QSYS PRO

A10 Nios II Hello World

February 2017

Contents

Introduction

This document is a fundamental guide for user that would like to get started in building a system in Qsys Pro and Quartus Prime and running a very simple software using a Arria 10 development kit. This user guide starts by creating the Quartus Prime project and Qsys Pro files from scratch.

Hardware:

- Arria 10 SoC development kit 10AS066N3F40E2SG
- Power adapter
- USB Blaster

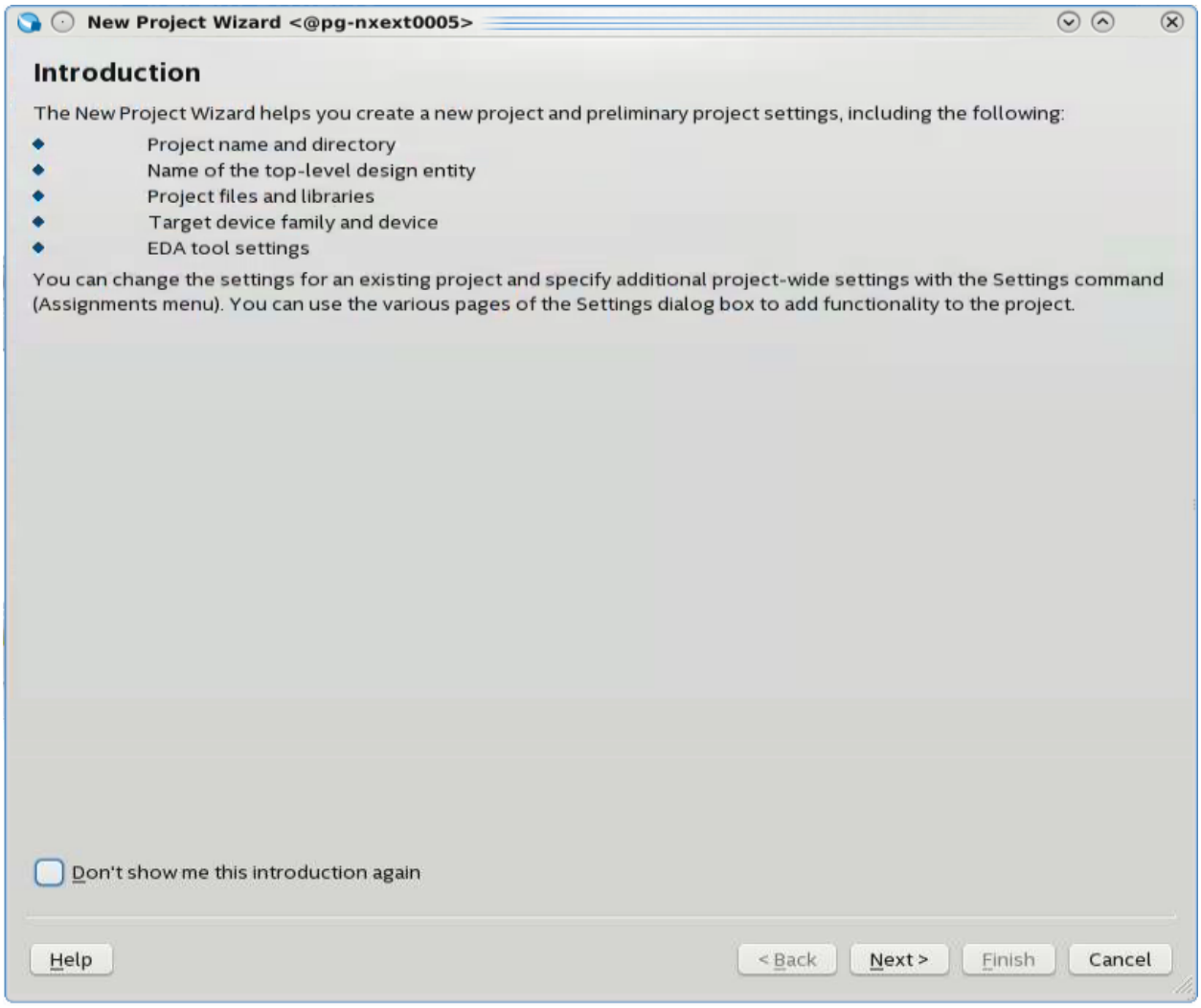
Software:

- Quartus Prime Pro version 17.0
- Please refer to this link for Arria 10 SoC development kit documentation and installation files

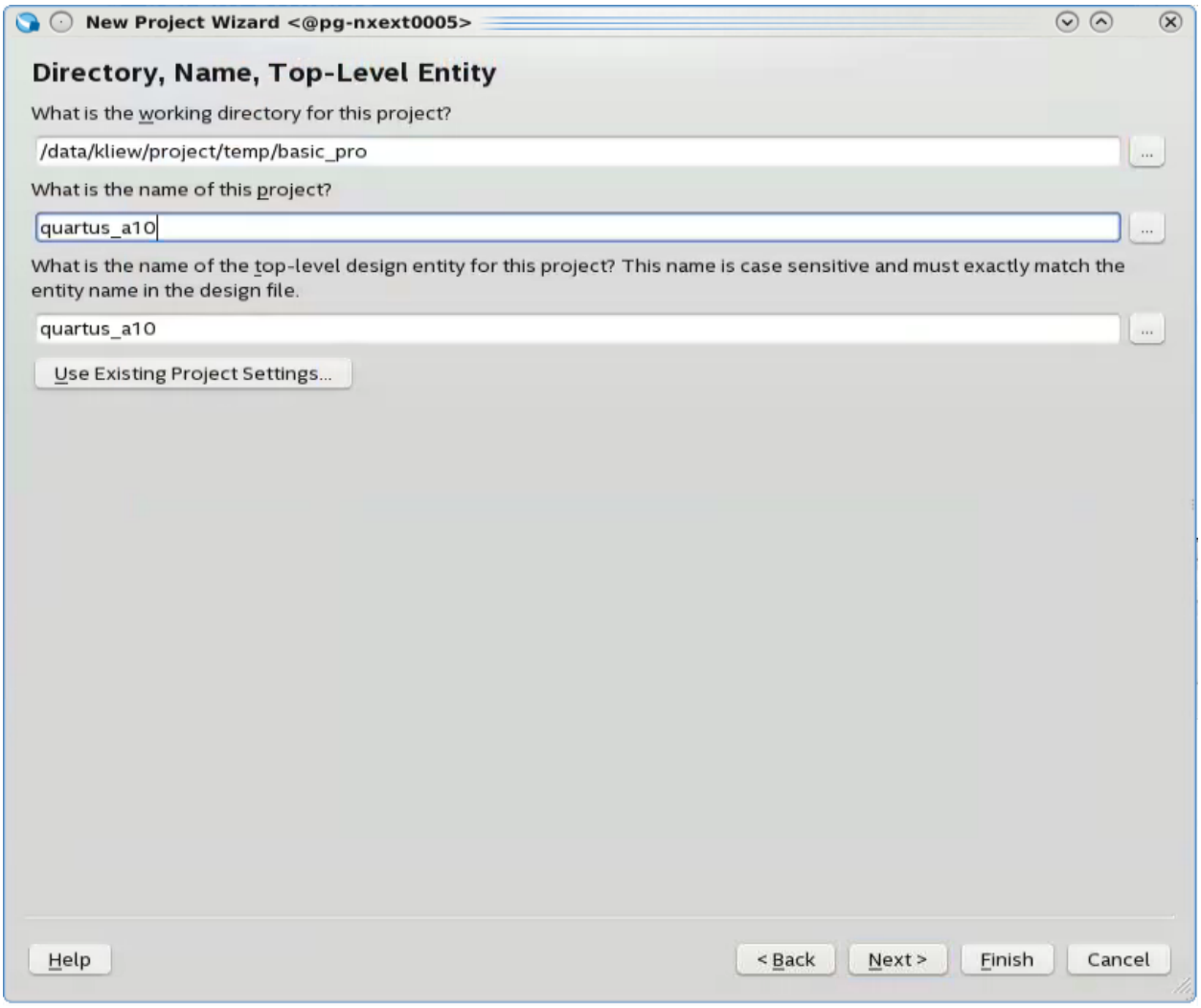
Creating a new project in Quartus Prime Pro

We shall start designing the hardware by including the components needed into Qsys Pro. Here are the steps to do it:

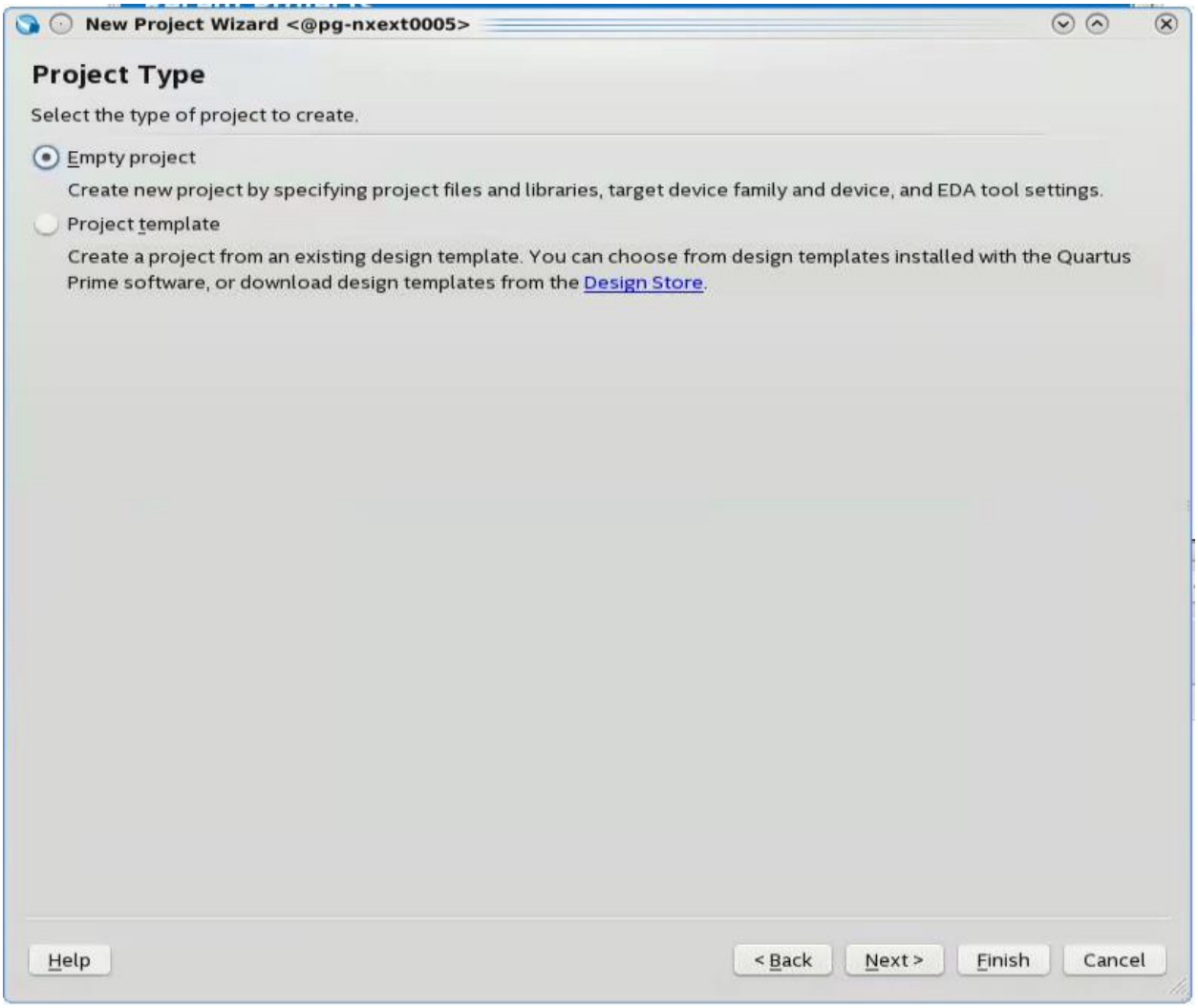
1. Open Quartus Prime Pro and click on “New Project Wizard” to create a new project.
2. The “New Project Wizard” box will be displayed.



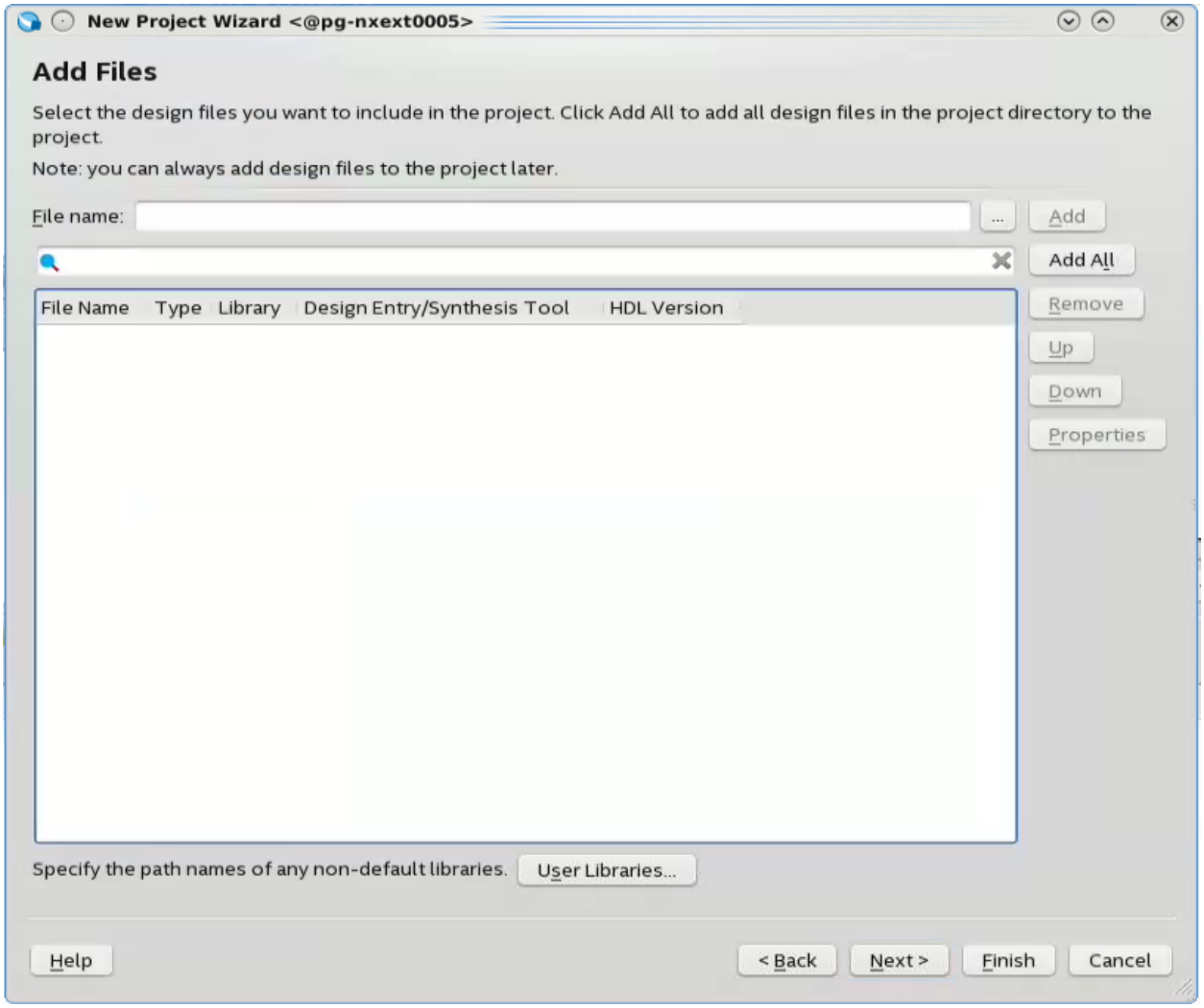
3. Click “Next”.
4. Enter the directory and the new name for the project.



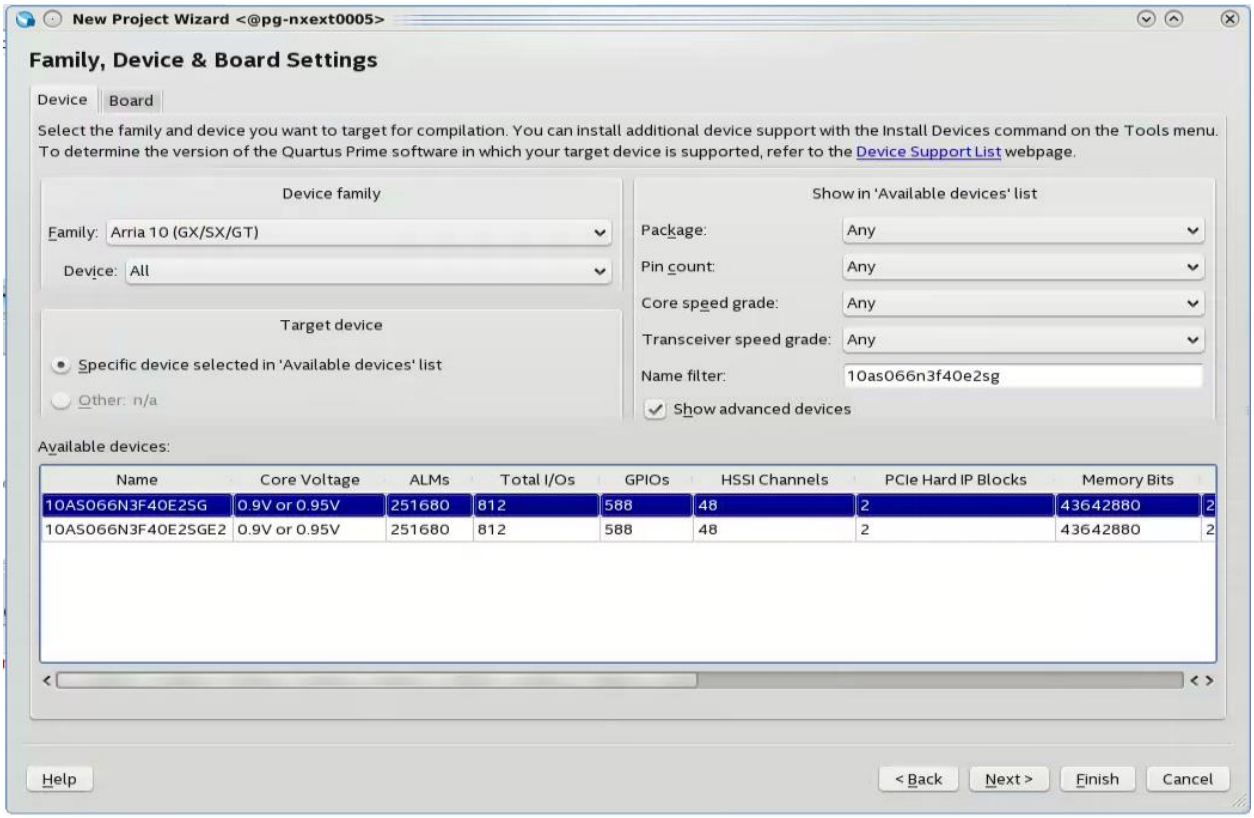
5. Leave the Project Type to be default Empty Project. Click “Next”.



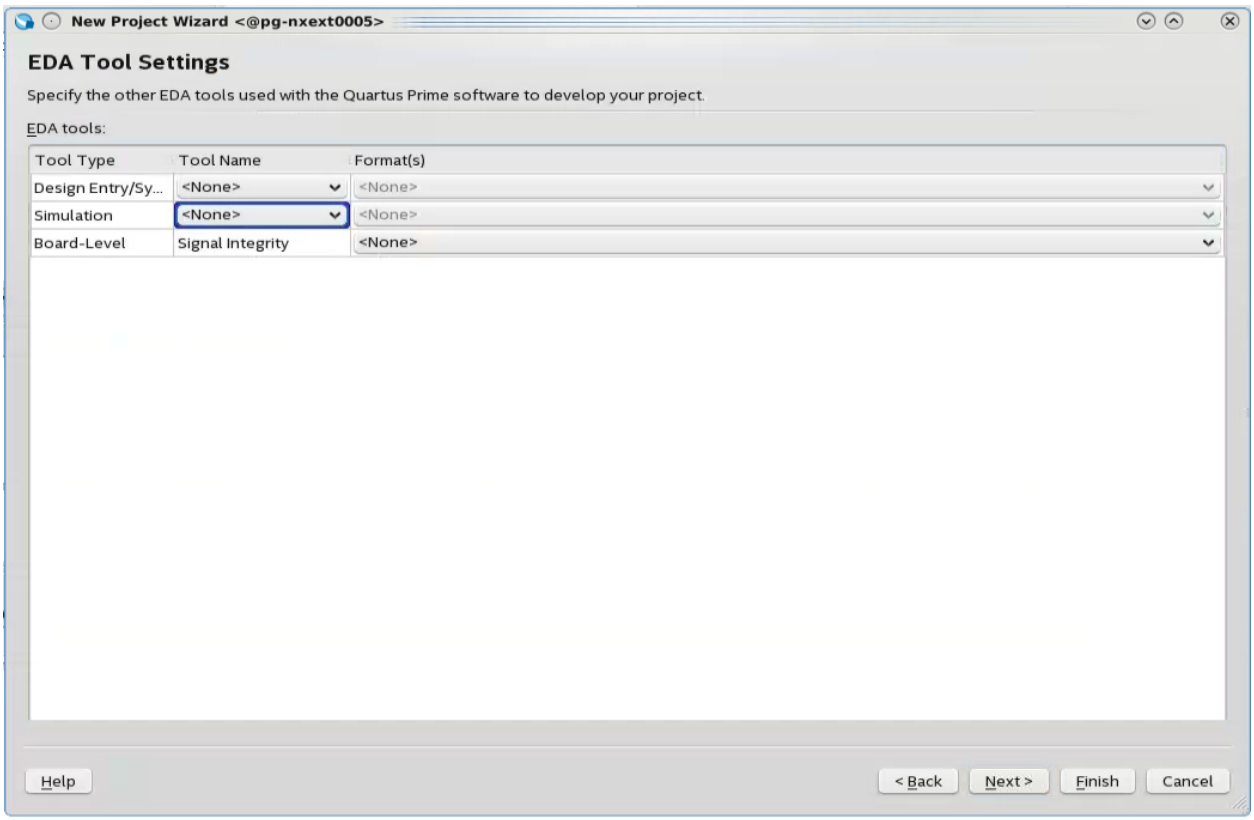
6. For now, leave the Add Files section to be empty. We will add files into the project after generation in Qsys Pro. Click “Next” to proceed further.



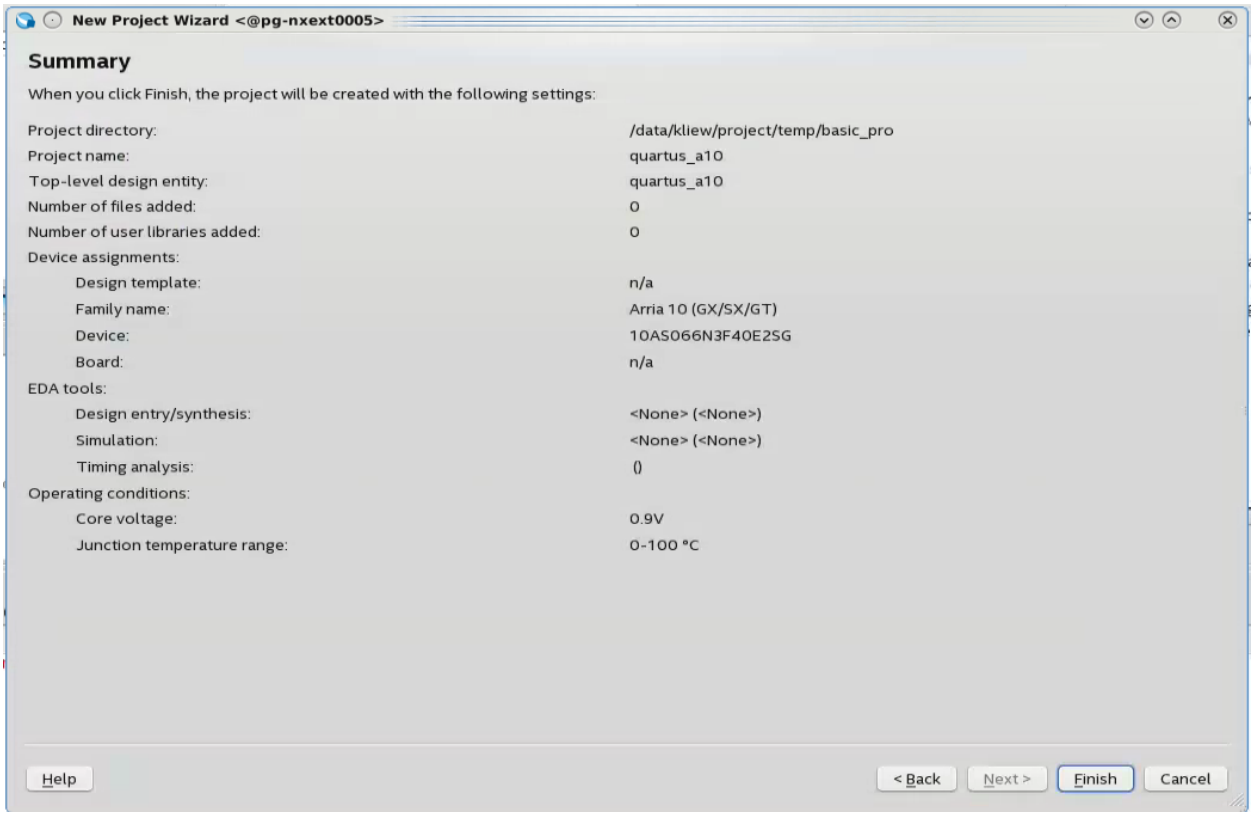
7. In the Family, Device & Board Settings section, search for the Arria 10 SoC development kit device number that you are using. In this example, we are using 10AS066N3F40E2SG.



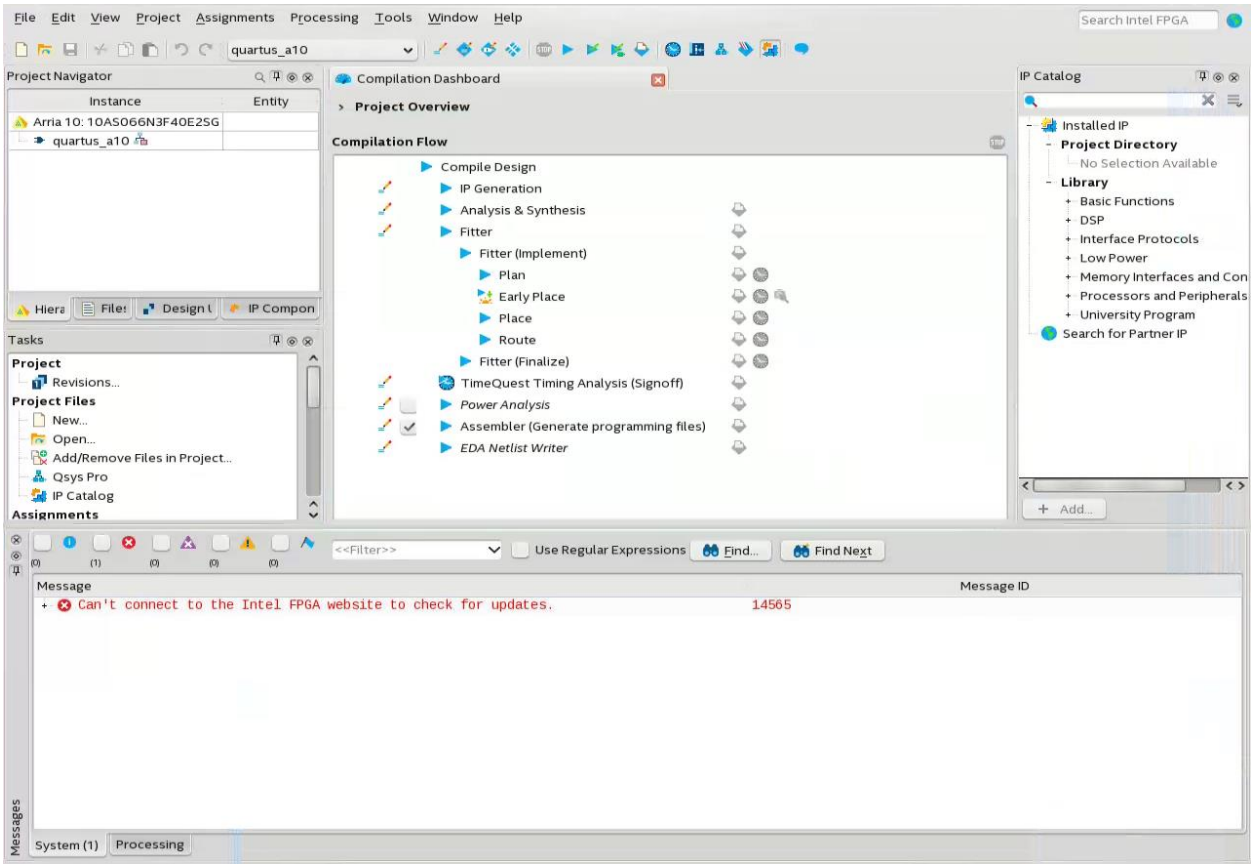
8. Click “Next”.
9. Since we are not doing simulation in this example, set the Simulation to None. Click “Next”.



10. The summary of the project settings will be displayed. Click “Finish” to create a new Quartus Prime Pro project with those settings that you have previously entered.



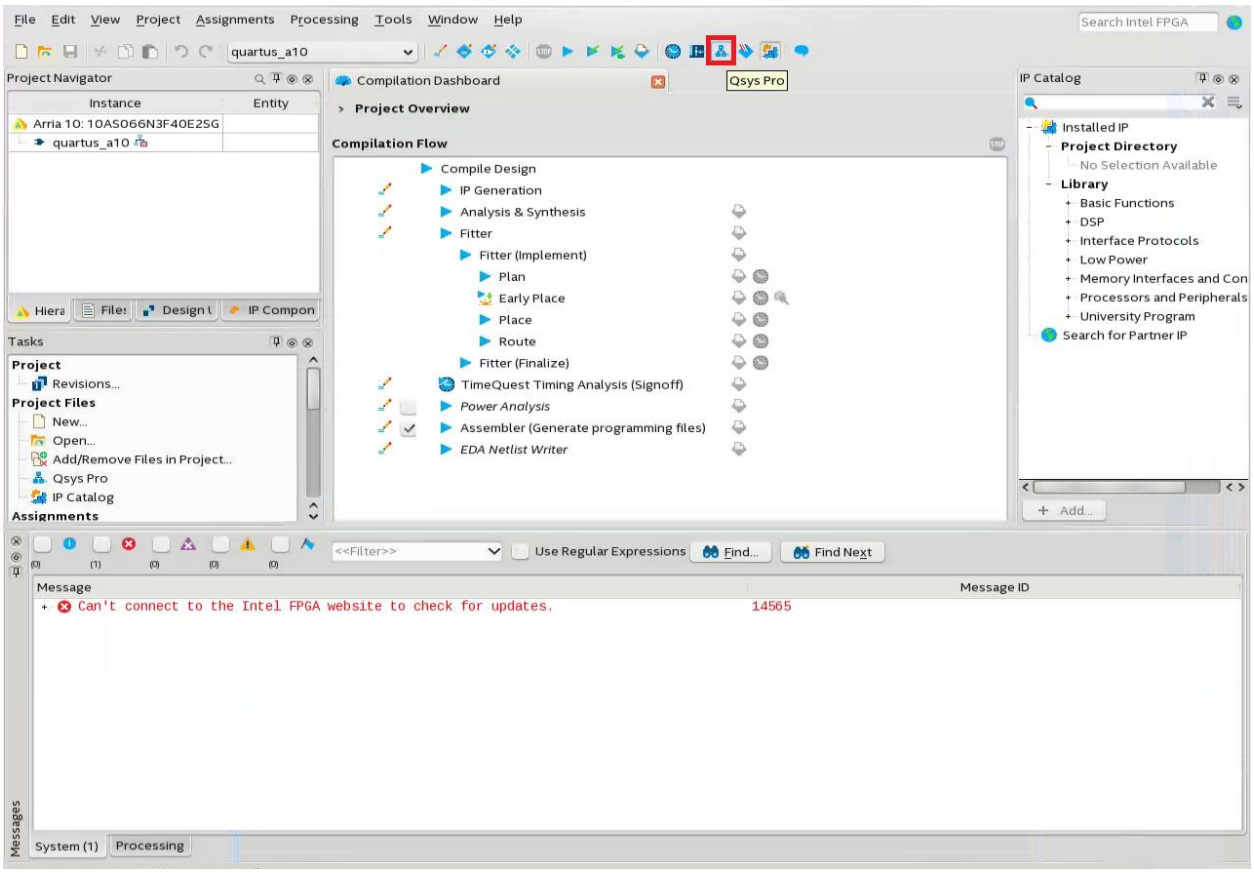
11. This is how it would like in Quartus Prime Pro after it successfully created the project.



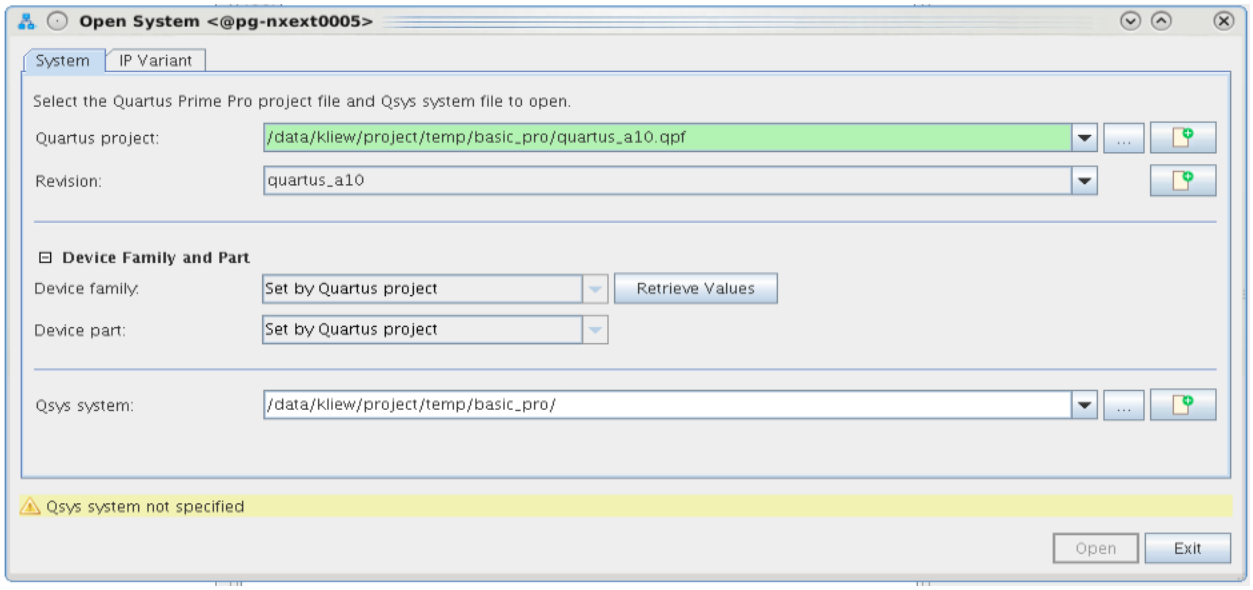
Building Hardware Design in Qsys Pro

Creating a new Qsys Pro file

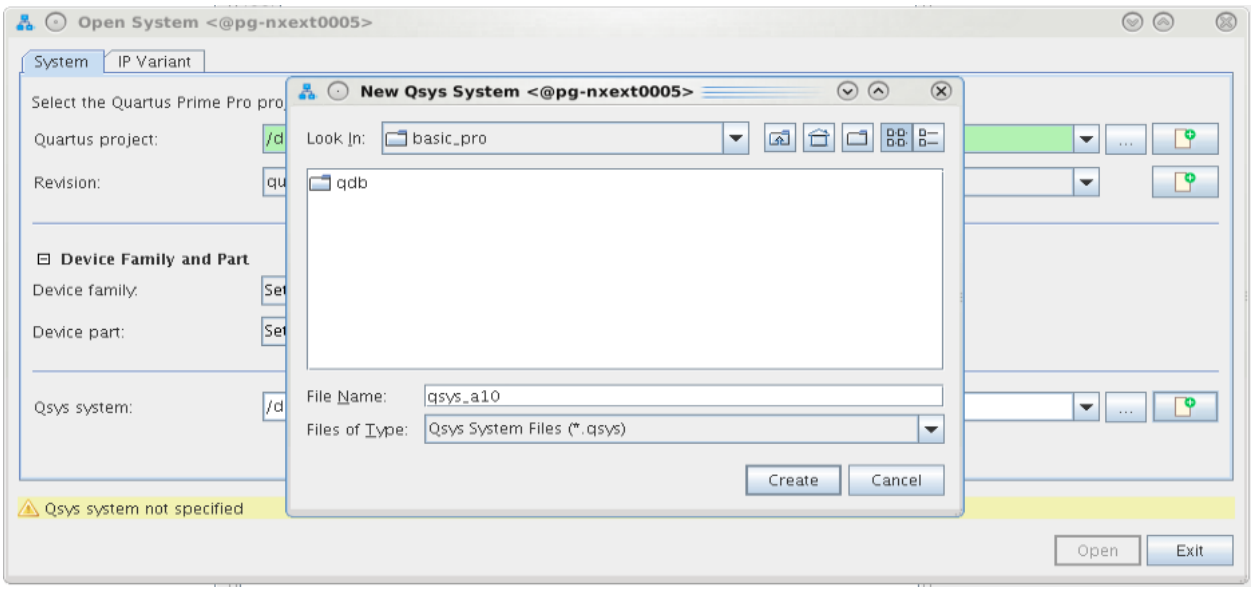
1. From Quartus Prime Pro, open Qsys Pro by clicking the Qsys Pro shortcut button at the top middle section of the GUI.

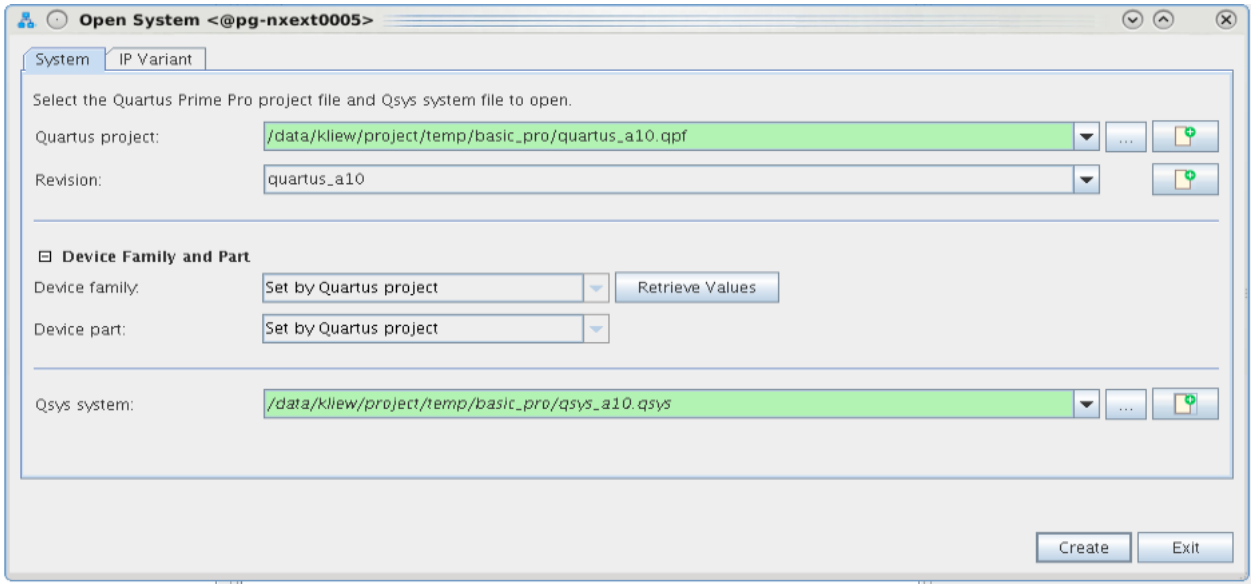


2. Qsys Pro will start up and a “Open System” box will be displayed. The Device Family and the Device Part will be taken from the project settings that has been set in Quartus Prime Pro.

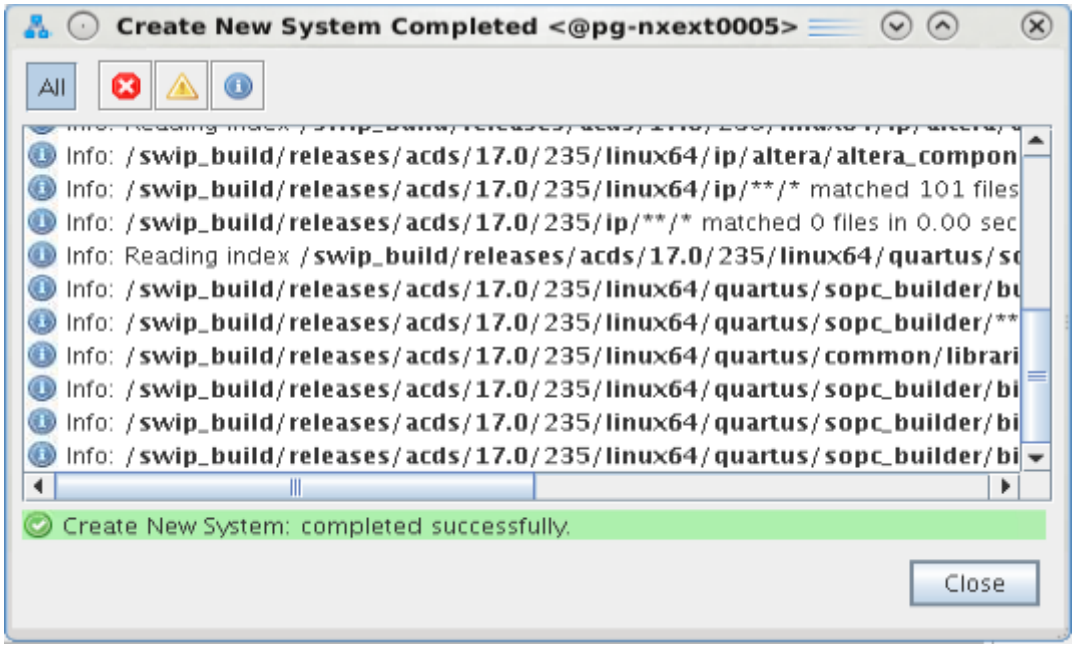


3. In the Qsys System section, click on the  icon.
4. Create a new Qsys Pro file by giving a new name to the .qsys file. Click “Create”.

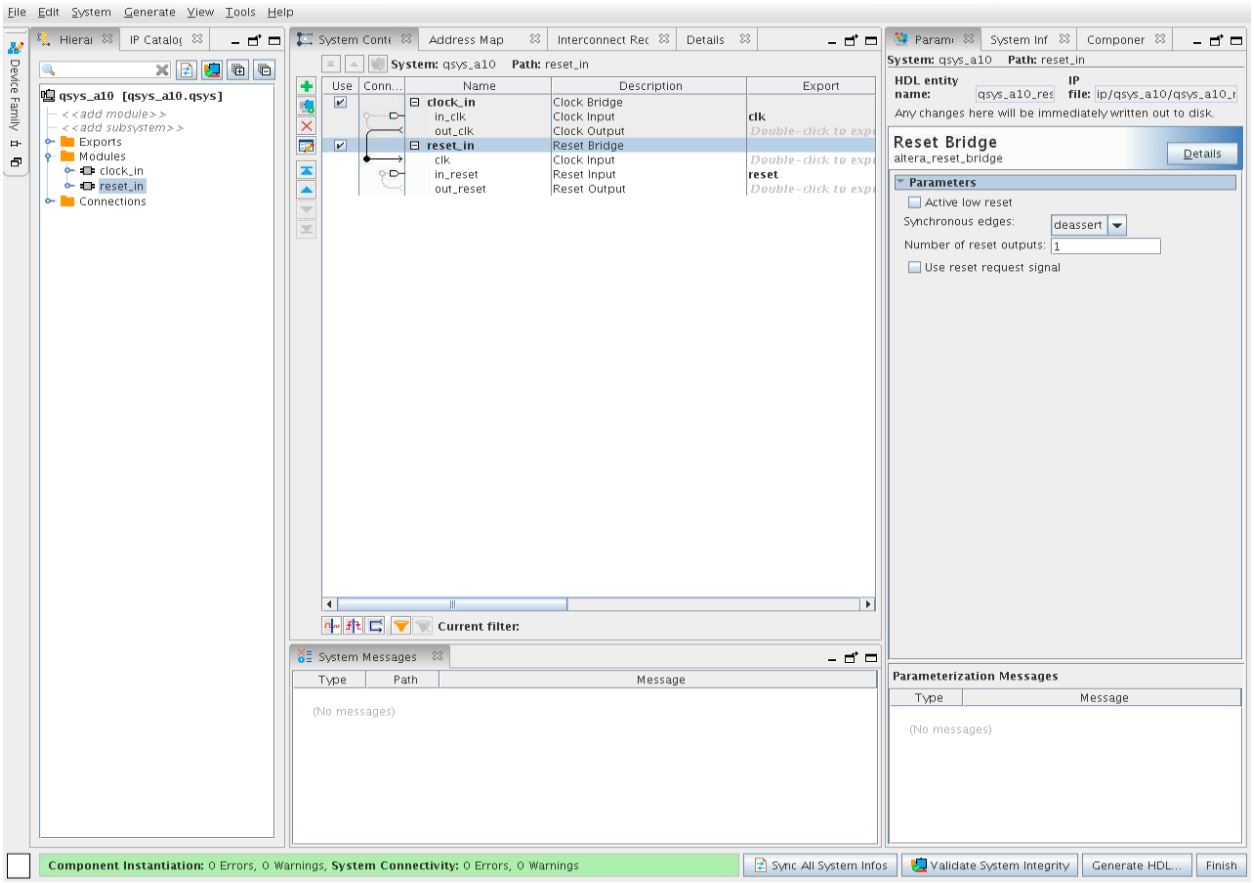




5. Click “Create” again and it will automatically create a new system for you.



6. Click “Close”.
7. This will be the default design that is available once you create a Qsys Pro from scratch. By default, the Clock Bridge and Reset Bridge is already made available for you in the empty design.

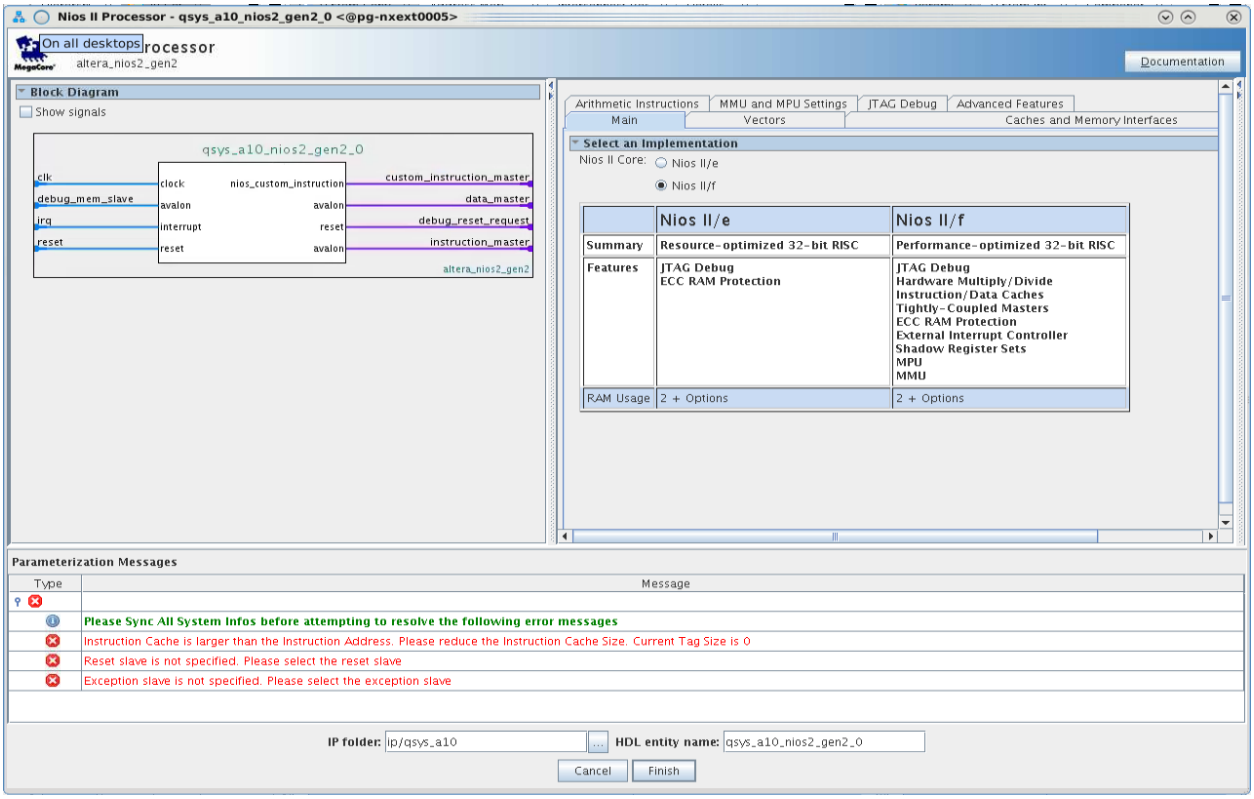


Adding components into the system

1. At the left top corner of the GUI, switch the tab from “Hierarchy” to “IP Catalog” to search for the IPs that is already available in the Catalog to use.
2. At the  icon, type in the name of the IP that you would like to add into the system.

Adding Nios II Processor

1. Search for “Nios II Processor” in IP Catalog.
2. Double click on the Nios II Processor IP in “Embedded Processors” under “Processors and Peripherals” section.
3. The Parameter box of Nios II Processor will be displayed.

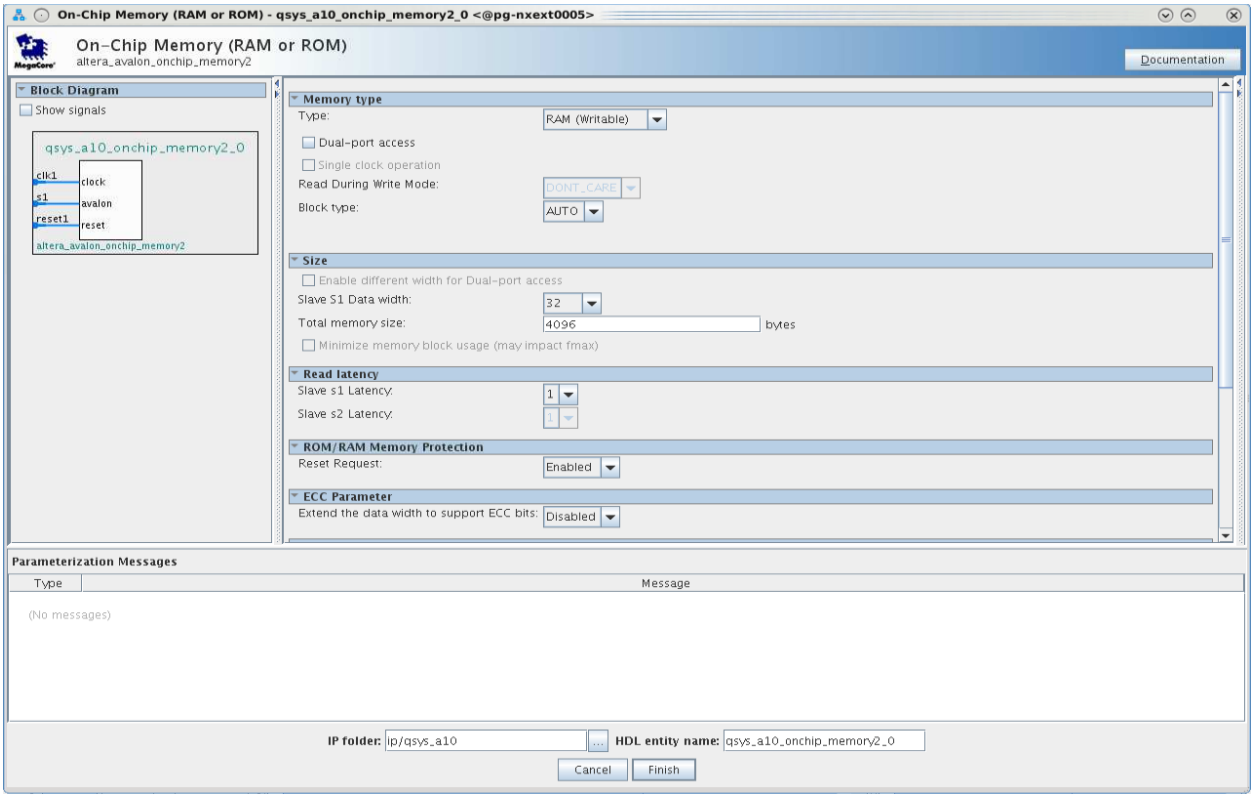


4. Click “Finish”. Leave the settings to be default for now. We can change it later.

Note: Nios II/f core will need a license. If you are creating a very simple design that does not need to use the full feature of the processor, you can use Nios II/e core, which is free.

Adding On-Chip Memory

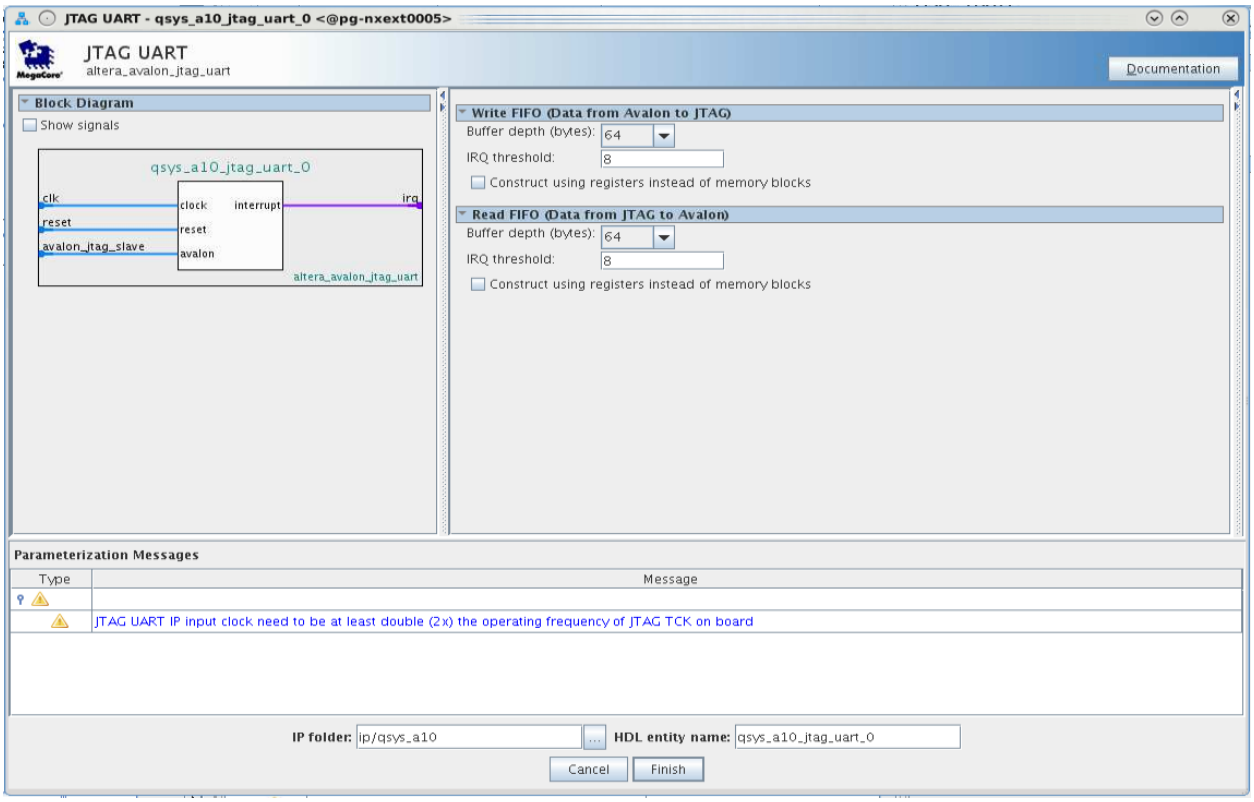
1. Search for On-Chip Memory in the IP Catalog.
2. Double click on the On-Chip Memory (RAM or ROM) IP in “On Chip Memory” under “Basic Functions” section.
3. The Parameter box of On-Chip Memory will be displayed.



4. Click “Finish”. Leave the settings to be default for now. We can change it later.

Adding JTAG UART

1. Search for Jtag Uart in the IP Catalog.
2. Double click on the Jtag Uart IP in “Serial” under “Interface Protocols” section.
3. The Parameter box of Jtag Uart will be displayed.



4. Click “Finish”. Leave the settings to be default for now. We can change it later.

Connecting the IPs instantiated

Clock Bridge:

	_nios2_gen2_0.clk	_onchip_memory2_0.clk	_jtag_uart_0.clk
clock_in.out_clk	✓	✓	✓

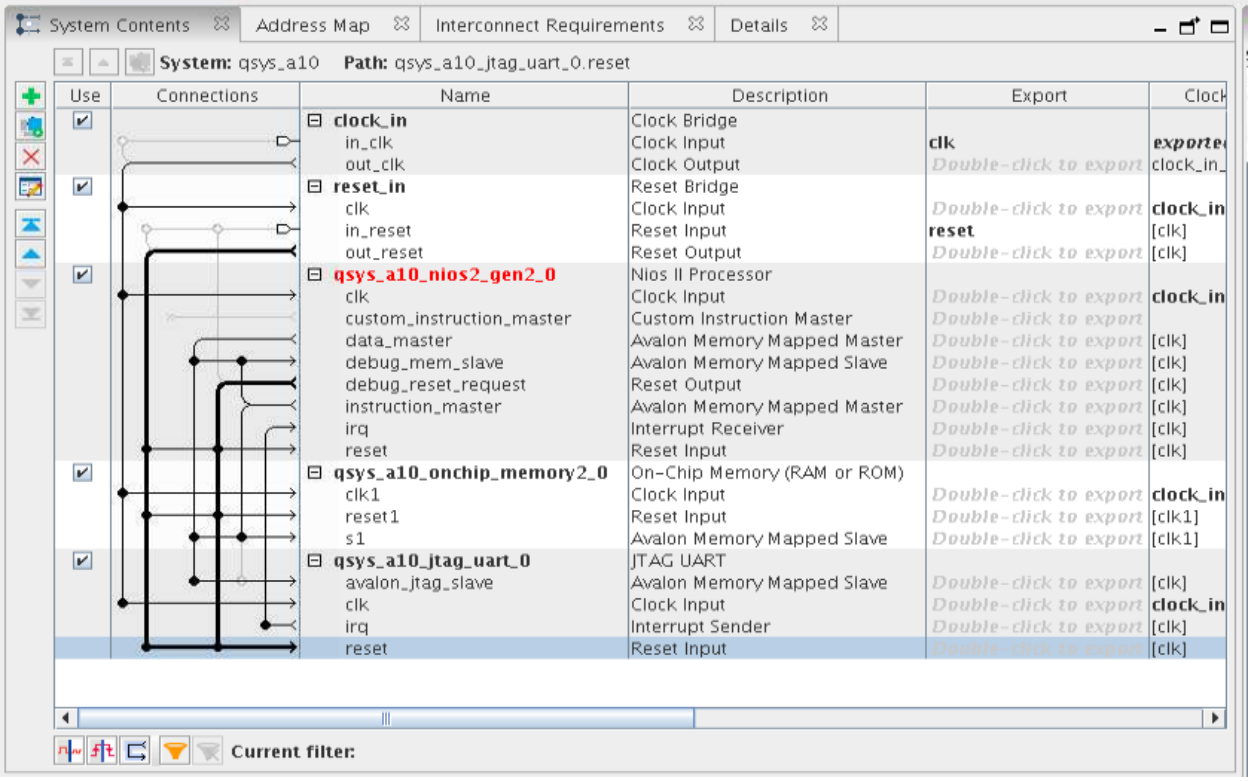
Reset Bridge:

	_nios2_gen2_0.reset	_onchip_memory2_0.reset1	_jtag_uart_0.reset
reset_in.out_out_reset	✓	✓	✓

Nios II Processor:

	_onchip_memory2_0.s1	_jtag_uart_0.s1	_jtag_uart_0.irq	_nios2_gen2_0.reset	_onchip_memory2_0.reset1	_jtag_uart_0.reset
_nios2_gen2_0.data_master	✓					
_nios2_gen2_0.instruction_master	✓	✓				
_nios2_gen2_0.irq			✓			
_nios2_gen2_0.debug_reset_request				✓	✓	✓

5. The full system after connecting all the IPs in Qsys Pro are as shown below. We shall eliminate all the errors and warning next.



Eliminating warnings and errors

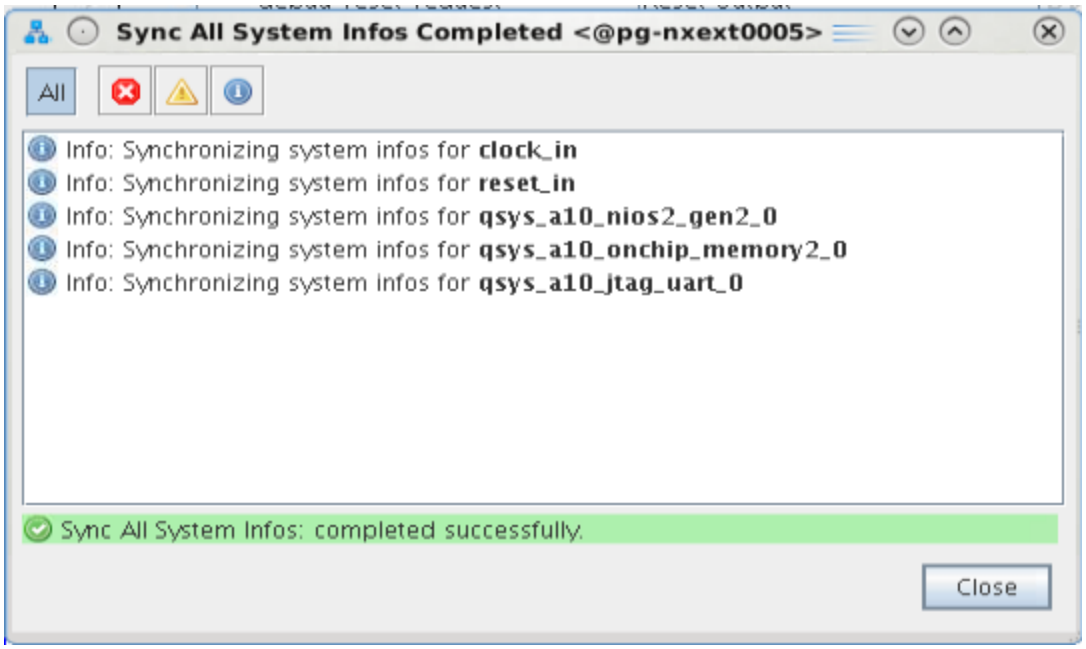
Error: Base address overlaps

⊗	qsys_a10.qsys_a10_nios2_gen2_0.data_master	qsys_a10_jtag_uart_0.avalon_jtag_slave (0x0..0x7) overlaps qsys_a10_onchip_memory2_0.s1 (0x0..0xfff)
⊗	qsys_a10.qsys_a10_nios2_gen2_0.data_master	qsys_a10_onchip_memory2_0.s1 (0x0..0xfff) overlaps qsys_a10_jtag_uart_0.avalon_jtag_slave (0x0..0x7)
⊗	qsys_a10.qsys_a10_nios2_gen2_0.data_master	qsys_a10_nios2_gen2_0.debug_mem_slave (0x800..0xfff) overlaps qsys_a10_onchip_memory2_0.s1 (0x0..0xfff)
⊗	qsys_a10.qsys_a10_nios2_gen2_0.instruction_master	qsys_a10_jtag_uart_0.avalon_jtag_slave (0x0..0x7) overlaps qsys_a10_onchip_memory2_0.s1 (0x0..0xfff)

1. Go to “System” and click on “Assign Base Addresses”.
2. Errors will still be seen in the “System Messages”.

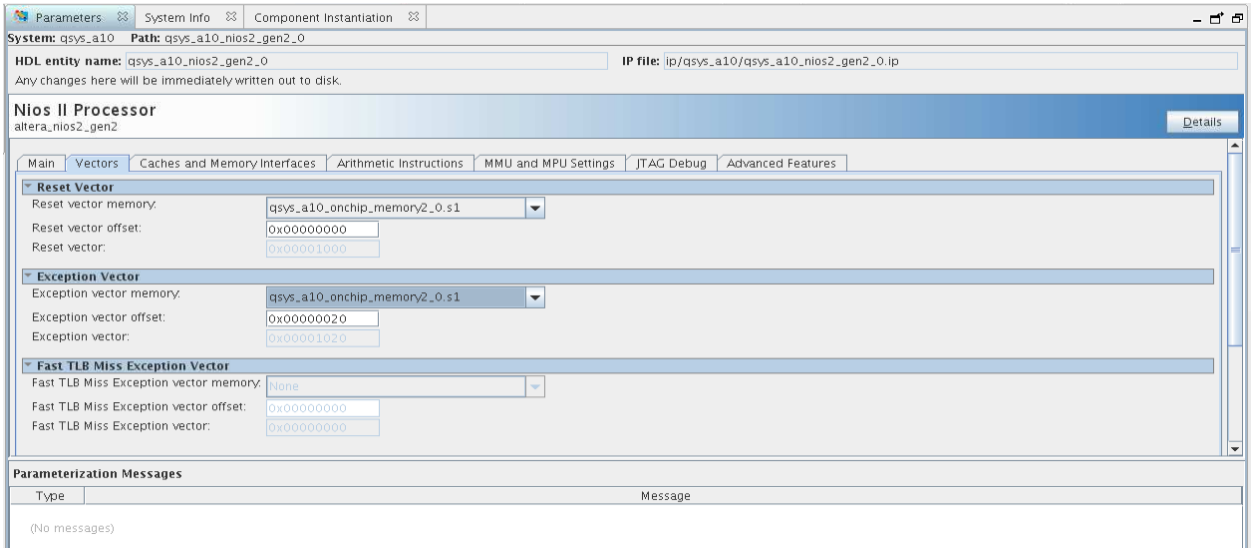
Type	Path	Message
⚠	4 Component Instantiation Warnings	
⚠	qsys_a10.qsys_a10_nios2_gen2_0	System Information doesn't match requirements of IP. Double-click to open System Info tab.
⚠	qsys_a10.qsys_a10_nios2_gen2_0	Errors found in IP parameterization.
⚠	qsys_a10.qsys_a10_jtag_uart_0	System Information doesn't match requirements of IP. Double-click to open System Info tab.
⚠	qsys_a10.qsys_a10_jtag_uart_0	Warnings found in IP parameterization.
⊗	5 System Connectivity Errors	
ⓘ	Qsys Pro Tip	Please Sync All System Infos before attempting to resolve the following error messages
⊗	qsys_a10.qsys_a10_nios2_gen2_0.data_master	qsys_a10_onchip_memory2_0.s1 (0x1000..0x1fff) is outside the master's address range (0x0..0xfff)
⊗	qsys_a10.qsys_a10_nios2_gen2_0.data_master	qsys_a10_nios2_gen2_0.debug_mem_slave (0x2800..0x2fff) is outside the master's address range (0x0..0xfff)
⊗	qsys_a10.qsys_a10_nios2_gen2_0.data_master	qsys_a10_jtag_uart_0.avalon_jtag_slave (0x3000..0x3007) is outside the master's address range (0x0..0xfff)
⊗	qsys_a10.qsys_a10_nios2_gen2_0.instruction_master	qsys_a10_onchip_memory2_0.s1 (0x1000..0x1fff) is outside the master's address range (0x0..0xfff)
⊗	qsys_a10.qsys_a10_nios2_gen2_0.instruction_master	qsys_a10_nios2_gen2_0.debug_mem_slave (0x2800..0x2fff) is outside the master's address range (0x0..0xfff)

3. Click on “Sync All System Infos”. This will sync what is instantiated in the system which is in the .qsys with the IPs as defined in the .ip files.

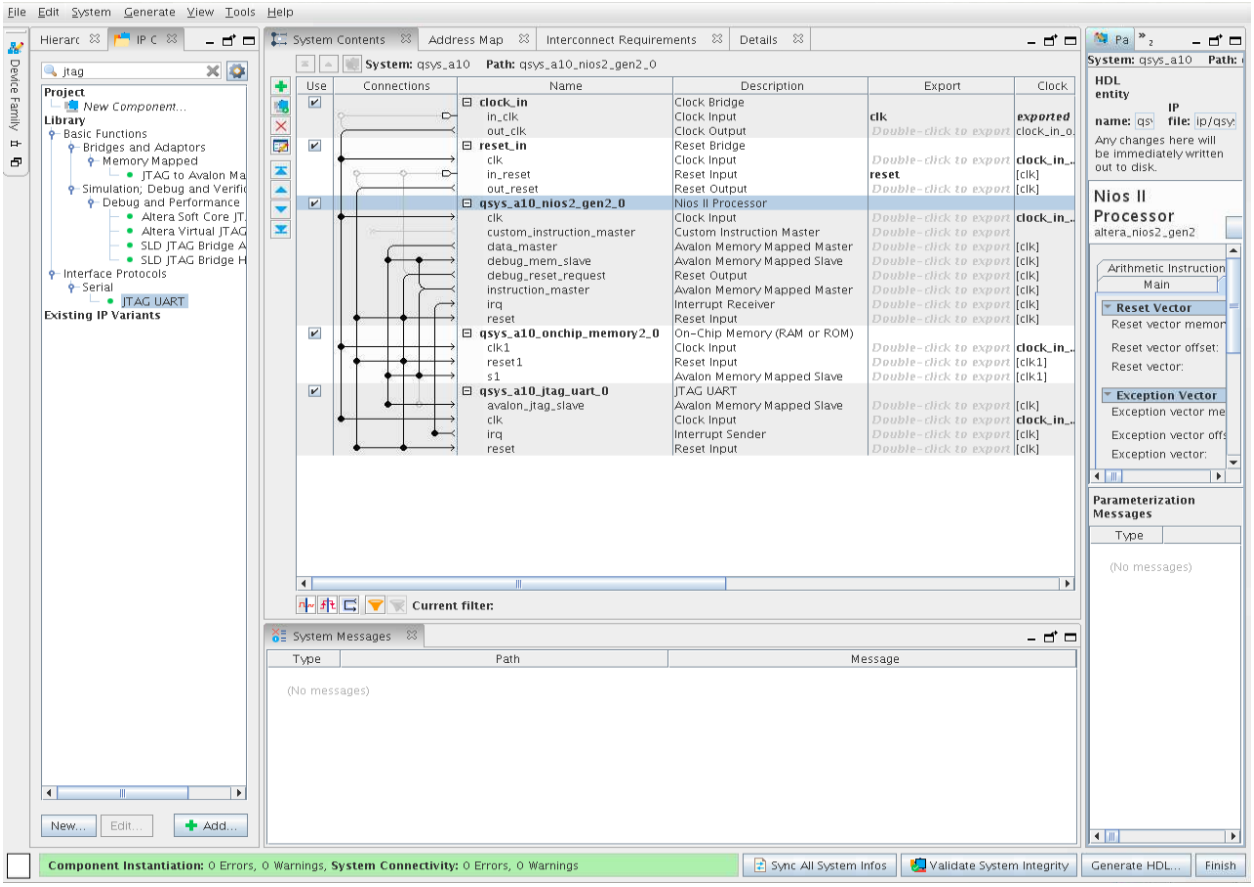


Error: Errors found in IP parameterization

1. Click on the qsys_a10_nios2_gen2_0 IP. At the right side of the GUI, you will see the “Parameter” box displayed.
2. Go to the “Vector” tab.
3. Under “Reset vector”, at the “Reset vector memory” dropbox, select qsys_a10_onchip_memory2_0.s1 to set the reset vector to on chip memory.
4. Under “Exception Vector”, at the “Exception vector memory” dropbox, select qsys_a10_onchip_memory2_0.s1 to set the exception vector to on chip memory.
5. The errors previously displayed in the “Parameterization Messages” in the Nios II Processor parameter box has disappeared.

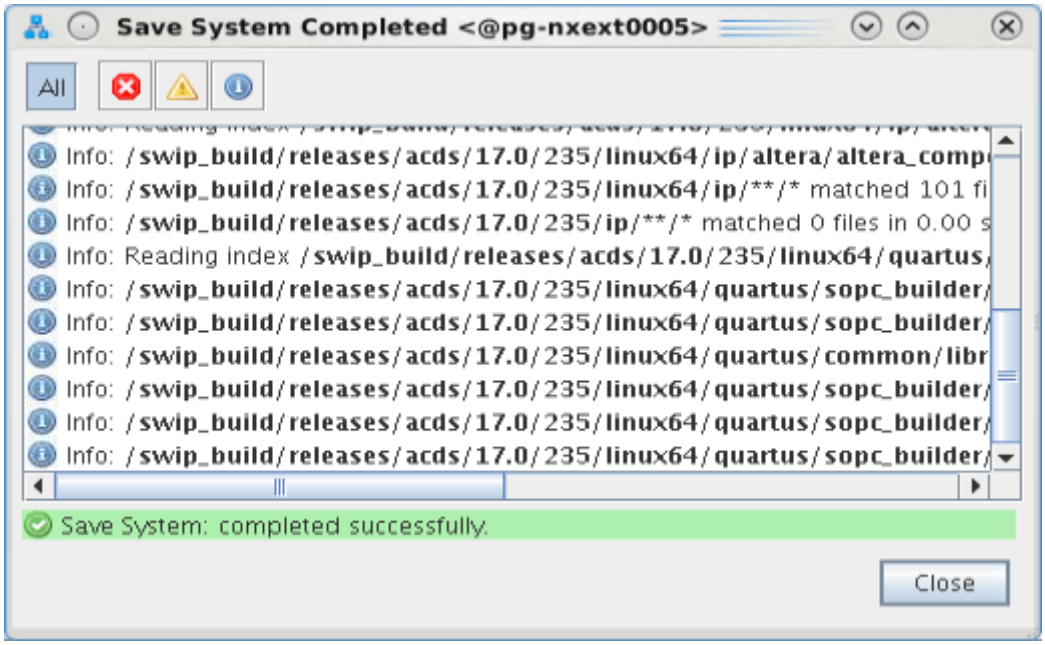


6. The whole system that you have designed will be cleaned with errors and warnings.

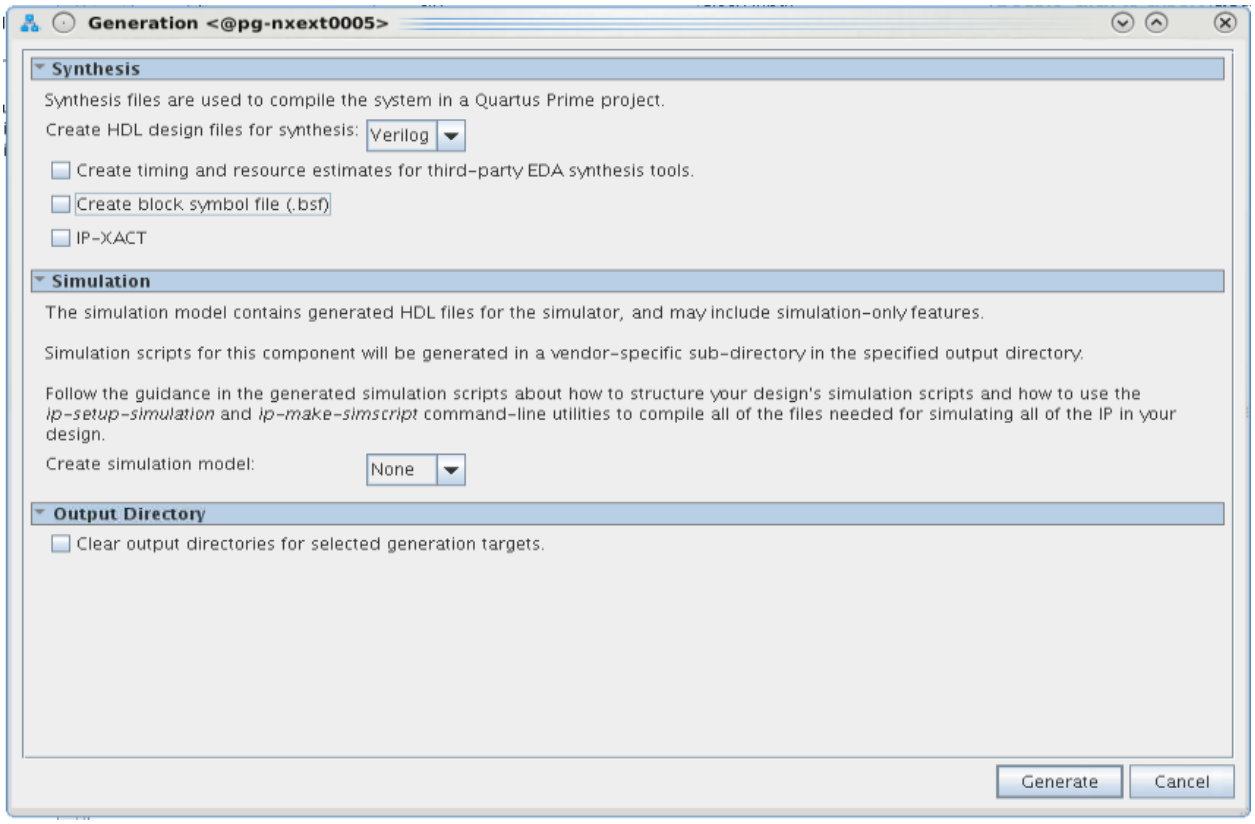


Save the system and generate

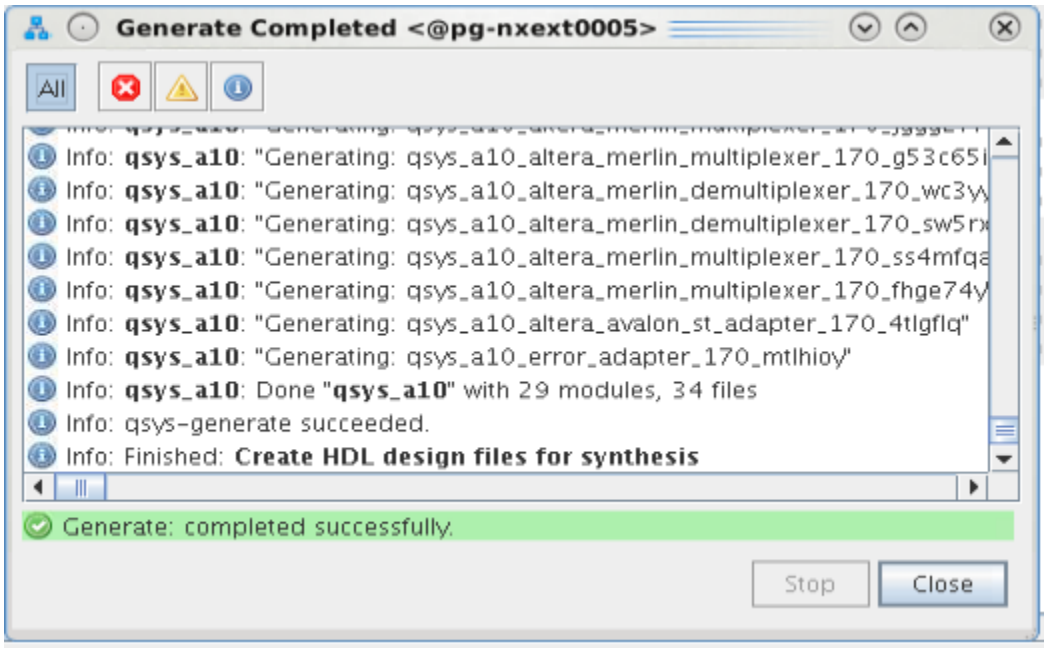
1. Go to “File” and click “Save”. This will save all the changes made.



2. Click “Generate HDL…” at the bottom right button of the GUI. This will generate the system in Qsys Pro.



3. Leave the settings to be default. Unless you would like to have different set of files generated, then set accordingly.



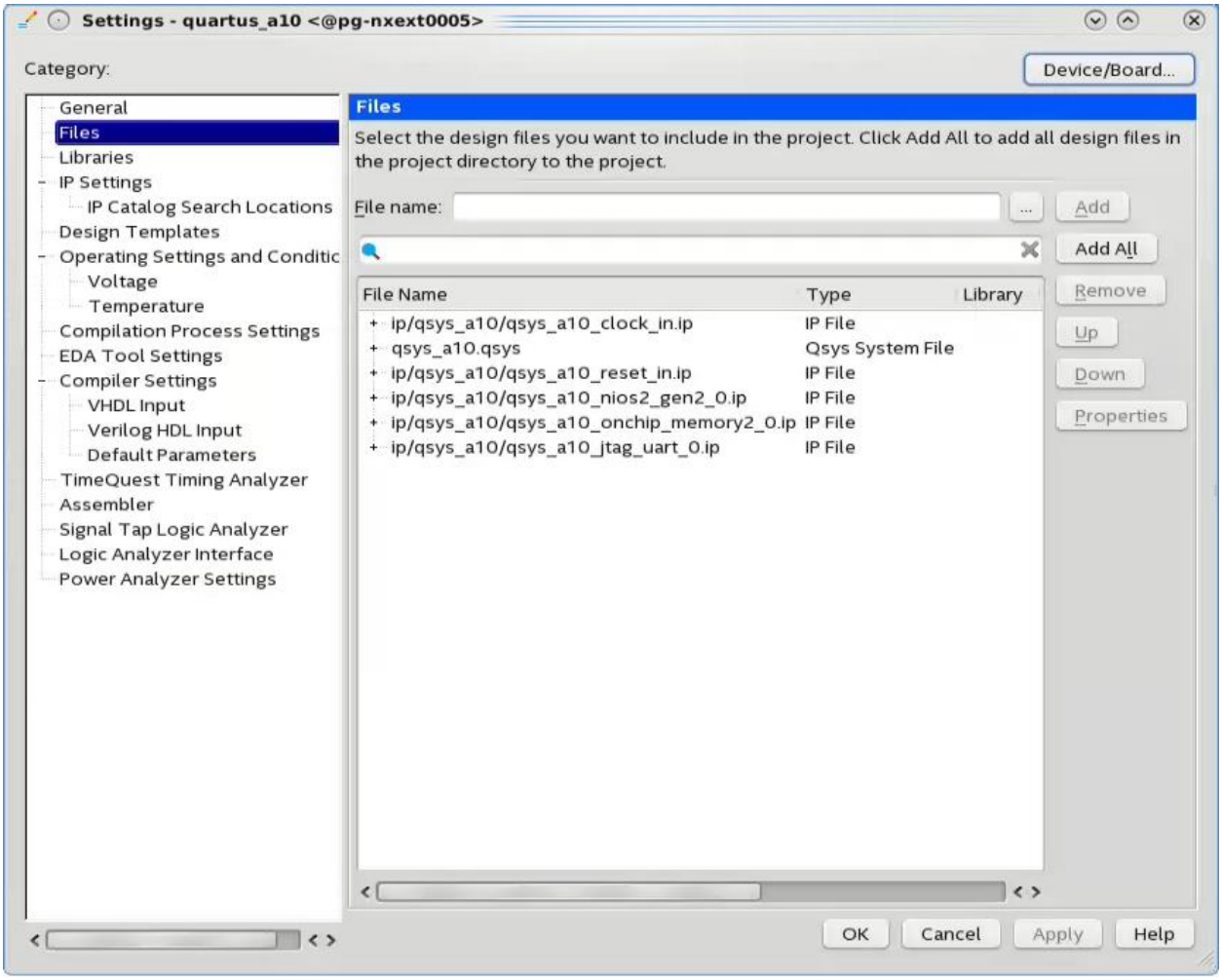
4. Click “Close” to close the “Generate Completed” box.
5. After generation completes, the generation files can be obtained in the generated folder. The folder name here is “qsys_a10”.

```
drwxr-xr-x 6 kliew software   4096 Feb 22 13:39 .
drwxr-xr-x 4 kliew software   4096 Feb 21 17:55 ..
drwxr-xr-x 3 kliew software   4096 Feb 22 13:39 ip
drwxr-xr-x 3 kliew software   4096 Feb 22 13:18 qdb
drwxr-xr-x 2 kliew software   4096 Feb 23 11:21 qsys_a10
-rw-r--r-- 1 kliew software 129180 Feb 23 11:15 qsys_a10.qsys
drwxr-xr-x 2 kliew software   4096 Feb 22 13:39 .qsys_edit
-rw-r--r-- 1 kliew software   1319 Feb 22 13:18 quartus_a10.qpf
-rw-r--r-- 1 kliew software   2679 Feb 23 11:15 quartus_a10.qsf
```

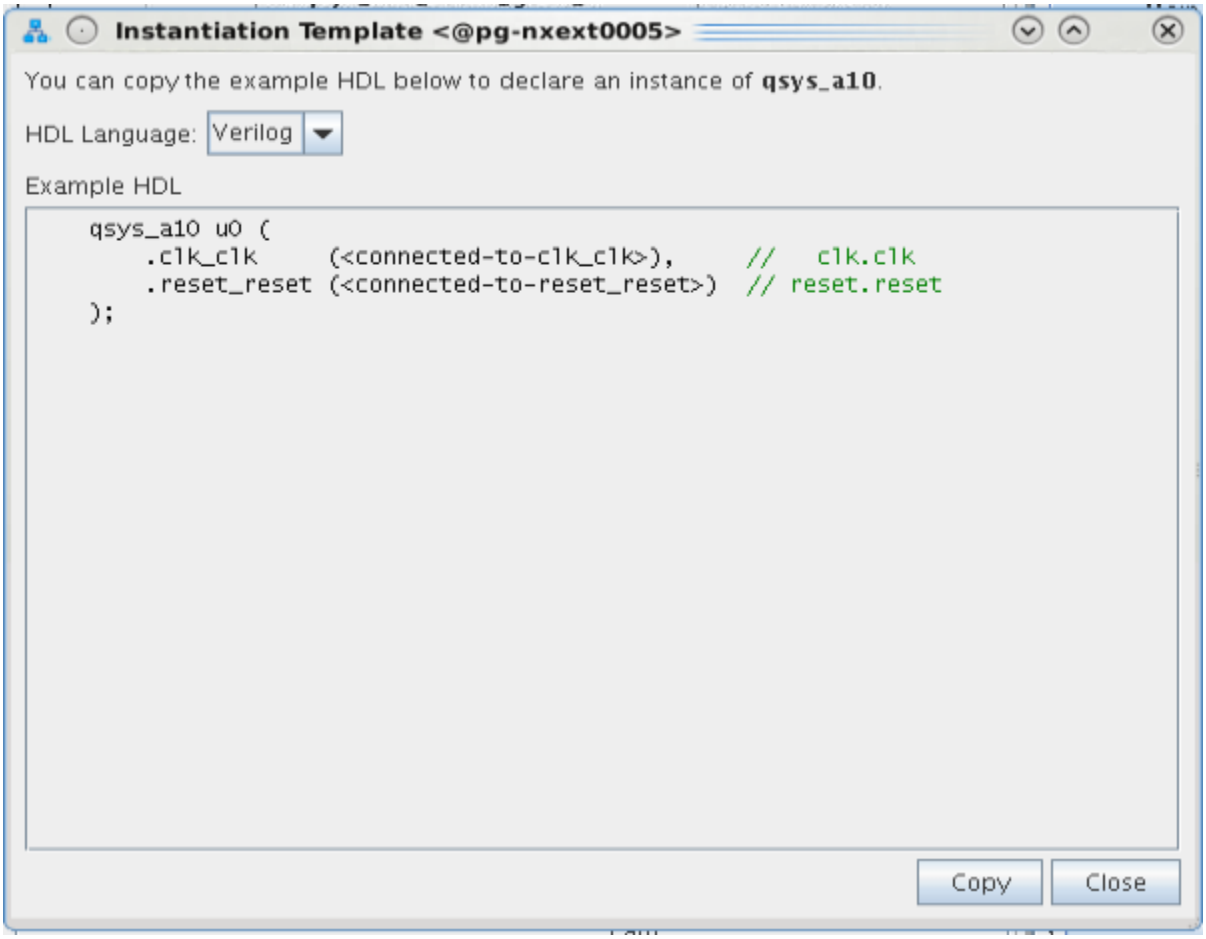
Qsys Pro introduces hierarchical isolation between system interconnect and IP components. Qsys Pro stores the instantiated IP component in .ip file and those system connectivity information in .qsys file.

Compilation in Quartus Prime Pro

1. Switch back to Quartus Prime Pro.
2. Navigate to “Assignments” and click on “Settings”.
3. Go to the “Files” tab. The .ip files and .qsys file has been automatically added into the Quartus Prime Pro project.



4. In this example, the top level file is taken from the ghrd of the A10 SoC development kit and named it quartus_a10.v. This will be set as the top level file.
5. Refer to the instantiation template. Navigate to “Generate” and click on “Show Instantiation Template”.



Here is the example of the top level file:

```
module quartus_a10 (
 // FPGA clock and reset
 input  wire        fpga_clk_100,
 input  wire        fpga_reset_n
);

// SoC sub-system module
qsys_a10 u0 (
 .clk_clk                        (fpga_clk_100),
 .reset_reset                        (fpga_reset_n)
);

endmodule
```

6. Go to “Assignments” and then to “Settings”. Click on the “Files” tab.
7. Under “File name”, click  and add in quartus_a10.v file in. Click “OK”.

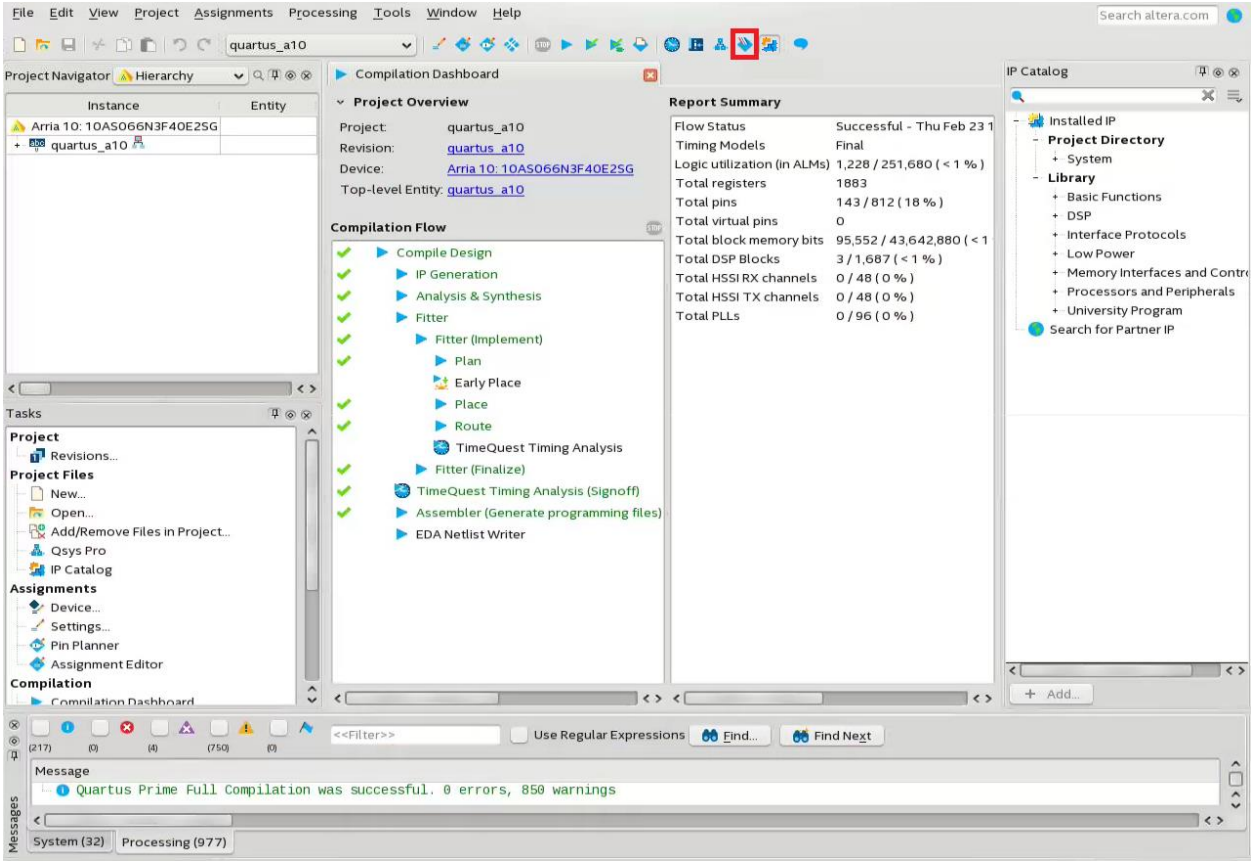
8. At “Project Navigator”, switch to the “Files” tab and right click on quartus_a10.v file and set as top level file.
9. Navigate to “Processing”, then go to “Start” and click on “Start Analysis & Elaboration”.

Pin assignments

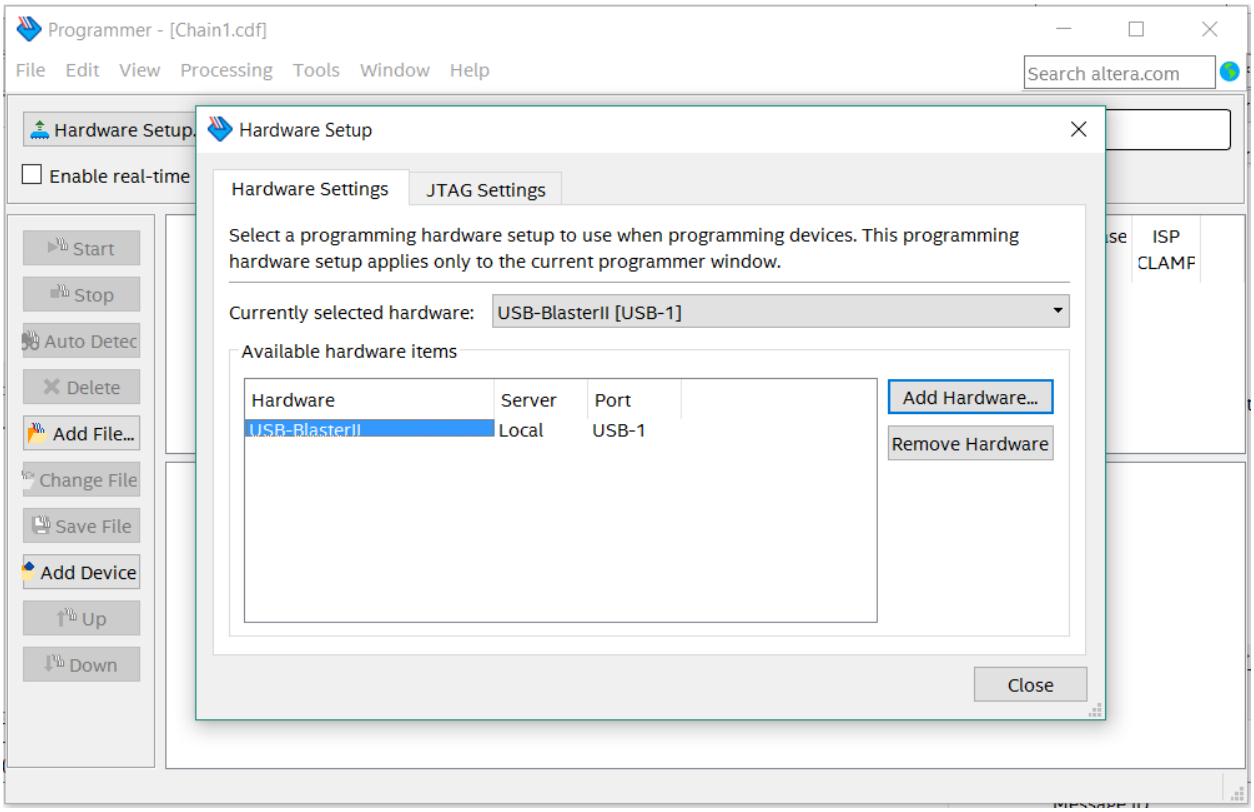
1. Go to “Assignments” and click on “Pin Planner”.
2. For this design, there are only 2 pins that needs to be assigned and the pin location is based on the Arria 10 SoC schematic:
    - fpga_reset_n assigned to PIN_AV21
    - fpga_clk_100 assigned to PIN_AM10
3. Close Pin Planner.
4. Click “Full Compilation” to compile the whole Quartus Prime Pro project.

Programming .sof into Arria 10 SoC development kit

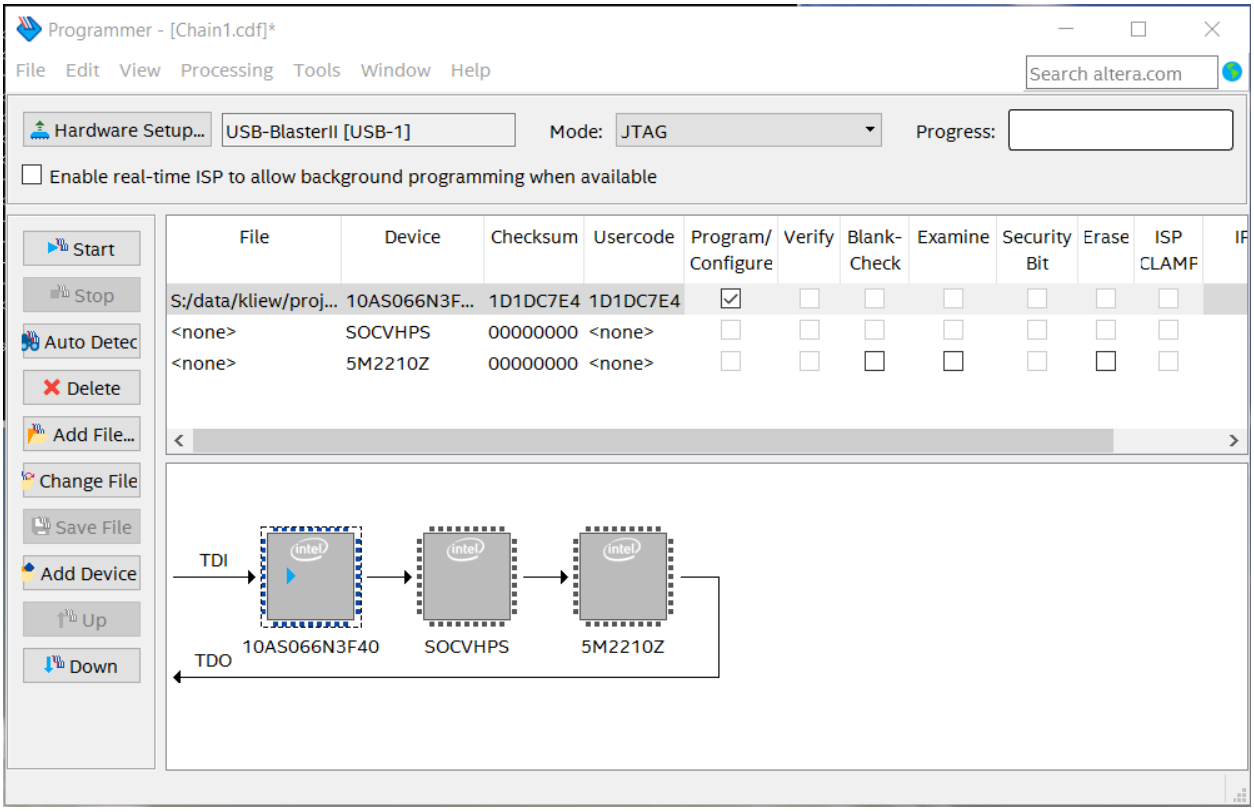
1. Open the Quartus Prime Pro Programmer.



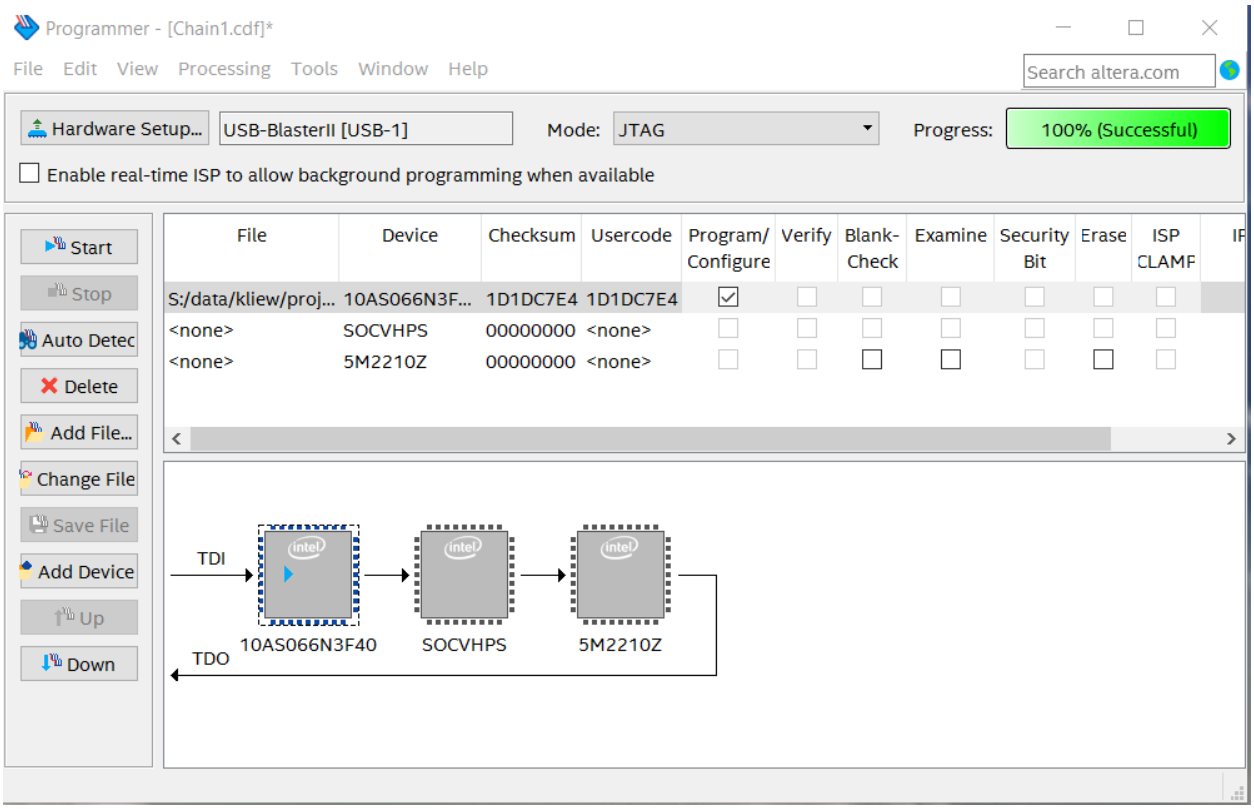
2. Connect the USB Blaster from the Arria 10 SoC development kit to the host machine.



3. Click “Auto Detect” to detect the jtag chain and the device.
4. Add in the .sof into the Programmer and check the “Program/Configure” checkbox.



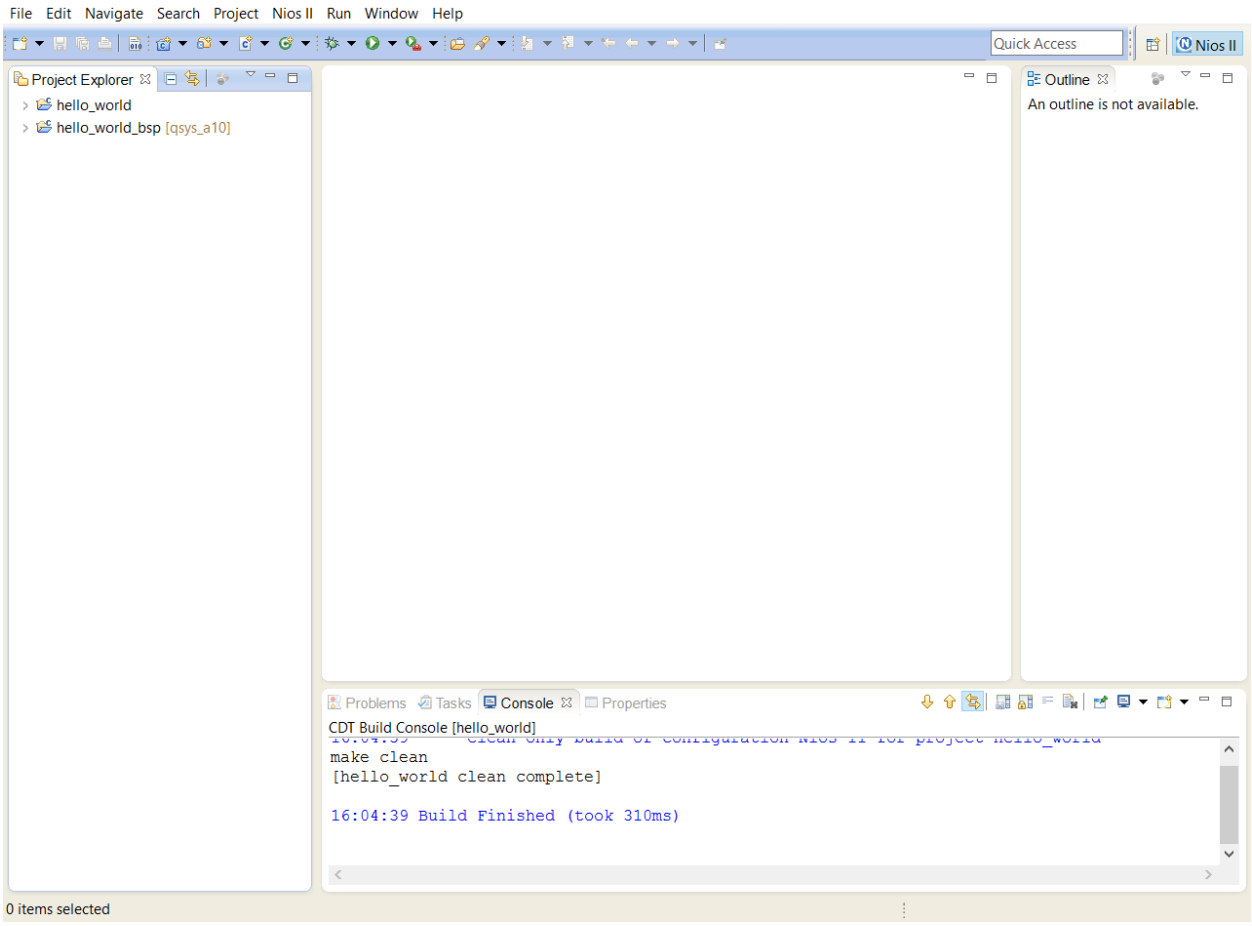
5. Click “Start” to program the .sof into the board.



6. Close the Programmer after successfully program the .sof.

Building Software Design in Nios II Software Build Tools

1. Go to “Tools” and click on “Nios II Software Build Tools for Eclipse” to open the GUI.
2. Create a workspace for the software project.
3. Go to “File” and then to “New” and click on “Nios II Application and BSP from Template”. This will create the application and BSP project for you. You can choose the existing available template to use in your design.
4. Enter the “SOPC Information File Name” and it will automatically detect the processor which is available in your design.
5. Give your Application Project a name. Then click “Finish”. This will automatically create an application with the name provided and a BSP linked together with the application project that has been created.
6. We shall select the “Hello World” template from the given Project Template.



7. Right click on the application project and click “Build Project”. When it is successfully built, it will produce the .elf file in the application folder.
8. Right click on the application project and navigate to “Run As” and click on “Nios II Hardware”.
9. “Hello World” will be printed out at the Console.