ECE 332 – Embedded Systems Lab

Lab 1: Using NIOS II processor for code execution on FPGA

Objectives

- Introduction to Altera tools
- Walk through from project creation to chip configuration (Lab1-Part I)
- Instructions to get started with Part II

Tools

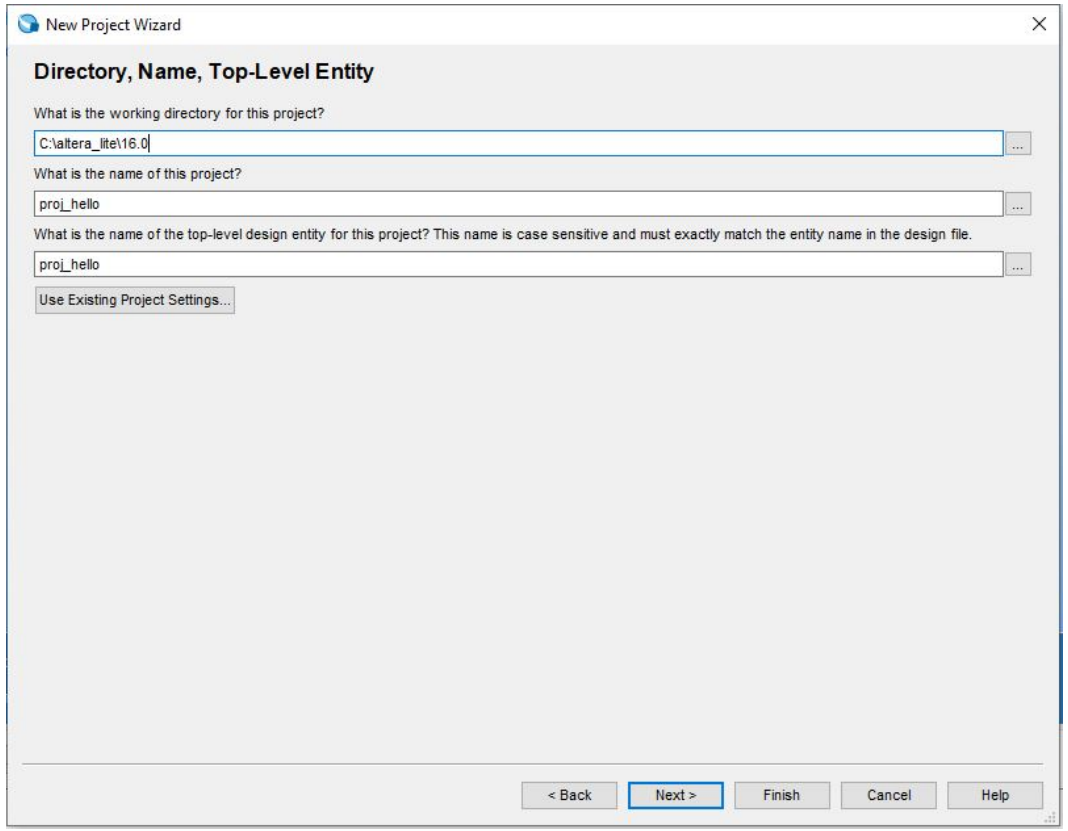
- Quartus Prime
  - FPGA development: Design, simulation, synthesis and download
- QSYS
  - Build target QSYS system
  - Based on a Processor-core (NIOS II processor), you can add other peripherals (e.g. Parallel IO, Ethernet Controller, Serial Ports, etc.)
- NIOS II Software Build Tools for Eclipse
  - Software platform for developing programs

Lab1 Objectives

- Part I :
  - Display “Hello, world” in your debug console from NIOS II
- Part 2 :
  - Implement a counter program and display the output on the LED, and the seven segment display.

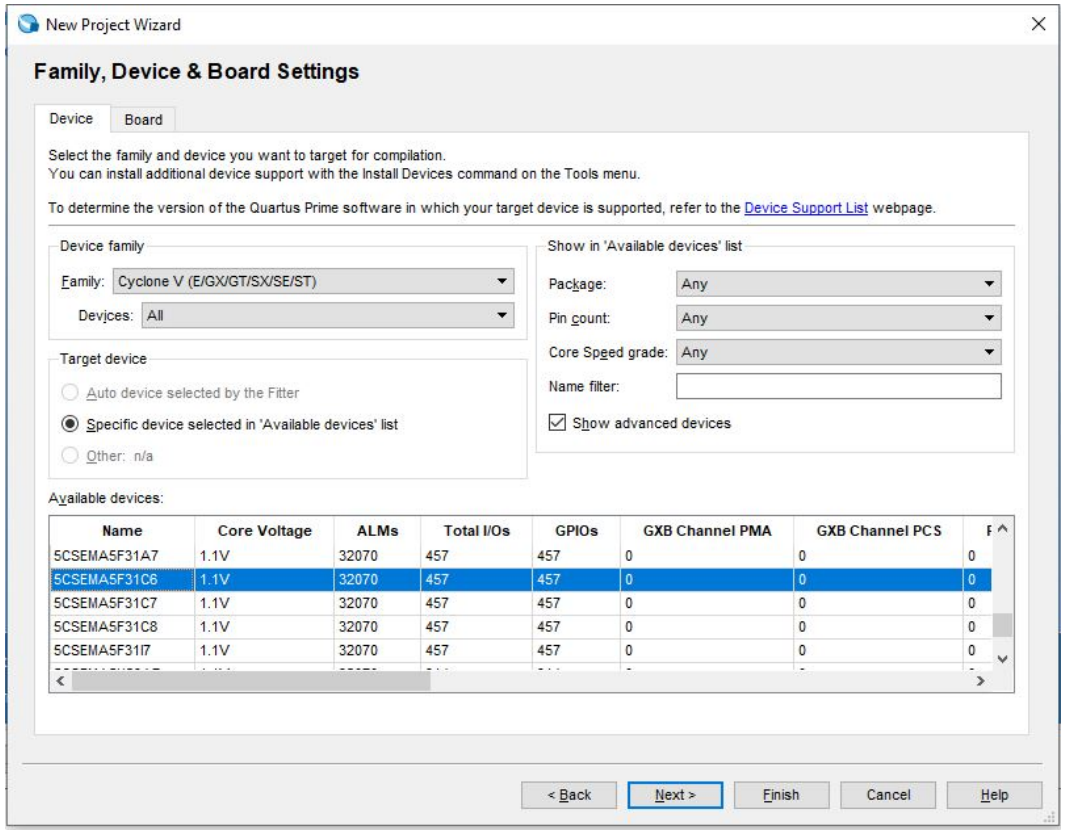
New Project Creation

- Open Quartus Prime, and select file -> New Project Wizard…
- Specify a working directory and name your project
- NOTE: Make sure your working directory contains no white-space. (This will cause problems later when trying to compile your project from the QSYS)



FPGA Selection

- Select Cyclone V in the drop-down box labeled “family”. We will be using device 5CSEMA5F31C6.
- Click finish. All necessary project parameters have been configured.



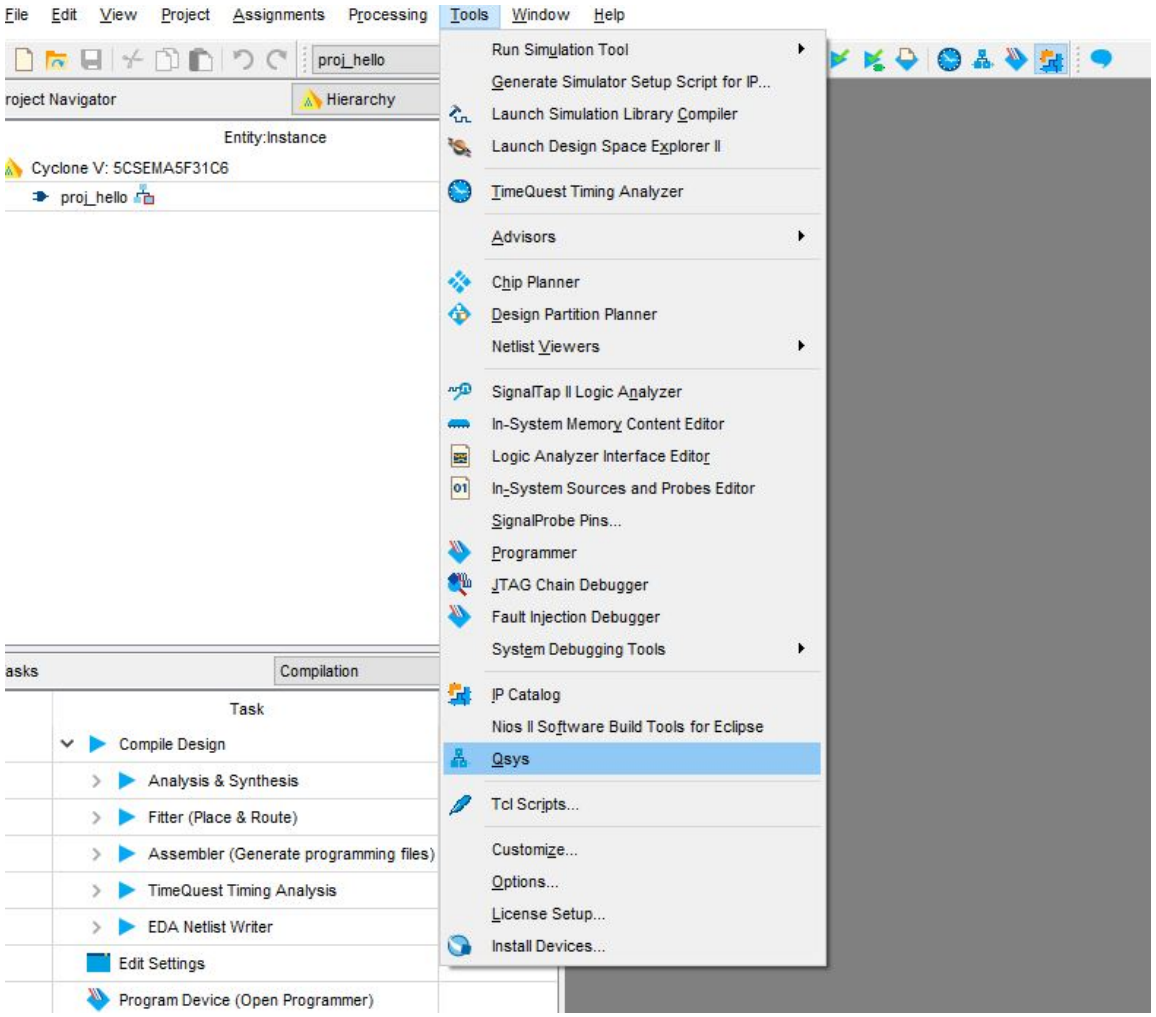
Using the QSYS System

- SOPC Builder (System on a Programmable Chip Builder) is software made by Altera that automates connecting IP cores to create a complete system that runs on any of its various FPGA chips.

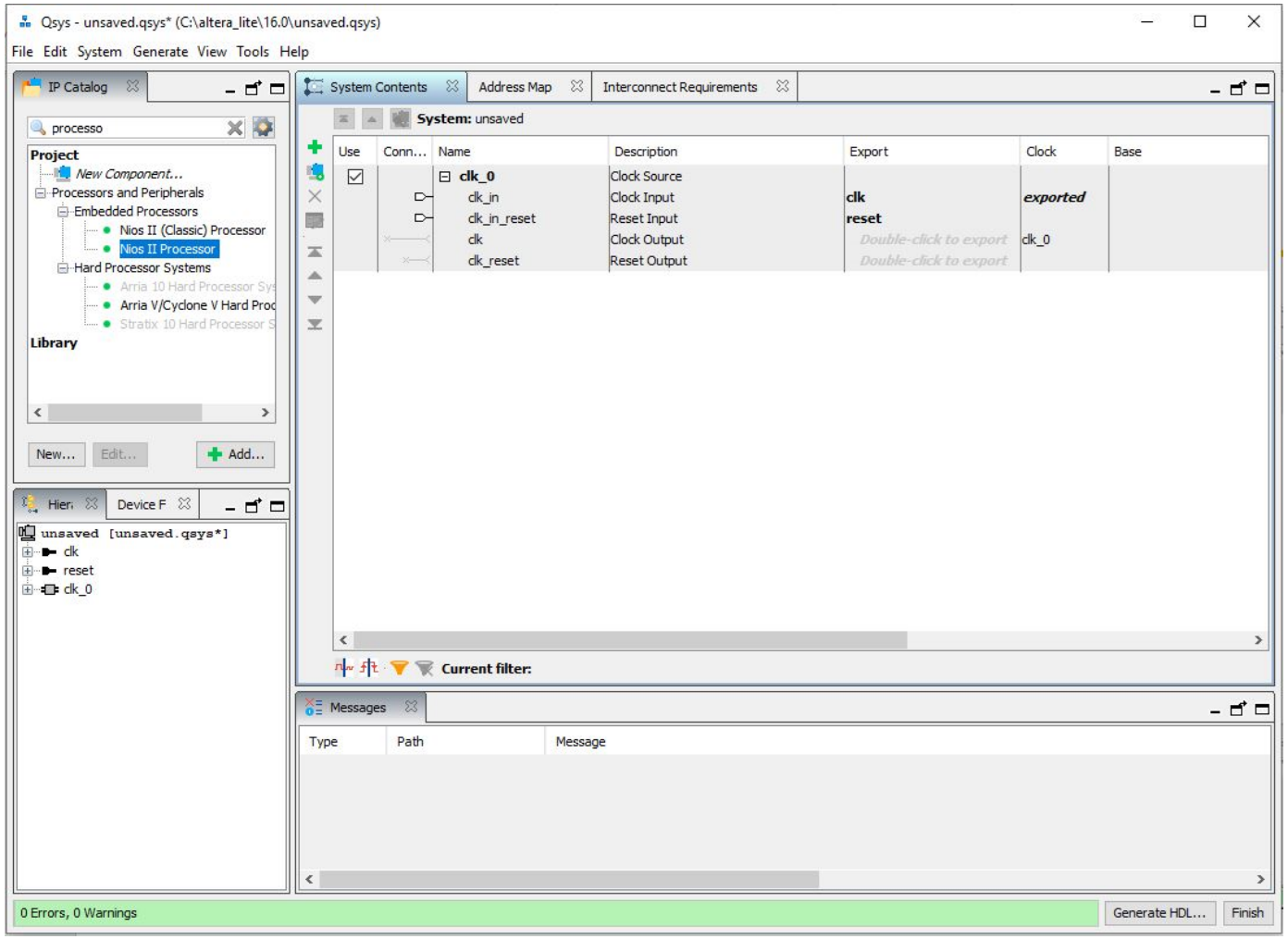
- Qsys is the next-generation SOPC Builder tool powered by a new FPGA-optimized network-on-a-chip (NoC) technology.

- Open Tools -> Qsys to start the system, save the system with the same name as the Top Level entity.

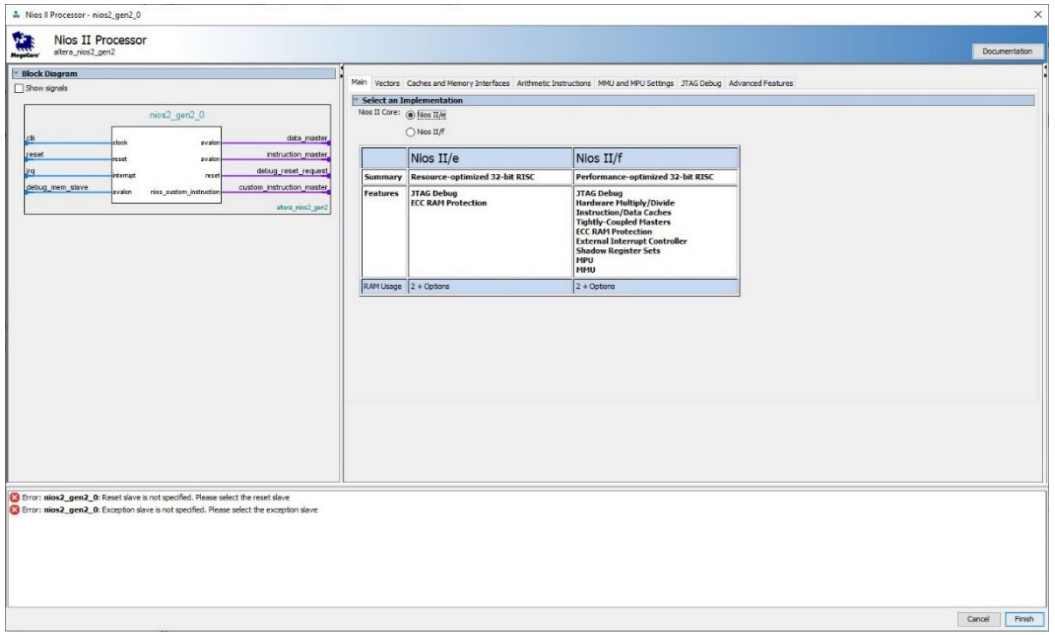
Launching QSYS

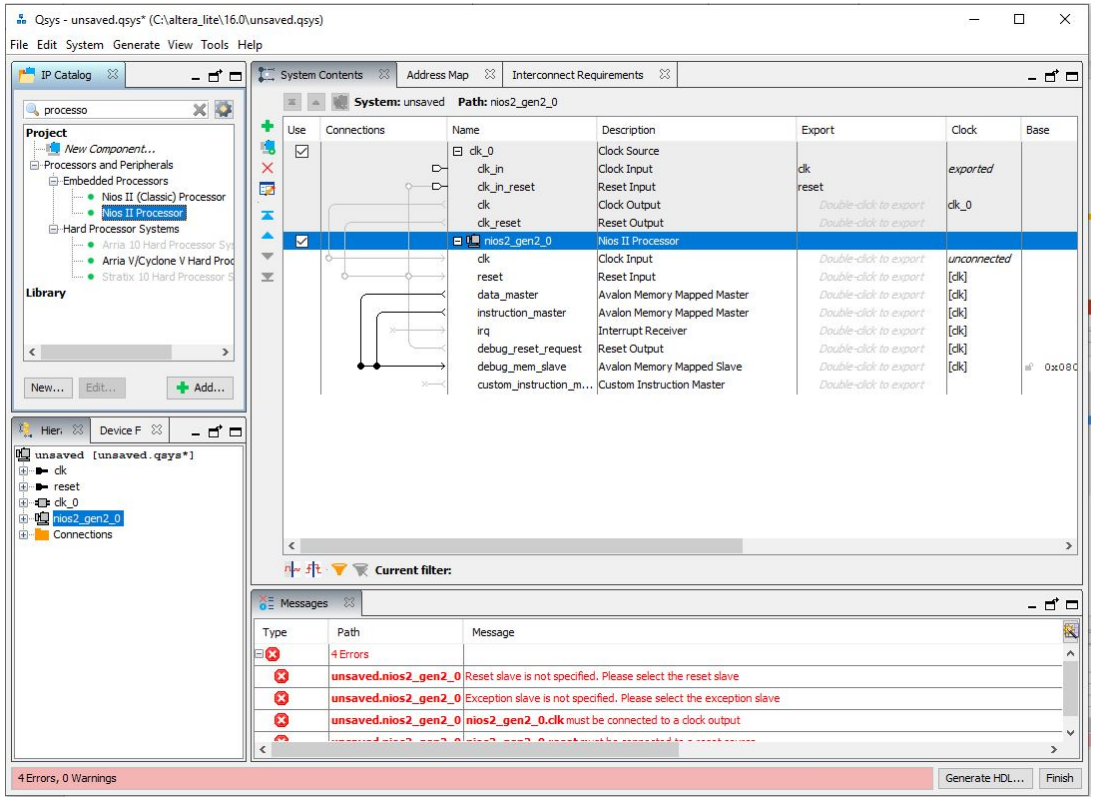


QSYS Screenshot



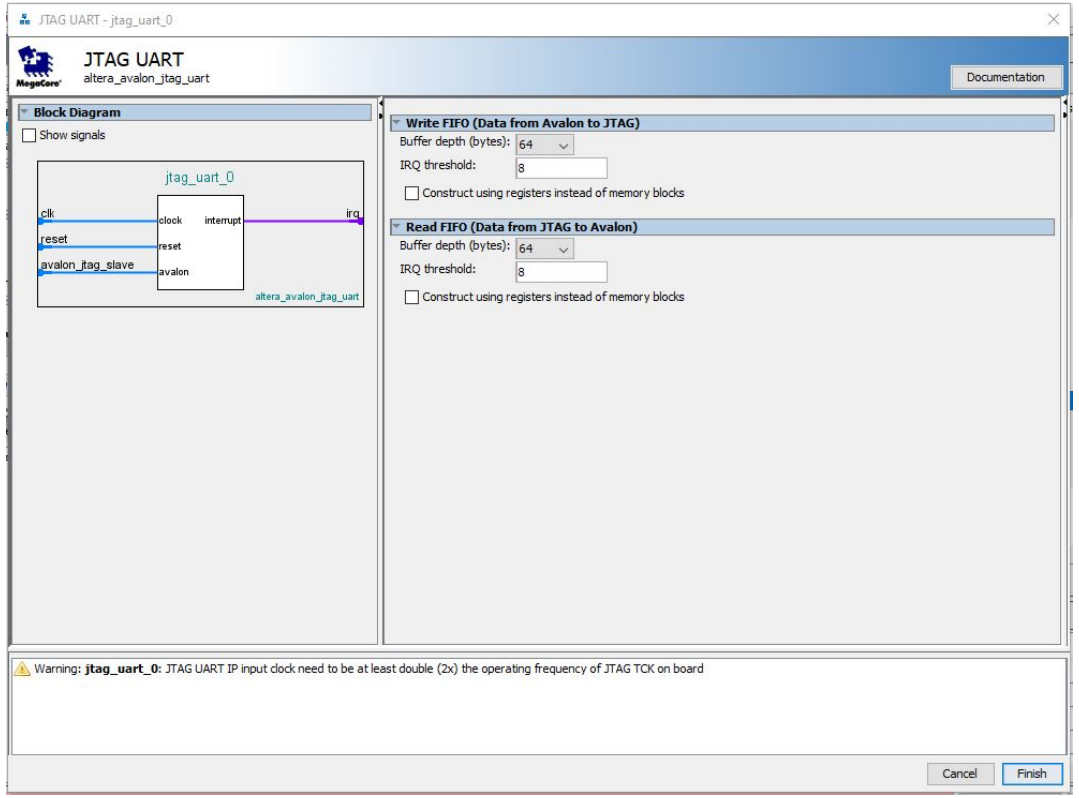
Adding NIOS II Processor

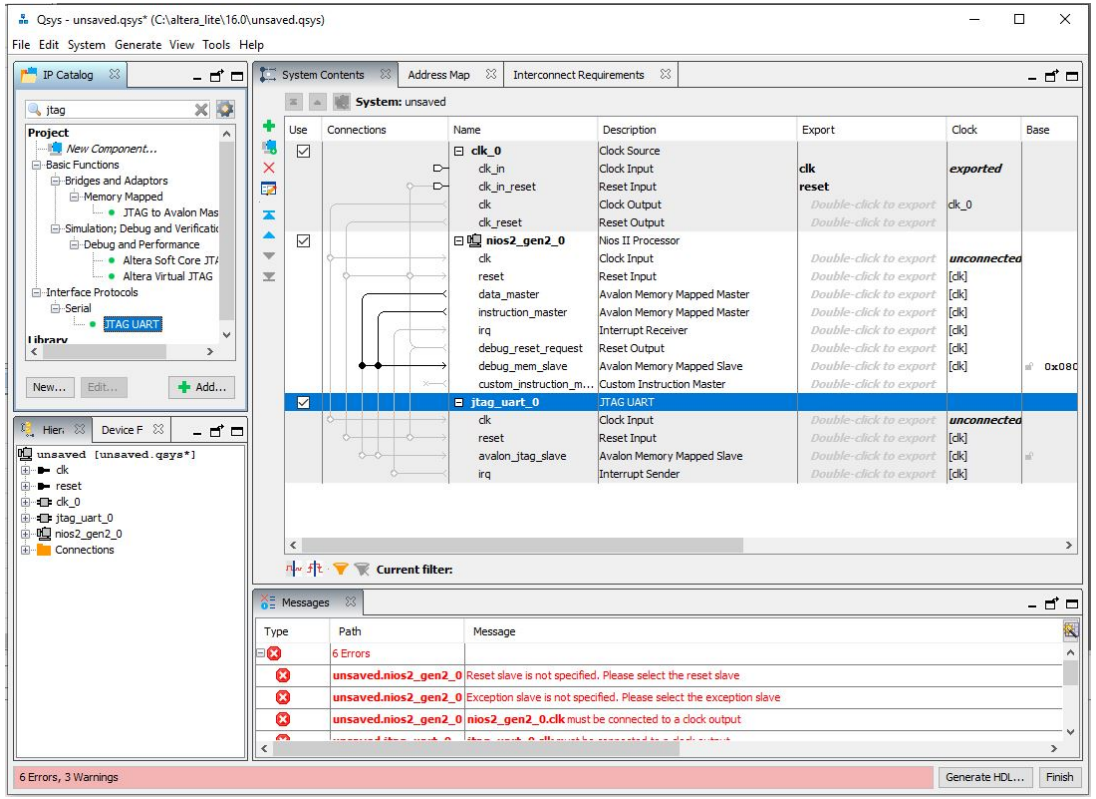




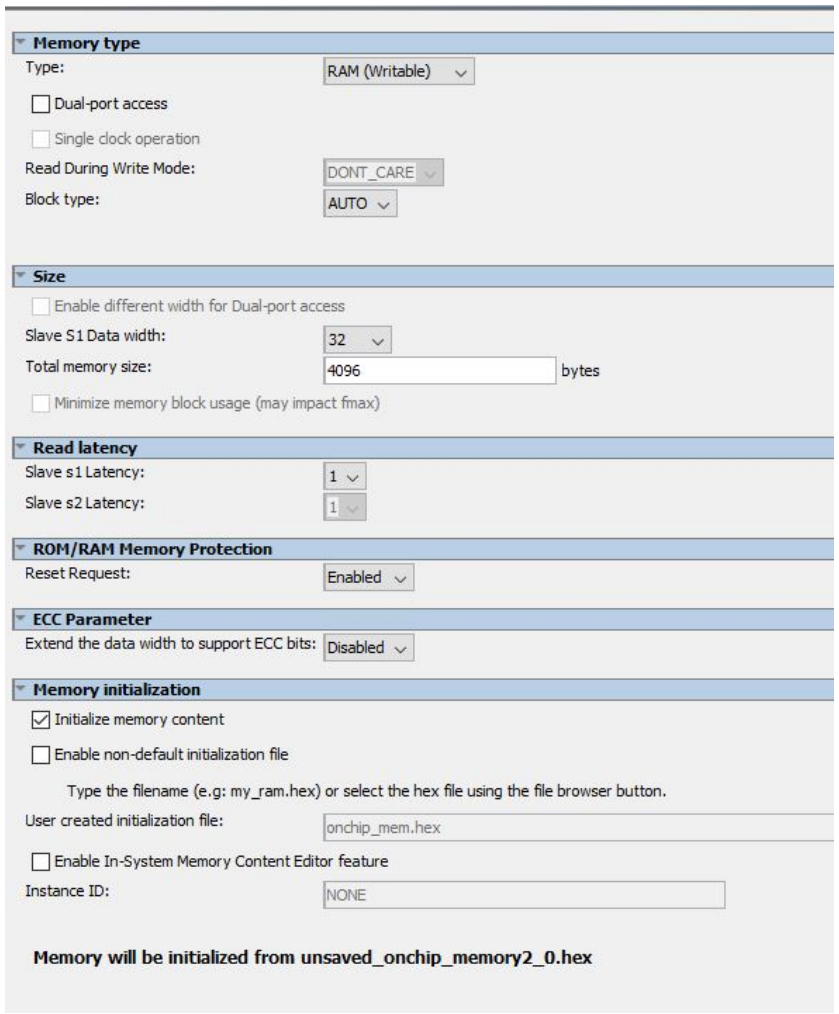
Adding JTAG UART

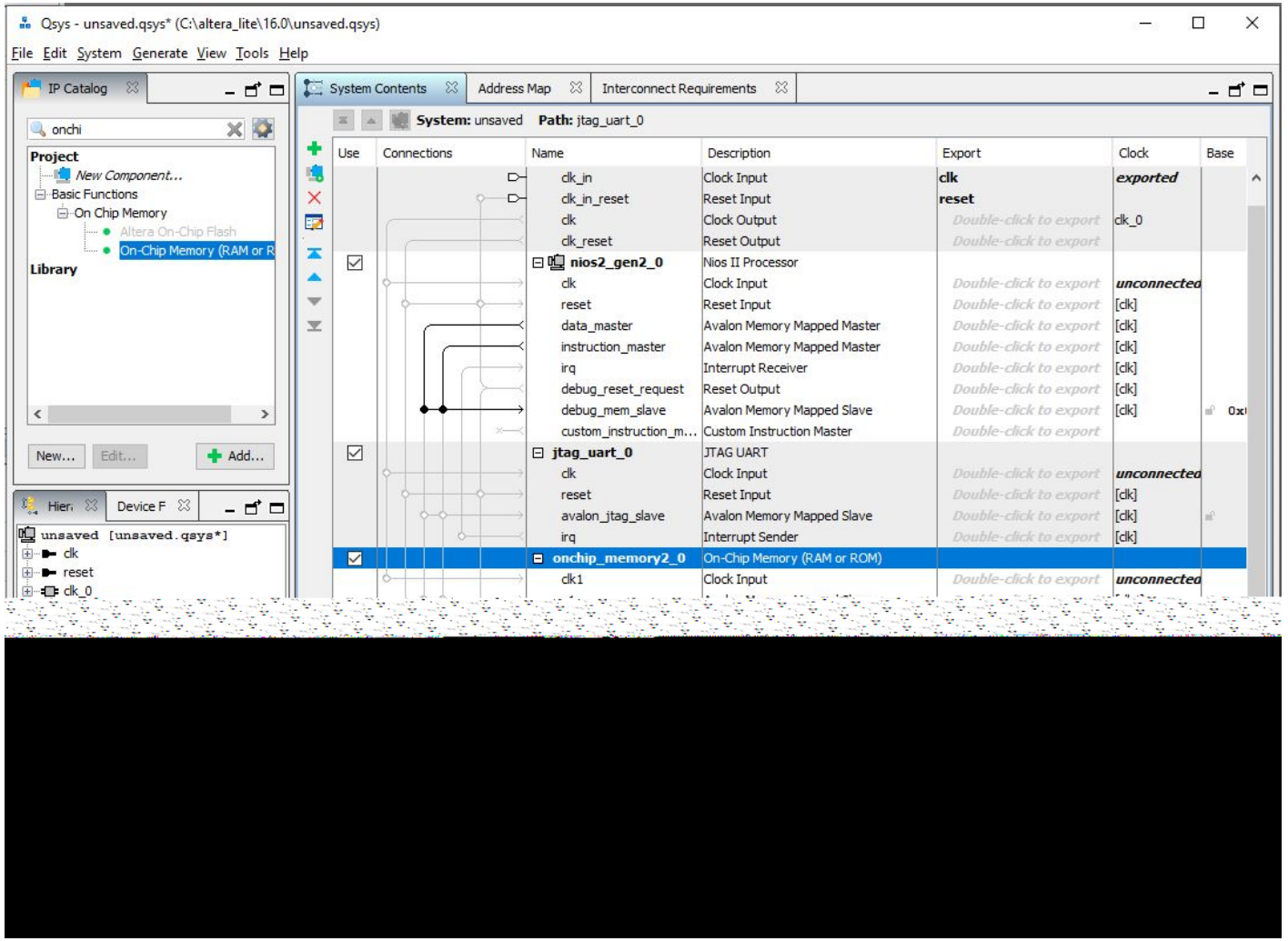
- Next, we will be adding a JTAG UART module, found under Serial.



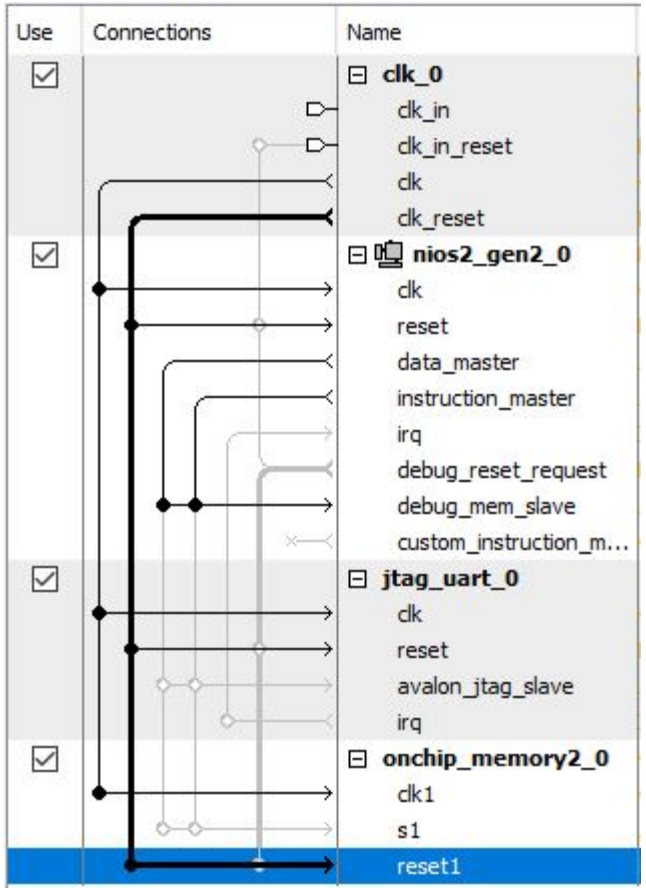


Adding On -Chip Memory

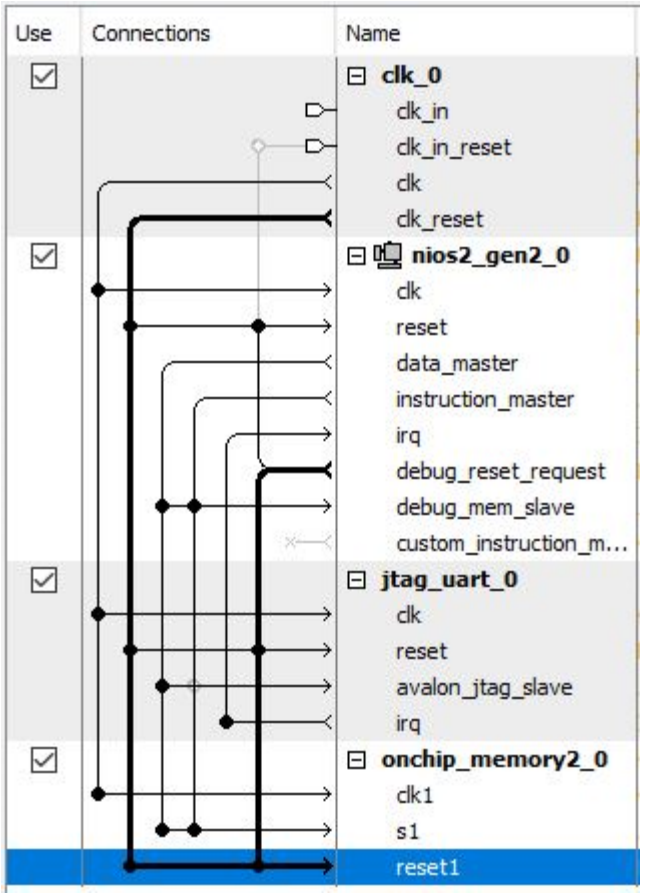




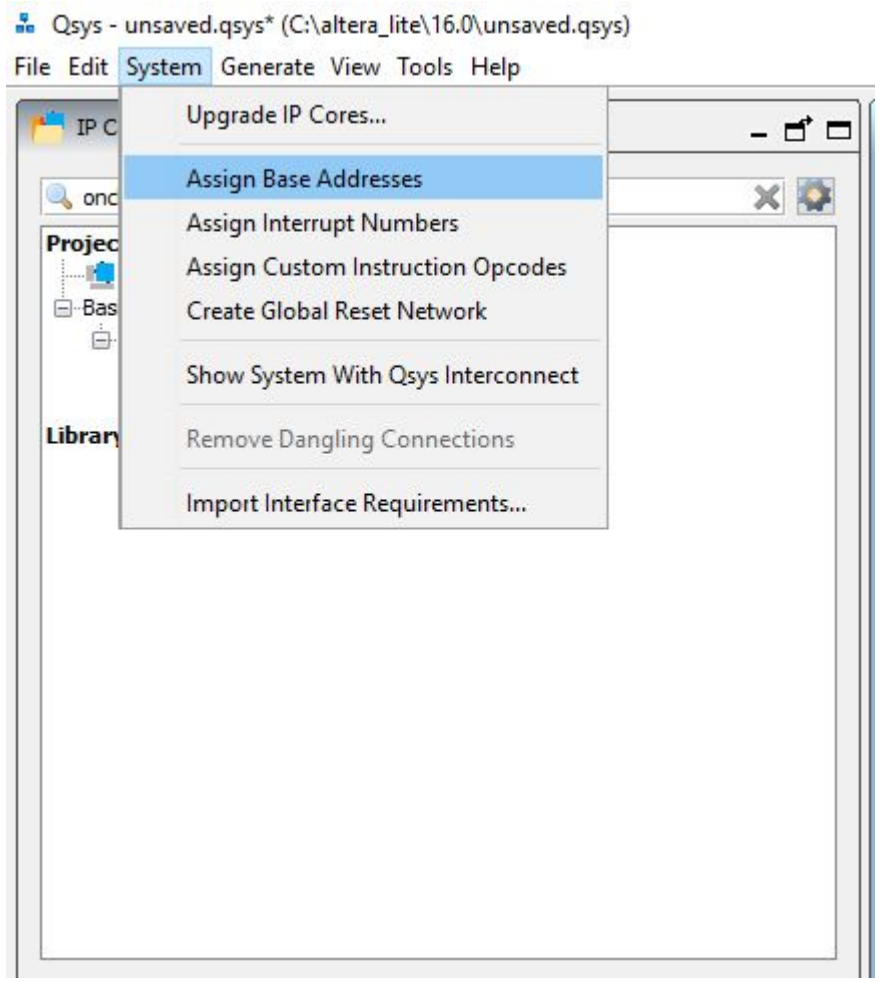
Clock and Reset Connections

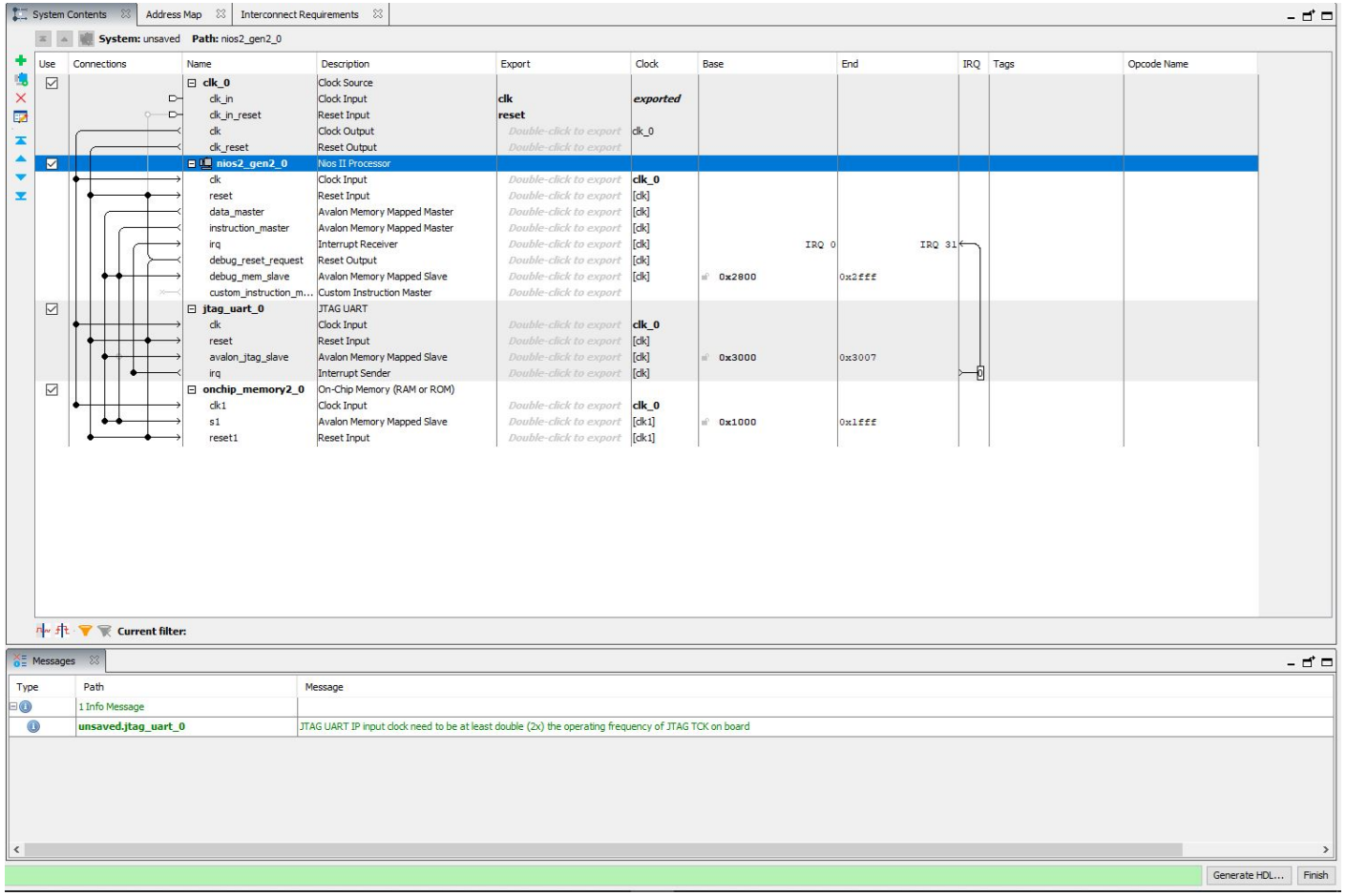


Other Connections

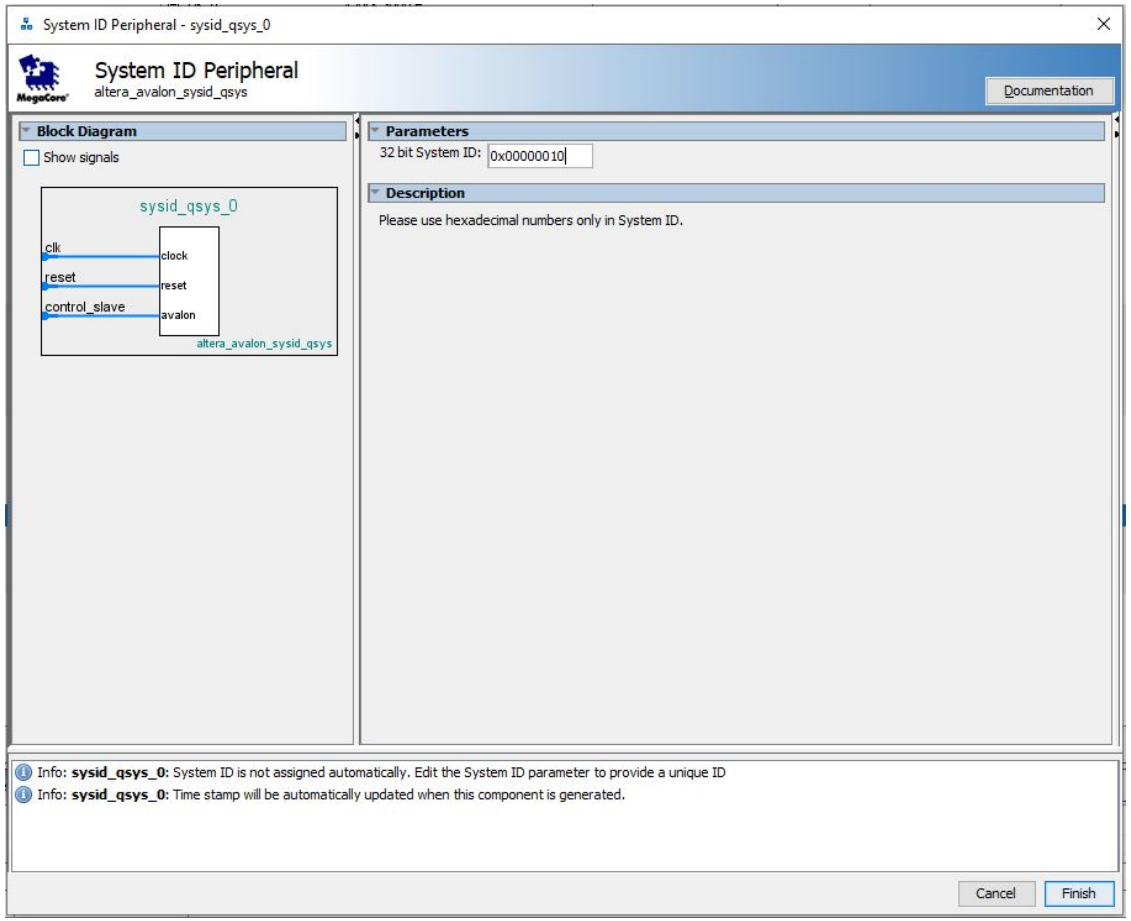


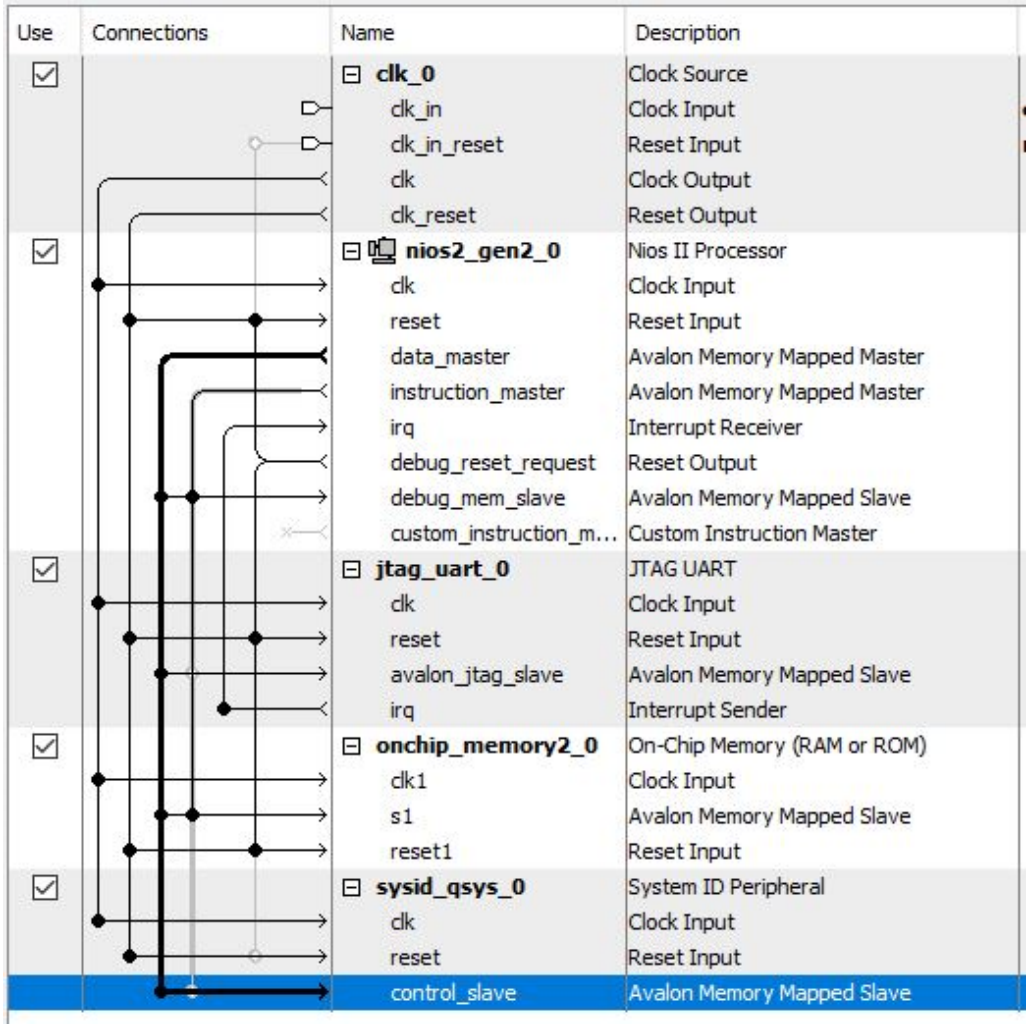
Assigning Reset and Exception Vectors



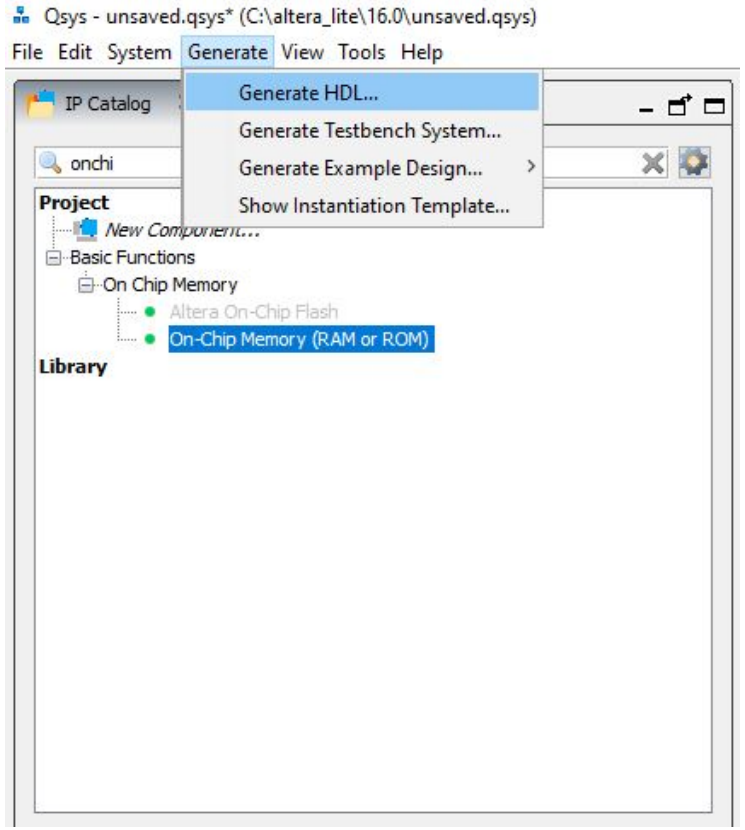


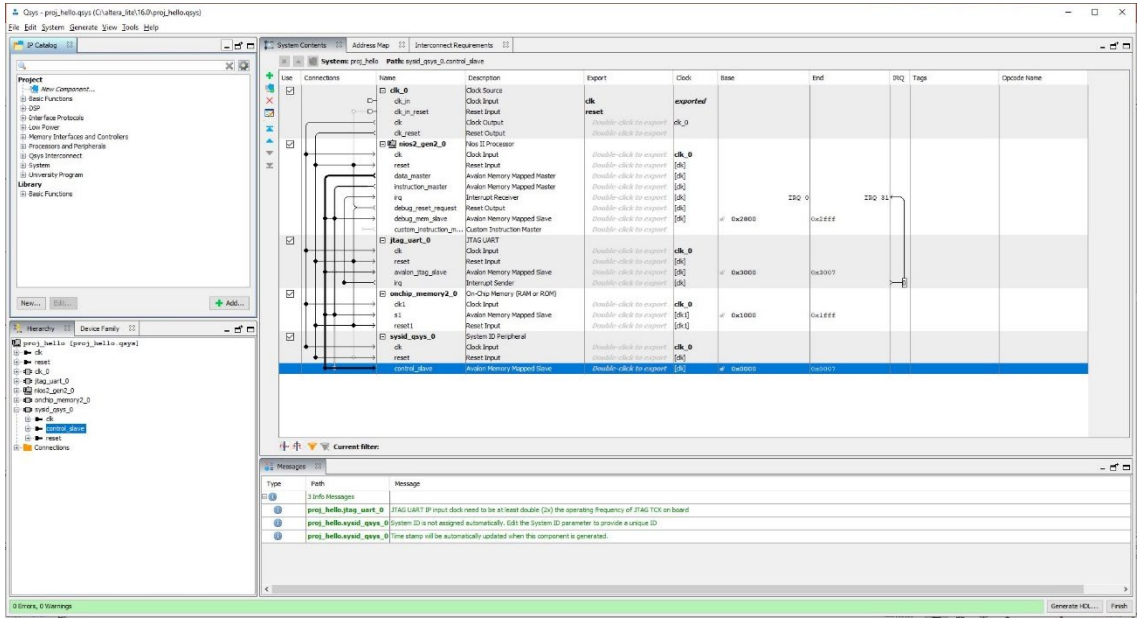
Adding system ID





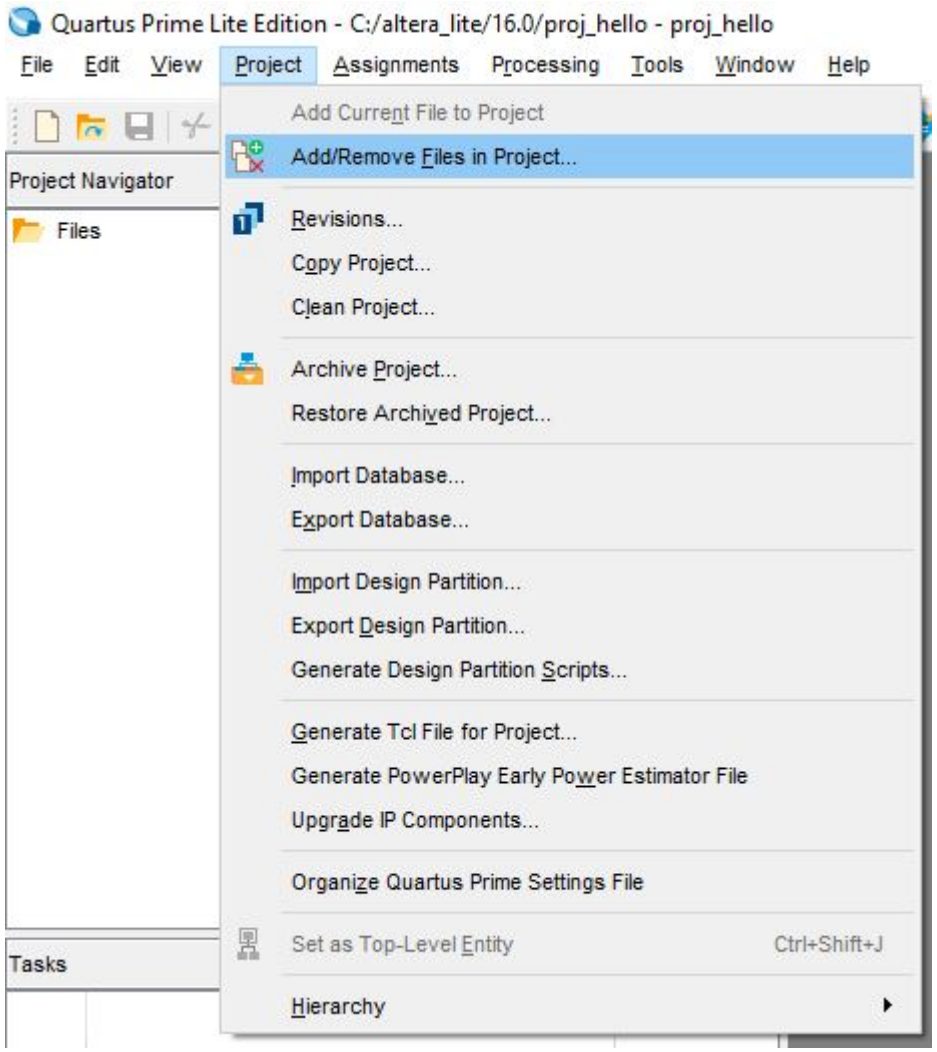
Generating HDL



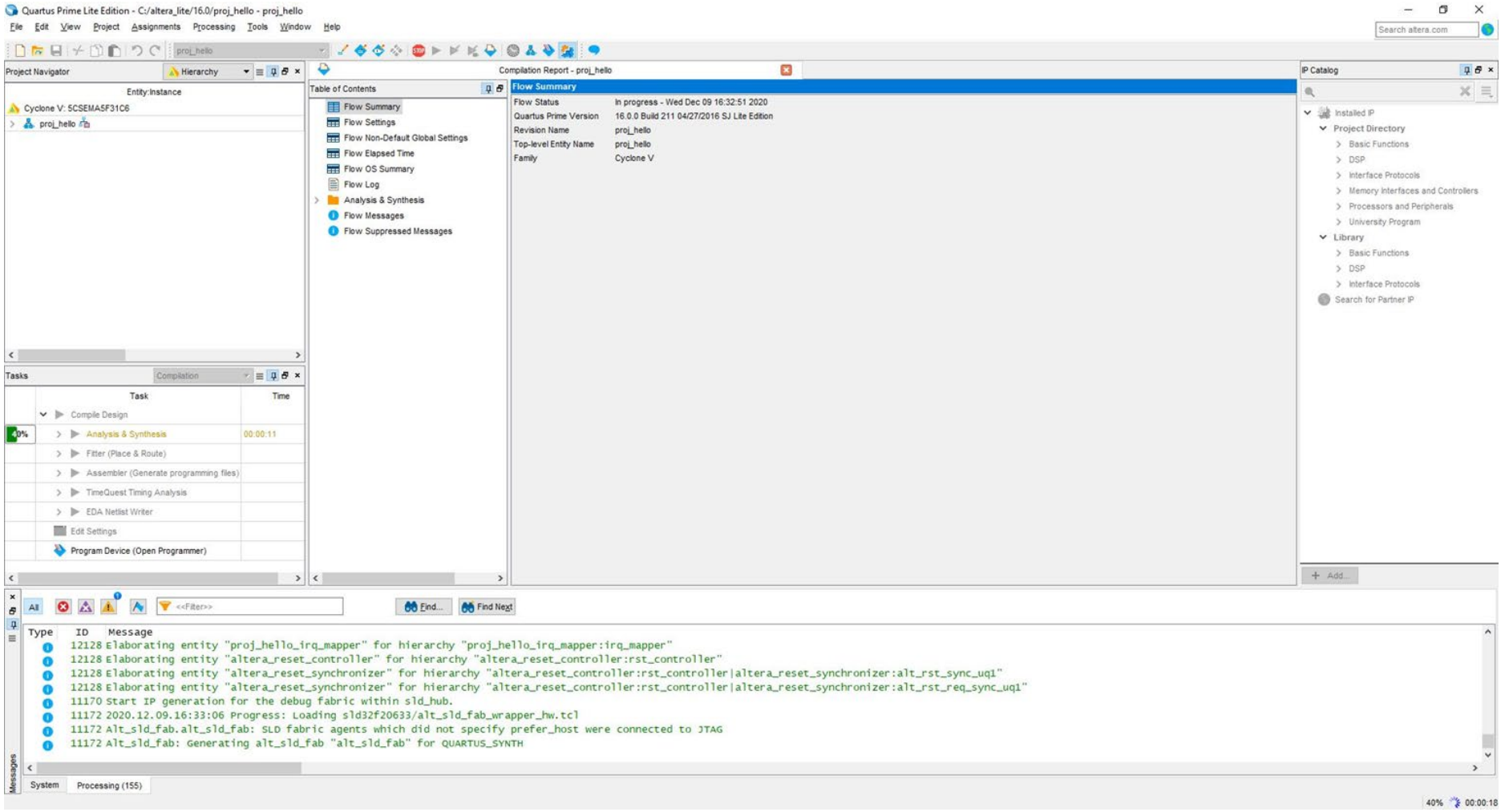


Tutorial: Programming the FPGA

- Back to Quartus Prime
  - Add the QIP file to the project in Quartus, it contains the Verilog files generated by QSYS
  - Now select Processing -> start -> Start Analysis & Synthesis.
  - A number of warnings will appear during the check. This is normal.



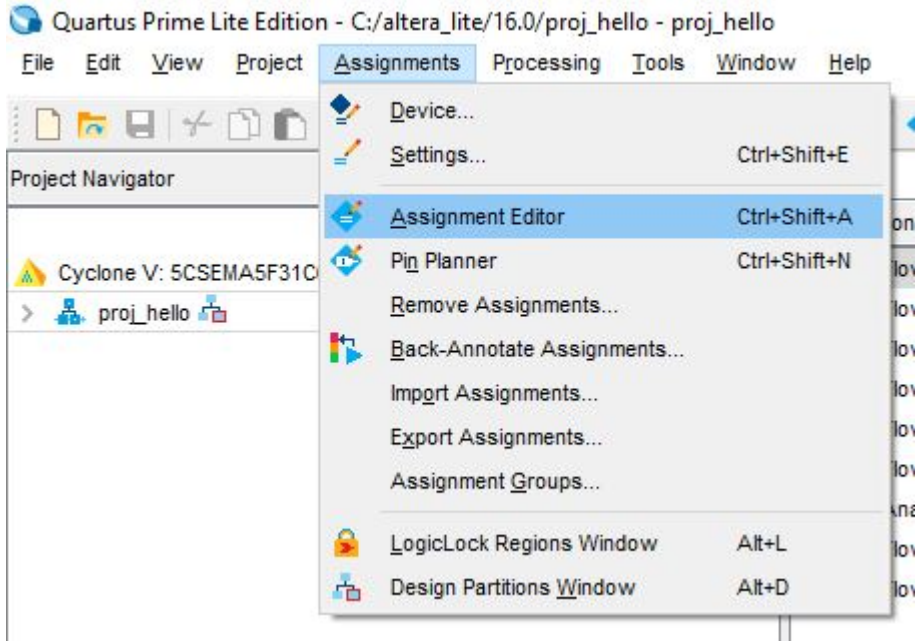
Quartus: Analysis & Synthesis



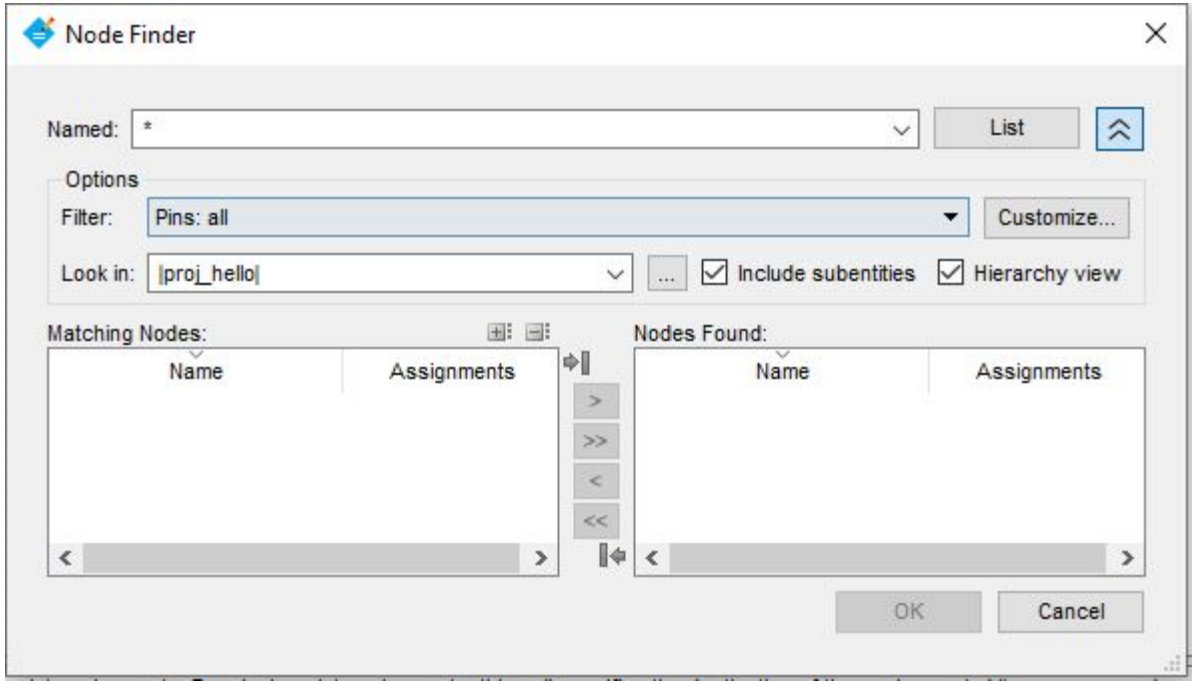
Assigning CLK and Reset Pins

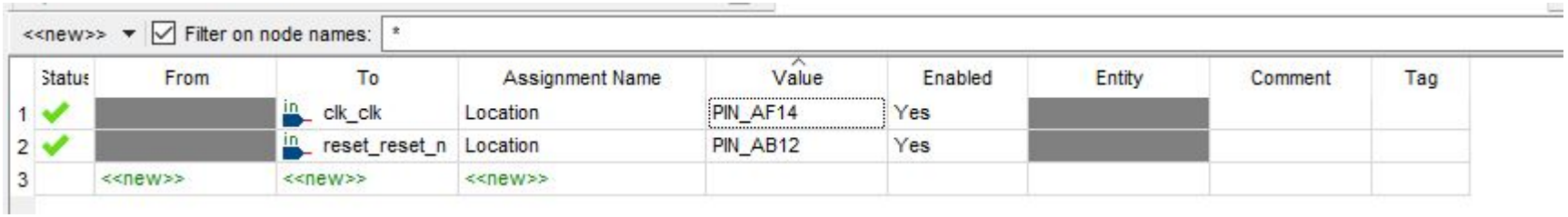
- The pins assignments can be done using Assignment Editor in Assignments -> Assignment Editor
- Find the entry corresponding to the clock, assign PIN_AF14 to it. doing so will connect the clock in our design to the 50 MHz oscillator on the DE1-SoC board.
- Now assign the reset input signal to any of the SW[x] pins. This will connect the reset signal to one of the switches found on the DE1-SoC board.
- Now select Processing -> Start Compilation. If compilation is successful, a programming file to be written to the FPGA will be generated.

Assigning CLK and Reset Pins

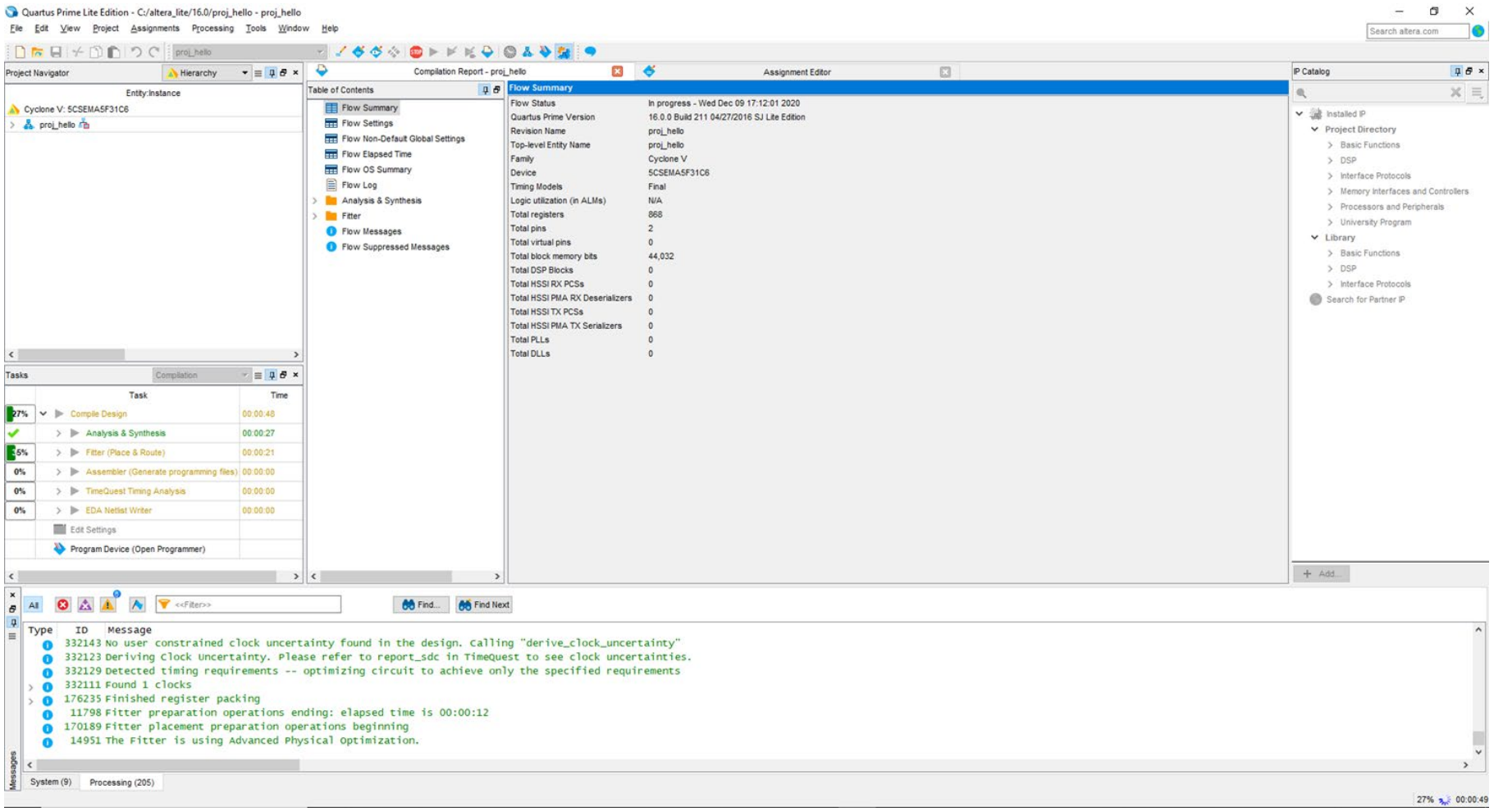


Assigning CLK and Reset Pins





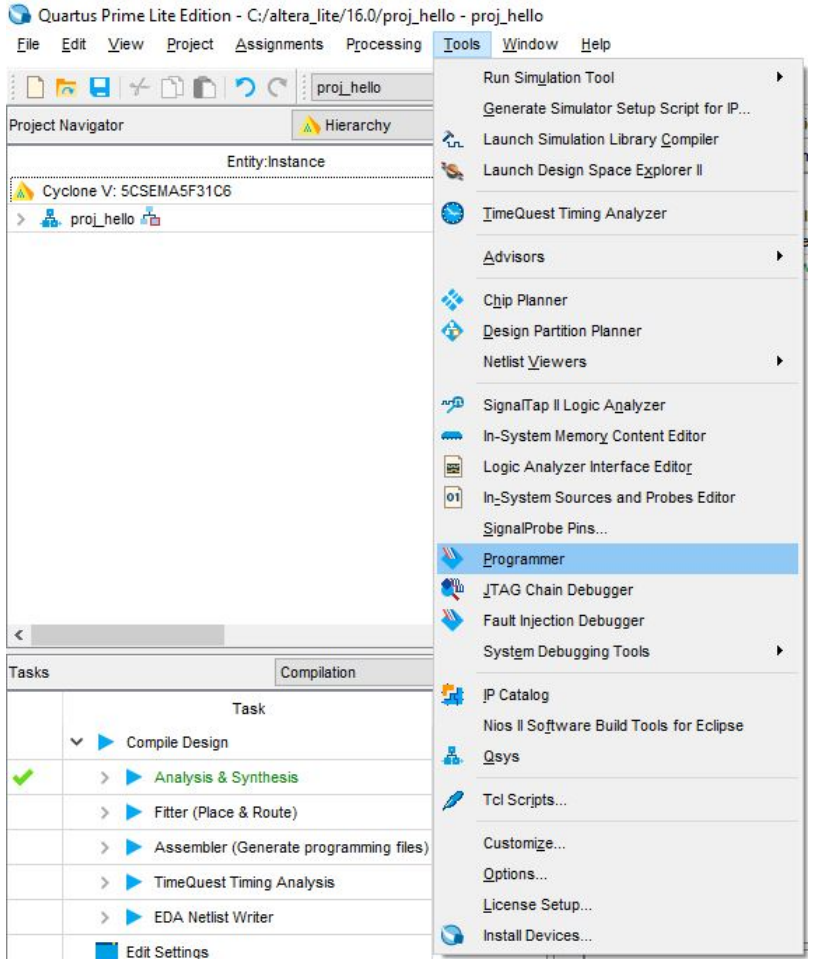
Compiling

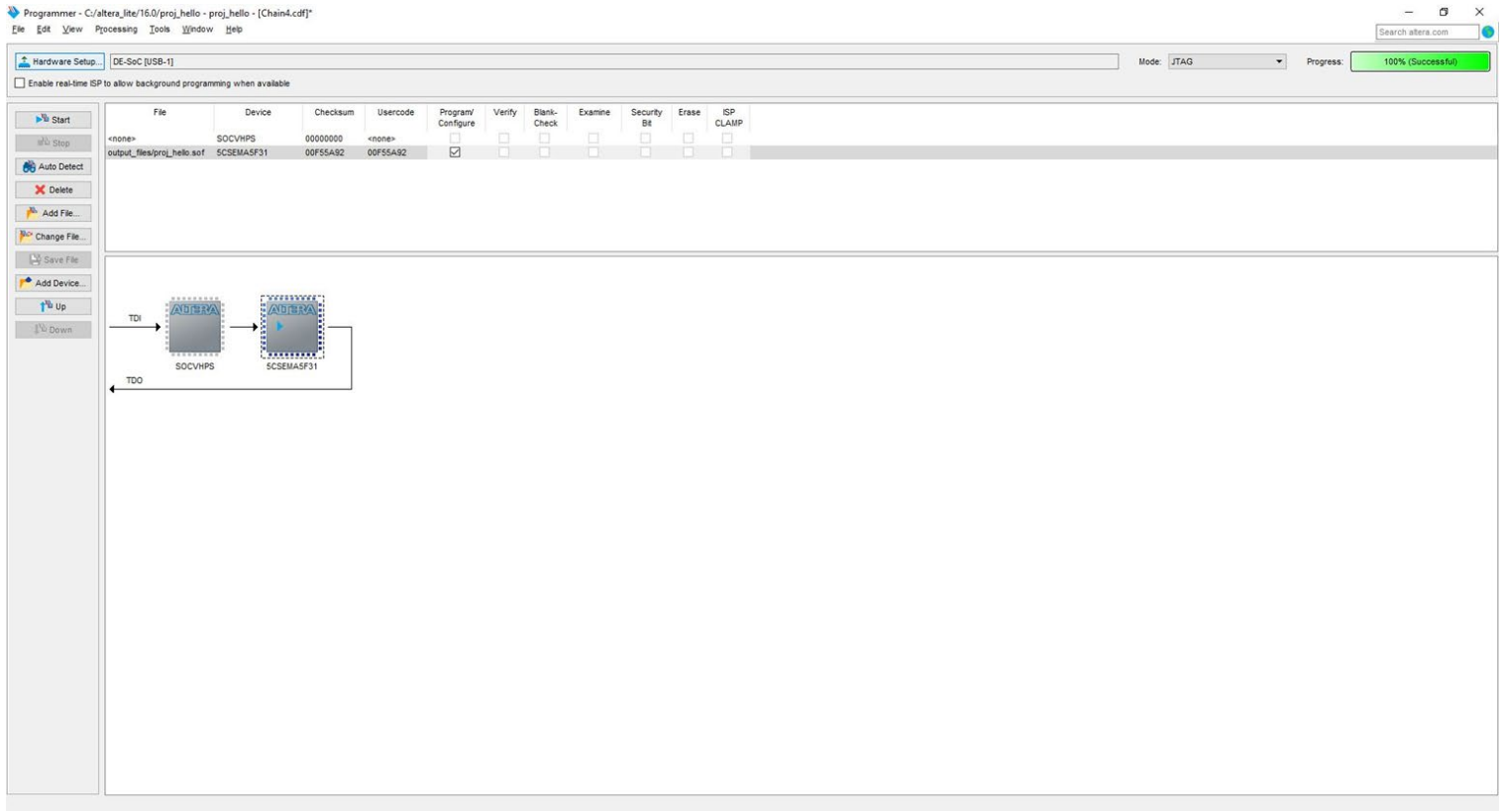

Quartus Prime Lite Edition - C:/altera_lite/16.0/proj_hello - proj_hello
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Hierarchy
Entity:Instance
Cyclone V: 5CSEMA5F31C6
proj_hello
Compilation Report - proj_hello
Assignment Editor
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global Settings
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Flow Messages
Flow Suppressed Messages
Flow Summary
Flow Status In progress - Wed Dec 09 17:12:01 2020
Quartus Prime Version 16.0.0 Build 211 04/27/2016 SJ Lite Edition
Revision Name proj_hello
Top-level Entity Name proj_hello
Family Cyclone V
Device 5CSEMA5F31C6
Timing Models Final
Logic utilization (in ALMs) N/A
Total registers 868
Total pins 2
Total virtual pins 0
Total block memory bits 44,032
Total DSP Blocks 0
Total HSSI RX PCSs 0
Total HSSI PMA RX Deserializers 0
Total HSSI TX PCSs 0
Total HSSI PMA TX Serializers 0
Total PLLs 0
Total DLLs 0
IP Catalog
Installed IP
Project Directory
Basic Functions
DSP
Interface Protocols
Memory Interfaces and Controllers
Processors and Peripherals
University Program
Library
Basic Functions
DSP
Interface Protocols
Search for Partner IP
Tasks
Compilation
Task Time
Compile Design 00:00:48
Analysis & Synthesis 00:00:27
Fitter (Place & Route) 00:00:21
Assembler (Generate programming files) 00:00:00
TimeQuest Timing Analysis 00:00:00
EDA Netlist Writer 00:00:00
Edit Settings
Program Device (Open Programmer)
Type ID Message
332143 No user constrained clock uncertainty found in the design. Calling "derive_clock_uncertainty"
332123 Deriving Clock Uncertainty. Please refer to report_sdc in Timequest to see clock uncertainties.
332129 Detected timing requirements -- optimizing circuit to achieve only the specified requirements
332111 Found 1 clocks
176235 Finished register packing
11798 Fitter preparation operations ending: elapsed time is 00:00:12
170189 Fitter placement preparation operations beginning
14951 The Fitter is using Advanced Physical Optimization.
System (9)
Processing (205)
27% 00:00:49

Programming the FPGA

- Select Tools -> Programmer. Now, select Auto detect, a new dialogue box opens up, choose “5CSEMA5”, you can find two entries, Right click on the 5CSEMA5 entry, change the file to the target SOF file. Be sure that DE1 SoC is connected to the computer via the provided USB cable.

Programming the FPGA

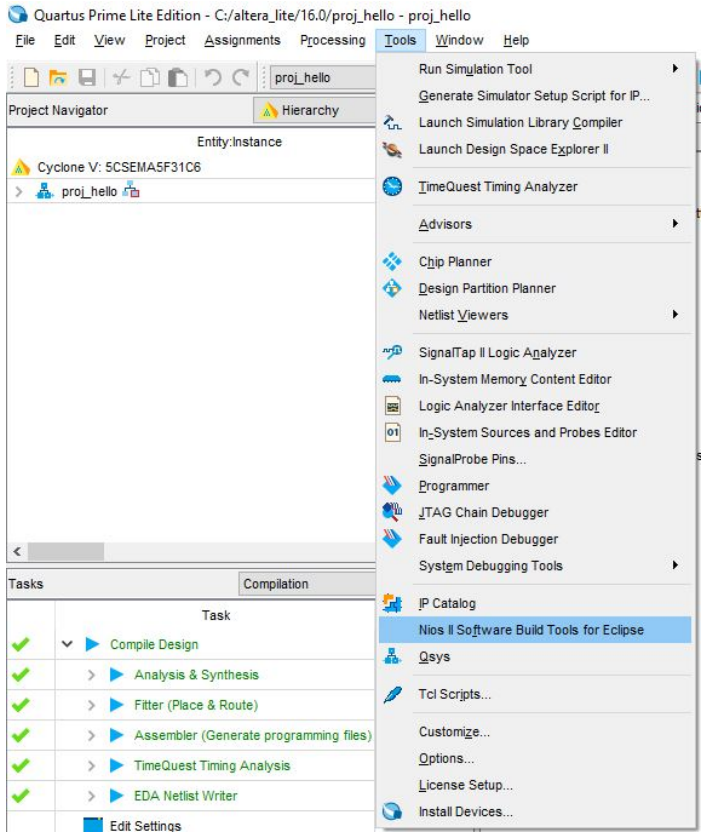


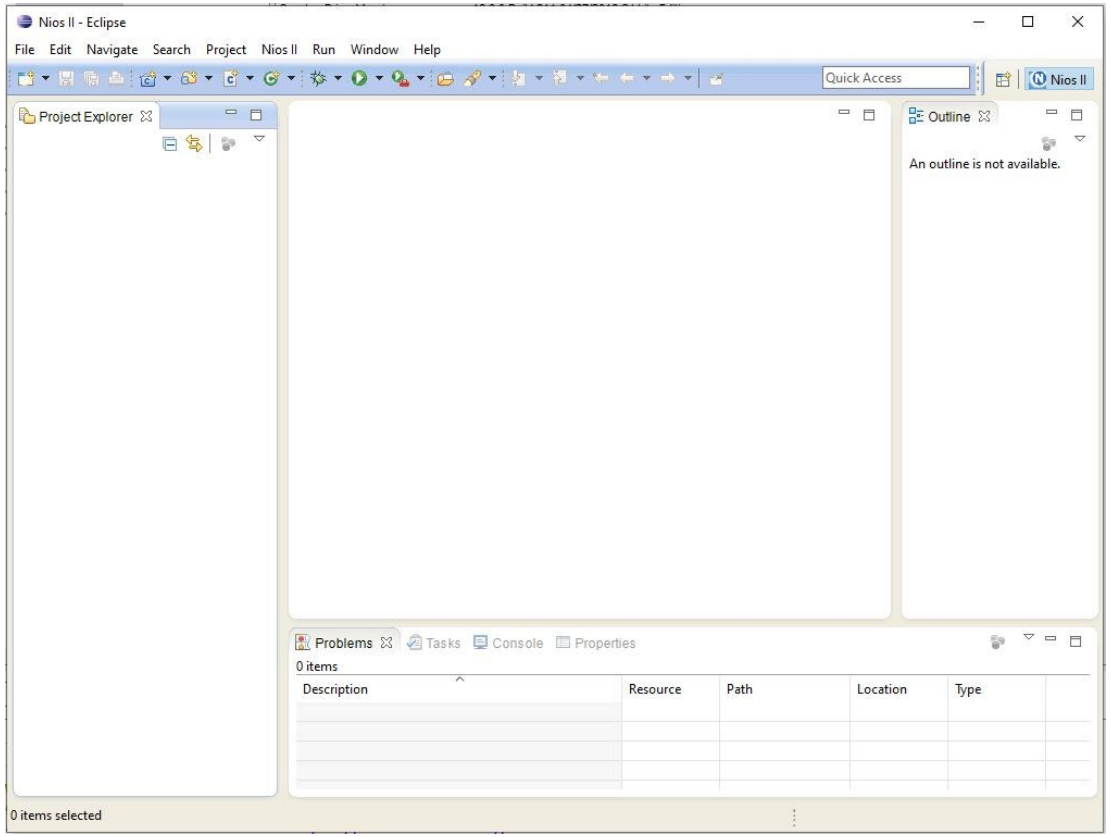


Tutorial: Using the NIOS II IDE

- Now that you have successfully developed an QSYS system, we will write a very simple program to run on our new processor.
- Select Tools -> NIOS II Software Build Tools for Eclipse.
- Create a new workspace. Make sure that there are no spaces in path directory names.
- Disable Anti-Virus for NIOS to detect connected board
- Make sure that your reset switch is logic HIGH (Reset Input is active when it is low)

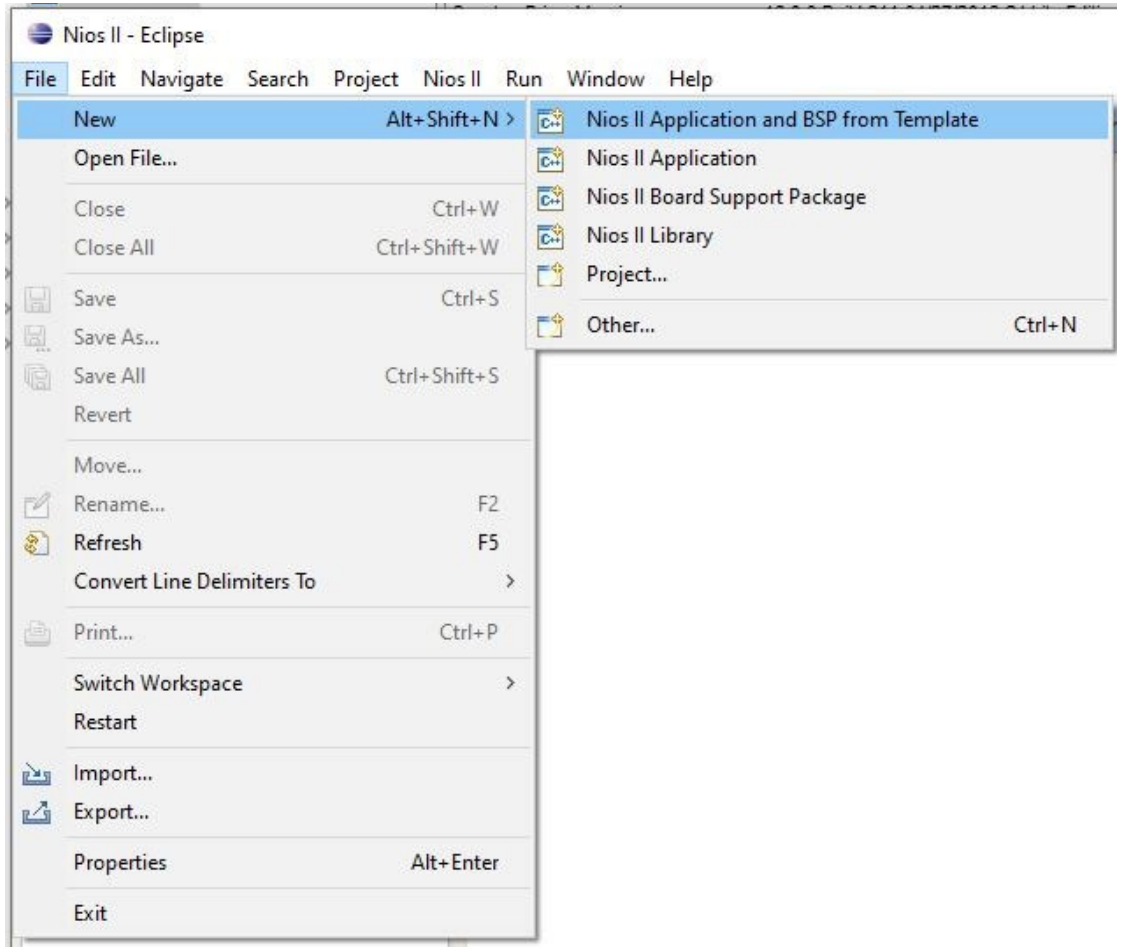
Tutorial: Using the NIOS II IDE



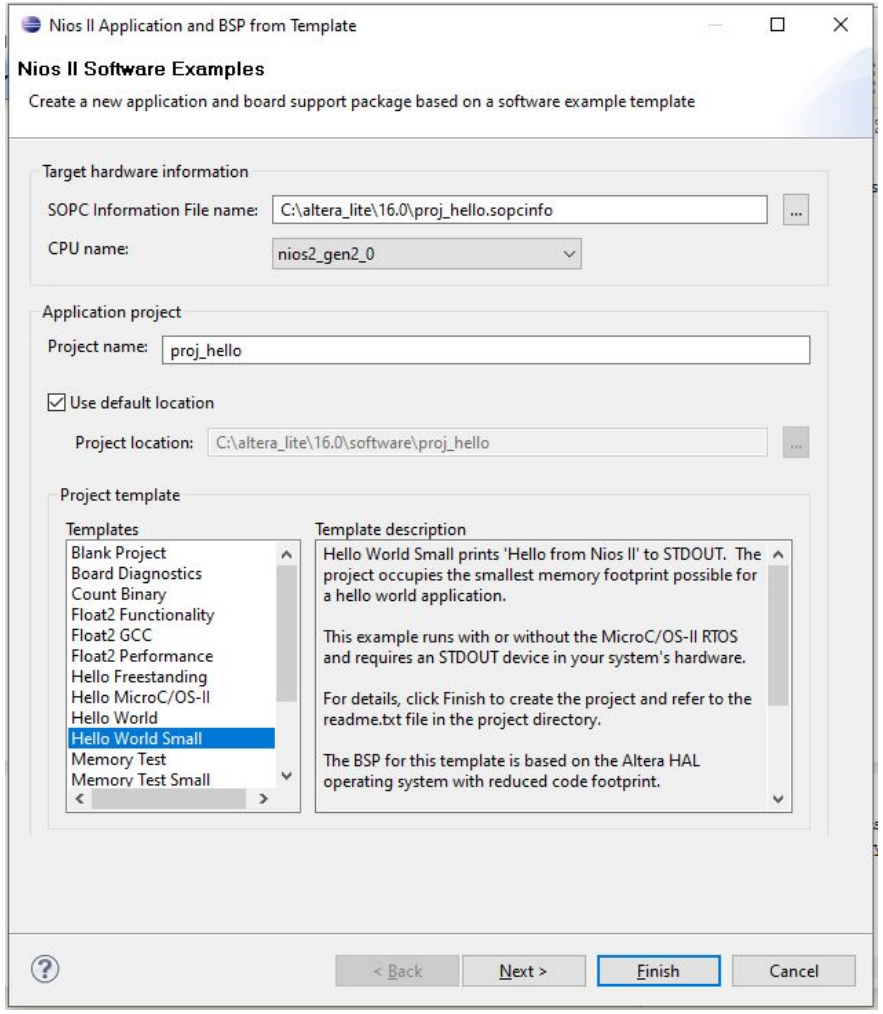


Tutorial: Using the NIOS II IDE

- Select File -> New -> NIOS II Application and BSP from template. A new dialogue box should open.
- A variety of project templates are provided as a starting point. We will be using the “Hello World Small” template.
- In order to develop a NIOS project, the IDE needs a .sopcinfo file that indicates various peripherals unique to our design.

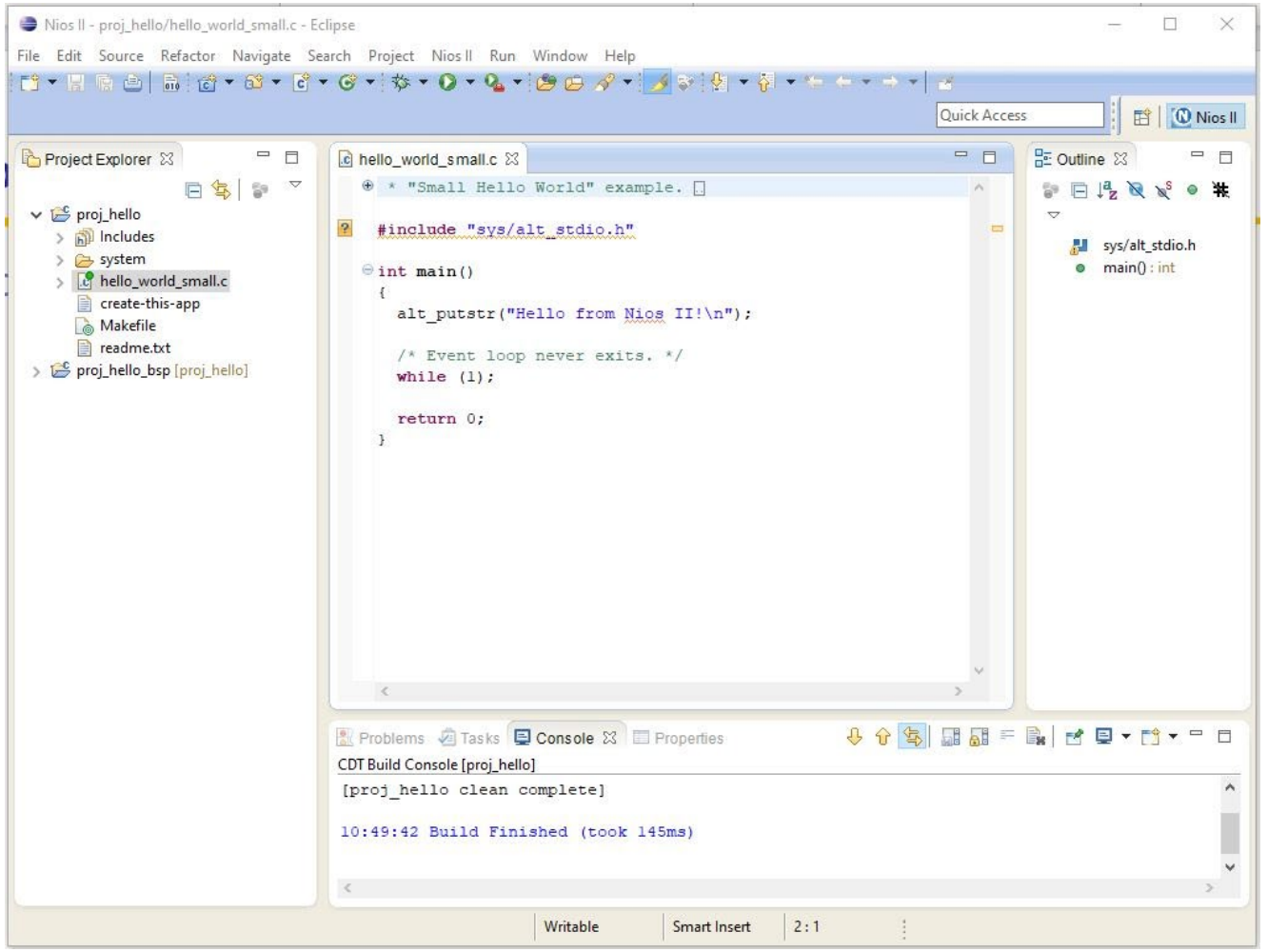


Tutorial: Using the NIOS II IDE



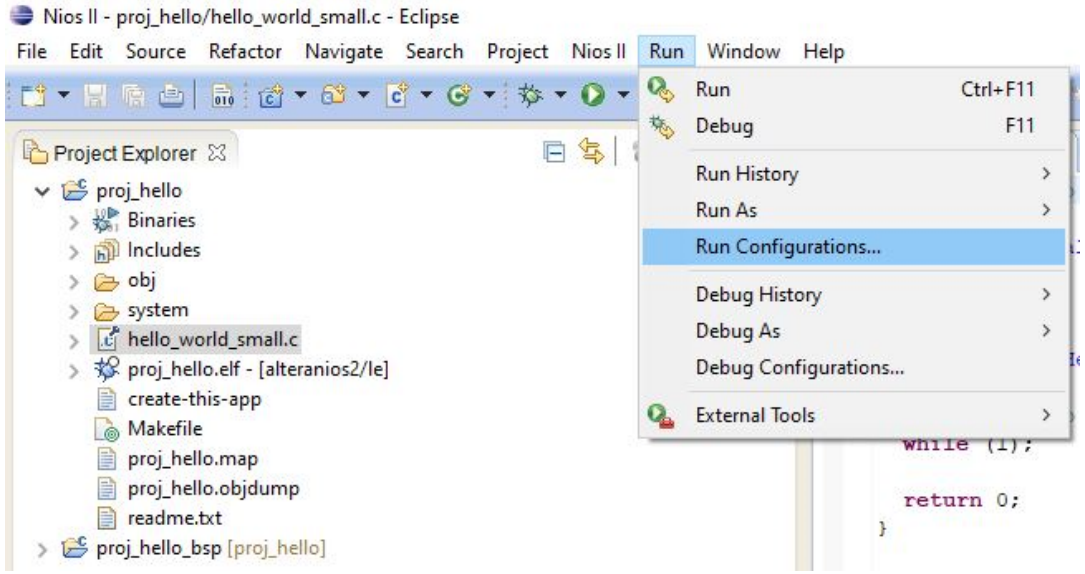
Tutorial: Using the NIOS II IDE

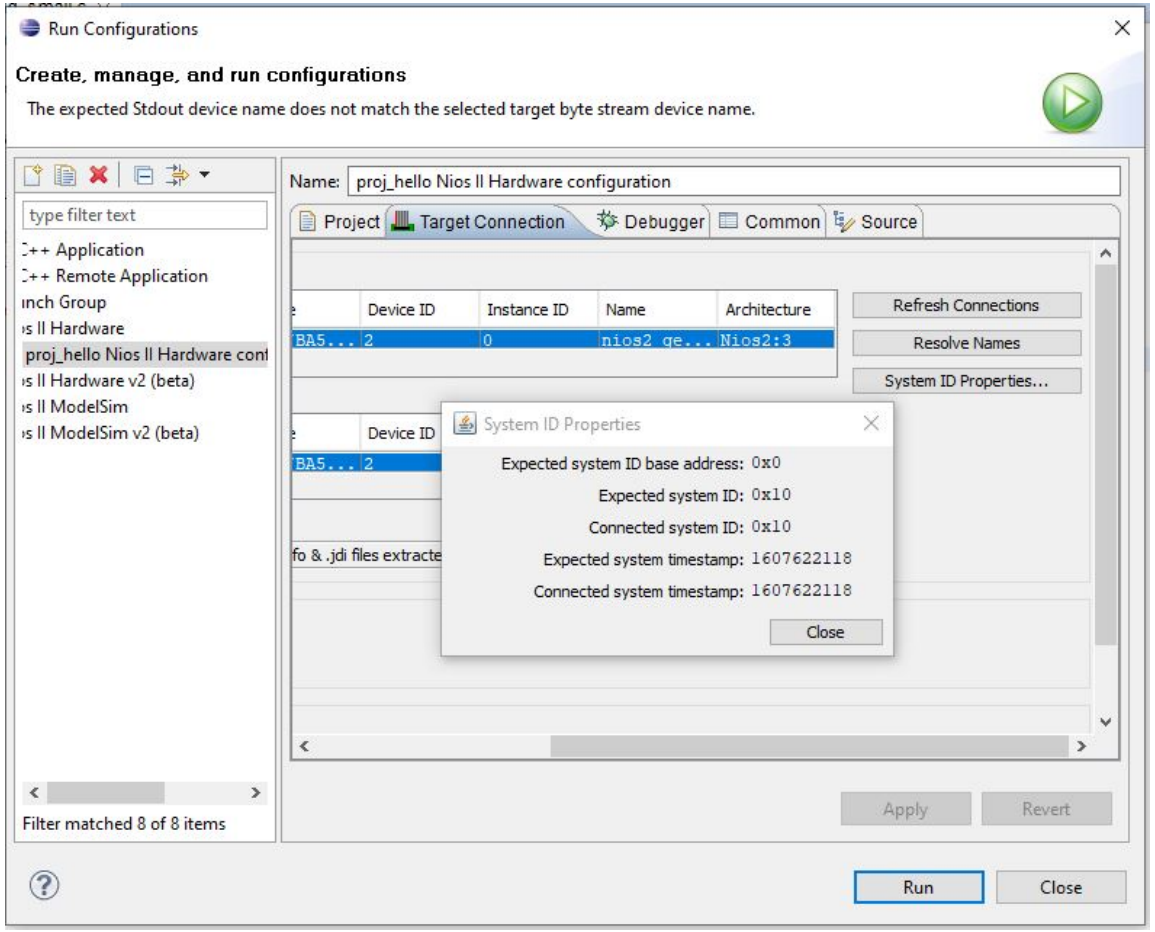
- You should now be looking at a window like this:



Tutorial: Using the NIOS II IDE

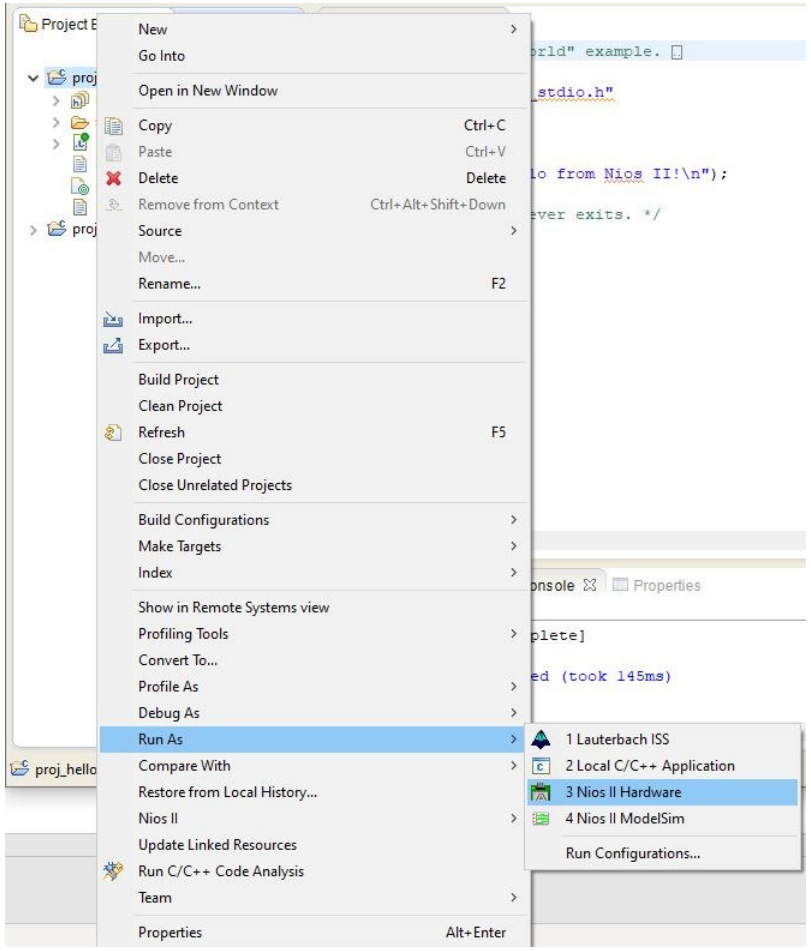
- Check expected and connected system id are same:



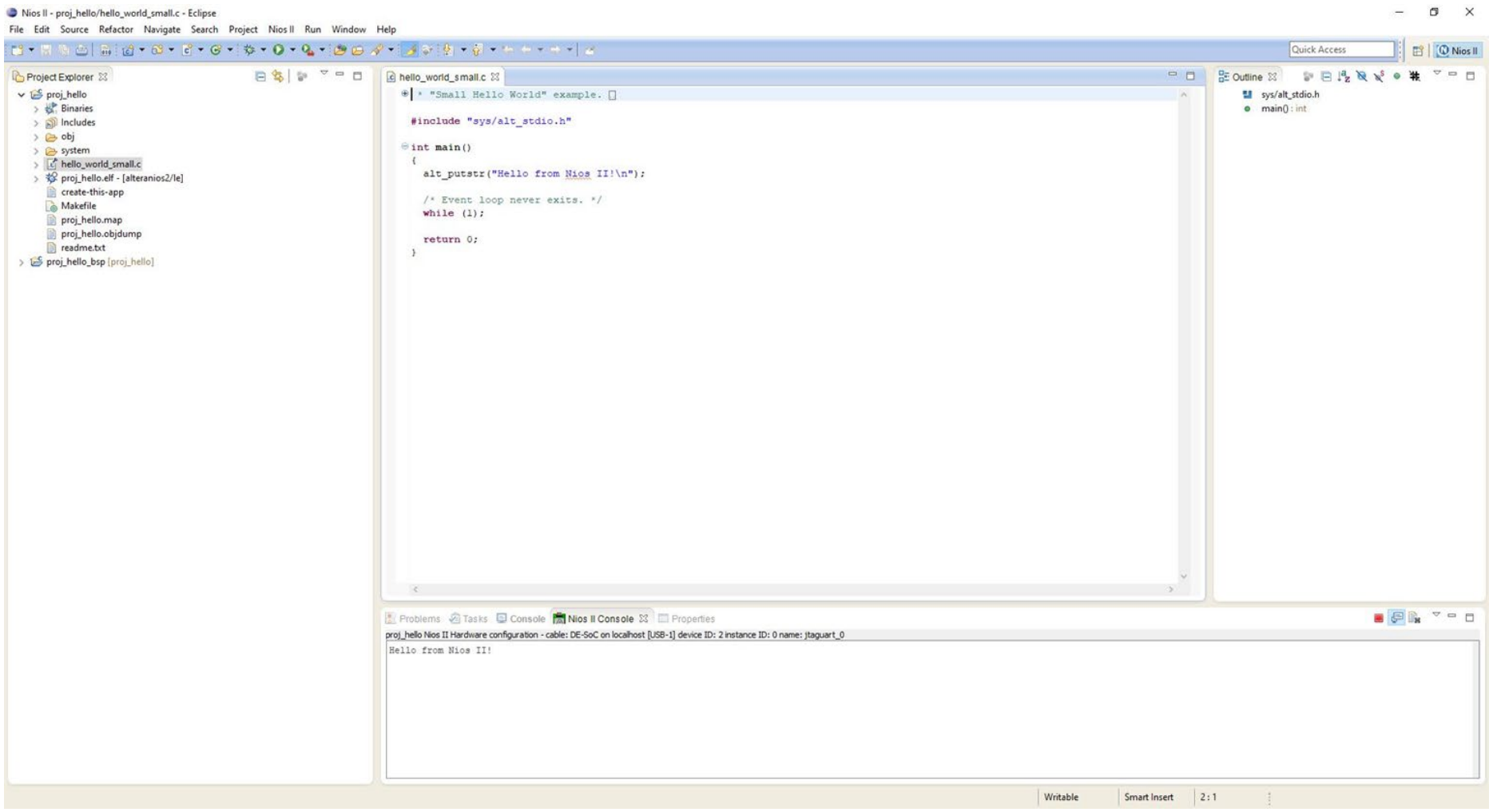


Tutorial: Using the NIOS II IDE

- Under the “Project Explorer” tab, right-click on the application. This time select Run As -> NIOS II Hardware.
- This option will compile and write the program to the on-chip memory we specified in QSYS.



Hello From NIOS II!



Checkpoint: Lab1 (Part I)

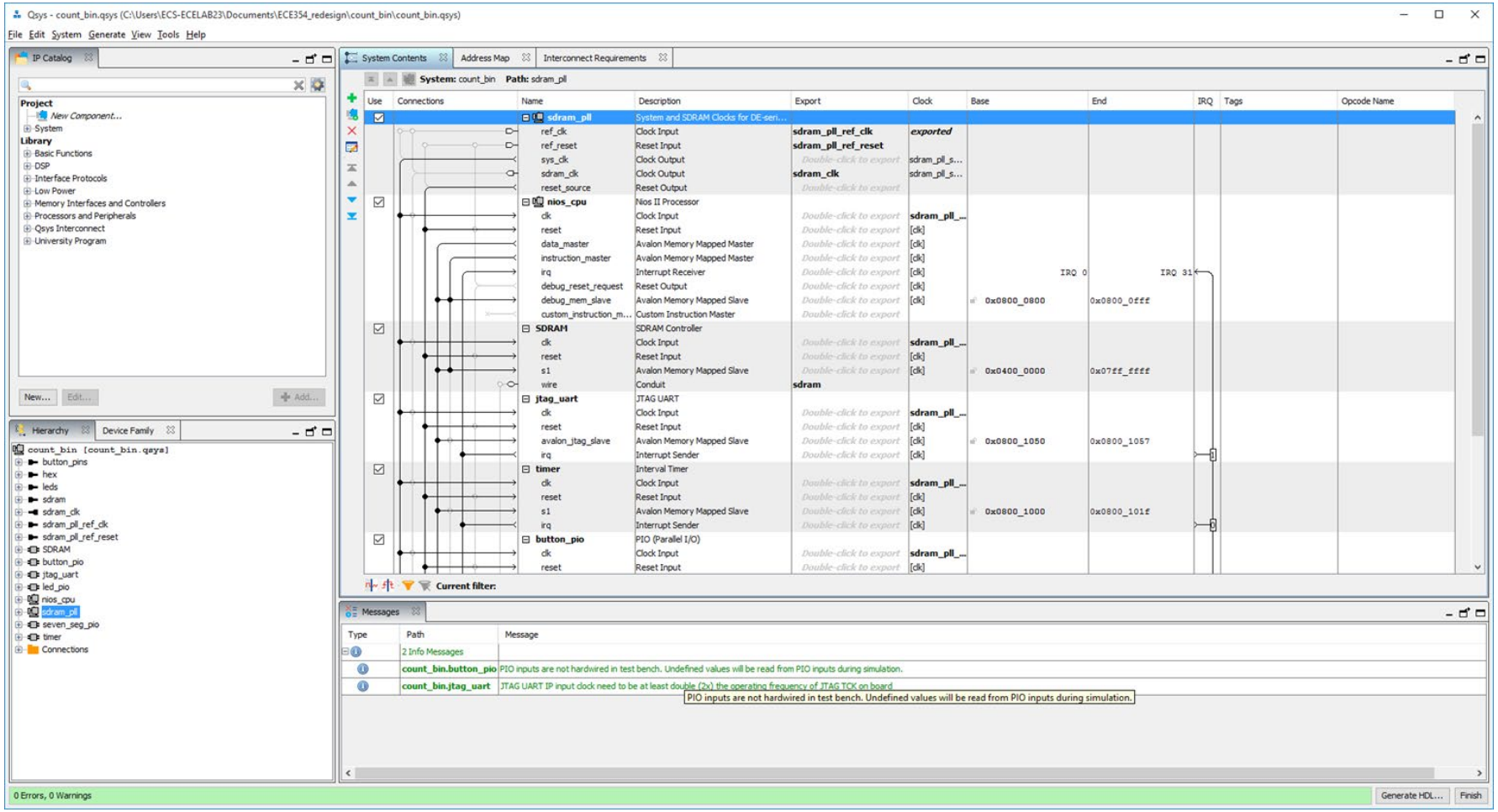
- A NIOS system in the Quartus Prime QSYS
- Use the NIOS II IDE to run a very simple C program on the system defined in QSYS.

Part II

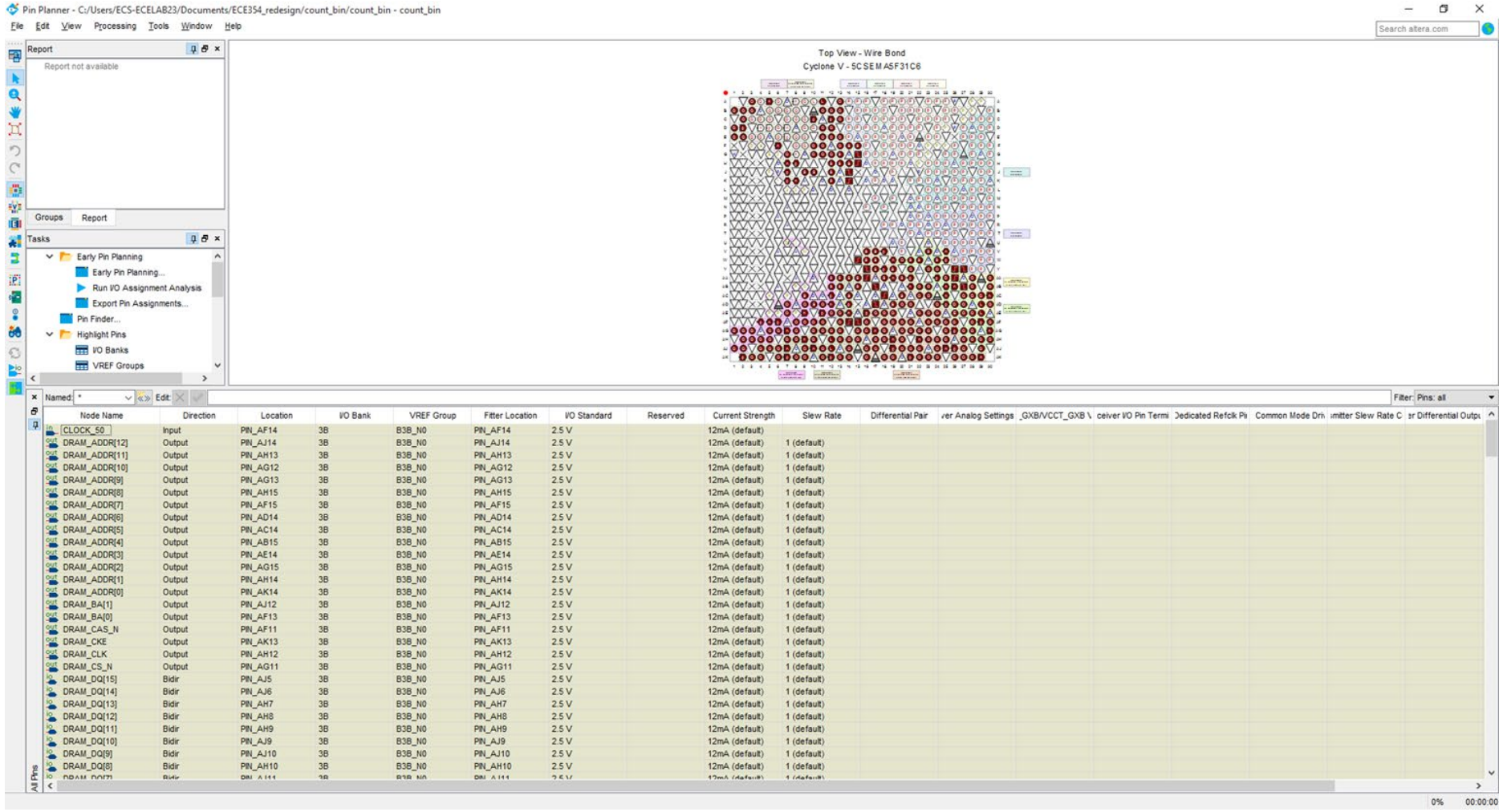
- You will use external memory
- You will use other devices (LED, seven segment, and hex display.)
- Please read lab instructions on class website

Getting Started with Part II (QSYS)

- Add the SDRAM PLL to drive the system
- Add the SDRAM as the program memory
- Add the required PIOs and name them appropriately



Pin assignments viewed from Pin Planner



Questions