Xilinx Vivado webpack download and installation

This guide explains how to download and install the Vivado Design Suite tools. You will use this suite throughout the course. There are multiple versions of this tool. Here we will focus on installation of Webpack version which is free and available to all students.

---

1 Install The Vivado Design Suite

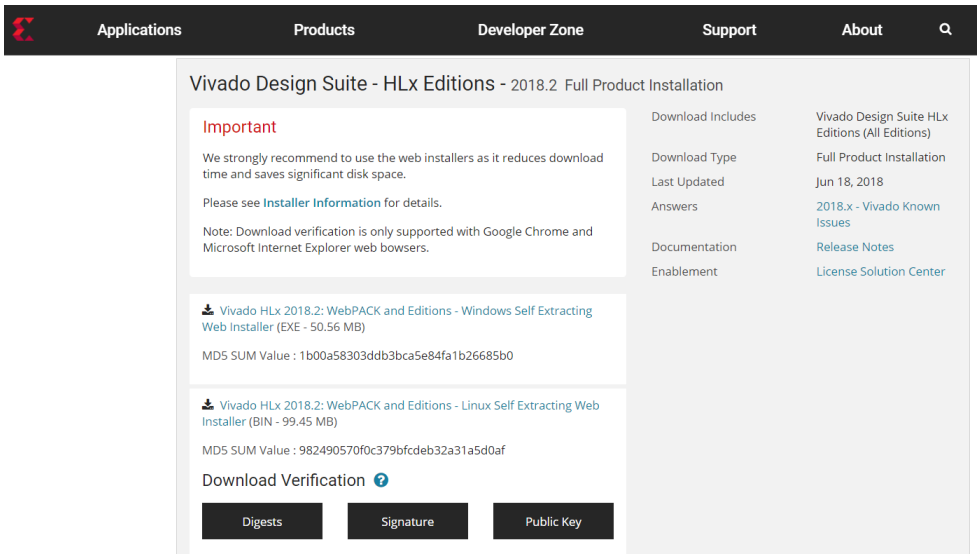
Vivado is a software produced by Xilinx for synthesis and analysis of HDL designs. The WebPack version of Vivado is free and You can download and install it from here:
https://www.xilinx.com/support/download.html

1.1 Download the Vivado Design Suite

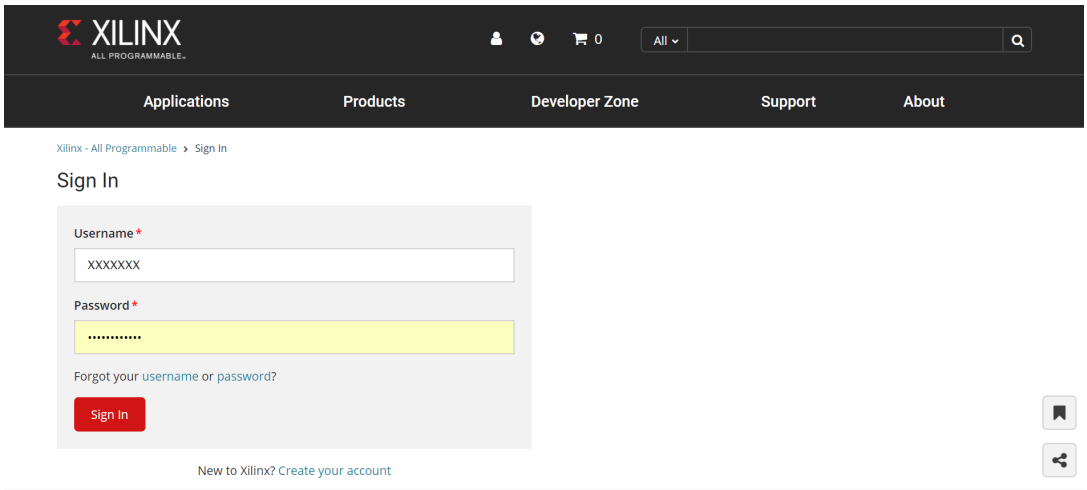
Select the appropriate version for your laptop operating system. Choose "Vivado HLx 2018.2: WebPACK and Editions - Windows Self Extracting Web Installer" to download the Vivado SDK installer version 2018.2 for 64-bit Windows machine.

There is no official Vivado for the Mac. To use this software, you must do one of the following:

- Run Windows or Linux in a Virtual Machine
  - Download "Vivado HLx 2018.2: WebPACK and Editions - Linux Self Extracting Web Installer" from Xilinx website
- Use Vivado on EECS server



You need a Xilinx account to download the tool. Register with your UCI net ID and sign in to download the Xilinx toolchain. This would be a file less than 100MB which will both download and install the tool for you.



1.2 Install the Vivado Design Suite

This section explains the installation process for all platforms for the Vivado Design Suite. Couple points before extracting and installing the tools. Make sure your machine can support this toolchain. You can check this on Xilinx website. Also, make sure that you have enough space on your system for this tool. It might take up to a few Gigabytes.
Now, let's start the installer you have downloaded in the previous step. After you click on your installer it will extract the tool and prepare it for the installation.

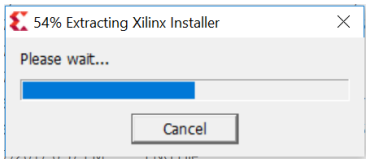
To install Vivado on Linux:

- Open a terminal and run:

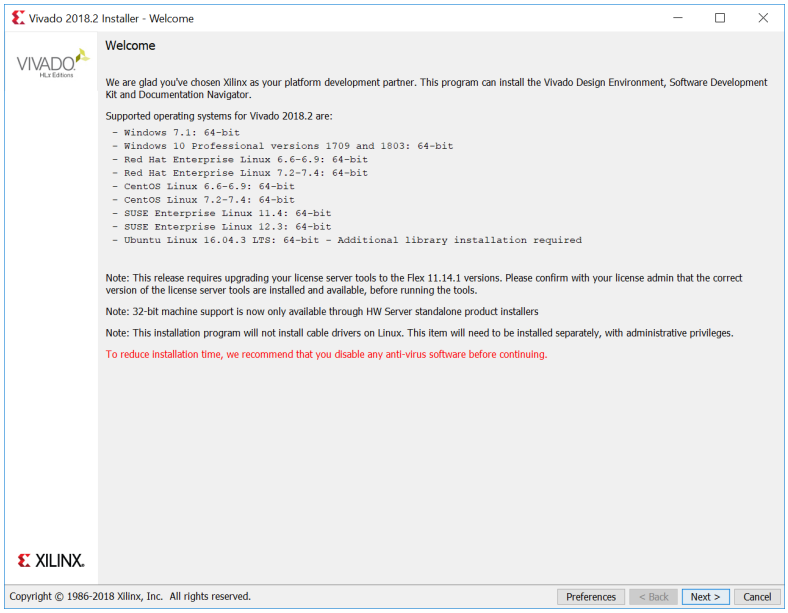
```
$chmod +x Xilinx_Vivado_SDK_Web_2018.2_0614_1954_Lin64.bin

$sudo ./Xilinx_Vivado_SDK_Web_2018.2_0614_1954_Lin64.bin
```

- Vivado Installer window will appear. The instalation process is same as windows.
- After installing Vivado, open a terminal and type these two commands to run the software :

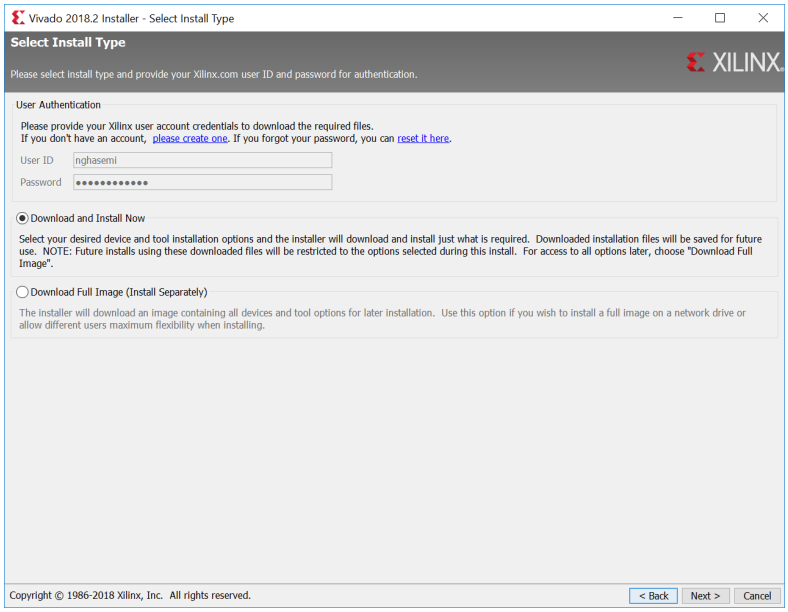
```
$source /opt/Xilinx/Vivado/2018.2/settings64.sh

$vivado
```



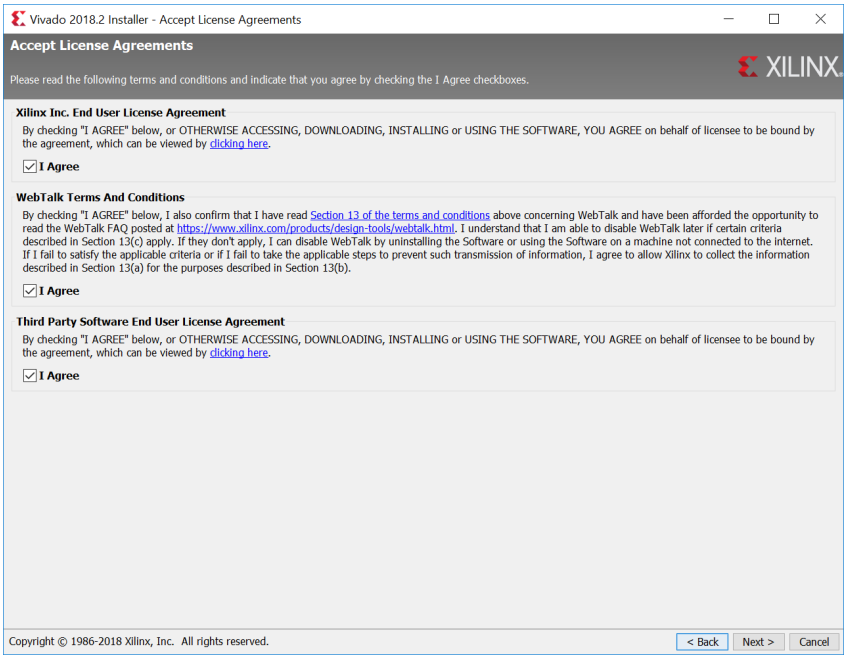
Click Next.



A window will pop up asking about your installation type and your Xilinx account. You can use the account you have made at the beginning on the Xilinx website. Select the "download and install now" option. Click Next.

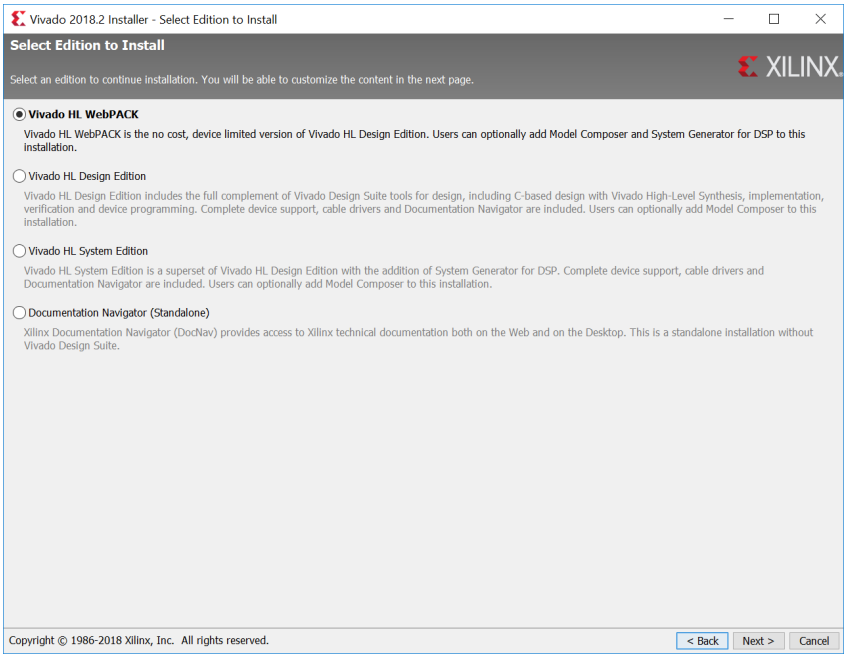


Check "I Agree" boxes. Click Next.

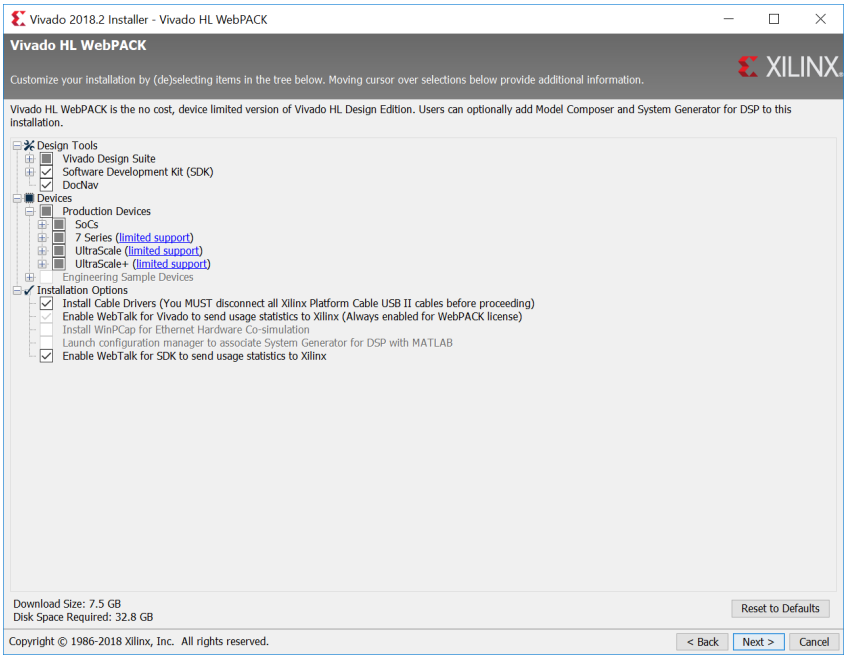


1.3 License and Edition Selection

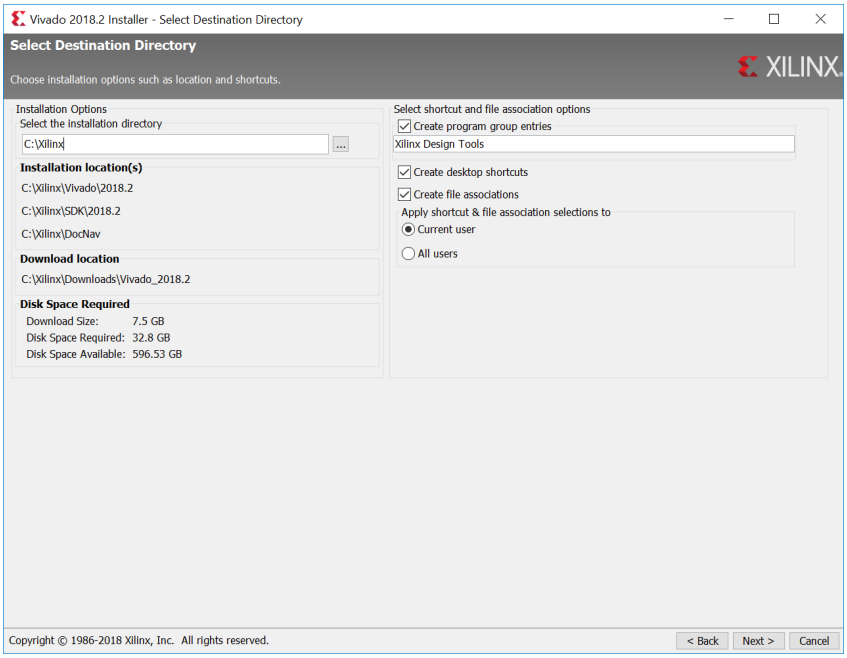
In the next step, you would select which edition of the Vivado tool you want to install. For now, we can install the webpack version which doesnt require any license.



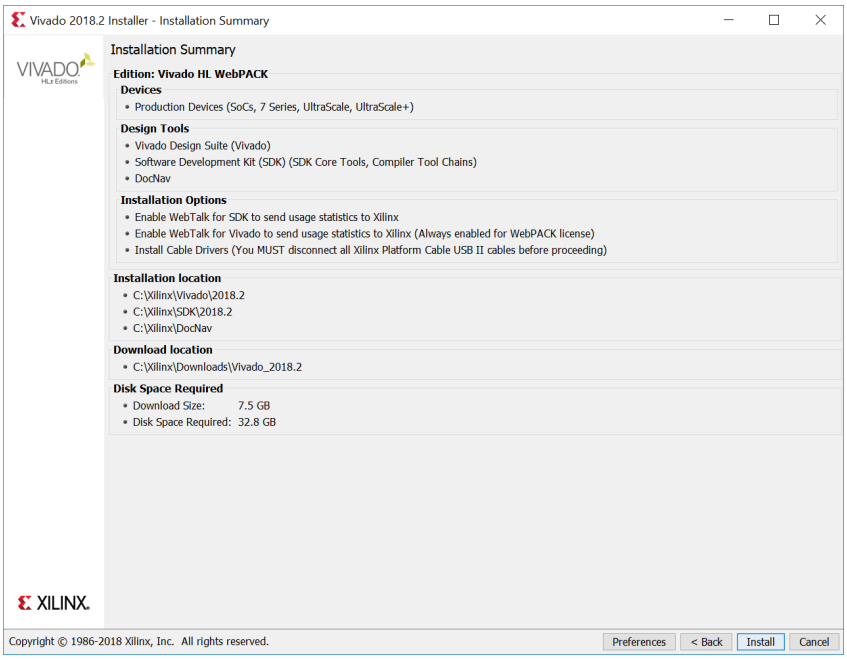
In the Design Tools and Devices, the default options are ok. Click Next.



Select the installation directory. Click Next.

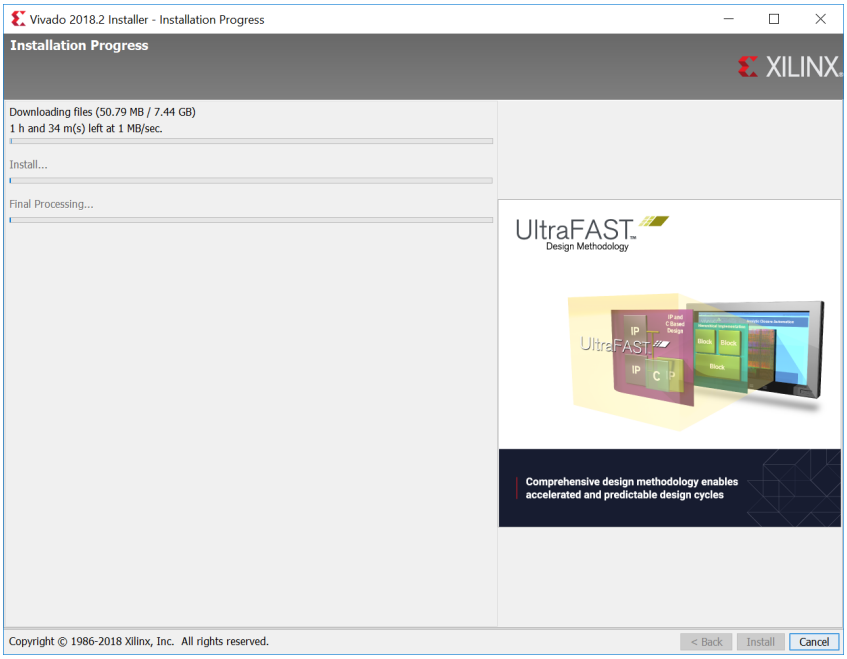


Click Install.



1.4 Download and installation

The final section of the installation will start to download the full image of the Vivado tool which even with a fast internet connection will take a long time. Try to do this part before the class. After downloading the full image the installation process will begin.



Use the Vivado software on the EECS server:

- Connect to one of below Linux Servers
  - zuma.eecs.uci.edu
  - laguna.eecs.uci.edu
  - crystalcove.eecs.uci.edu
  - bondi.eecs.uci.edu
- Open a terminal and loging with your username

  $ssh -X -Y UserName@ServerName

- Type below command in the terminal to setup Vivado 2017.1

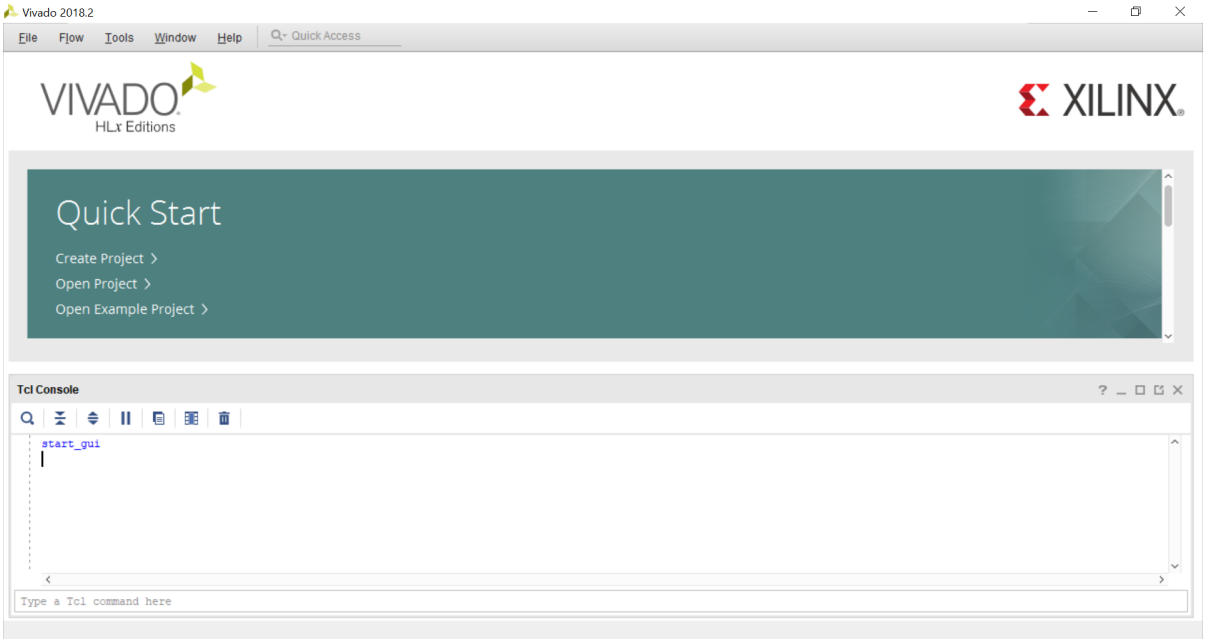
  $source /ecelib/eceware/xilinx/Vivado/2017.1/settings64-Vivado.csh

- Type vivado to lunch the application

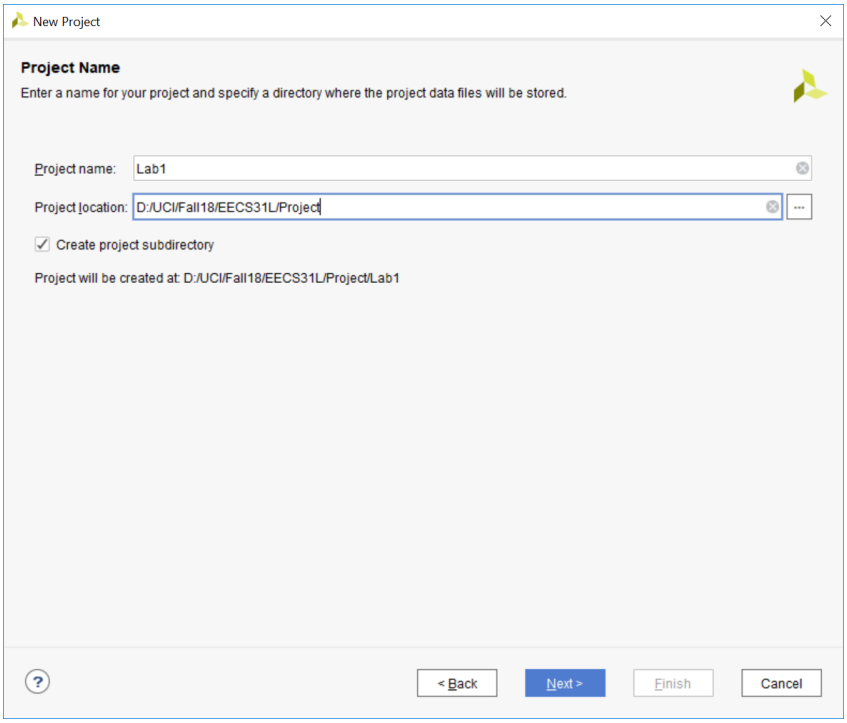
  $vivado

2 Using The Vivado Design Suite

Open Vivado, and select Create Project. Click next.



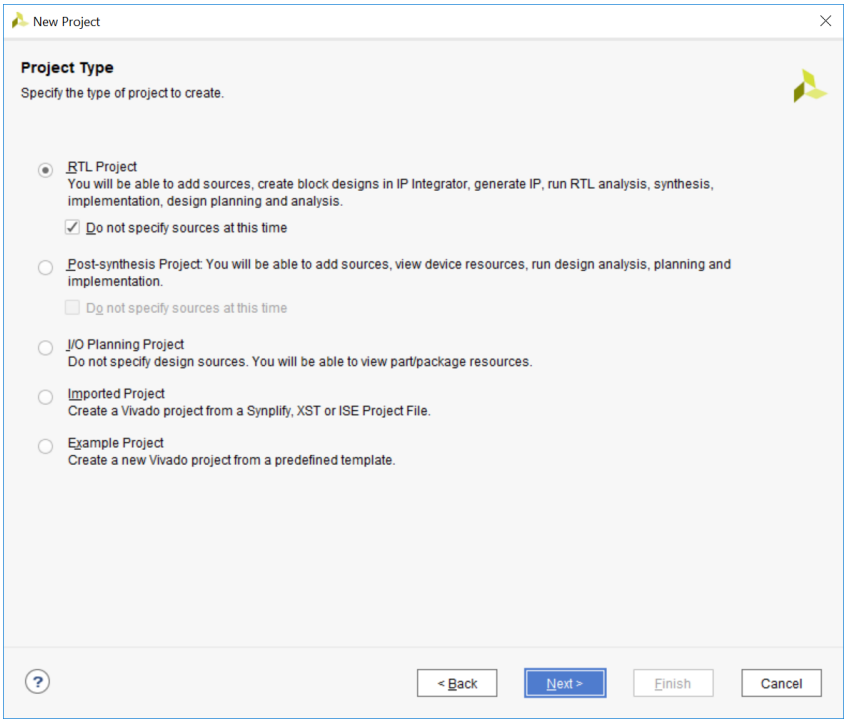
Your new project needs a name, an address (location on your computer), and a collection of design source codes. This new window asks you to choose a name for your project, and specify the directory that you want to save your project in.



Vivado Naming Conventions:

Project names and source file names must start with a letter ($A-Z, a-z$) and must contain only alphanumeric characters (A-Z, a-z, 0-9) and underscores(_).

This new window asks you to determine the type of your project. Select "RTL Project" option in the Project Type form, and click Next.

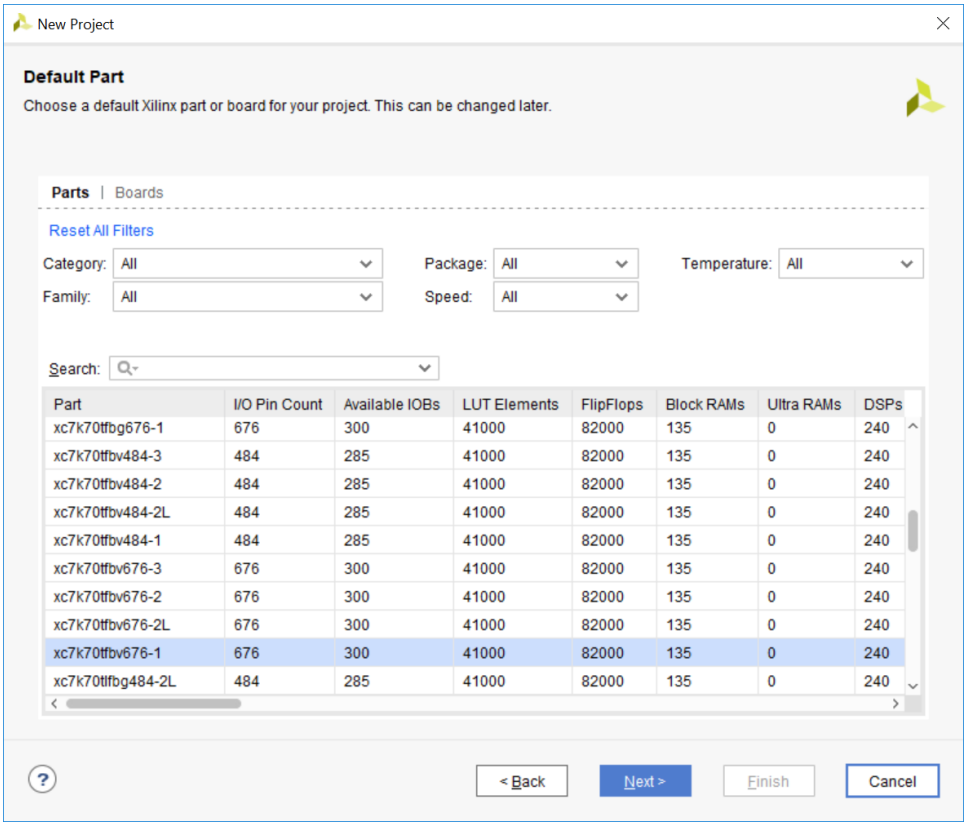


Note:

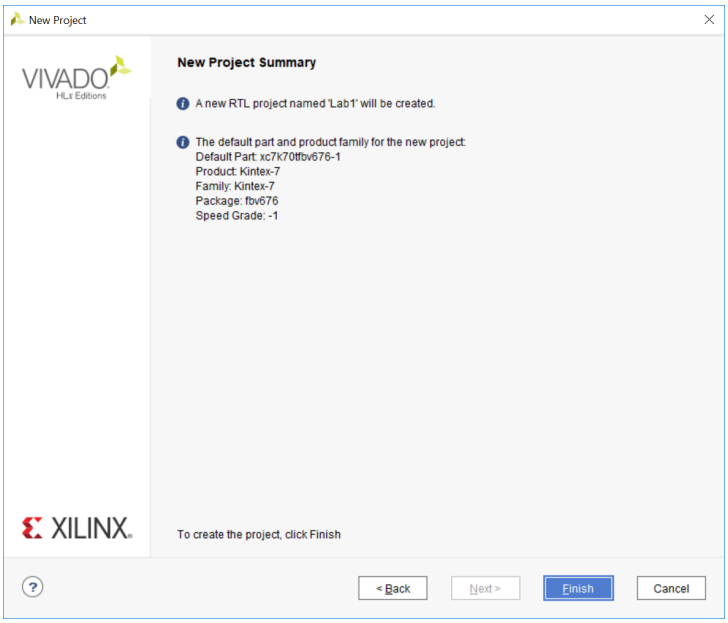
You can define different types of projects in Vivado.

- RTL Project - RTL to Hardware Validation
  - Import RTL and IP to process design all the way to hardware
  - Standalone IP - Create reusable preconfigured IP module
  - Device exploration - Empty project to examine device resources
- Post-synthesis Project Netlist to Hardware Validation
  - Third Party synthesis
- I/O Planning Project Early I/O Exploration and Assignment
  - Create I/O port manually or import CSV, RTL, or XDC
  - Can migrate to RTL Project
- Imported Project - Migrate Project from Synplify, XST, or ISE Project
  - Imports sources and compilation order
  - No synthesis or implementation result imported
  - No tool setting migrated
- Example project
  - Using Vivado examples

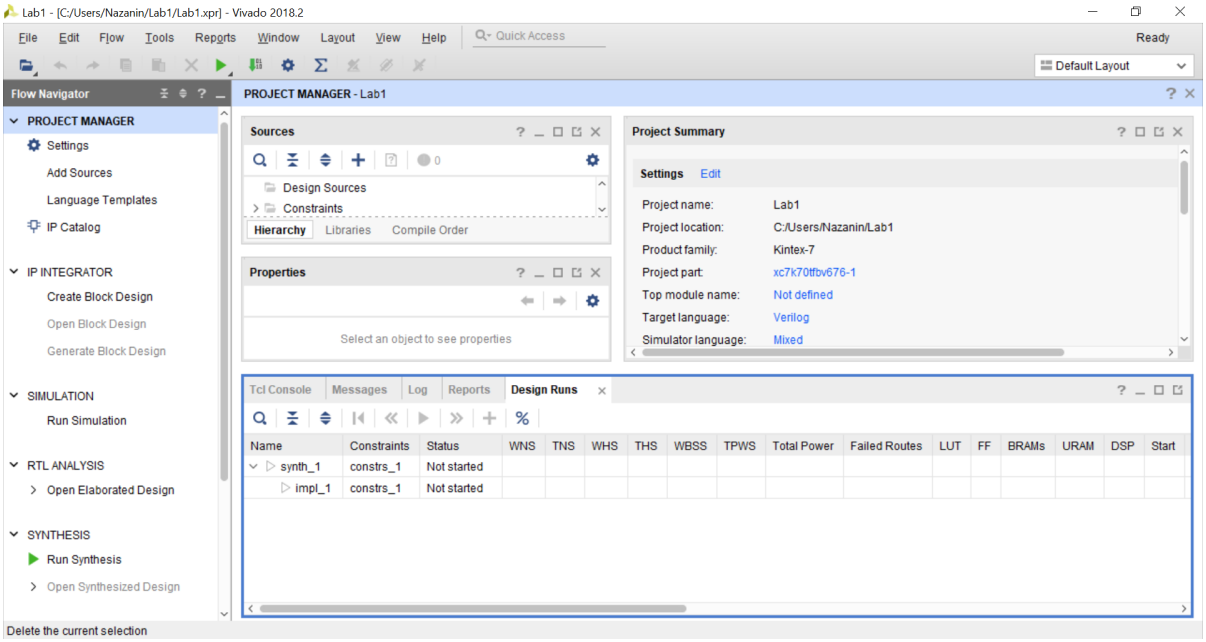
In the Default Part dialog box, you should select the family and part number of the FPGA you want to implement your code on. Since through this course we are not going to work with any FPGA's, you may skip this step and click Next.



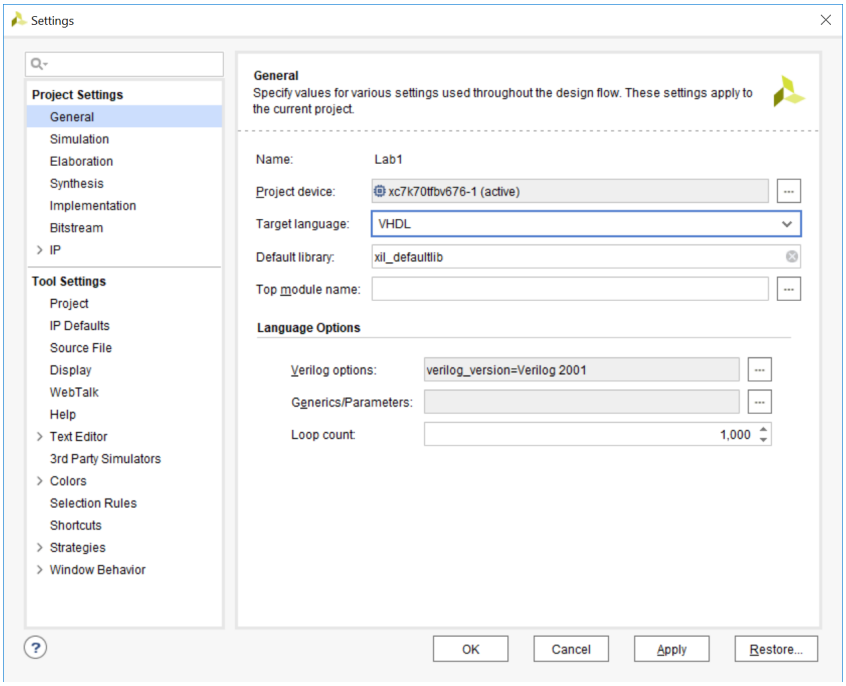
Review the project summary in the New Project Summary dialog box before clicking Finish to create the project.



You see different windows in the Vivado software: Flow Navigator, Sources, Project summary, and Design Runs console.



The Project summary will give you the status of your project.
Through this course we are going to use VHDL language for hardware design. So, go to the Project Summary window and click Edit. Then change the target language to VHDL.



Click Apply. Click Ok. Now in your Project Summary window, the target language should be VHDL.