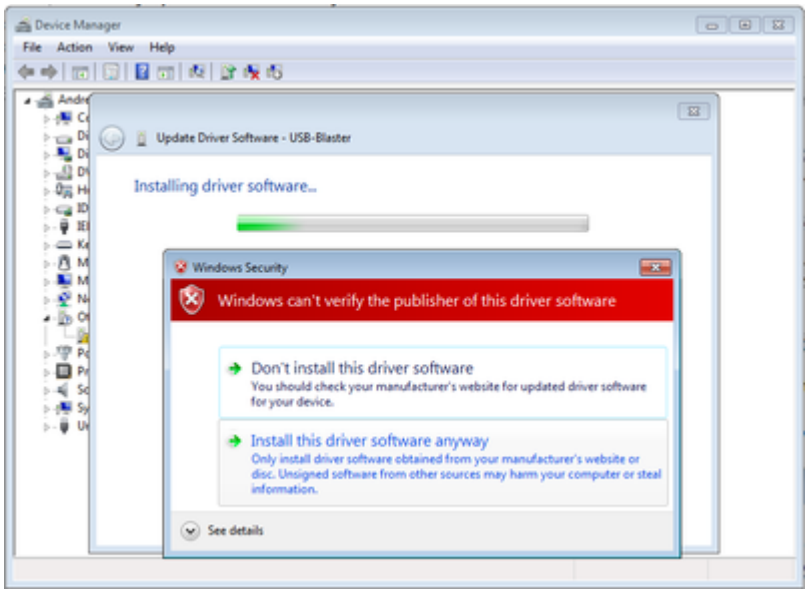


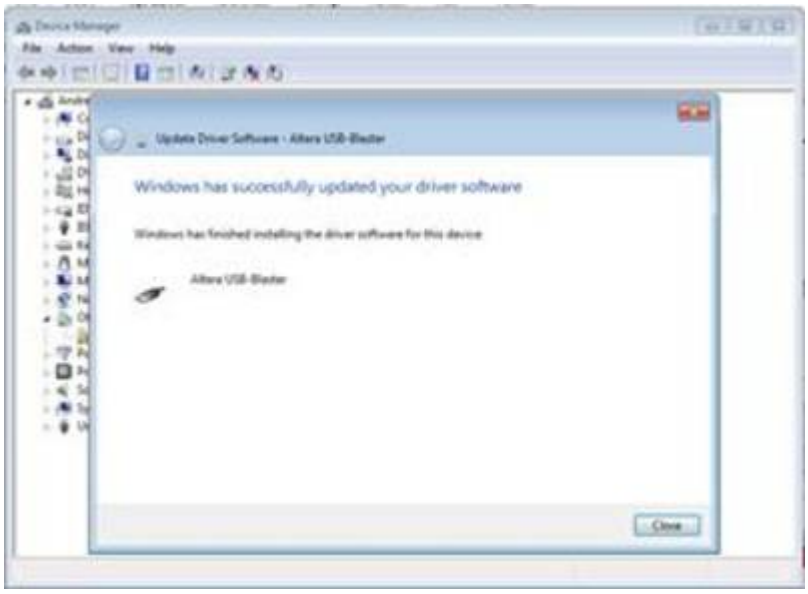
In the next dialog box select the option ***Browse my computer for driver software***. A new dialog will open where it is possible to point to the driver's location. Set the location to **altera\<*version number*>\quartus\drivers\usb-blaster** and press ***Next***.





If Windows presents you with a message that the drivers have not passed Windows Logo testing, please click "***Install this driver software anyway***". Upon installation completion a message will be displayed to inform that the installation is finished.





15 Sep 2011 14:23 · Andrei Cozma

AD7367 Evaluation Project Overview

The evaluation project contains all the source files needed to build a system that can be used to configure the AD7367 and capture data from it. The system consists of a Nios II softcore processor that is implemented in the FPGA found on the BeMicro board and a PC application. The softcore controls the communication with the Device Under Test (DUT) and the data capture process. The captured data is saved into the onchip RAM of the BeMicro board and aftwerwards it is read by the PC application and saved into a comma separated values (.csv) file that can be used for further data analysis.

The following components are implemented in the FPGA design:

Name	Address	IRQ
CPU	0x00000800	-
JTAG UART	0x00000090	0
uC-Probe UART	0x000000A0	1
EPCS FLASH CONTROLLER	0x00001800	2
OnChip RAM	0x00010000	-
LED GPIO	0x00000100	-
GPIO	0x00002080	-
CTRL GPIO	0x000020A0	-
SYS ID	0x00000040	-
TIMER	0x00000060	3
AVALON MASTER	-	-
Main PLL	0x00000080	-
AD7367 0	0x00000120	-

Table 1 System components

The Nios II processor contains a peripheral that implements the communication protocol with the DUT. The peripheral is divided into three logical modules: a module which implements the interface with the Avalon bus and the communication with the onchip RAM, a module which implements an Avalon master interface which is used to write data directly in the onchip RAM and a module which is the actual driver of the DUT. The driver can also be used as standalone in FPGA designs which do not contain a softcore. Following is presented a block diagram of the HDL driver and a description of the driver's interface signals.

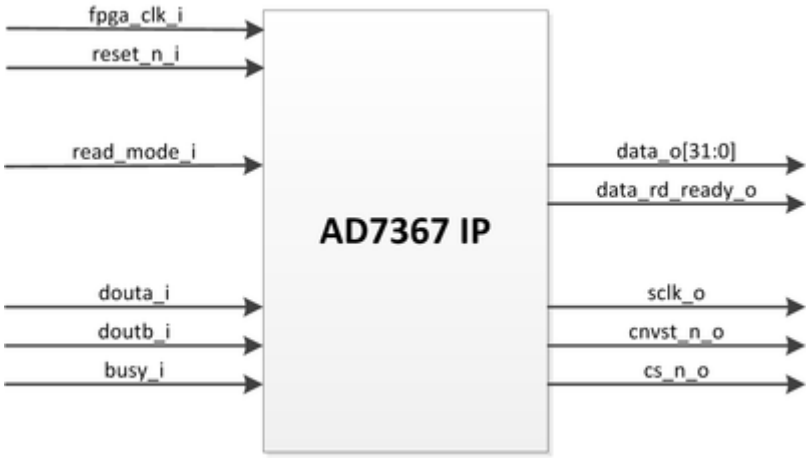


Table 2 describes the ports of the AD7367 HDL driver.

Port	Direction	Width	Description
Clock and reset ports			
FPGA_CLK_I	IN	1	Main clock input. 100MHz
RESET_I	IN	1	Active low reset signal
IP control and data ports			
READ_MODE_I	IN	1	Signal used to select between single line read mode(HIGH) or dual read mode(LOW)
DATA_O	OUT	32	Outputs the data read from the ADC. The most significant two bytes store the data from channel A, and the least significant two bytes store data from channel B
DATA_RD_READY_O	OUT	1	Signals that at port DATA_O there is new data available
AD7367 ports			
DOUTA_I	IN	1	The data output is supplied to each pin as a serial data stream. The bits are clocked out on the falling edge of the SCLK input. The data simultaneously appears on both pins from the simultaneous conversions of both ADCs
DOUTB_I	IN	1	The data output is supplied to each pin as a serial data stream. The bits are clocked out on the falling edge of the SCLK input. The data simultaneously appears on both pins from the simultaneous conversions of both ADCs
BUSY_I	IN	1	Busy transitions high when a conversion is started and remains high until the conversion is complete. It is not used by the current driver
SCLK_O	OUT	1	This pin provides the SCLK for accessing the data from AD7367
CNVST_N_O	OUT	1	This pin is edge triggered. On the falling edge of this input, the track/hold goes into hold mode and the conversion is initiated.

Port	Direction	Width	Description
CS_N_O	OUT	1	This signal frames the serial data transfer.
Table 2 AD7367 driver ports description			

The following figures present the timing diagram for the read operation for the AD7367 driver:

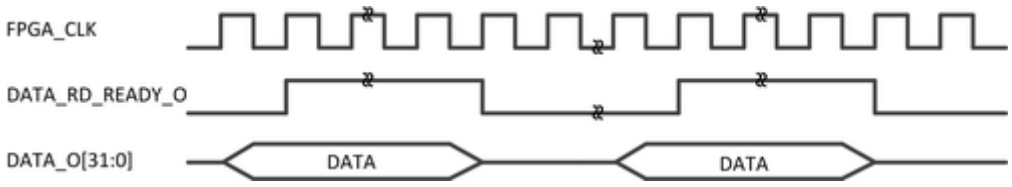


Table 3 describes the ports of the Avalon peripheral:

Port	Direction	Width	Description
Clock and reset ports			
CLK_I	IN	1	Main clock input
RESET_I	IN	1	System reset
Avalon Slave Interface			
AVALON_WRITEDATA_I	IN	32	Slave write data bus
AVALON_WRITE_I	IN	1	Slave write data request
AVALON_READ_I	IN	1	Slave read data request
AVALON_ADDRESS_I	IN	2	Slave address bus
AVALON_READDATA_O	OUT	32	Slave read data bus
Avalon Master Interface			
AVALON_MASTER_WAITREQUEST	IN	1	Master wait request signal
AVALON_MASTER_ADDRESS_O	OUT	32	Master address bus
AVALON_MASTER_WRITE_O	OUT	1	Master write signal
AVALON_MASTER_BYTEENABLE_O	OUT	4	Master byte enable signals
AVALON_MASTER_WRITEDATA_O	OUT	32	Master write data bus
External connectors			
DOUTA_I	IN	1	The data output is supplied to each pin as a serial data stream. The bits are clocked out on the falling edge of the SCLK input. The data simultaneously appears on both pins from the simultaneous conversions of both ADCs
DOUTB_I	IN	1	The data output is supplied to each pin as a serial data stream. The bits are clocked out on the falling edge of the SCLK input. The data simultaneously appears on both pins from the simultaneous conversions of both ADCs
BUSY_I	IN	1	Busy transitions high when a conversion is started and remains high until the conversion is complete. It is not used by the current driver
SCLK_O	OUT	1	This pin provides the SCLK for accessing the data from AD7367
CNVST_N_O	OUT	1	This pin is edge triggered. On the falling edge of this input, the track/hold goes into hold mode and the conversion is initiated.

Port	Direction	Width	Description
CS_N_O	OUT	1	This signal frames the serial data transfer.
Table 3 Avalon peripheral ports description			

Table 4 describes the registers of the Avalon peripheral:

Name	Offset	Width	Access	Description
CONTROL_REGISTER	0	32	RW	Bit 0 is used to start data acquisition Bit 1 is used to initiate software reset of the core Bit 2 is used to configure the Avalon write master core to write data to the same location Bit 3 is used to write data to the AD7367 driver
ACQ_COUNT_REGISTER	1	32	RW	Register used to configure the number of samples to be acquired when acquisition is started
BASE_REGISTER	2	32	RW	Register used to configure the base address of the memory location where the acquired data is to be written
STATUS_REGISTER	3	32	R	Bit 0 is used to signal that the acquisition is complete Bit 1 is used to signal that the internal memory buffer has been overflown Bit 2 is used to signal that the user has performed a read of an unavailable register
DUT_WRITE_REGISTER	4	32	RW	Data to be written to the ADC driver. The Least significant bit controls the READ_MODE_I pin from the AD7367 driver. READ_MODE_I is a signal used to select between single line read mode(HIGH) or dual read mode(LOW)
Table 4 Avalon peripheral registers description				

Quick Evaluation

The next sections of this lab present all the steps needed to create a fully functional project that can be used for evaluating the operation of the ADI platform. It is possible to skip these steps and load into the FPGA an image that contains a fully functional system.The first step of the quick evaluation process is to program the FPGA with the image provided in the lab files. Before the image can be loaded the **Quartus II Web Edition** tool or the Quartus II Programmer must be installed on your computer. To load the FPGA image run the ***program_fpga.bat*** batch file located in the ***ADIEvalBoardLab/FPGA*** folder. After the image was loaded the system must be reset. Now the FPGA contains a fully functional system and it is possible to skip directly to the **Demonstration Project User Interface** section of this lab.

03 Sep 2012 16:42 · Adrian Costina

NIOS II Software Design

This section presents the steps for developing a software application that will run on the **BeMicroSDK** system and will be used for controlling and monitoring the operation of the ADI evaluation board.

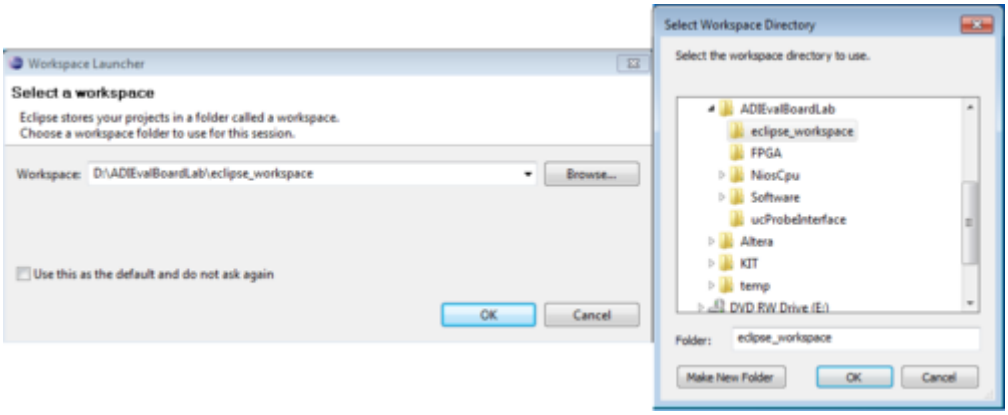
Create a new project using the NIOS II Software Build Tools for Eclipse

Launch the **Nios II SBT** from the ***Start → All Programs → Altera 12.0sp2→ Nios II EDS 12.0sp2 → Nios II 12.0sp2 Software Build Tools for Eclipse (SBT)***.

NOTE: Windows 7 users will need to right-click and select ***Run as administrator***. Another method is to right-click and select ***Properties*** and click on the ***Compatibility*** tab and select the ***Run This Program As An Administrator*** checkbox, which will make this a permanent change.

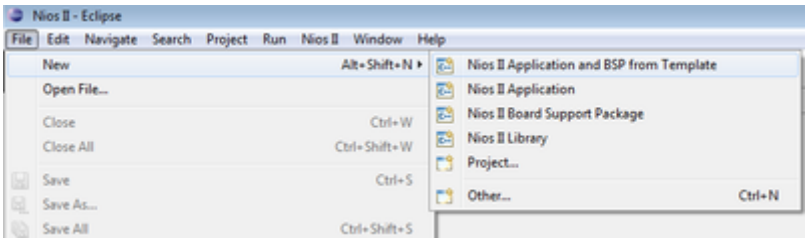
1. Initialize Eclipse workspace

- When Eclipse first launches, a dialog box appears asking what directory it should use for its workspace. It is useful to have a separate Eclipse workspace associated with each hardware project that is created in SOPC Builder. Browse to the ***ADIEvalBoardLab*** directory and click ***Make New Folder*** to create a folder for the software project. Name the new folder "***eclipse_workspace***". After selecting the workspace directory, click ***OK*** and Eclipse will launch and the workbench will appear in the ***Nios II*** perspective.

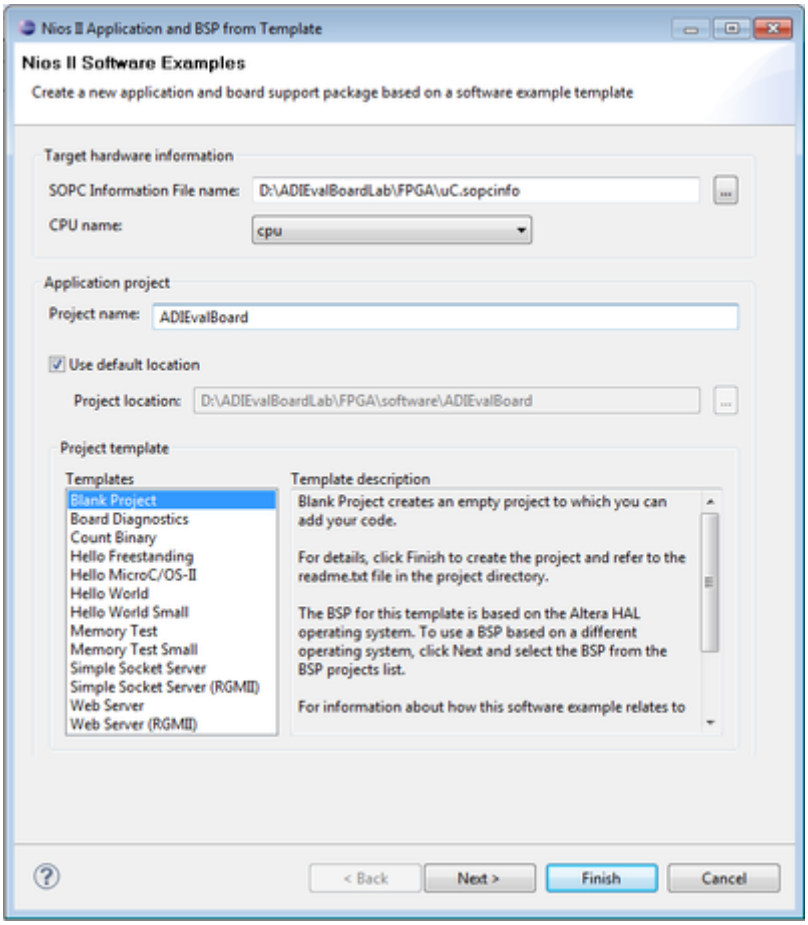


2. Create a new software project in the SBT

- Select ***File → New → Nios II Application and BSP from Template***.

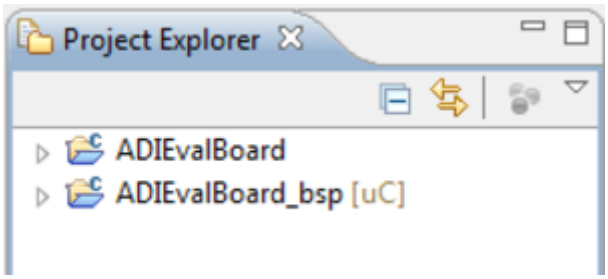


- Click the ***Browse*** button in the ***SOPC Information File Name*** dialog box.
- Select the ***uC.sopcinfo*** file located in the ***ADIEvalBoardLab/FPGA*** directory.
- Set the name of the Application project to "***ADIEvalBoard***".
- Select the ***Blank Project*** template under ***Project template***.
- Click the ***Finish*** button.



The tool will create two new software project directories. Each Nios II application has 2 project directories in the Eclipse workspace.

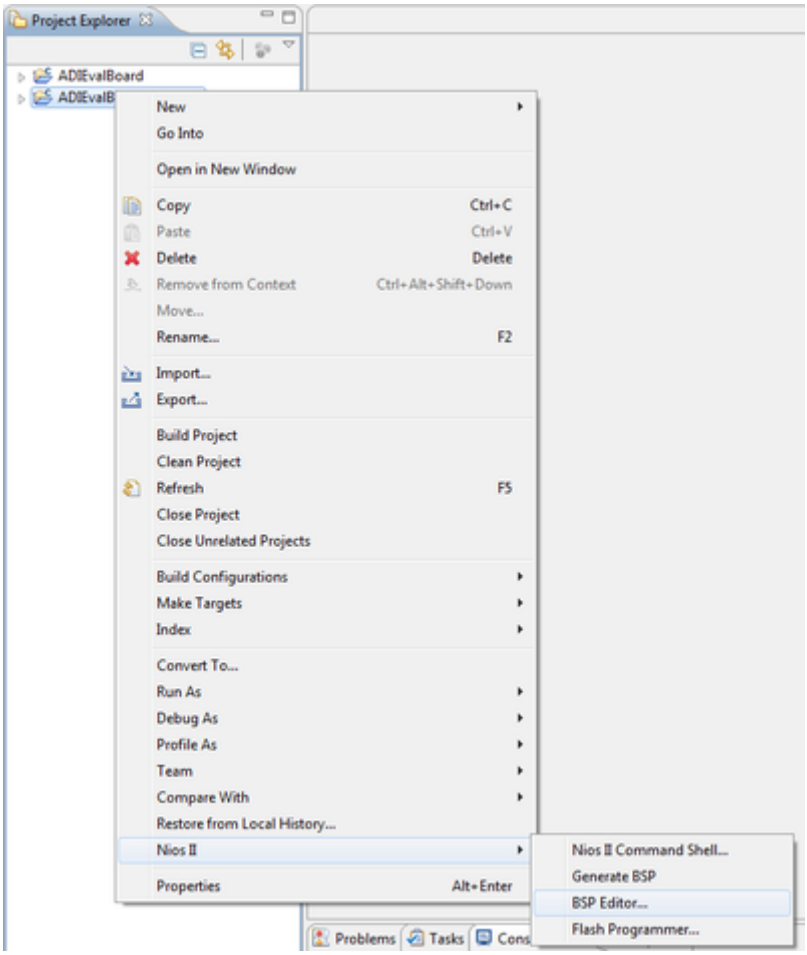
- The application software project itself - this where the application lives.
- The second is the ***Board Support Package (BSP)*** project associated with the main application software project. This project will build the system library drivers for the specific SOPC system. This project inherits the name from the main software project and appends "***_bsp***" to that.



Since you chose the blank project template, there are no source files in the application project directory at this time. The BSP contains a directory of software drivers as well as a system.h header file, system initialization source code and other software infrastructure.

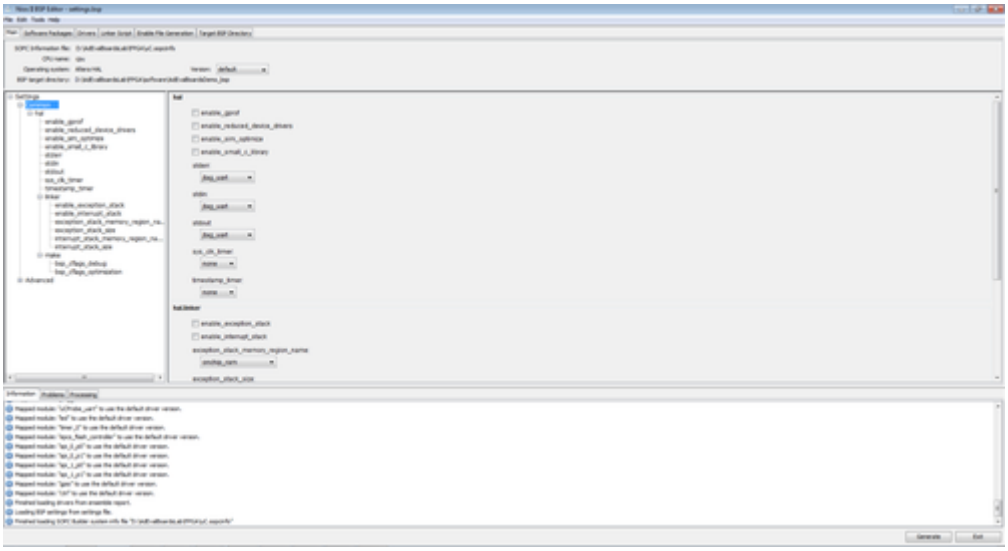
Configure the Board Support Package

- Configure the board support package to specify the properties of this software system by using the ***BSP Editor*** tool. These properties include what interface should be used for *stdio* and *stderr* messages, the memory in which stack and heap should be allocated and whether an operating system or network stack should be included with this BSP.
- Right click on the ***ADIEvalBoard_bsp*** project and select ***Nios II → BSP Editor…*** from the right-click menu.



The software project provided in this lab does not make use of an operating system. All *stdout*, *stdin* and *stderr* messages will be directed to the *jtag_uart*.

- Select the ***Common*** settings view. In the ***Common*** settings view, change the following settings:
  - Select the ***jtag_uart*** for *stdin*, *stdout* and *stderr* messages. Note that you have more than one choice.
  - Select ***none*** for the *sys_clk_timer* and *timestamp_timer*.

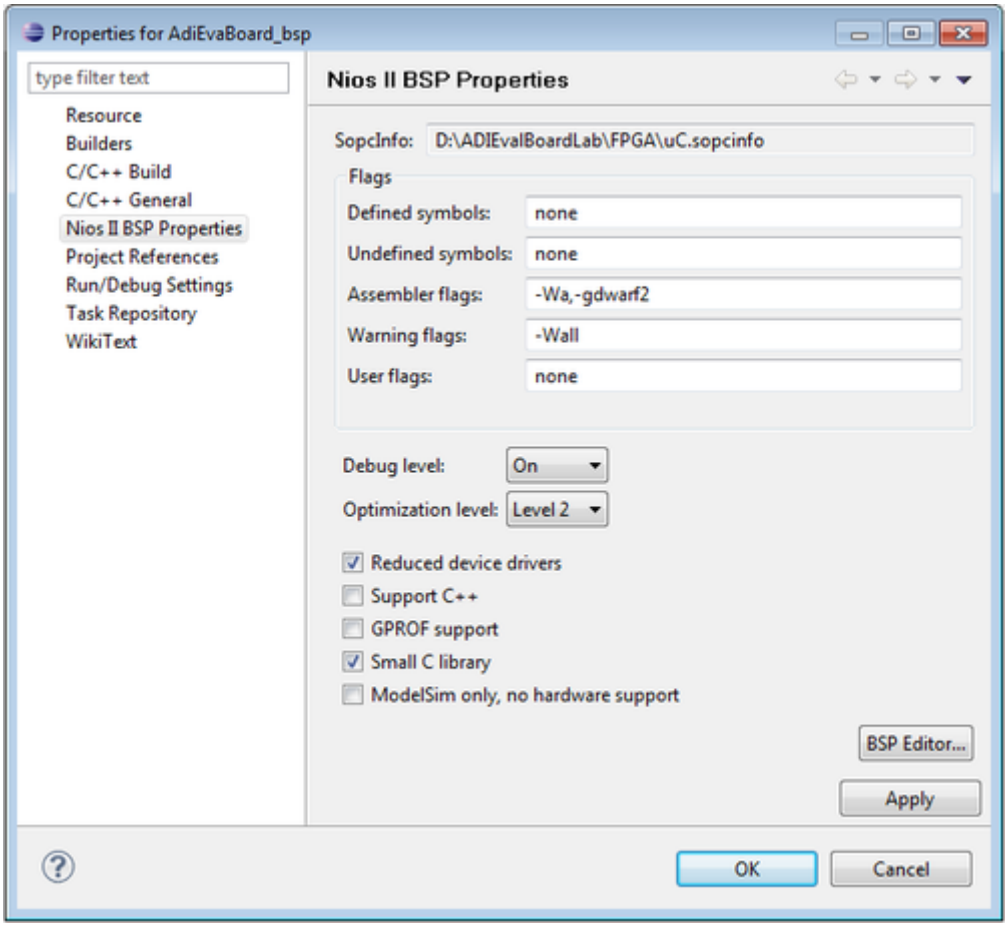


- Select ***File → Save*** to save the board support package configuration to the *settings.bsp* file.
- Click the ***Generate*** button to update the BSP.
- When the generate has completed, select ***File → Exit*** to close the BSP Editor.

Configure BSP Project Build Properties

In addition to the board support package settings configured using the ***BSP Editor***, there are other compilation settings managed by the Eclipse environment such as compiler flags and optimization level.

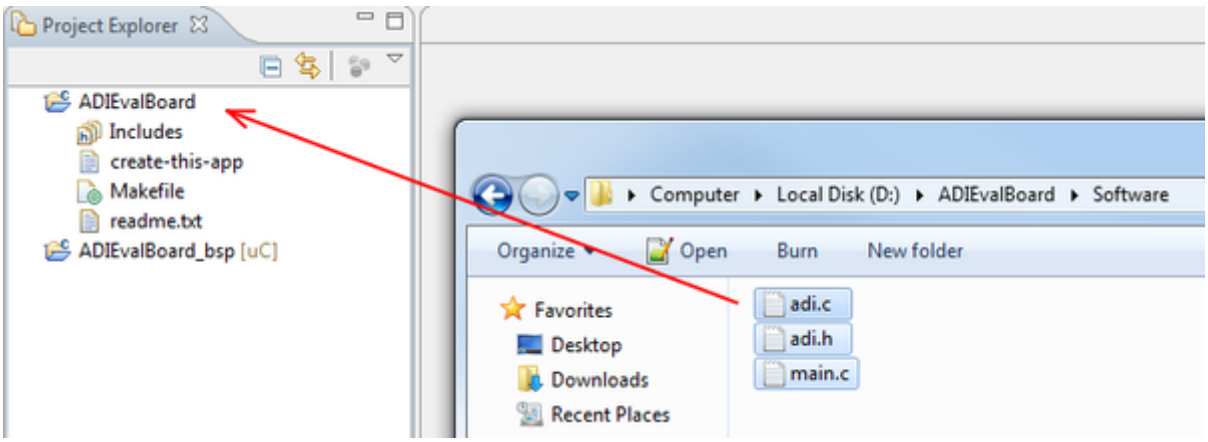
- Right click on the ***ADIEvalBoard_bsp*** software project and select ***Properties*** from the right-click menu.
- On the left-hand menu, select ***Nios II BSP Properties***.
- During compilation, the code may have various levels of optimization which is a tradeoff between code size and performance. Change the ***Optimization level*** setting to ***Level 2***
- Since our software does not make use of C++, uncheck ***Support C++***.
- Check the ***Reduced device drivers*** option
- Check the ***Small C library*** option
- Press ***Apply*** and ***OK*** to regenerate the BSP and close the ***Properties*** window.



Add source code to the project

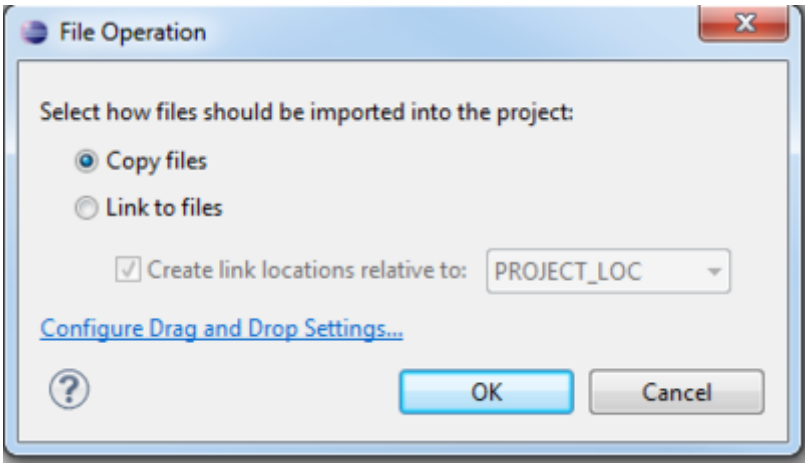
In Windows Explorer locate the project directory which contains a directory called ***Software***. In Windows Explorer select all the files and directories from the ***Software*** folder and drag and drop them into the Eclipse software project ***ADIEvalBoard***.

- Select all the files and folders and drag them over the ***ADIEvalBoard*** project in the SBT window and drop the files onto the project folder.

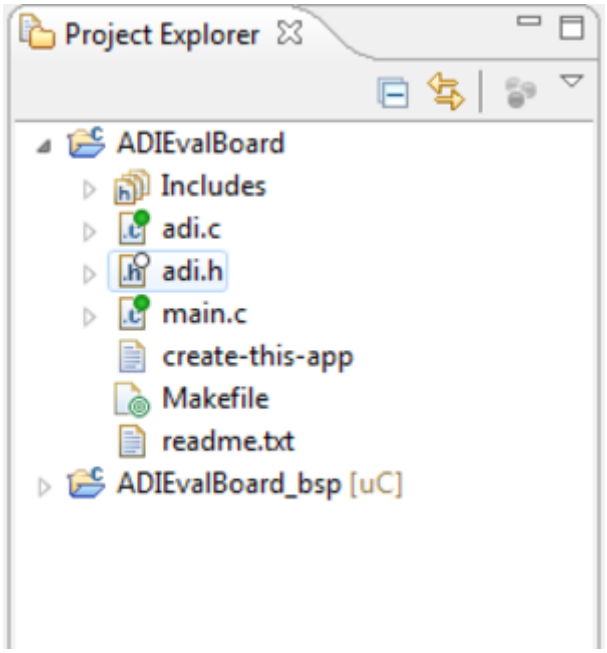


- A dialog box will appear to select the desired operation. Select the option ***Copy files and folders***

and press ***OK***.



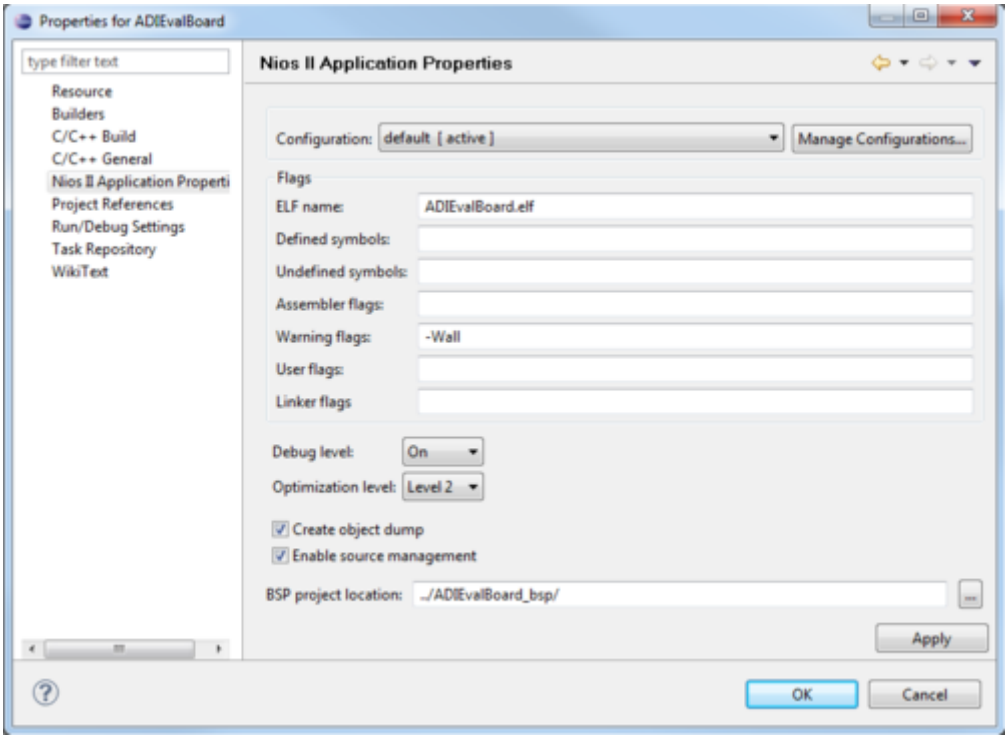
- This should cause the source files to be physically copied into the file system location of the software project directory and register these source files within the Eclipse workspace so that they appear in the Project Explorer file listing.



Configure Application Project Build Properties

Just as you configured the optimization level for the BSP project, you should set the optimization level for the application software project ***ADIEvalBoard*** as well.

- Right click on the ***ADIEvalBoard*** software project and select ***Properties*** from the right-click menu.
- On the left-hand menu, select the ***Nios II Application Properties*** tab
- Change the ***Optimization level*** setting to ***Level 2***.
- Press ***Apply*** and ***OK*** to save the changes.

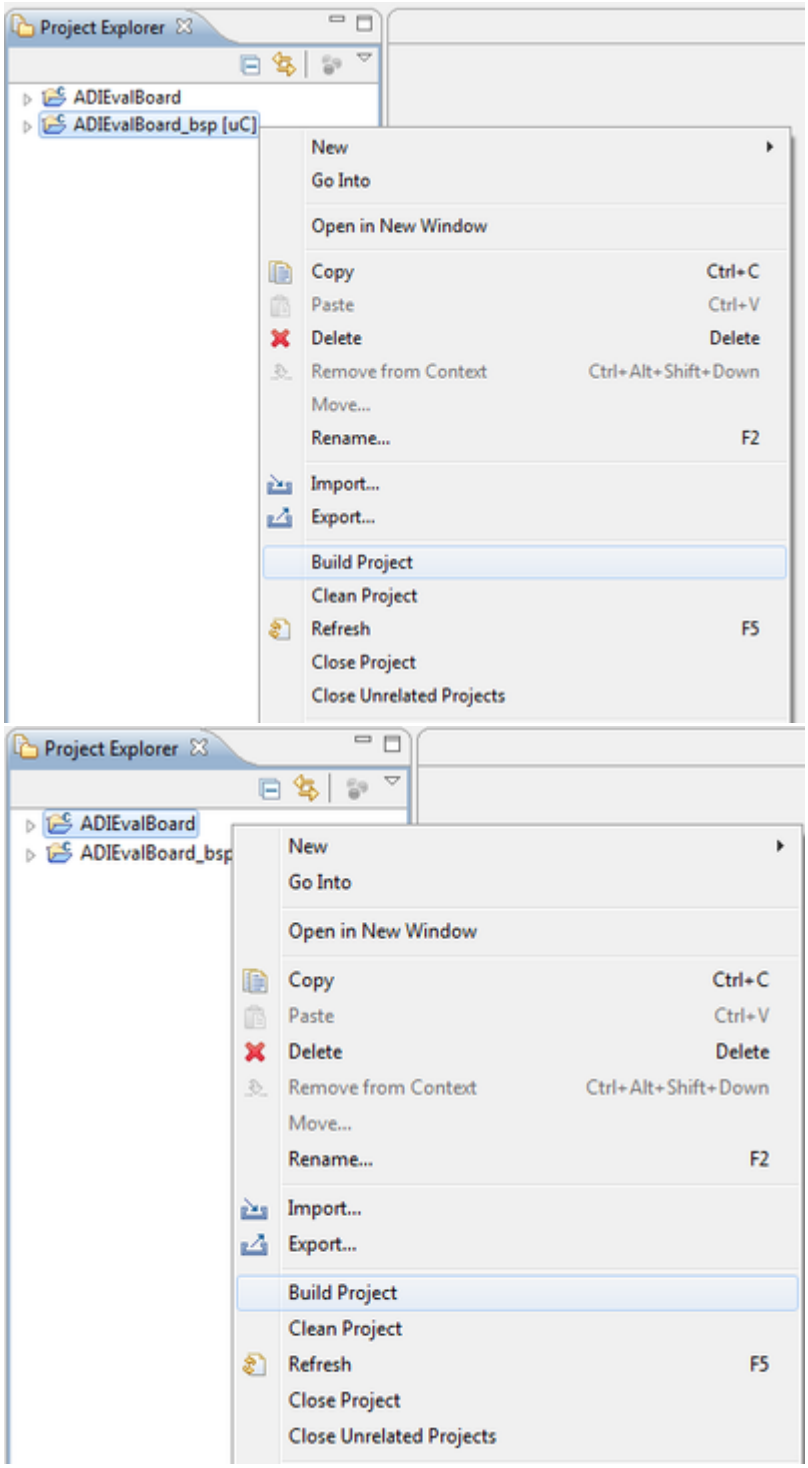


Compile, Download and Run the Software Project

1. Build the Application and BSP Projects

- Right click the ***ADIEvalBoard_bsp*** software project and choose ***Build Project*** to build the board support package.
- When that build completes, right click the ***ADIEvalBoard*** application software project and choose ***Build Project*** to build the Nios II application.

These 2 steps will compile and build the associated board support package, then the actual application software project itself. The result of the compilation process will be an *Executable and Linked Format (.elf)* file for the application, the ***ADIEvalBoard.elf*** file.

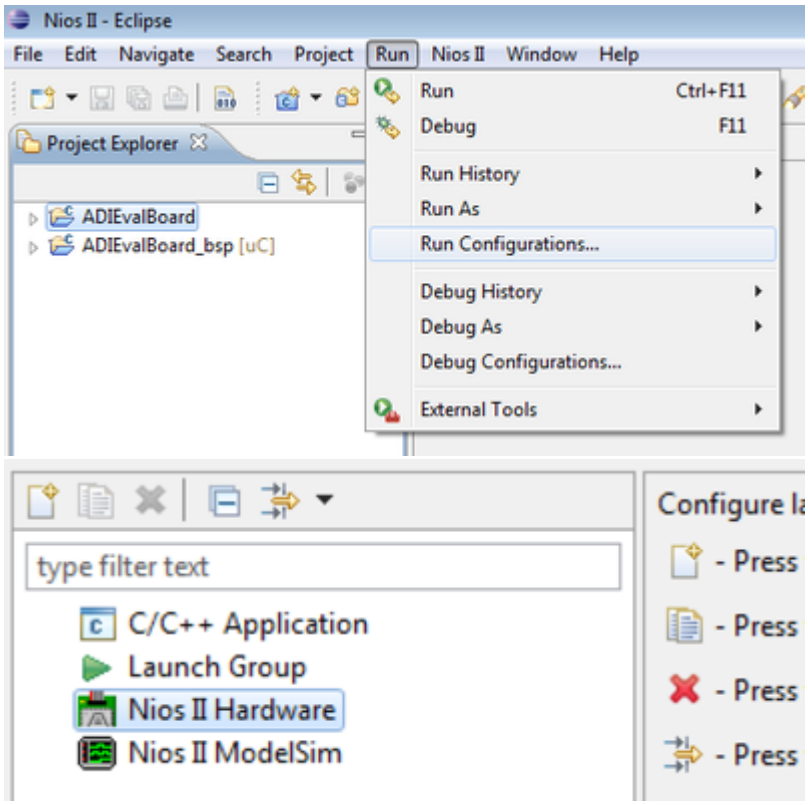


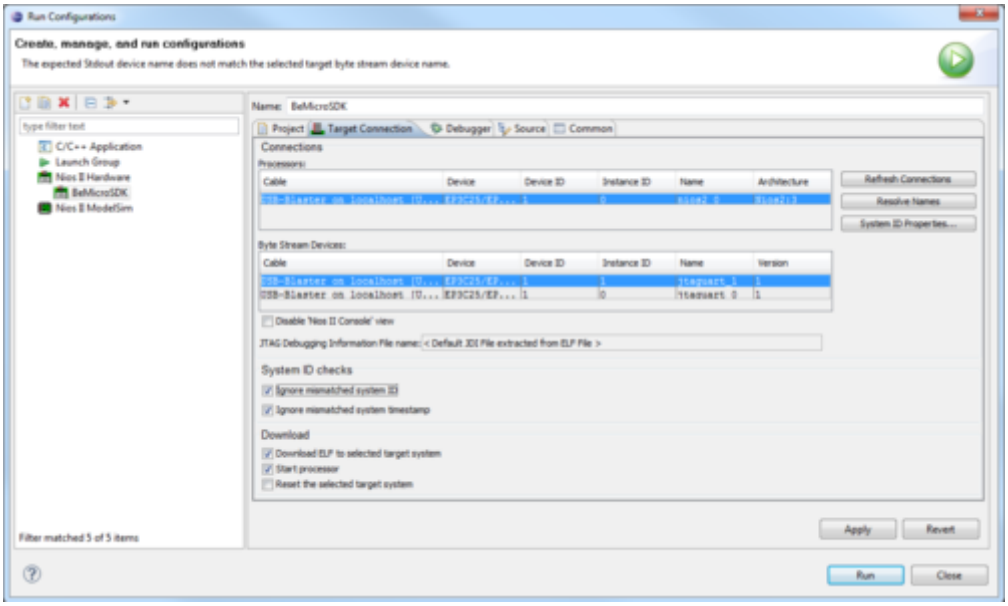
2. Verify the Board Connection

The **BeMicroSDK** hardware is designed with a *System ID* peripheral. This peripheral is assigned a unique value based on when the hardware design was last modified in the SOPC Builder tool. SOPC Builder also places this information in the *.sopcinfo* hardware description file. The BSP is built based on the information in the *.sopcinfo* file.

- Select the ***ADIEvalBoard*** application software project.
- Select ***Run → Run Configurations…***

- Select the ***Nios II Hardware*** configuration type.
- Press the ***New*** button to create a new configuration.
- Change the configuration name to ***BeMicroSDK*** and click ***Apply***.
- On the ***Target Connection*** tab, press the ***Refresh Connections*** button. You may need to expand the window or scroll to the right to see this button.
- Select the ***jtag_uart*** as the ***Byte Stream Device*** for *stdio*.
- Check the ***Ignore mismatched system ID option***.
- Check the ***Ignore mismatched system timestamp option***.



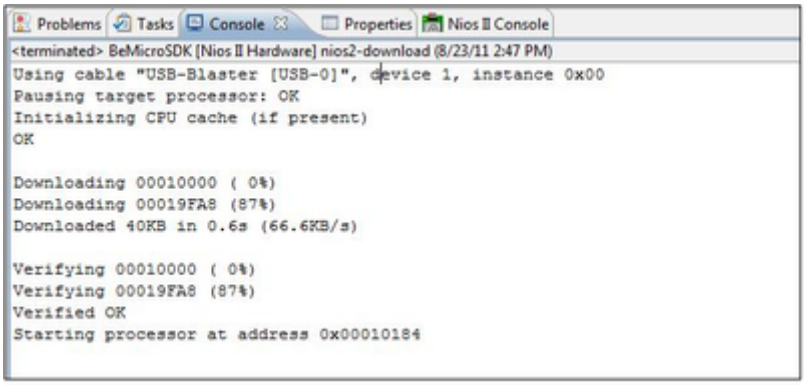


3. Run the Software Project on the Target

To run the software project on the Nios II processor:

- Press the ***Run*** button in the ***Run Configurations*** window.

This will re-build the software project to create an up–to-date executable and then download the code into memory on the **BeMicroSDK** hardware. The debugger resets the Nios II processor, and it executes the downloaded code. Note that the code is verified in memory before it is executed.

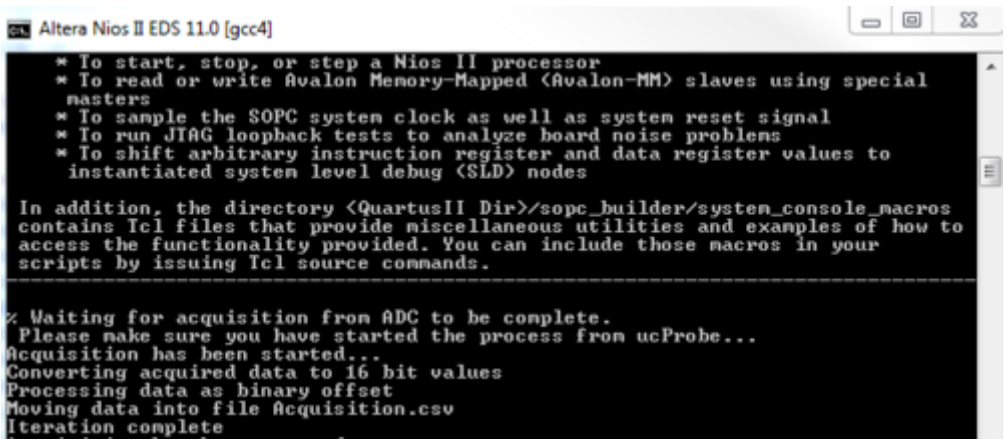


The code size and start address might be different than the ones displayed in the above screenshot.

04 Sep 2012 08:08 · Adrian Costina

Demonstration Project User Interface

After the FPGA is correctly programmed the data acquisition process can start by executing the data_capture.bat script. The data_capture.tcl file can be modified so to use either single line or dual line data acquisition.



If the resulting csv file is opened with Microsoft Excel, the data will be displayed on a different number of columns, each column corresponding to a channel.

Note: If several consecutive data acquisitions are performed the captured data is appended to the ***Acquisition.csv*** file.

Troubleshooting

In case there is a communication problem with the board the follwing actions can be perfomed in order to try to fix the issues:

- Check that the evaluation board is powered.
- Check that the USB connection cable is properly connected to the device and to the computer and that the ***USB Blaster Device Driver*** driver is installed correctly. If the driver is not correctly installed perform the steps described in the ***Getting Started → Install te USB-Blaster Device Driver*** section.

04 Sep 2012 08:24 · Adrian Costina

More information

- ask questions about the FPGA reference design
- Example questions:
  - FMCOMMS1 AD9548 clock derivation explanation by cherif.chibane@ll.mit.edu
  - FMCDAQ2 Arria10GX Nios Sourcecode by kairue
  - Using ZC706 and AD-fmcomms3 by 85083074@qq.com
  - FM-COMMS3 and FM-COMMS5 with VC707 vs Zync ZC706 by dr8
  - How Xps ip update to vivado ? by huanmolb@163.com