THE IMAGINATION UNIVERSITY PROGRAMME

RVfpgaEL2 Lab 5

Creating a Vivado Project

Boolean board

(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 Real Digital's Boolean FPGA board, -50T version (i.e. RVfpgaEL2-Boolean). By following these same steps, you will be able to modify RVfpgaEL2 -Boolean 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 Boolean board is 12.5Mhz (see the bitstream provided at ***[RVfpgaBooleanPath]/src/rvfpgaboolean.bit***, which is the one generated in the project that you can find at ***[RVfpgaBooleanPath]/Labs/Lab05/project_1/***).
We also provide a second bitstream (***[RVfpgaBooleanPath]/Labs/Lab05/project_2/project_2.runs/impl_1/rvfpgaboolean.bit***) of the RVfpgaEL2 System for a Boolean board that uses a core frequency of 25MHz. Although this SoC works in all our exercises, we don't set it as our default configurations as it provides some timing violations when the bitstream is created in Vivado.
The specific configuration used for the core is specified in Lab 11.

1. Creating a Vivado Project for RVfpgaEL2-Boolean

You will use Xilinx's Vivado Design Suite to build the RVfpgaEL2-Boolean system using the RTL, the Verilog files that define the system. Follow these steps, detailed below, to build the RVfpgaEL2-Boolean system and target it to a Boolean 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 *XC7S50-CSGA324* as the target FPGA
Step 5. Set rvfpgaboolean 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).

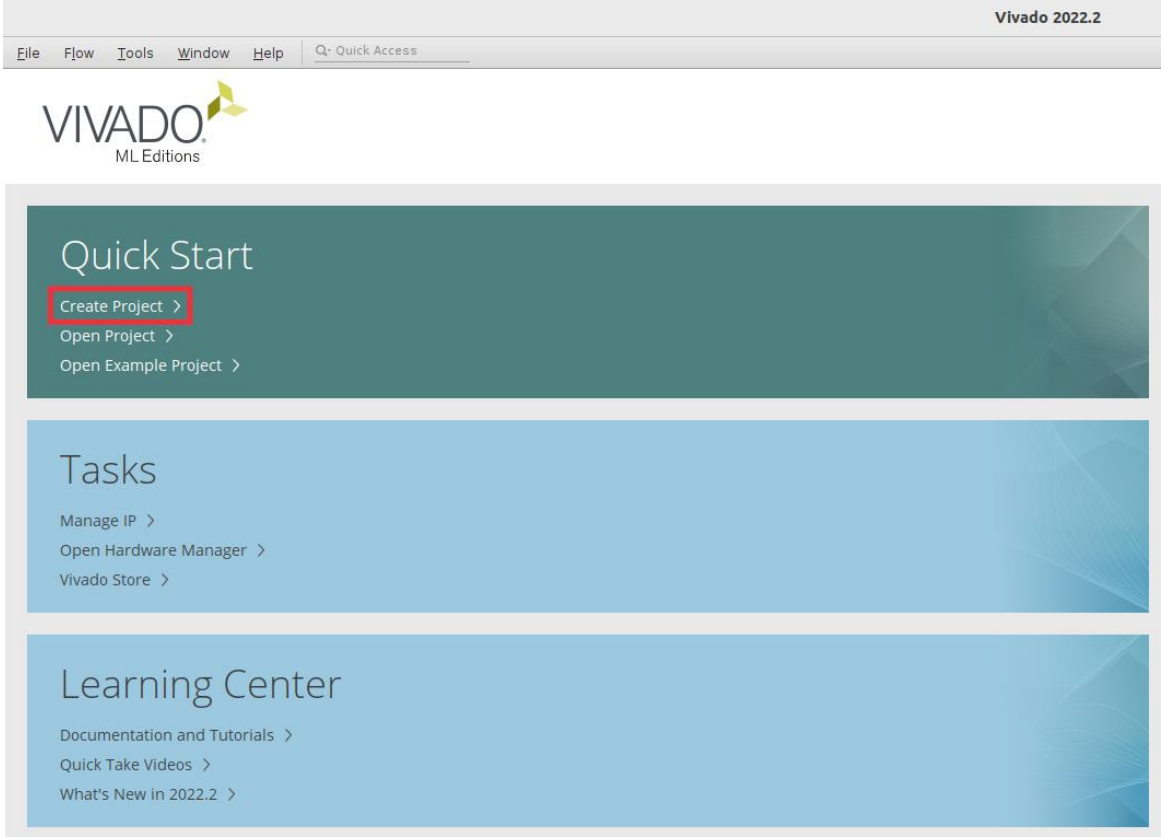

Figure 1. Vivado welcome screen: Create Project

Step 2. Create a new RTL project

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

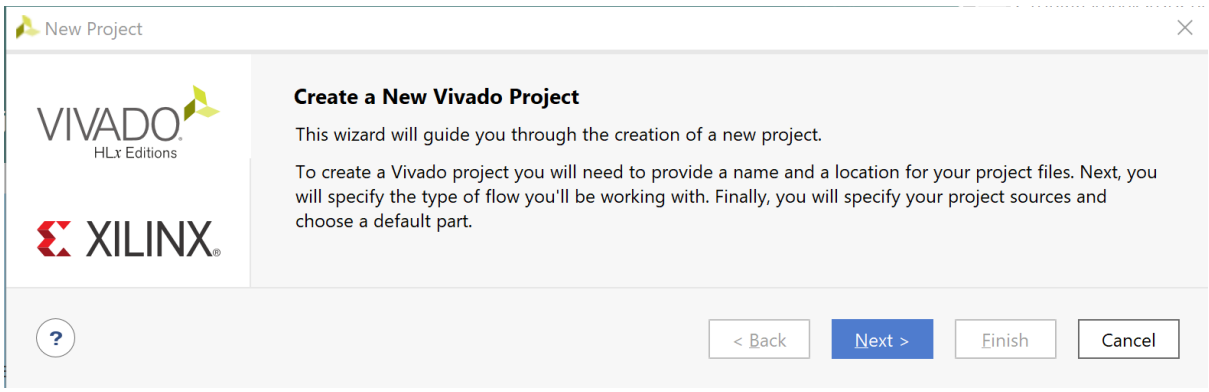

Figure 2. Create a New Vivado Project Wizard

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

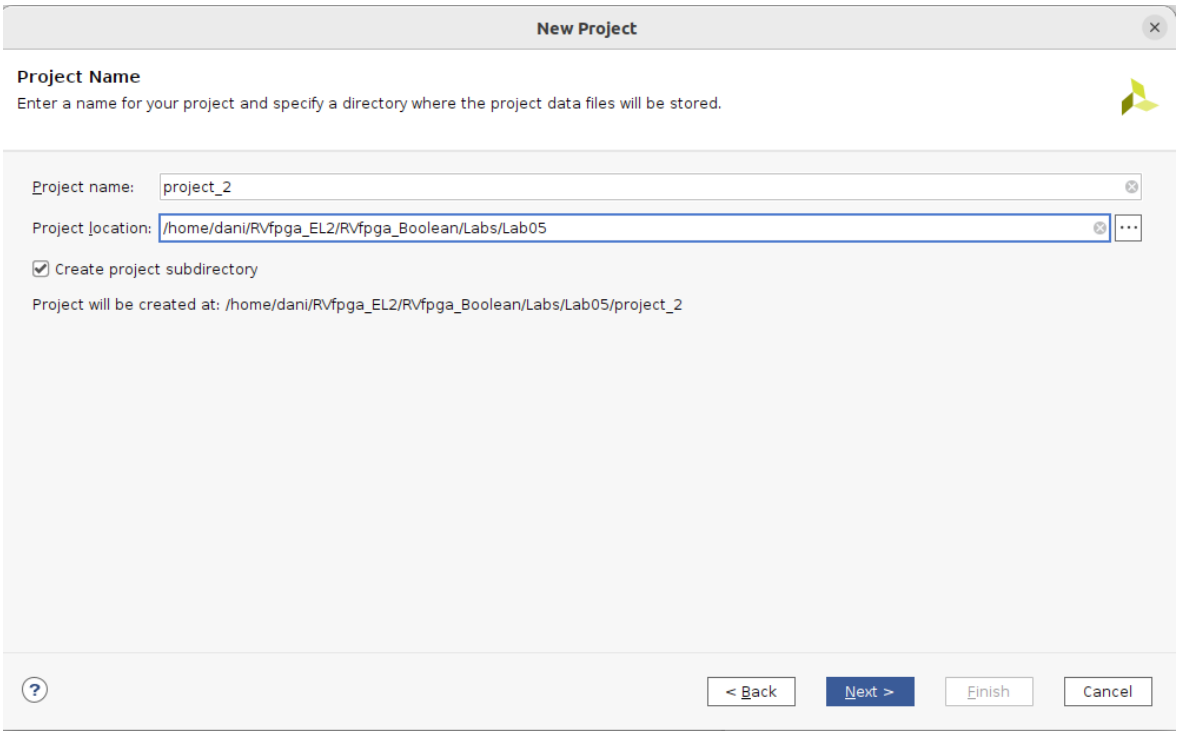

Figure 3. Project Name

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

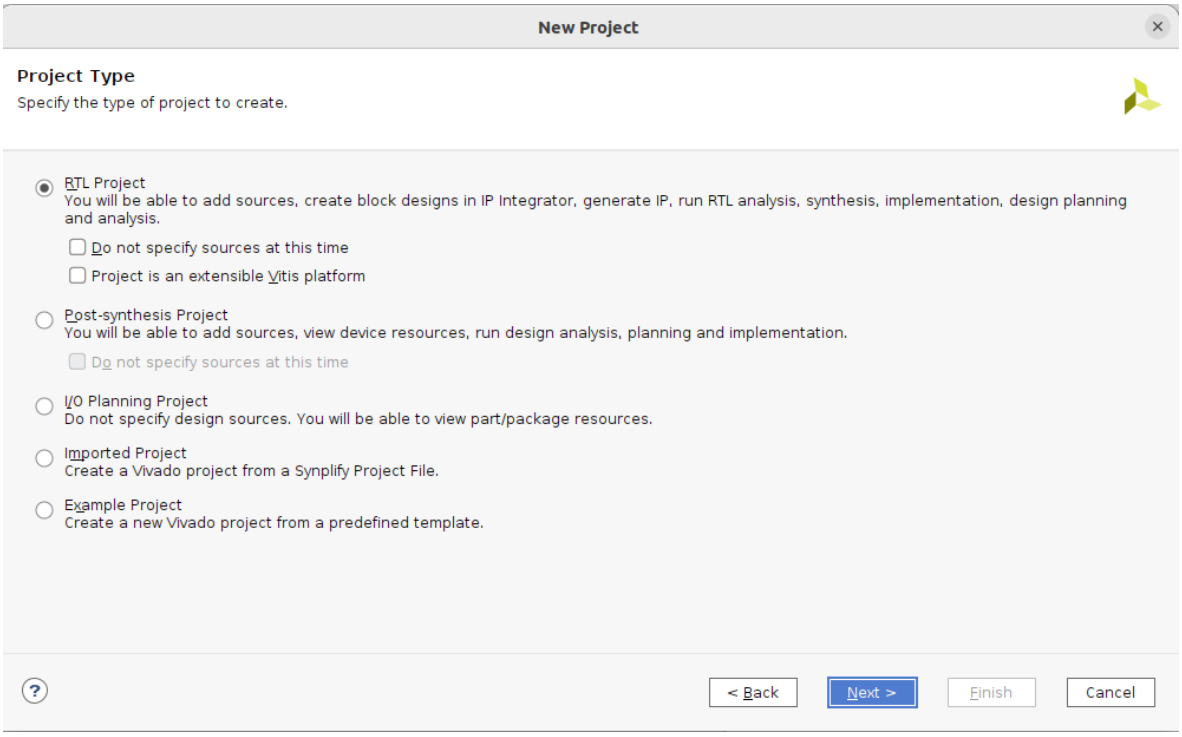

Figure 4. RTL Project

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

In the Add Sources window, click on Add Directories, and select *[RVfpgaBooleanPath]/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.

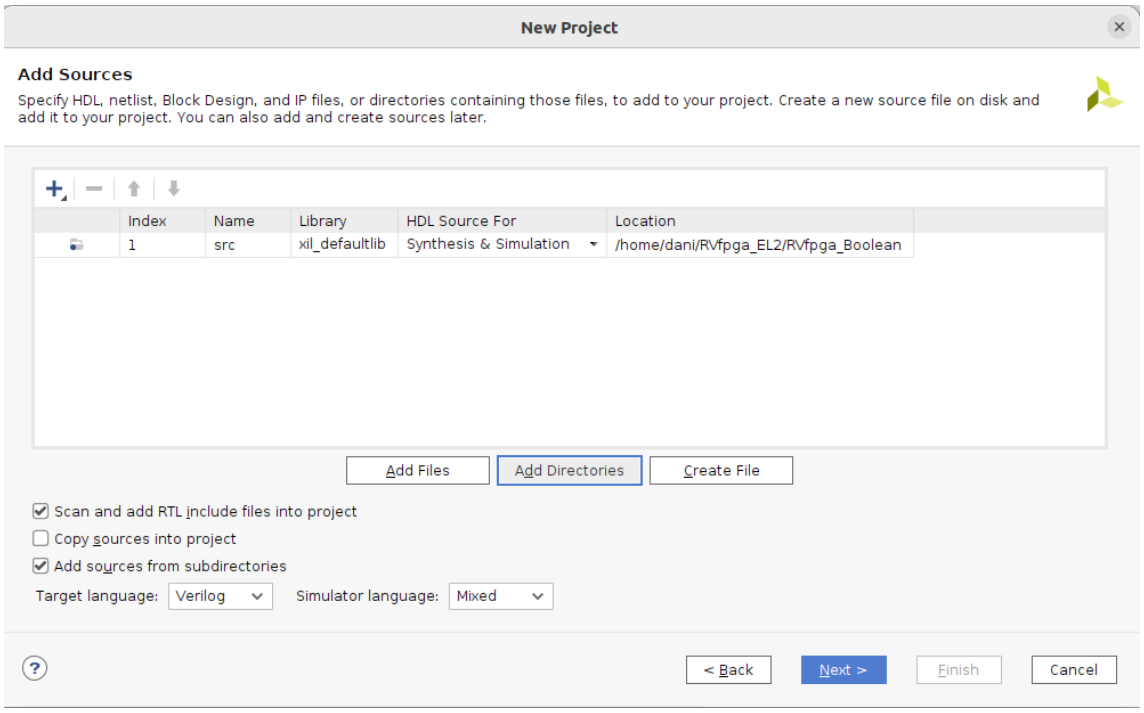

Figure 5. Add Sources

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 Boolean 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 Boolean board's LEDs.

In the Add Constraints window, click on Add Files and select the following file (see Figure 6):
[RVfpgaBooleanPath]/src/rvfpgaboolean.xdc

Then click Next.

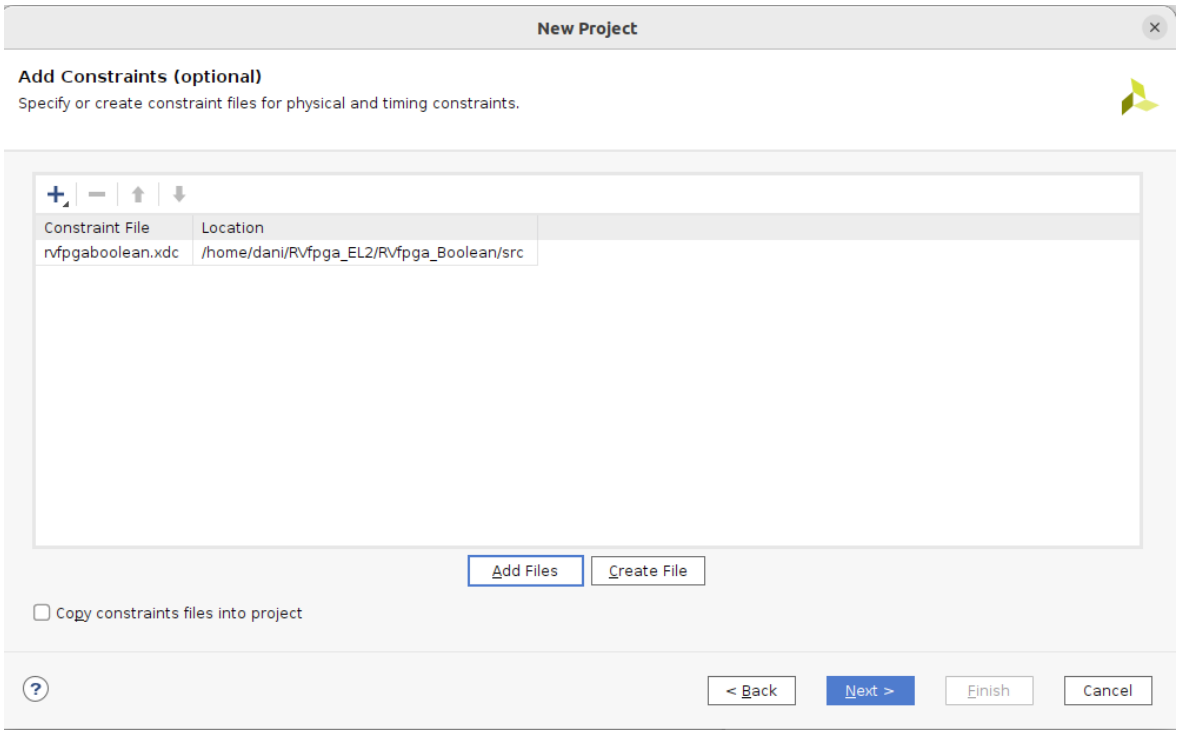

Figure 6. Add Constraints

Step 4. Select the Spartan 7 FPGA on the Boolean board as the Default Part
In the Default Part window, click on Parts and then select xc7s50csga324-1 (see Figure 7). You may use the Search box to narrow down the results.

Click Next.

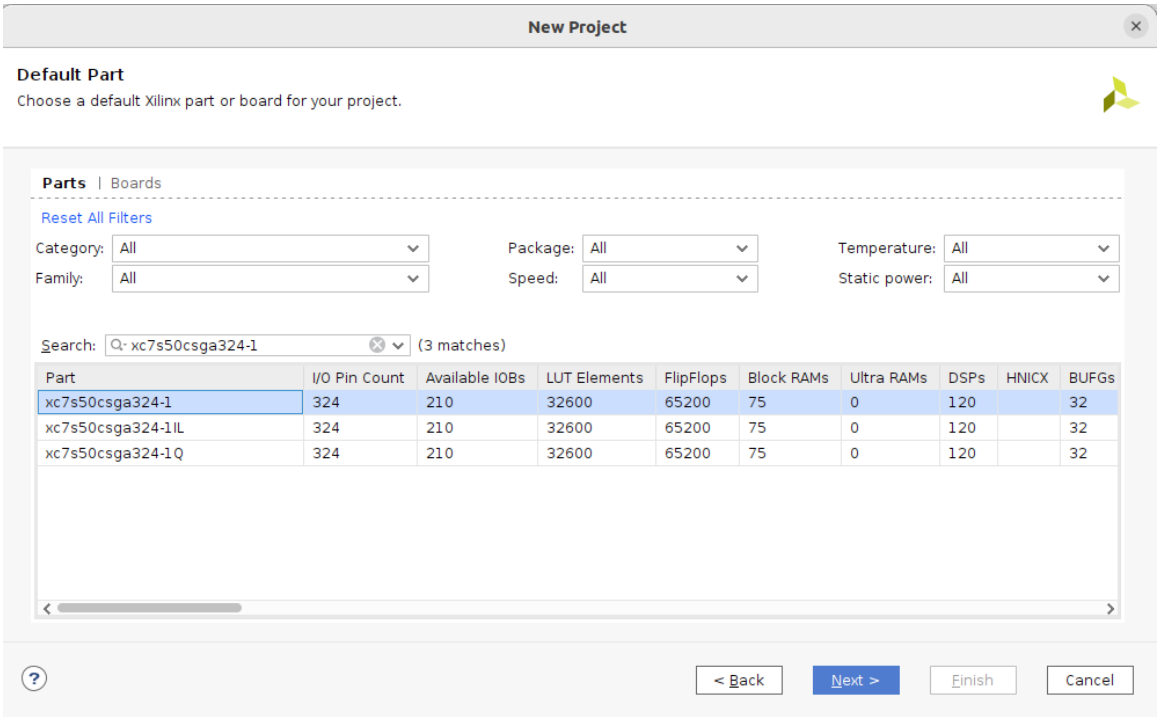

Figure 7. Select target part

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

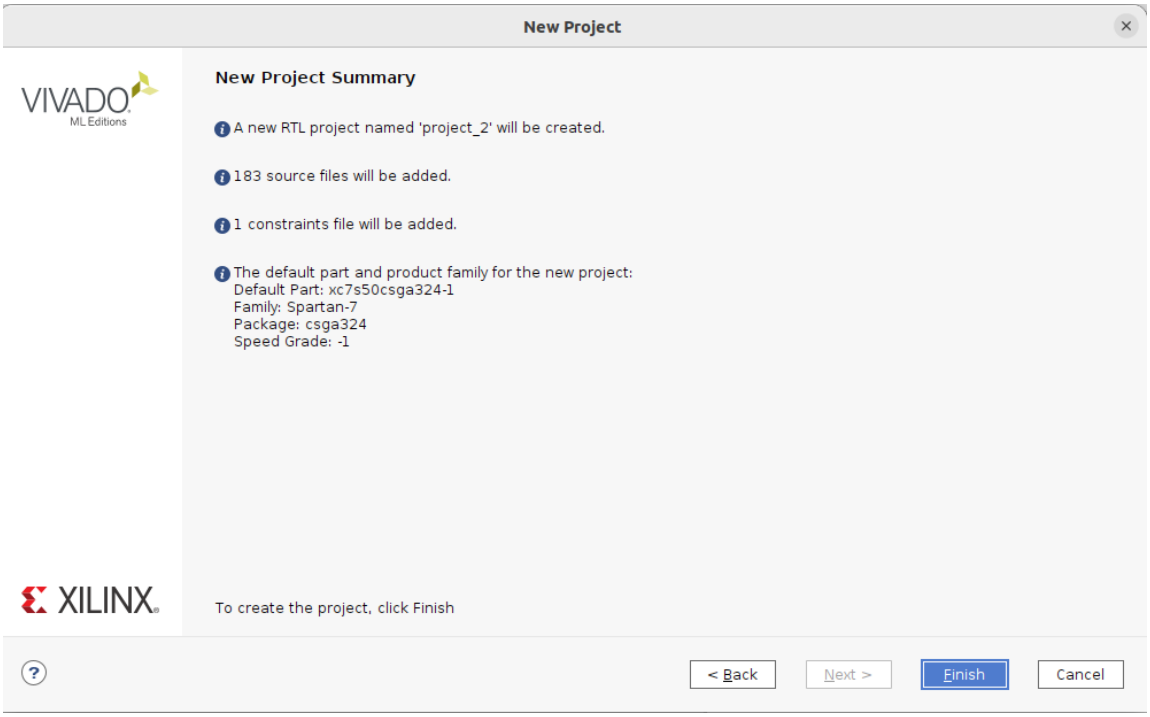

Figure 8. New Project Summary Window

Step 5. Set rvfpgaboolean 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.

<u>Set rvfpgaboolean as Top Module</u>: You will now set the rvfpgaboolean module as the top module. In the Sources pane, scroll down under Design Sources, right-click on the rvfpgaboolean module, and select Set as Top (see Figure 9). You can also find the rvfpgaboolean module by typing this name in the search box, as shown. This sets rvfpgaboolean as the highest-level module in the hierarchy and the target to be synthesized and implemented onto the FPGA. After setting rvfpgaboolean as the top module, the hierarchy will update.

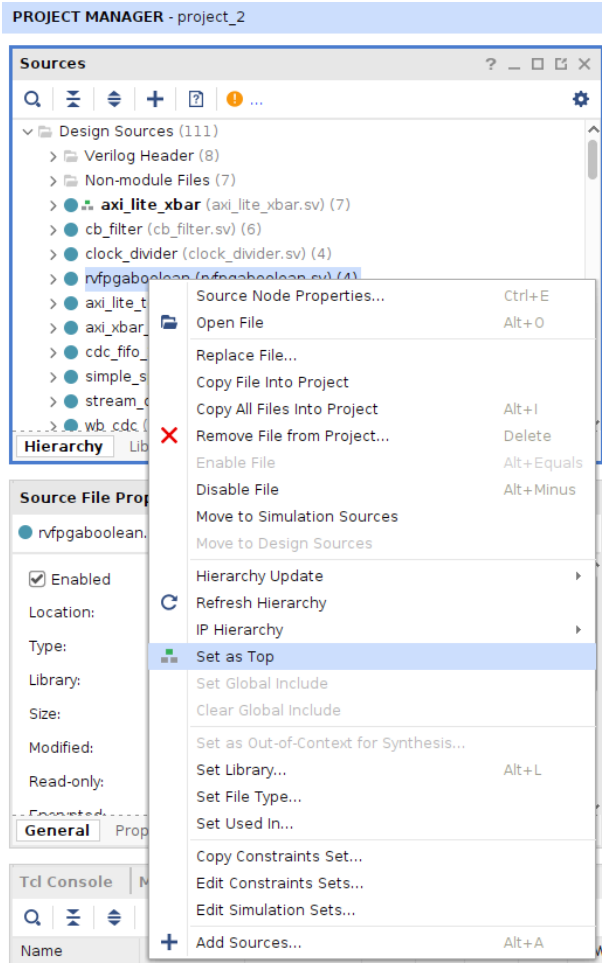

Figure 9. Set rvfpgaboolean as top module

<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 10**). The hierarchy will now update and include that file in Design Sources/Global Include.

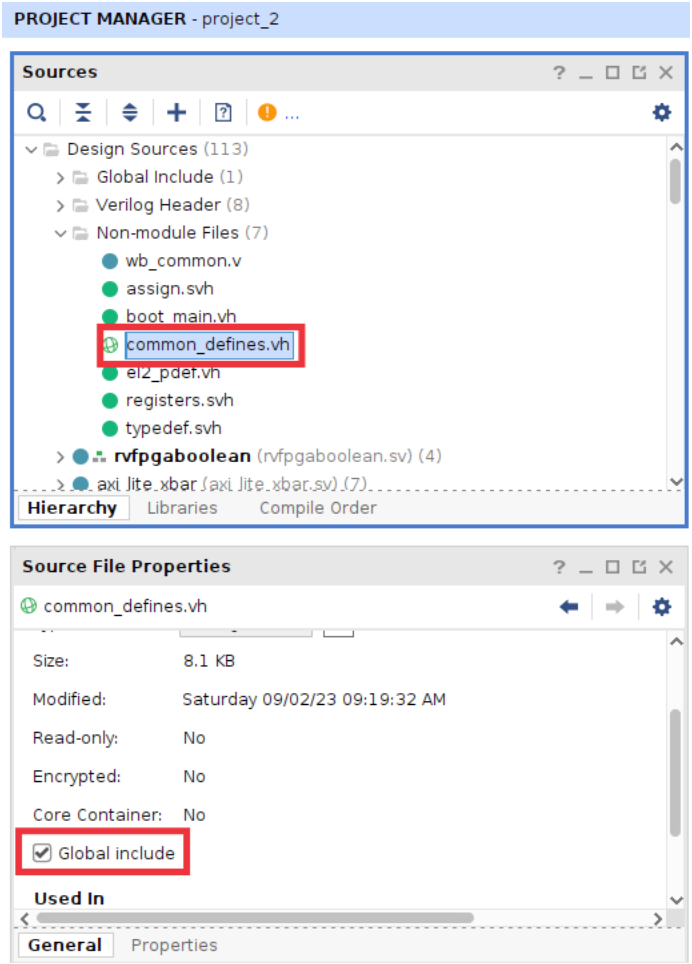

Figure 10. Set *common_defines.vh* as a Global include file

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 11).

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.

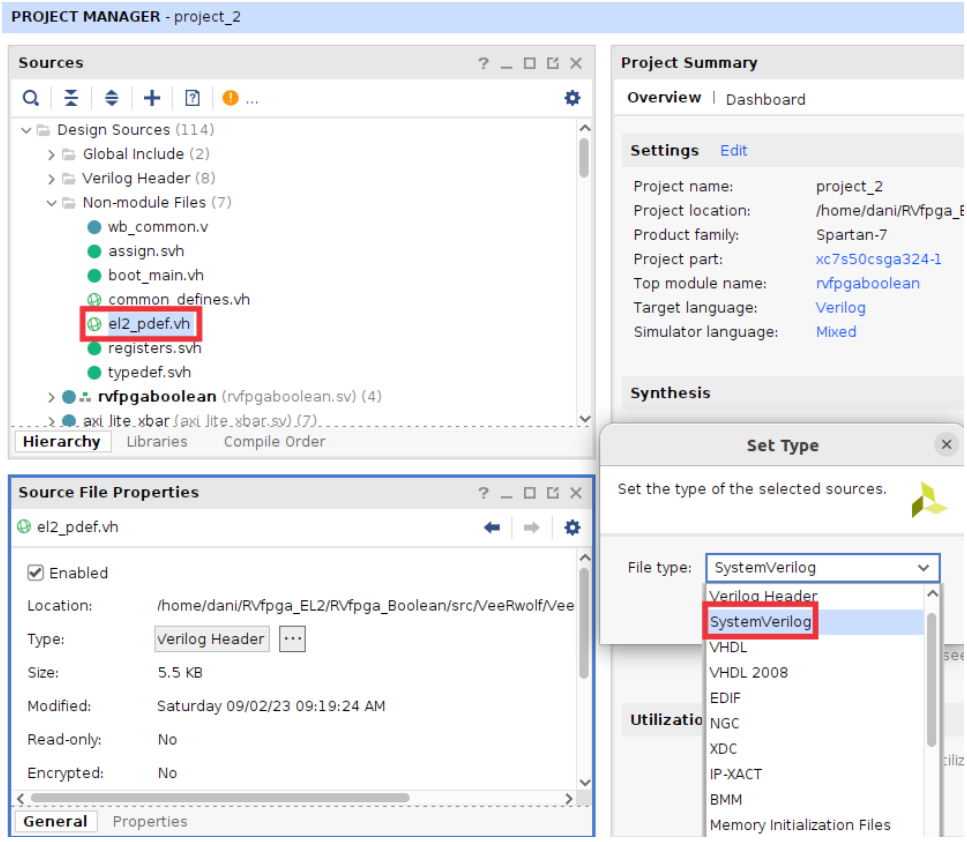

Figure 11. Set the type of *el2_pdf.vh* as System Verilog.

<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 12). Navigate to *[RVfpgaBooleanPath]/src/VeeRwolf/BootROM/sw* and select *boot_main.mem* (as shown in Figure 12). The hierarchy will update and include that file in Design Sources/Memory File.

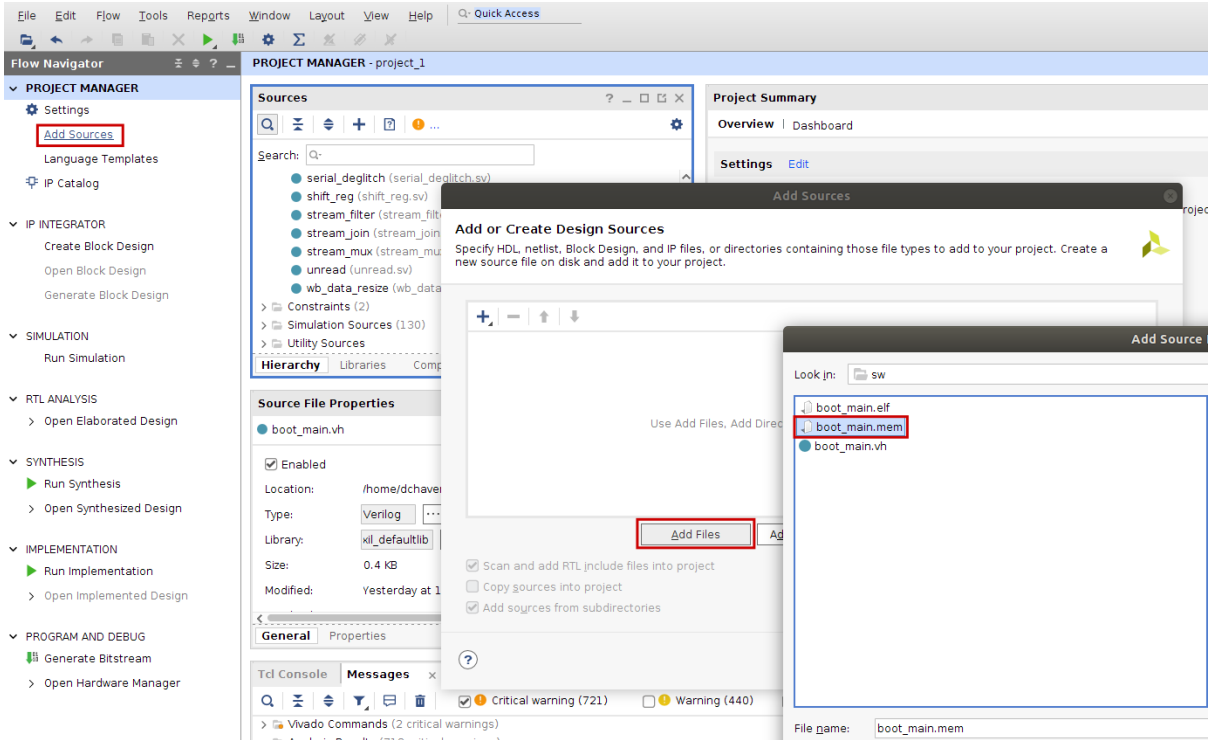

Figure 12. Add Memory File *boot_main.mem*

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

```
module rvfpgaboolean
  #(parameter bootrom_file = "boot_main.mem")
   (input wire          clk,
    input wire          i_uart_rx,
    output wire         o_uart_tx,
    inout  wire [15:0]   i_sw,
    output reg  [15:0]   o_led,
```

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 13). 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:

```
[RVfpgaBooleanPath]/src/VeeRwolf/Interconnect/AxiInterconnect/pulp-platform.org__axi_0.25.0/include
[RVfpgaBooleanPath]/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.

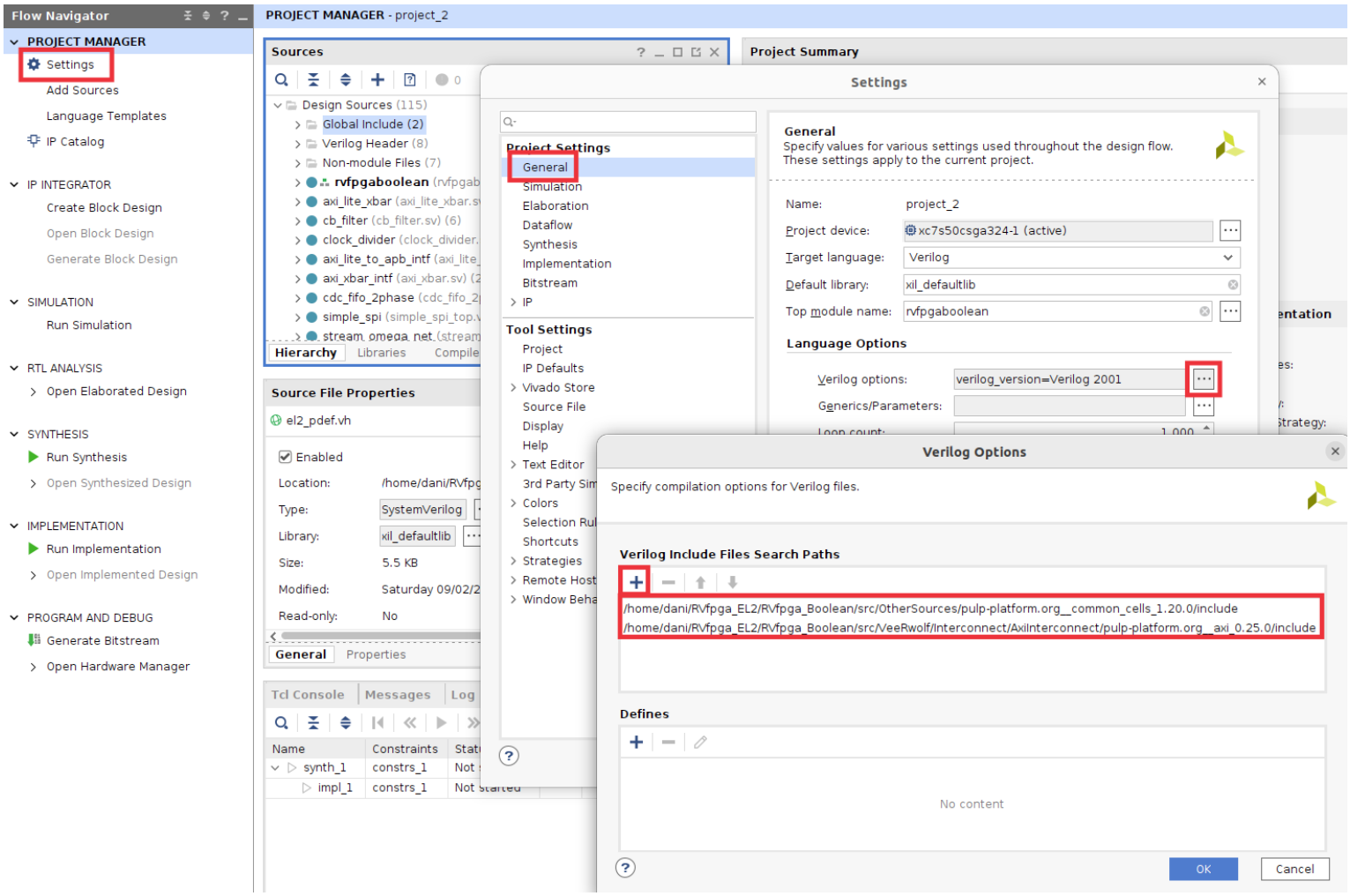

Figure 13. Include folders for the Pulp Platform

Add -sweep: Finally, add the –sweep option to the **Implementation – Opt Design**. See Figure 14.

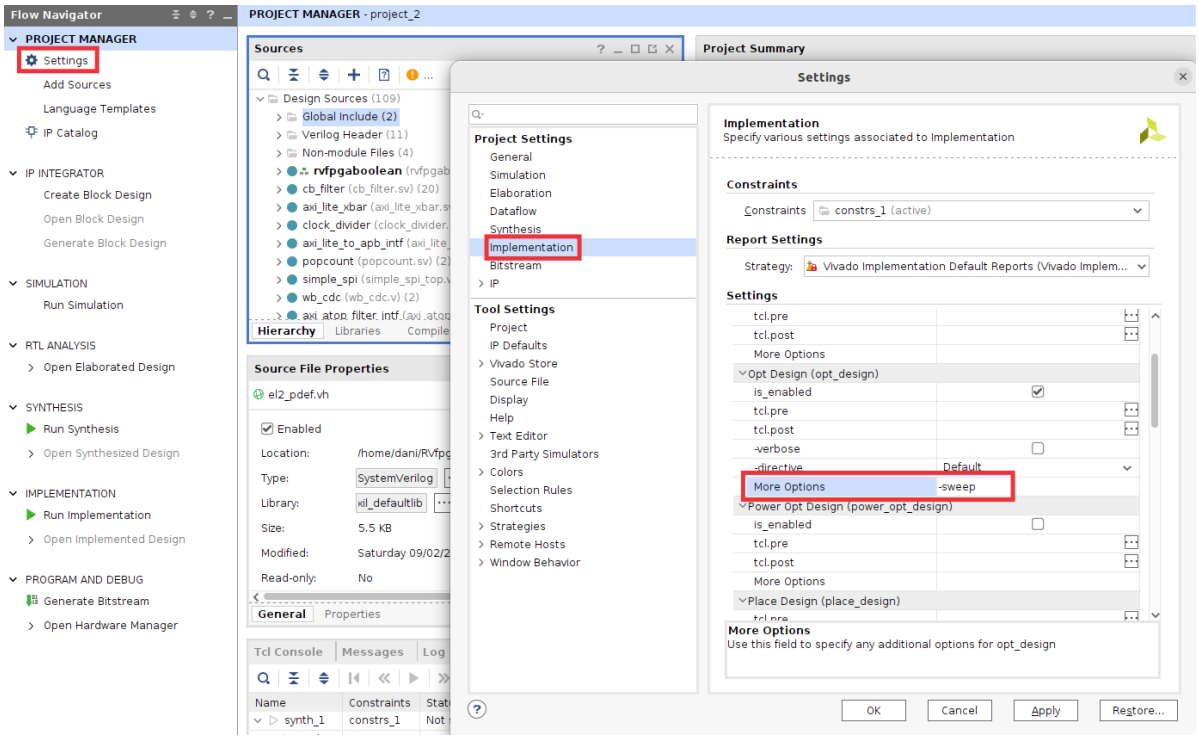

Figure 14. Add the –sweep option

Step 6. Generate Bitstream

Now Click on Flow → Generate Bitstream as shown in Figure 15. This process typically takes around 20 minutes, depending on the speed of your computer.

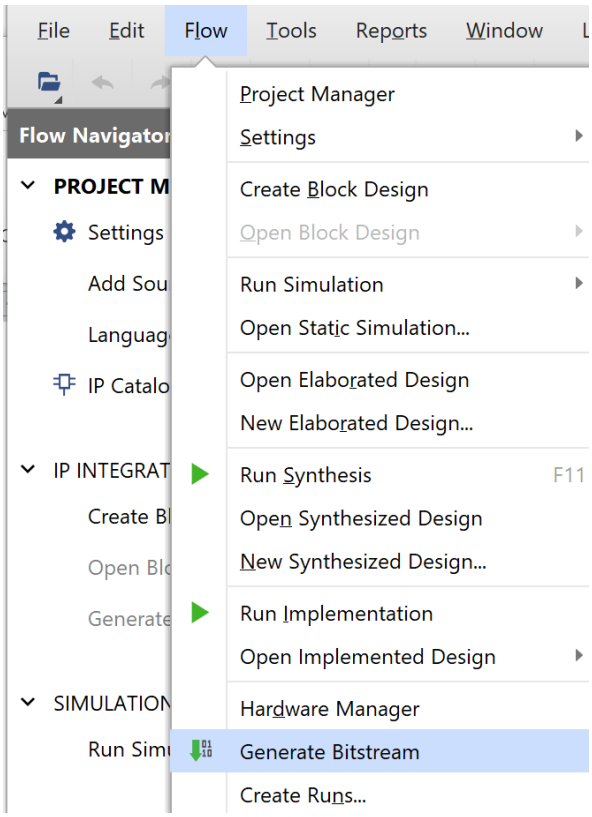

Figure 15. Generate Bitstream

After the bitstream has been generated a window will pop up as shown in Figure 16. Click on the ✕ button in the top-right corner to close the window.

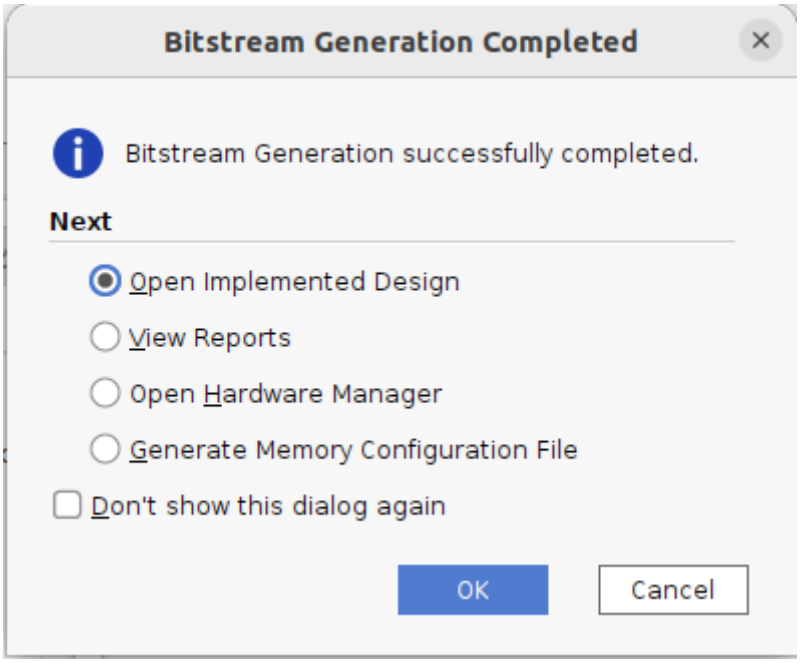

Figure 16. Bitstream Generation Completed

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

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

```
[RVfpgaBooleanPath]/Labs/Lab05/project_2/project_2.runs/impl_1
```

It is recommended that you use Catapult SDK to download RVfpgaEL2-Boolean onto the Boolean 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-Boolean system onto the FPGA on the Boolean board, you will use Catapult SDK to download and run/debug programs on RVfpgaEL2-Boolean.