

# **QG004 – v1.1**

## **GUIDE FOR INSTALLING VIVADO GETTING STARTED WITH VIVADO AND ARTY-Z7 BOARD**

### **Author**

Computer Engineering Lab

- Faculty of Computer Science and Engineering, HCMUT.

### **Revision History**

<b>Version</b>	<b>Description</b>	<b>Date</b>
1.0	Initial release.	11/01/2022
1.1	- Update: Set top module in III.4 - Add III.5. Simulation - Add color to code	18/01/2022

*This document is for internal use only.*

*In case of a mistake or proposal to edit, update, please contact the lecturer/teaching assistant of the subject or computer engineering laboratory, 201C5, University of Technology University - VNU-HCM . Thank you.*

## TABLE OF CONTENTS

<b>I. INTRODUCTION.....</b>	<b>2</b>
<b>II. INSTALLATION.....</b>	<b>2</b>
1. Preparation.....	2
2. Installing on Windows.....	4
3. Installing on Linux-based system.....	9
4. Installing Board Files for Arty-Z7.....	10
<b>III. GETTING STARTED WITH VIVADO AND ARTY-Z7.....</b>	<b>12</b>
1. Preparation.....	12
2. Create a Vivado project.....	12
3. Example Project.....	17
4. Generate bitstream and configure FPGA.....	20
5. Simulation.....	23

# I. INTRODUCTION

This document will show the process of installing and configuring the Xilinx Vivado software to develop logic circuit on Arty-Z7 board in the subject Logic Design with HDL.

This guide is based on Vivado 2018.2 version, but other versions are similar. Students can also install another version to study in the subject.

# II. INSTALLATION

There are some useful online references at:

1. <https://digilent.com/reference/programmable-logic/guides/installing-vivado-and-sdk>
2. <https://digilent.com/reference/vivado/getting-started-with-ipi/2018.2>

## 1. Preparation

Download the installation file on Xilinx's website.

- Go to Vivado download page at: <https://www.xilinx.com/support/download.html>

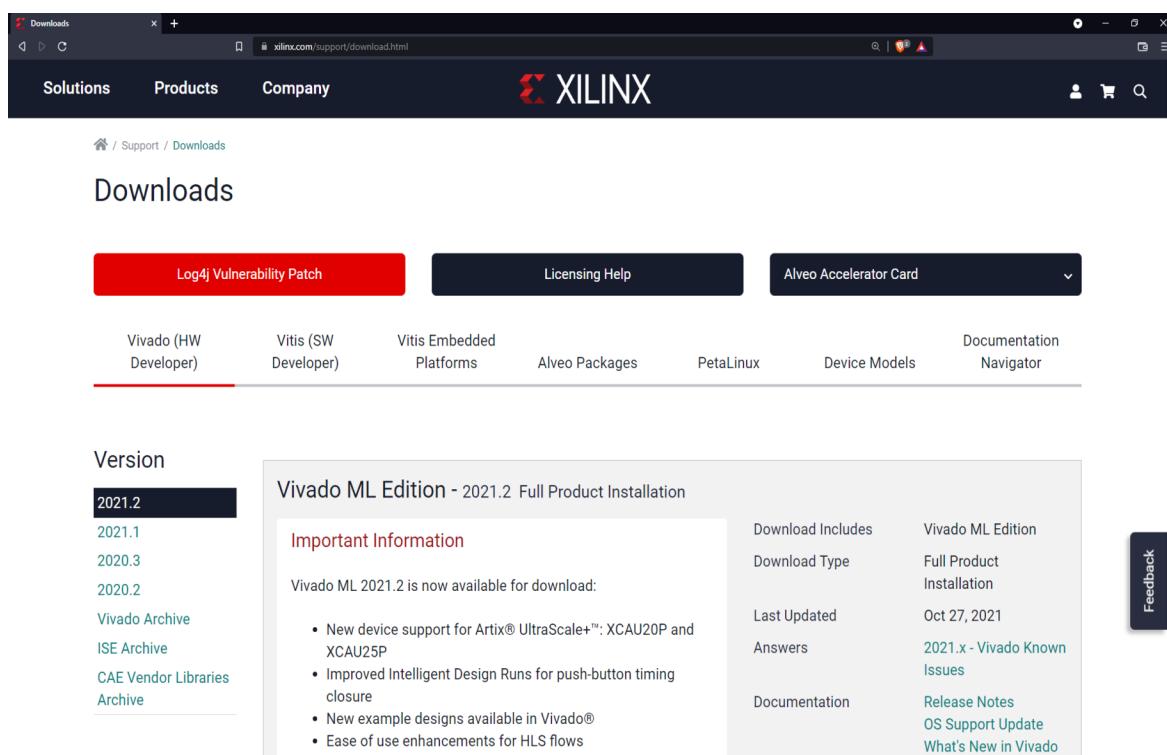


Figure 1: Xilinx Vivado download webpage

- We need an Xilinx account to download softwares. Click on the user icon at the top-right corner, select **Login | Register**.

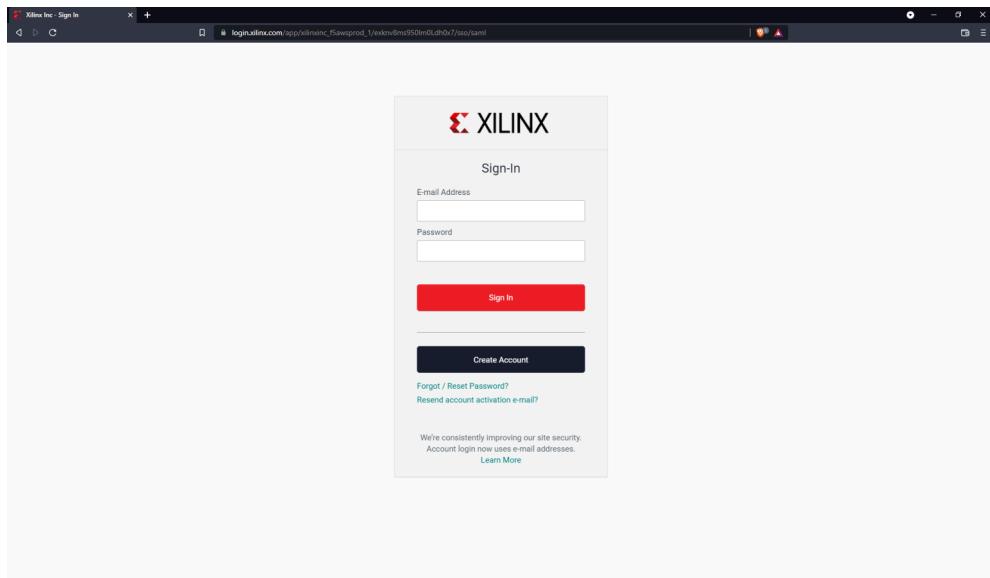


Figure 2: Xilinx Sign-In page

- Click **Create Account** and follow the register process to acquire an Xilinx account.
- Go back to the download page, select **Vivado Archive > 2018.2**.
- At Vivado Design Suite - HLx Editions - 2018.2 Full Product Installation, you can chose the Installer which is appropriate with your operating system (In this guide, we use the Web Installer). At this step, you are required to log in and fill some information to dowload.

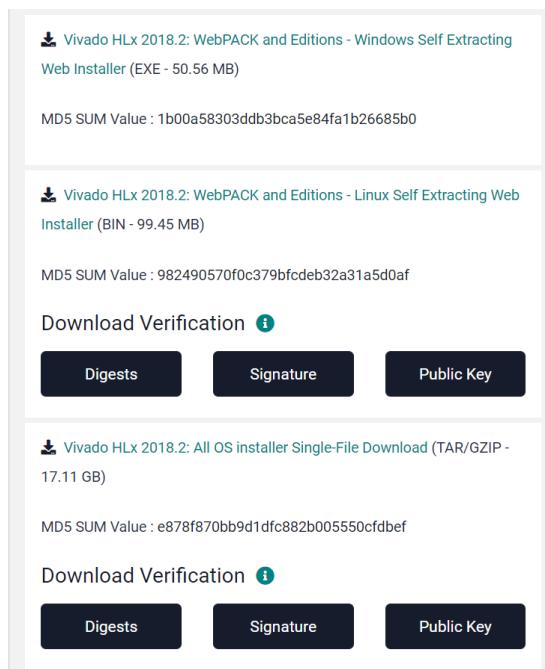
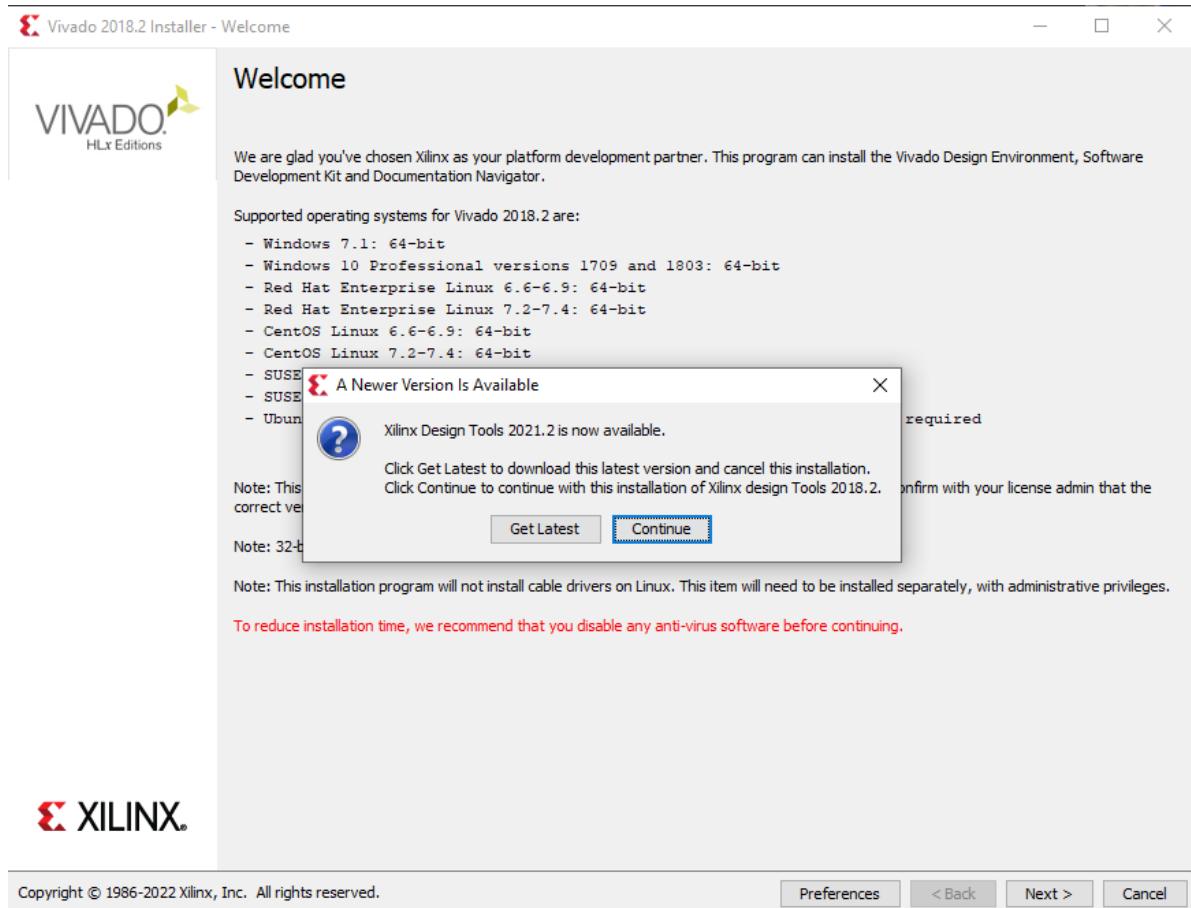


Figure 3: Vivado 2018.2 Installer options

## 2. Installing on Windows

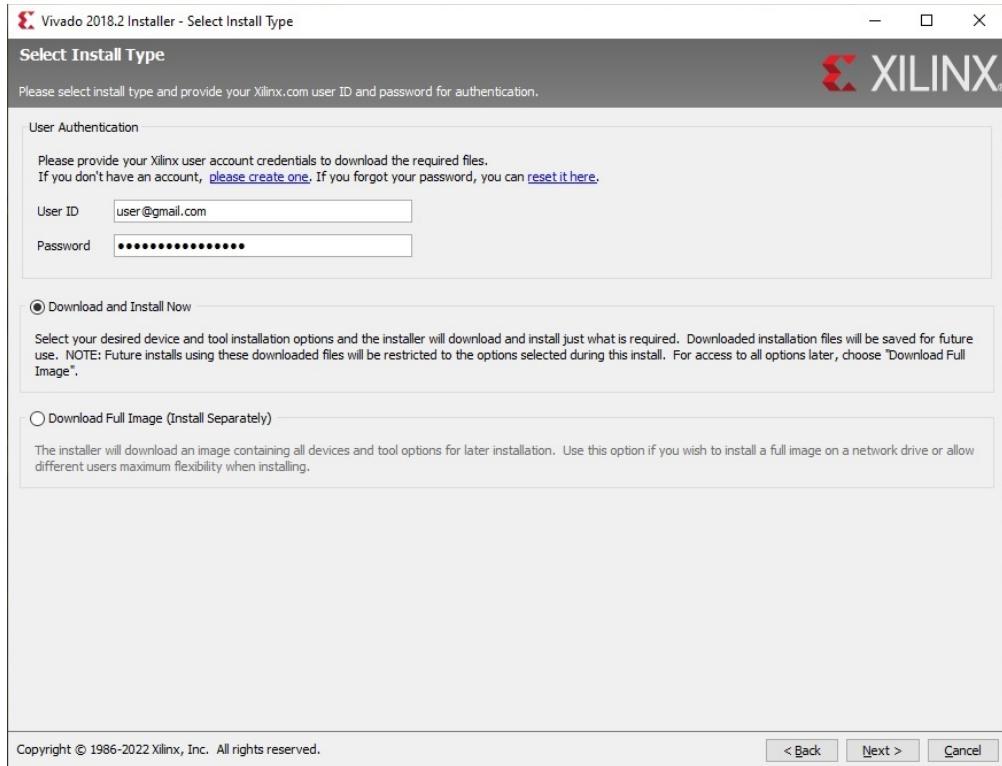
- Open the installer. You may be asked for getting a newer version, just click **Continue**.



**Figure 4:** Vivado installation window

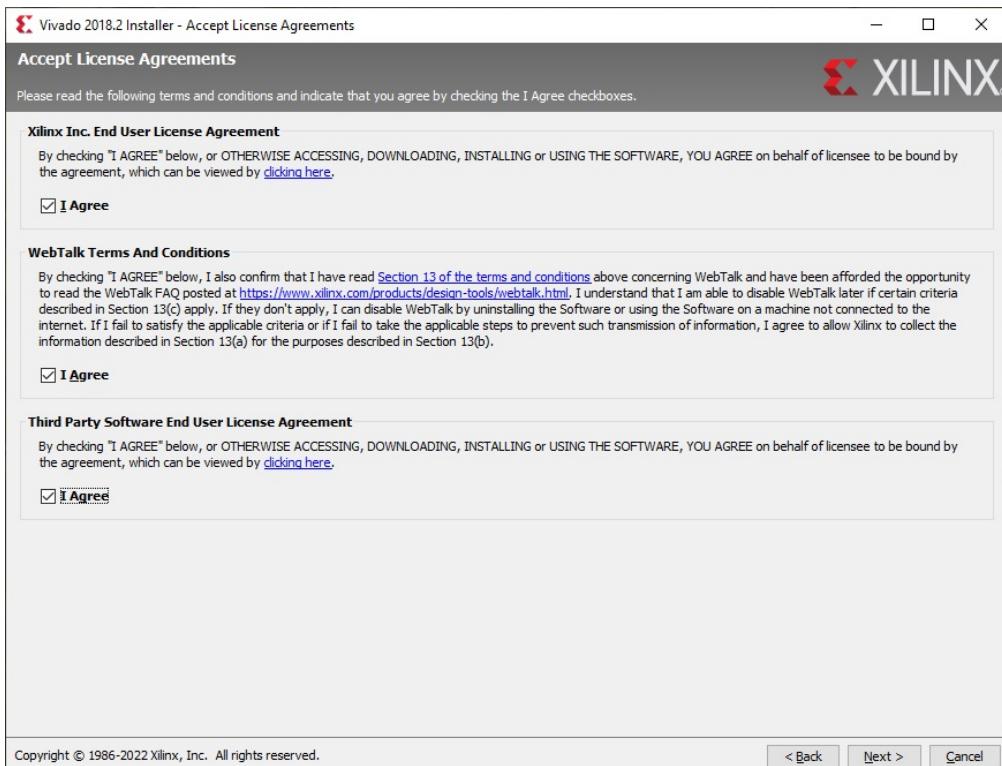
- You should satisfy the system requirement noticed by the installer. Click **Next**.

- Fill your Xilinx account information, select **Download and Install Now**, then click **Next**.



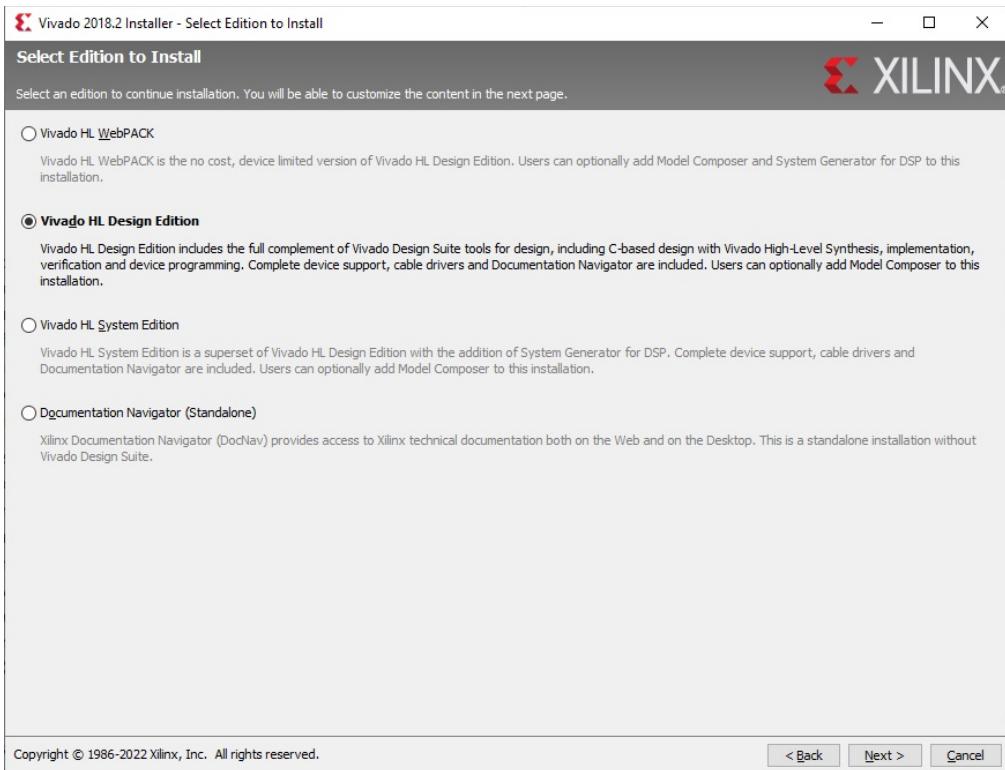
**Figure 5:** Vivado installation window - User Authentication

- Tick 'I Agree' of all statements, then click **Next**.



**Figure 6:** Vivado Installation window - License Agreements

- Select **Vivado HL Design Edition**, then click **Next**.



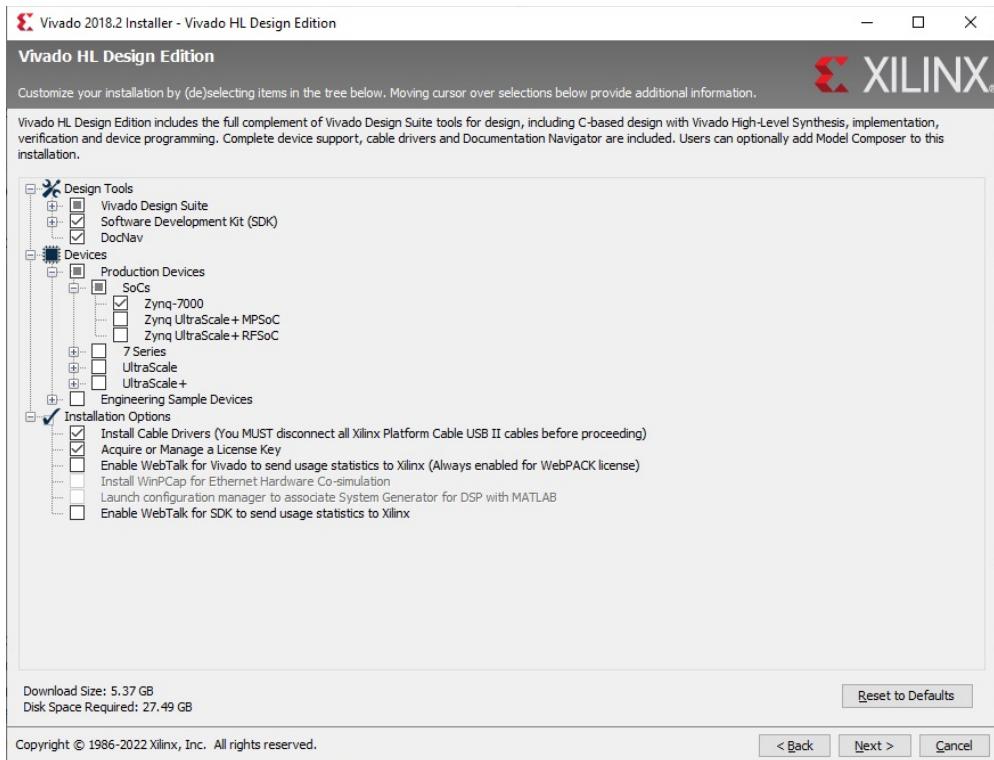
**Figure 7:** Vivado installation window - Select Edition to Install

- Customize the installation: you can deselect unneeded devices and options as **Figure 8** for smaller download.

Note: **MUST have SoCs Zynq-7000 devices** to develop on Arty-Z7 board.

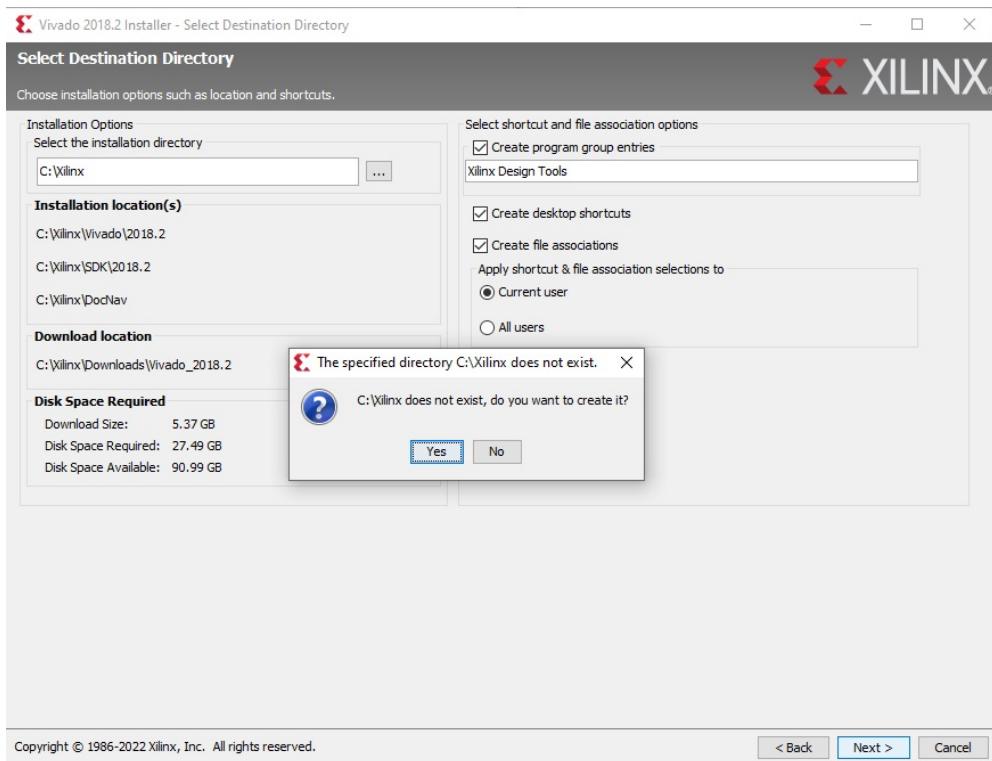
**MUST select Install Cable Drivers** to program FPGAs via USB cable.

Click **Next**.



**Figure 8:** Vivado installation window - Customize the installation

- Select the installation folder. The Installer may ask to create a directory, click **Yes**. Click **Next**.



**Figure 9:** Vivado installation window - Select Destination Directory

**Note: Installation folder MUST NOT have space and special characters.**

- Review Installation Information again and click **Next**.

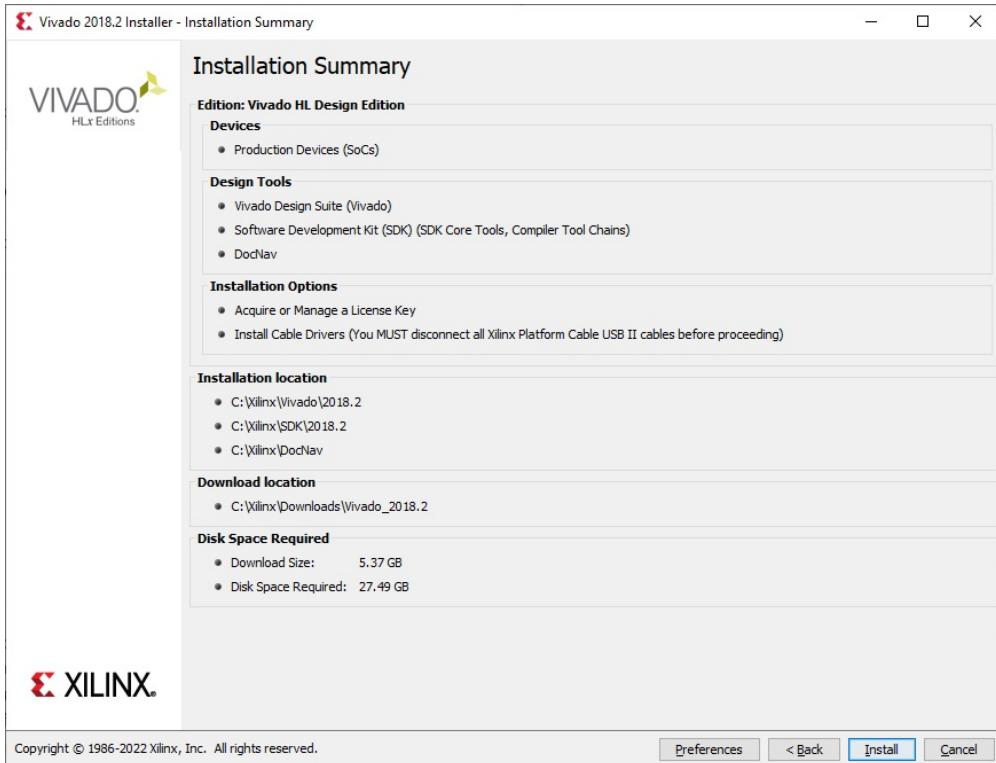


Figure 10: Vivado installation window - Installation Summary

- Wait for downloading and installing. Windows may ask for install device software at the end of installation to install cable driver, click **Install**.

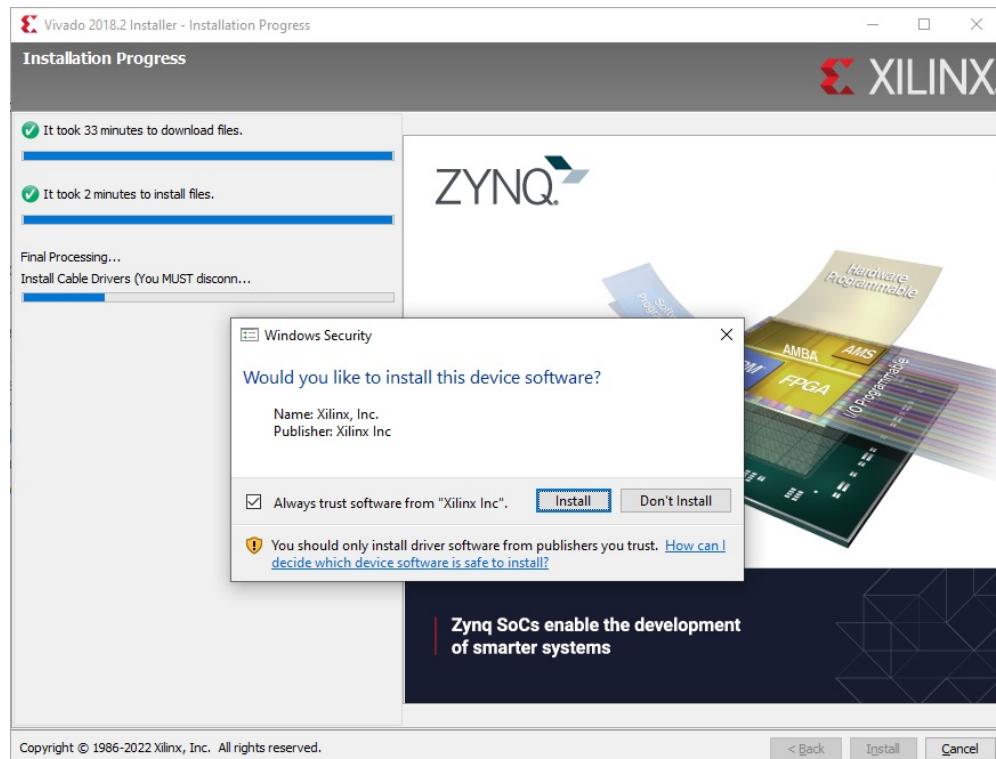


Figure 11: Vivado installation window - Installation Progress

- After installation completed, a notification window will appear, click **OK**. And a **Vivado License Manager** window will pop-up, just close it.

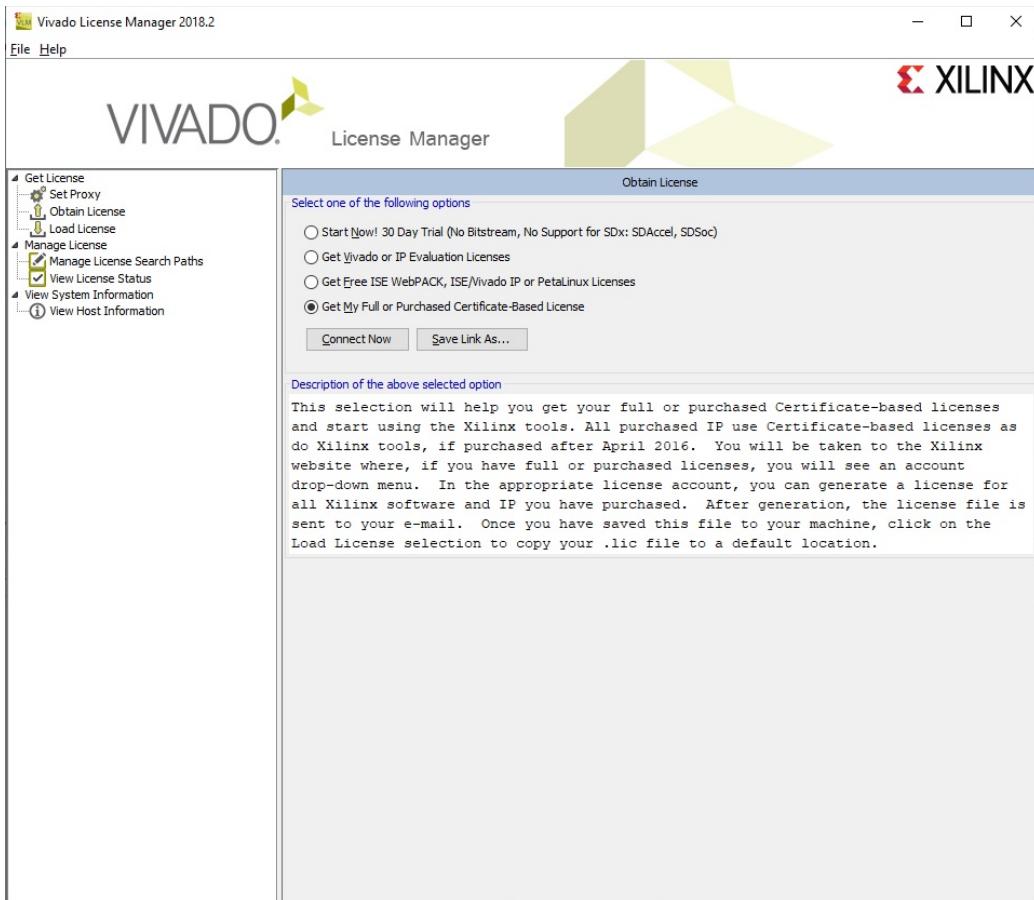


Figure 12: Vivado License Manager window

You have completed the installation process.

### 3. Installing on Linux-based system

- Open a terminal and navigate to the downloaded installer directory. Then, execute the command below (<installer> is the installer's file name):

```
chmod +x <installer>.bin && sudo ./<installer>.bin
```

The installation process is the same as on Windows. But we have to install the cable driver manually.

#### \* Installing Cable Drivers on Linux

- Open a terminal and navigate to the directory:

```
<installation_path>/data/xicom/cable_drivers/lin64/install_script/install_drivers/
```

For example:

```
/opt/Xilinx/Vivado/2018.2/data/xicom/cable_drivers/lin64/install_script/install_drivers/
```

- Then run the install\_drivers by command:

```
./install_drivers
```

- In order to use the USB drivers with a serial terminal, each user that will be using serial terminals must be added to the dialout group. Run command:

```
sudo adduser $USER dialout
```

## 4. Installing Board Files for Arty-Z7

To create a Vivado project on Arty-Z7 board, we need to add **Digilent Arty-Z7 board files** to Vivado.

- Go to the page: <https://github.com/Digilent/vivado-boards>

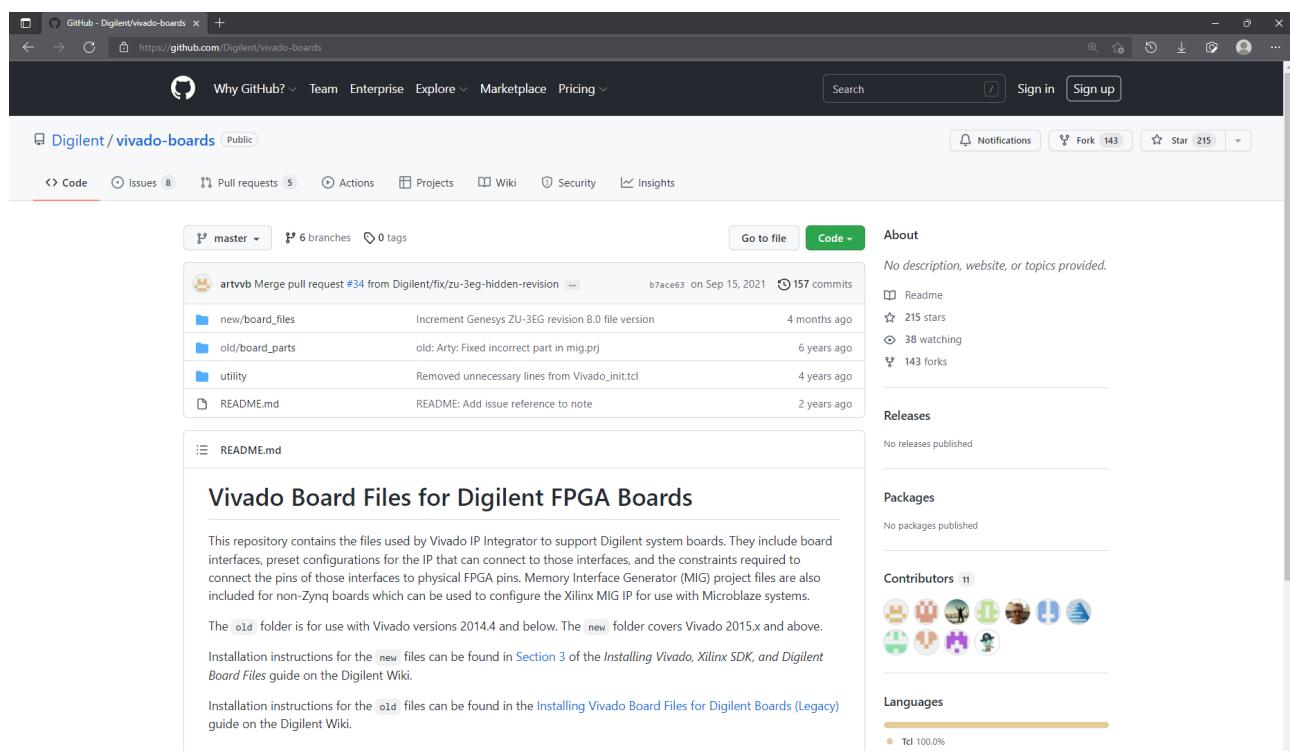


Figure 13: Digilent's vivado-board github repository

- If you familiar with using git, you can clone the repository or you just dowload the .zip file that compressed the sources by click on **Code > Download ZIP**.

- Extract the zip file, we have many of board files and the Arty-Z7 one is in the folder **vivado-board/new/board\_files/arty-z7-20**

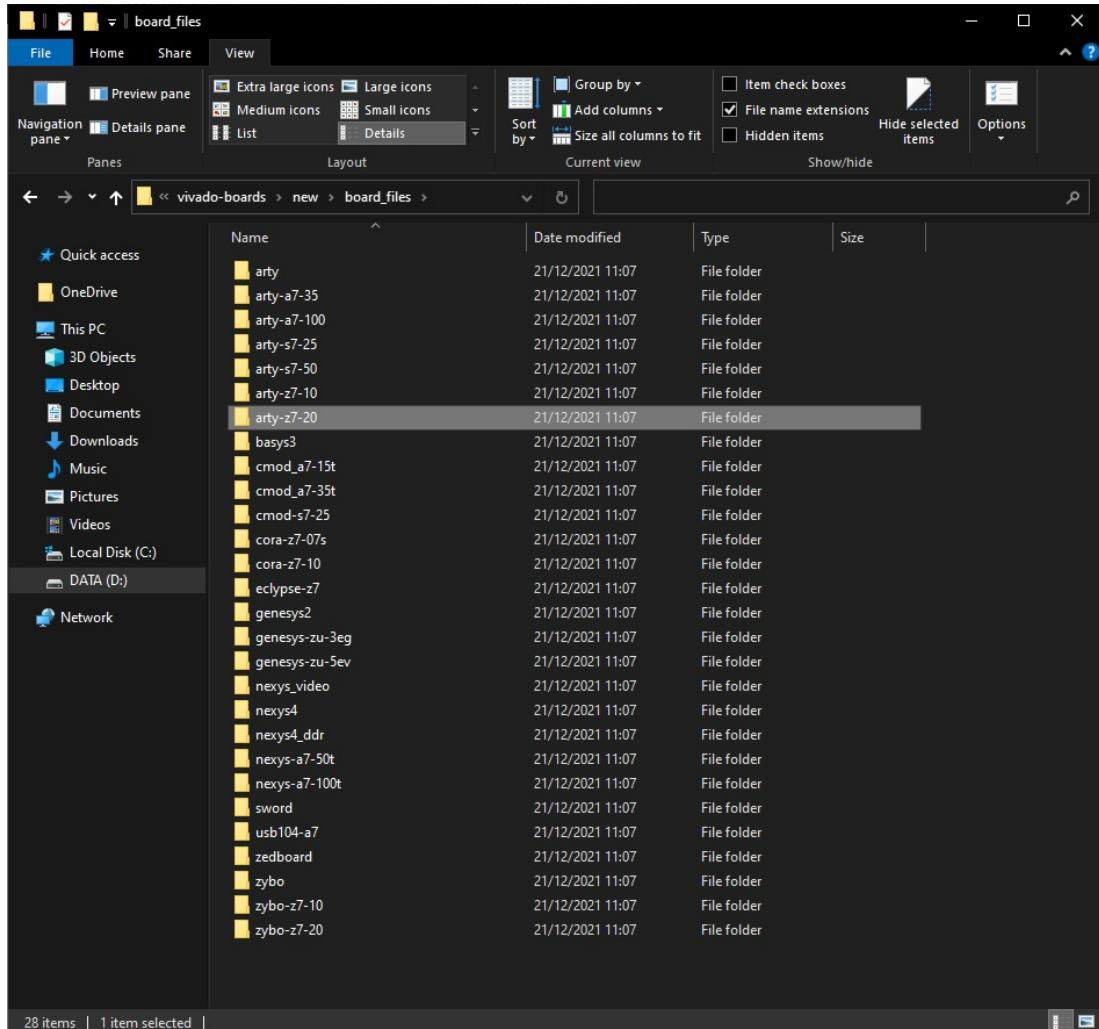


Figure 14: Arty-Z7 board files folder

- Copy the folder '**arty-z7-20**' to Vivado installation folder at

<installation\_path>/data/boards/board\_files

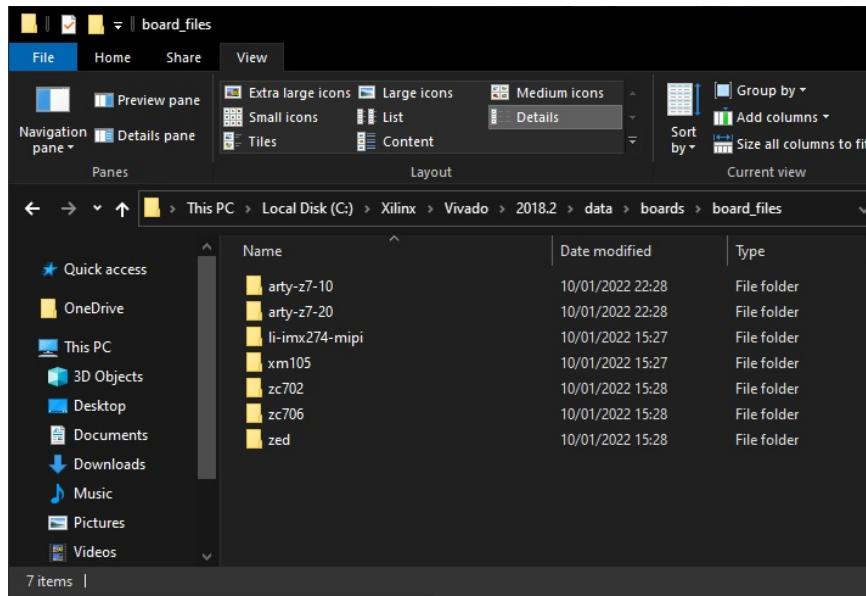


Figure 15: Vivado's Board files folder

You also can copy other board files folders to add them to Vivado.

Note: When installing Vivado versions 2014.4 or older, use the files found in 'old/board\_files' instead.

## III. GETTING STARTED WITH VIVADO AND ARTY-Z7

### 1. Preparation

- Vivado Design (with cable driver) installed.
- Arty-Z7 board files added.
- Get Arty-Z7 constraint file at: <https://github.com/Digilent/digilent-xdc>  
or just save the file **Arty-Z7-20-Master.xdc** in the repository.

### 2. Create a Vivado project

NOTE: **ALL PATHS** that are used in Vivado **MUST NOT** have **SPACE**, and special characters.

- Open Vivado.

Use command line on Linux:

```
source /opt/Xilinx/Vivado/2018.2/settings64.sh
vivado &
```

Note: This will create many files on the standing directory, you should create a directory for Vivado projects and run above commands on that directory.

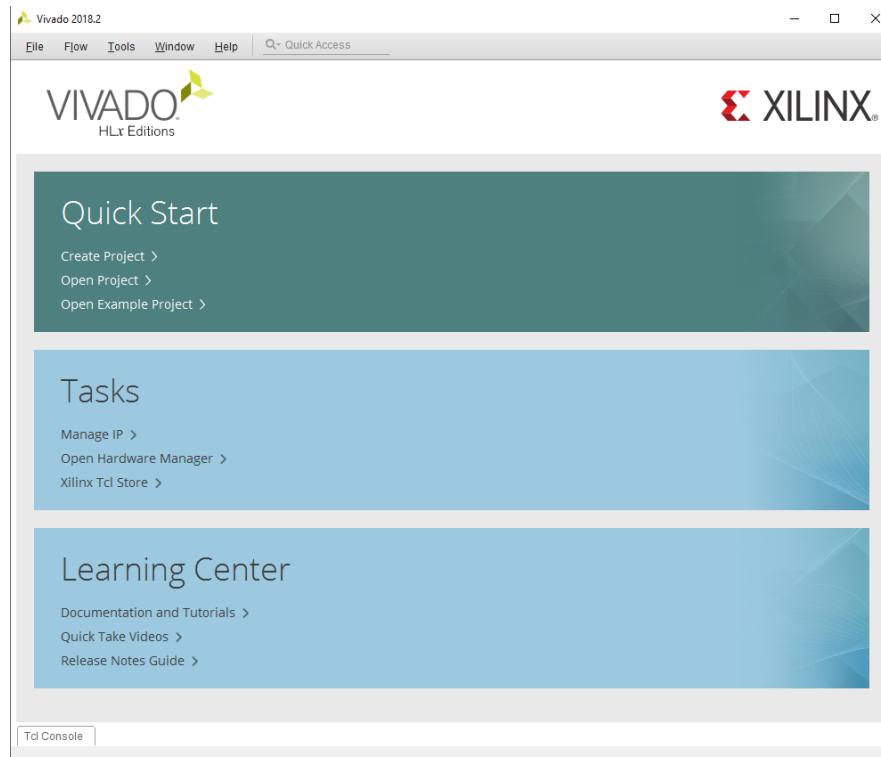


Figure 16: Vivado window

- Click **Create Project**. A New Project wizard window will appear, click **Next**.

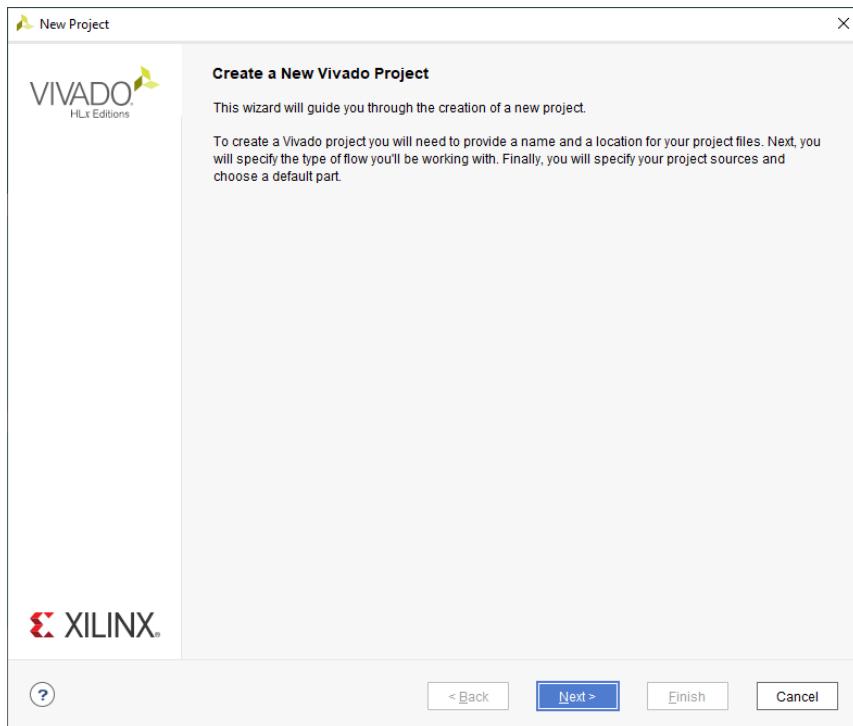


Figure 17: New Project window

- Type the project name and browse to the project location. Click **Next**.

Note: Location should be the common Vivado projects folder and you should tick on the option **Create project subdirectory**.

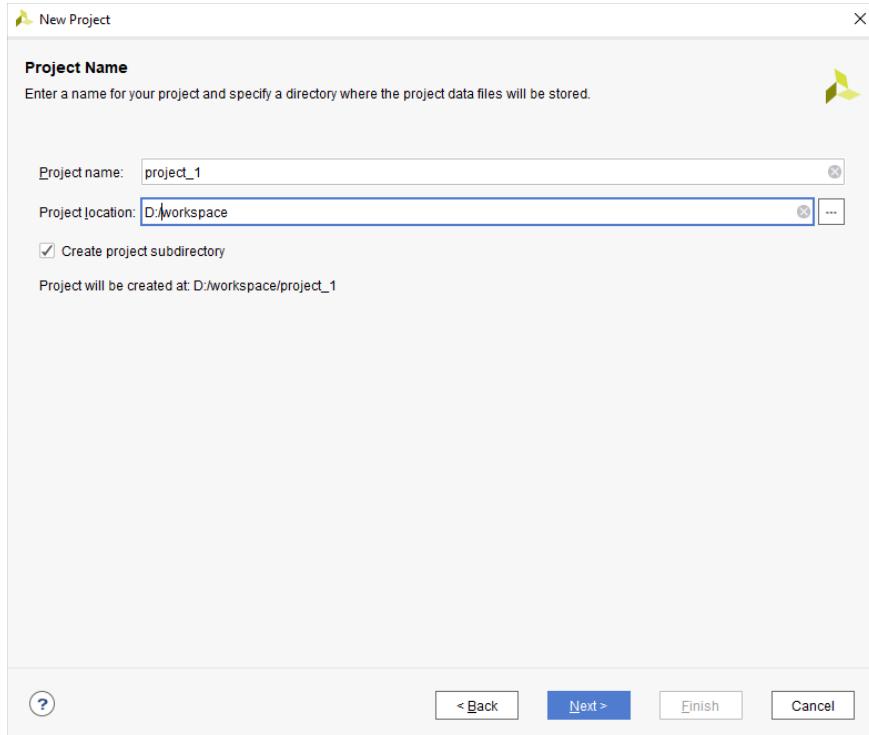


Figure 18: New Project window - Project Name

- Select **RTL Project**, then click **Next**.

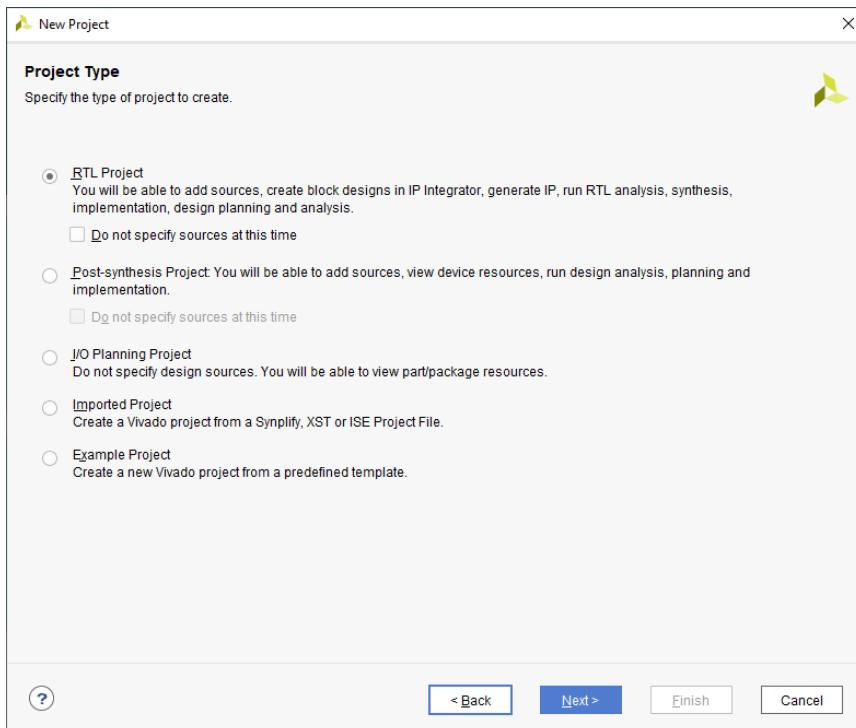
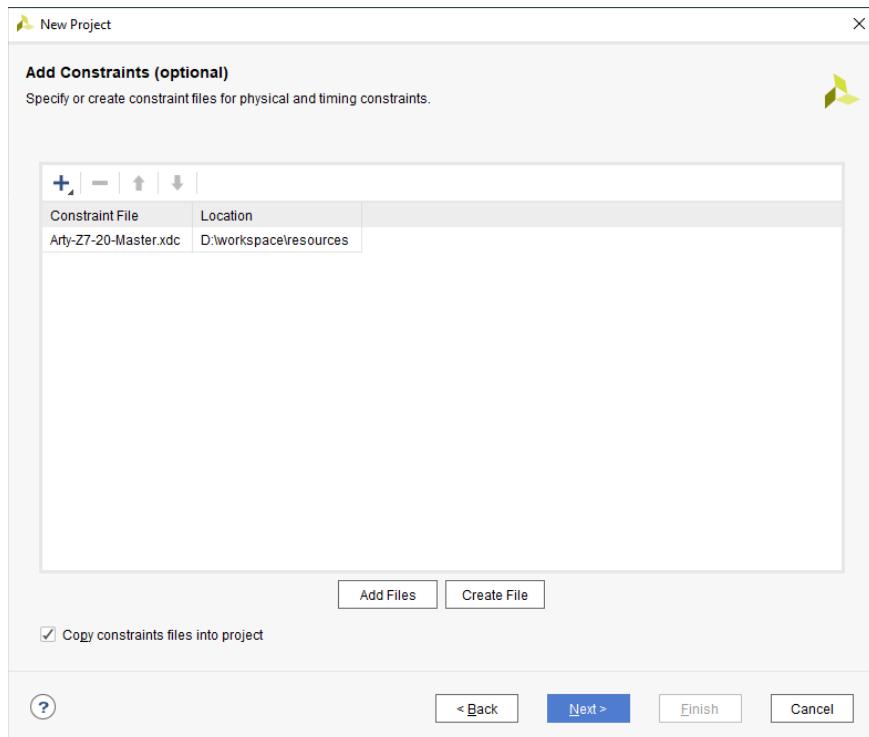


Figure 19: New Project window - Project Type

---

- **Add Source:** You can add created source code (if any) or just skip this step by click **Next**.
- **Add Constraint:** You can add the downloaded constraint file at this step or just skip it by click **Next**.



**Figure 20:** New Project window - Add Constraints

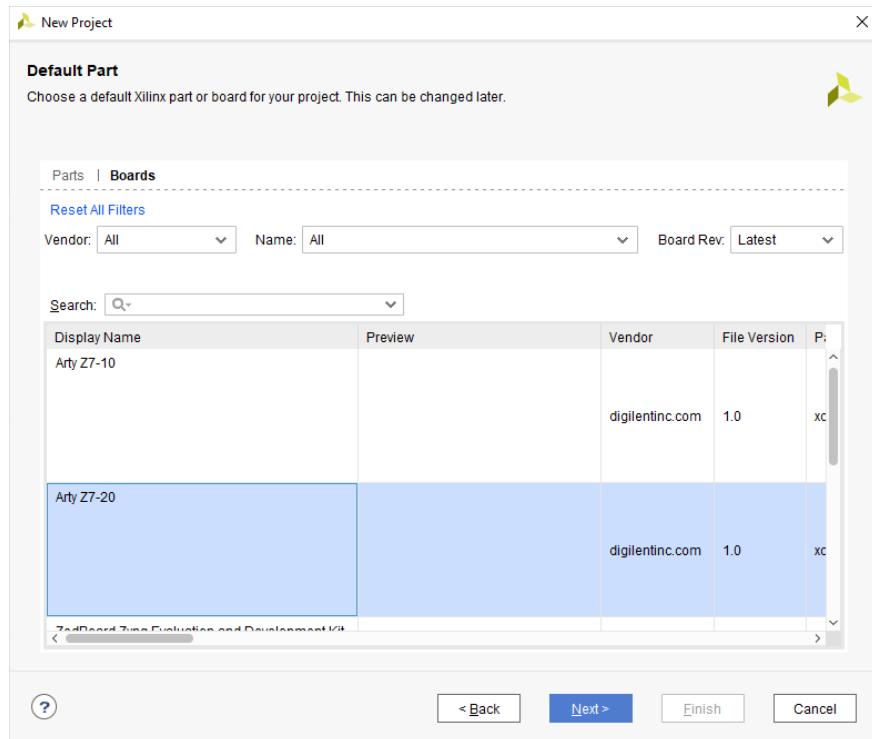
Note: You should tick on **Copy constraint files into project** to modified it without losing the original contents.

- **Default part:** select device part

- + Click on **Boards** tab

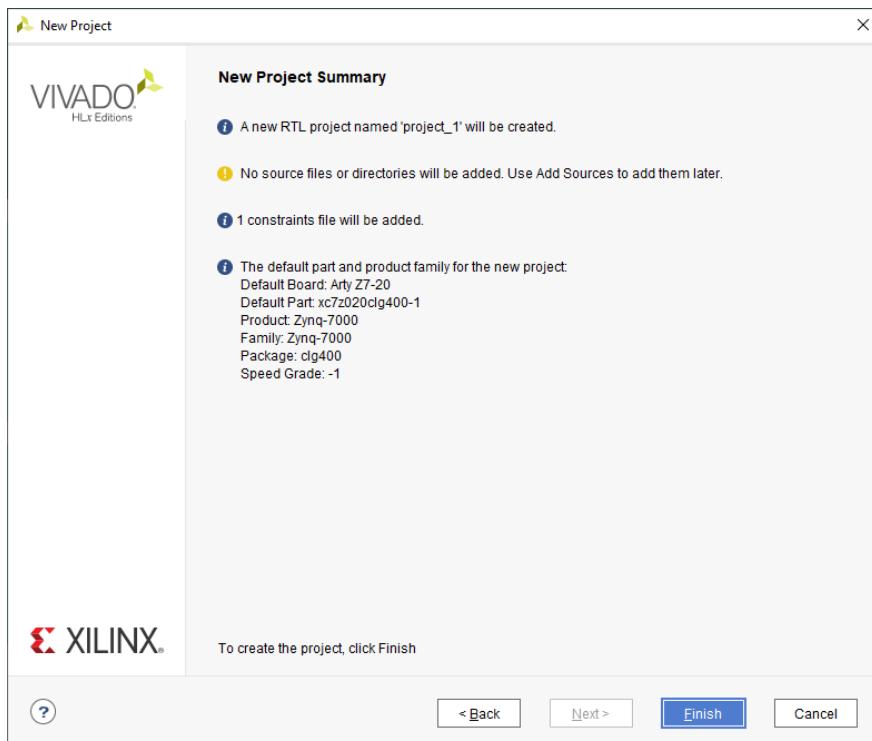
- + Select **Arty Z7-20**

- + Click **Next**



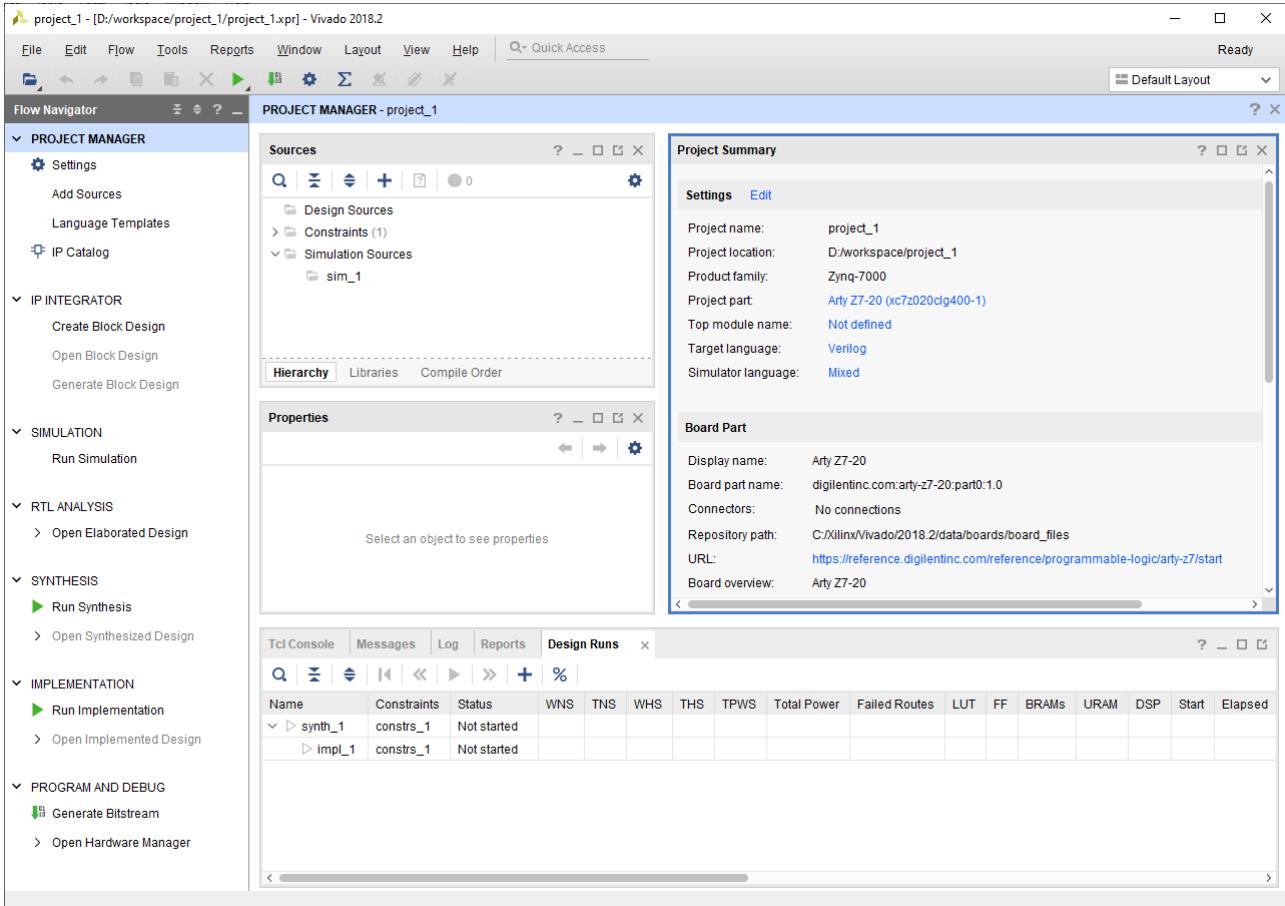
**Figure 21:** New Project window - Default Part

- Review Project information again then click **Finish**.



**Figure 22:** New Project window - New Project Summary

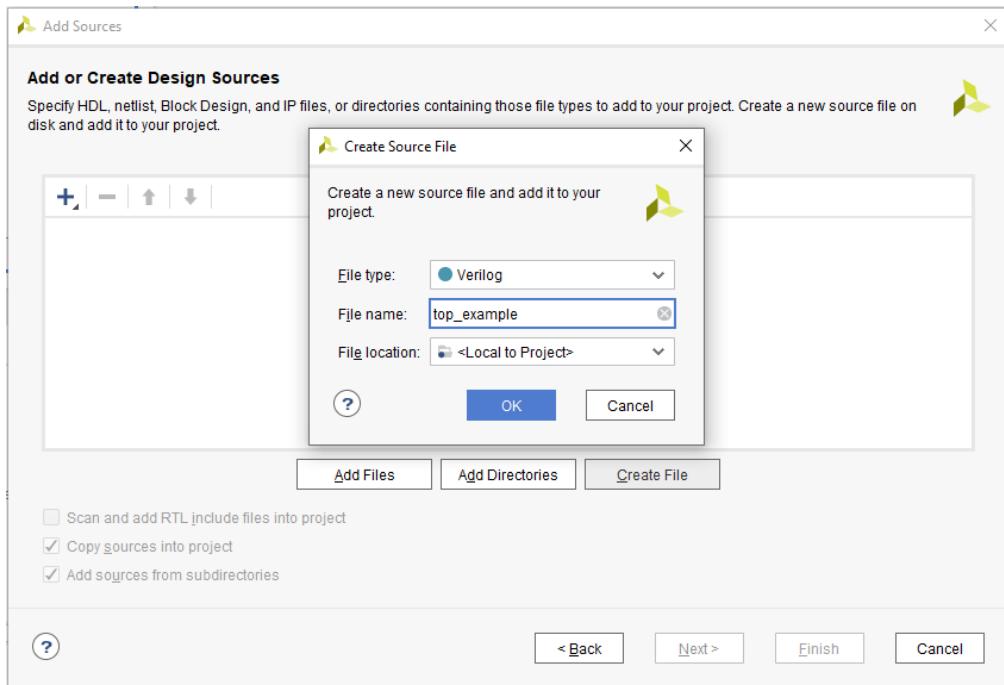
### 3. Example Project



**Figure 23:** Vivado working window

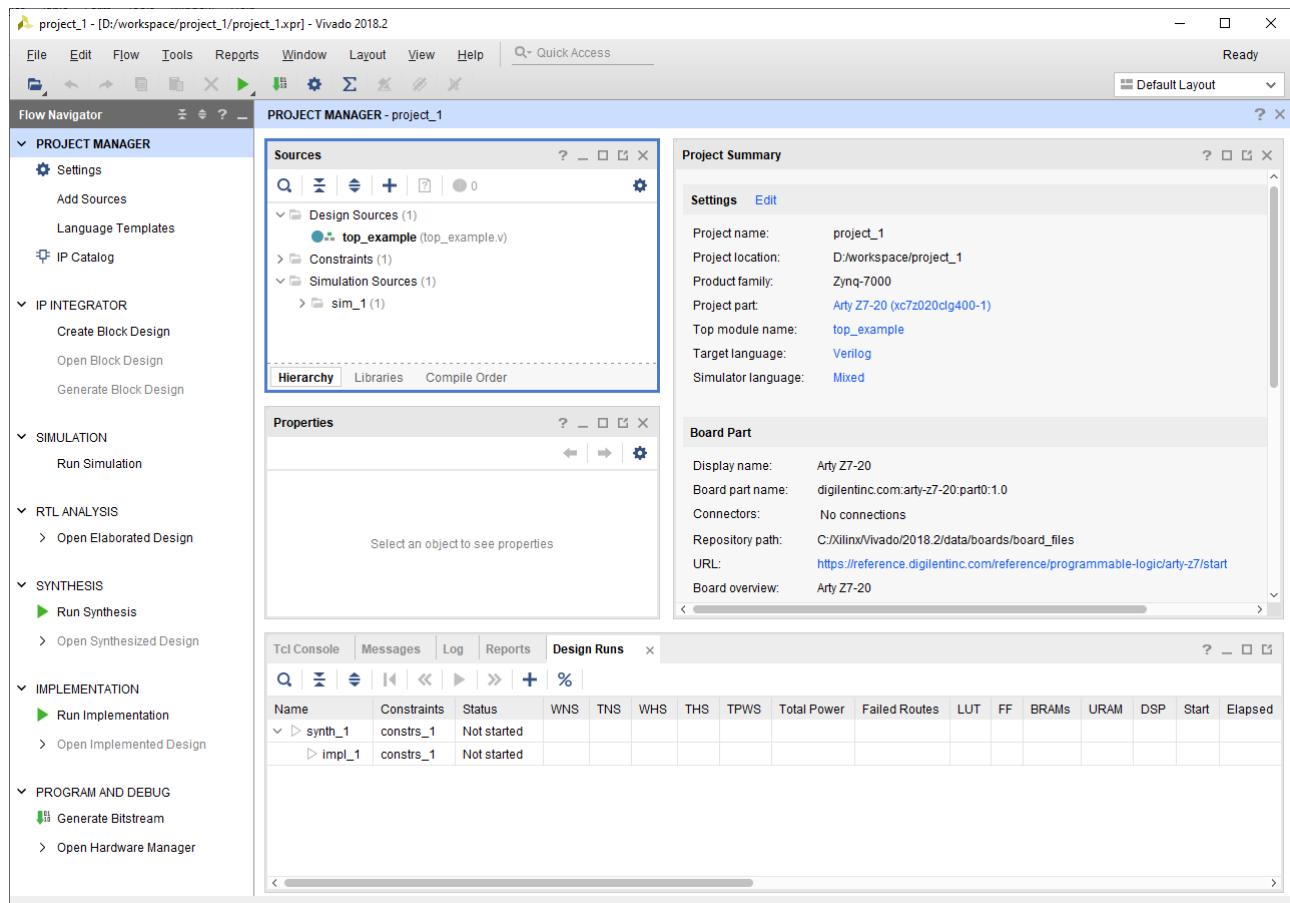
- Create a source code file:

- + At **Project Manager**, **Sources** board, click on the **+** symbol or select **File > Add Sources...**
- + Select **Add or create design sources**.
- + Click **Next**.
- + Click **Create File**.
- + Enter file name, then click **OK**.
- + Click **Finish**.
- + Some pop-up window will appear, click **OK, Yes**.



**Figure 24:** Create a design source file

- The created file is shown in **Sources Hierarchy > Design Sources**. Double click to open it.



**Figure 25:** Project with the created design source file

- Edit the source as below, then save by click on the **Save** icon on editor window or **Ctrl+S**:

```
module top_example(
    input [3:0] in,
    output [3:0] out
);

    assign out = in;

endmodule
```

- Edit constraint file to assign pins:

+ Open the added constraint file in **Source Hierarchy > Constraints/constrs\_1/Arty-Z7-20-Master.xdc**

+ Uncomment below lines:

```
19 #set_property -dict { PACKAGE_PIN L14     IOSTANDARD LVCMOS33 } [get_ports { led5_g }]; #IO_L22P_T3_AD7P_35 Sch=LED5_G
20 #set_property -dict { PACKAGE_PIN M15     IOSTANDARD LVCMOS33 } [get_ports { led5_r }]; #IO_L23N_T3_35 Sch=LED5_R
21
22 ## LEDs
23 set_property -dict { PACKAGE_PIN R14     IOSTANDARD LVCMOS33 } [get_ports { led[0] }]; #IO_L6N_T0_VREF_34 Sch=LED0
24 set_property -dict { PACKAGE_PIN P14     IOSTANDARD LVCMOS33 } [get_ports { led[1] }]; #IO_L6P_T0_34 Sch=LED1
25 set_property -dict { PACKAGE_PIN N16     IOSTANDARD LVCMOS33 } [get_ports { led[2] }]; #IO_L21N_T3_DQS_AD14N_35 Sch=LED2
26 set_property -dict { PACKAGE_PIN M14     IOSTANDARD LVCMOS33 } [get_ports { led[3] }]; #IO_L23P_T3_35 Sch=LED3
27
28 ## Buttons
29 set_property -dict { PACKAGE_PIN D19     IOSTANDARD LVCMOS33 } [get_ports { btn[0] }]; #IO_L4P_T0_35 Sch=BTN0
30 set_property -dict { PACKAGE_PIN D20     IOSTANDARD LVCMOS33 } [get_ports { btn[1] }]; #IO_L4N_T0_35 Sch=BTN1
31 set_property -dict { PACKAGE_PIN L20     IOSTANDARD LVCMOS33 } [get_ports { btn[2] }]; #IO_L9N_T1_DQS_AD3N_35 Sch=BTN2
32 set_property -dict { PACKAGE_PIN L19     IOSTANDARD LVCMOS33 } [get_ports { btn[3] }]; #IO_L9P_T1_DQS_AD3P_35 Sch=BTN3
33
34 ## Pmod Header JA
35 #set_property -dict { PACKAGE_PIN Y18     IOSTANDARD LVCMOS33 } [get_ports { ja_p[1] }]; #IO_L17P_T2_34 Sch=JA1_P (Pin 1)
36 #set_property -dict { PACKAGE_PIN Y19     IOSTANDARD LVCMOS33 } [get_ports { ja_n[1] }]; #IO_L17N_T2_34 Sch=JA1_N (Pin 2)
```

**Figure 26:** Arty-Z7 constraint file with LEDs and Buttons assignment lines uncommented

+ Modify the port assignments to be the same as our module input/output port's names. Or you can name the ports of the module follow the port name in the constraint file. Remember to save the file.

```
## LEDs
set_property -dict { PACKAGE_PIN R14     IOSTANDARD LVCMOS33 } [get_ports { out[0] }]; #IO_L6N_T0_VREF_34 Sch=LED0
set_property -dict { PACKAGE_PIN P14     IOSTANDARD LVCMOS33 } [get_ports { out[1] }]; #IO_L6P_T0_34 Sch=LED1
set_property -dict { PACKAGE_PIN N16     IOSTANDARD LVCMOS33 } [get_ports { out[2] }]; #IO_L21N_T3_DQS_AD14N_35 Sch=LED2
set_property -dict { PACKAGE_PIN M14     IOSTANDARD LVCMOS33 } [get_ports { out[3] }]; #IO_L23P_T3_35 Sch=LED3

## Buttons
set_property -dict { PACKAGE_PIN D19     IOSTANDARD LVCMOS33 } [get_ports { in[0] }]; #IO_L4P_T0_35 Sch=BTN0
set_property -dict { PACKAGE_PIN D20     IOSTANDARD LVCMOS33 } [get_ports { in[1] }]; #IO_L4N_T0_35 Sch=BTN1
set_property -dict { PACKAGE_PIN L20     IOSTANDARD LVCMOS33 } [get_ports { in[2] }]; #IO_L9N_T1_DQS_AD3N_35 Sch=BTN2
set_property -dict { PACKAGE_PIN L19     IOSTANDARD LVCMOS33 } [get_ports { in[3] }]; #IO_L9P_T1_DQS_AD3P_35 Sch=BTN3
```

## 4. Generate bitstream and configure FPGA

- Set top module to synthesis: Right click on the file to be the top module, select **Set as Top**. (By default, the file created fristly is set as top).
- Click on **Generate Bitstream** in **Flow Navigator > PROGRAM AND DEBUGS**.
- A notification window about No Implementation Result Available pop-ups, click **Yes**.
- Select the **Number of jobs** to the maximum of your computer's available cores/threads to save running time. Click **OK**.

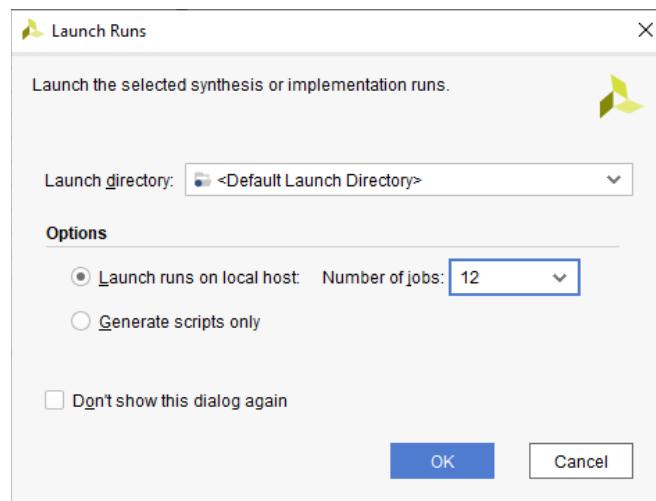


Figure 27: Launch Runs

- Wait for the Design Runs complete. A window will show when the process completed, just close it or select **Open Hardware Manager > OK** to configure the FPGA.

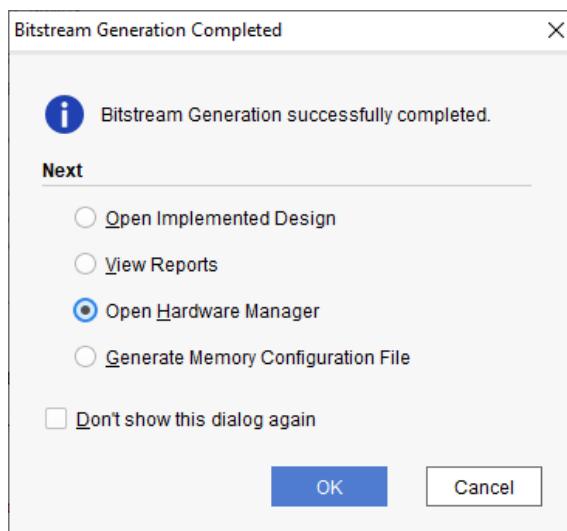


Figure 28: Bitstream Generation Completed window

The Open Hardware Manager button is also under the Generate Bitstream button.

- Connect the Arty-Z7 board to your computer via a USB cable.

Note: Make sure that all jumpers are the same as Figure 29.

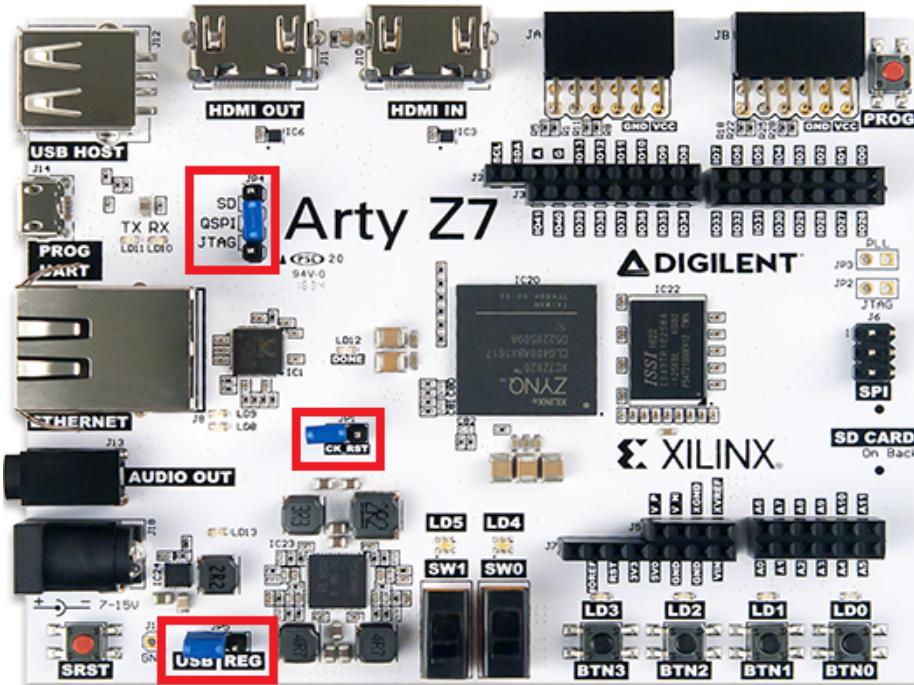


Figure 29: Arty-Z7 board

- Click on Open target > Auto Connect

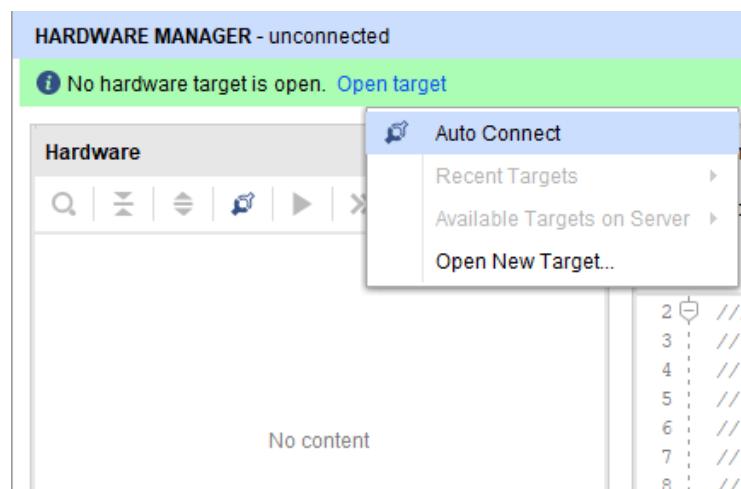


Figure 30: Connect Vivado to the board

- Click on Program device

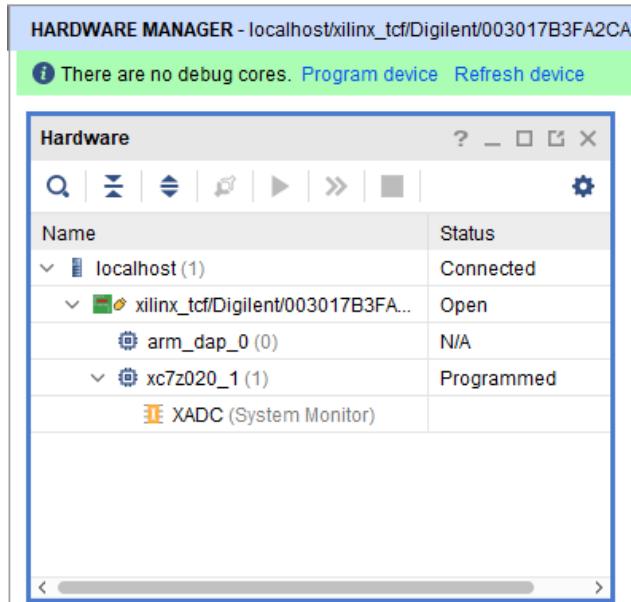


Figure 31: Vivado connected with the board

- Browse to the generated bitstream file (if needed), click **Program**.

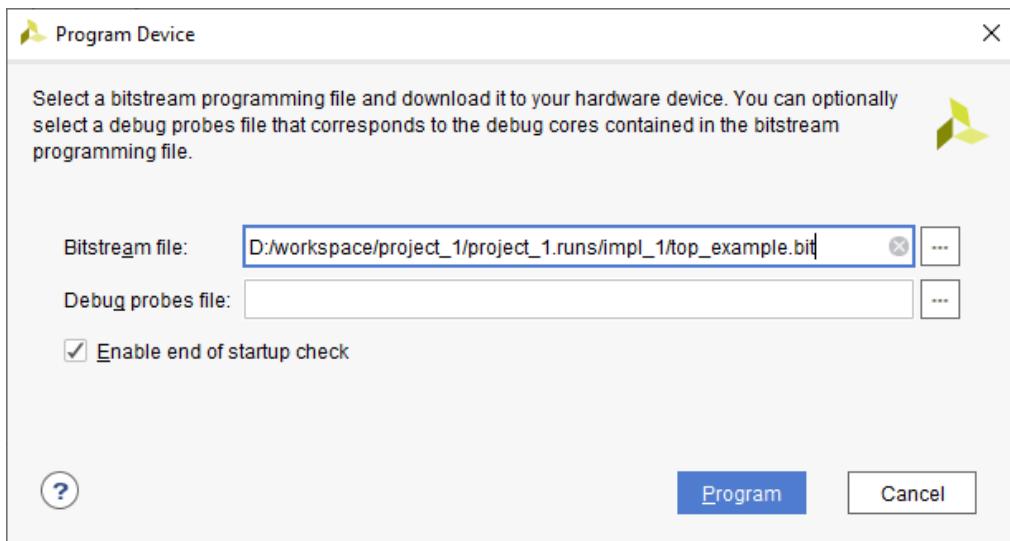


Figure 32: Program Device

- The process is done. Check the buttons and leds on the board then investigate the behavior.
- A summary report is generated in **PROJECT MANAGER > Project Summary**. Read the report to see how many resources on the FPGA are used. Also check the timing and power summary.

## 5. Simulation

- Create a source code file:

+ At Project Manager, Sources board, click on the + symbol or select **File > Add Sources...**

+ Select **Add or create simulation sources.**

+ Click **Next.**

+ Click **Create File.**

+ Enter file name, then click **OK.**

+ Click **Finish.**

+ Some pop-up window will appear, click **OK, Yes.**

- Edit the file as follow:

```
'timescale 1ns/10ps // time-unit = 1 ns, precision = 10 ps

module top_example_tb;

reg [3:0] in;
wire [3:0] out;

// Unit under test (UUT)/Design under test (DUT)
top_example uut (
    .in     (in),
    .out    (out)
);

// Command to print simulation signal values
initial begin
    $monitor("At time %t : input = %b - output = %b", $time, in, out);
end

// Stimulus generation: Sequent input of the simulation
initial begin
    in <= 0;
    #10 // Wait for 10 units of time
    in <= 4'b0001;
    #2  in <= 4'b0010;
    #3  in <= 4'b0011;
    #5  in <= 4'b0101;
    #2  in <= 4'b1001;
    #2  in <= 4'b1111;
    #5 $stop; // end of simulation

```

`end`

`endmodule`

- Set top simulation: Right click on the created testbench file, select **Set as Top**.
- In **SIMULATION**, click **Run Simulation > Run Behavioral Simulation**.

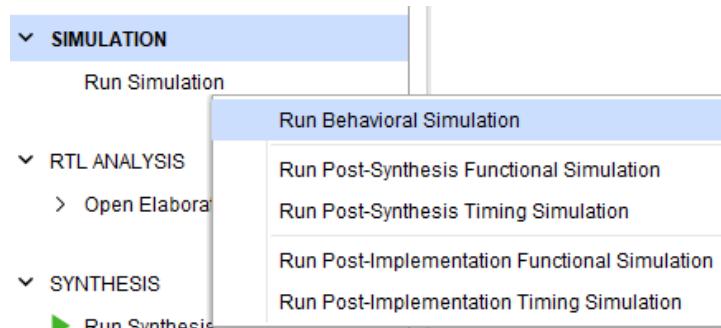


Figure 33: Run Simulation

- The waveform is in tab *Untitled 1*, click to see it. Use Zoom tools to scale the waveform. Try to expand the signals, change a signal display radix by right click on it and change the Radix. Let's discover other functions in simulation window yourself.

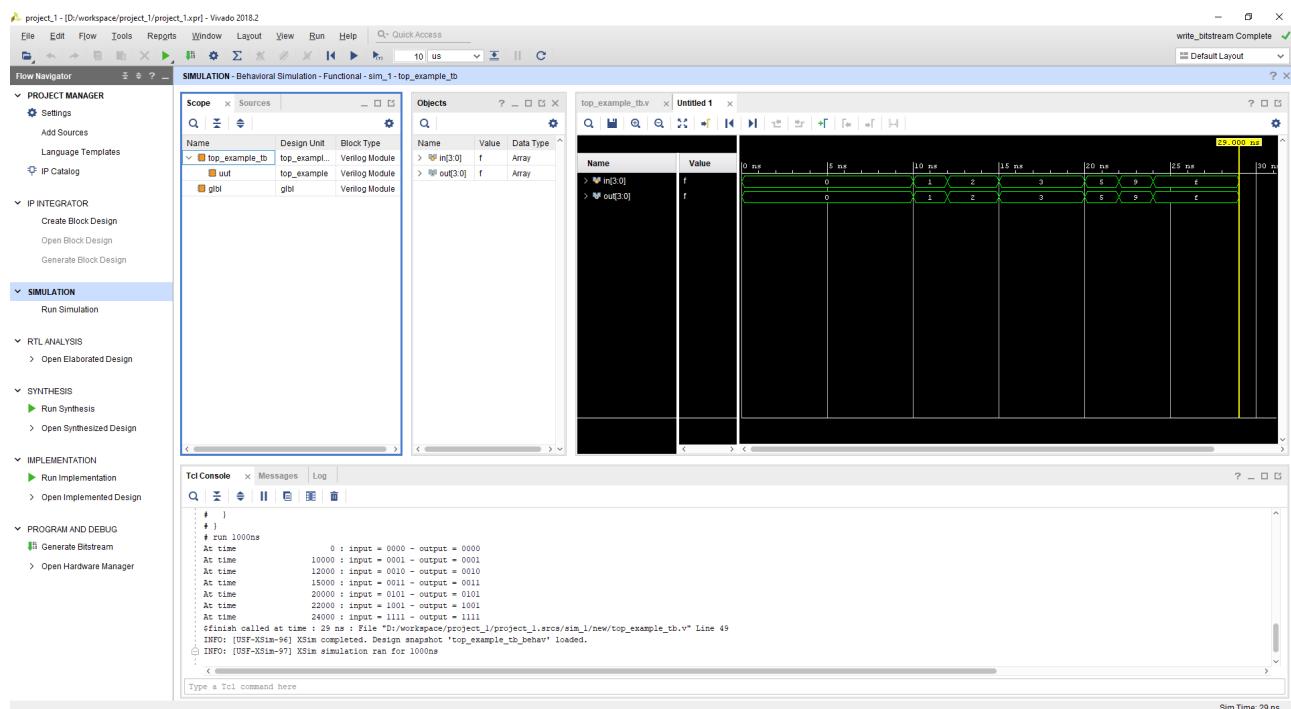


Figure 34: Vivado window - Simulation

- Check the **Tcl console** to see `$monitor` command's output

End of document.