Introduction to the EDA Tools for Digital Labs

National Chiao Tung University
Chun-Jen Tsai
9/18/2015

The Target Technology of Digital Labs

- Digital circuits can be implemented in different ways
  - Application Specific ICs
    - Full-custom IC design
    - Cell-based IC design
  - Programmable logics
    - Complex Programmable Logic Device (CPLD) design
    - Field Programmable Gate Array (FPGA) design
- In this course, we use Xilinx FPGAs for implementation
  - Xilinx is the largest FPGA design company in the world
  - Freeman, the co-founder of Xilinx, invented the very first SRAM-based FPGA in 1986

Xilinx ISE Design Suite

- Xilinx has two different EDA tools for FPGA-based digital system implementation
  - Vivado Design Suite
    - Only for 7th-generation FPGAs and above
    - Unified IDE for complex digital system design
  - ISE Design Suite
    - For 7th-generation FPGAs and before
    - ISE EDK for HW-SW system implementation
    - ISE Project Navigator for digital circuit implementation

ISE Project Navigator Design Flow

- Step 1: Design Entry
  - Input your circuit design using Hardware Description Language (HDL), such as Verilog or VHDL, or gate-level schematics
- Step 2: Synthesis
  - Convert from the HDL programs or schematics to a netlist file that define a list of circuit blocks and how they are connected
- Step 3: Mapping
  - Determine what FPGA resource will be used to implement which part of the netlist
- Step 4: Place-and-Route
  - Determine physical location and routing of the circuit resource
  - A BIT file will be generated for the FPGA device

ISE Project Navigator Debug Flow

- Your design may not be perfect in the first try!
  - Circuit debugging is done via “simulation” or “signal probing”
- ISE supports several simulation types:
  - Behavioral simulation
    - Also called functional simulation
    - Simulate signal switching of your circuit without device delay information
  - Post-Place & Route simulation
    - Also called post-sim
    - Simulate signal switching of your circuit with delay information of the target device
- Chipscope is the signal probing tool in ISE
  - When the FPGA is activated with your circuit, Chipscope can be used to probe the real-time signals for debugging purposes

Install Your Own ISE Design Suite

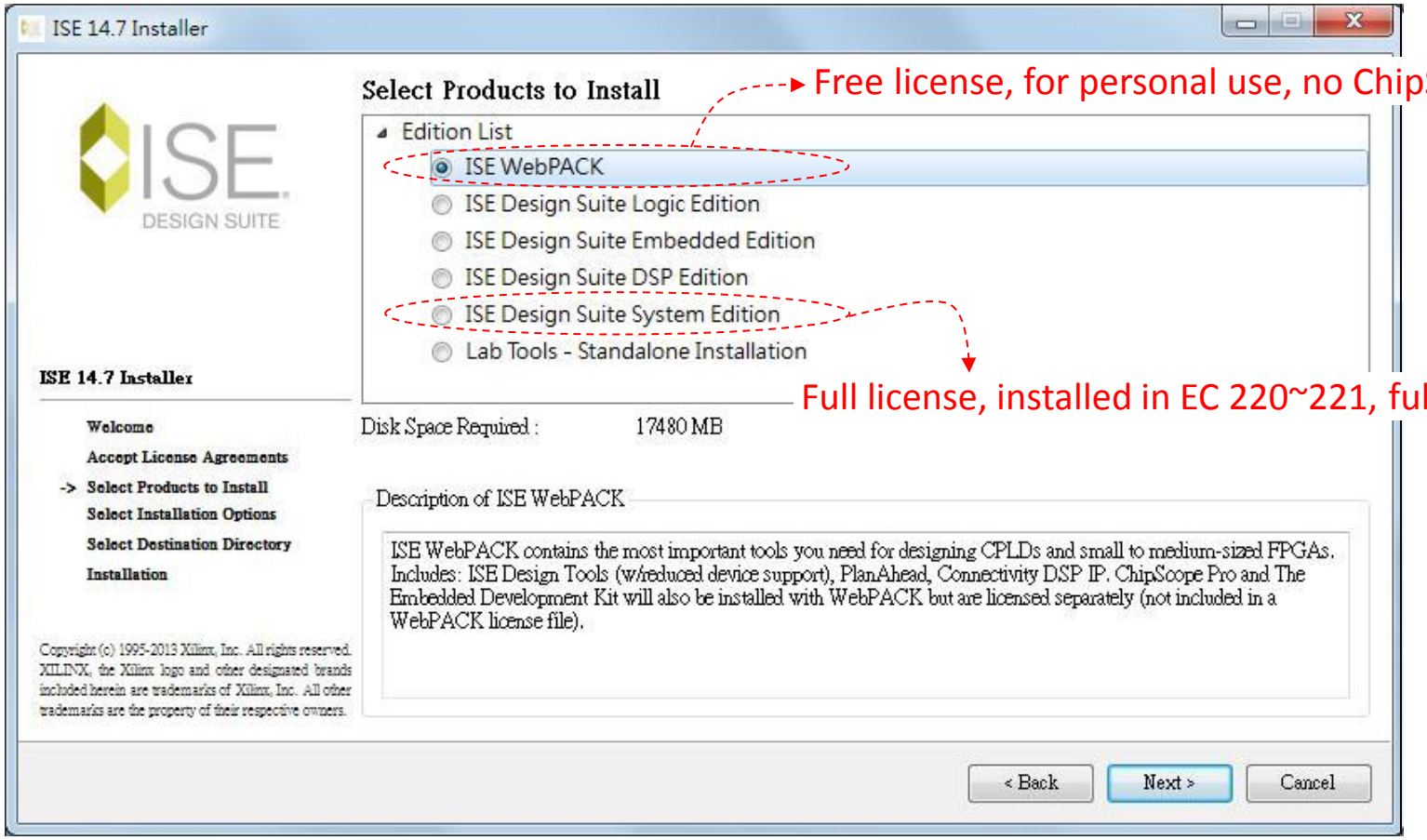
- If you want to install a copy of ISE Design Suite 14.7 onto your computer, you can download it from:

  http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html

- ISE Design Suite on a Windows PC requires 20 GiB of disk space
  - Please install the “WebPACK” version and register online for a free license
  - Note: the free WebPACK version has no Chipscope function
  - ISE only runs on Windows 7 and Linux, for Window 8, you need to do some tricks to get it to run (Google it!)

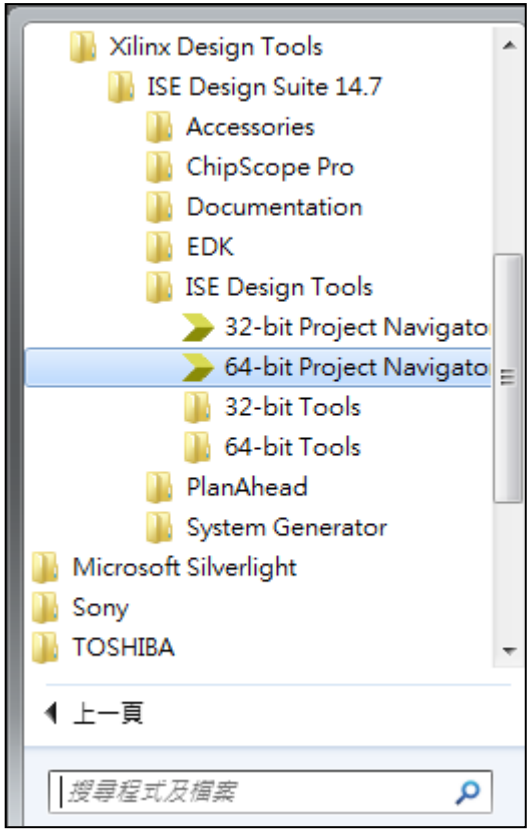
ISE Installation Note

- You must select the right version upon installation:

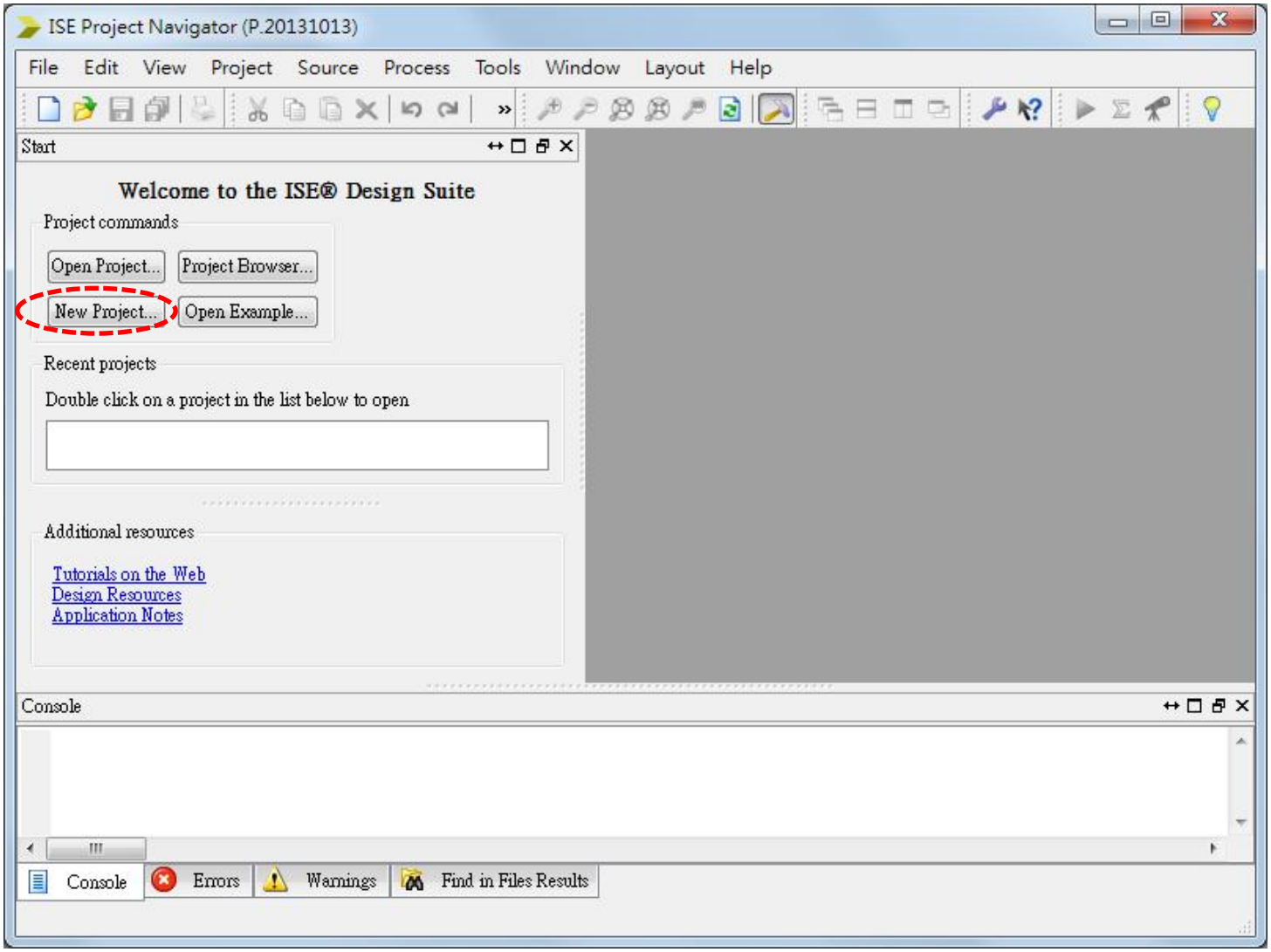


Invoke ISE Project Navigator

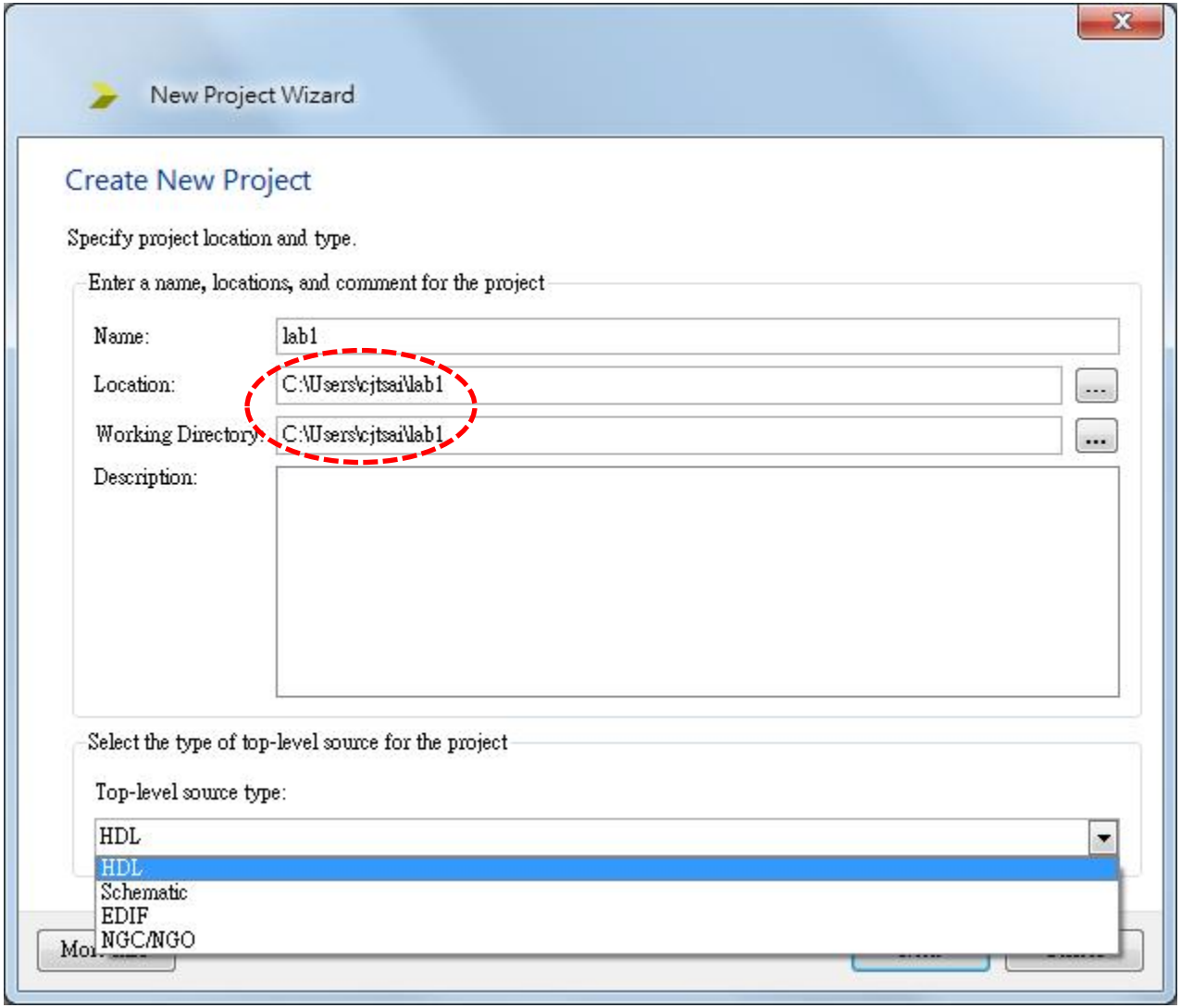
- Under Windows, select
  Start Menu
  → Program
  → Xilinx Design Tools
  → ISE Design Suite 14.7
  → 64-bit Project Navigator



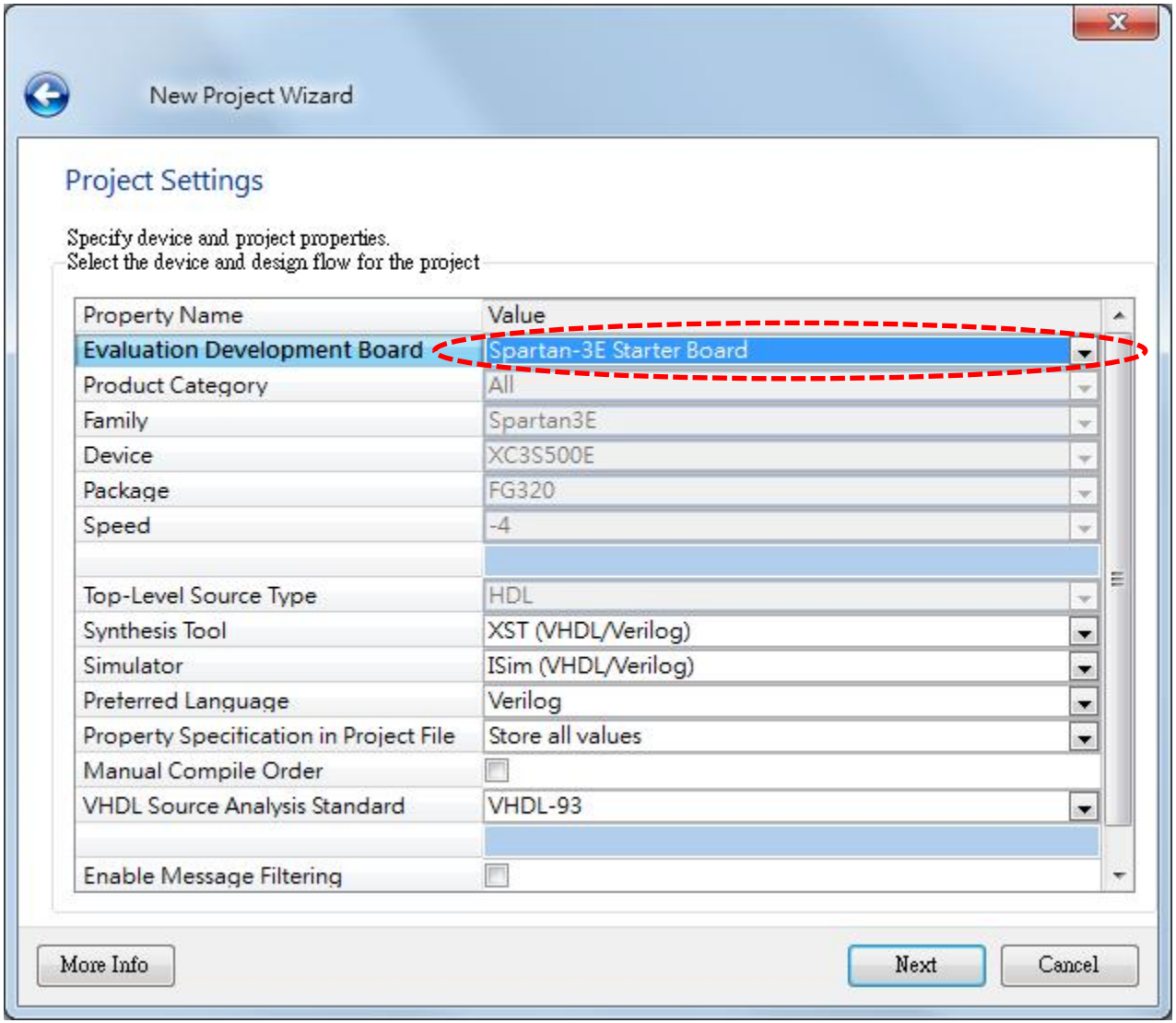
Create a New Project in ISE



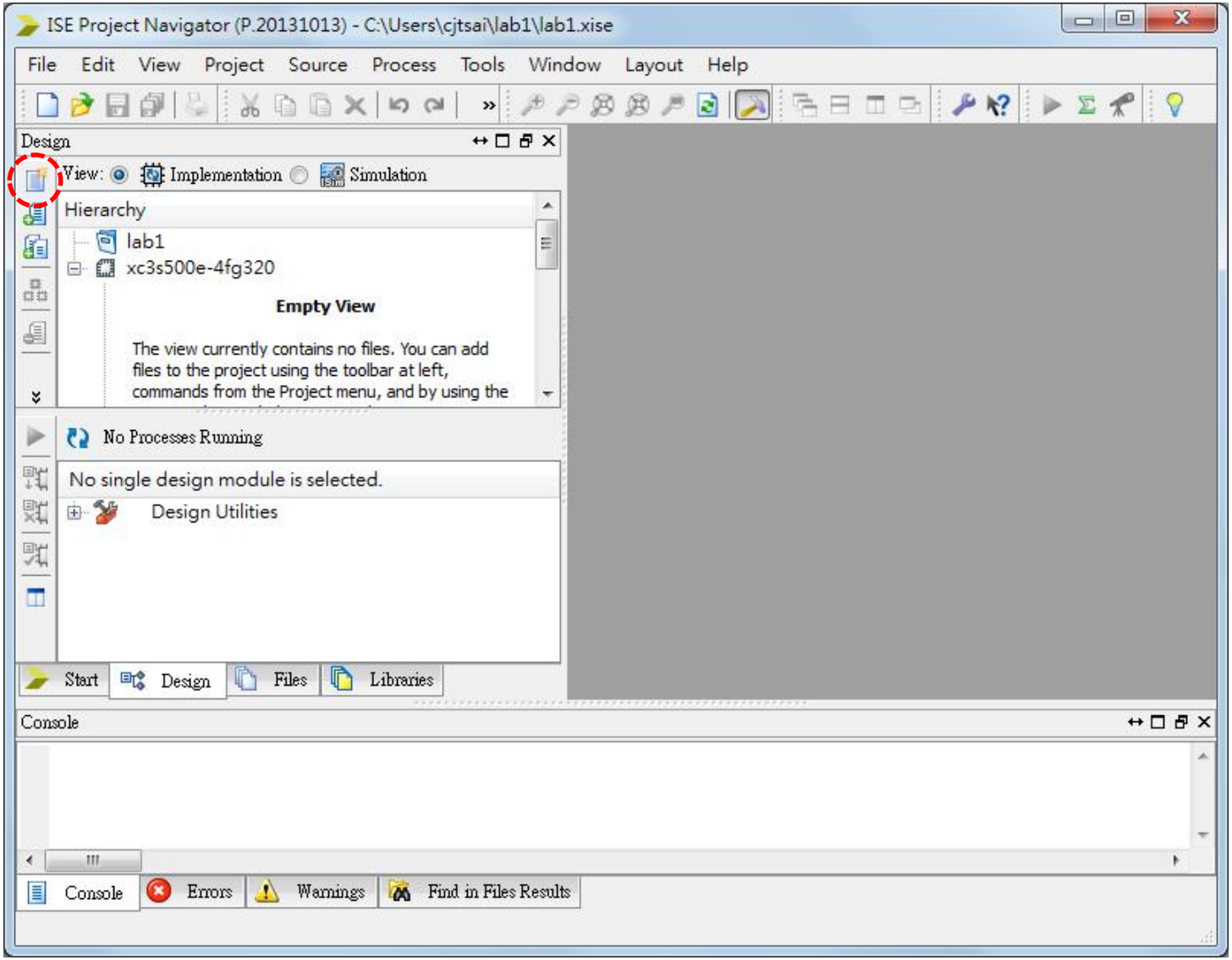
Select HDL or Schematic Design



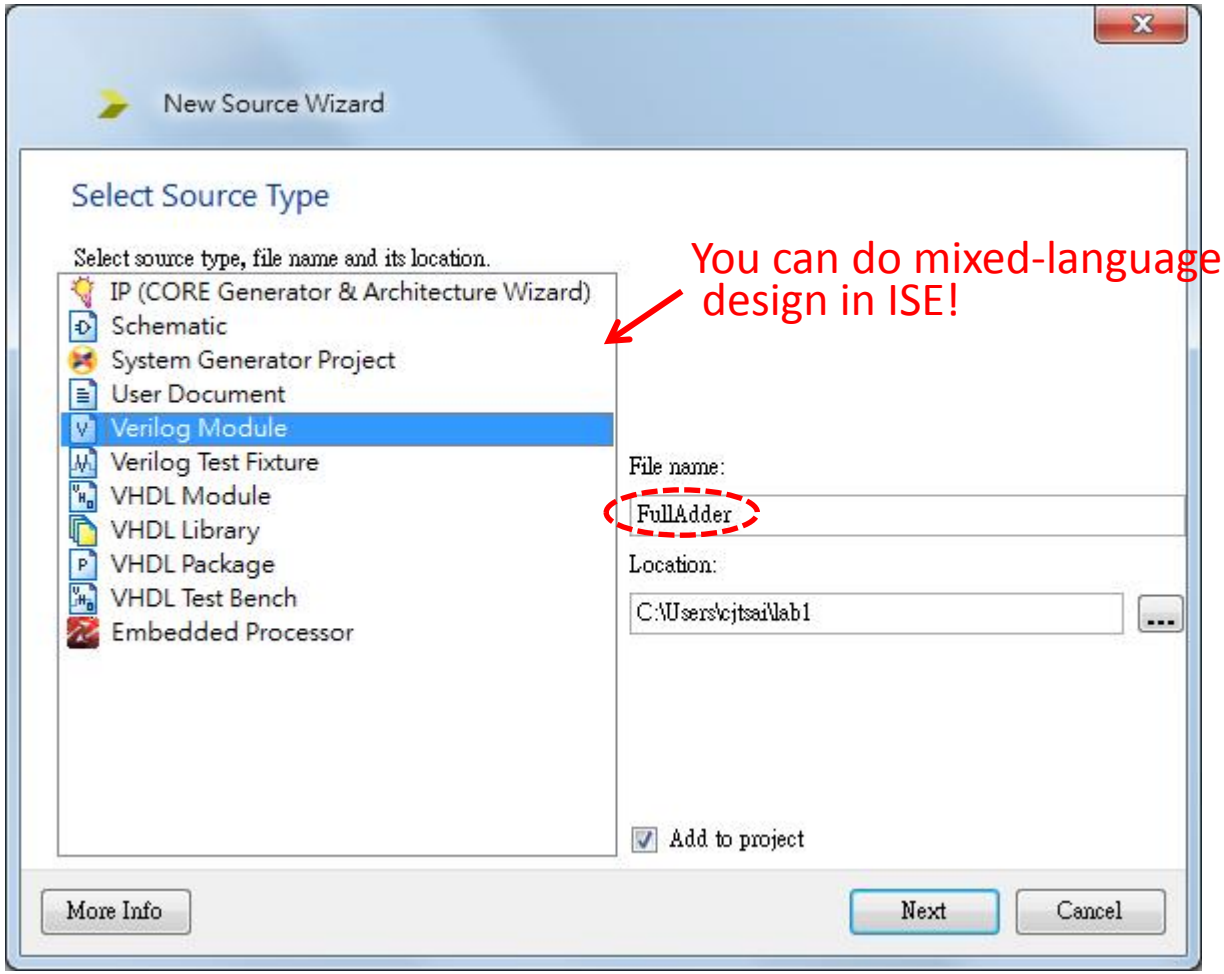
Select the Target Device



Add a New HDL Source Code

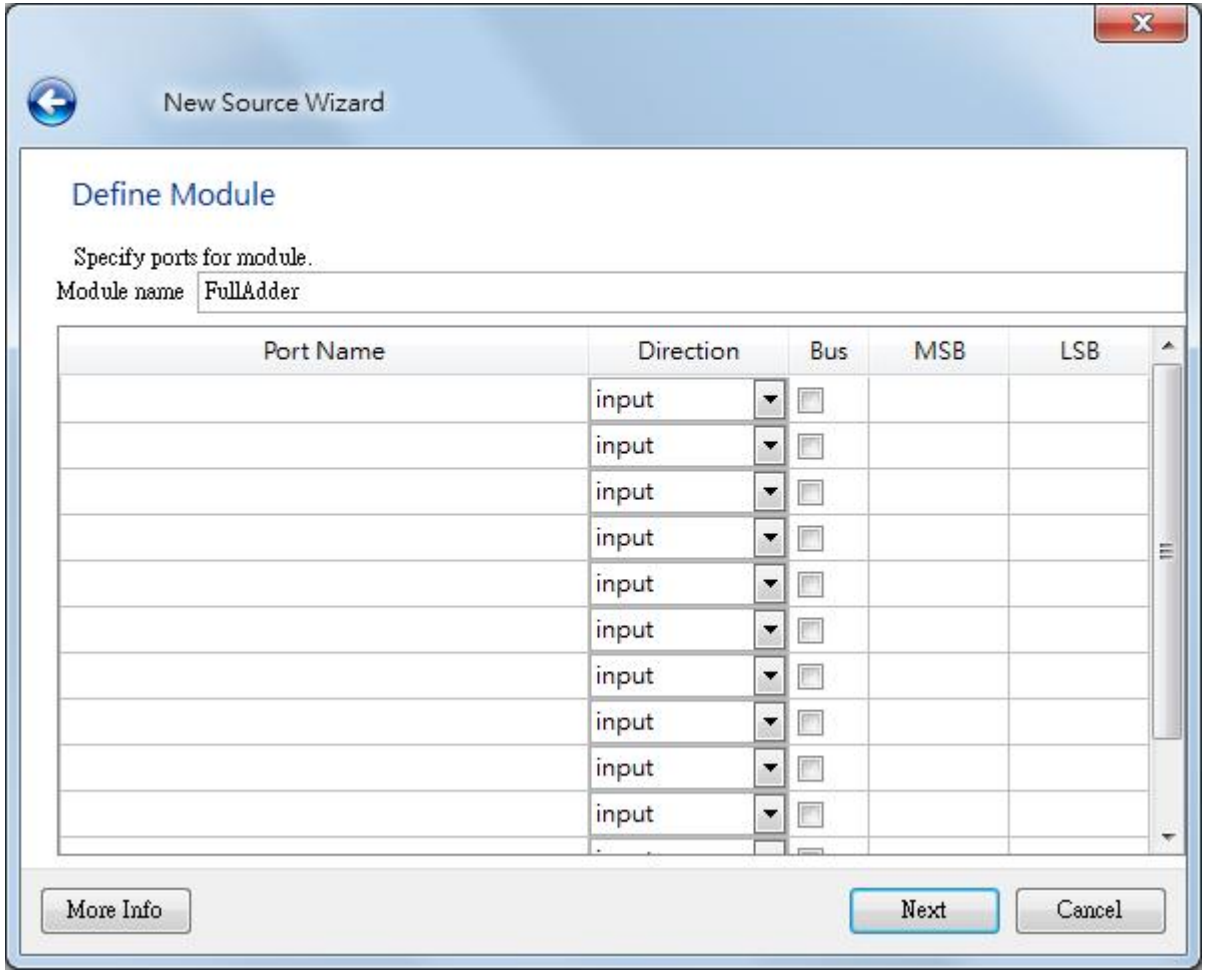


Setting the Source Code Type

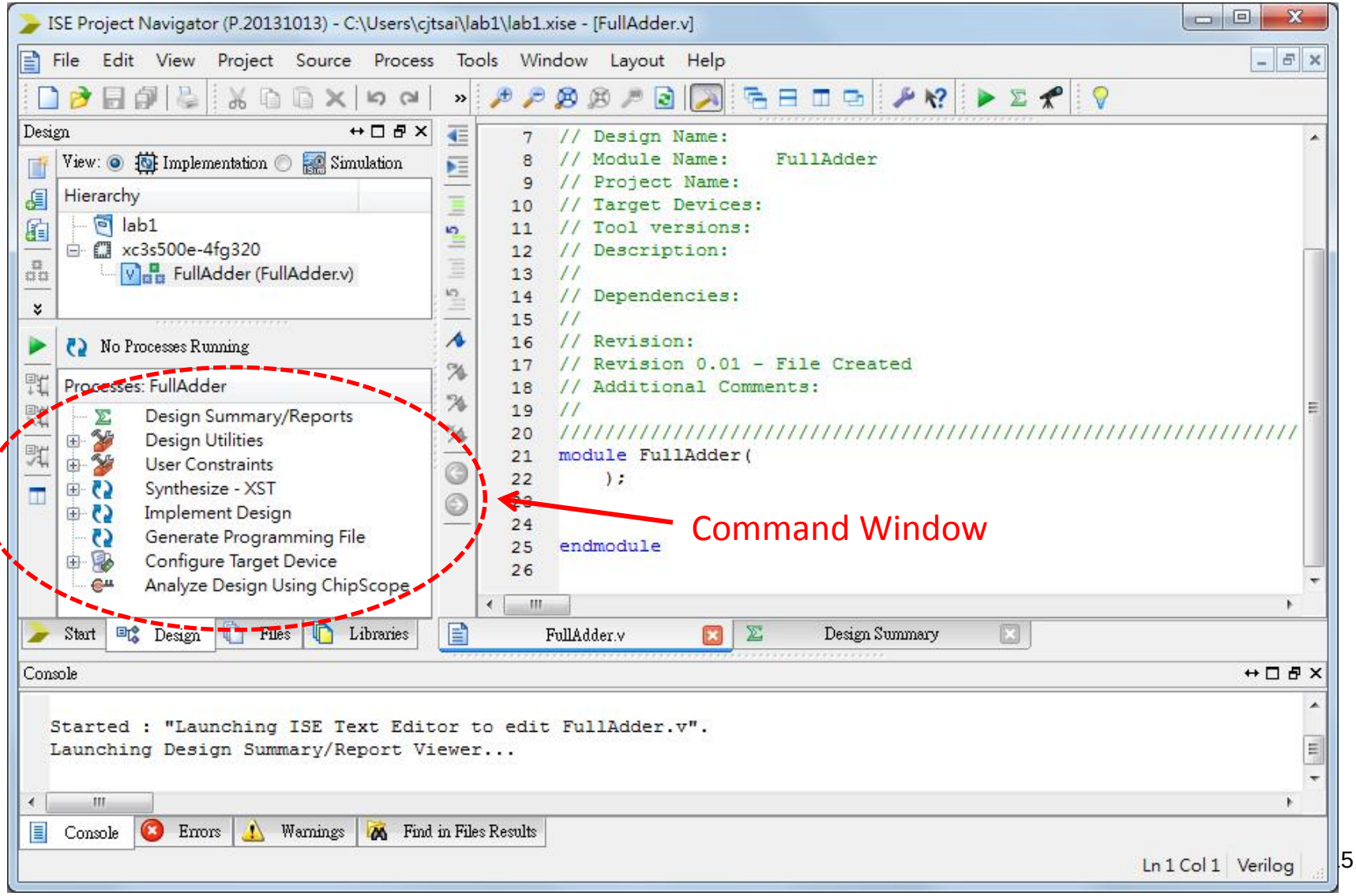


Define Your Input-Output Ports

- You can define your ports here or do it in the HDL code:



Now, You have a Circuit Project



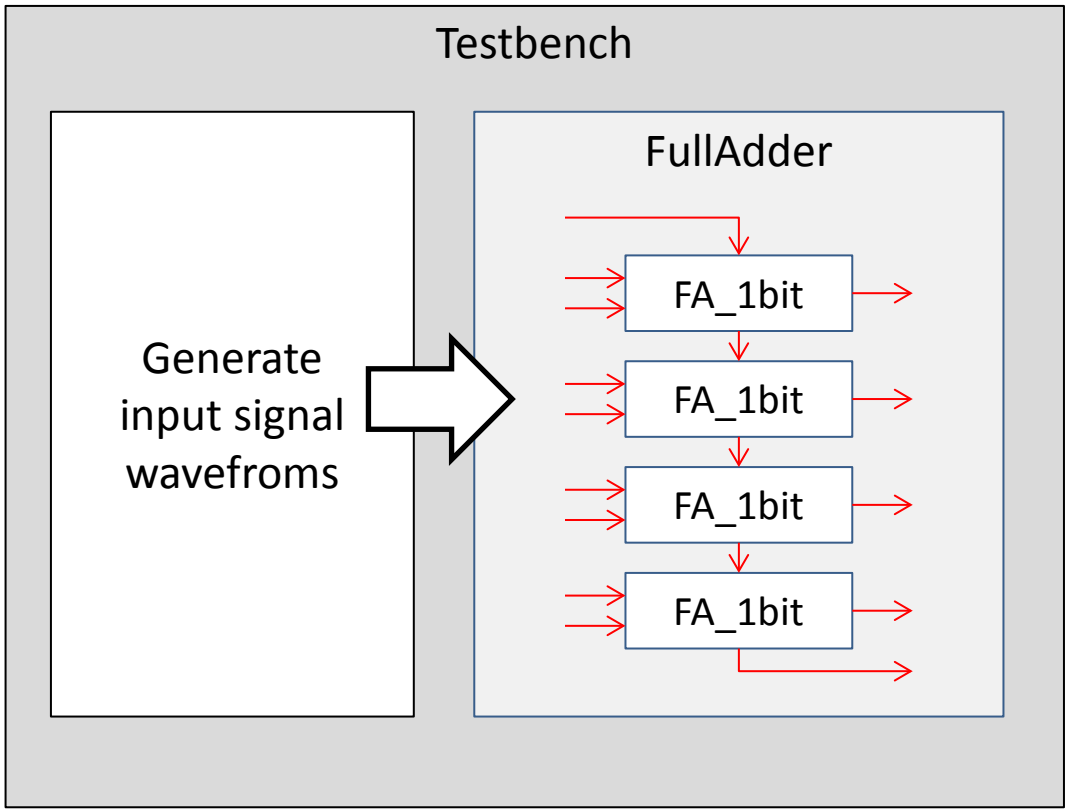
Type in the HDL Source Code

- You can now enter the following source code:

```
// ------------- A four-bit full adder ----------------------------
module FullAdder(A, B, Cin, S, Cout);
  input [3:0] A, B;
  input Cin;
  output [3:0] S;
  output Cout;
  wire [2:0] t;

  FA_1bit FA0(.A(A[0]), .B(B[0]), .Cin(Cin), .S(S[0]), .Cout(t[0]));
  FA_1bit FA1(.A(A[1]), .B(B[1]), .Cin(t[0]), .S(S[1]), .Cout(t[1]));
  FA_1bit FA2(.A(A[2]), .B(B[2]), .Cin(t[1]), .S(S[2]), .Cout(t[2]));
  FA_1bit FA3(.A(A[3]), .B(B[3]), .Cin(t[2]), .S(S[3]), .Cout(Cout));
endmodule

// ----------------- A 1-bit full adder ----------------------------
module FA_1bit(A, B, Cin, S, Cout);
input A, B, Cin;
output S, Cout;

assign S = Cin ^ A ^ B;
assign Cout = (A & B) | (Cin & B) | (Cin & A);
endmodule
```

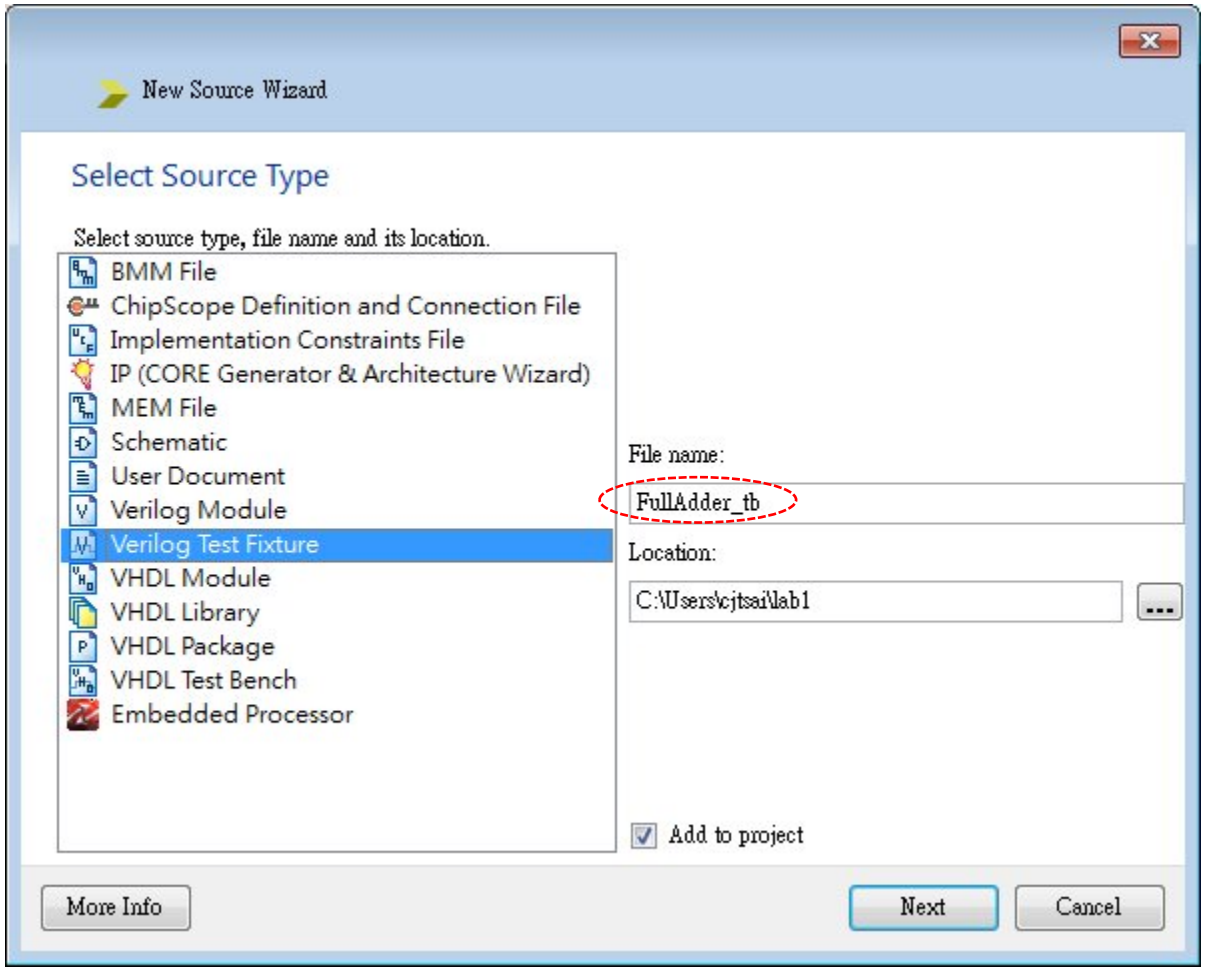
TestBench Design

- You must create a testbench to generate input signals that can feed into your circuit module, such that you can analyze the output to verify its correctness



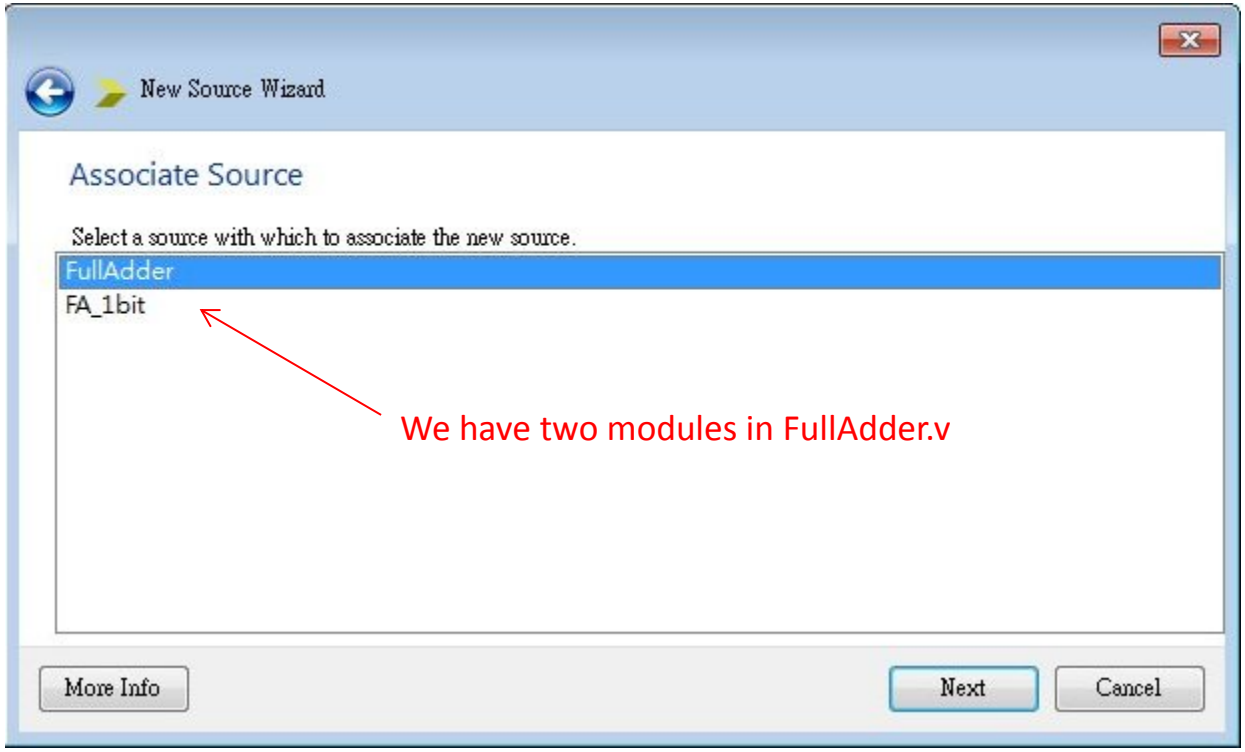
Create the Testbench Source Code

- Click “New Source” button again:



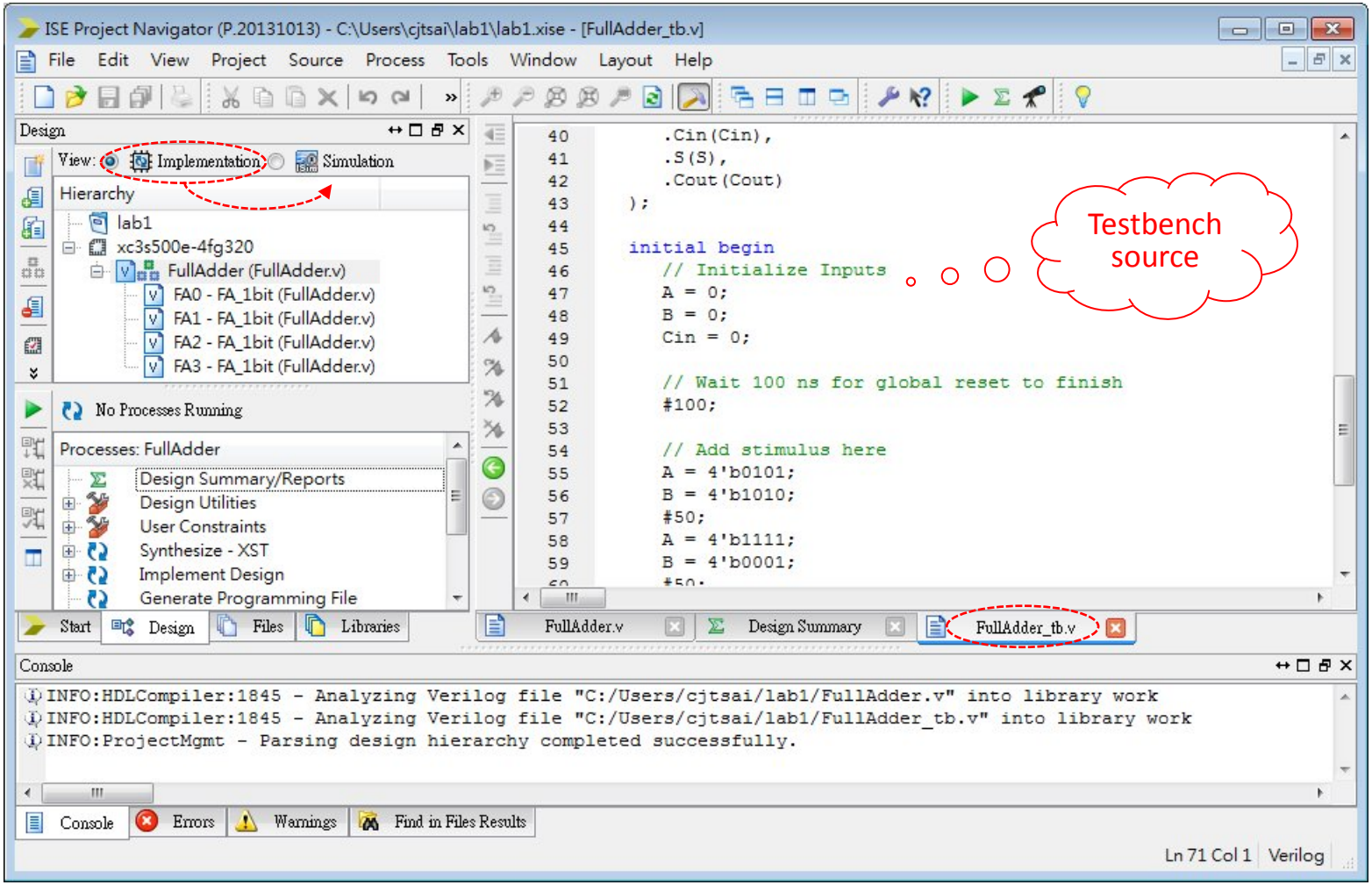
Select the Target Circuit Component

- In this design, you want to verify the correctness of the 4-bit FullAdder:



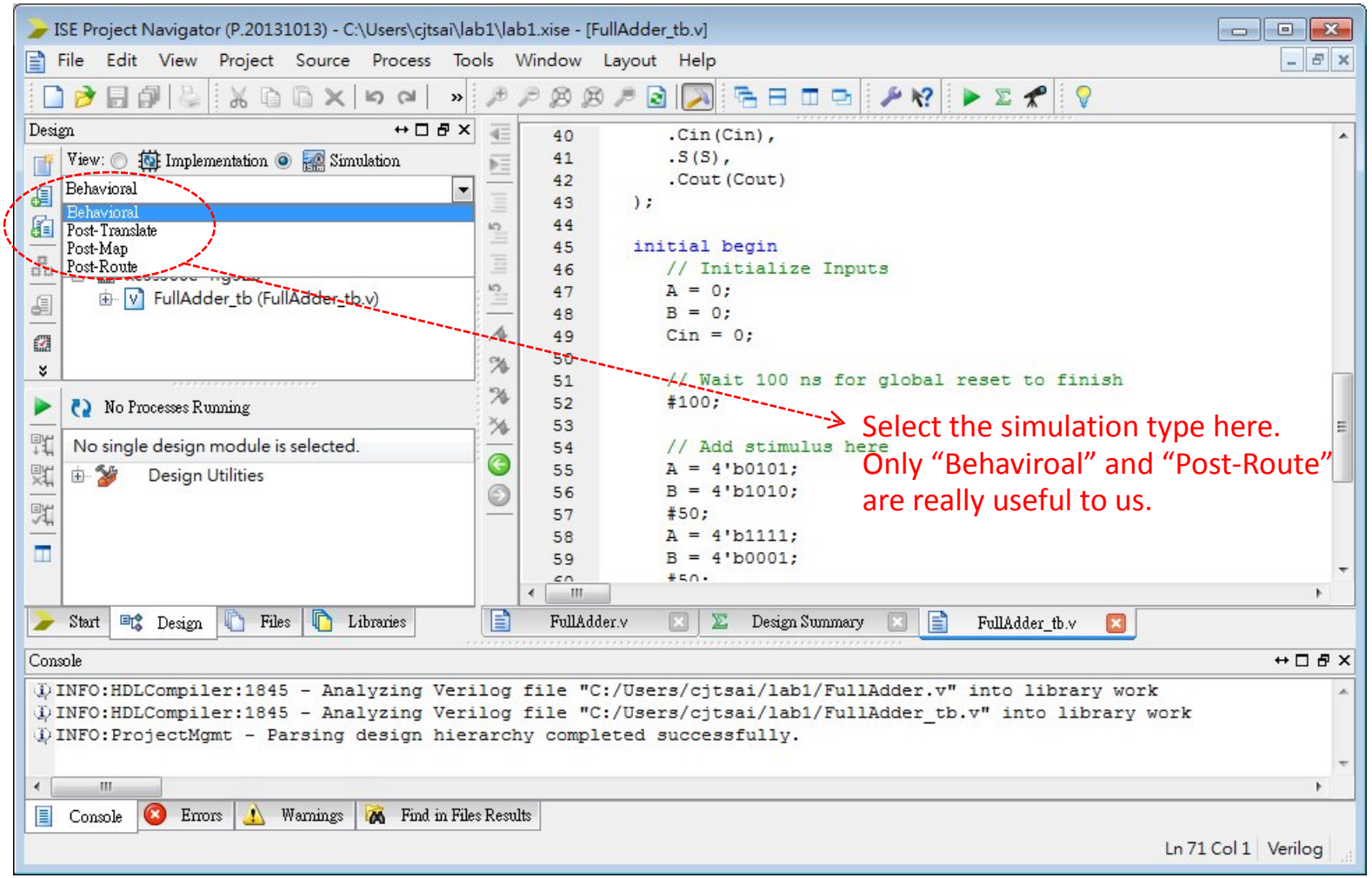
TestBench Template

- ISE automatically generates a testbench template:



ISE Simulation View

- Switch to simulation view to use the ISim simulator



Add Simulation Input Generator Code

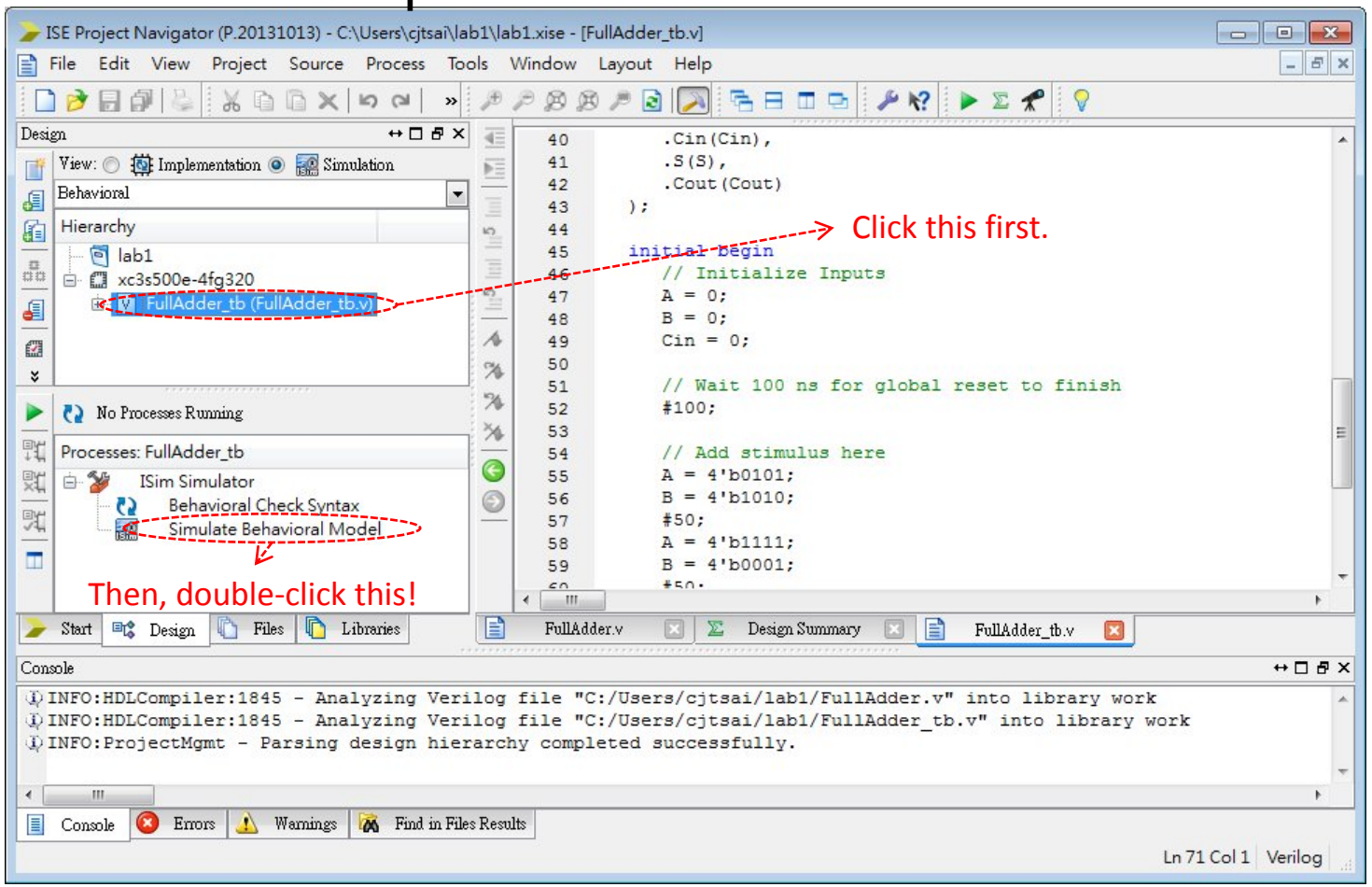
- The template created by ISE only contains an empty testbench, you must add your own test patterns:

```
module FullAdder_tb;

  // ... Some template code here ... //

  initial begin
    // Initialize Inputs
    A = 0;
    B = 0;
    Cin = 0;

    // Wait 100 ns for global reset to finish
    #100;

    // Add stimulus here
    A = 4'b0101;
    B = 4'b1010;
    #50;
    A = 4'b1111;
    B = 4'b0001;
    #50;
    A = 4'b0000;
    B = 4'b1111;
    Cin = 1'b1;
    #50;
    A = 4'b0110;
    B = 4'b0001;
  end
endmodule
```

What you add.

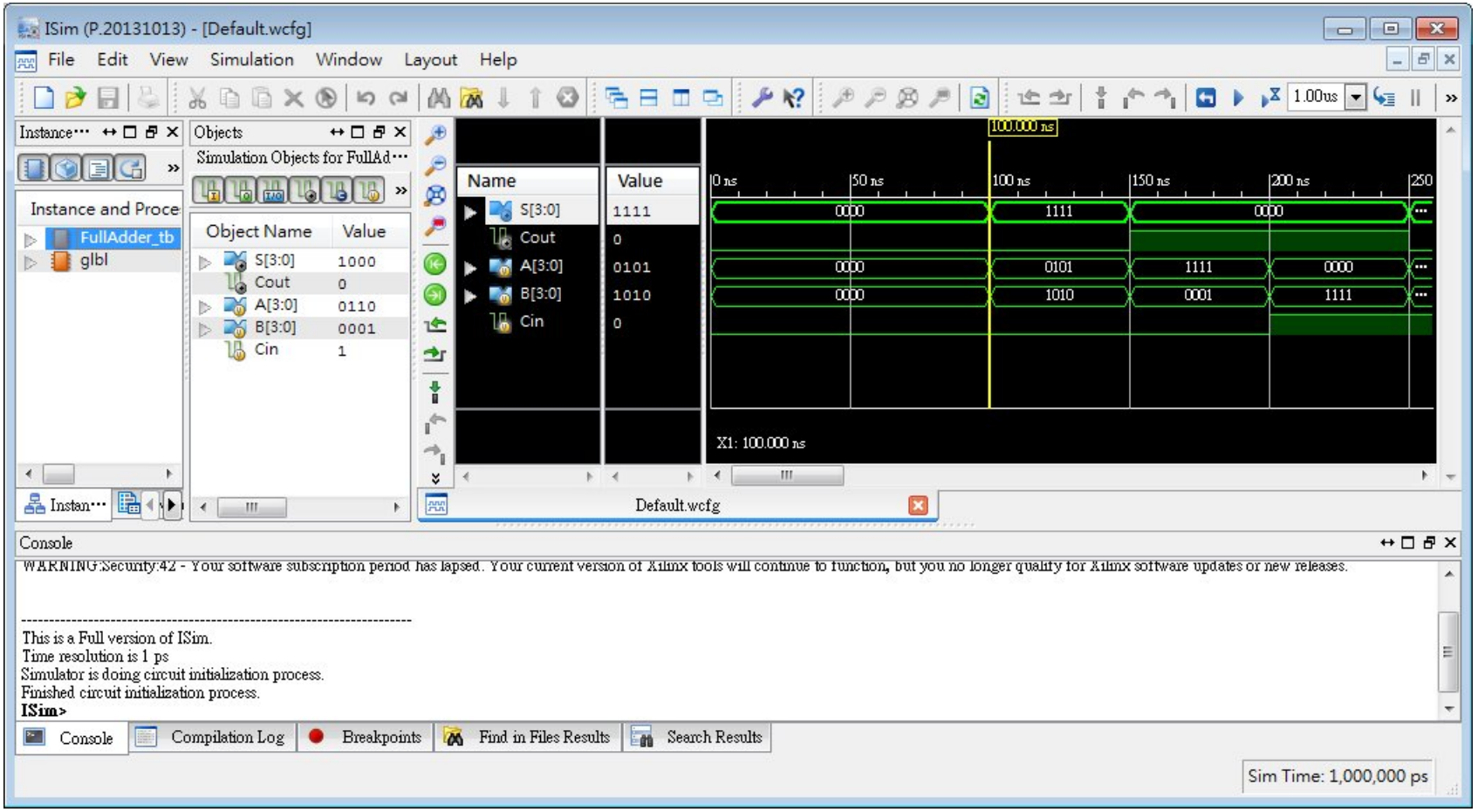
Compile the Design for Simulation

- You must click FullAdder_tb in the Hierarchy Window to show the ISim processes:



ISim Simulator Window

- After double-clicking “Simulate Behavior Model” you get:



Conclusion

- Now, you have basic concept about the EDA tool we will use throughout the entire course
- Please read the references carefully and go to the lab frequently to practice circuit design; there are just too many details to learn that we cannot possibly cover everything in our lectures or lab notes

Reference

- Xilinx, *ISE In-Depth Tutorial, UG695 (v14.1)*, April 24, 2012 (Note: this document can be downloaded from the E3 website of this class)
- M. Mano and M. Ciletti, *Digital Design with an Introduction to the Verilog HDL*, 5th Ed., Pearson, 2013.
- P. J. Ashenden, *Digital Design: An Embedded Systems Approach Using Verilog*, Morgan Kaufmann, 2008.
- P. P. Chu, *FPGA Prototyping by Verilog Examples: Xilinx Spartan-3 version*, J. Wiley & Sons, 2008.