

# CENG 232

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

# Hardware Description Language

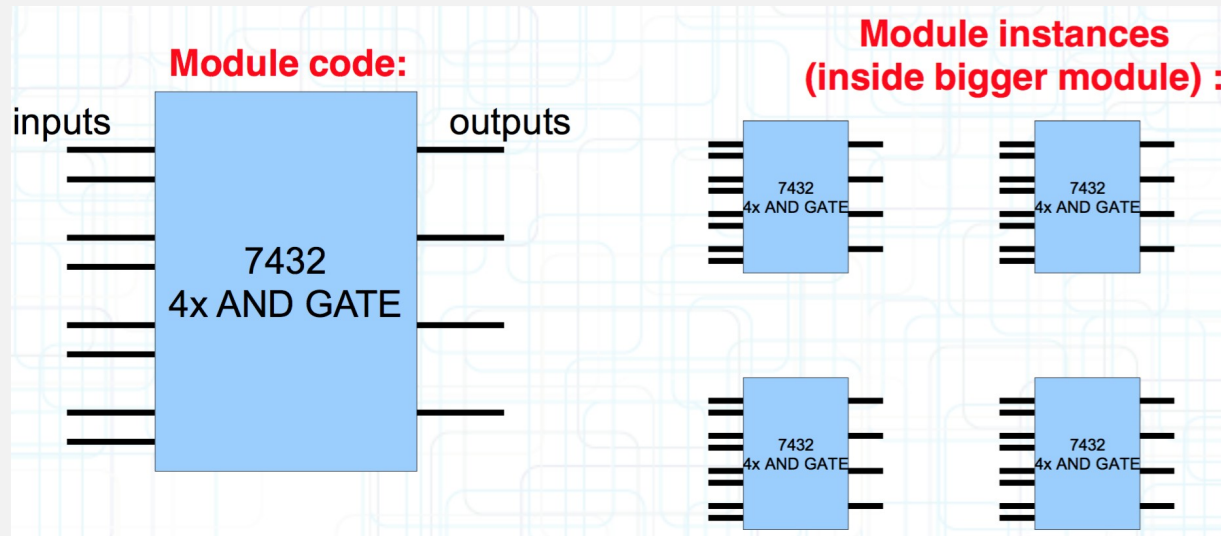
- Designing schematics wire by wire is:
  - Hard
  - Error-prone
  - Gets more complex as project gets bigger
- HDL
  - Describe your circuits as textual “code”, just like any programming language
  - Divide larger projects into smaller **modules**
  - Harder to visualize, but easier to manage

# 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



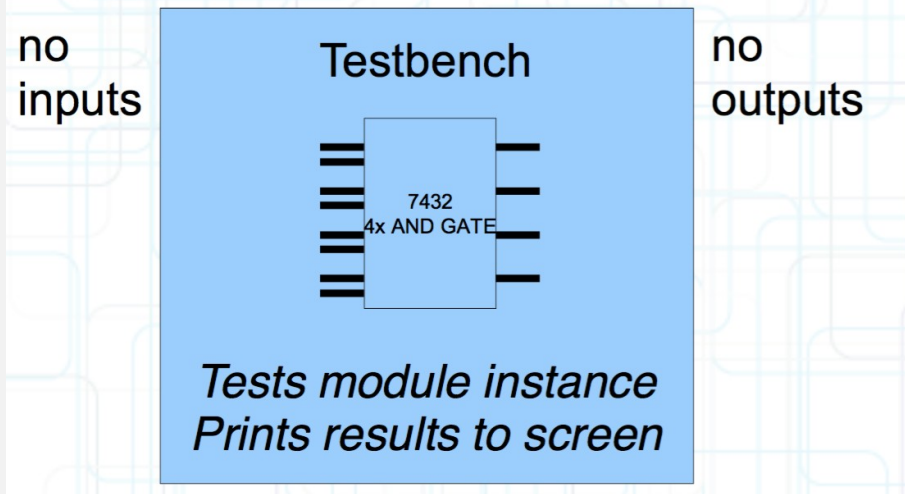
# Testbench Idea

- 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 module:



# 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

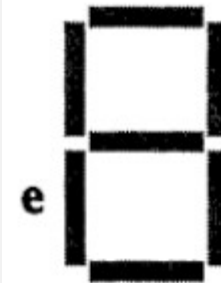
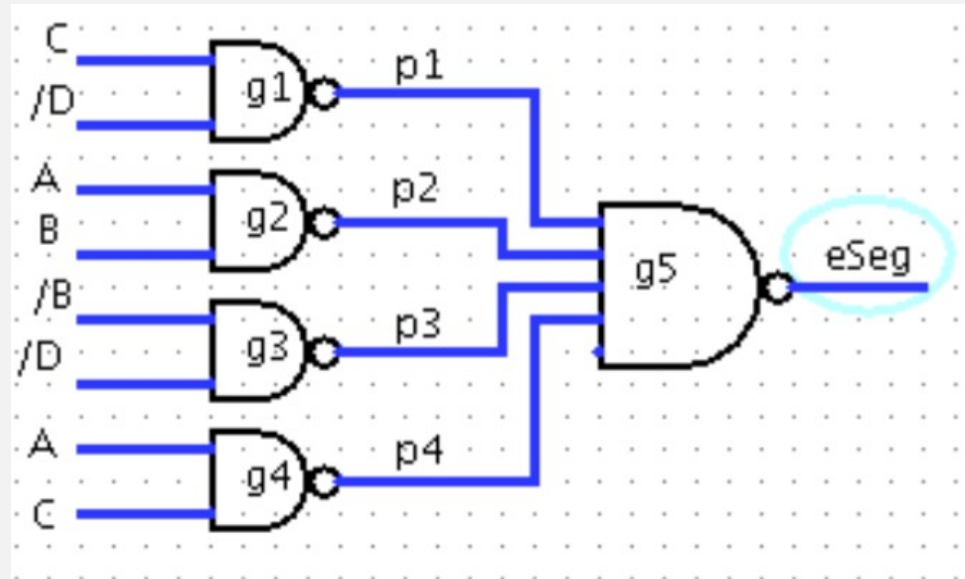
TEST2 failed

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>
- Lab Computers (Ineks) will have Xilinx v12.4, you can install 14.7 on your computers as well.
- 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](mailto: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.

# Xilinx Installation

## Support

- **Windows 10**
  - We've had tested the installation. It works but you may have get an issue when compiling your verilog files. Please check the upcoming slides.
- **Windows 11**
  - We haven't tested 12.4 or 14.7.
  - It may work on your personal machine, you may try it.
  - You can always use the Ineks.
- **Linux**
  - We've tested the installation on Ubuntu 22.04(latest version) without any issues.
  - Please follow the instructions on the upcoming slides.
- **Mac**
  - We could not be able to test it due to lack of hardware.

# Installation

← → ↻ xilinx.com/support/download.html ☆ 0

Solutions Products Support



Home / Support / Downloads

## Downloads

Licensing Help

Alveo Accelerator Card ▾

Vivado (HW Developer)

Vitis (SW Developer)

Vitis Embedded Platforms

Alveo Packages

PetaLinux

Device Models

Documentation Navigator

## Version

2020.3

2020.2

2020.1

2019.2

Vivado Archive

ISE Archive

CAE Vendor Libraries Archive

### Vivado Design Suite - HLx Editions - 2020.3 Full Product Installation

#### Important Information

Download Vivado® Design Suite 2020.3 now, with support for

- Production Devices
  - Versal™ AI Core Series :- XCVC1902, XCVC1802
  - Versal™ Prime Series :- XCVM1802

We **strongly recommend** to use the web installers as it reduces download time and saves significant disk space.

Please see [Installer Information](#) for details.

**IMPORTANT NOTE:** This release supports **Versal devices only!**

For customers using Versal devices, Xilinx recommends installing Vivado 2020.3.

Download Includes

Download Type

Last Updated

Answers

Documentation

Support Forums

Vivado Design Suite HLx Editions (All Editions)

Full Product Installation

Apr 17, 2021

[2020.x - Vivado Known Issues](#)

[Release Notes](#)

[OS Support Update](#)

[What's New in Vivado](#)

[Installation and Licensing](#)

# Installation (Cont.)

2021.2	14x
2021.1	14.7 Windows 10
2020.3	14.7
Vivado Archive	
ISE Archive	
CAE Vendor Libraries Archive	14.6
	14.5
	14.4
	14.3
	14.2
	14.1
	13.x
	13.4
	13.3
	13.2
	13.1
	12.x
	12.4

# Installation (Cont.)

12.x

12.4

## ISE Design Suite - 12.4 Full Product Installation

↓ [All Platforms \( - 4.39 GB\)](#)  
MD5 SUM Value: eb743b99096e39a8996d8ee8e673b486

↓ [Full Installer for Windows \( - 3.36 GB\)](#)  
MD5 SUM Value: 9aab55db13d0b5aaa6b375856421ec20

↓ [Full Installer for Linux \( - 3.46 GB\)](#)  
MD5 SUM Value: 33b9326a3eff75f289d681a8a9a091d4

Download Type

Last Updated

Enablement

Order DVD

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 ▼

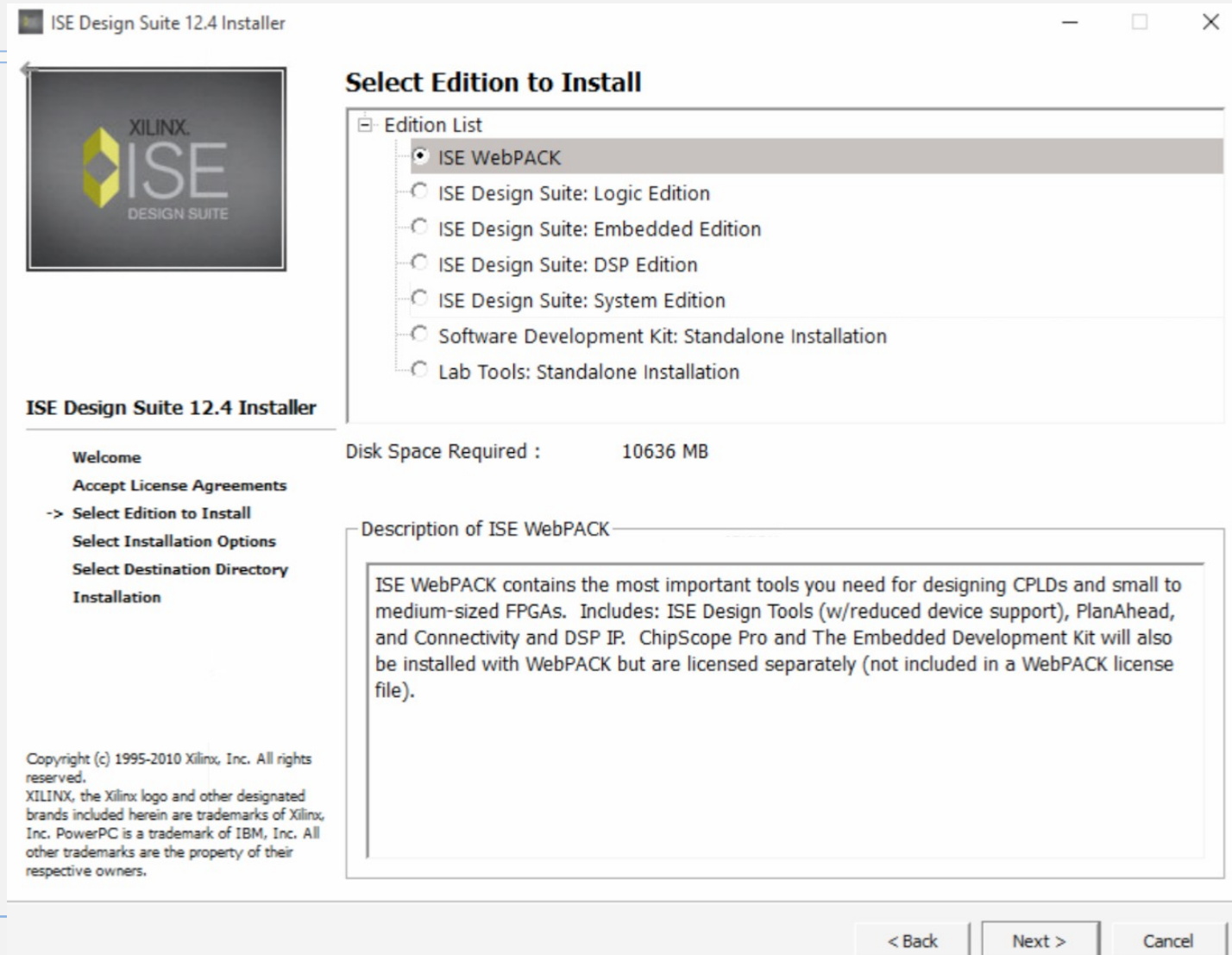
Next

# Installation vs. Operating System

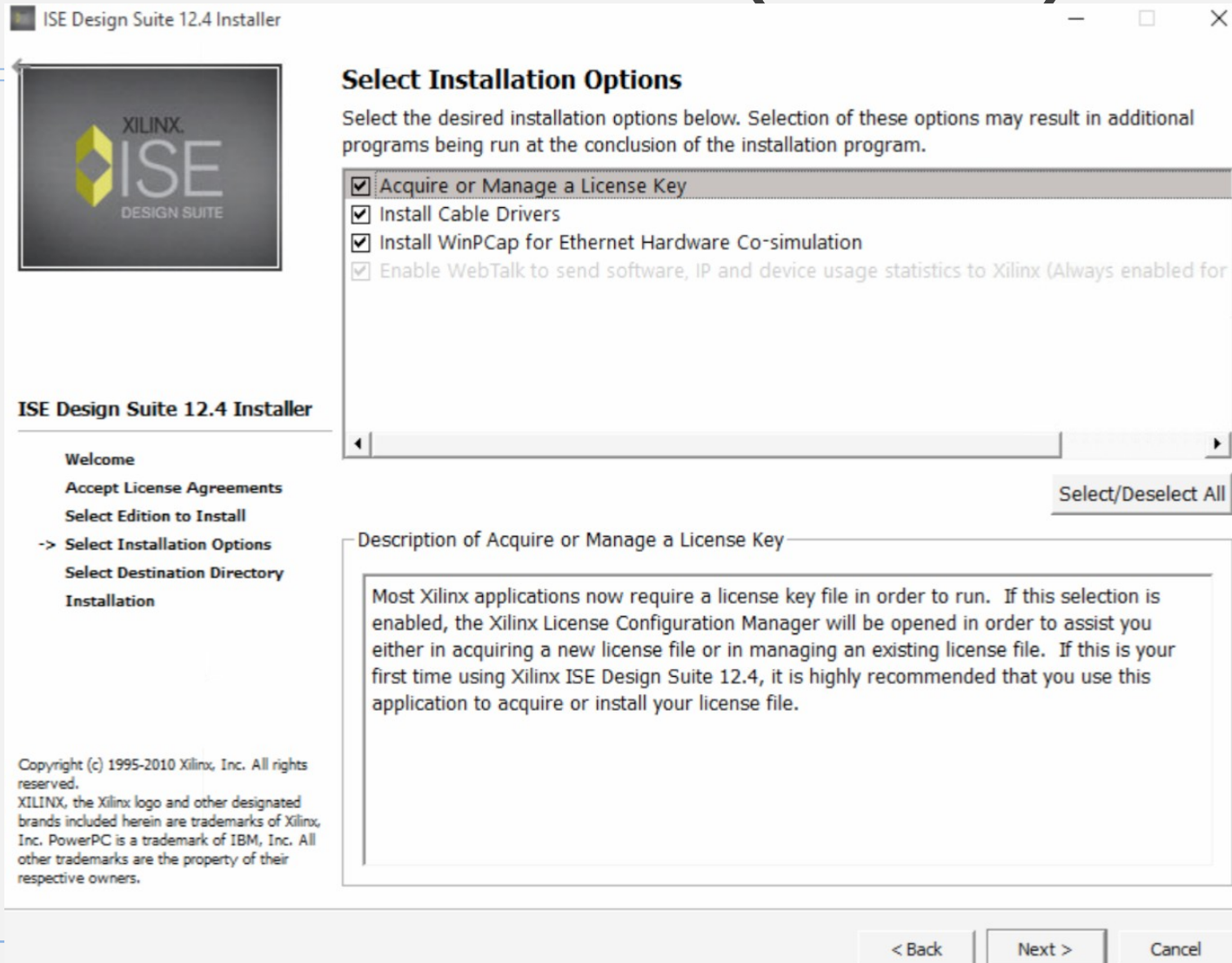
- Win8.1 and Win10 users:
  - Run bin/nt/xsetup.exe (even if your OS is 64bit)
- Other Windows versions:
  - Run xsetup.exe inside the downloaded file.
- Linux:
  - Run ./xsetup inside the downloaded file.



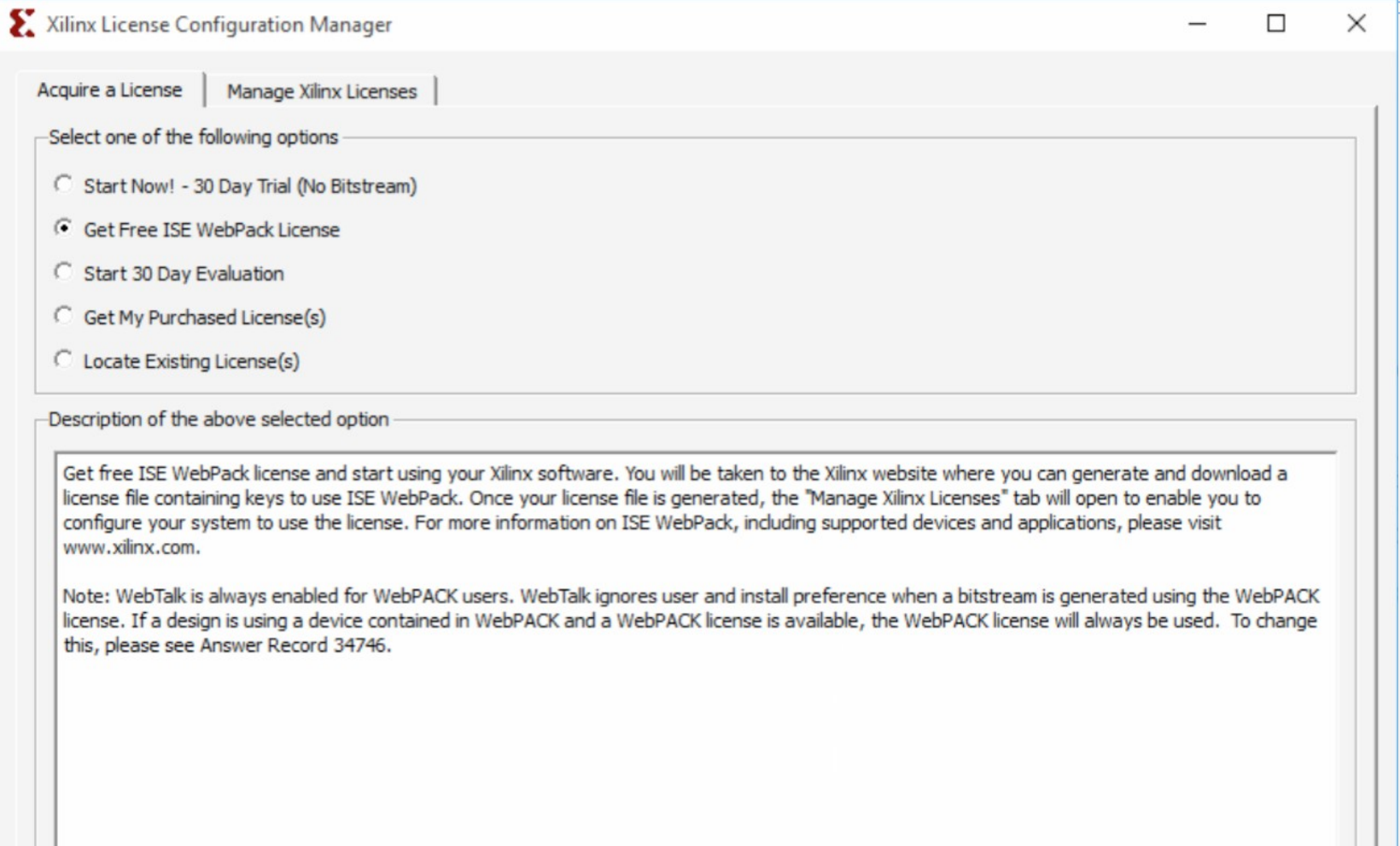
# Installation (Cont.)



# Installation (Cont.)



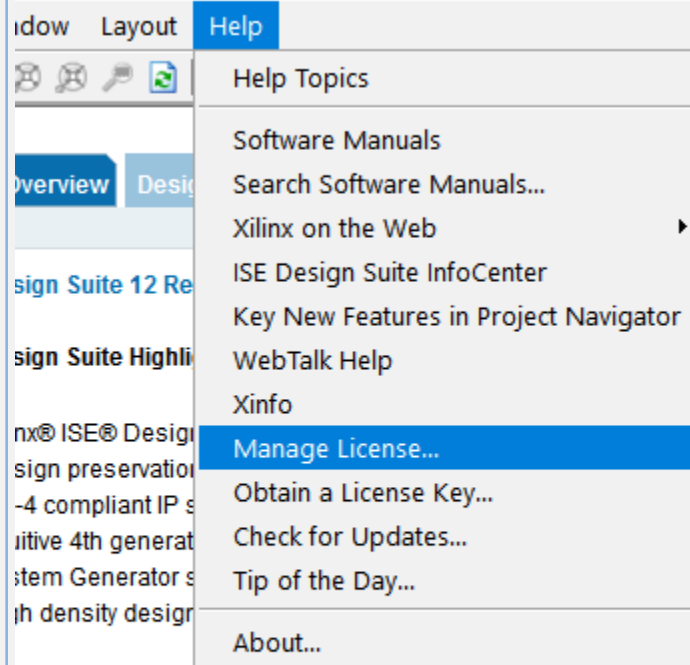
# Installation (Licensing)



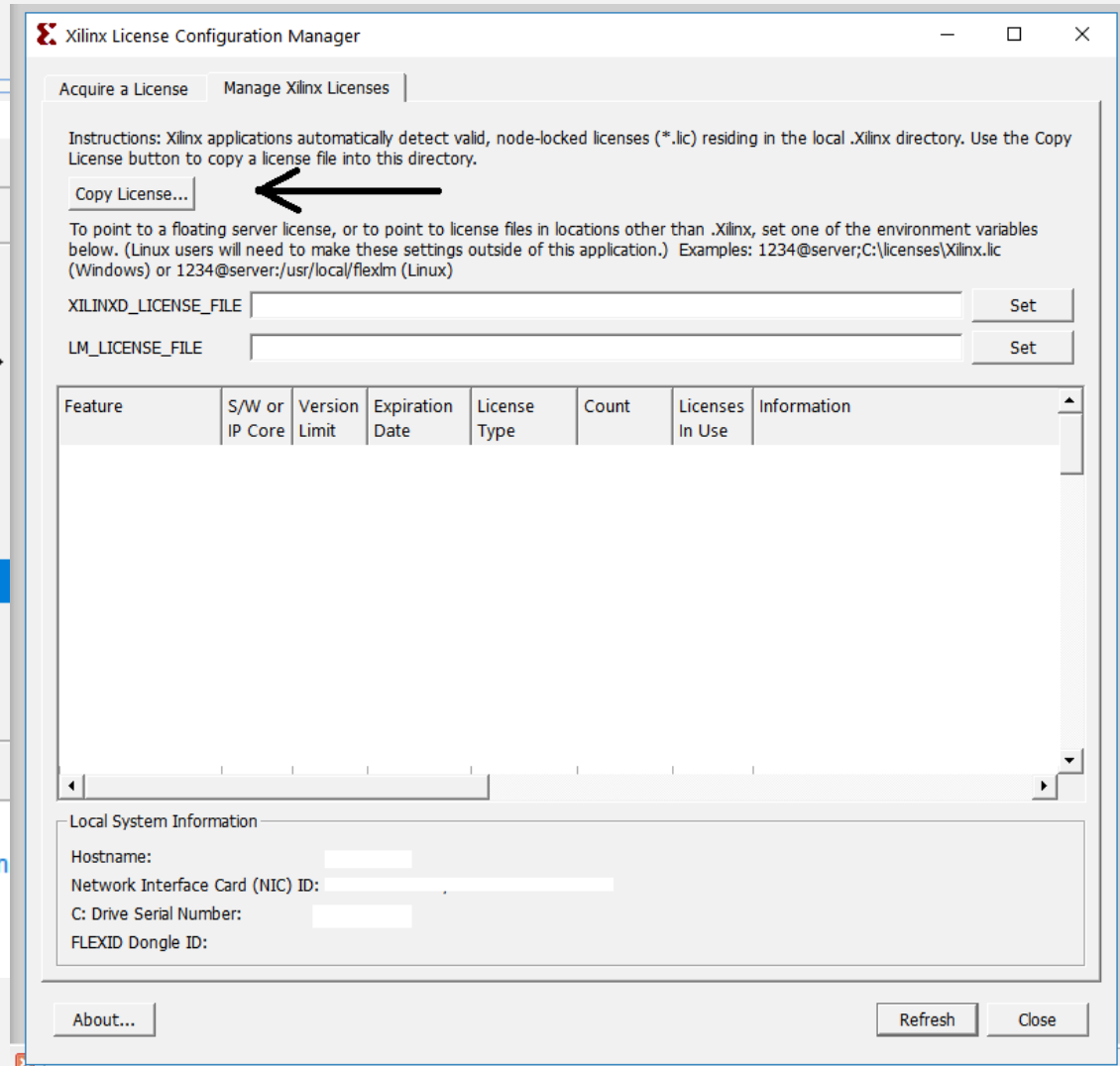
# Licensing (cont.)

- 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.)



ates: [Click here for the IP Release Notes library updates and n](#)  
rces



# 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

Select the device and design flow for the project

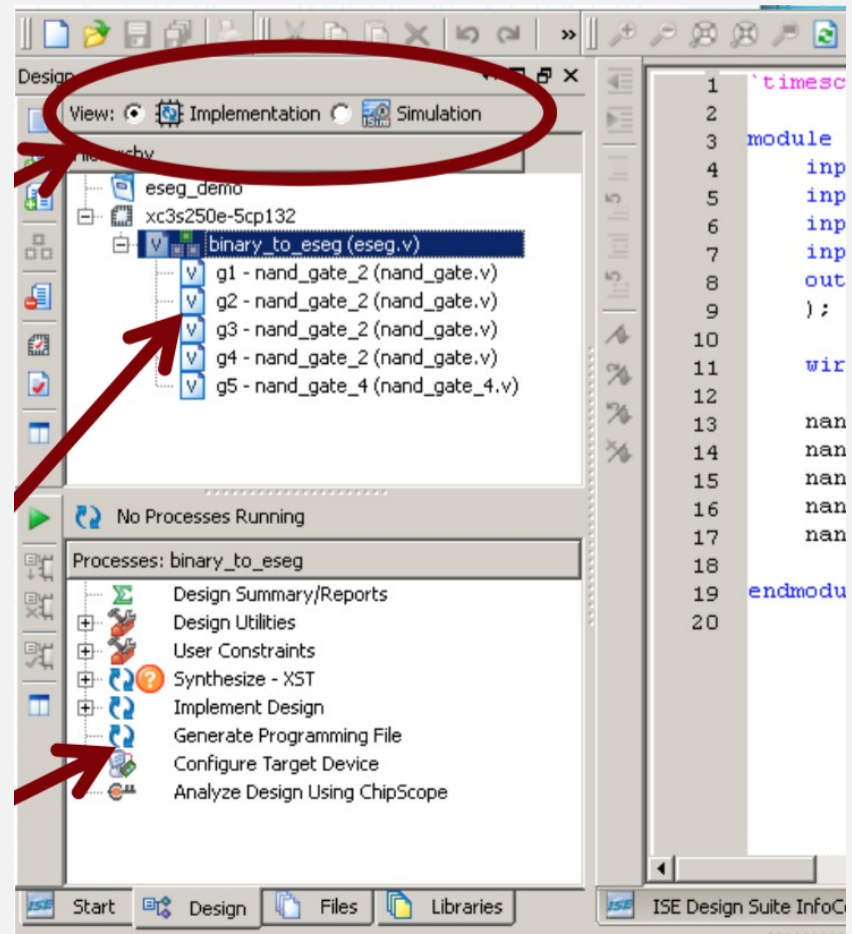
Property Name	Value
Product Category	All
Family	Spartan3E
Device	XC3S250E
Package	CP132
Speed	-5
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

More Info      < Back      Next >      Cancel



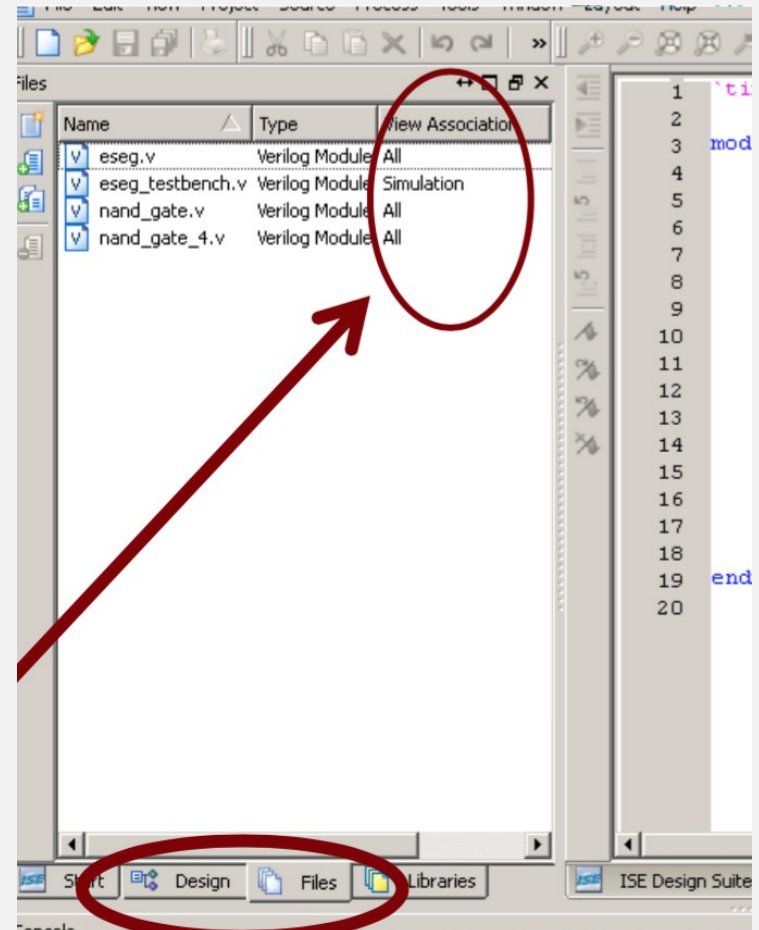
# Basic Properties of Xilinx

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



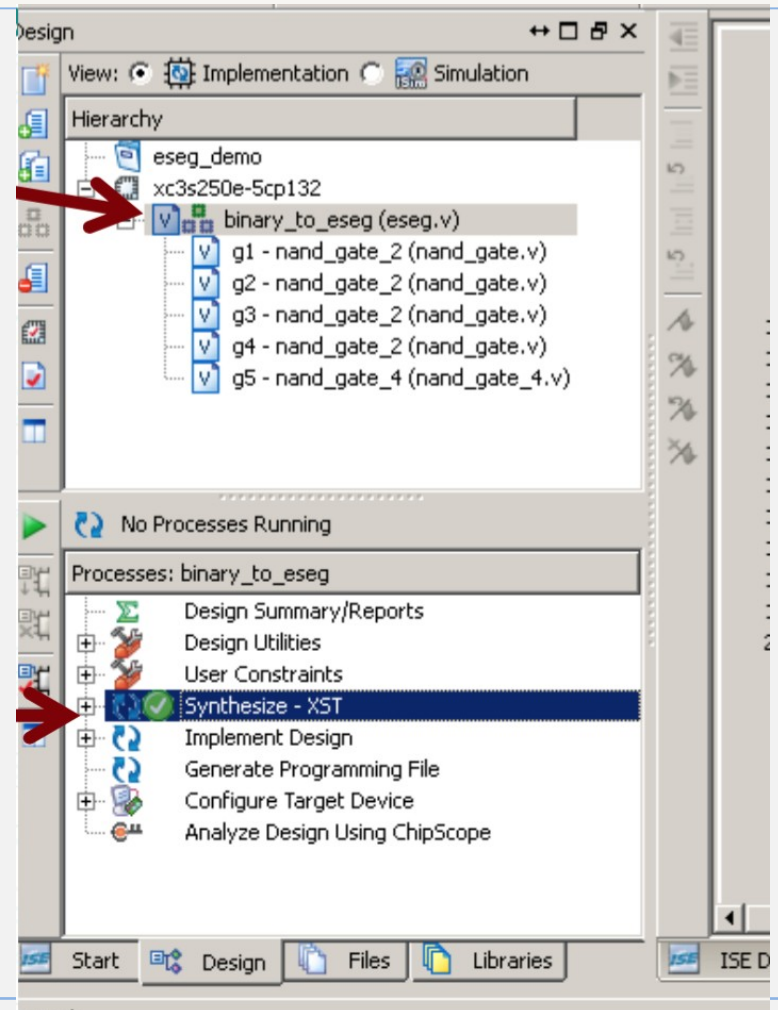
# Basic Properties of Xilinx

- 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**

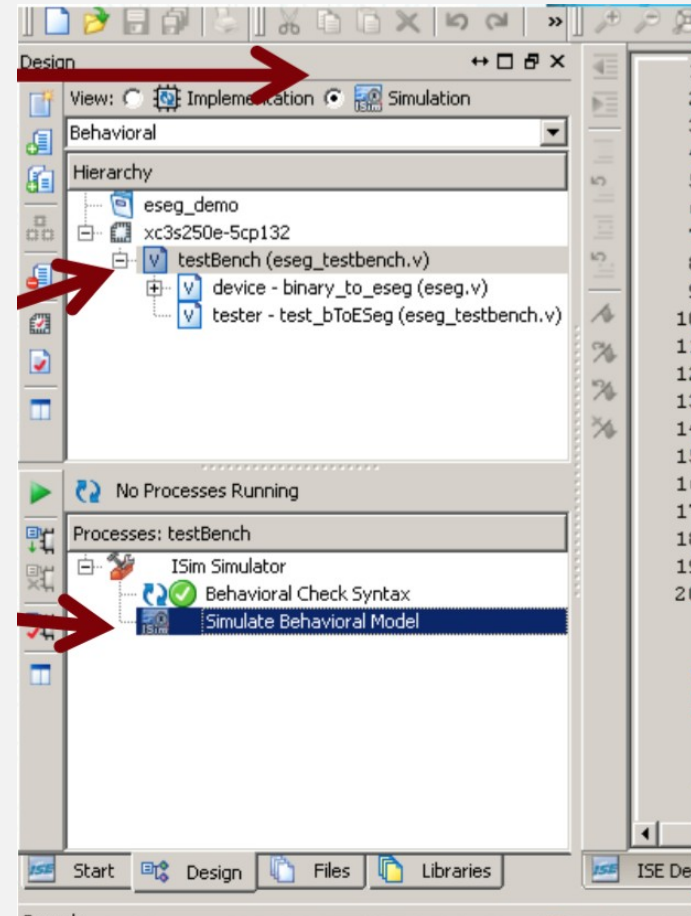


# Simulation

- 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

**Controls Timing Diagram scale**

**Timing Diagram**

**Timing Diagram Vs Source Code view**

**Simulator Output**

The screenshot displays the Isim Simulator interface. The top menu bar includes File, Edit, View, Simulation, Window, Layout, and Help. Below the menu is a toolbar with various icons. A red circle highlights the 'Timing Diagram' icon in the toolbar, with an arrow pointing to the text 'Controls Timing Diagram scale'. The main window is divided into three panes. The left pane shows a list of variables: w1, w2, w3, w4, and w5, with their current values (0, 0, 0, 0, and 1 respectively). The middle pane displays the 'Timing Diagram' for these variables, showing their values over time. A red circle highlights the 'Timing Diagram' icon in the toolbar, with an arrow pointing to the text 'Timing Diagram Vs Source Code view'. The right pane shows the 'Source Code view' of the testbench file, 'eseg\_testbench.v'. The bottom pane is the 'Console', which displays the simulator's output, including initialization messages and the final simulation time (40 ns). A red circle highlights the 'Console' icon in the toolbar, with an arrow pointing to the text 'Simulator Output'.

Name	Value
w1	0
w2	0
w3	0
w4	0
w5	1

Timing Diagram: X1: 13.943 ns

Simulator Output:

```
This is a Lite version of Isim.  
Time resolution is 1 ps  
Simulator is doing circuit initialization process.  
Finished circuit initialization process.  
0 A=x B=x C=x D=x eSeg=x  
Setting all inputs to 0  
10 A=0 B=0 C=0 D=0 eSeg=1  
Setting D to 1  
20 A=0 B=0 C=0 D=1 eSeg=0  
Setting C to 1, D to 0  
30 A=0 B=0 C=1 D=0 eSeg=1  
Ending simulation  
Stopped at time : 40 ns : File "C:/Documents and Settings/Kerem/Belgelerim/ceng232-1st recitation/verilog/eseg_demo/eseg_testbench.v"
```

Sim Time: 40,000 ps