

SoC Design

Introduction to Xilinx Vivado on Windows

California State University Long Beach

1 Gole

This tutorial explains how to install and use Vivado Design Suite to develop and simulate Verilog model codes.

2 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. You can download and install it from here:

<https://www.xilinx.com/support/download.html>

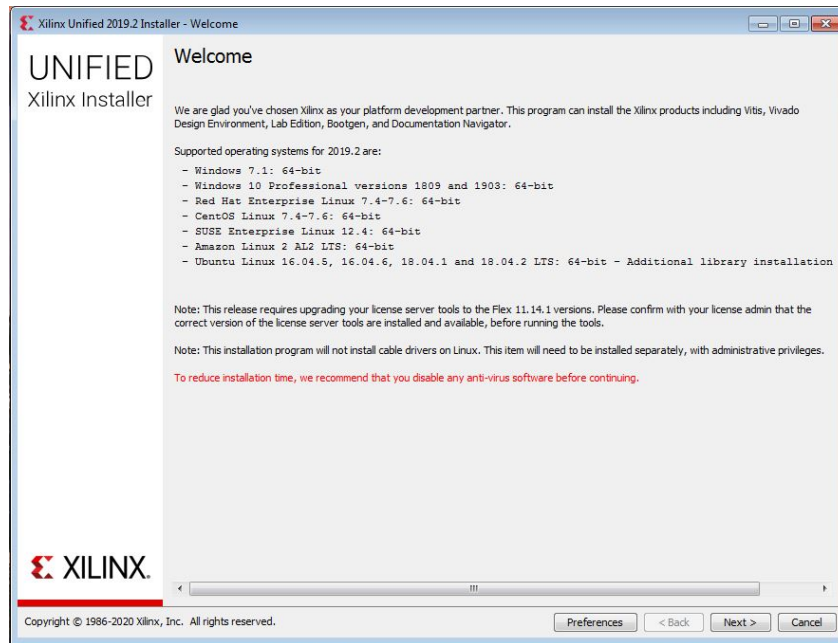
2.1 Downloading the Xilinx Installer

Select the appropriate version for your laptop operating system. You would need a Xilinx account to download the tools. Register with your school email (CSULB email address) 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. Choose "Xilinx Unified Installer 2019.2:Windows Self Extracting Web Installer" to download the installer version 2019.2 for 64-bit Windows machine.

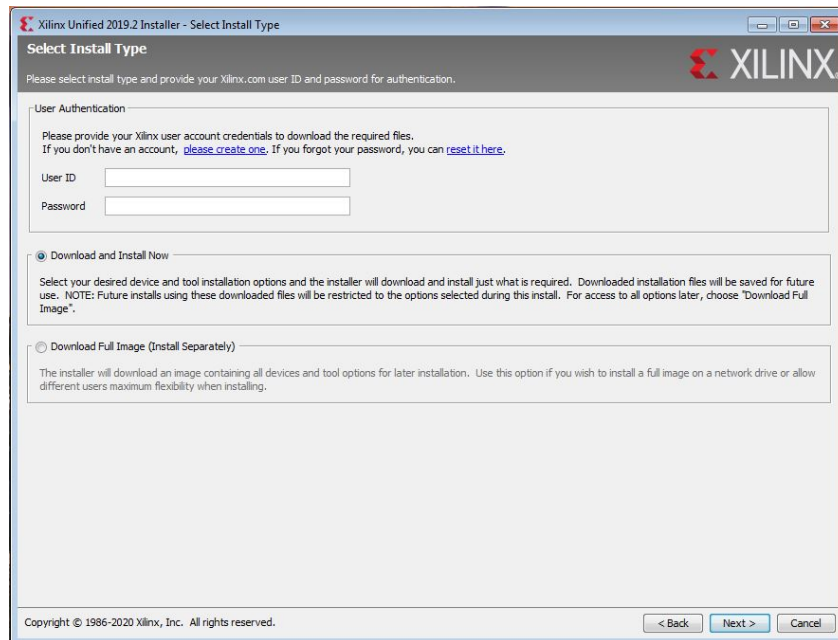
2.2 Installing the Vivado Design Suite Tools

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. Also, make sure that you have enough space on your system for this tool. It might take up to 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 installation.

Click Next.

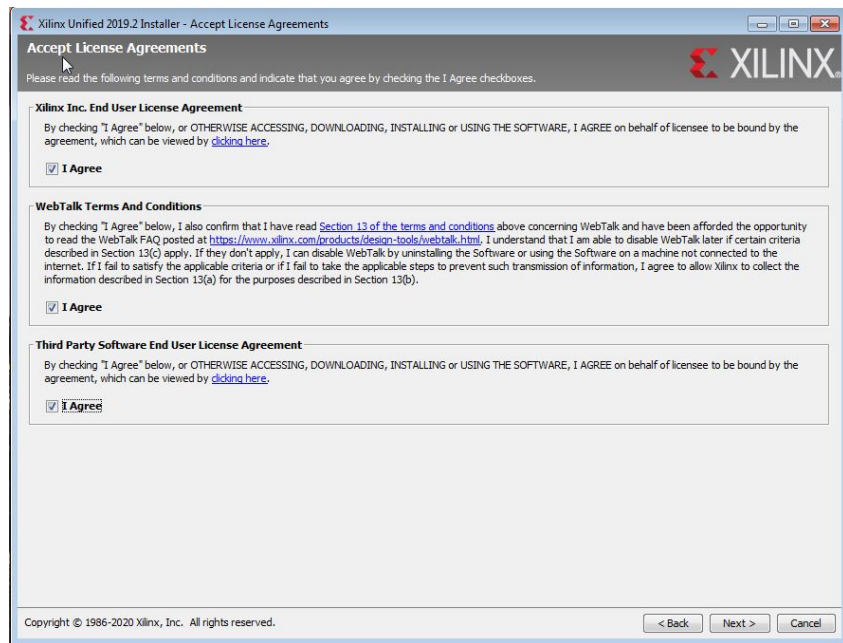


A window will pop up asking about your installation type and your Xilinx account. You can use the account you've made at the beginning on the Xilinx website. Select the download and install now option.

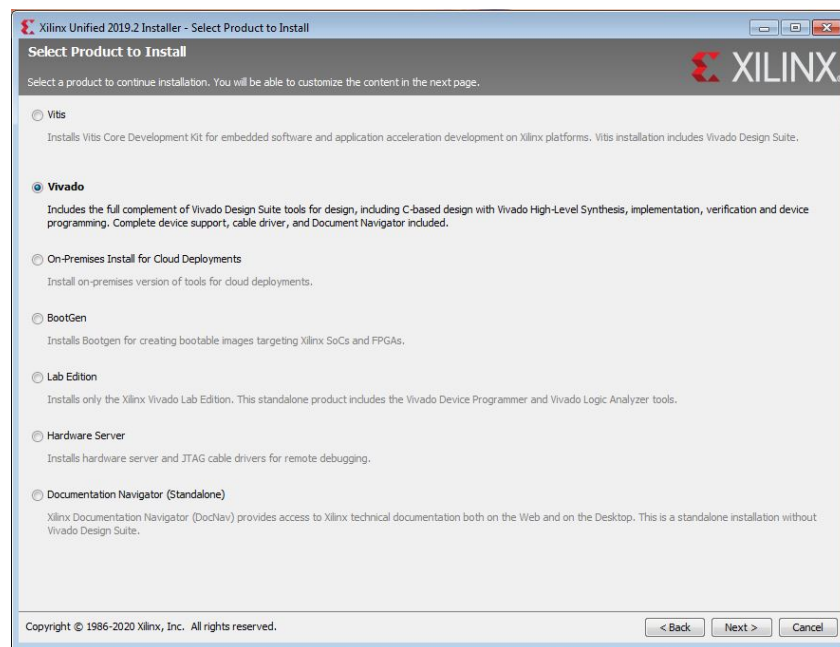


Click Next.

Check "I Agree" boxes. Click Next.

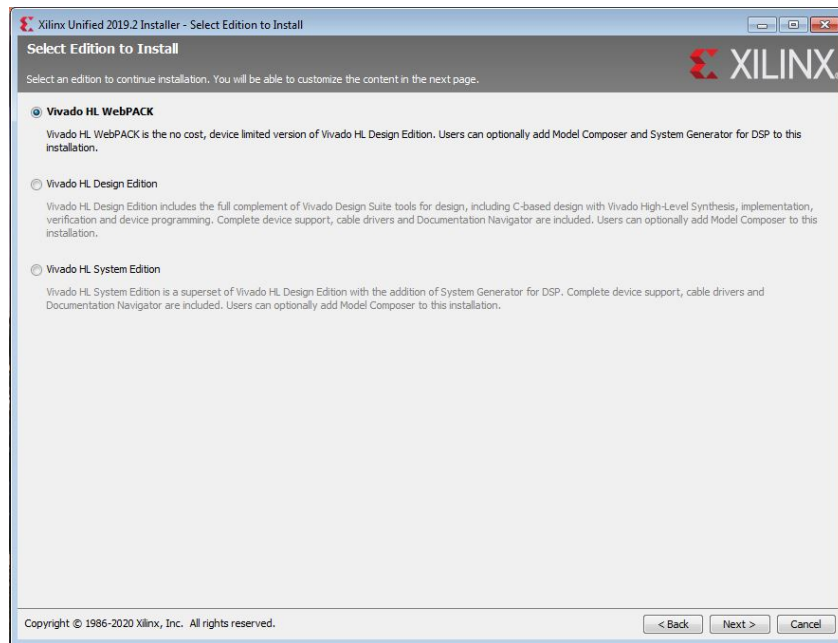


Choose Vivado. Click Next.

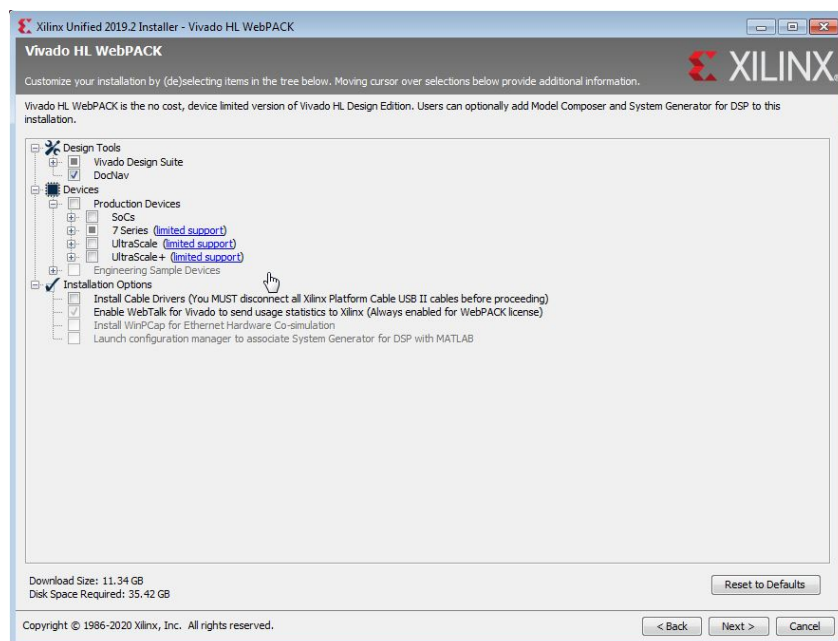


2.3 License and edition Selection

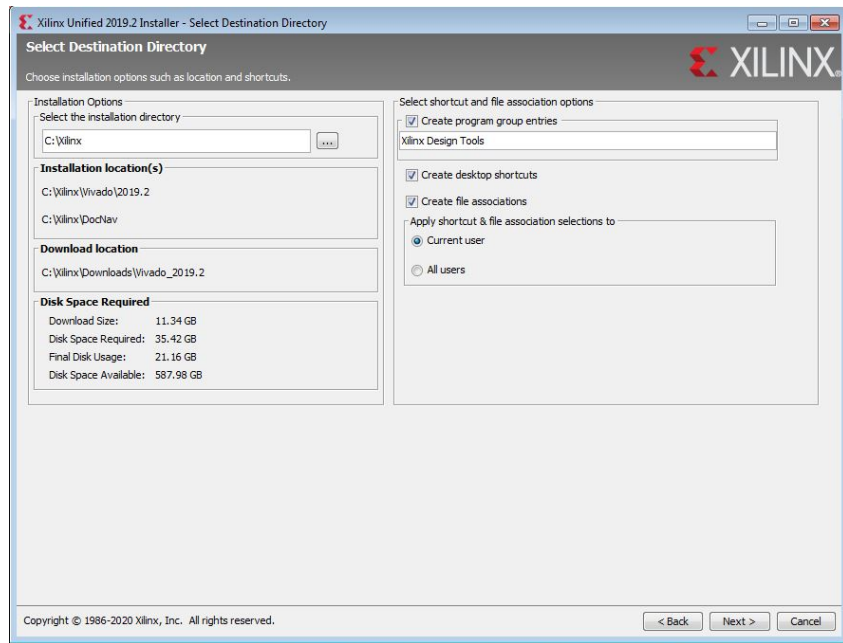
In the next step you would select which edition of the Vivado tool you want to install. For now, we can install the webpack installation which doesn't require any license.



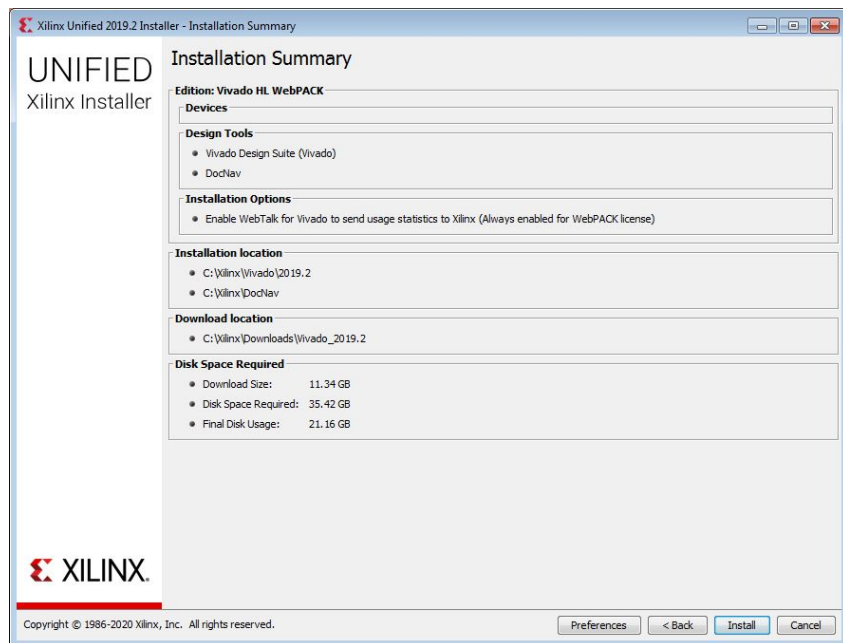
In the Design Tools and Devices the default options are ok. Click Next.



Select the installation directory. Click Next.

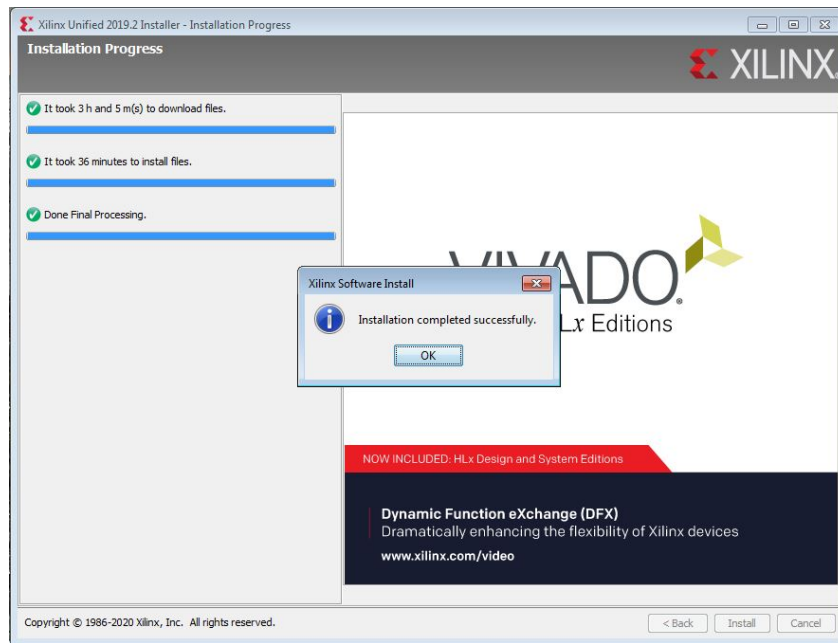


Click Install.



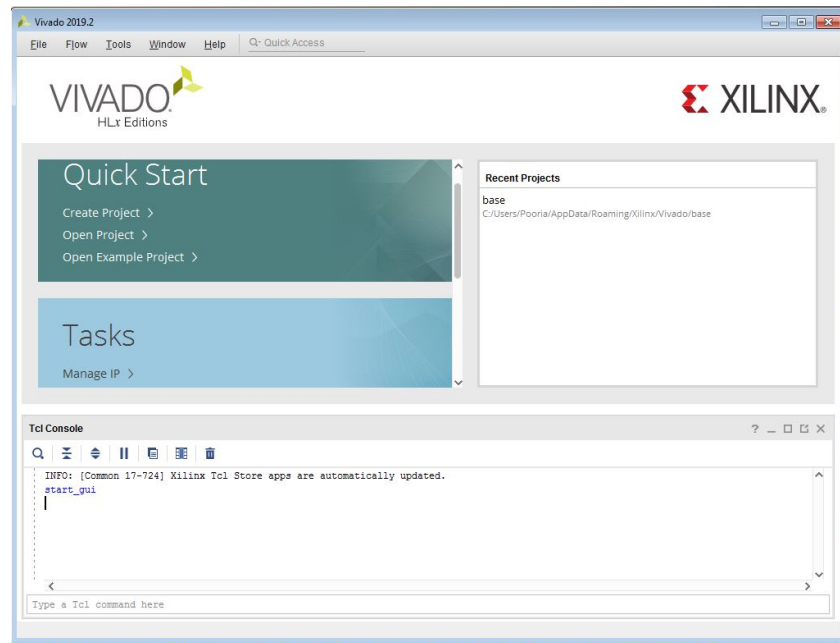
2.4 Download and installation

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. After downloading the full image the installation process will begin.



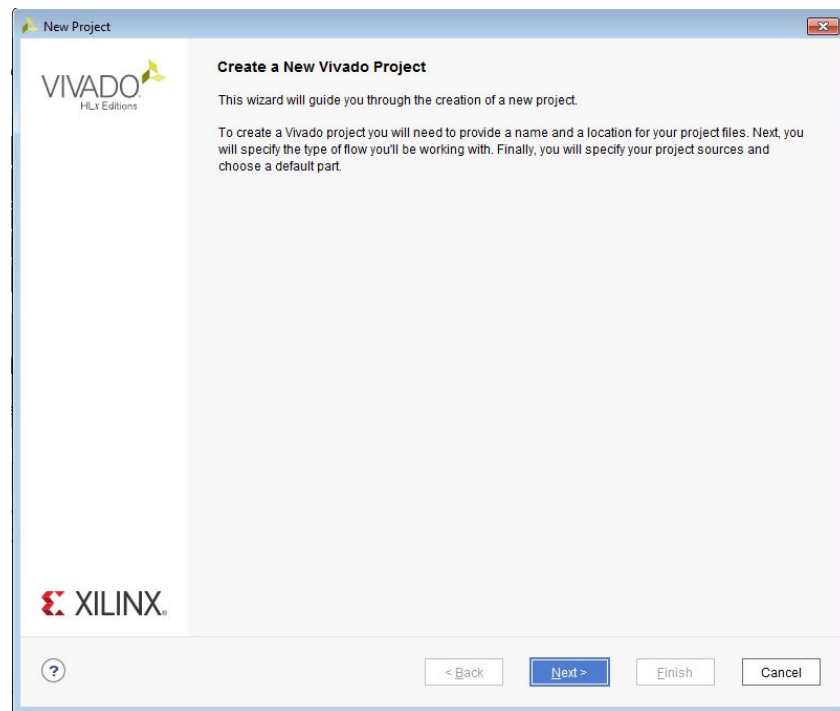
3 Using The Vivado Design Suite

Run Vivado by double-clicking the Vivado icon on your desktop. Your initial screen will be like this:



3.1 Create New Project

To create a new project, click File, Project, New. A window will pop up, click Next.

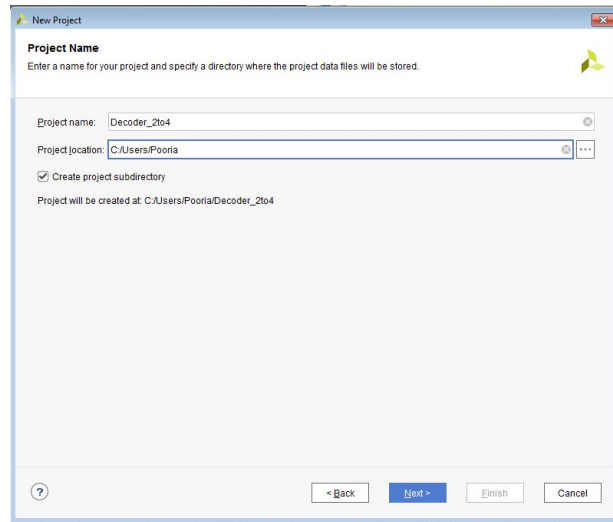


Every new project needs a name, an address (location on your computer), and a collection of design source codes. This new window asks you to choose a name for your project, and specify the directory that you want to save your project in.

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(_).

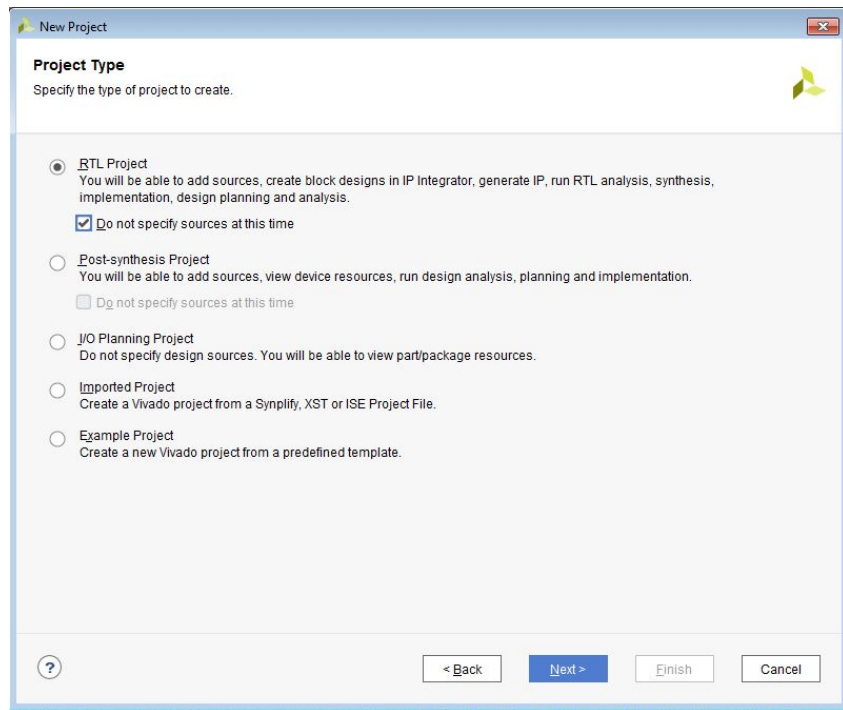
Click Next.



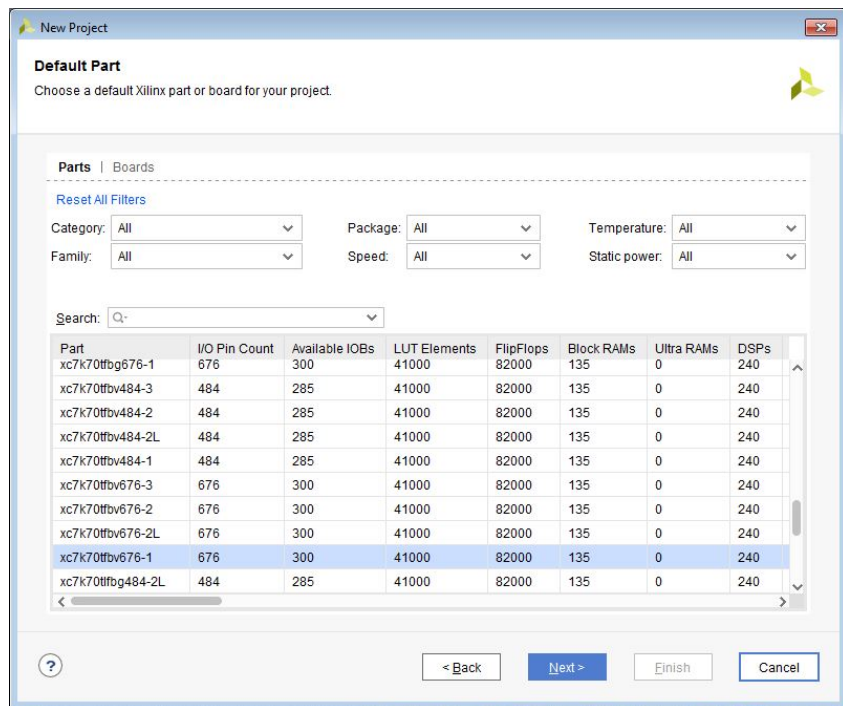
This new window asks you to determine the type of your project. You can define different types of projects in Vivado.

- 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

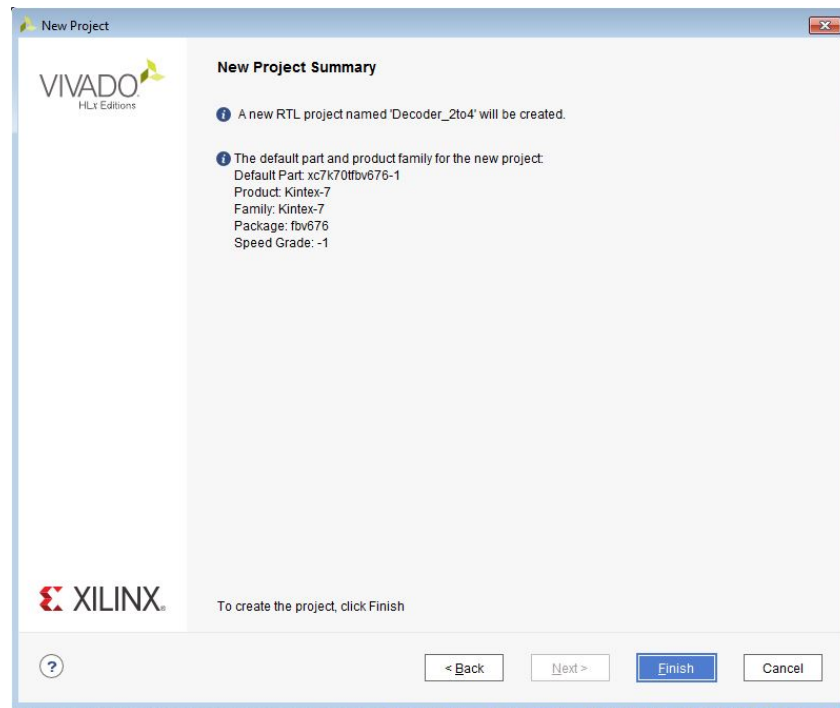
Select RTL Project option in the Project Type form, and click Next.



Click Next.



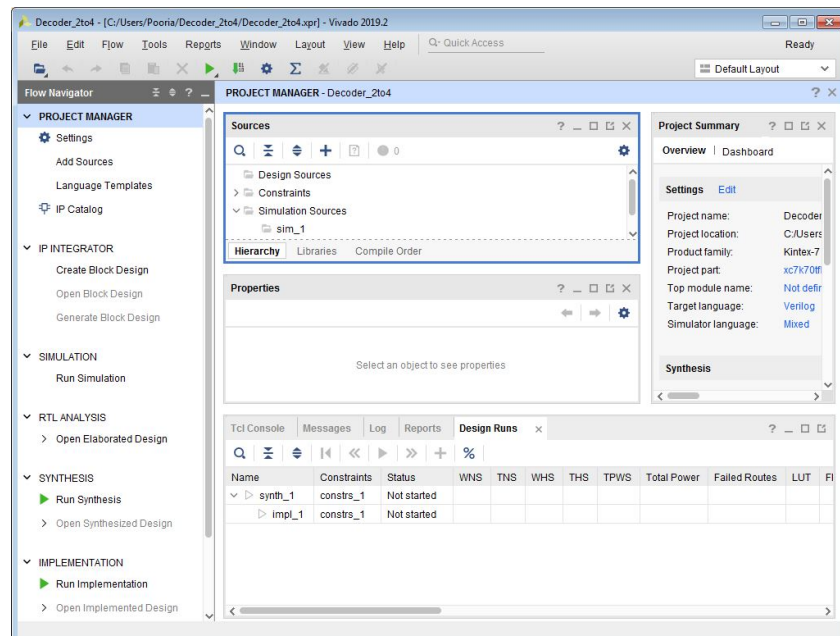
Hit finish button in new project summary window. This will create a project and give you the main Vivado tool homepage.



3.2 Vivado Homepage

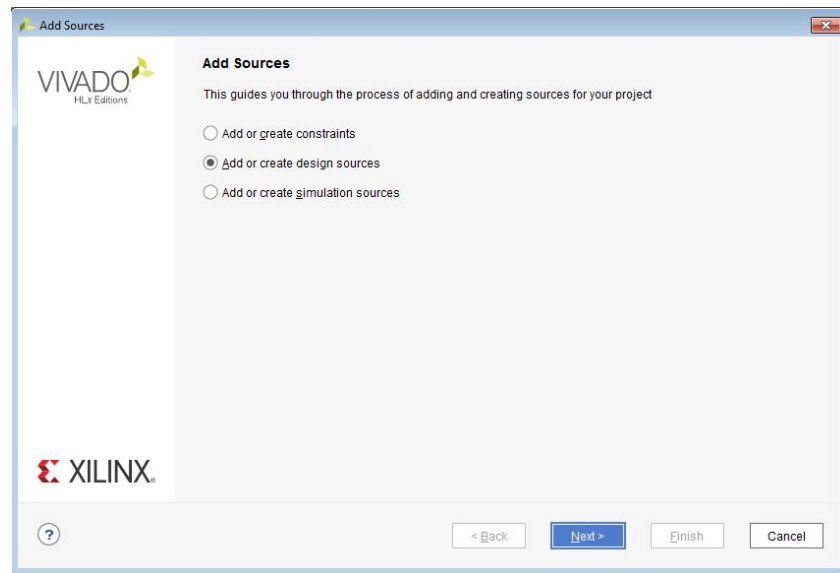
There are different parts here: flow navigator, sources, design summary, and design runs console. Use flow navigator to create your block design, Add sources, and run a Simulation. The project summary will give you the status of your project.

In the source window, you can manage your design sources and check the hierarchy.

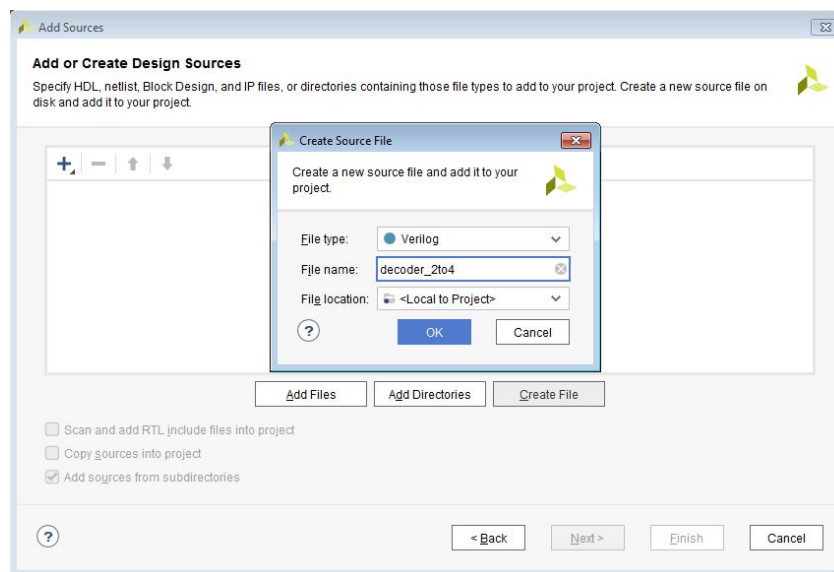


3.3 Add Source

From flow navigator window, choose Add Sources. We want to design a 2:4 Decoder. So, we need to create a design source. Click Next.



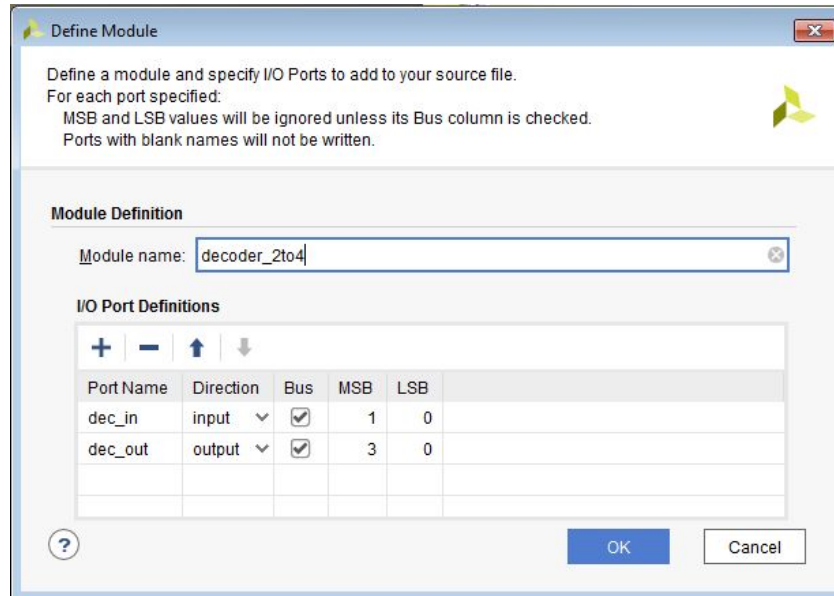
In the next window you can add files you already have or you can choose to create a new file. To add a file select Add Files, choose `decoder_2to4.v`, and click OK. Here we are going to create a new file. Click create file, find Verilog from file type menu, and choose a name for your module. Click Ok and Finish.



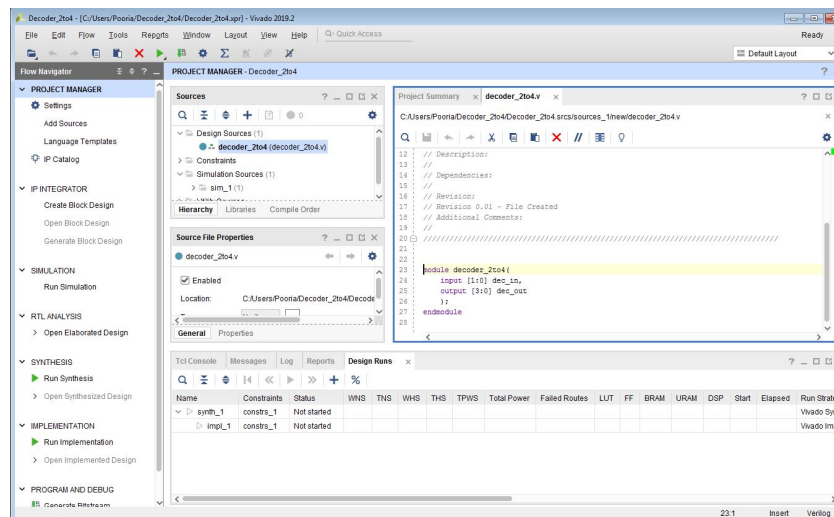
This new window will ask you to define I/O ports for your module. To design a 2:4 decoder we need 2 bits for input (I named it here as (`dec_in`), and 4 bits outputs(`dec_out`)). For every port you define:

- A name,
- The Direction of that port (if it is input port or output),
- The number of bits you need for that port (if this number is more than 1 check the bus box),
- order of bits in that port (which bit is the Most Significant Bit and which one is the Least Significant Bit)

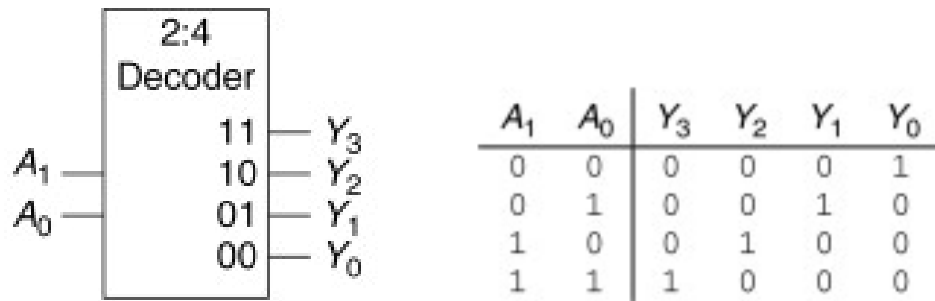
you also have this option to not define any port here and write down everything later in your code. Click OK.



Now you can see your new source file (**decoder_2to4**) under design sources in the Sources window. Double click on the **decoder_2to4** to open it in a new window. Then we are going to write our code.

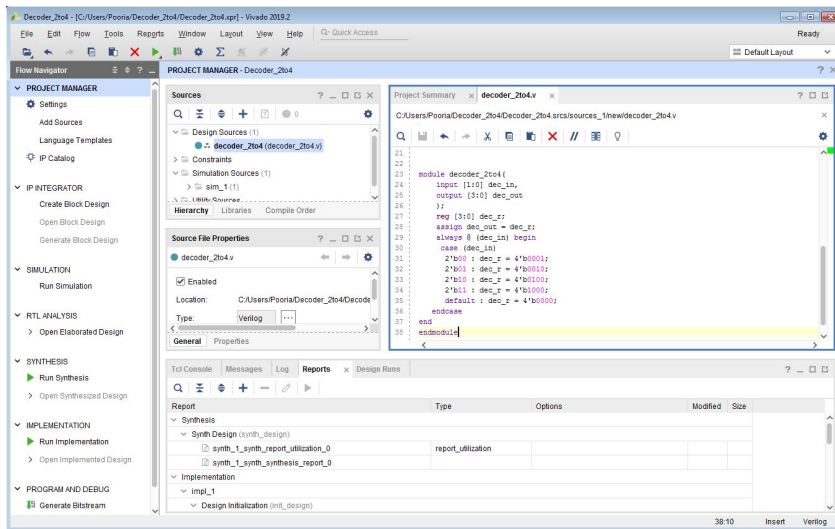


Note: A decoder has N inputs and 2N outputs. It asserts exactly one of its outputs depending on the input combination. You can see the block diagram and truth table for a 2:4 decoder here.



3.4 Write the Code for your Design

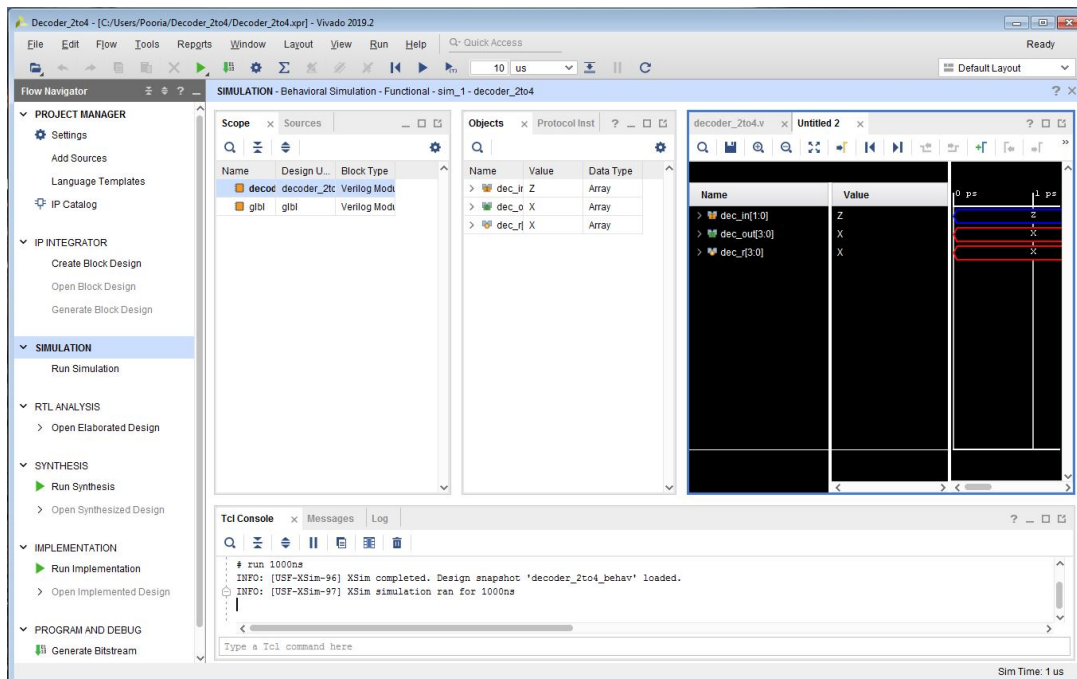
Now we are going to write the code for our decoder. If you have a syntax error in this part. The line which contains error will be underlined by red color. Make sure to correct all of them. Your code should look like this. Save your code.



```
module decoder_2to4(
    input [1:0] dec_in,
    output [3:0] dec_out
);
    reg [3:0] dec_r;
    assign dec_out = dec_r;
    always @ (dec_in) begin
        case (dec_in)
            2'b00 : dec_r = 4'b0001;
            2'b01 : dec_r = 4'b0010;
            2'b10 : dec_r = 4'b0100;
            2'b11 : dec_r = 4'b1000;
            default : dec_r = 4'b0000;
        endcase
    end
endmodule
```

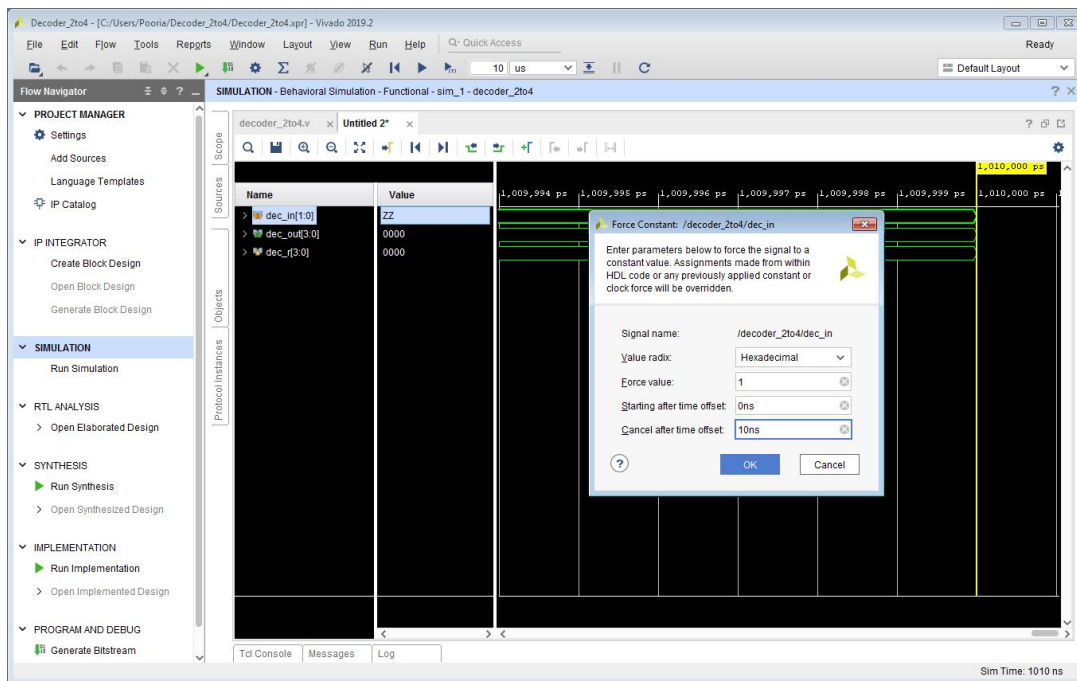
3.5 Simulation

To simulate your design, click on the simulation button in flow navigator window. Choose Run simulation, run behavioral simulation. After successful simulation, the simulation window opens.



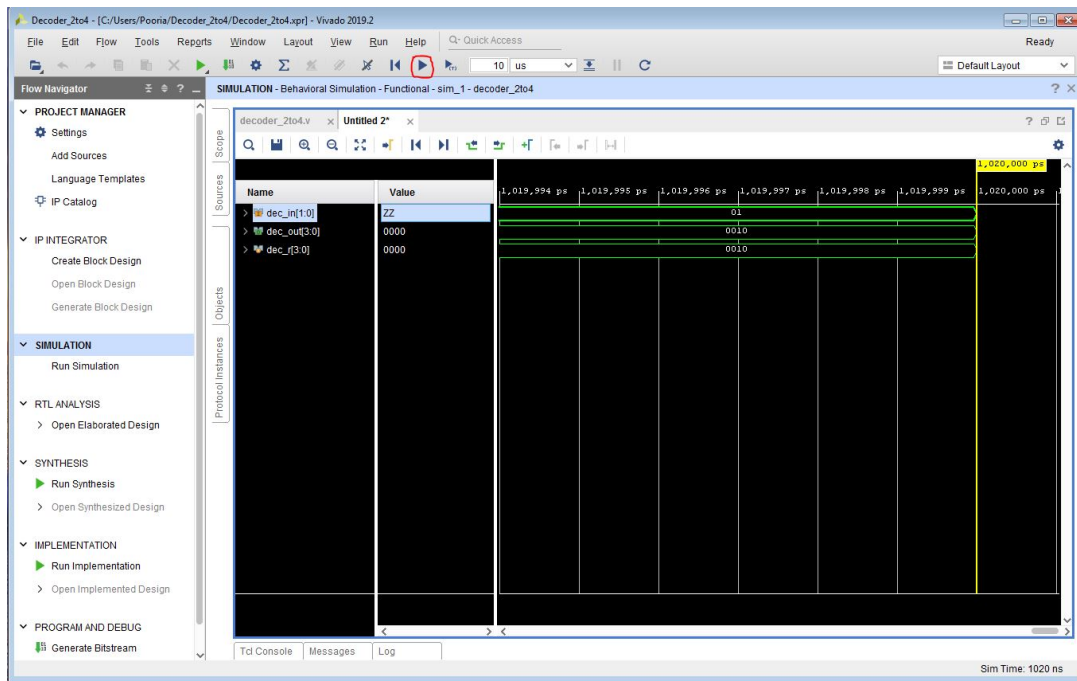
You can check your design function through this window. And if it is necessary, go back to your design code and change it.

In the Object window right click on `dec_in` and select "Force Constant ...". choose a value for the input (`dec_in` has 2 bits so you can choose 0,1,2, or 3), and set the starting and cancellation points. Click OK.



Push the Run all button and enlarge the waveform.

You can see the result here. When input is 1 the output is 0010 (as we had set in our code). You can redo this part for all 4 inputs (0,1,2, and 3) to see the output in the waveform.



4 Vivado on Mac

Run Windows or Linux on a Virtual Machine. Download either Windows or Linux web installer.

Watch the following YouTube video for more help: <https://www.youtube.com/watch?v=aJcc-xC6krE>