Imagination

THE IMAGINATION UNIVERSITY PROGRAMME

RVfpgaEL2 Lab 5
Creating a Vivado Project
Digilent NexysA7 (Nexys4 DDR)

(Revised by Roy Kravitz)

0. INTRODUCTION

To work with and modify the RVfpgaEL2 System, you will need to build a project that includes all the Verilog, SystemVerilog, header, configuration, and text files that define the system. In this lab, we show how to create a Vivado project that targets the SoC used in this course to Digilent's Nexys A7-100T version (i.e. RVfpgaEL2-NexysA7-DDR). By following these same steps, you will be able to modify RVfpgaEL2 -NexysA7-DDR and resynthesize it.

IMPORTANT: If you haven't already done so, install Xilinx's Vivado 2022.2 as explained in the Getting Started Guide.

IMPORTANT: The core frequency of the default RVfpgaEL2 System for a Nexys A7 is 12.5Mhz (see the bitstream provided at ***[RVfpgaNexysA7-DDRPath]/src/rvfpganexys.bit***, which is the one generated in the project that you can find at ***[RVfpgaNexysA7-DDRPath] /Labs/Lab05/project_1/***).

1. Creating a Vivado Project for RVfpgaEL2-NexysA7-DDR

You will use Xilinx's Vivado Design Suite to build the RVfpgaEL2-NexysA7-DDR system using the RTL, the Verilog files that define the system. Follow these steps, detailed below, to build the RVfpgaEL2-NexysA7-DDR system and target it to a NexysA7-100T FPGA board.

Step 1. Open Vivado
Step 2. Create a new RTL project
Step 3. Add the RTL source files and the constraint files
Step 4. Select the XC7A100T-1CSG324C as the target FPGA
Step 5. Set rvfpganexys as Top Module, set *common_defines.vh* as global, set *el2_pdef* both as global and as a System Verilog file, add *boot_main.mem* to the project, include folders and add –*sweep* option
Step 6. Generate Bitstream

Step 1. Open Vivado
If you did not install Vivado on your machine as described in the RVfpga Getting Started Guide, do so now. Be sure to install the board files as well.

Now, run Vivado (in **Linux**, open a terminal and type: vivado; in **Windows**, open Vivado from the Start menu). The Vivado welcome screen will open. Click on Create Project (see Figure 1).

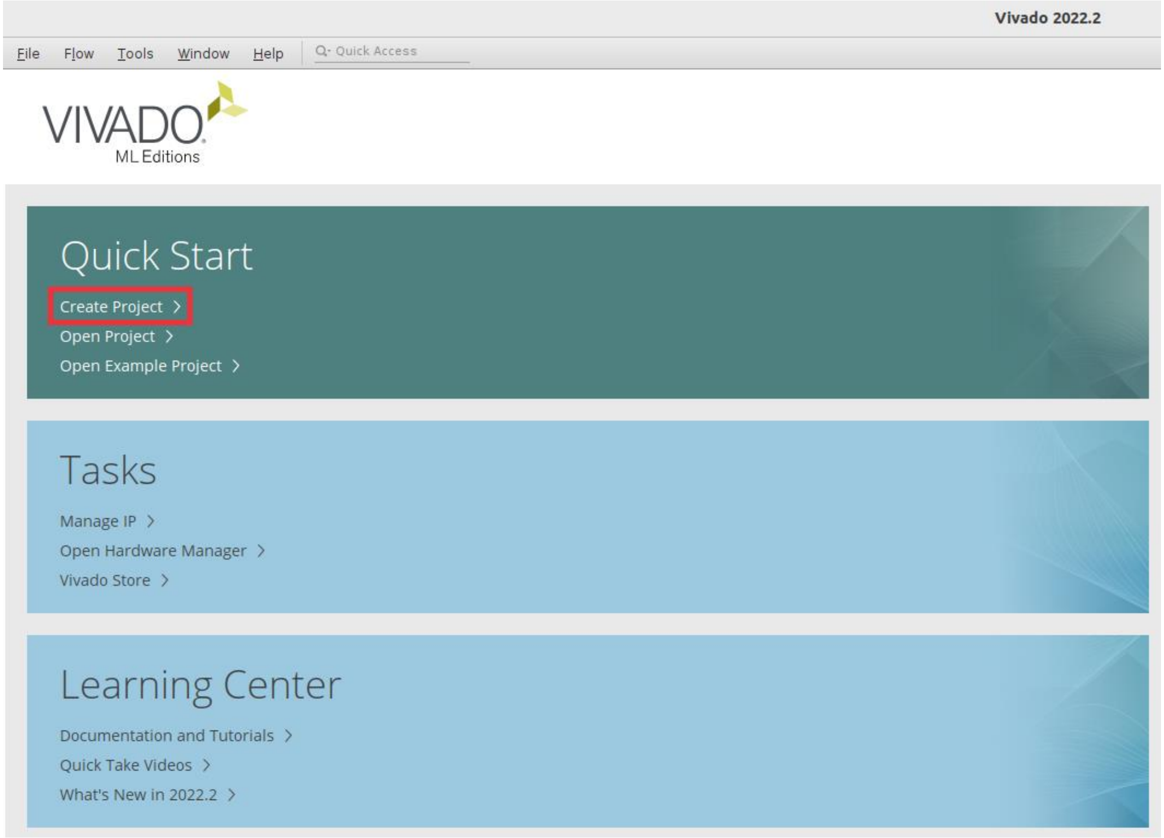
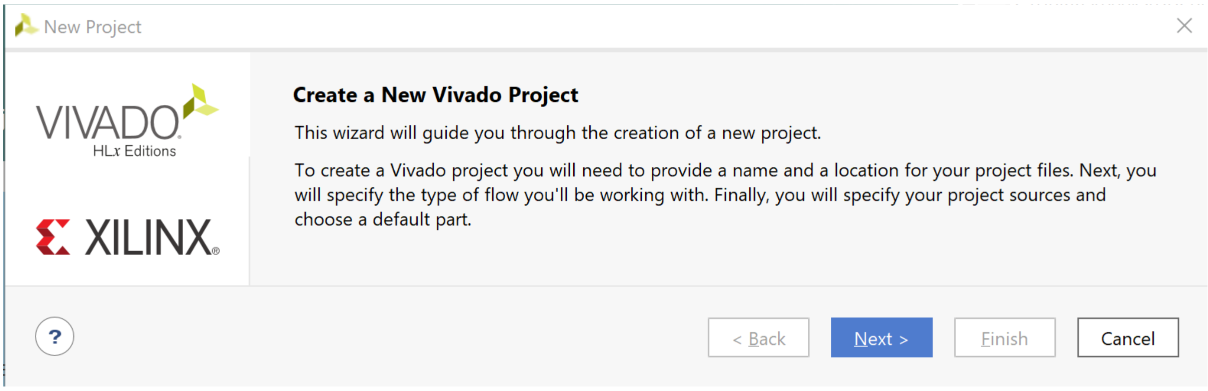


Figure 1. Vivado welcome screen: Create Project

Step 2. Create a new RTL project

The Create a New Vivado Project Wizard will now open (see Figure 2). Click Next.



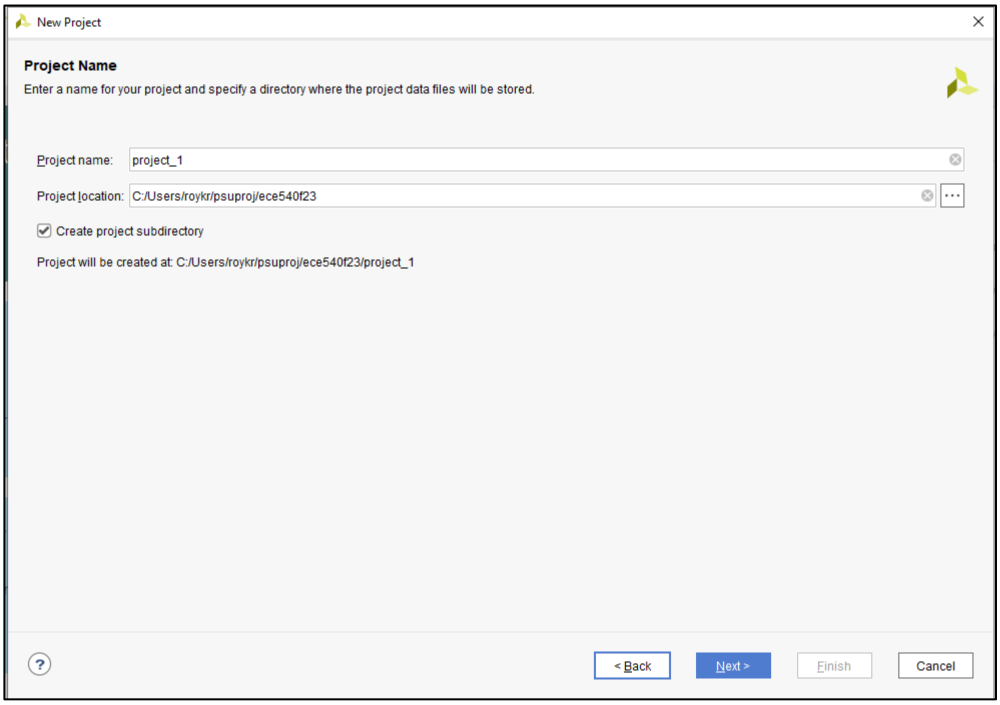
Create a New Vivado Project

This wizard will guide you through the creation of a new project.

To create a Vivado project you will need to provide a name and a location for your project files. Next, you will specify the type of flow you'll be working with. Finally, you will specify your project sources and choose a default part.

Figure 2. Create a New Vivado Project Wizard

Give a name to your project and place it in the *[RVfpgaNexysA7-DDRPath]/Labs/Lab05* folder. Select option *Create project subdirectory*. Then click Next (see Figure 3).



Project Name

Enter a name for your project and specify a directory where the project data files will be stored.

Project name: project_1

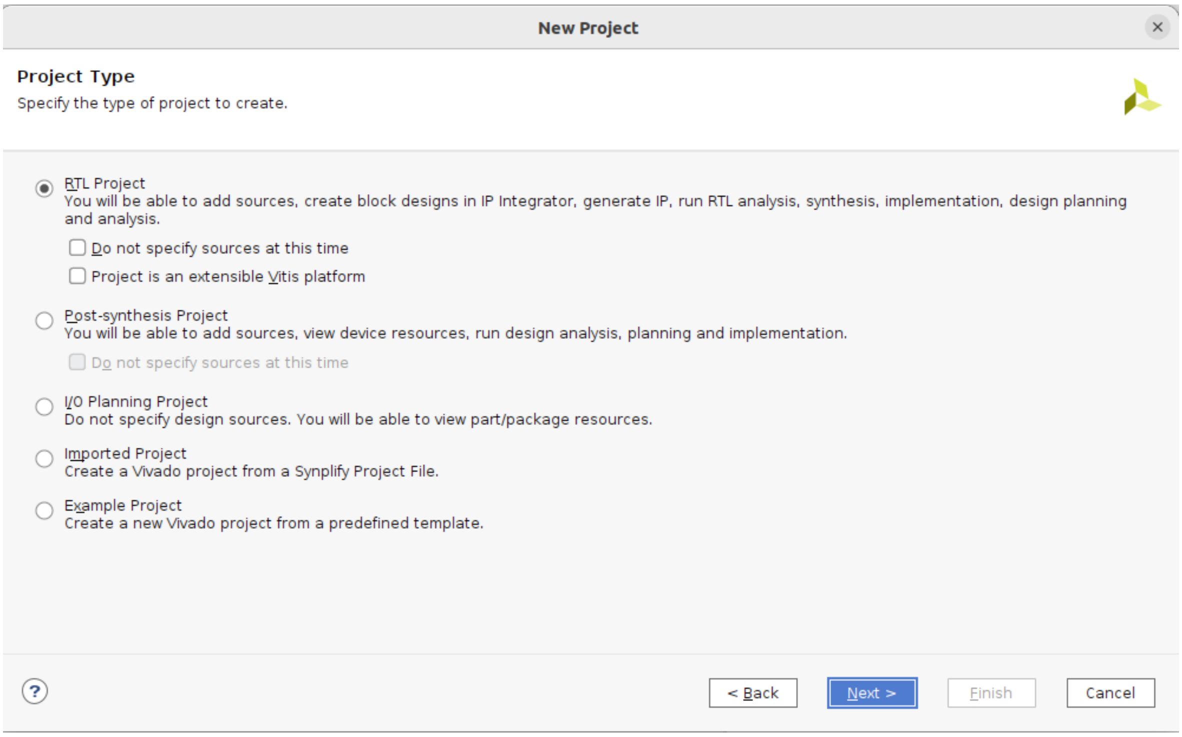
Project location: C:/Users/roykr/psuproj/ece540f23

☑ Create project subdirectory

Project will be created at: C:/Users/roykr/psuproj/ece540f23/project_1

Figure 3. Project Name

Select the project type as RTL Project, and click Next (see Figure 4).



Project Type

Specify the type of project to create.

● RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

□ Do not specify sources at this time

□ Project is an extensible Vitis platform

○ Post-synthesis Project
You will be able to add sources, view device resources, run design analysis, planning and implementation.

□ Do not specify sources at this time

○ I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.

○ Imported Project
Create a Vivado project from a Synplify Project File.

○ Example Project
Create a new Vivado project from a predefined template.

Figure 4. RTL Project

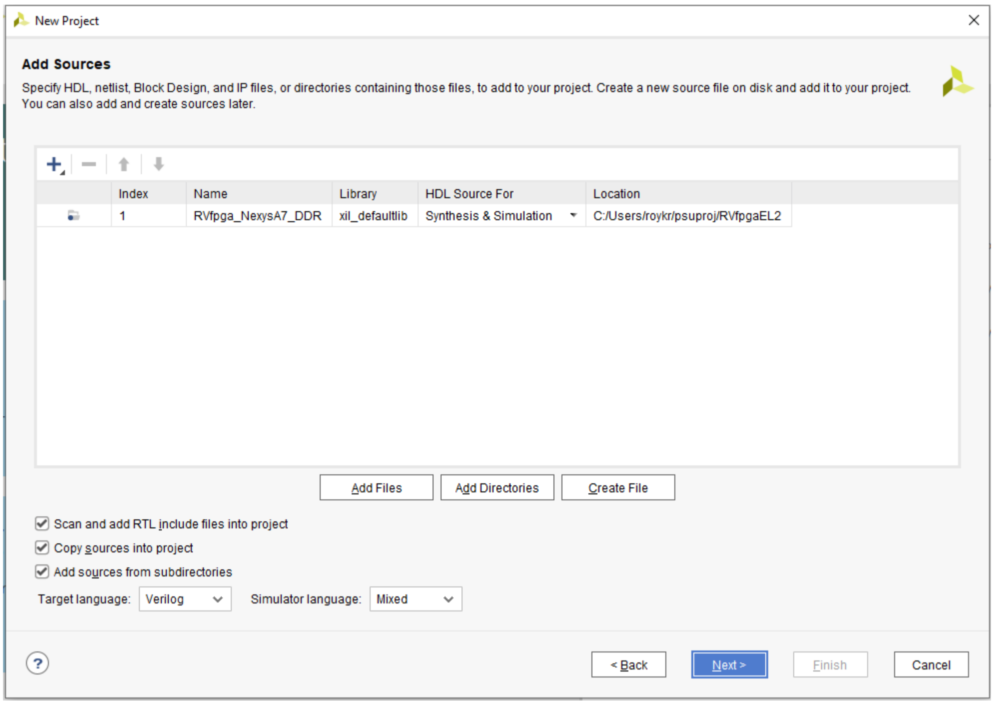
Step 3. Add the RTL source files and the constraint files

In the Add Sources window, click on Add Directories, and select *[RVfpgaNexysA7-DDRPath]/src* (see Figure 5). Make sure both of the following options are selected (as shown in Figure 5):

- Scan and add RTL include files into project

- Add sources from subdirectories

Then click Next.



New Project

Add Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

	Index	Name	Library	HDL Source For	Location
	1	RVfpga_NexysA7_DDR	xil_defaultlib	Synthesis & Simulation	C:/Users/roykr/psuproj/RVfpgaEL2

☑ Scan and add RTL include files into project
☑ Copy sources into project
☑ Add sources from subdirectories

Target language: Verilog   Simulator language: Mixed

Figure 5. Add Sources

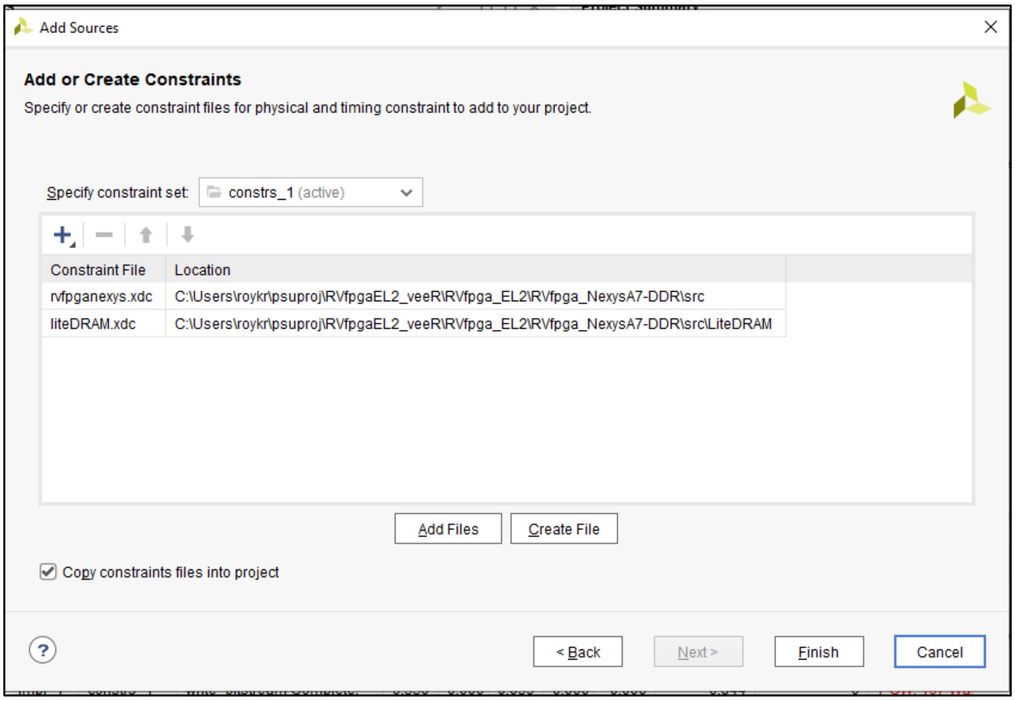
You will now add the constraints for the system. These files map the signal names to the pins on the board. For example, the LEDs on the NexysA7 FPGA board are connected to FPGA pins on the board through traces in the PCB. Vivado must know this so that it maps the correct signal name in the RTL to the correct FPGA pin.

Note that the signal name `o_led` is the name used in the Verilog code to drive the NexysA7 board

In the Add Constraints window, click on Add Files and select the following files (see Figure 6):

[RVfpgaNexysA7-DDRPath]/src/rvfpganexys.xdc
[RVfpgaNexysA7-DDRPath]/src/LiteDRAM/liteDRAM.xdc

Then click Next.



Add Sources

Add or Create Constraints

Specify or create constraint files for physical and timing constraint to add to your project.

Specify constraint set: constrs_1 (active)

Constraint File	Location
rvfpganexys.xdc	C:\Users\roykr\psuproj\RVfpgaEL2_veeR\RVfpga_EL2\RVfpga_NexysA7-DDR\src
liteDRAM.xdc	C:\Users\roykr\psuproj\RVfpgaEL2_veeR\RVfpga_EL2\RVfpga_NexysA7-DDR\src\LiteDRAM

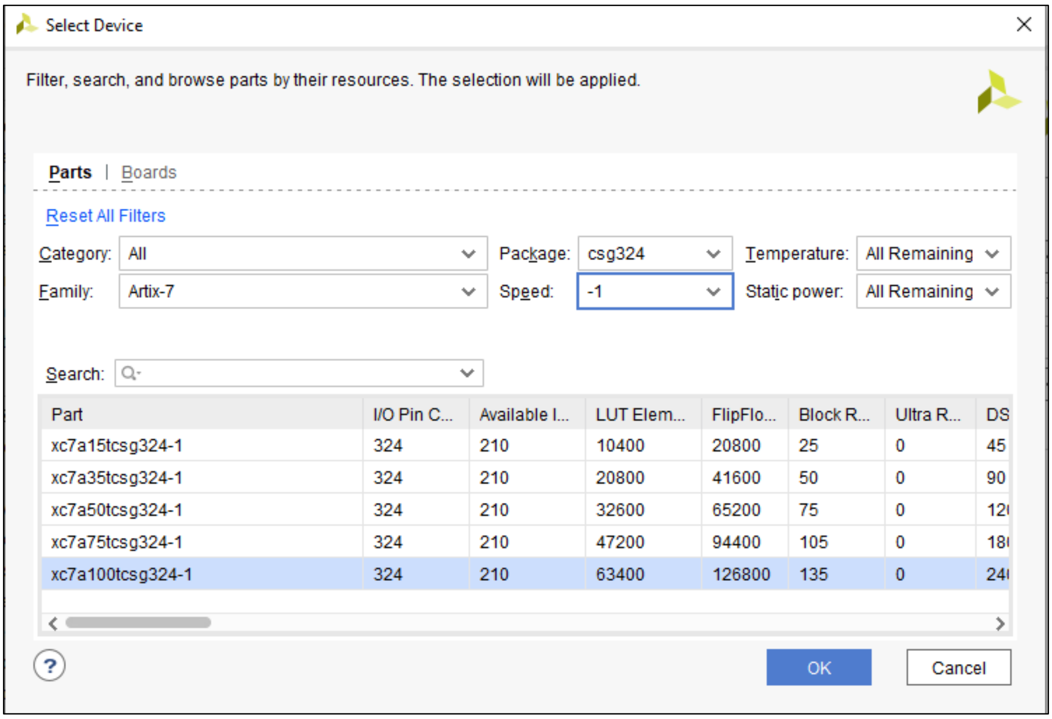
☑ Copy constraints files into project

Figure 6. Add Constraints

Step 4. Select the Series 7 FPGA on the NexysA7 board as the Default Part

In the Default Part window, click on Parts and then select xc7a100tcsg324-1 (see Figure 7). You may use the Search box to narrow down the results.

Click Next.



Select Device

Filter, search, and browse parts by their resources. The selection will be applied.

Parts | Boards

Reset All Filters

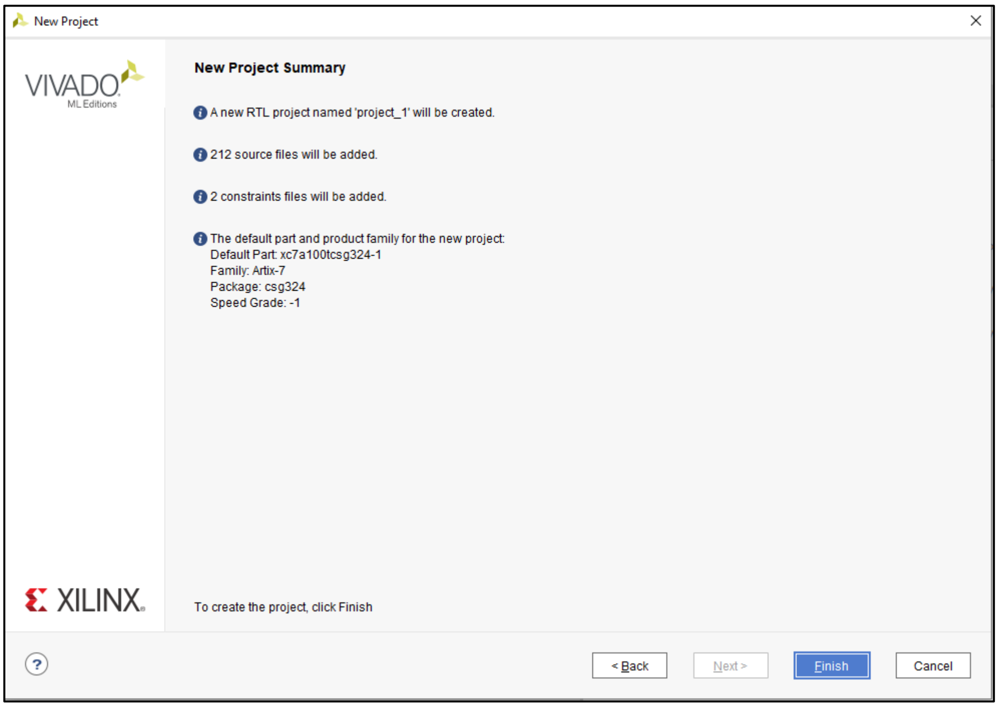
Category: All
Package: csg324
Temperature: All Remaining
Family: Artix-7
Speed: -1
Static power: All Remaining

Search:

Part	I/O Pin C...	Available I...	LUT Elem...	FlipFlo...	Block R...	Ultra R...	DS
xc7a15tcsg324-1	324	210	10400	20800	25	0	45
xc7a35tcsg324-1	324	210	20800	41600	50	0	90
xc7a50tcsg324-1	324	210	32600	65200	75	0	12
xc7a75tcsg324-1	324	210	47200	94400	105	0	18
xc7a100tcsg324-1	324	210	63400	126800	135	0	24

Figure 7. Select target part

In the New Project Summary window, click Finish (see Figure 8).



New Project

New Project Summary

A new RTL project named 'project_1' will be created.

212 source files will be added.

2 constraints files will be added.

The default part and product family for the new project:
Default Part: xc7a100tcsg324-1
Family: Artix-7
Package: csg324
Speed Grade: -1

To create the project, click Finish

Figure 8. Project Summary

Step 5. Set rvfpganexys.sv as Top Module, set common_defines.vh as global, set *el2_pdef* as global and System Verilog, add *boot_main.mem* to the project, include folders and add *–sweep* option.

Set rvfpganexys as Top Module: You will now set the rvfpganexys module as the top module. In the Sources pane, scroll down under Design Sources, right-click on the rvfpganexys module, and select Set as Top (see Figure 8). You can also find the rvfpganexys module by typing this name in the search box. This sets rvfpganexys as the highest-level module in the hierarchy and the target to be synthesized and implemented onto the FPGA. After setting rvfpganexys as the top module, the hierarchy will update.

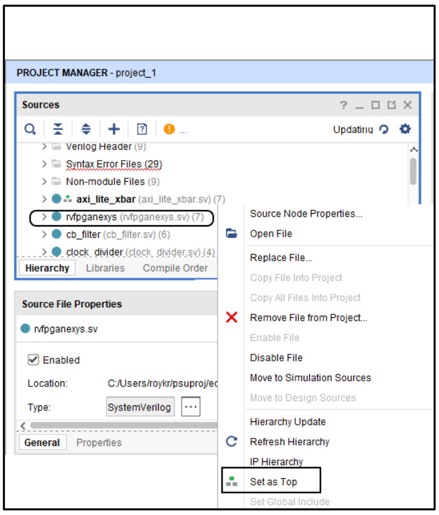
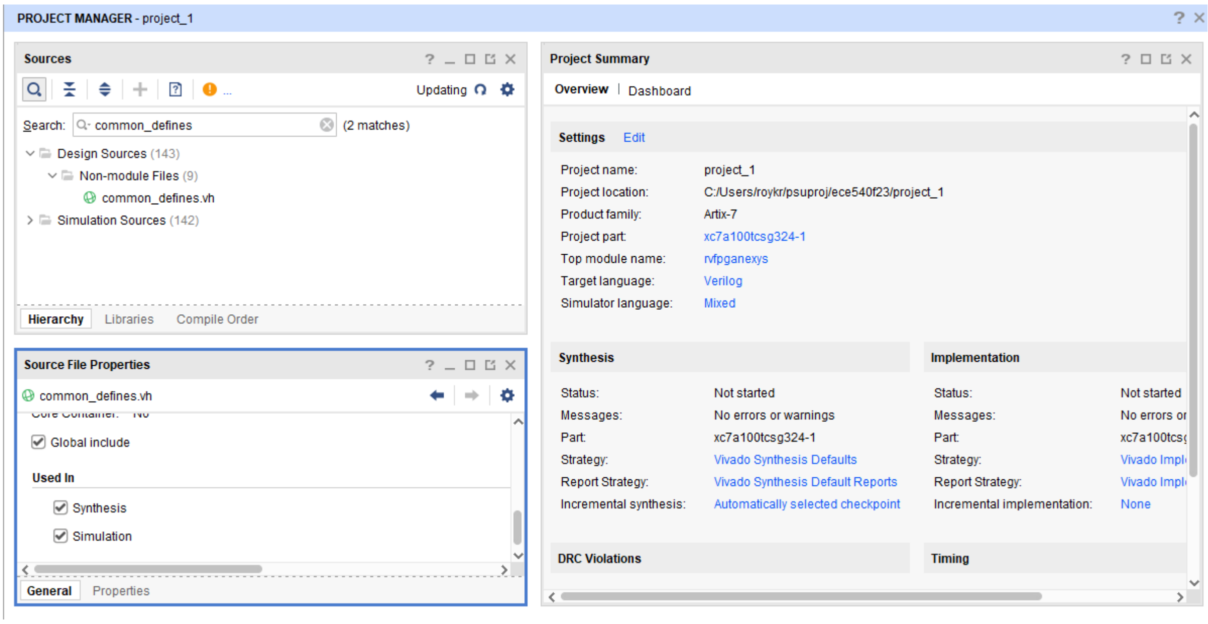


Figure 8. Set rvfpganexys as top module

<u>**Set files *common_defines.vh* and *el2_pdef* as Global Include and *el2_pdef* as System Verilog**</u>: Now, still in the Sources pane under Design Sources, expand the Non-modules file group and click on *common_defines.vh*. The properties of the file will then open in the Source File Properties pane, just below the Sources pane. Click on Global Include to tick that box (see **Figure 9**). The hierarchy will now update and include that file in Design Sources/Global Include.



PROJECT MANAGER - project_1

Sources — Updating

Search: common_defines (2 matches)

- Design Sources (143)
  - Non-module Files (9)
    - common_defines.vh
- Simulation Sources (142)

Hierarchy | Libraries | Compile Order

Source File Properties — common_defines.vh

Core Container: No

☑ Global include

Used In

☑ Synthesis

☑ Simulation

General | Properties

Project Summary — Overview | Dashboard

Settings Edit

Project name:	project_1
Project location:	C:/Users/roykr/psuproj/ece540f23/project_1
Product family:	Artix-7
Project part:	xc7a100tcsg324-1
Top module name:	rvfpganexys
Target language:	Verilog
Simulator language:	Mixed

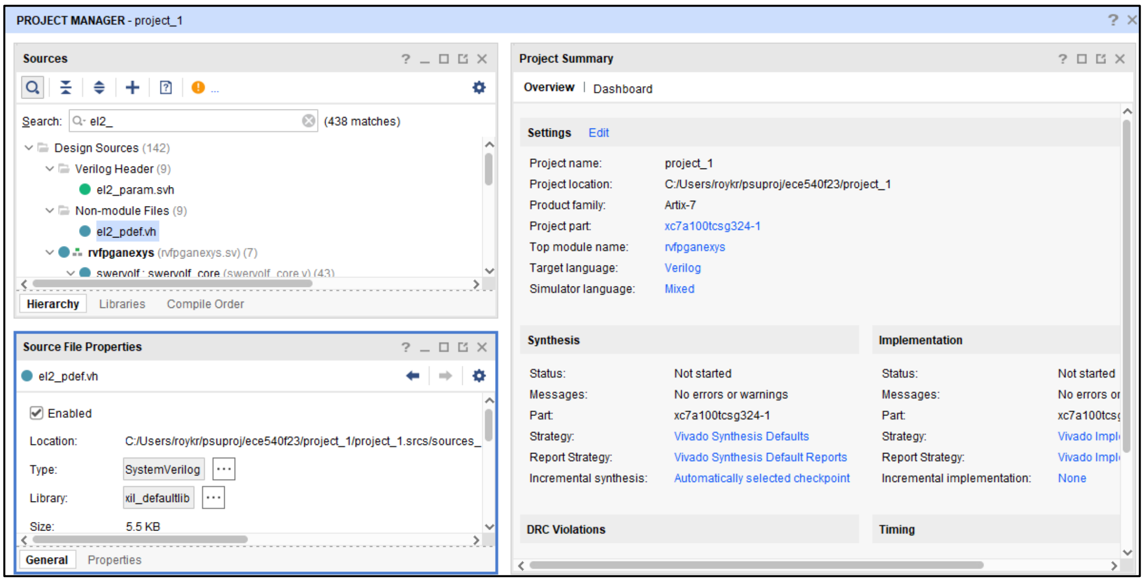
Synthesis		Implementation	
Status:	Not started	Status:	Not started
Messages:	No errors or warnings	Messages:	No errors or
Part:	xc7a100tcsg324-1	Part:	xc7a100tcs
Strategy:	Vivado Synthesis Defaults	Strategy:	Vivado Impl
Report Strategy:	Vivado Synthesis Default Reports	Report Strategy:	Vivado Impl
Incremental synthesis:	Automatically selected checkpoint	Incremental implementation:	None

DRC Violations | Timing

Figure 9. Set *common_defines.vh* as a Global include file

Do the same with file *el2_pdef*. Then, on the same Window (Source File Properties), set the type of this file as System Verilog (see Figure 10).

Note (RK): you may have to set the file type to SystemVerilog after the hierarchy updates to include el2_pdef.vh in the Global Include folder.



PROJECT MANAGER - project_1

Sources

Search: el2_ (438 matches)

- Design Sources (142)
  - Verilog Header (9)
    - el2_param.svh
  - Non-module Files (9)
    - el2_pdef.vh
  - rvfpganexys (rvfpganexys.sv) (7)
    - swervolf : swervolf_core (swervolf_core.v) (43)

Hierarchy | Libraries | Compile Order

Source File Properties — el2_pdef.vh

☑ Enabled

Location:	C:/Users/roykr/psuproj/ece540f23/project_1/project_1.srcs/sources_
Type:	SystemVerilog
Library:	xil_defaultlib
Size:	5.5 KB

General | Properties

Project Summary — Overview | Dashboard

Settings Edit

Project name:	project_1
Project location:	C:/Users/roykr/psuproj/ece540f23/project_1
Product family:	Artix-7
Project part:	xc7a100tcsg324-1
Top module name:	rvfpganexys
Target language:	Verilog
Simulator language:	Mixed

Synthesis		Implementation	
Status:	Not started	Status:	Not started
Messages:	No errors or warnings	Messages:	No errors or
Part:	xc7a100tcsg324-1	Part:	xc7a100tcs
Strategy:	Vivado Synthesis Defaults	Strategy:	Vivado Impl
Report Strategy:	Vivado Synthesis Default Reports	Report Strategy:	Vivado Impl
Incremental synthesis:	Automatically selected checkpoint	Incremental implementation:	None

DRC Violations | Timing

Figure 10. Set the type of *el2_pdf.vh* as System Verilog.

<u>Add *boot_main.mem* to the project</u>: In the Flow Navigator pane, click on Add Sources, leave the default option (“Add or create design sources”), and click on Add Files (see Figure 11). Navigate to *[RVfpgaNexysA7-DDRPath]/src/VeeRwolf/BootROM/sw* and select *boot_main.mem* (as shown in Figure 11). The hierarchy will update and include that file in Design Sources/Memory File.

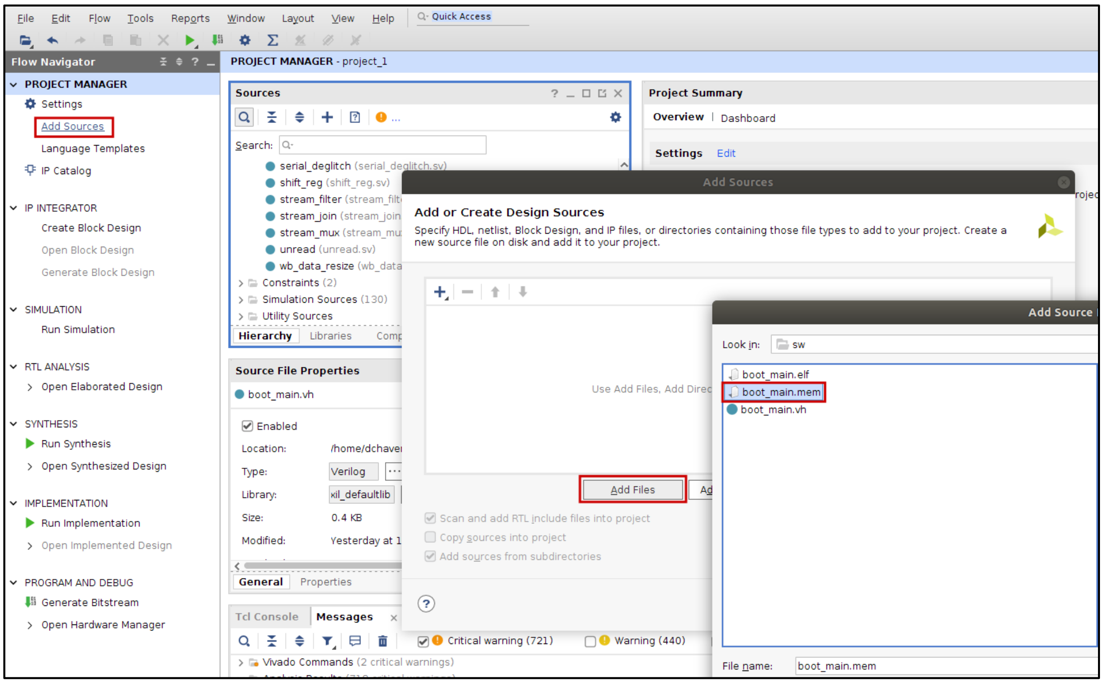
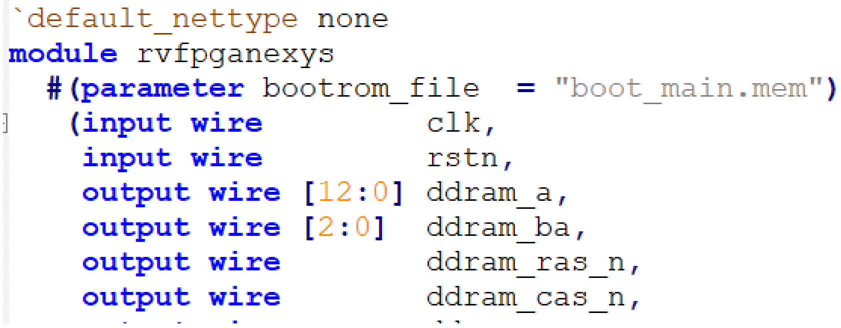


Figure 11. Add Memory File *boot_main.mem*

This file (*boot_main.mem*) is used for initializing the Boot ROM of our SoC by invoking it as a parameter in **file *[RVfpgaNexysA7-DDRPath]/****src/rvfpganexys.sv*:



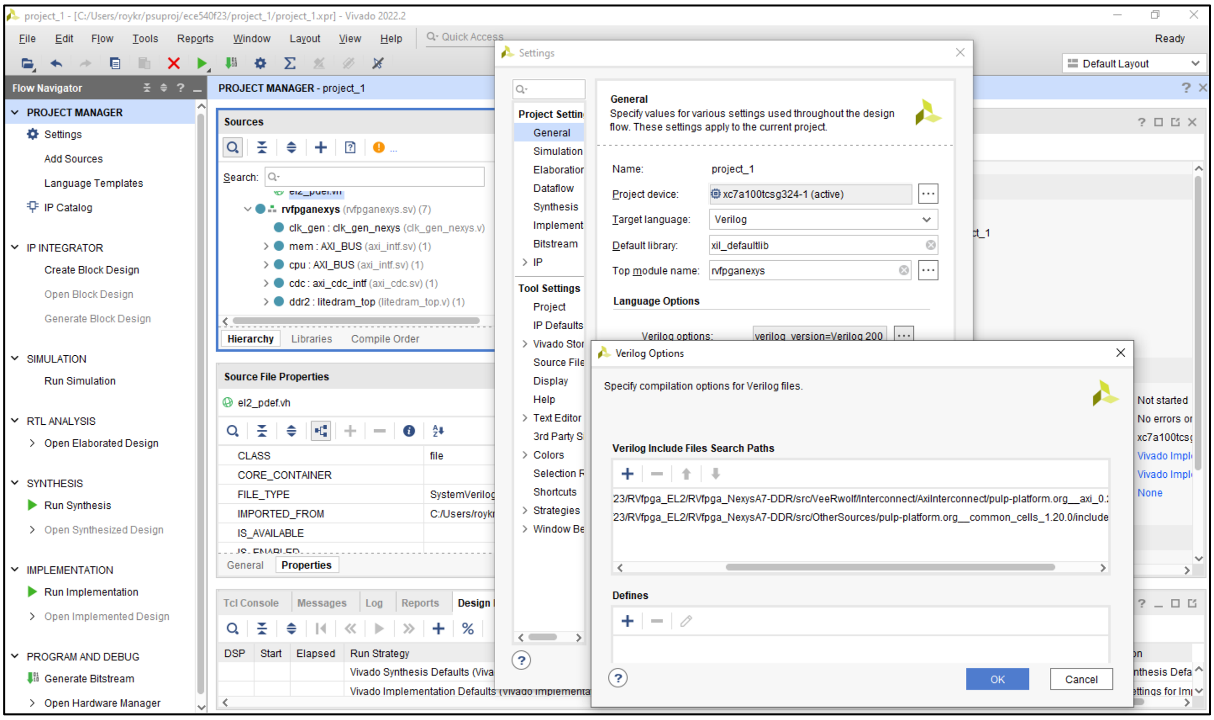
```
`default_nettype none
module rvfpganexys
  #(parameter bootrom_file  = "boot_main.mem")
   (input wire          clk,
    input wire          rstn,
    output wire [12:0]  ddram_a,
    output wire [2:0]   ddram_ba,
    output wire         ddram_ras_n,
    output wire         ddram_cas_n,
```

Section 6.A in the Getting Started Guide contains more information about this file.

<u>Include folders</u>: Include two folders for the Pulp Platform (see Figure 12). In the Flow Navigator pane click on *Settings*, and on the window that opens click on *General* and then on *Verilog options* (…). In the new window, add the two following include directories by clicking on + and browsing to the directories:

```
[RVfpgaNexysA7-DDRPath]/src/VeeRwolf/Interconnect/AxiInterconnect/pulp-platform.org__axi_0.25.0/include
[RVfpgaNexysA7-DDRPath]/src/OtherSources/pulp-platform.org__common_cells_1.20.0/include
```

Note (RK): Including these folders should eliminate any syntax errors once the hierarchy has been updated.



Settings – General

Specify values for various settings used throughout the design flow. These settings apply to the current project.

Name: project_1
Project device: xc7a100tcsg324-1 (active)
Target language: Verilog
Default library: xil_defaultlib
Top module name: rvfpganexys

Language Options
Verilog options: verilog_version=Verilog 200

Verilog Options

Specify compilation options for Verilog files.

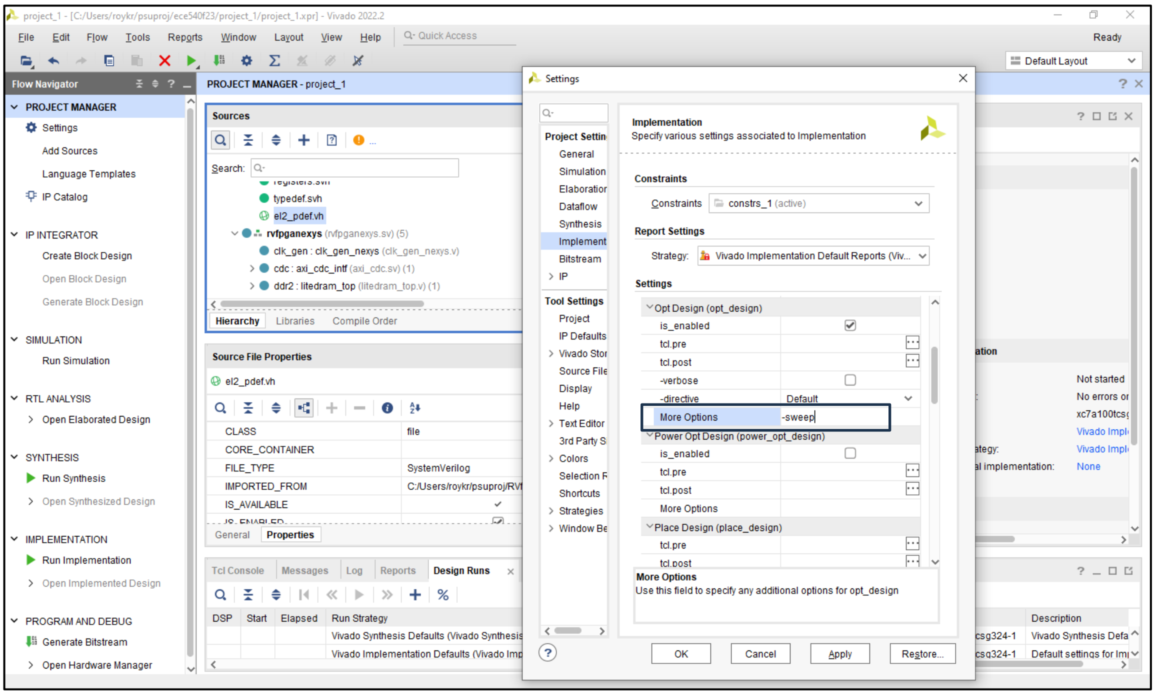
Verilog Include Files Search Paths

23/RVfpga_EL2/RVfpga_NexysA7-DDR/src/VeeRwolf/Interconnect/AxiInterconnect/pulp-platform.org__axi_0.:
23/RVfpga_EL2/RVfpga_NexysA7-DDR/src/OtherSources/pulp-platform.org__common_cells_1.20.0/include

Defines

Figure 12. Include folders for the Pulp Platform

Add *-sweep*: Finally, add the –sweep option to the **Implementation – Opt Design**. (See Figure 13).


project_1 - [C:/Users/roykr/psuproj/ece540f23/project_1/project_1.xpr] - Vivado 2022.2
File Edit Flow Tools Reports Window Layout View Help
Settings
Project Settings
General
Simulation
Elaboration
Dataflow
Synthesis
Implementation
Bitstream
IP
Tool Settings
OK
Cancel
Apply
Restore...

Implementation

Specify various settings associated to Implementation

Constraints
Constraints: constrs_1 (active)

Report Settings
Strategy: Vivado Implementation Default Reports (Viv...

Settings

Opt Design (opt_design)
is_enabled ☑
tcl.pre
tcl.post
-verbose ☐
-directive Default
More Options -sweep

Power Opt Design (power_opt_design)
is_enabled ☐
tcl.pre
tcl.post
More Options

Place Design (place_design)
tcl.pre
tcl.post

More Options
Use this field to specify any additional options for opt_design

Figure 13. Add the –sweep option

Step 6. Generate Bitstream

Now Click on Flow → Generate Bitstream as shown in Figure 14. This process typically takes around 20 minutes, depending on the speed of your computer.

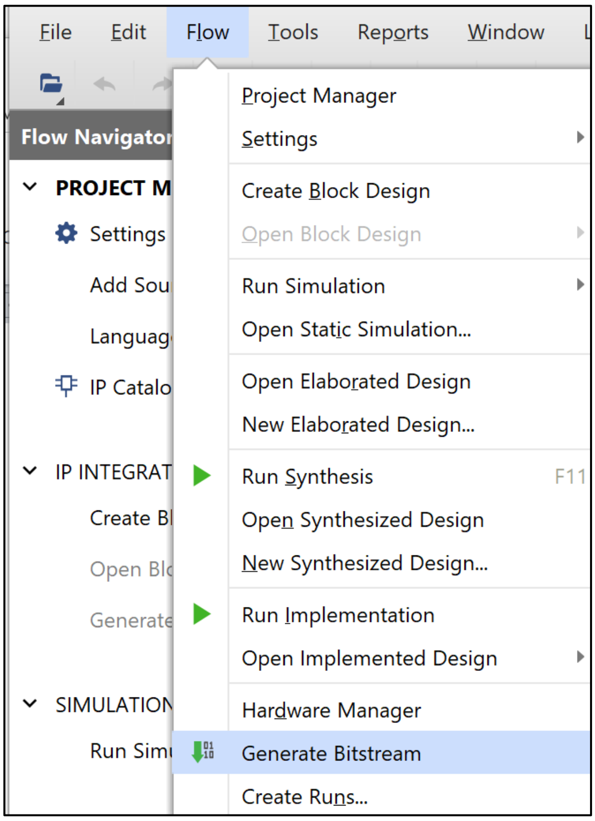
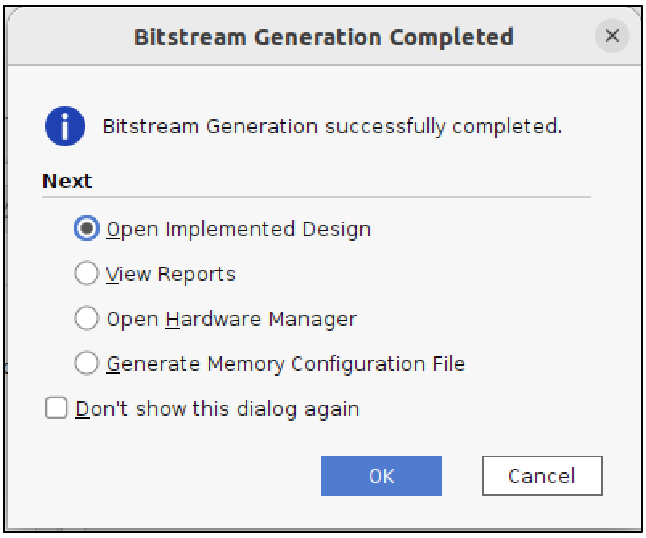


Figure 14. Generate Bitstream

After the bitstream has been generated a window will pop up as shown in Figure 15. Click on the ✕ button in the top-right corner to close the window.



Bitstream Generation Completed

Bitstream Generation successfully completed.

Next

● Open Implemented Design

○ View Reports

○ Open Hardware Manager

○ Generate Memory Configuration File

□ Don't show this dialog again

Figure 15. Bitstream Generation Completed

Note (RK): Finished with 260 synthesis warnings, 3 implementation warnings and 52 DRC warnings. Bitstream generation was successful.

Now that you have built the RVfpgaEL2-NexysA7-DDR system yourself, you will be able to rebuild RVfpgaEL2-NexysA7-DDR after you make modifications to it in Labs 6-9. You can now also use RVfpgaEL2-NexysA7-DDR system you just built to download and run programs on it.

IMPORTANT: You can find the bitstream just generated by Vivado in folder:

```
  [RVfpgaNexysA7-DDRPath]/Labs/Lab05/project_2/<project name>/impl_1
```

It is recommended that you use Catapult SDK to download RVfpgaEL2-NexysA7-DDR onto the Nexys A7 board, as described in Labs 1-4 and in the RVfpgaEL2 Getting Started Guide (GSG) in detail. As also described in the GSG and Labs 1-4, after downloading the RVfpgaEL2-NexysA7-DDR system onto the FPGA on the Nexys A7 board, you will use Catapult SDK to download and run/debug programs on RVfpgaEL2-NexysA7-DDR board.