Computer Organization

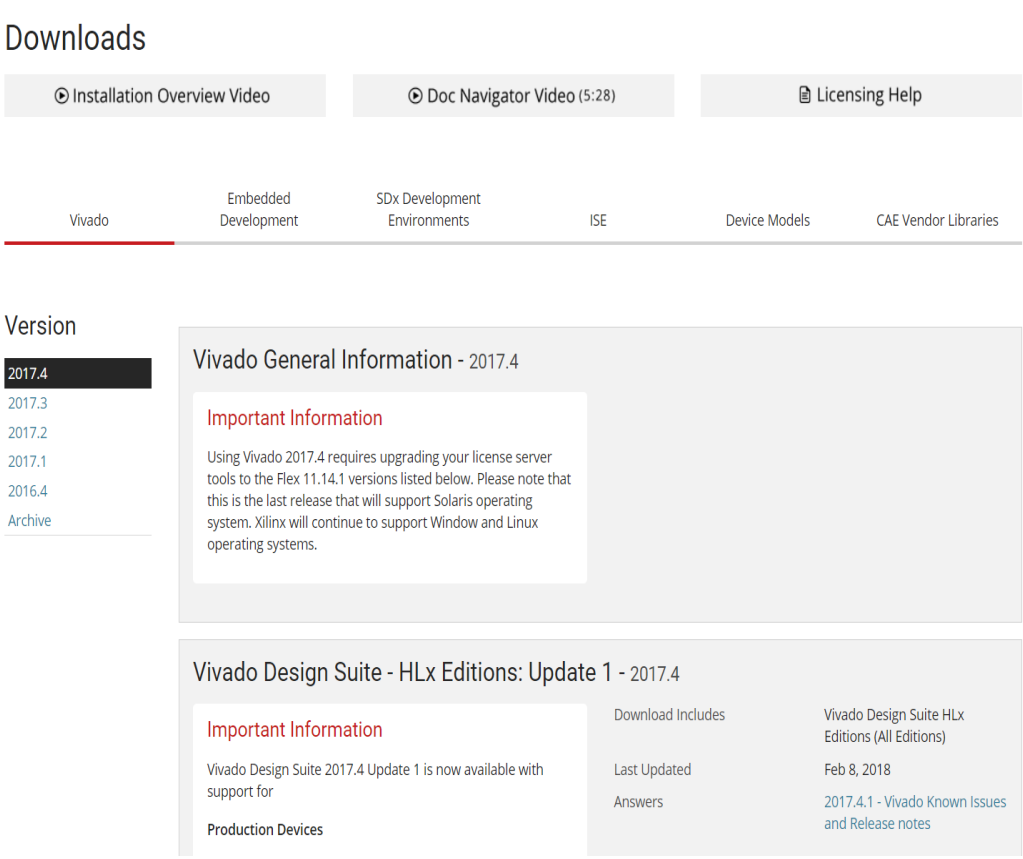
[Introduction to **VIVADO**]

Hunjun Lee (hunjunlee7515@snu.ac.kr), Sunghwa Lee
High Performance Computer Systems (HPCS) Lab.
March 6th, 2018

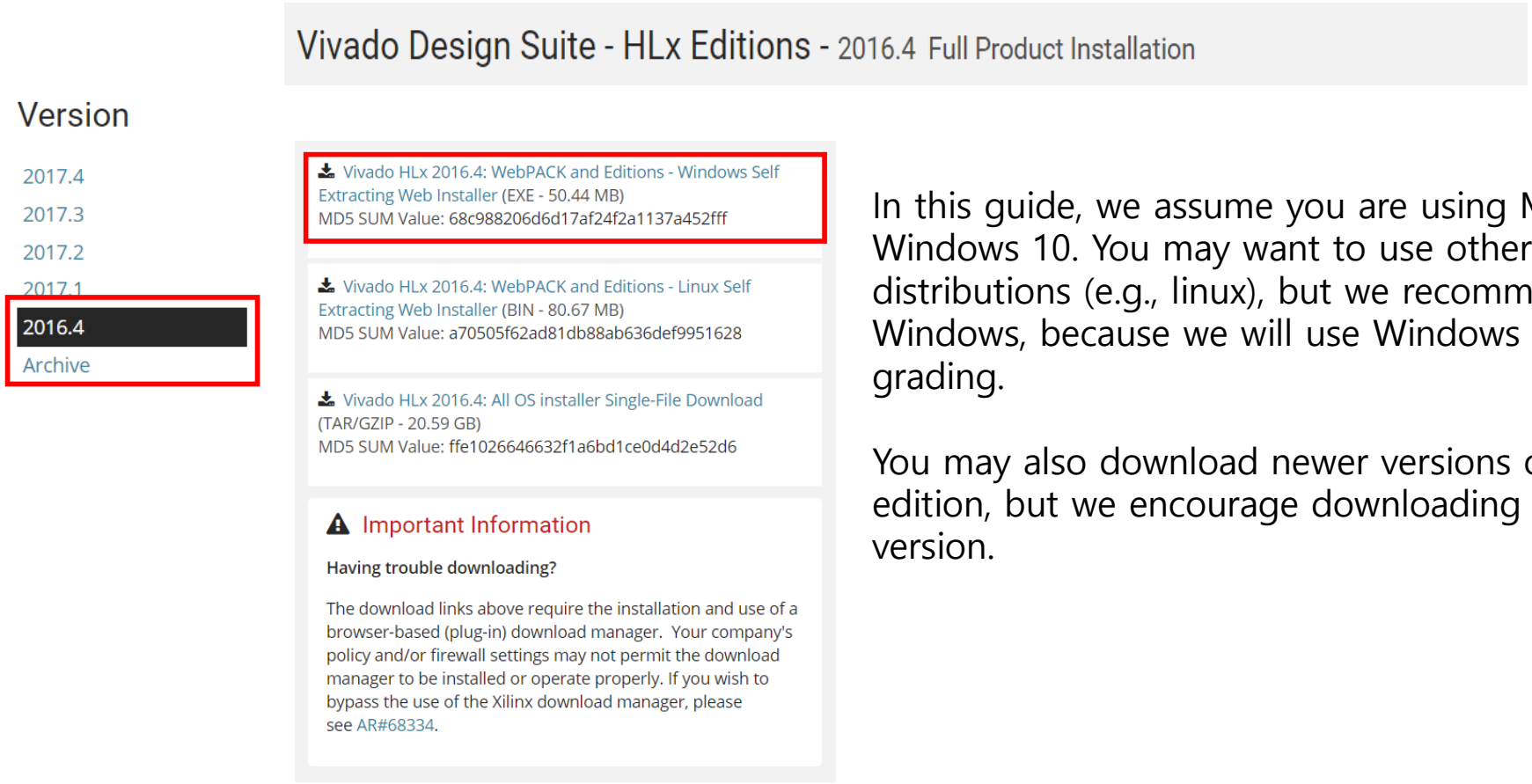
1. Go to Xilinx Website

Use below link:

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



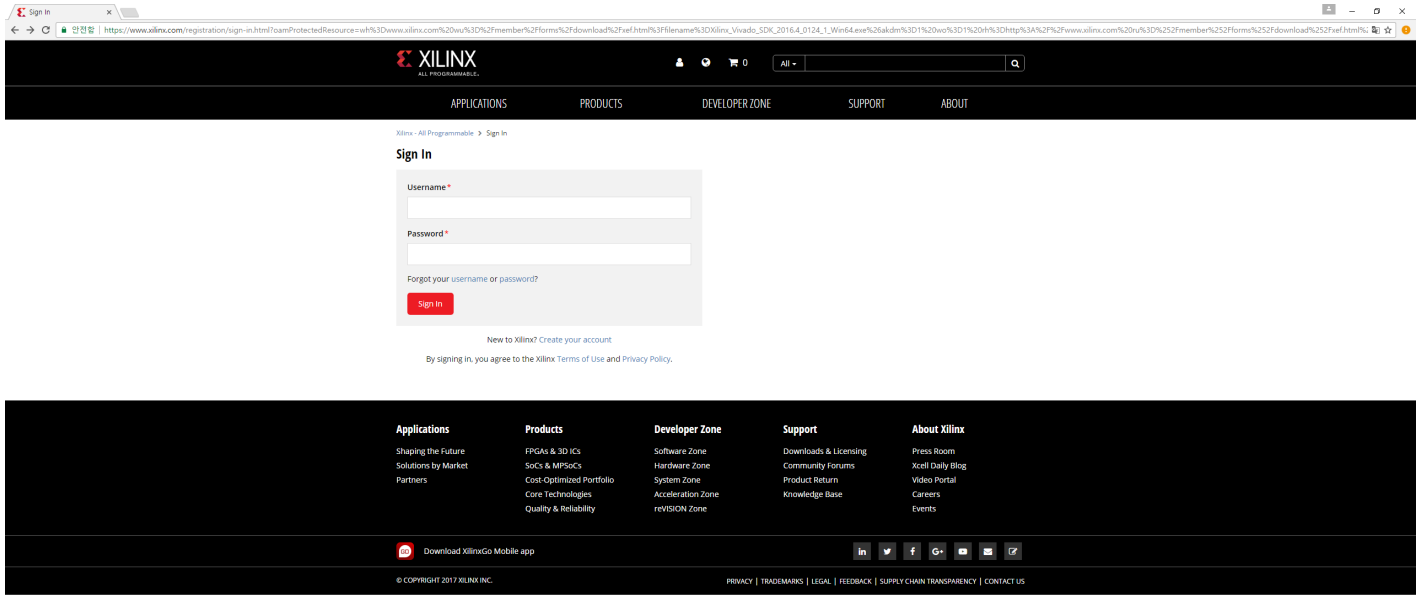
2. Download Vivado HLx 2016.4: **WebPACK** and Editions



In this guide, we assume you are using Microsoft Windows 10. You may want to use other distributions (e.g., linux), but we recommend to use Windows, because we will use Windows 10 for grading.

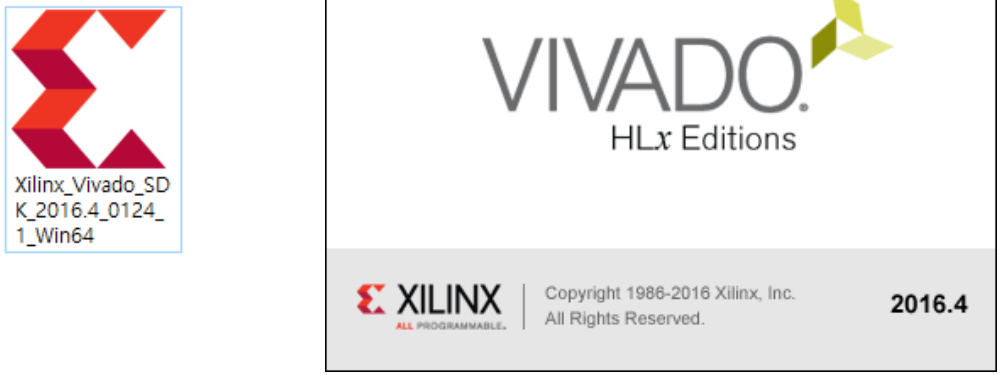
You may also download newer versions of WebPACK edition, but we encourage downloading the given version.

2-1. Sign in the Xilinx account. If you don't have the account, the you should create your account.

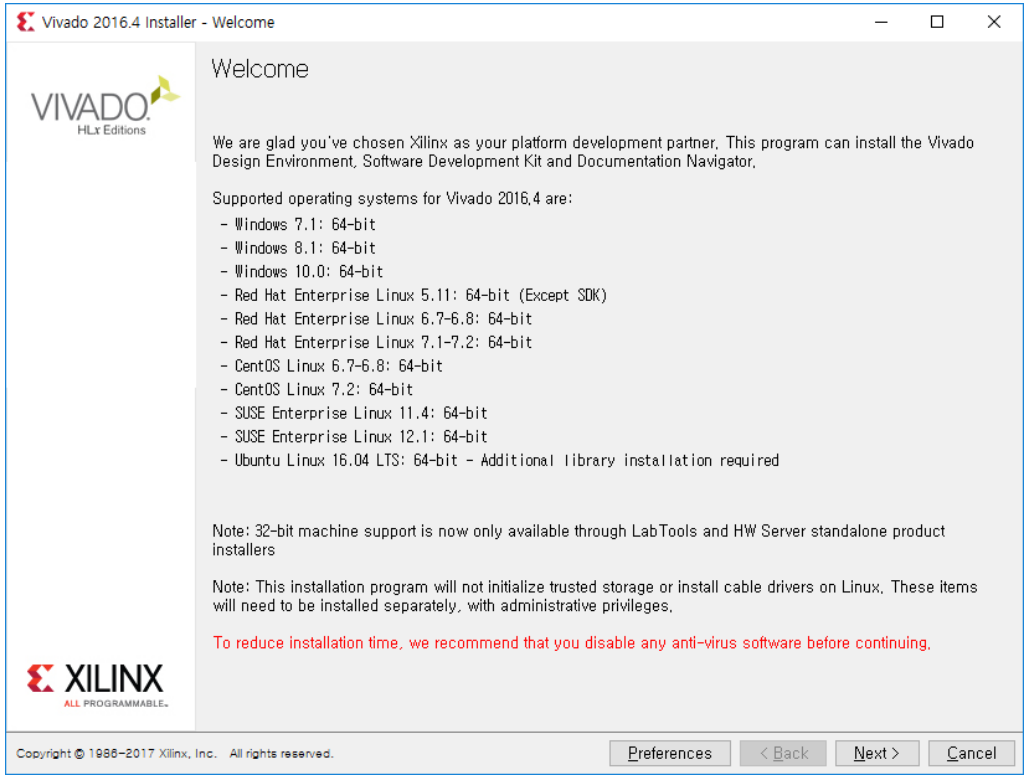


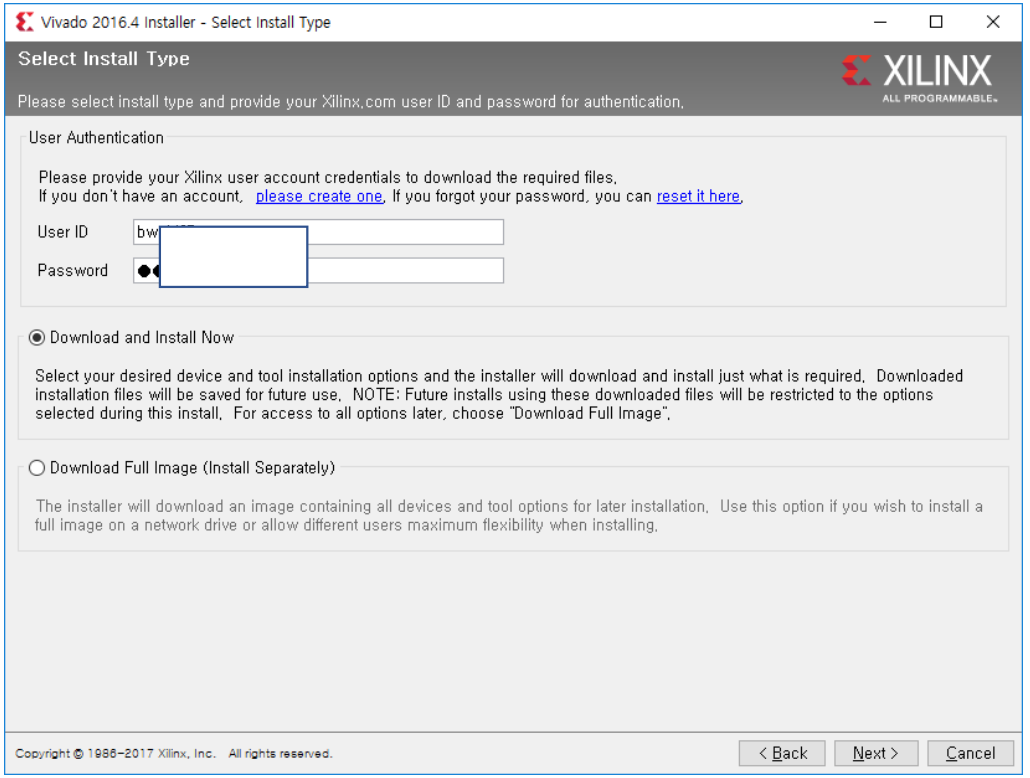
You should create account with SNU email address to get full license

3. Execute Xilinx setup file

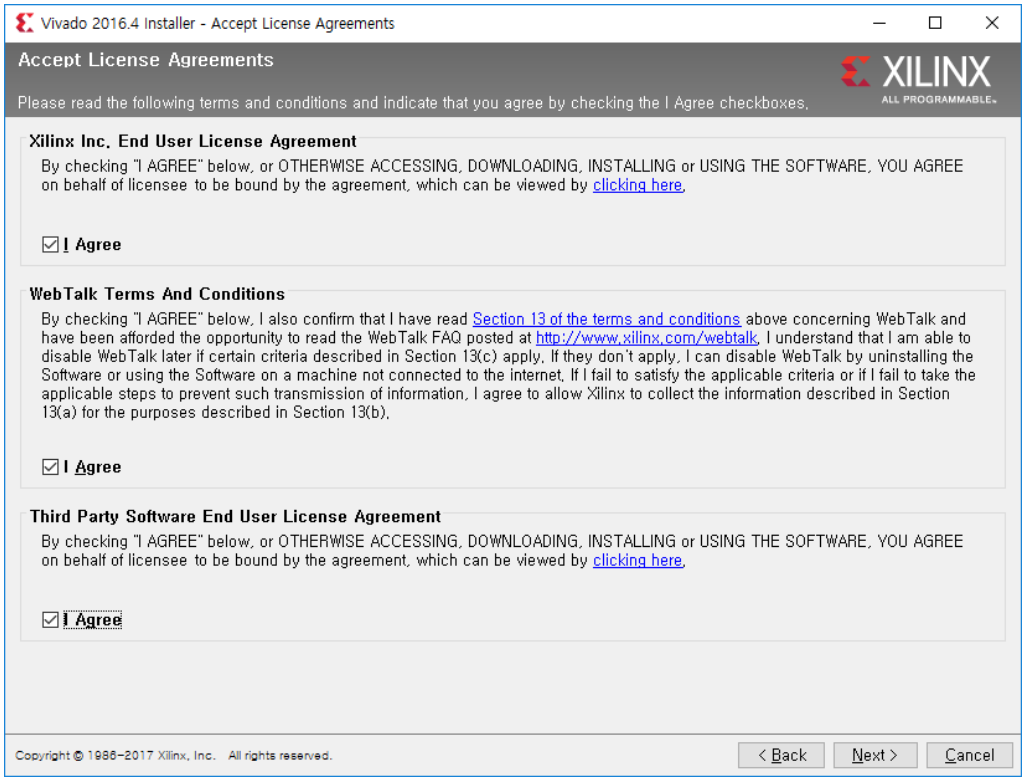


4. 5. Press Next

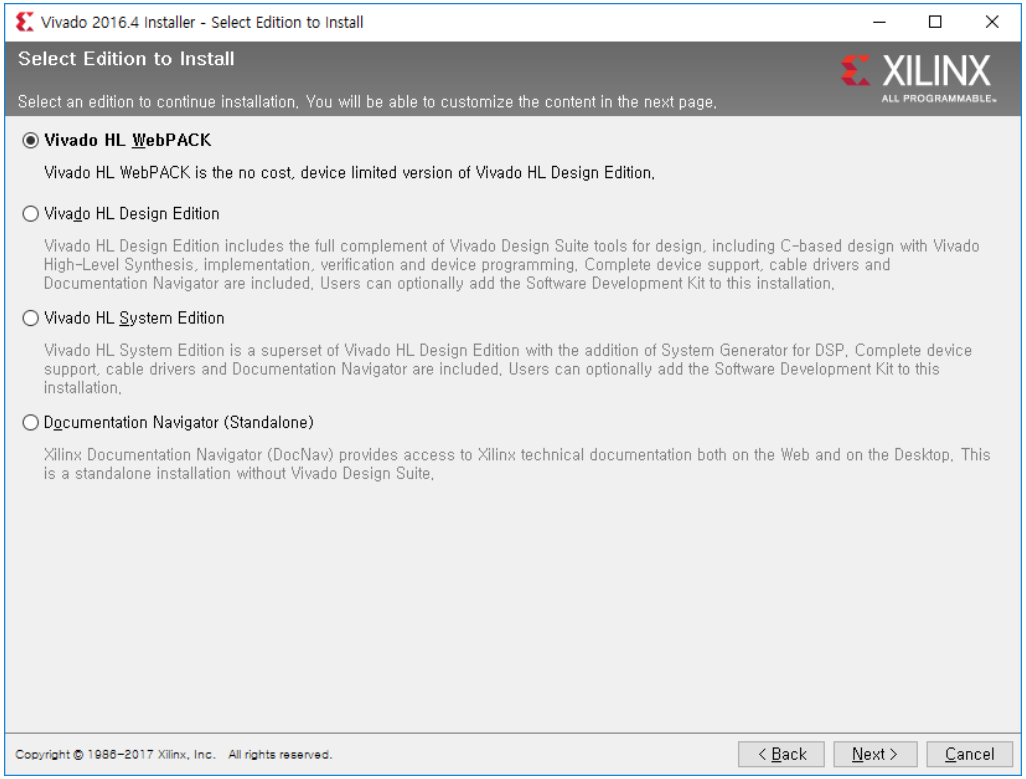




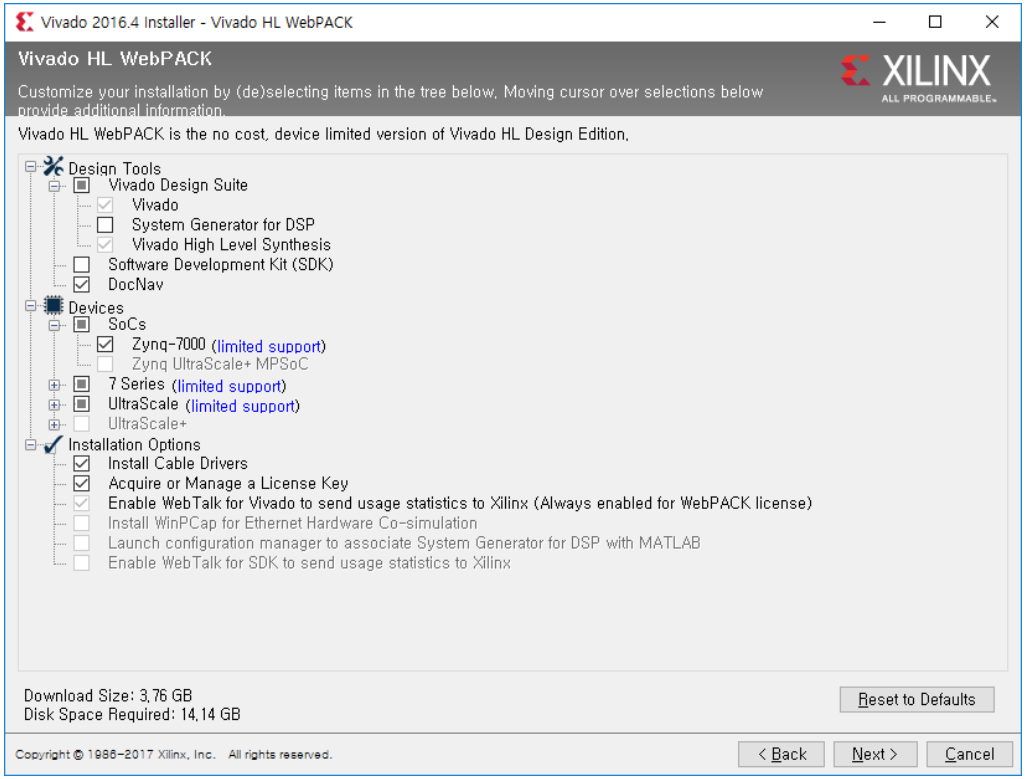
6. Check three "I Agree"s, and press next



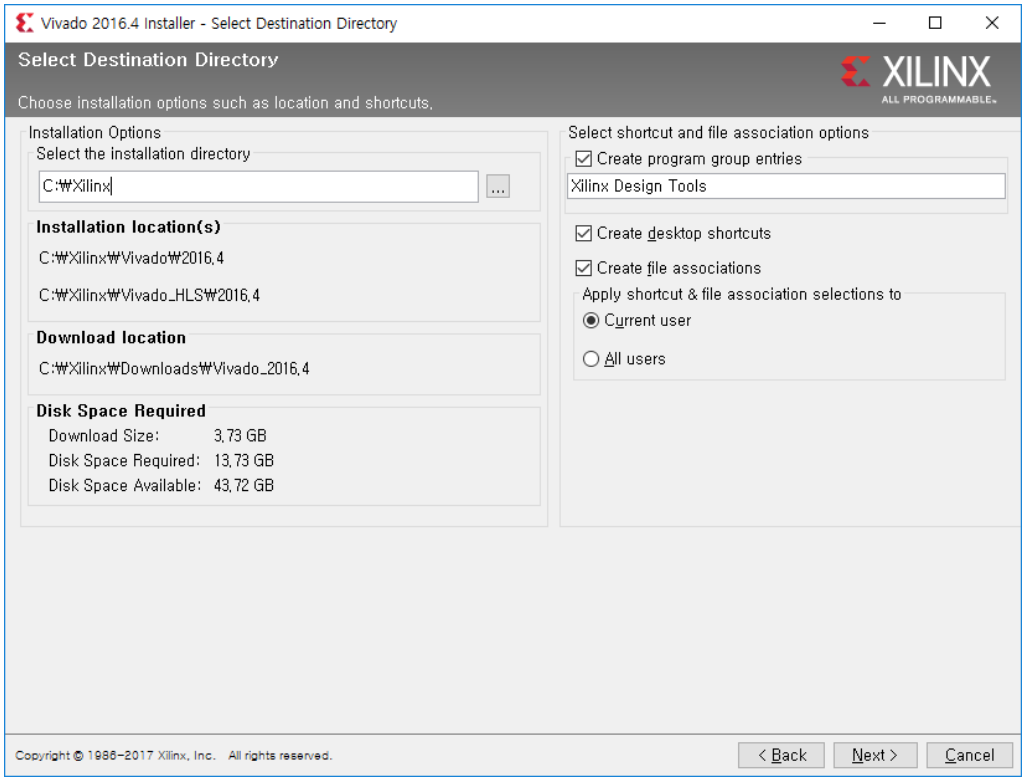
7. Choose WebPack Edition (It's free!!)



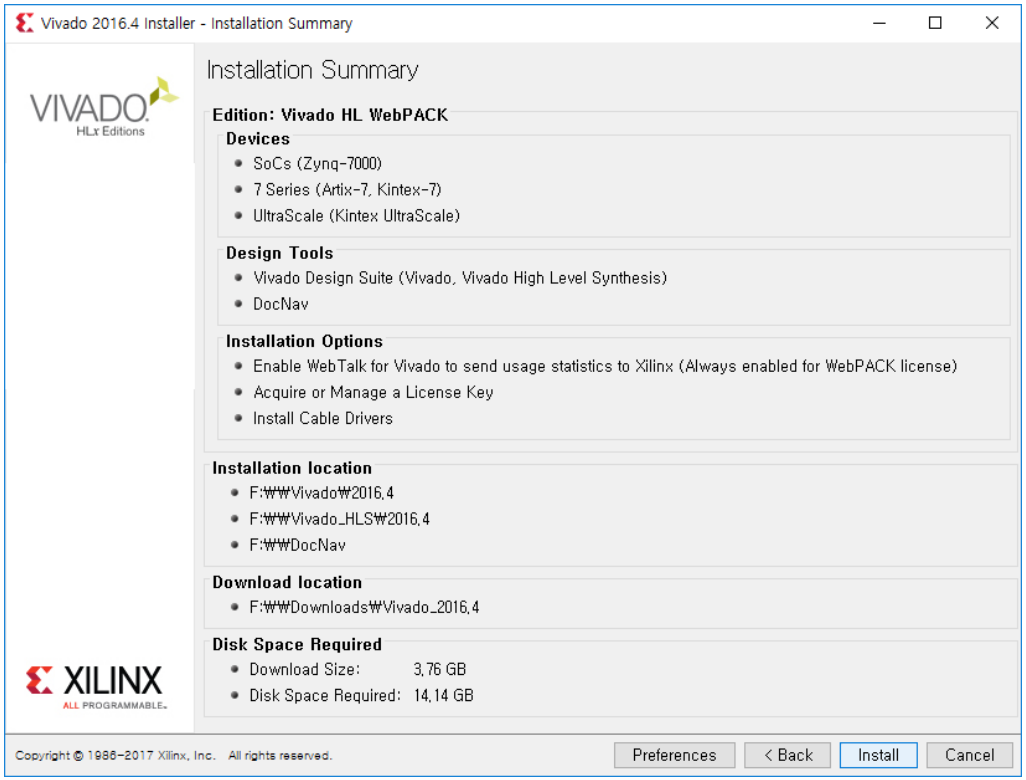
8. Press next


Vivado 2016.4 Installer - Vivado HL WebPACK
Vivado HL WebPACK
Customize your installation by (de)selecting items in the tree below. Moving cursor over selections below provide additional information.
XILINX
ALL PROGRAMMABLE.
Vivado HL WebPACK is the no cost, device limited version of Vivado HL Design Edition.
Design Tools
Vivado Design Suite
Vivado
System Generator for DSP
Vivado High Level Synthesis
Software Development Kit (SDK)
DocNav
Devices
SoCs
Zynq-7000 (limited support)
Zynq UltraScale+ MPSoC
7 Series (limited support)
UltraScale (limited support)
UltraScale+
Installation Options
Install Cable Drivers
Acquire or Manage a License Key
Enable WebTalk for Vivado to send usage statistics to Xilinx (Always enabled for WebPACK license)
Install WinPCap for Ethernet Hardware Co-simulation
Launch configuration manager to associate System Generator for DSP with MATLAB
Enable WebTalk for SDK to send usage statistics to Xilinx
Download Size: 3.76 GB
Disk Space Required: 14.14 GB
Reset to Defaults
Copyright © 1986-2017 Xilinx, Inc. All rights reserved.
< Back
Next >
Cancel

9. Press next


Vivado 2016.4 Installer - Select Destination Directory
Select Destination Directory
Choose installation options such as location and shortcuts.
XILINX
ALL PROGRAMMABLE.
Installation Options
Select the installation directory
C:₩Xilinx
...
Installation location(s)
C:₩Xilinx₩Vivado₩2016.4
C:₩Xilinx₩Vivado_HLS₩2016.4
Download location
C:₩Xilinx₩Downloads₩Vivado_2016.4
Disk Space Required
Download Size: 3.73 GB
Disk Space Required: 13.73 GB
Disk Space Available: 43.72 GB
Select shortcut and file association options
Create program group entries
Xilinx Design Tools
Create desktop shortcuts
Create file associations
Apply shortcut & file association selections to
Current user
All users
Copyright © 1986-2017 Xilinx, Inc. All rights reserved.
< Back
Next >
Cancel

10. Press "Install"

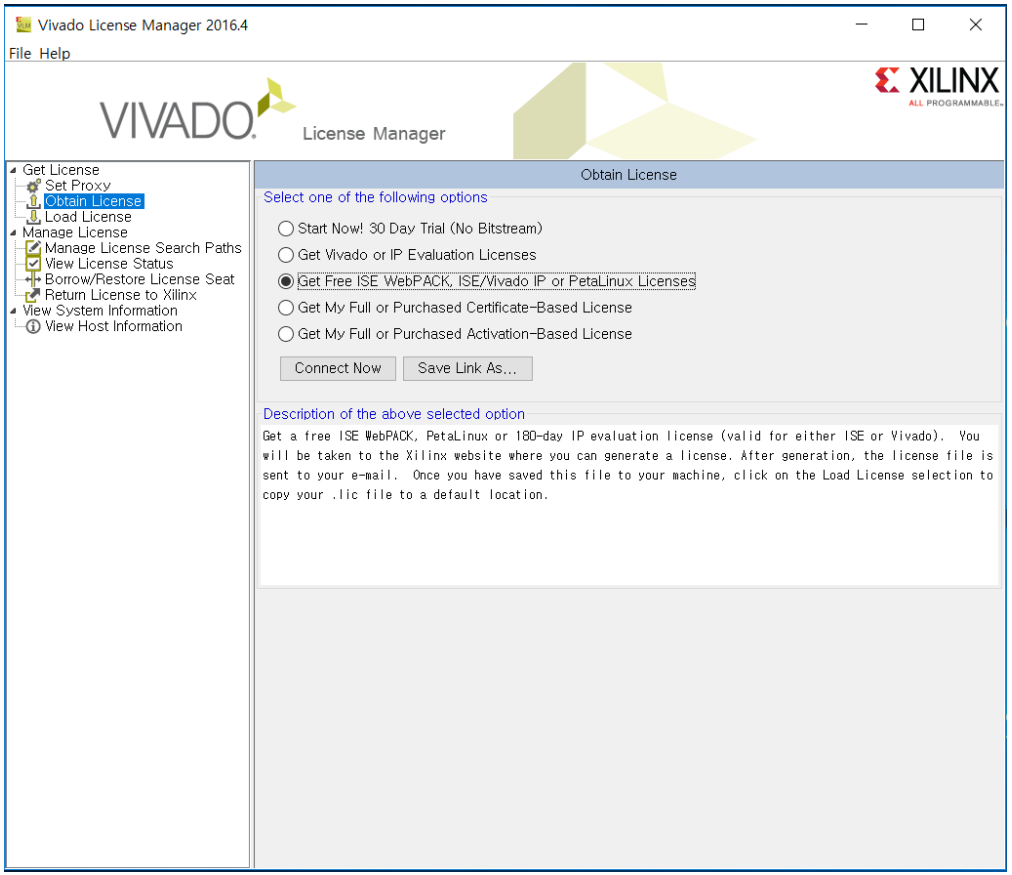


11. Wait until installing is finished.

12. Finish!

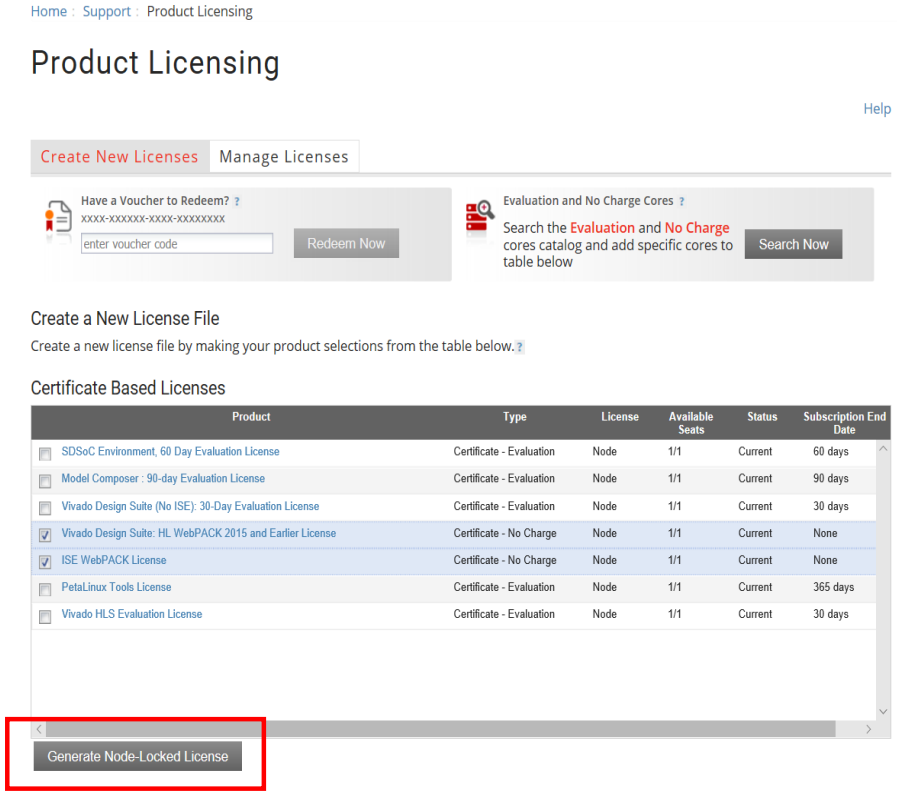

Vivado 2016.4 Installer - Installation Progress
Installation Progress
XILINX
ALL PROGRAMMABLE.
It took 6 minutes to download files.
It took 12 minutes to install files.
Done Final Processing.
ZYNQ
Hardware Programmable
Software Programmable
AMS
I/O Programmable
Xilinx Software Install
Installation completed successfully.
확인
Zynq All Programmable SoCs Enable
the Development of Smarter Systems
Copyright © 1986-2018 Xilinx, Inc. All rights reserved.
< Back
Install
Cancel

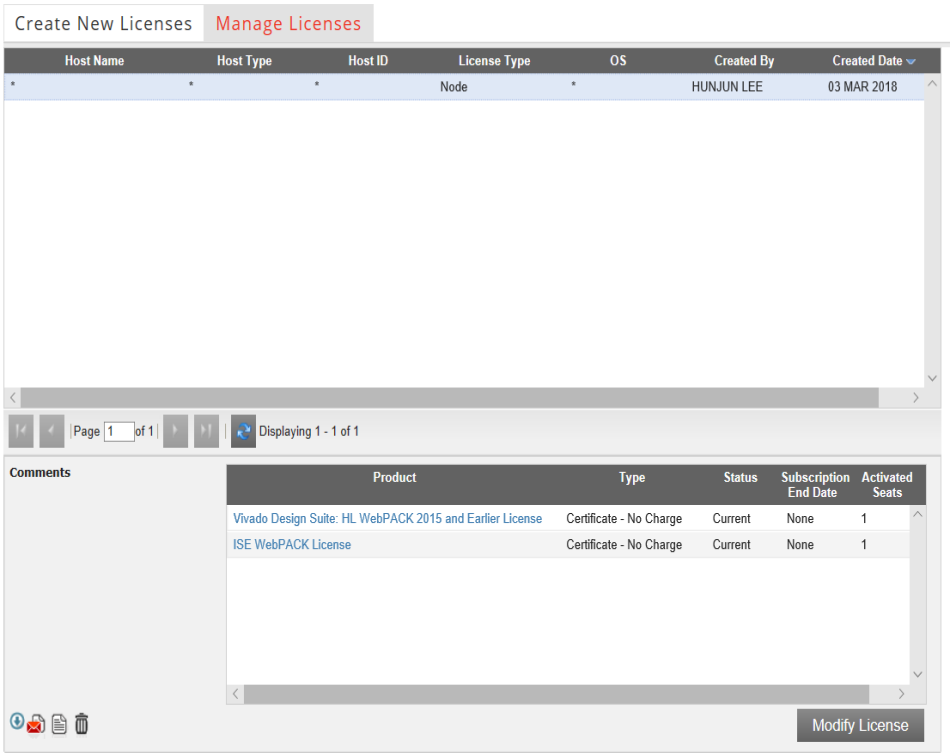
13. License Manager:
Check "Get Free ISE WebPack ISE/Vivado or PetaLinux License", and click “Connect Now”.



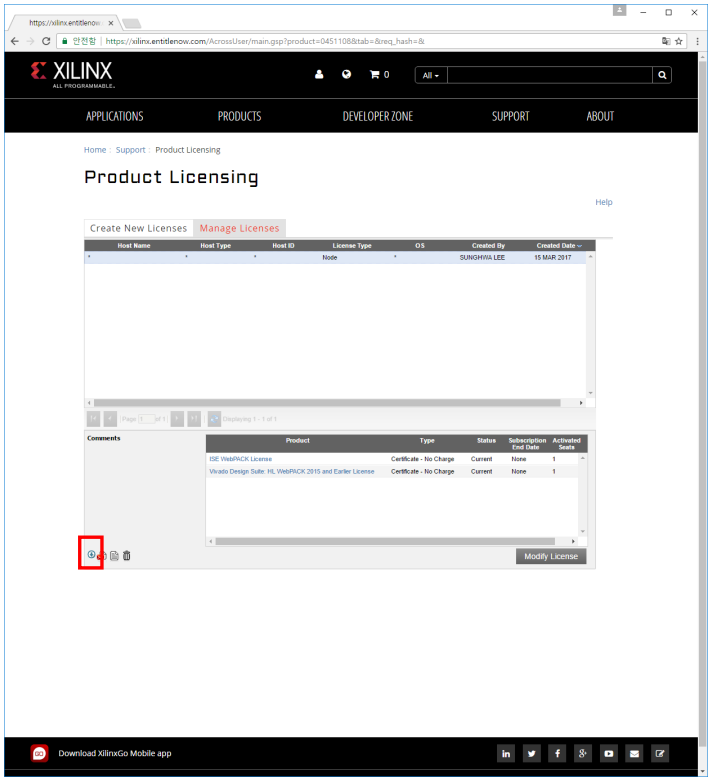
You may open
License Manager manually

14. Create a Vivado WebPACK license.

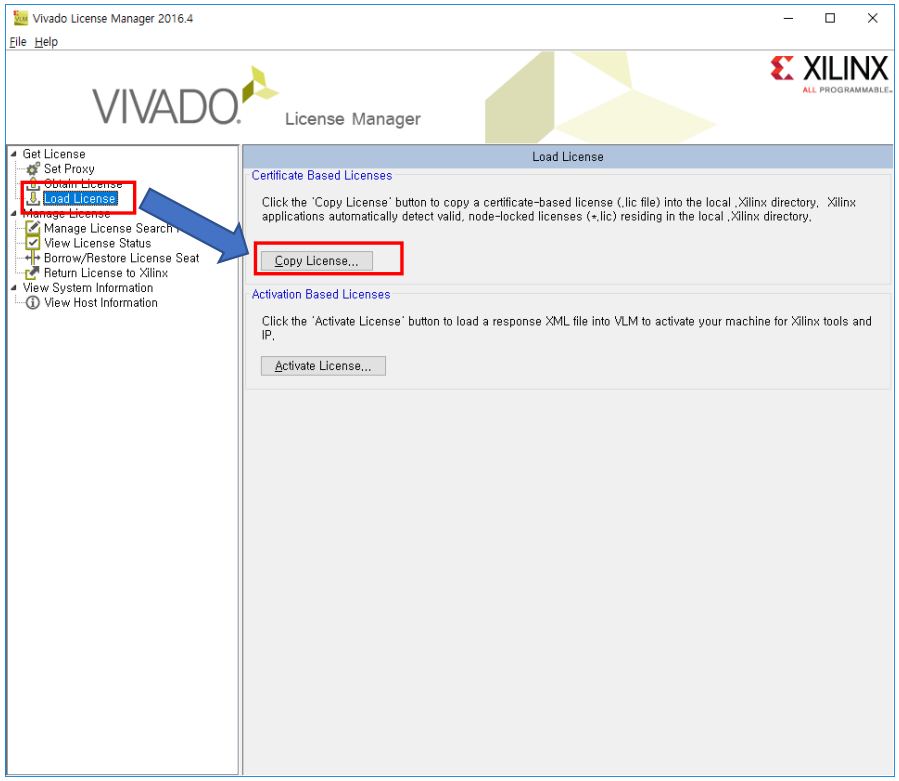




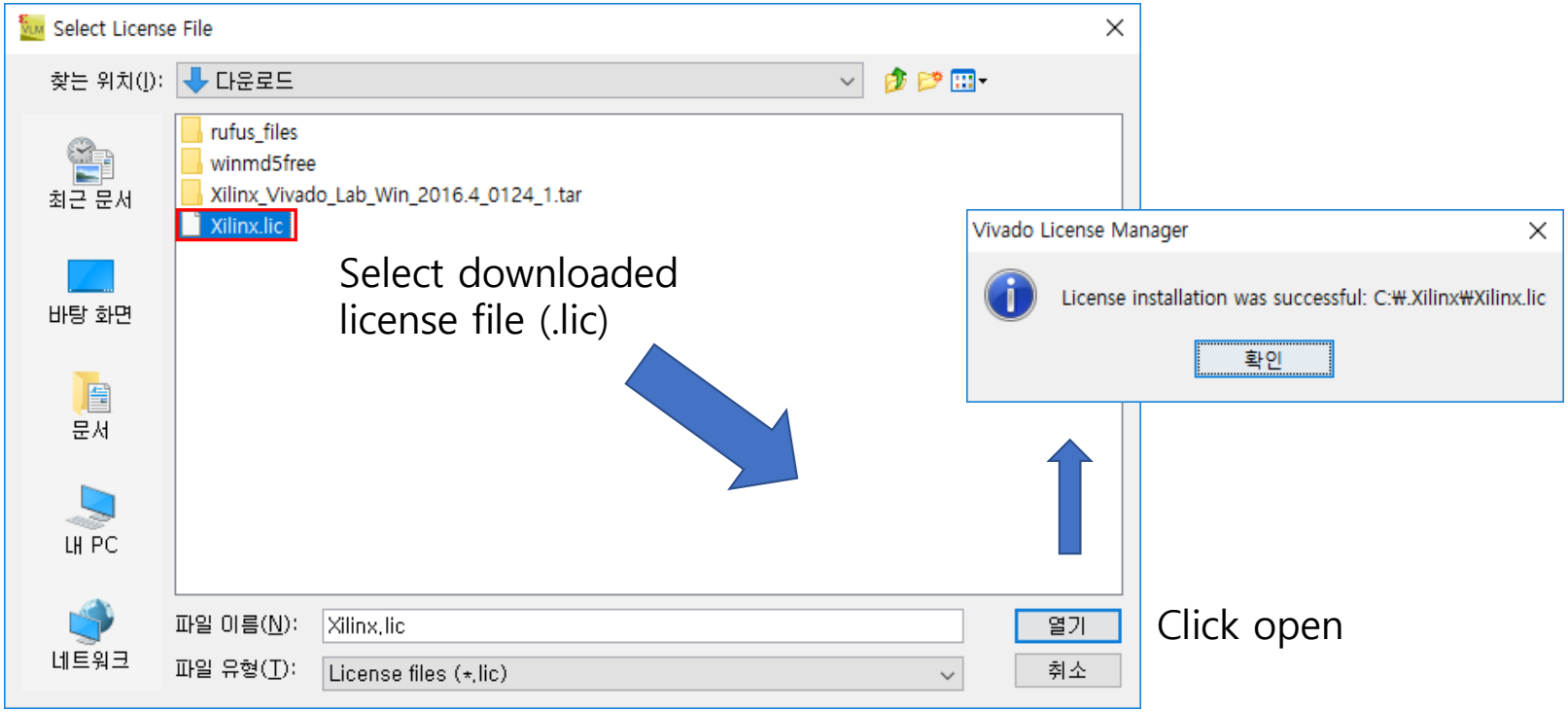
15. Select licenses and Download it



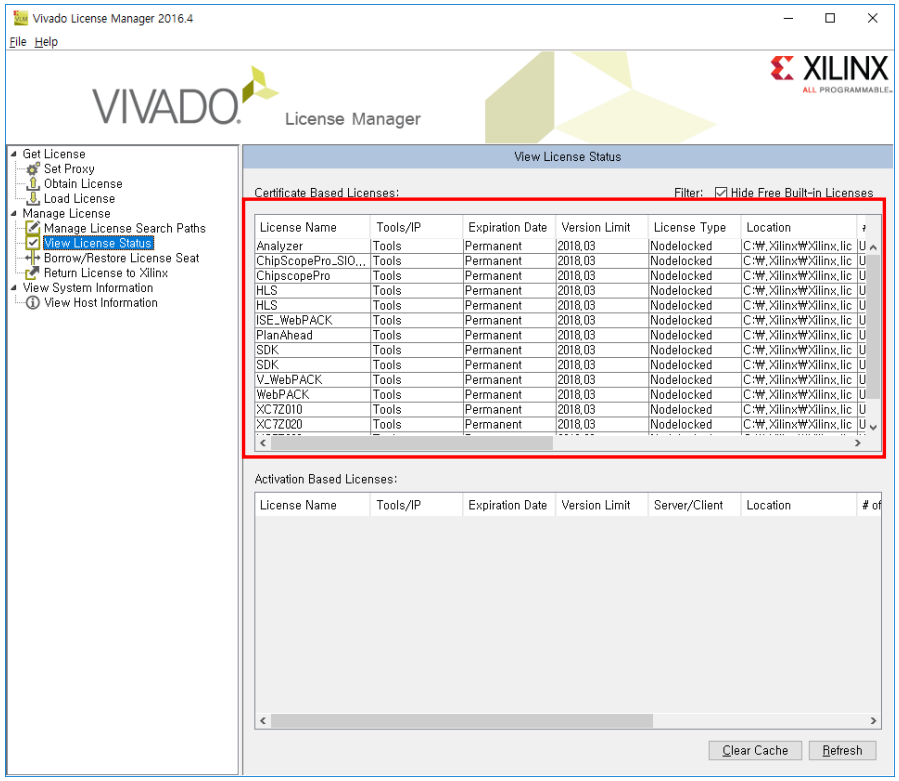
16. Load License



17. Copy License



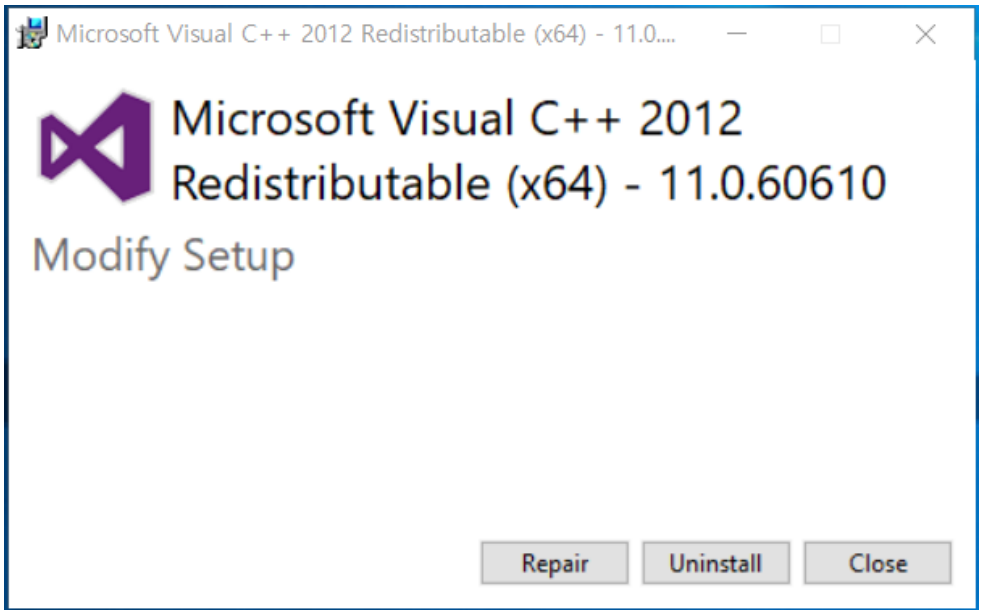
18. Check License Status (FINISH)



TroubleShooting

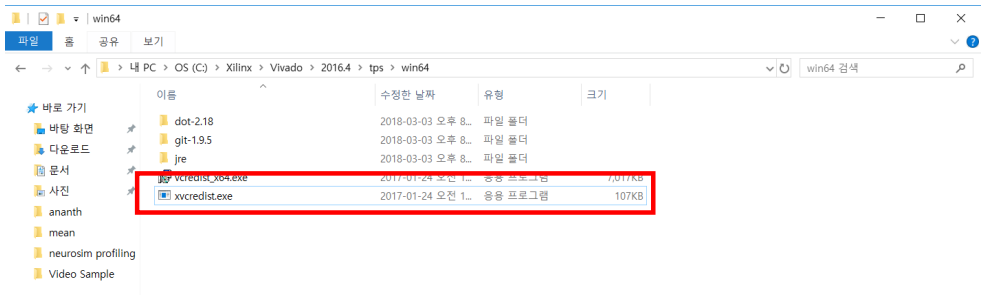
1. Visual C++ 2012 modify setup problem

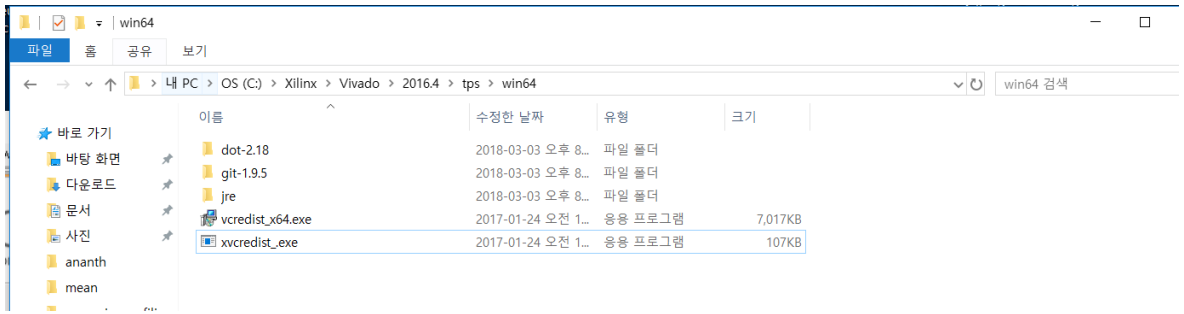
In some situations, Visual C++ 2012 Redistributable problem occurs after executing Vivado



2. Visual C++ 2012 modify setup problem

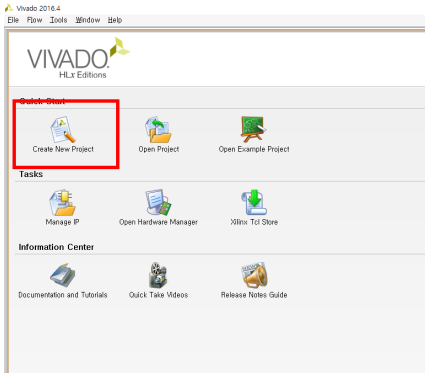
Change C:/Xilinx/Vivado/<version.no>/tps/win64/xvcredist.exe to C:/Xilinx/Vivado/<version.no>/tps/win64/xvcredist_.exe

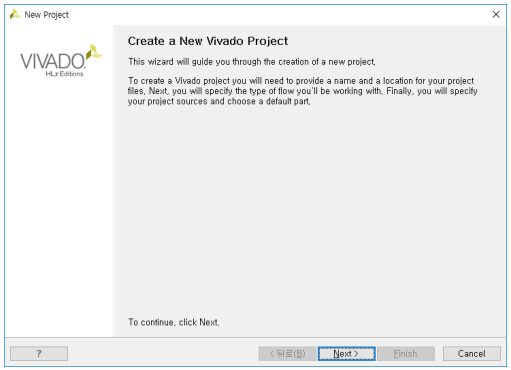




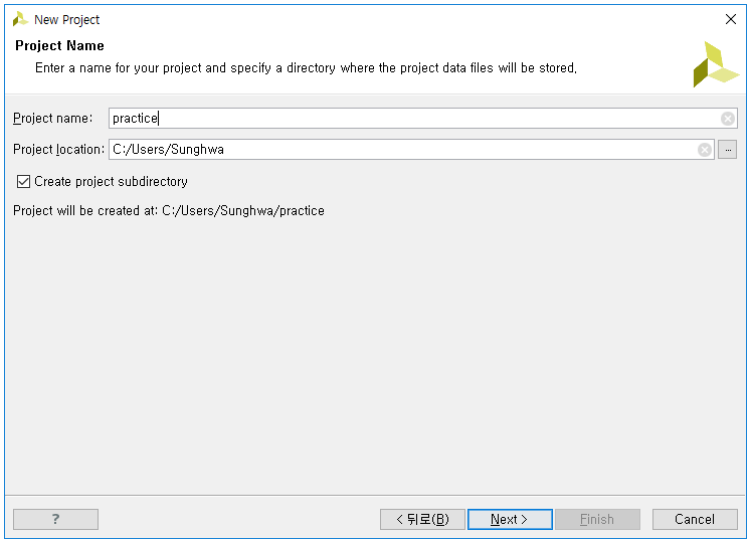
How to make a project

1. Run Vivado and create new project



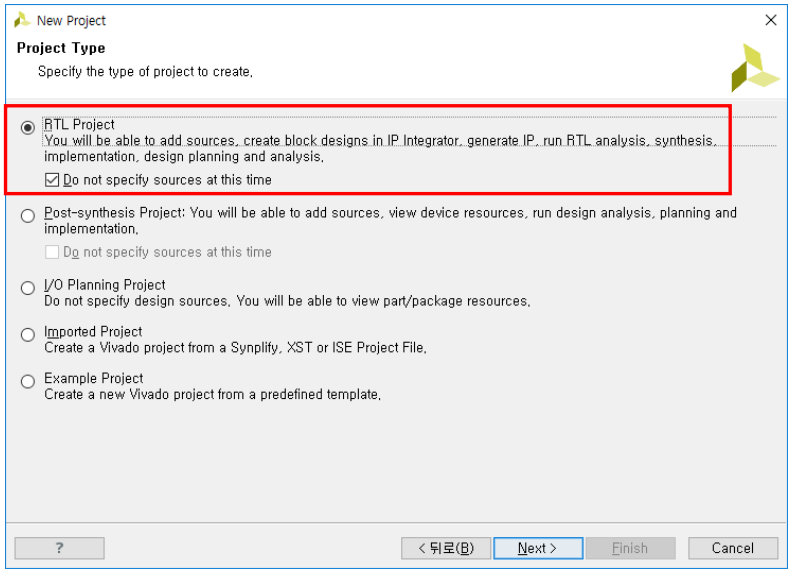


Click Next

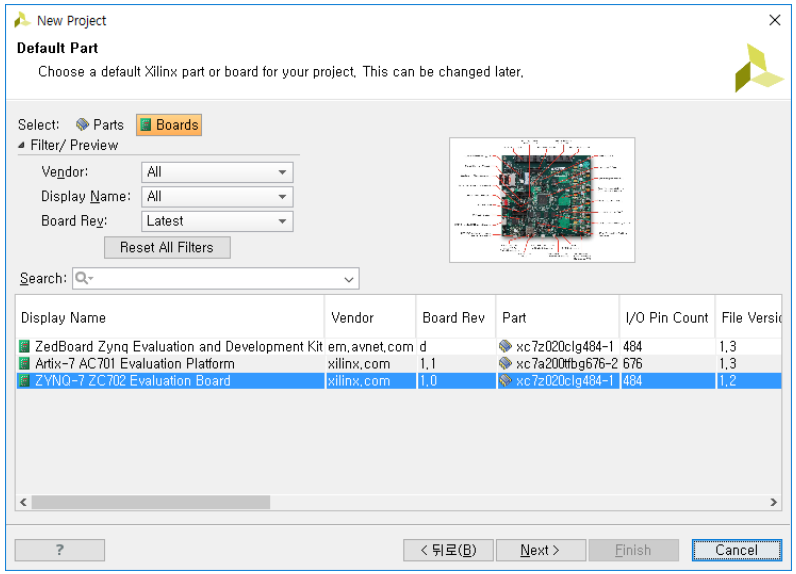


Enter a project name

2. Create a new project



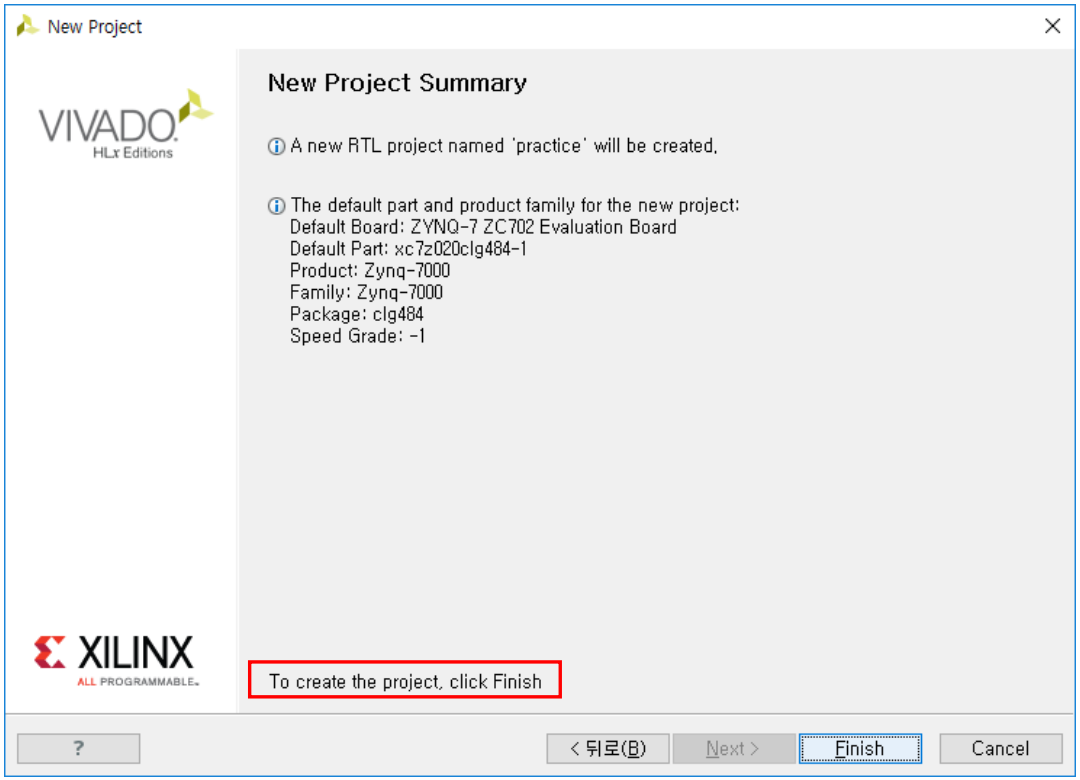
Choose RTL Project



Choose Parts/Boards

- We don't use any boards in this course, so just ignore this part and press next.

3. Create a new project



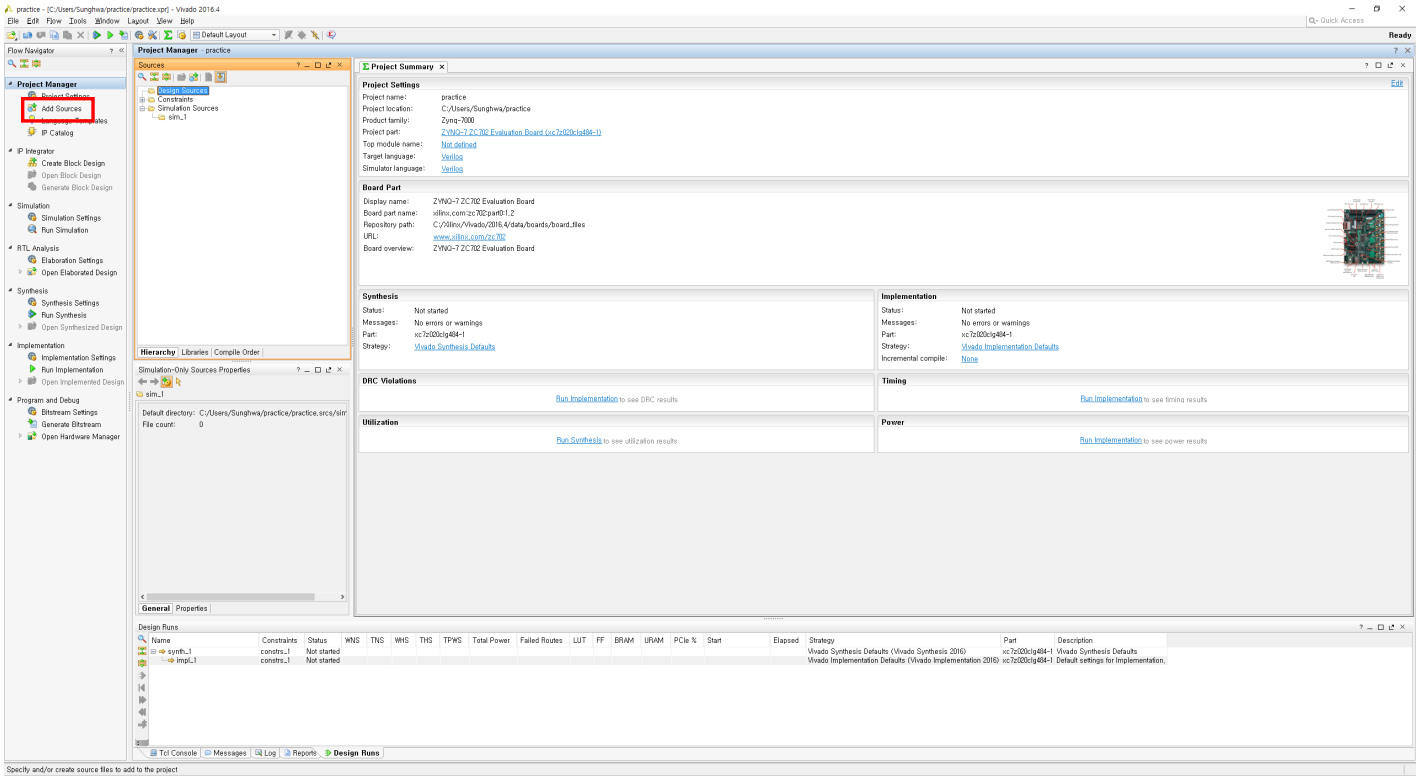
Simple Verilog Tutorial using Vivado

Counter

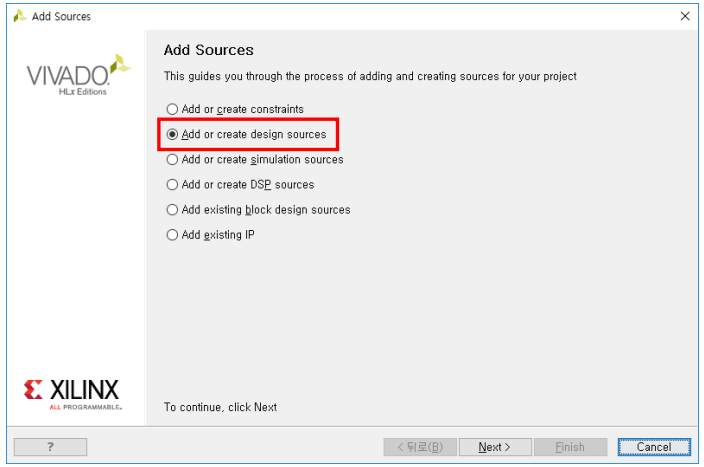
- 4-bit synchronous up-counter
- That is, the counter is incremented when the clock signal is positive edge.

1. Add Design Sources

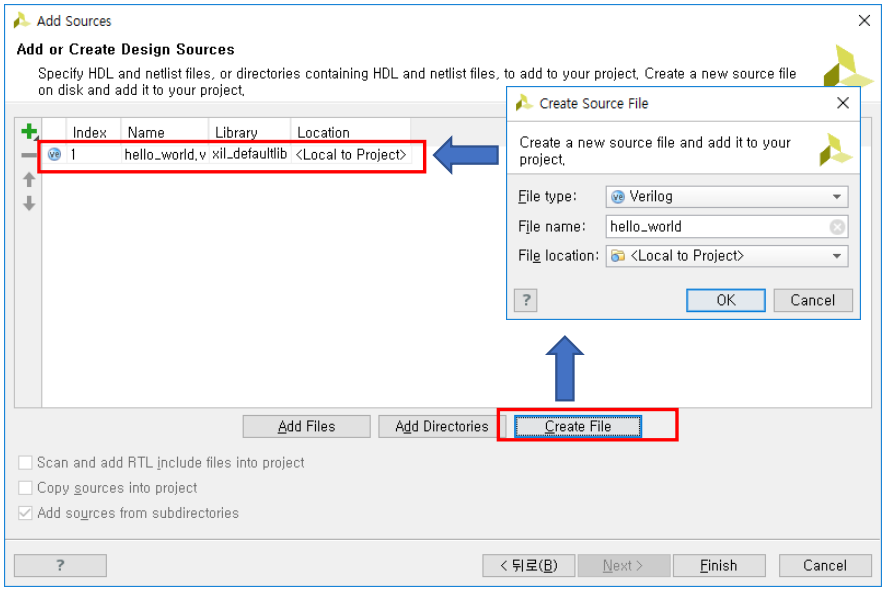
Click
Add Sources



2. Add Design Sources



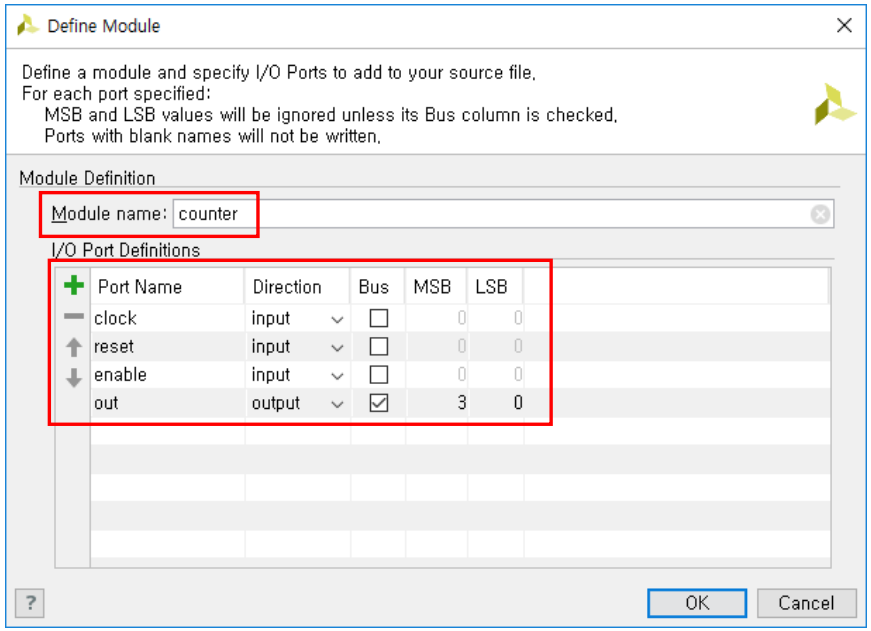
To continue, click Next



1. Click Create File
2. Enter file name (“hello_world”)
3. Click Finish

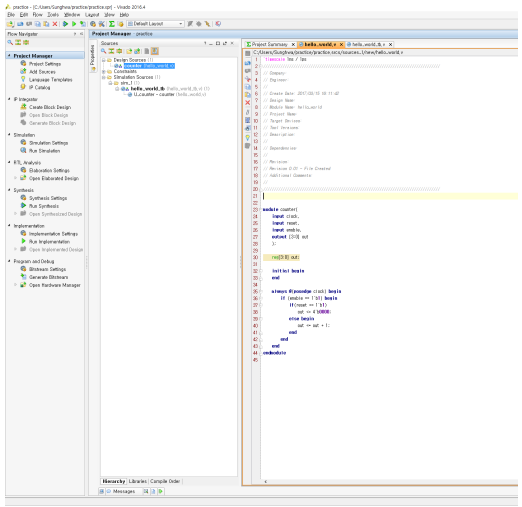
3. Add Design Sources

- Change module name and enter I/O Port Deifinitions



4. Write code

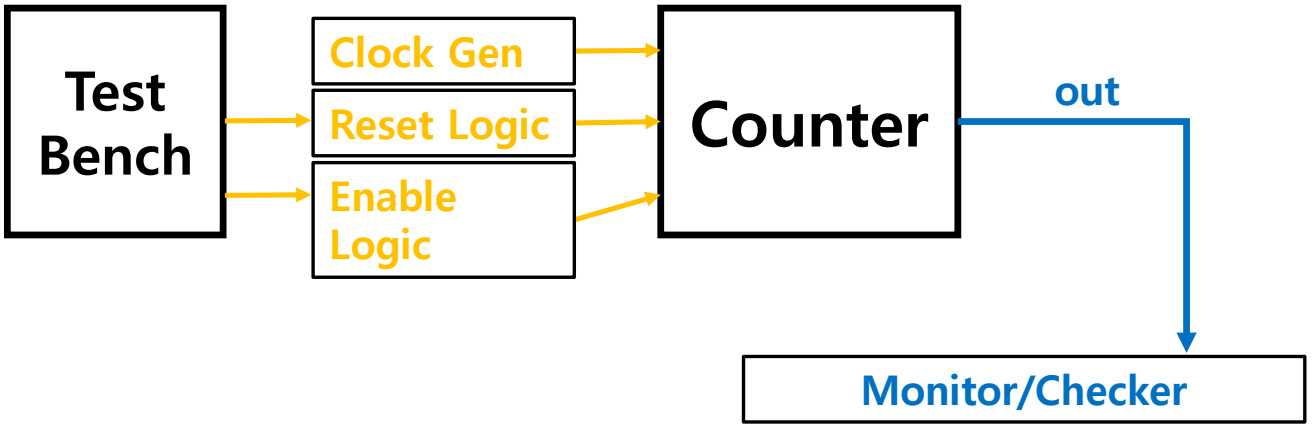
(* Copy this code and paste it *)



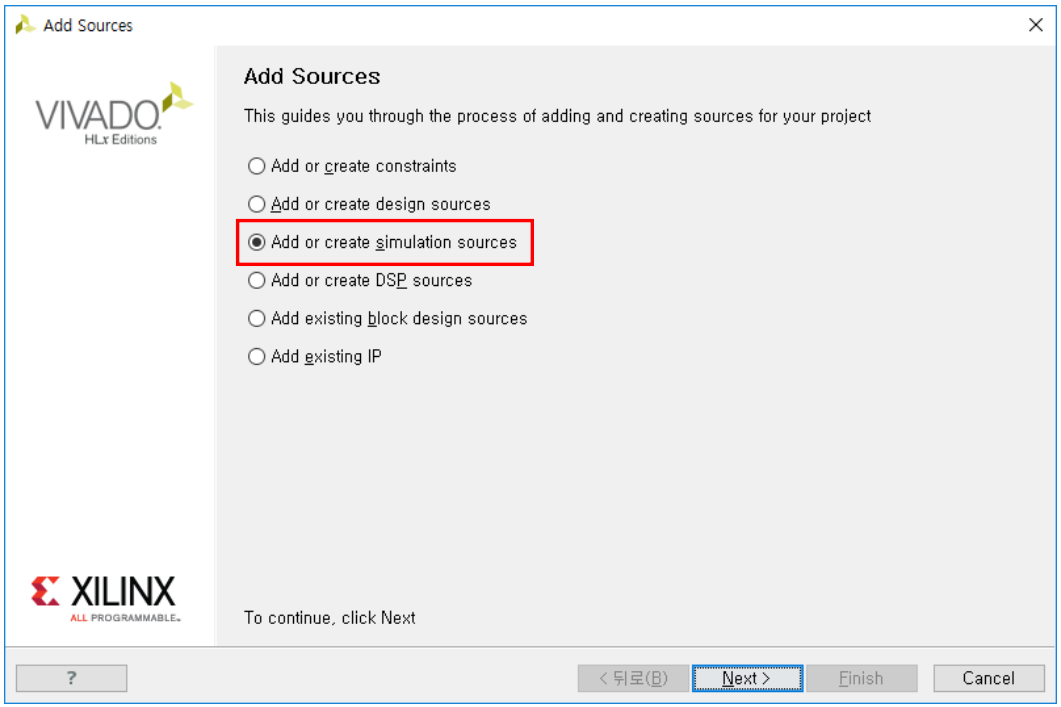
```
`timescale 1ns / 100ps


module counter (input clock, input reset, input enable, output [3:0] out);
        reg [3:0] out;

        initial begin
        end

        always @(posedge clock) begin
                if (enable == 1'b1) begin
                        if (reset == 1'b1)
                                out <= 4'b0000;
                        else begin
                                out <= out+1;
                        end
                end
        end
endmodule
```

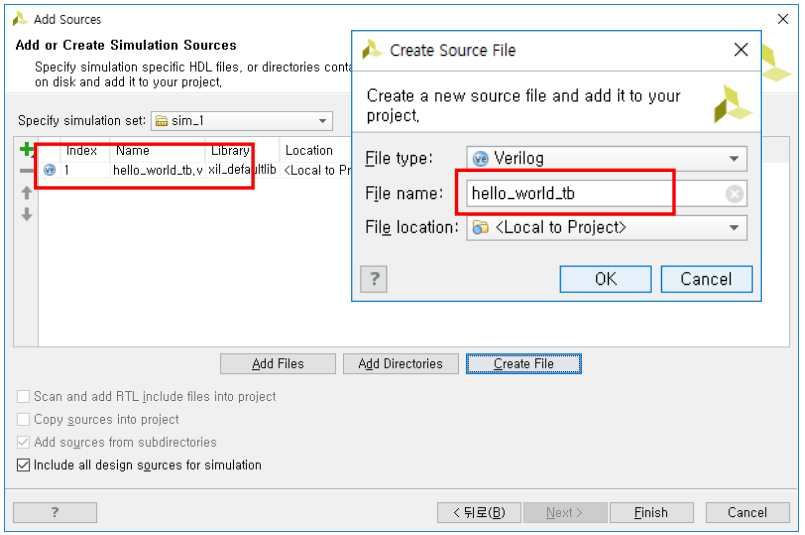
5. Write down your testbench

• To test the counter, a testbench file is needed.

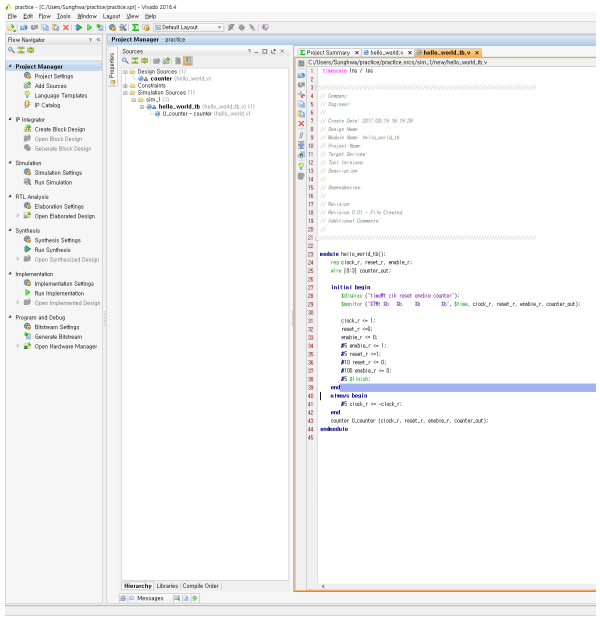


6. Add a testbench source (Click add sources)





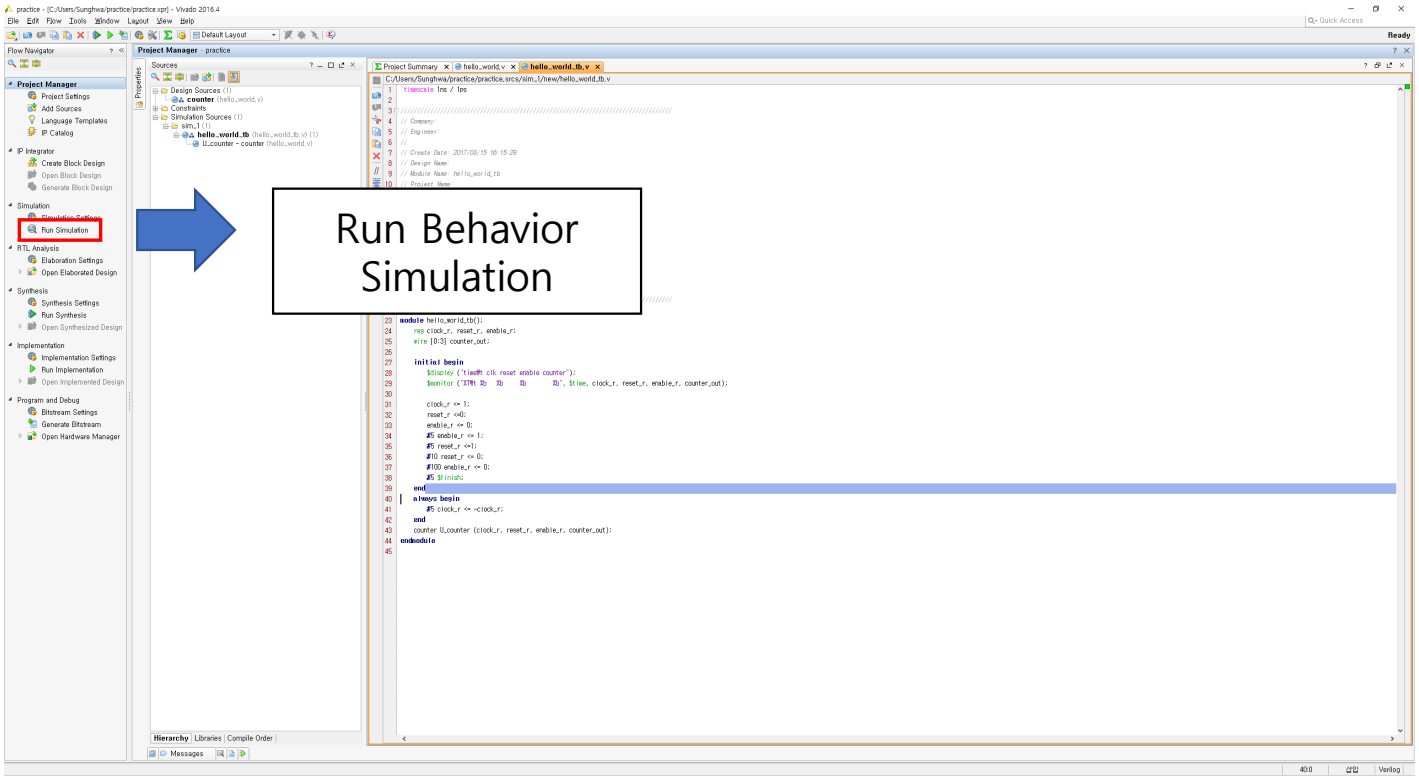
7. Write code



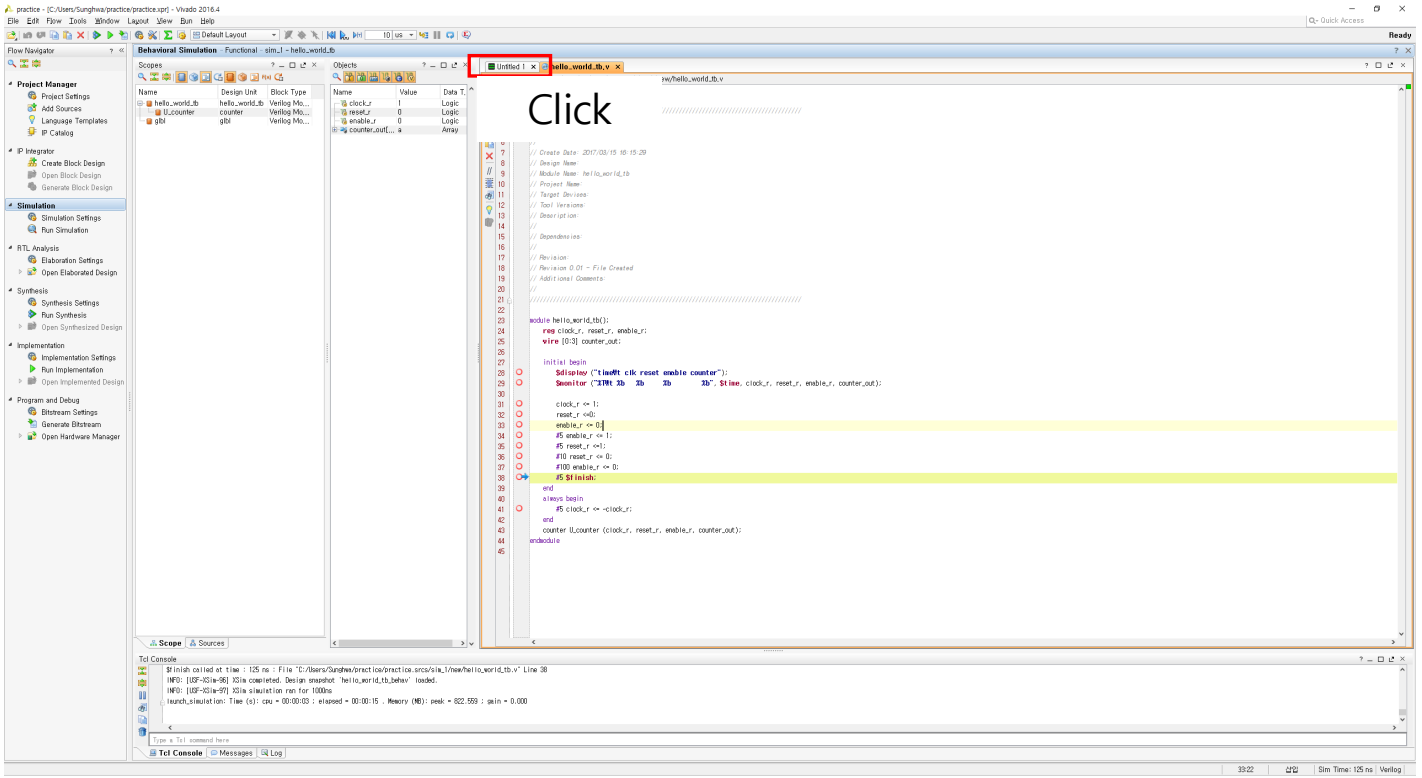
(* Copy this code and paste it *)

```
module hello_world_tb();
	reg clock_r, reset_r, enable_r;
	wire [3:0] counter_out;

	initial begin
		$display ("time₩t clk reset enable counter");
		$monitor ("%T₩t %b  %b  %b  %b", $time, clock_r,
reset_r, enable_r, counter_out);

		clock_r <= 1;
		reset_r <=0;
		enable_r <= 0;
		#5 enable_r <= 1;
		#5 reset_r <=1;
		#10 reset_r <= 0;
		#100 enable_r <= 0;
		#5 $finish;
	end
	always begin
		#5 clock_r <= ~clock_r;
	end
	counter U_counter (clock_r, reset_r, enable_r, counter_out);
endmodule
```

8. Run Simulation!



9. Run simulation



10. Check the result