Peter the Great St. Petersburg Polytechnic University

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[Institute of Computer Science and Technology](http://english.spbstu.ru/structure/institut_computernikh_nauk_i_tekhnologiy/)

Department of Computer Systems & Software Engineering

Lecture

*Introduction to Nios II Processor Software Development*

student:

Ivanov Il’ya

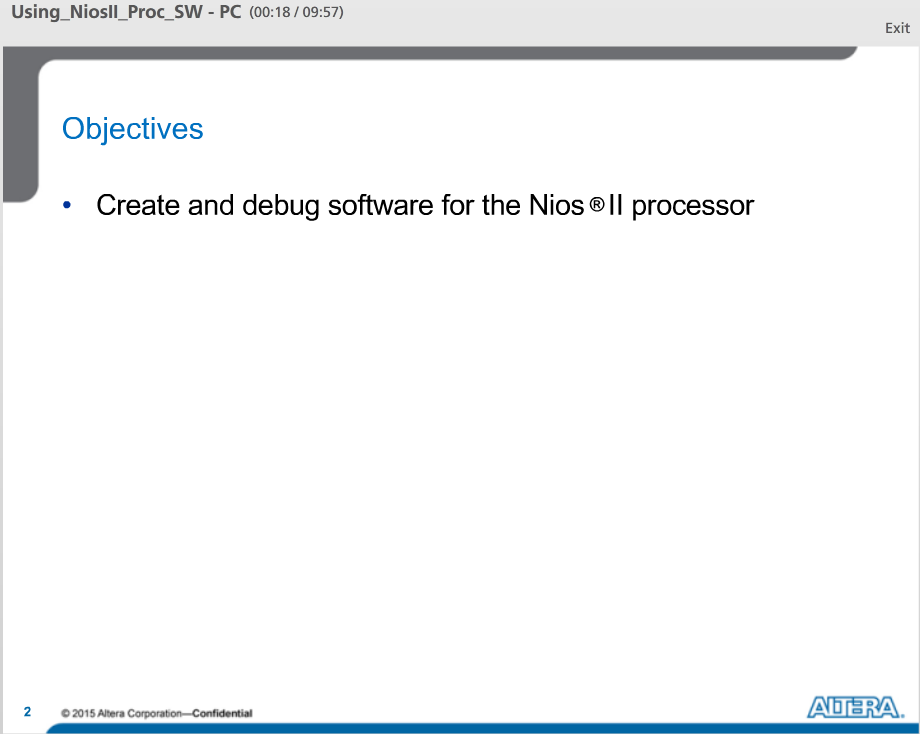
group: 3530901/70203

lecturer:

Antonov A.P.

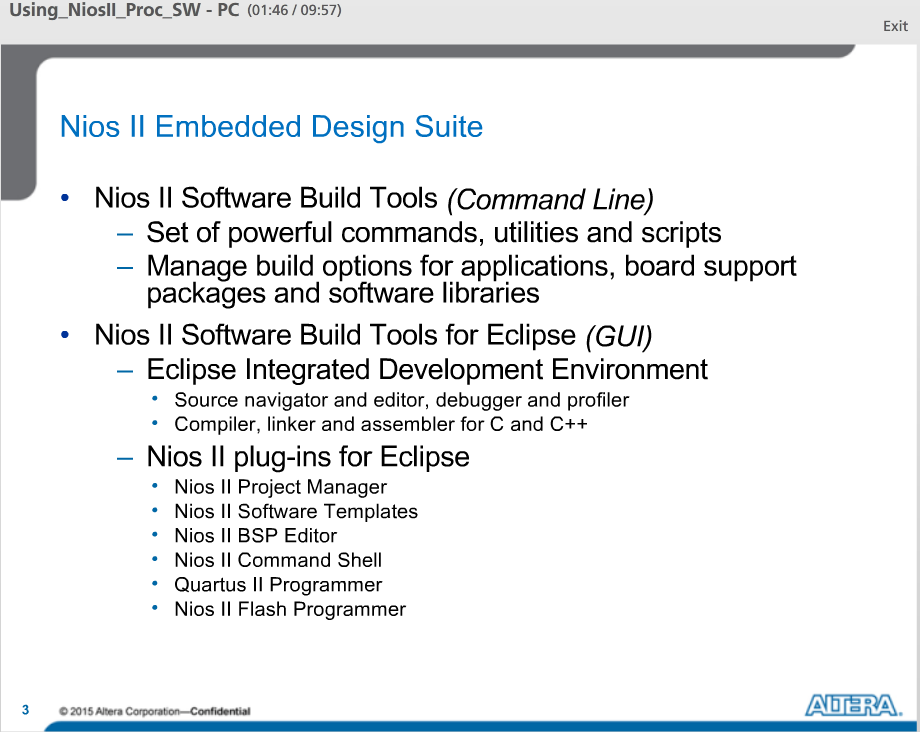
Saint-Petersburg

2020

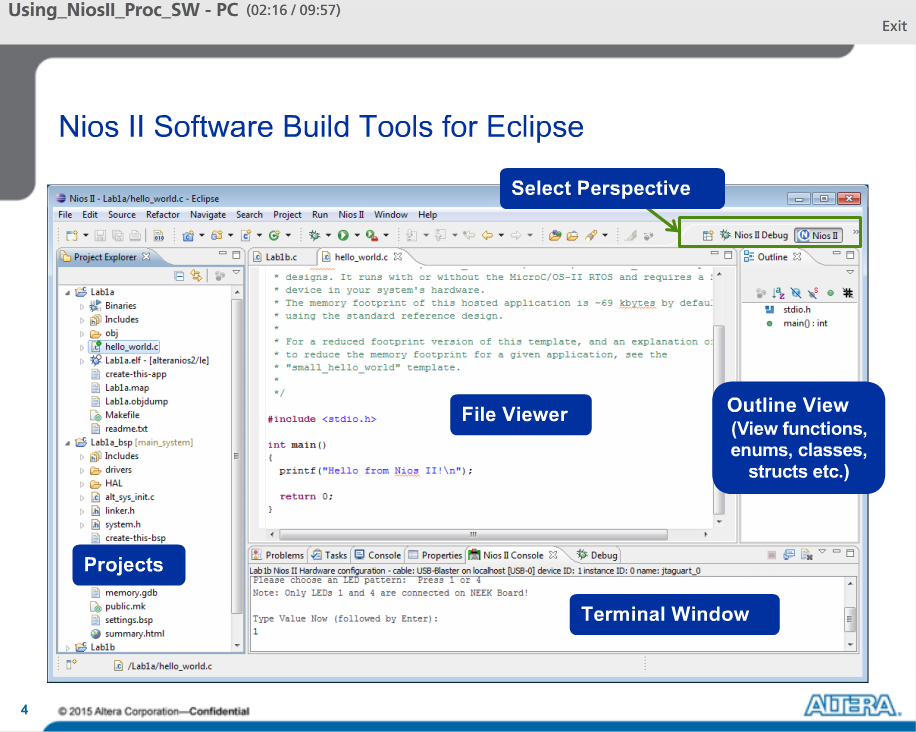


Objectives for this lecture:

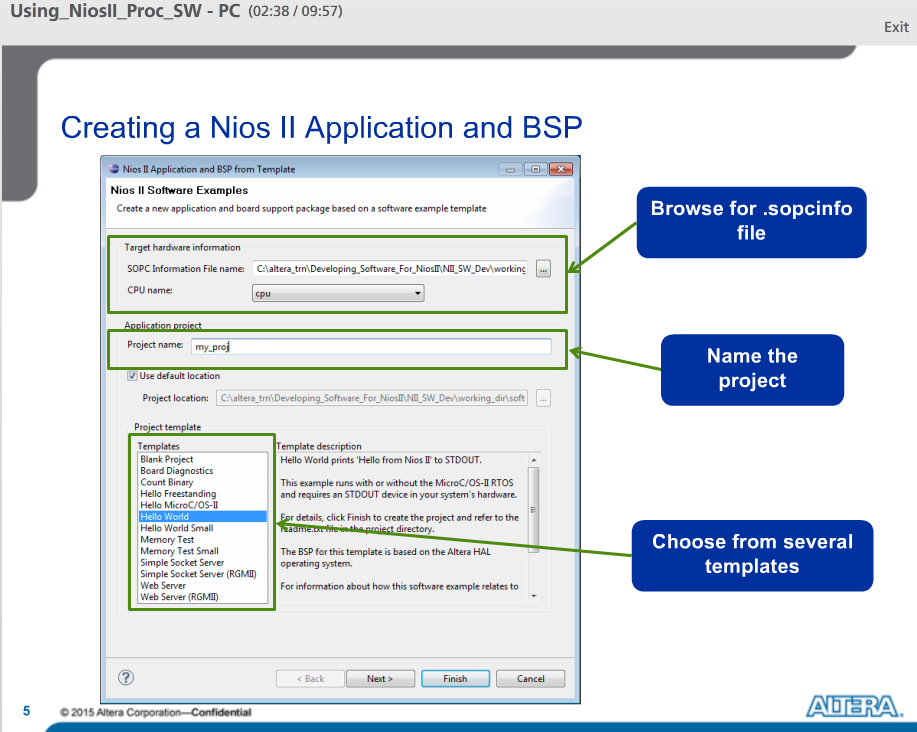
Learn how to create and debug software for a Nios II embedded system that you created in Qsys.



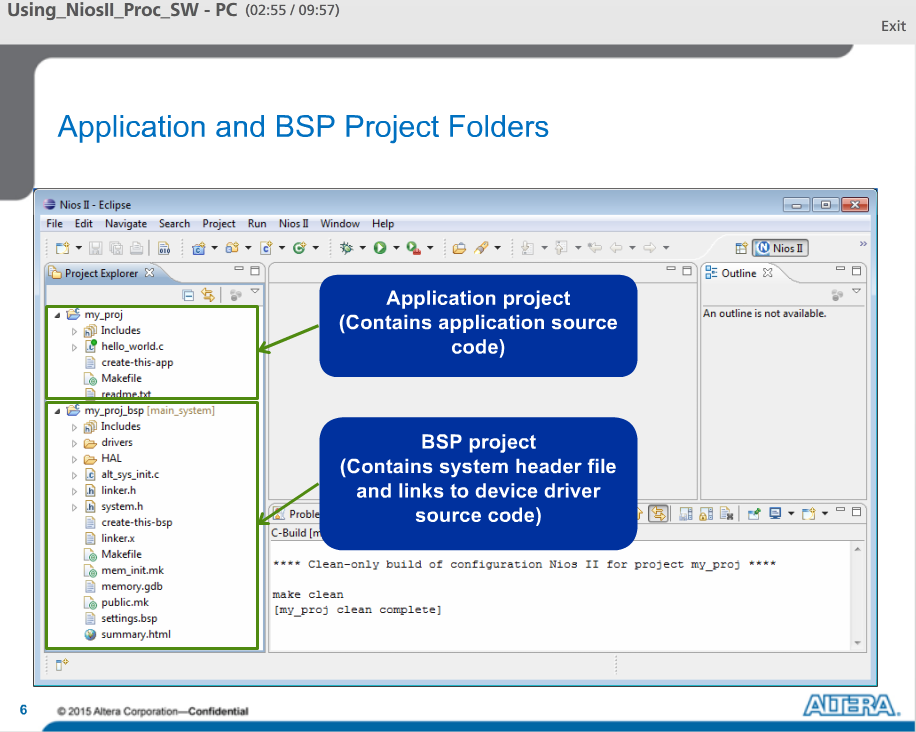
The Nios II Embedded Design suite or EDS is a series of cutting edge SW tools, utilities, libraries and drivers that help you bring your design to market in record time. The Suite includes the Nios II Software Build Tools for Eclipse, Nios II Software Build Tools Command line, embedded SW device drivers from Altera, IP and Hardware Abstraction Layer or HAL APIs. The Nios II SW Build tools are fully integrated into the development environment. The Nios II SBT focus on improved productivity for large designs and team based designs.



The Nios II SBT for Eclipse can be launched through Qsys or through the Windows Program menu. The GUI looks like this.

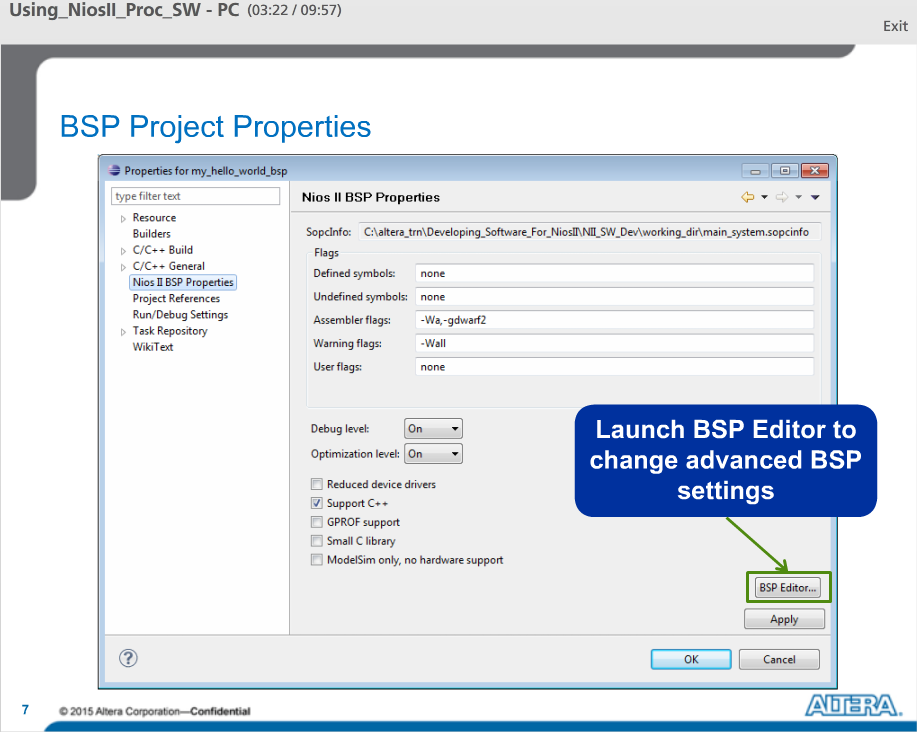


To create a project, the easiest thing to do is to use one of the templates. You must specify the SOPCinfo file to provide vital information about the HW system. In this dialog box, we specifying project name and the template we are using. In this case, we have chosen the Hello World template.

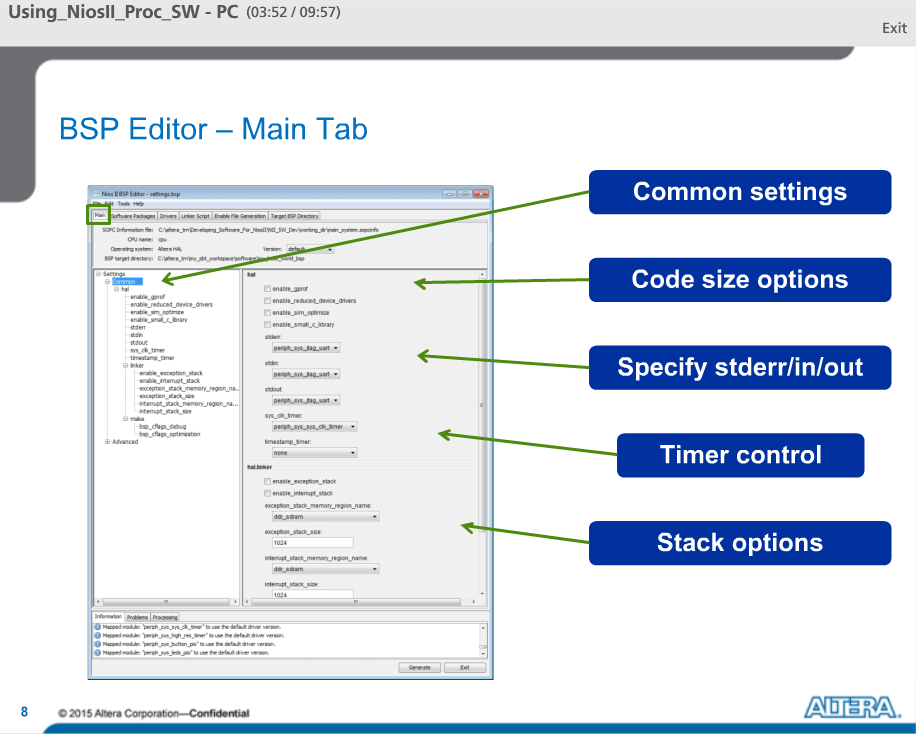


After creating the project, you will see two project directories in the Project Explorer. One for the application and the other for the BSP.

The application project contains all the source code in your project and the BSP project contains the system header file and links to device driver code.

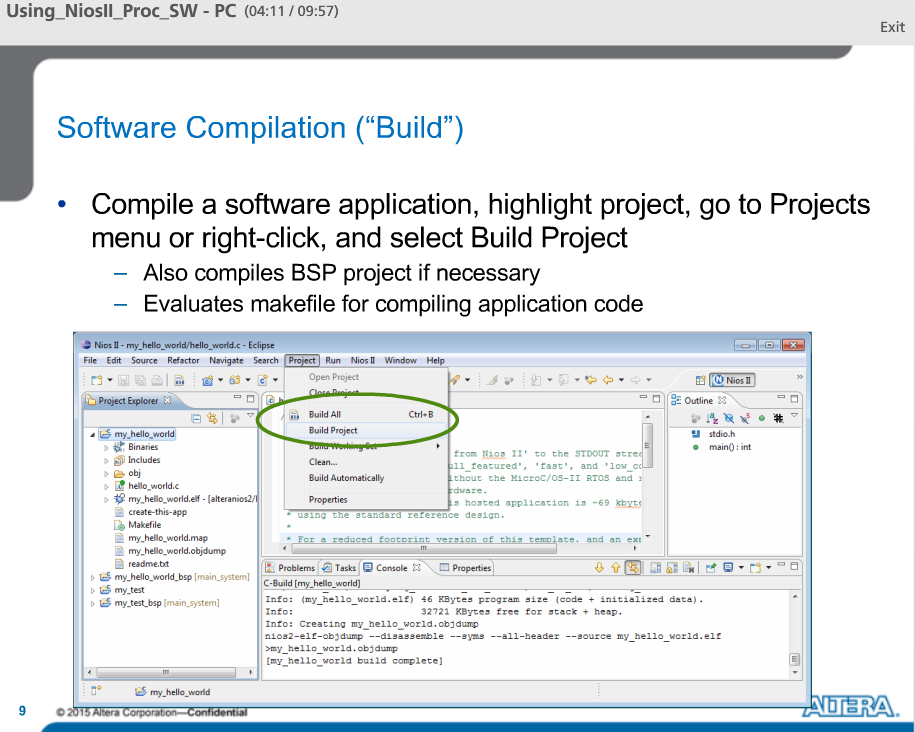


Right clicking on the BSP project and choosing properties allows the optimization levels to be changed for the board support package files. In addition, from here we can select reduced size device drivers, small C library and ModelSim simulation support. The BSP editor can be launched from the BSP Project Properties by pressing the BSP editor button. Settings such as debug level and optimization level are also available for the Application Project.

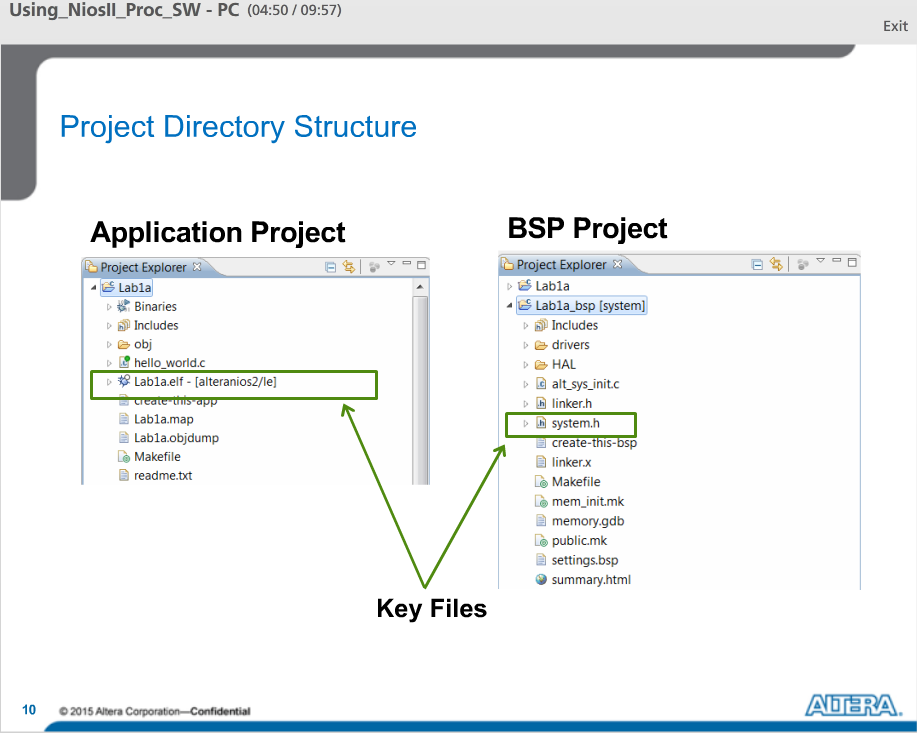


The BSP Editor allows the user to change more advanced settings in the board support package. The BSP editor consists of tabs across the top of the editor allowing the user to change to different categories in the editor.

After making changes, be sure to click the generate button to create a new BSP.

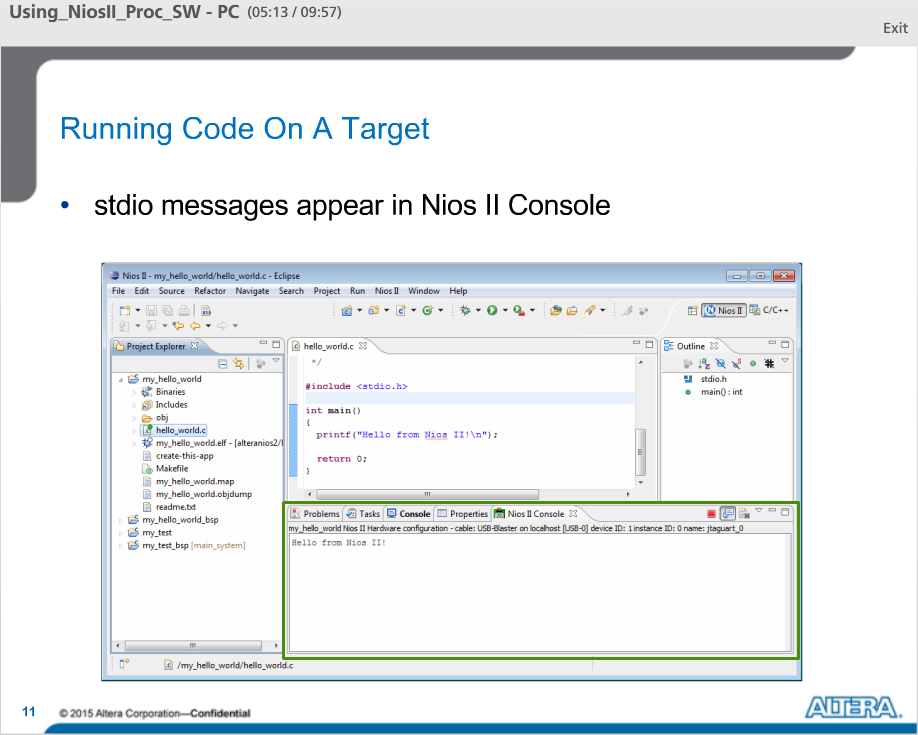


Eventually you will have to compile or build your project. In order to do that, you can highlight the application project, right click, and select Build Project. Alternatively, you can go to the project menu and select build project. Doing so will also compile the associated BSP project if needed.



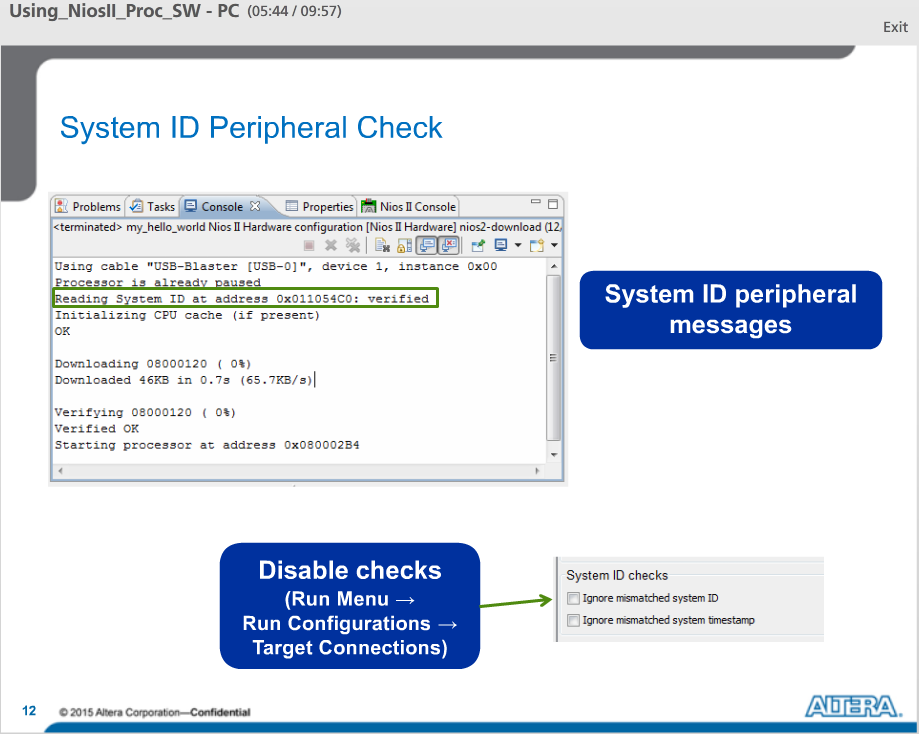
After you compile or build your projects, you will find that a number of files have been created inside both your Application project folder and your BSP project folder. In the application project, the .elf file contains all of your compiled code and device drivers and it is downloaded to the processor.

In the BSP project, the most important file to remember is the system.h file. This file contains all of the mappings between the Qsys peripheral names and their base addresses in the memory map.



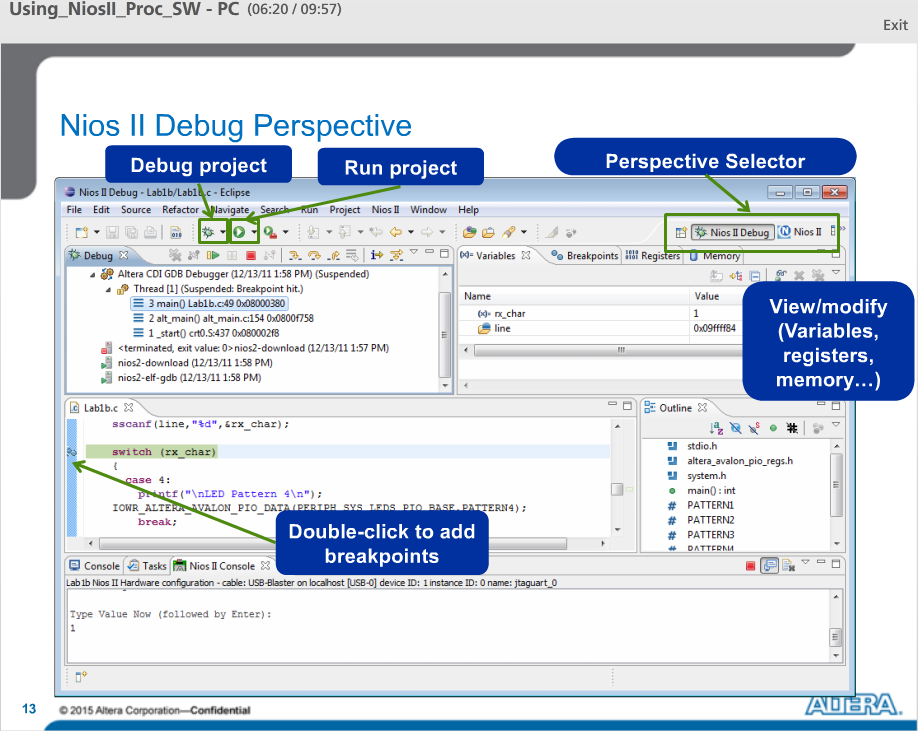
After you build your project, you download the code to the processor on your FPGA to run or debug it.

To run code on the target, right click on your application project and select Run As Nios II HW. That will download the code through the FPGA programming cable, in through the JTAG ports on the FPGA to the Processor where it will initiate the run sequence.

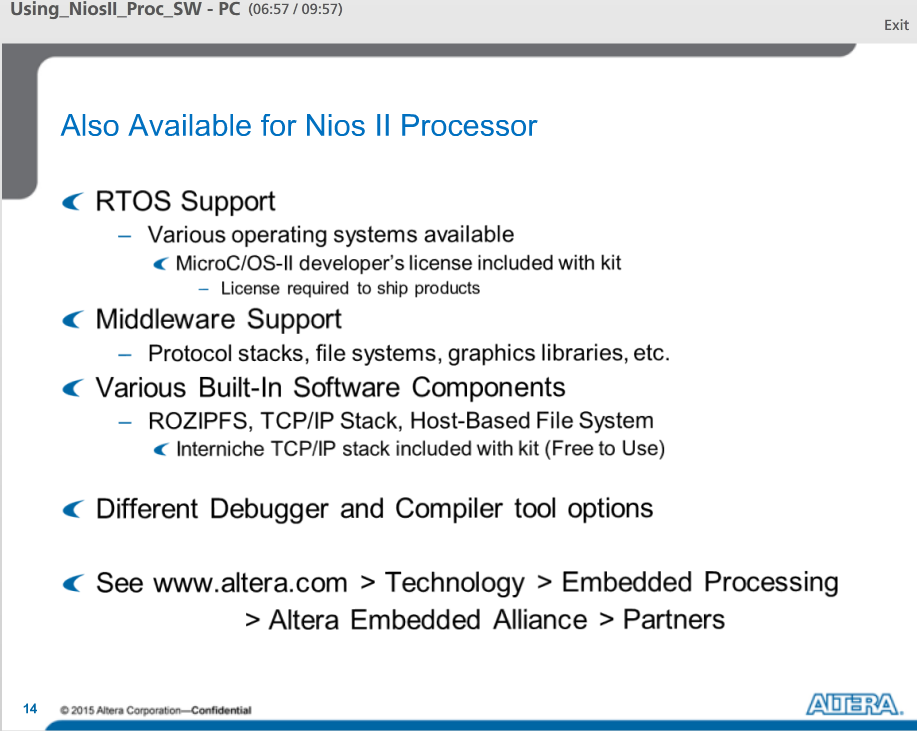


At run time, the Nios II Eclipse Platform makes sure the SW matches the HW on the FPGA. It does this by computing the expected System ID peripheral values from the .sopcinfo file and comparing them with values in the .sof file.

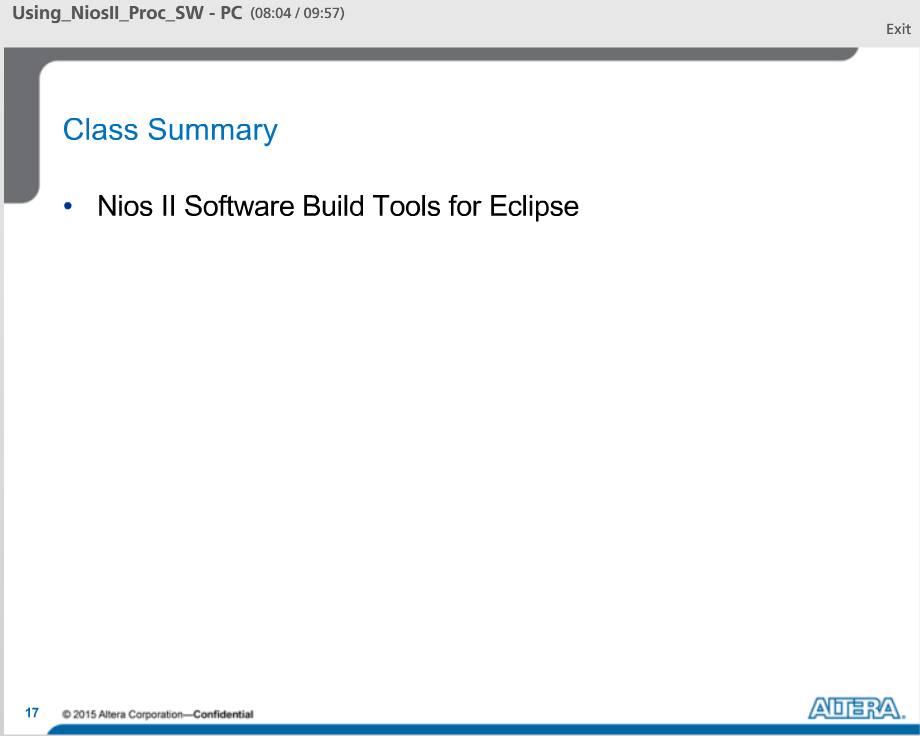
If computed ID values do not match System ID variables stored on target board, then an error is flagged.



The Nios II SBT for Eclipse have a debug perspective useful for debugging your software. Changing to the debug perspective can be done by right-clicking on an application and then selecting debug as Nios II hardware. This will download and run the code on the FPGA in debug mode. In the debug perspective, you can examine the program stack, breakpoints, registers, and memory. Using the Debugging Controls, you can suspend or restart the program, or you can step into or over certain lines of code.



A variety of real time operating system are supported, including uCOSII, which shipped with the kit along with full source code and developer’s license. There are a variety of other operating systems as well, a variety of middleware, Protocol Stacks for TCP/IP USB etc. and a RO file system. The Interniche TCP/IP Stack is shipped with the SW Development Tools and is available for use free of change.



In this lecture, we talked about the embedded SW design tools that we can download from Altera, including the Software Build tools for Eclipse.