CENG 232

Brief HDL introduction,
Xilinx Installation Guide,
Basic Properties of Xilinx,
Sample Demonstration

Hardware Description Language

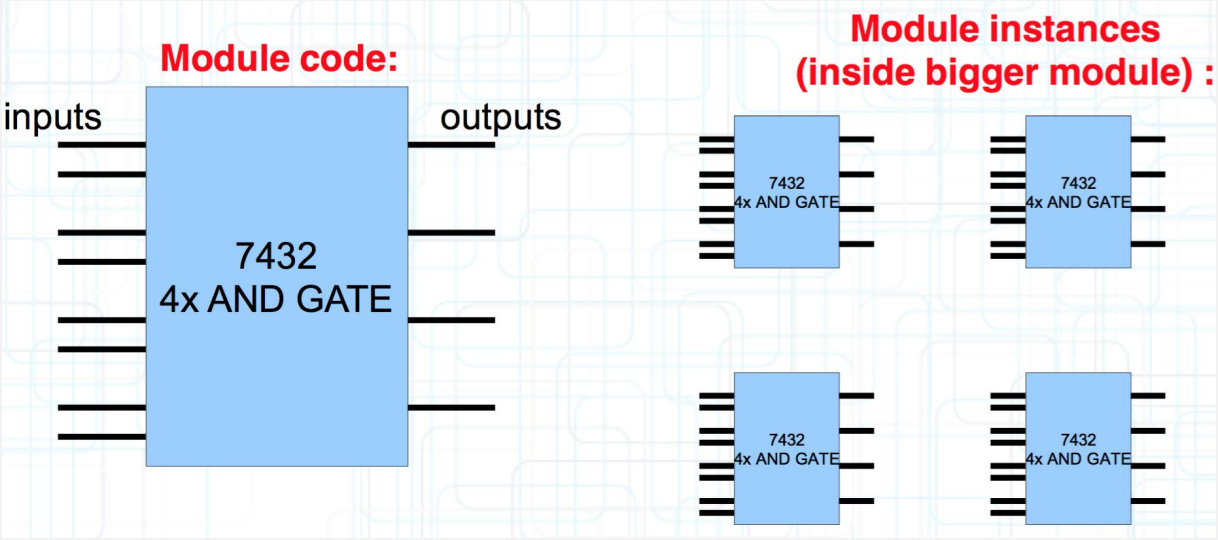
- Designing schematics wire by wire is:
  - Hard
  - Error-prone
  - Gets more complex as project gets bigger
- HDL
  - Describe your circuits as textual “code”, just like any programming language
  - Divide larger projects into smaller modules
  - Harder to visualize, but easier to manage

Hardware Description Language

- Two popular languages:
  - VHDL
    - Complex at initial glance
    - Provides better control when synthesizing to physical elements
  - Verilog
    - Simpler syntax
    - Easier to learn

Modules

- Modules are the building blocks of Verilog designs
  - Inputs
  - Outputs
  - Semantics between inputs and outputs

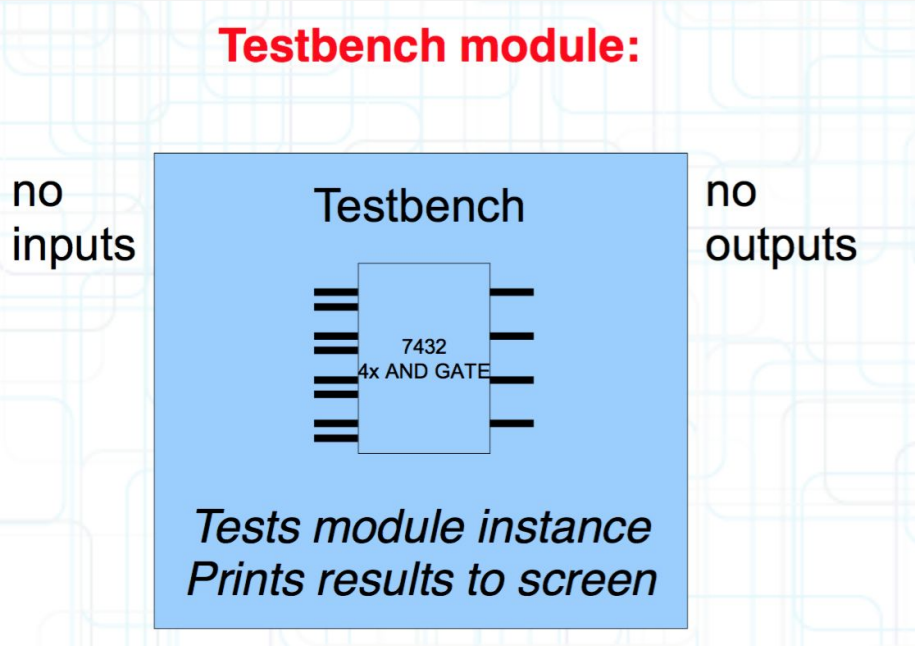


Testbench Idea

- HDLs provide you a way to write modules
  - with their inputs and outputs
- How to test it?
  - Create a test-module
    - instantiates your own module
    - writes to your module's inputs
    - reads from your module's outputs
  - Connect inputs and outputs of the test module to your module
  - Run the testbench

Testbench Idea

- Testbench modules
  - No inputs, no outputs (as connections)
  - Access to screen, keyboard, time
  - Access to tested module instance



Testbench Idea

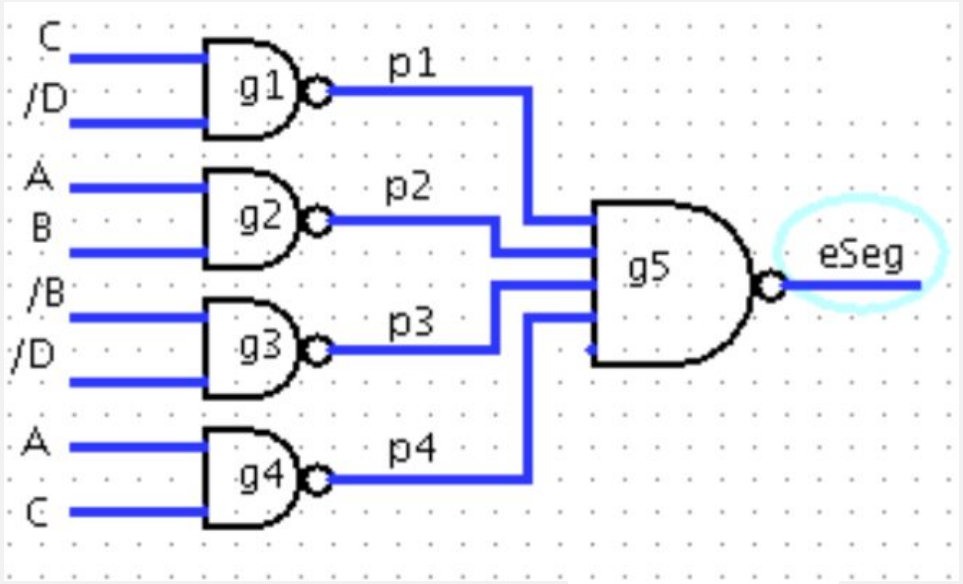
- Same as plugging a chip to a physical test board which tests various inputs and blinks green light if everything is OK.
- You don't have to blink a green light:
  - HDLs provide a way to write to screen:

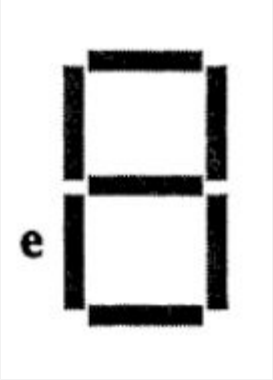
TEST1 passed
TEST2 failed
TEST3 passed
...

A Simple Verilog Example

```
module binary_to_eseg(
    input A, input B,
    input C,
    input D,
    output eSeg
    );
    wire p1, p2, p3, p4;
    nand2 g1(p1, C, ~D);
    nand2 g2(p2, A, B);
    nand2 g3(p3, ~B, ~D);
    nand2 g4(p4, A, C);
    nand4 g5(eSeg, p1, p2, p3, p4);

endmodule
```





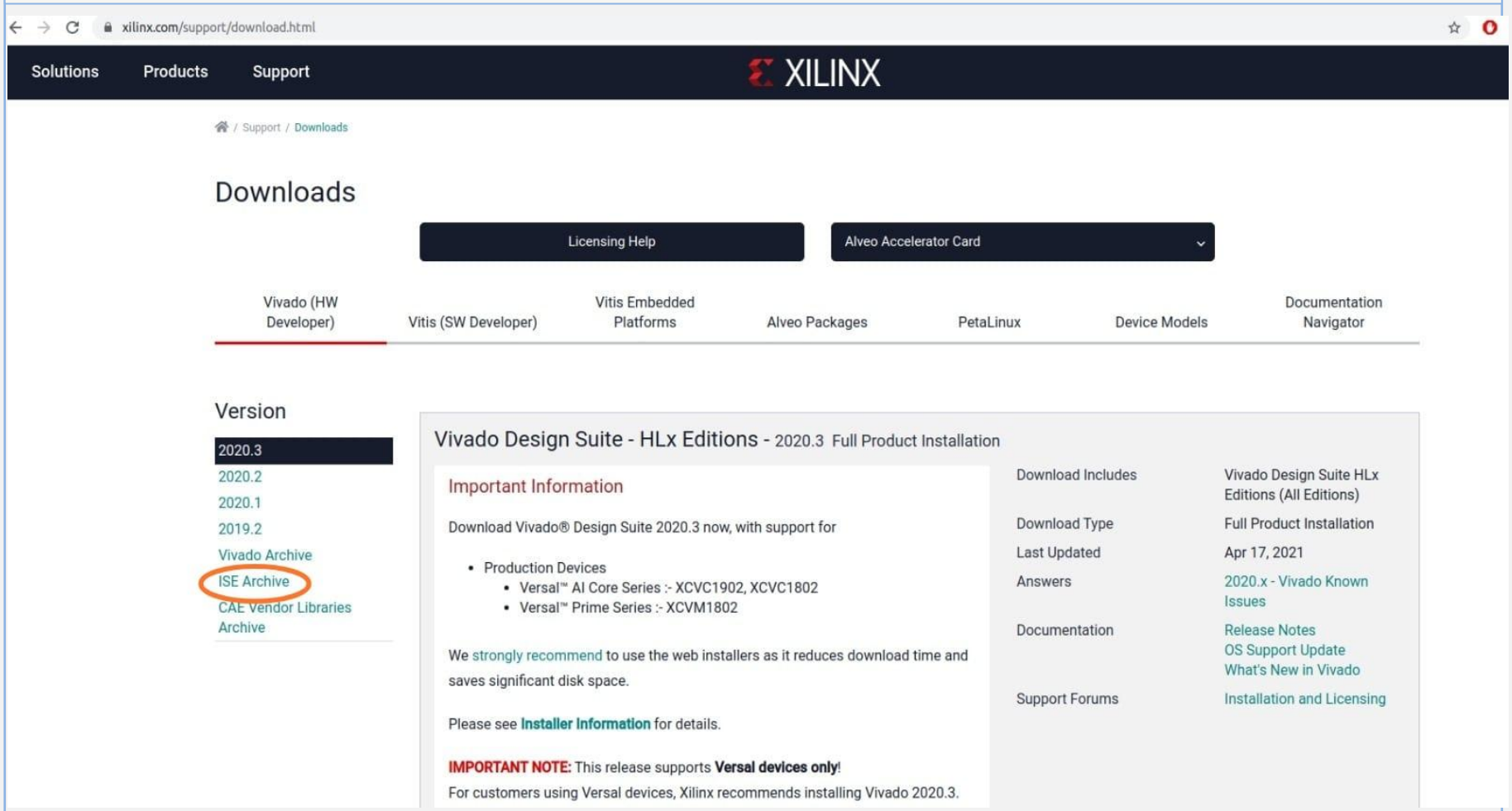
Tools

- Xilinx ISE WebPack (version **12.4**)
  - Integrated Development Environment
  - Editor, compiler, simulator, FPGA programmer
  - Linux & Windows
- Free to use
  - Installation requires free registration and obtaining a free license.
- https://www.xilinx.com/support/download.html
- We will use Xilinx version 12.4, please don't use other versions
- Already installed on inek machines (35-70).
- Type “Xilinx” on terminal screen to launch.

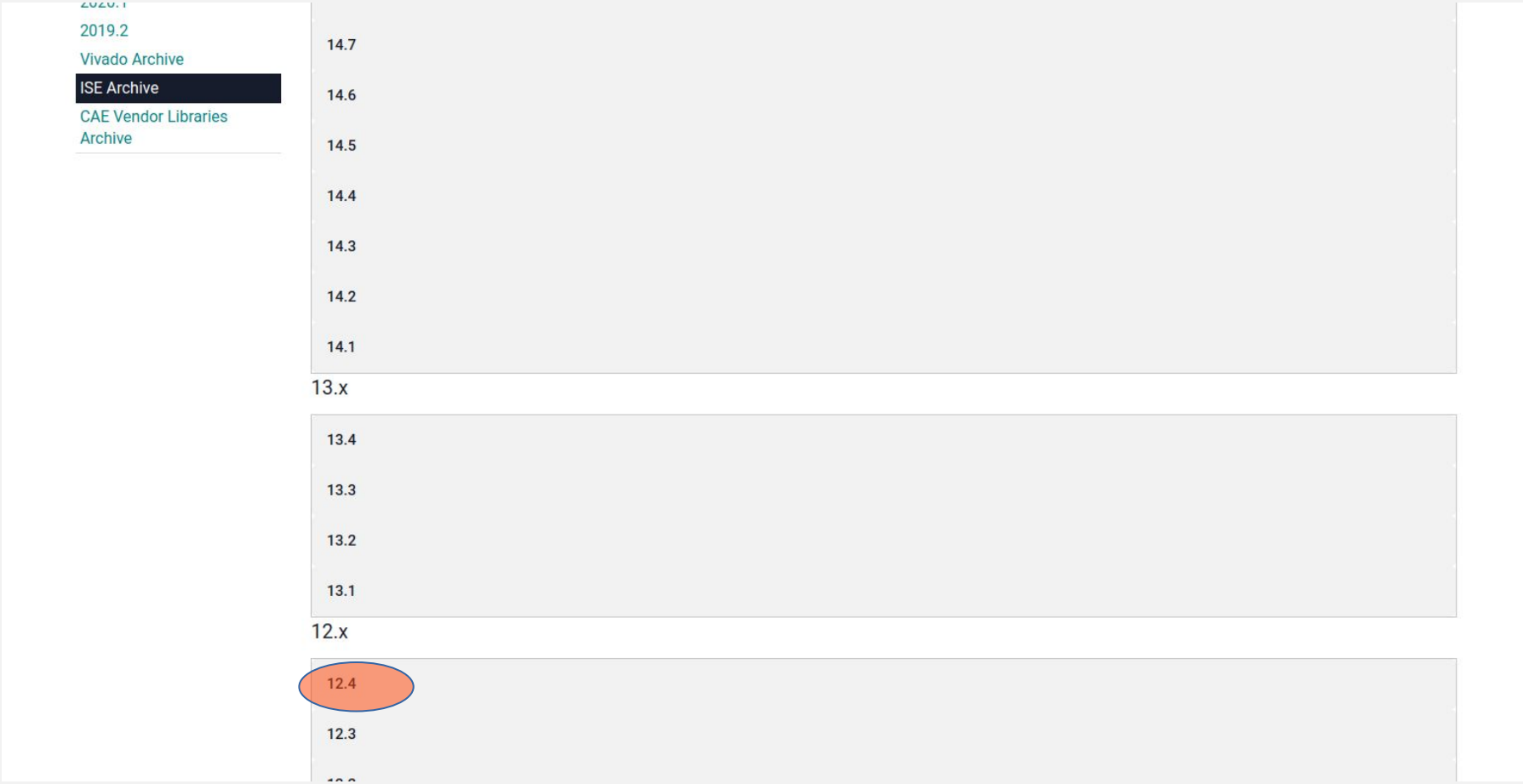
Tools

- You can also launch Xilinx on your own PC without installing it, by connecting to inek machines, using two SSH connections with -Y parameter in this order:
  - >> ssh -Y e1234567@login.ceng.metu.edu.tr -p 8085
  - >> ssh -Y inekXX (ineks 36-70, example: inek36)
  - >> Xilinx
- As the software now runs over the network, some lag may occur while interacting with the GUI.

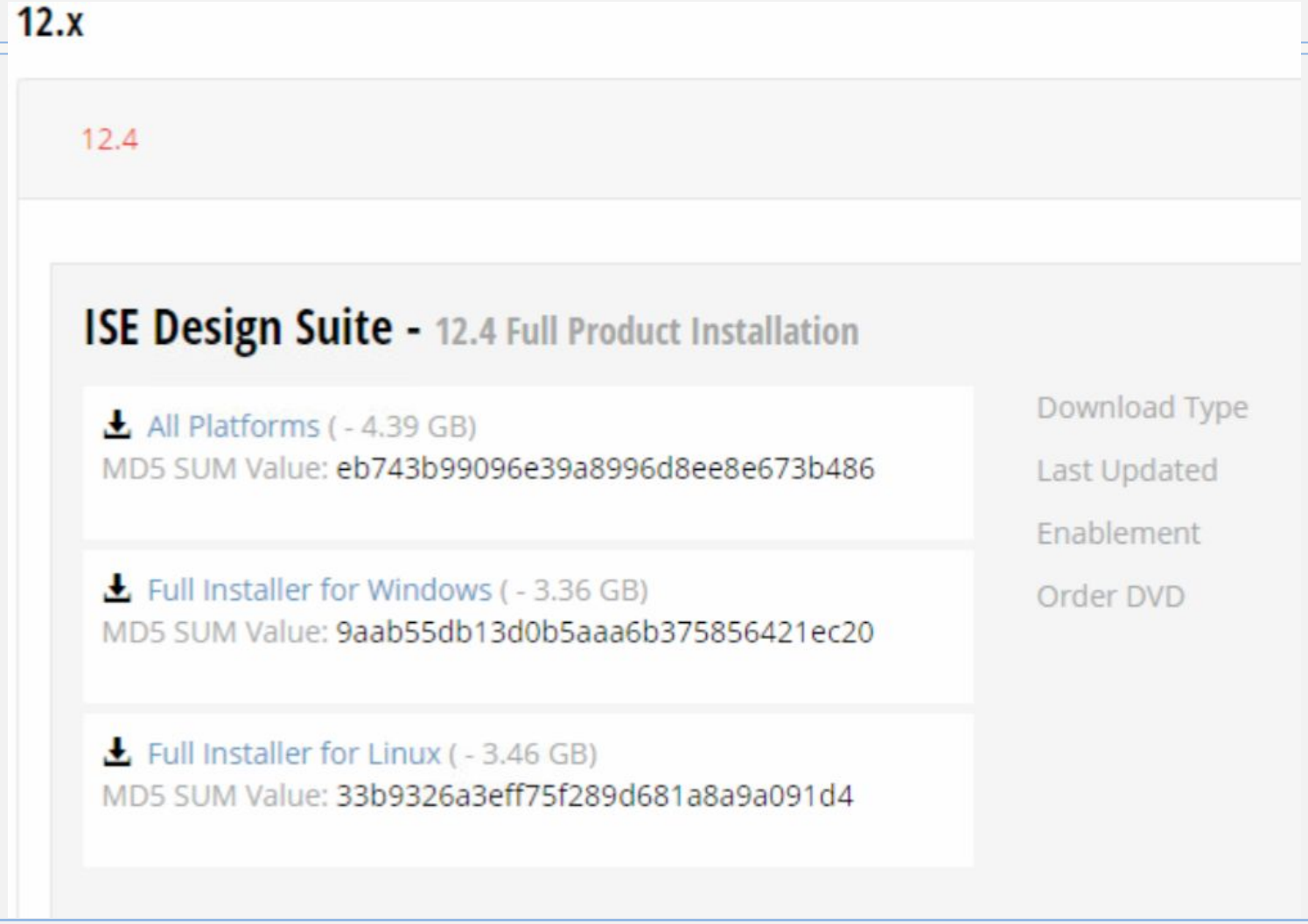
Installation



Installation (Cont.)



Installation (Cont.)



First Name*

Kadir

Last Name*

Cenk

Corporate E-mail*

kadir.alpay@metu.edu.tr

Company Name*

Metu

Address 1*

Metu

Address 2

Metu

City*

Ankara

State*

Cankaya

Please use 2-letter code for your US state or Canadian province.

Country*

Turkey

Zip Code*

1234

Phone

Job Function*

Student

Primary Market*

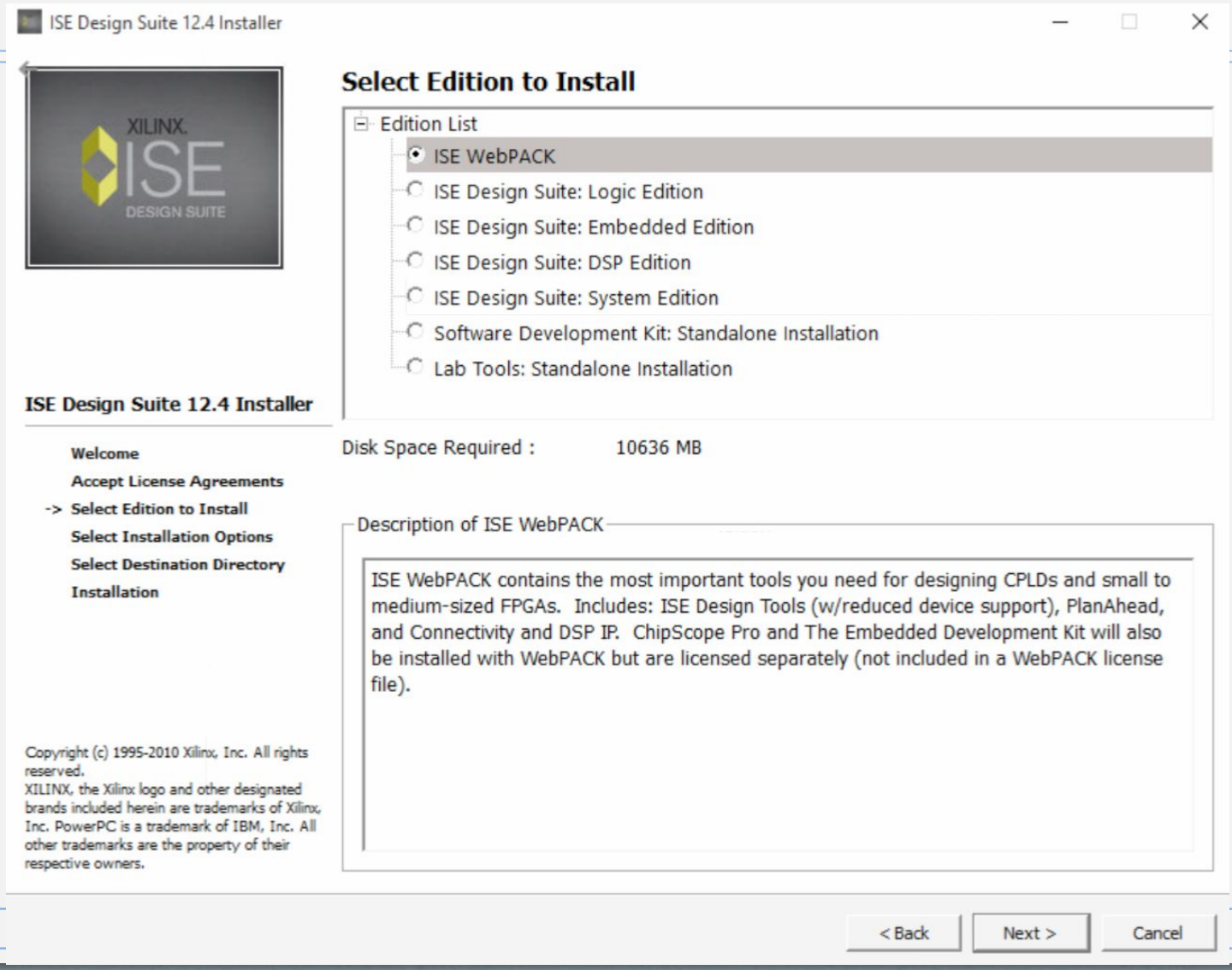
Test and Measurement

Next

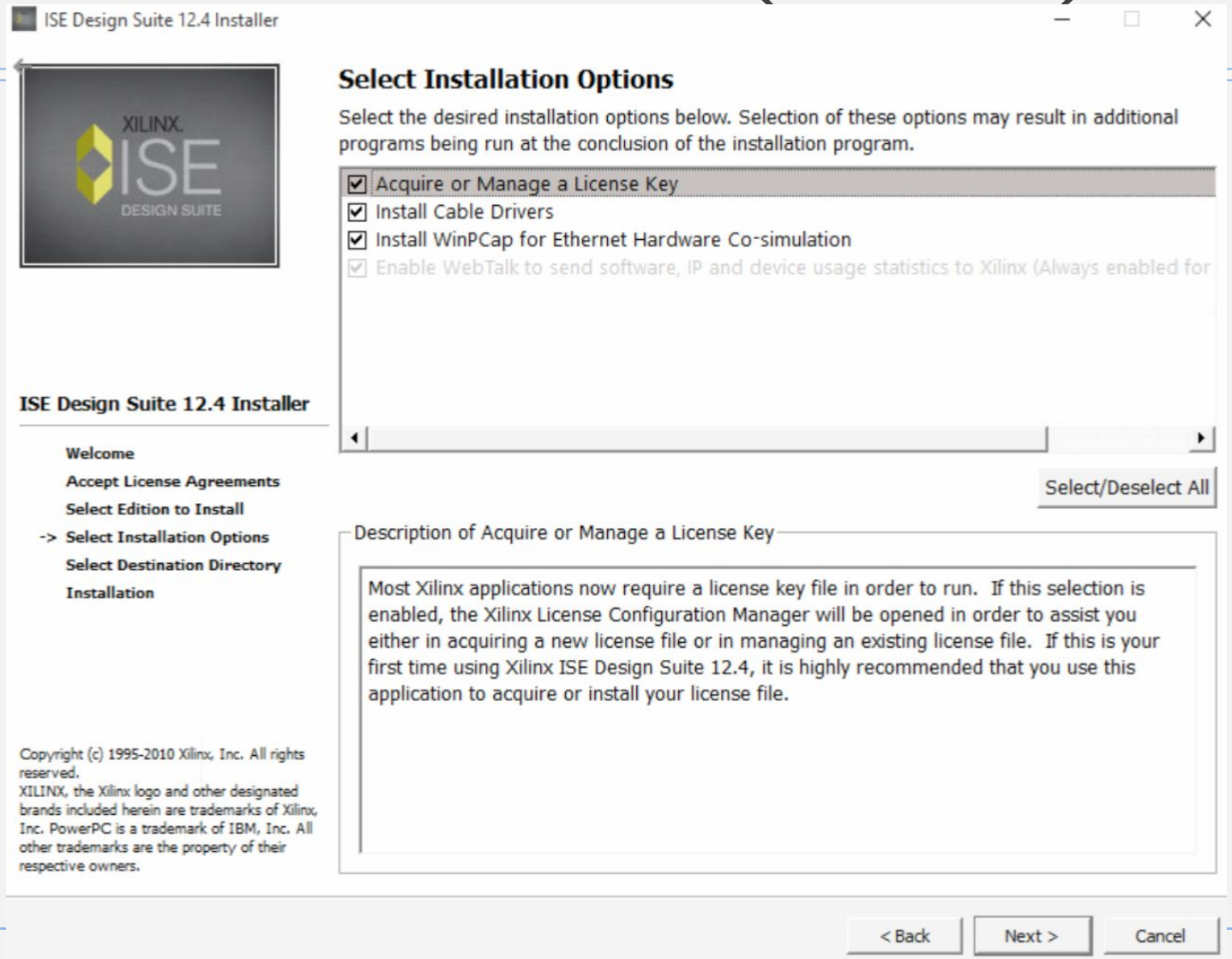
Installation vs. Operating System

- Win8.1 and Win10 users:
  - Run bin/nt/xsetup.exe (even if your OS is 64bit)
- Other Windows versions:
  - Run xsetup.exe inside the downloaded file.
- Linux:
  - Run ./xsetup inside the downloaded file.

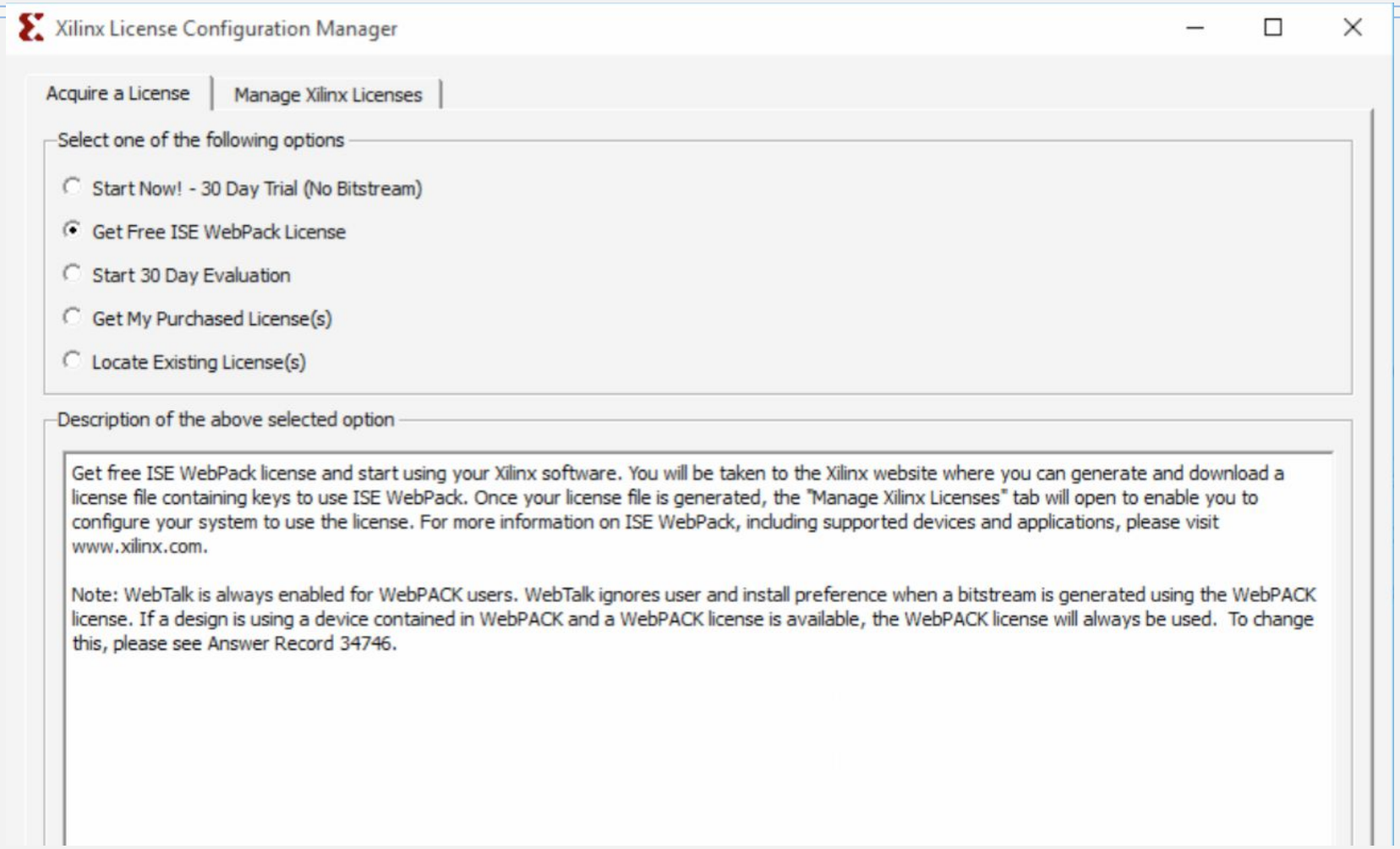
Installation (Cont.)



Installation (Cont.)



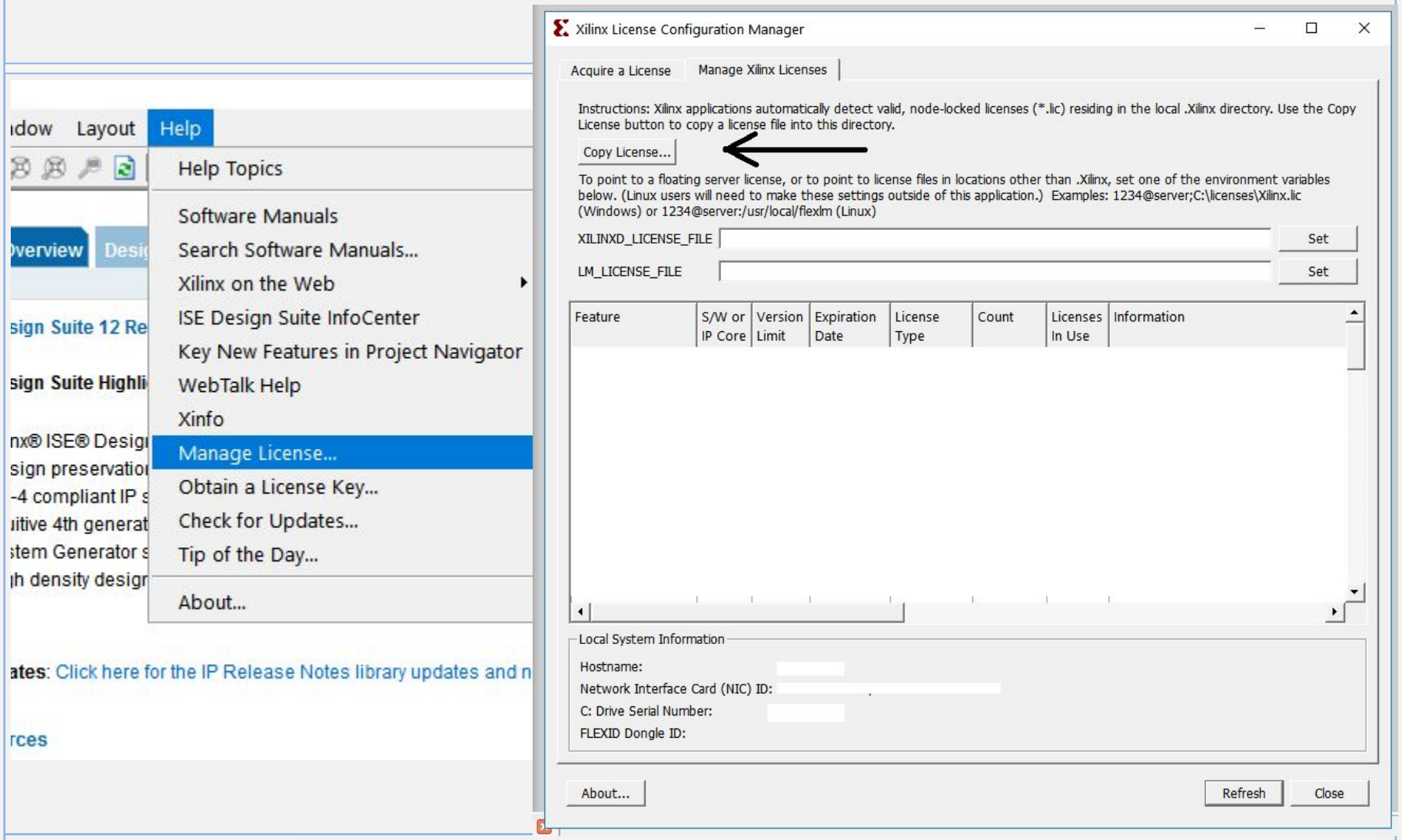
Installation (Licensing)



Licensing (cont.)

- If the previous step does not launch a webpage:
  - Go to http://xilinx.com/getlicense to manually create an “ISE WebPACK License” and download the created Xilinx.lic license file.
  - Launch “Xilinx ISE Design Suite 12.4” and follow the instructions provided in the images on the next slide to import the license file.

Licensing (cont.)



Opening Xilinx ISE

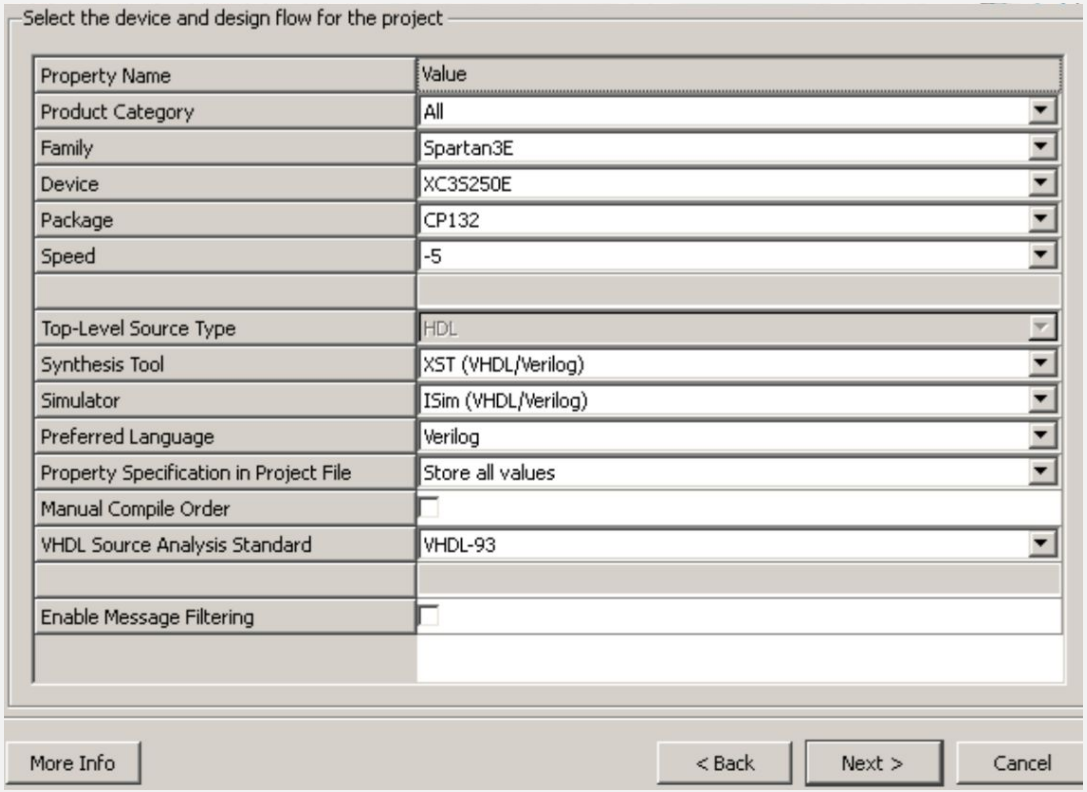
- **Windows**
  - Just run “Xilinx ISE Design Suite 12.4”
  - If you get “ERROR: Simulator 861 – Failed to link the design”:
    - Remove or rename “C:\ Xilinx \ 12.4 \ ISE_DS \ ISE \ gnu \ MinGW \ 5.0.0 \ nt \ libexec \ gcc \ mingw32 \ 3.4.2 \ collect2.exe”
- **Linux**
  - Type “source <Xilinx_folder>/ISE_DS/settings64.sh” (or settings32.sh if you are on 32-bit system) every time you open a terminal.
  - For default installation (at least on Ubuntu 20.04) Xilinx installation folder is “/opt/Xilinx/12.4/”
  - So, above command for default installation will be “source /opt/Xilinx/12.4/ISE_DS/settings64.sh”

Opening Xilinx ISE (cont.)

- **Linux**
  - If you do not want to type “source ..../settings64.sh” every time you open a terminal, append that command to your home folder “.bashrc” file (re-open your terminal after that)
  - After that type “ise” on your terminal.

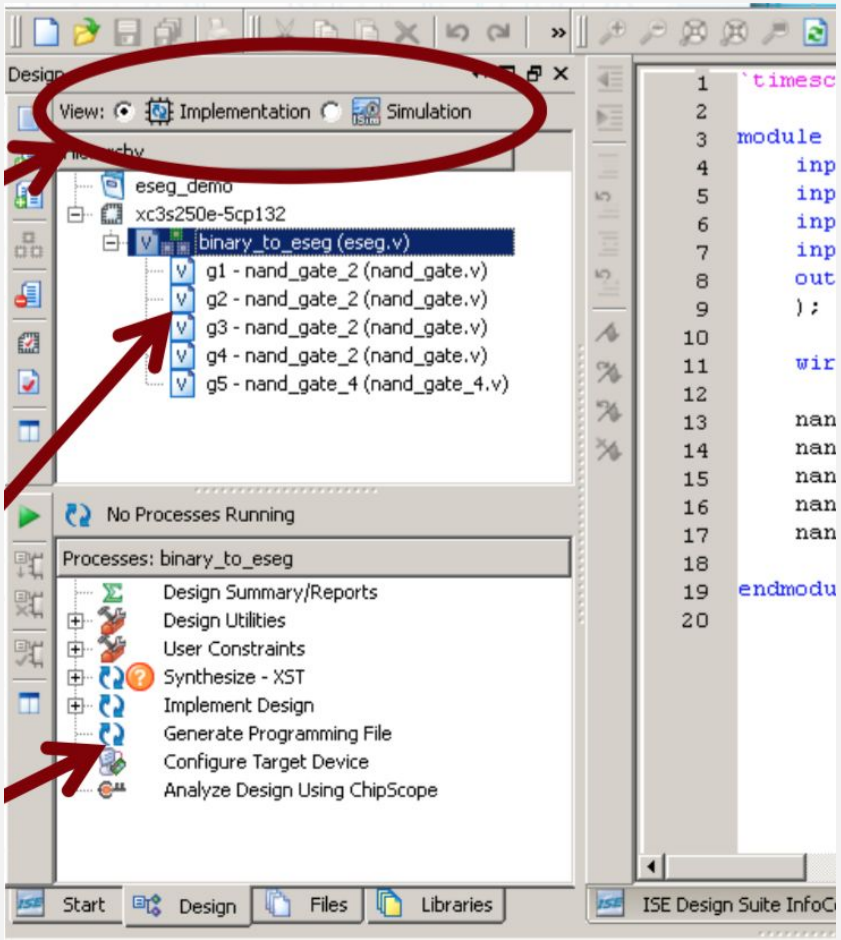
Creating a Project

- Open ISE Design Tools → Project Navigator
- Create new project with the following options



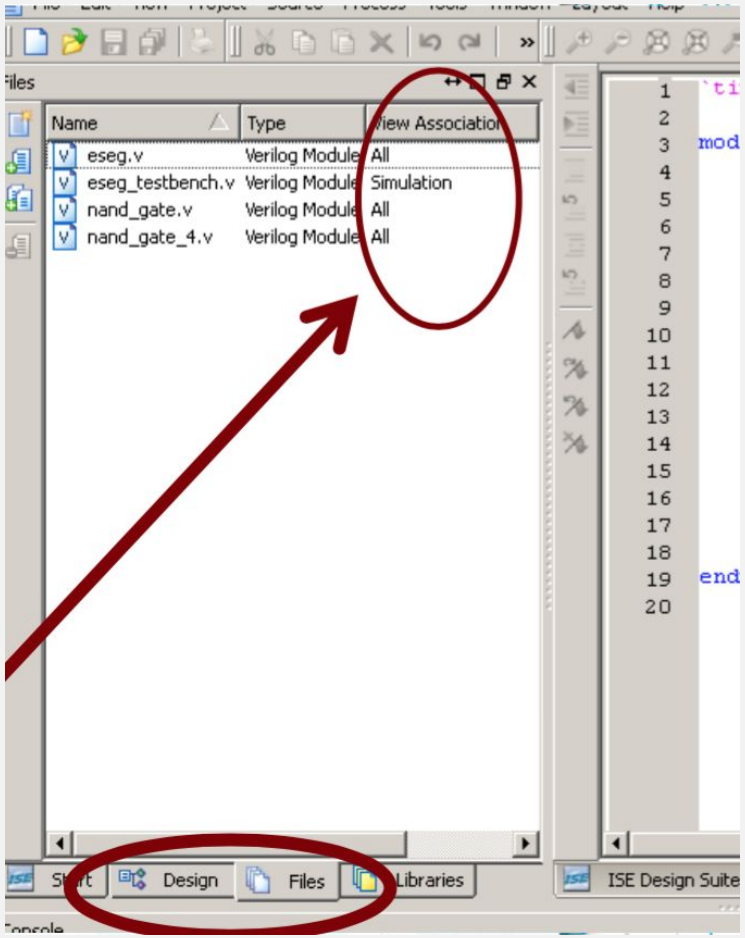
Basic Properties of Xilinx

- 2 Design modes:
  - Implementation
  - Simulation
- Top
  - Module hierarchy
- Bottom
  - Actions with
  - Selected module



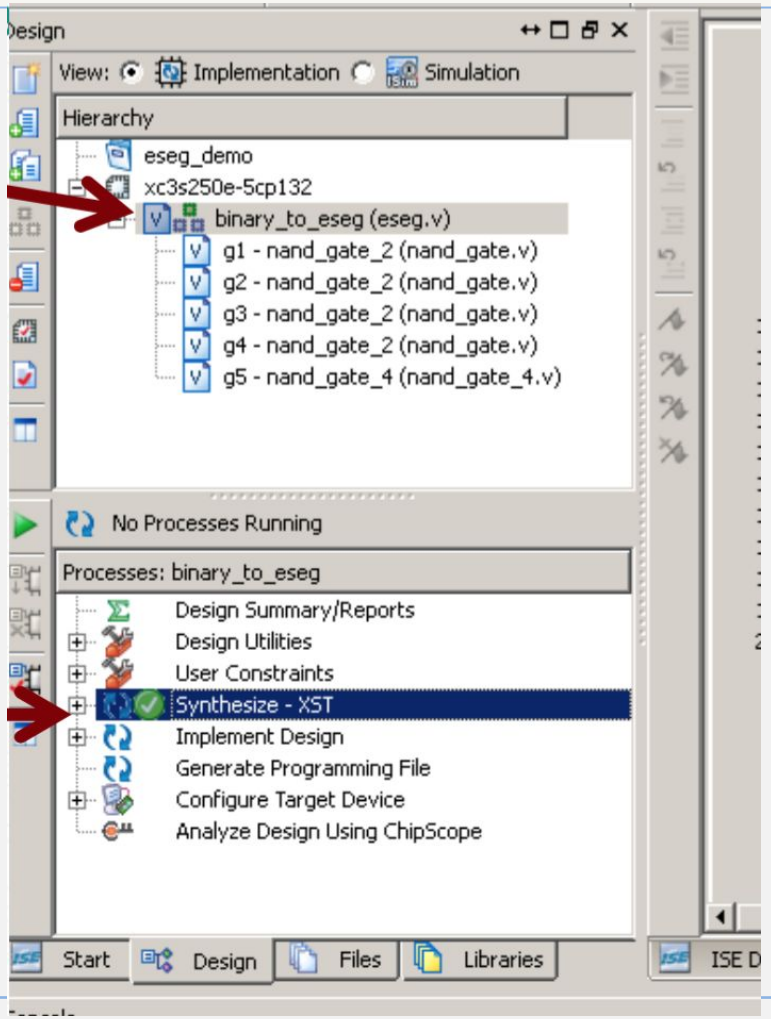
Basic Properties of Xilinx

- You can switch between Design and Files views
- A file may belong to:
  - Implementation only
    - files related with FPGA board
  - Simulation only
    - Testbenches …
  - Both
    - Your design



Compiling, checking syntax errors

- Choose desired module
  - All dependencies will also be compiled
  - Compiler output will be shown in the bottom
- In implementation mode
  - Double click on Synthesize
- OR in simulation mode
  - Double click on Behavioral Check Syntax

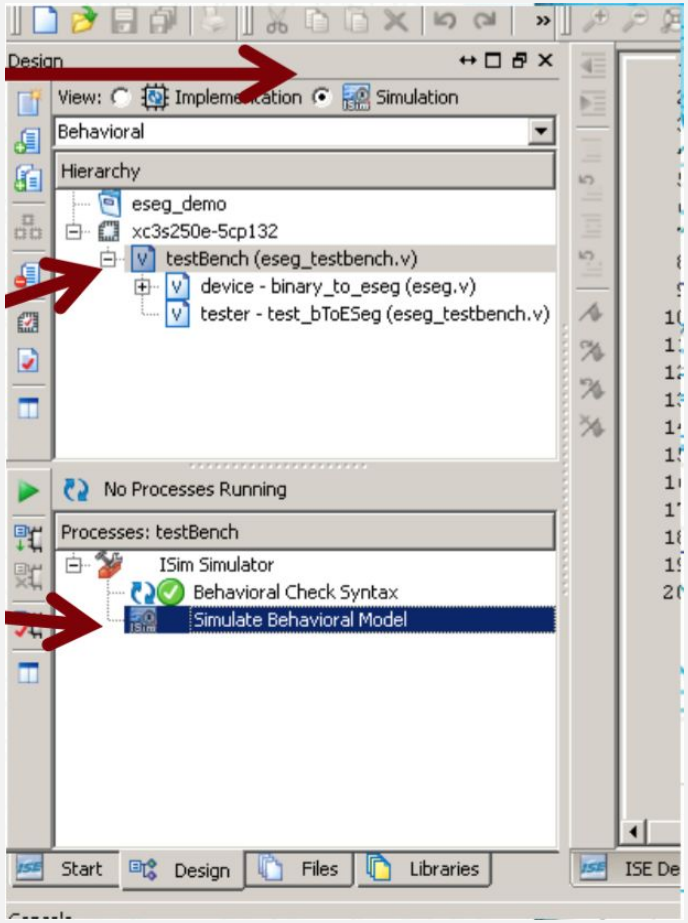


Simulation

- Special commands can be used inside simulation source-files:
  - $finish, $stop, $monitor, $display, ...
- Results of these are seen on the output panel.
- Also, wire values of any module can be seen in the timing diagram panel.

Simulation (Cont.)

- In simulation mode
- Select testbench module
- Double click on Simulate Behavioral Model
  - Dependencies will be compiled
  - ISim Simulator will be opened



Isim Simulator