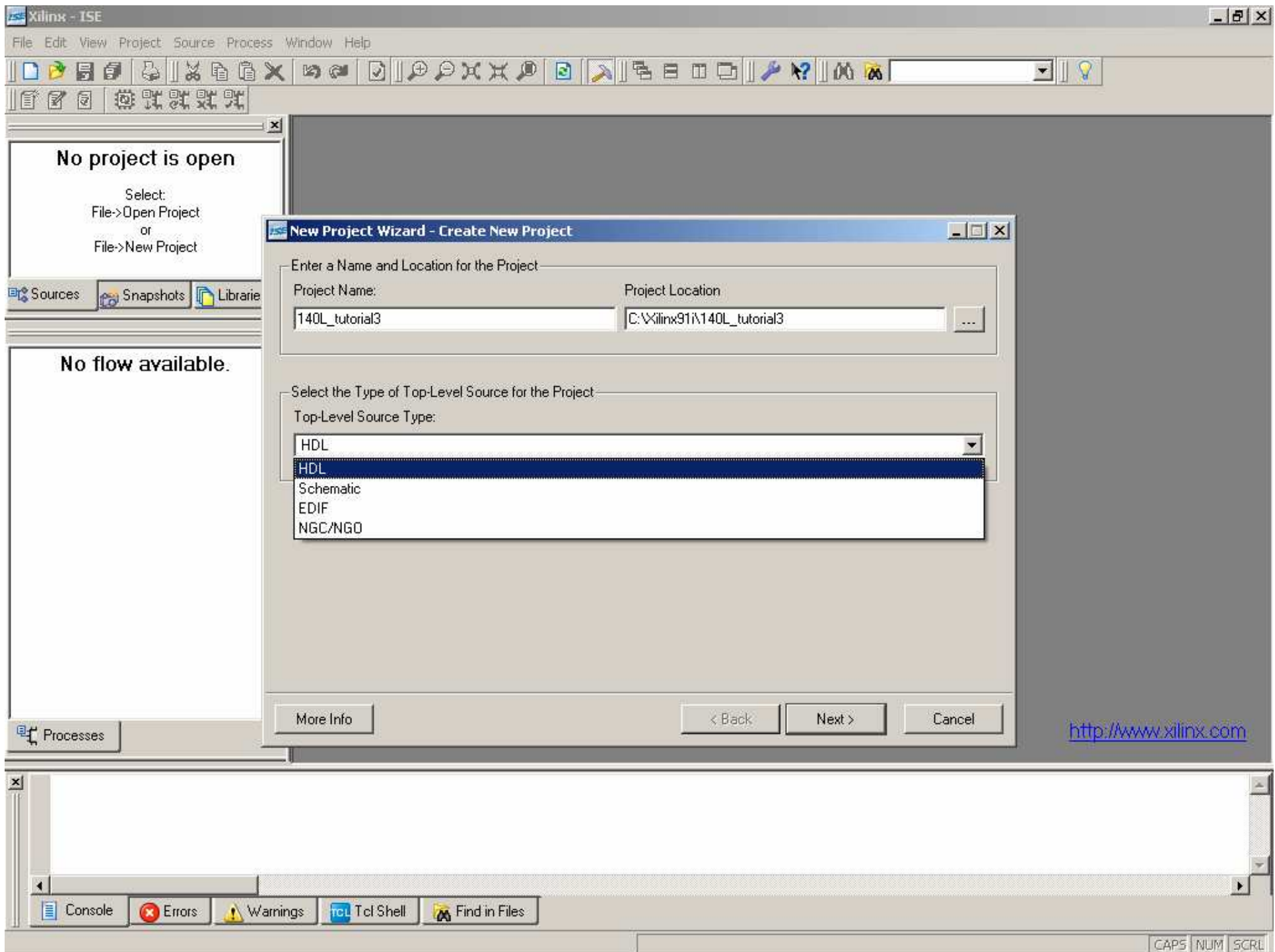
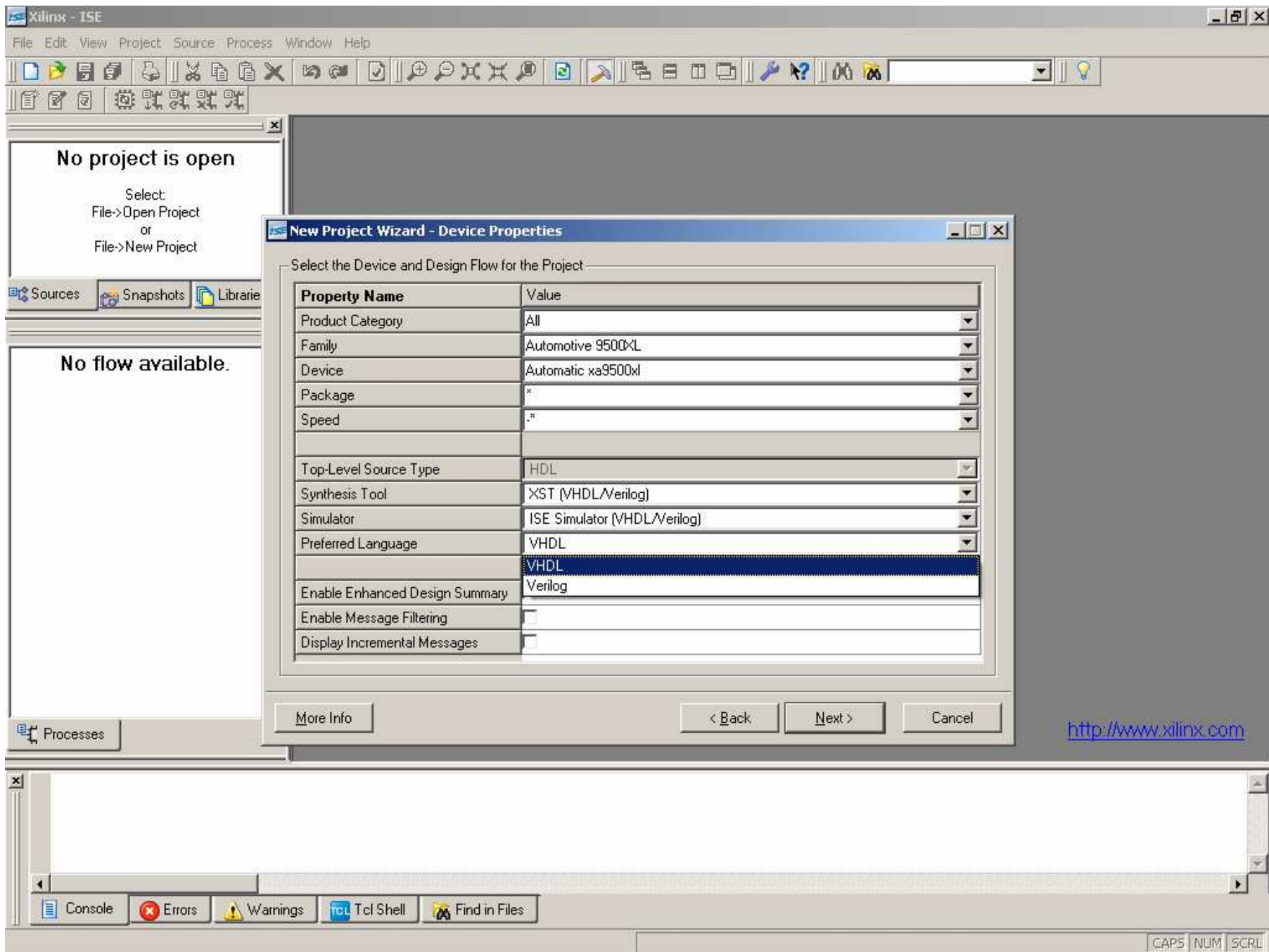
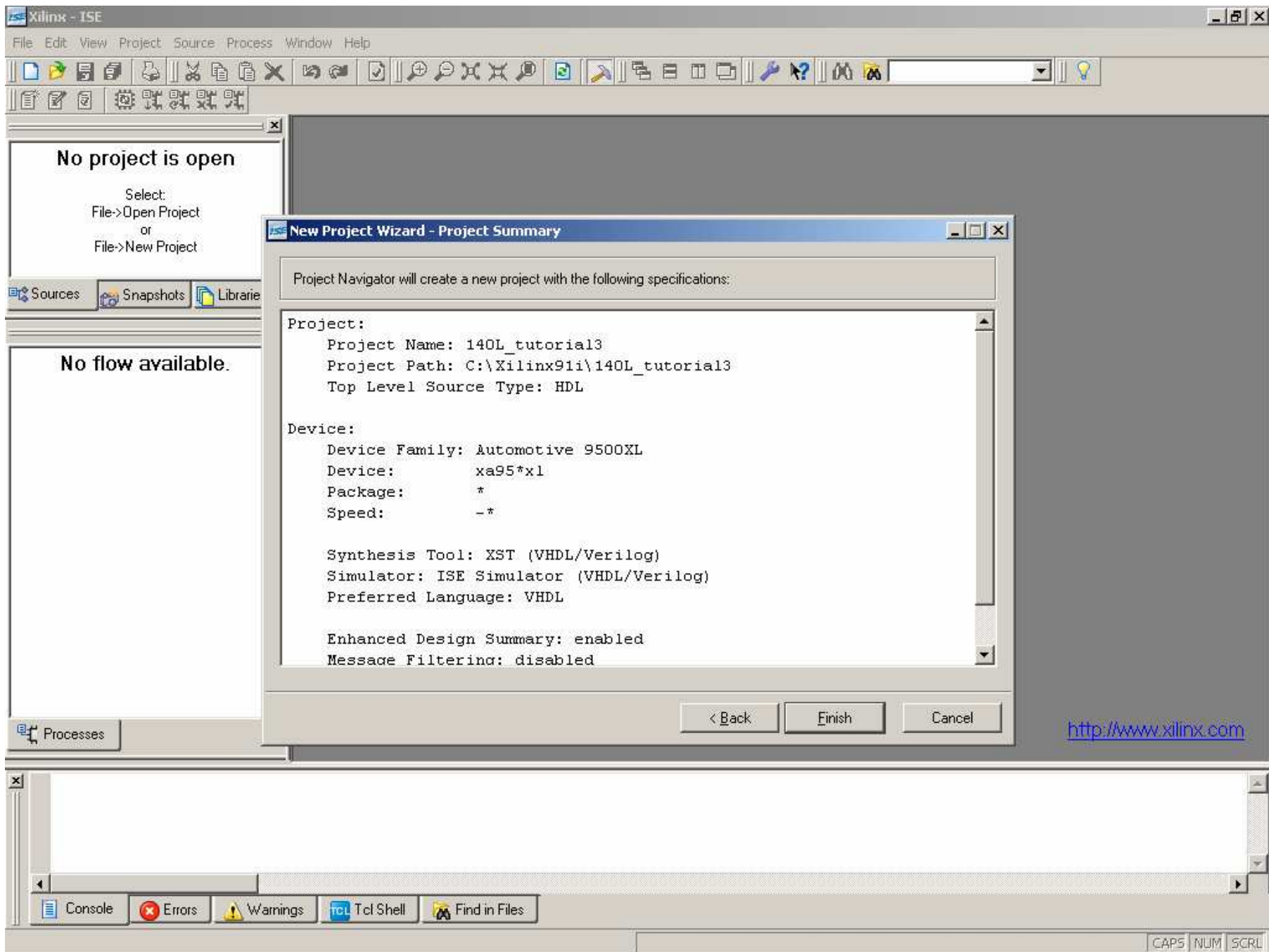




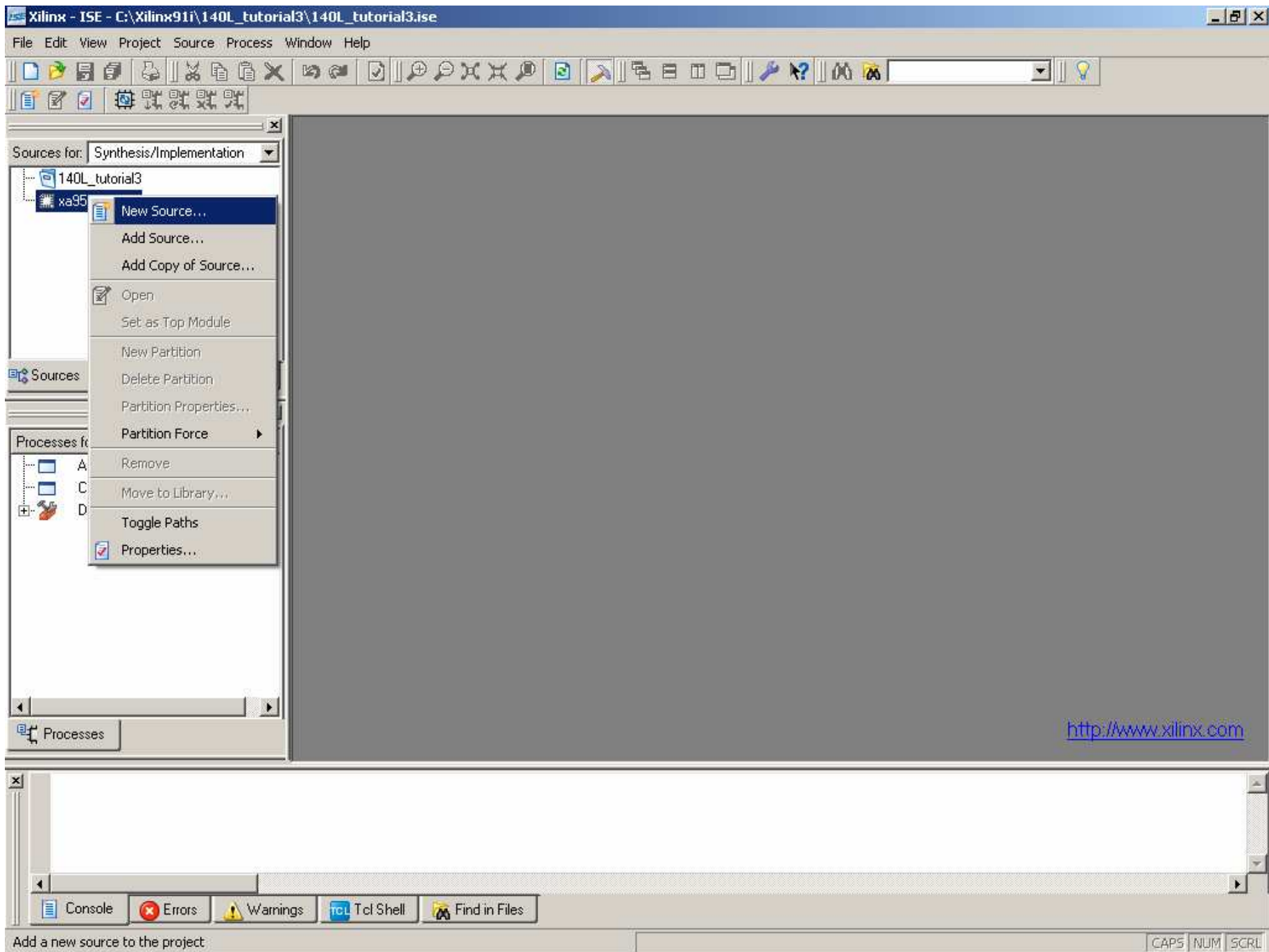
Start a new project and select the “HDL” as the top-level source type and click Next.



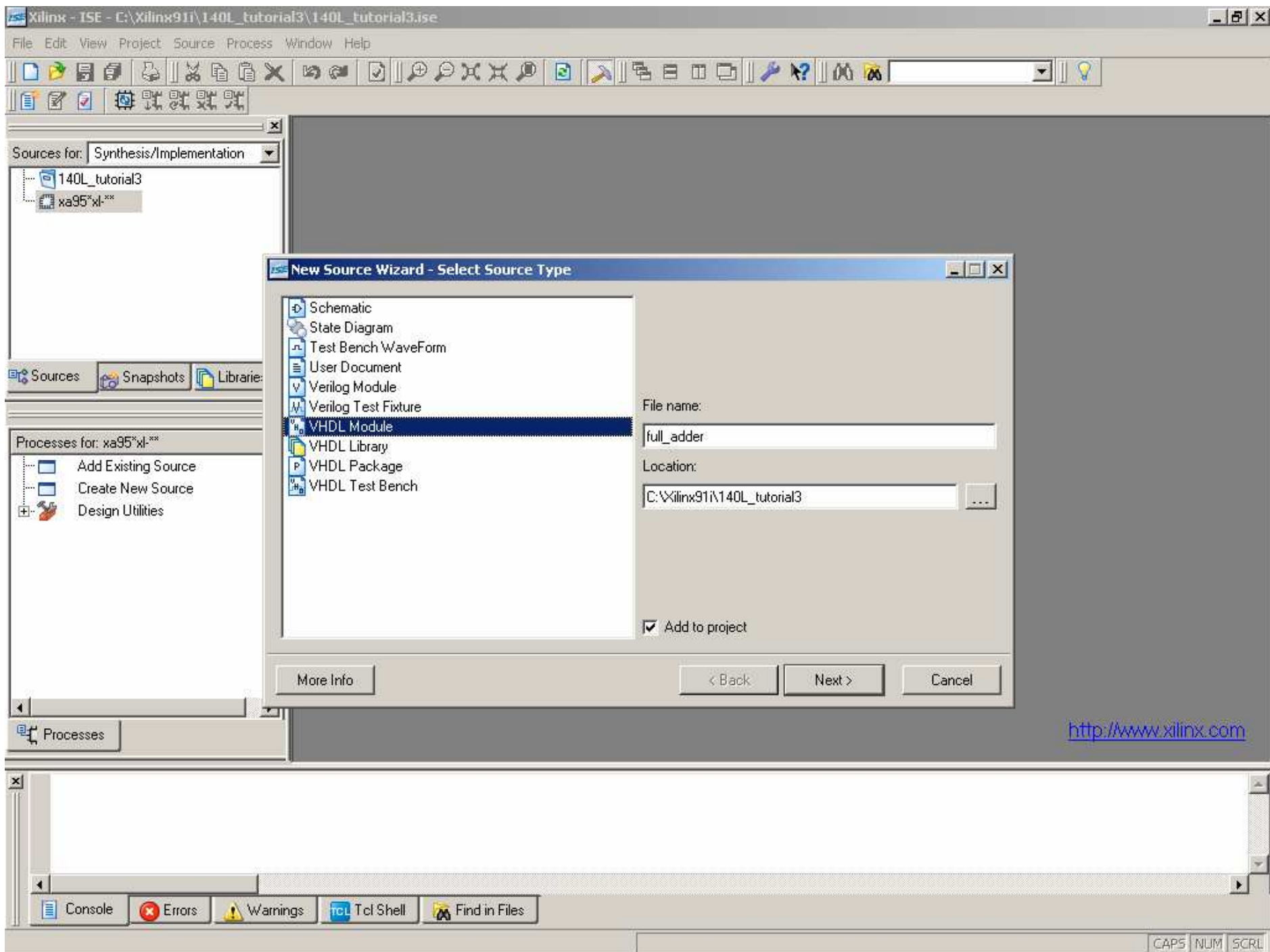
Select "VHDL" as the Preferred Language and click Next.



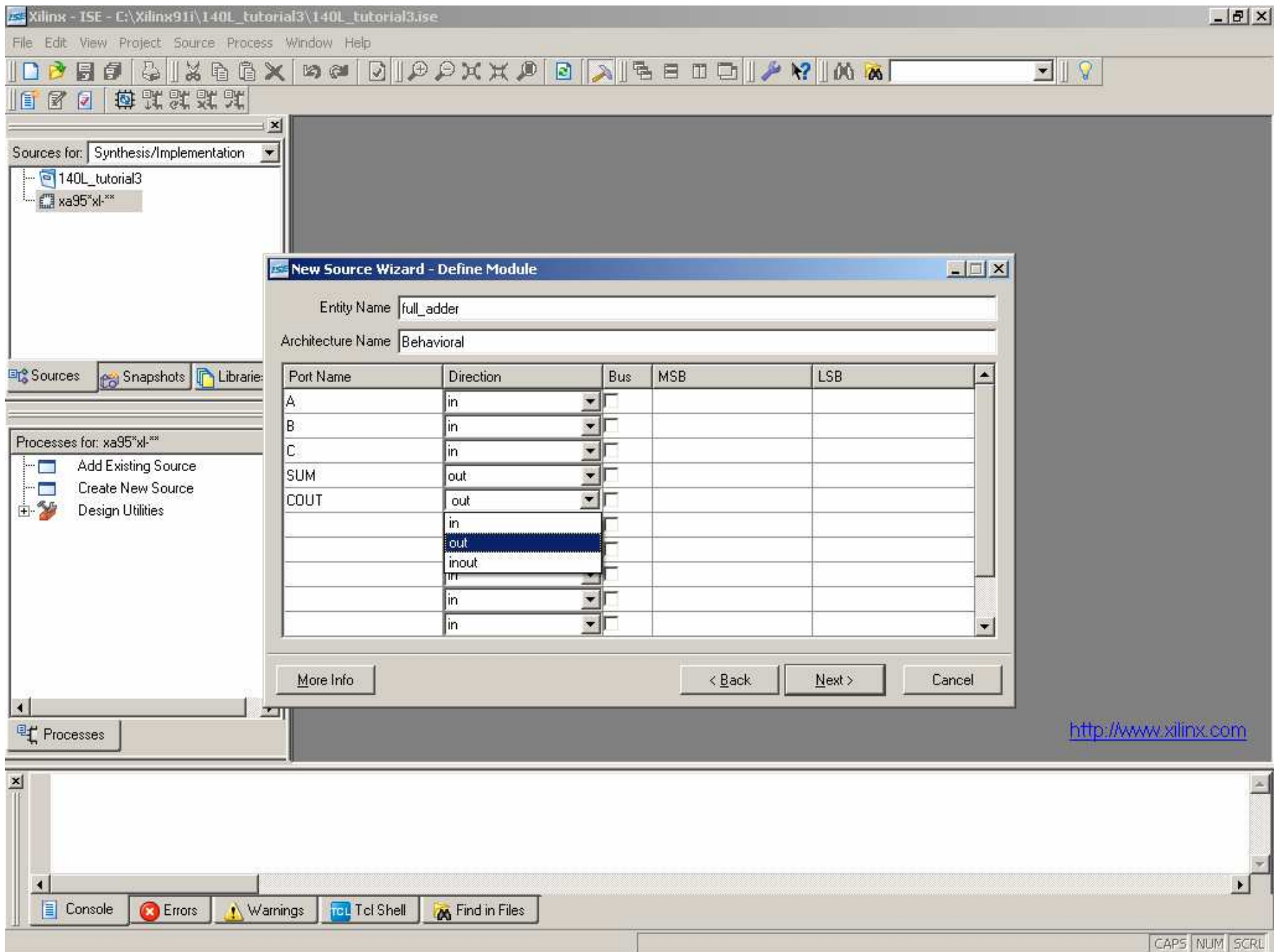
After clicking Next for the following two screens, you come to a summary screen. Click Finish.



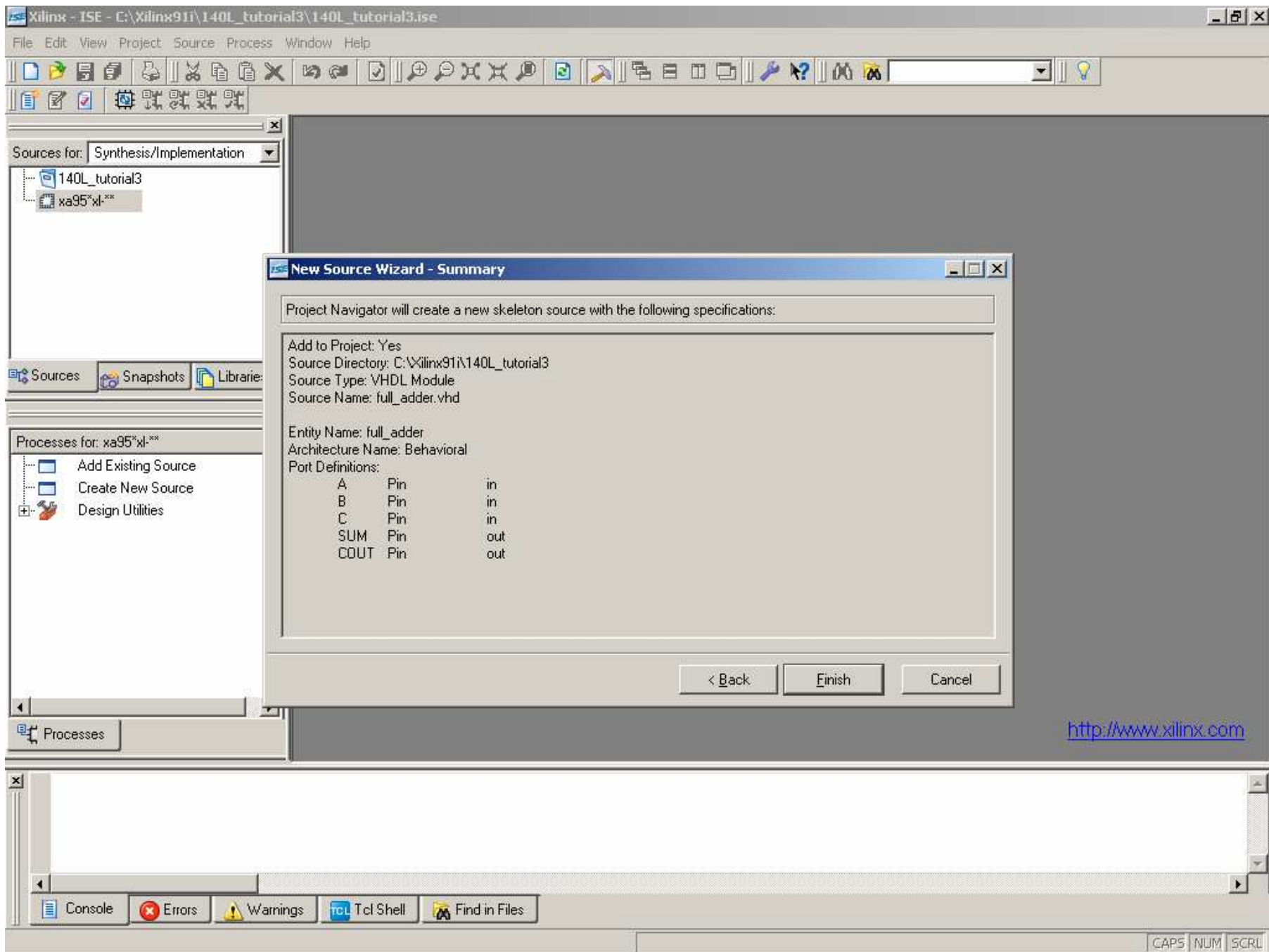
Now, add a new source.



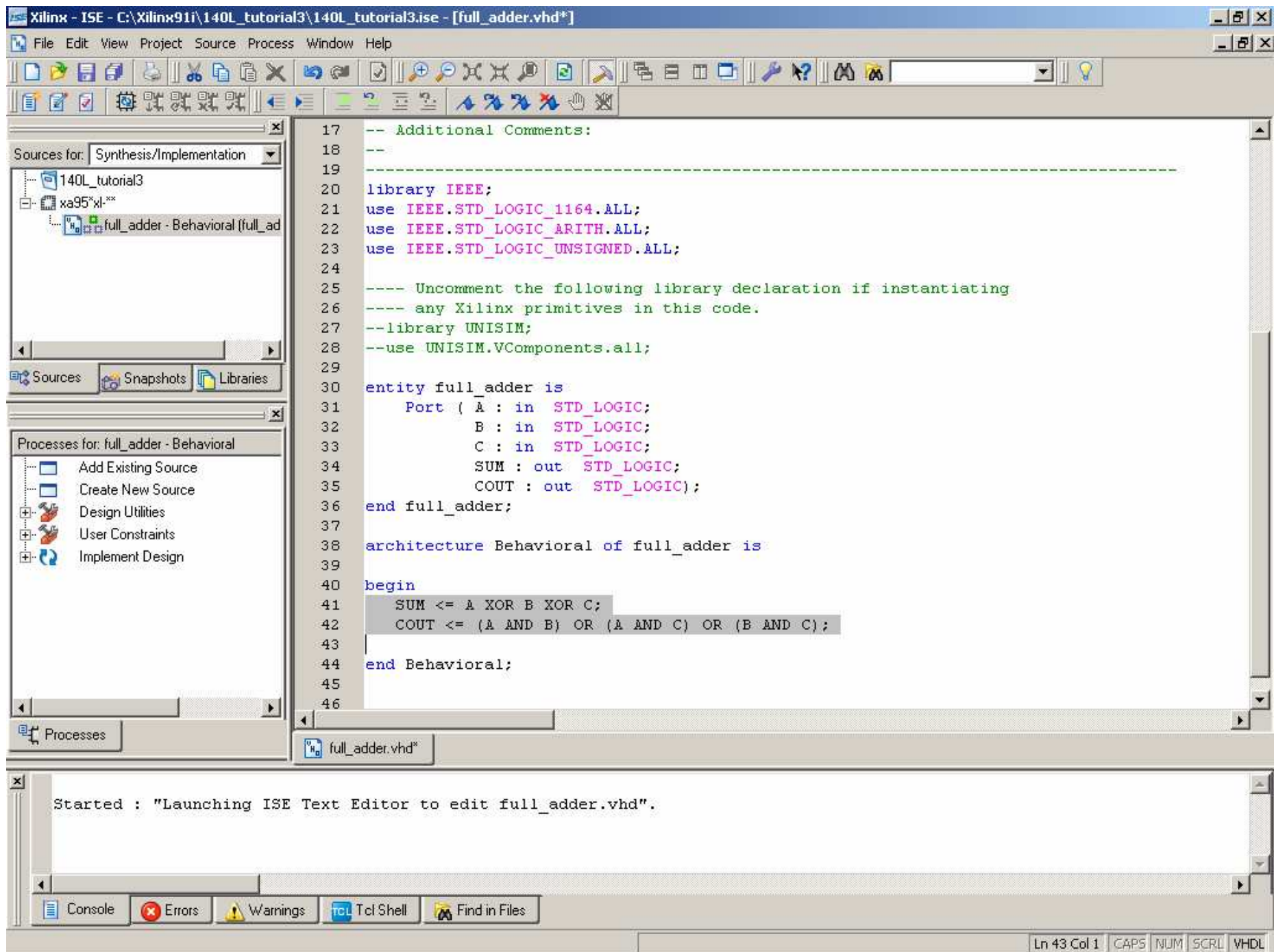
Be sure to select the “VHDL Module” and enter a file name “full_adder”.



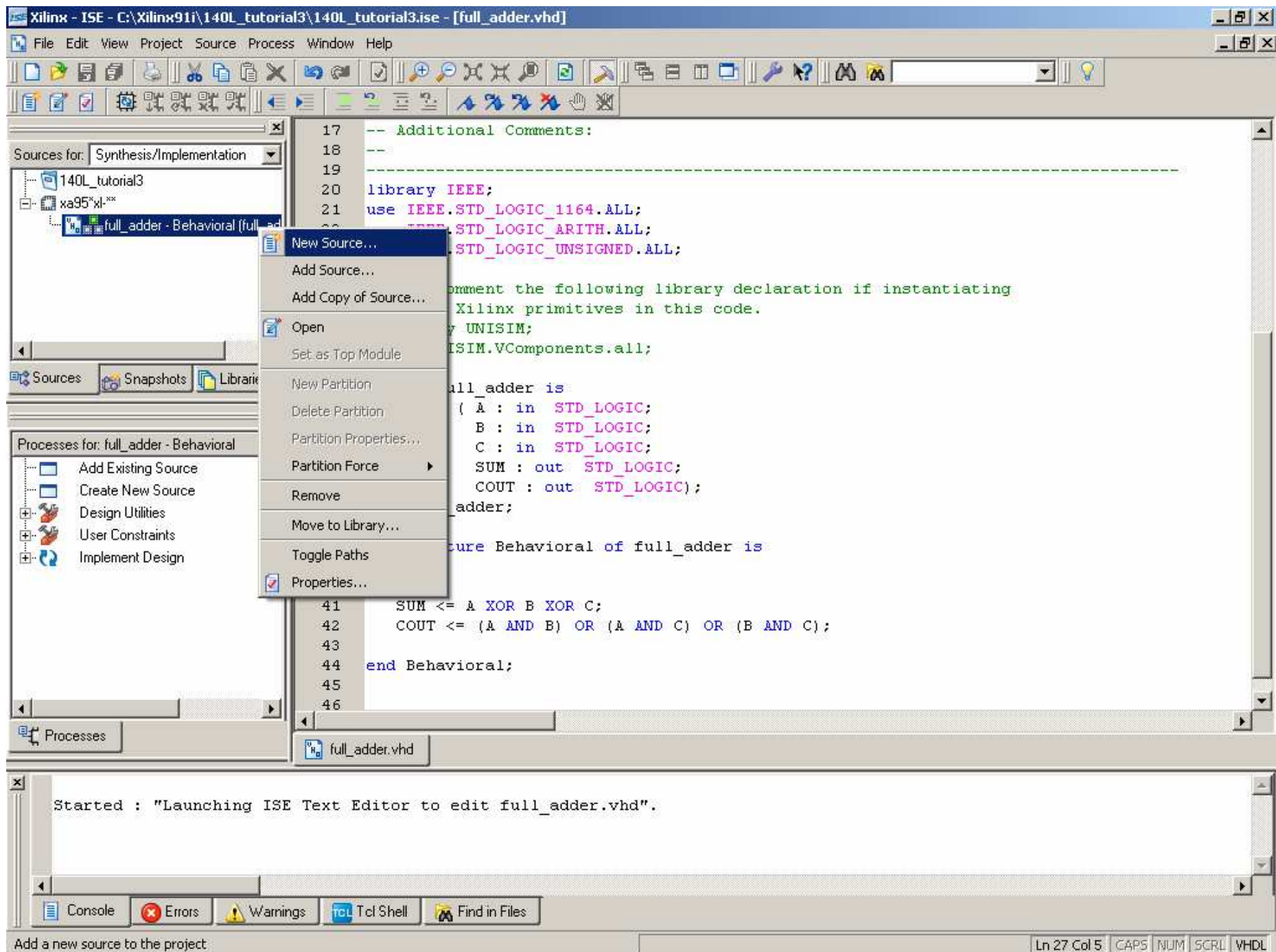
Enter the Port Name (A, B, C, SUM, COUT). Select the “out” as the direction for SUM and COUT. Click Next.



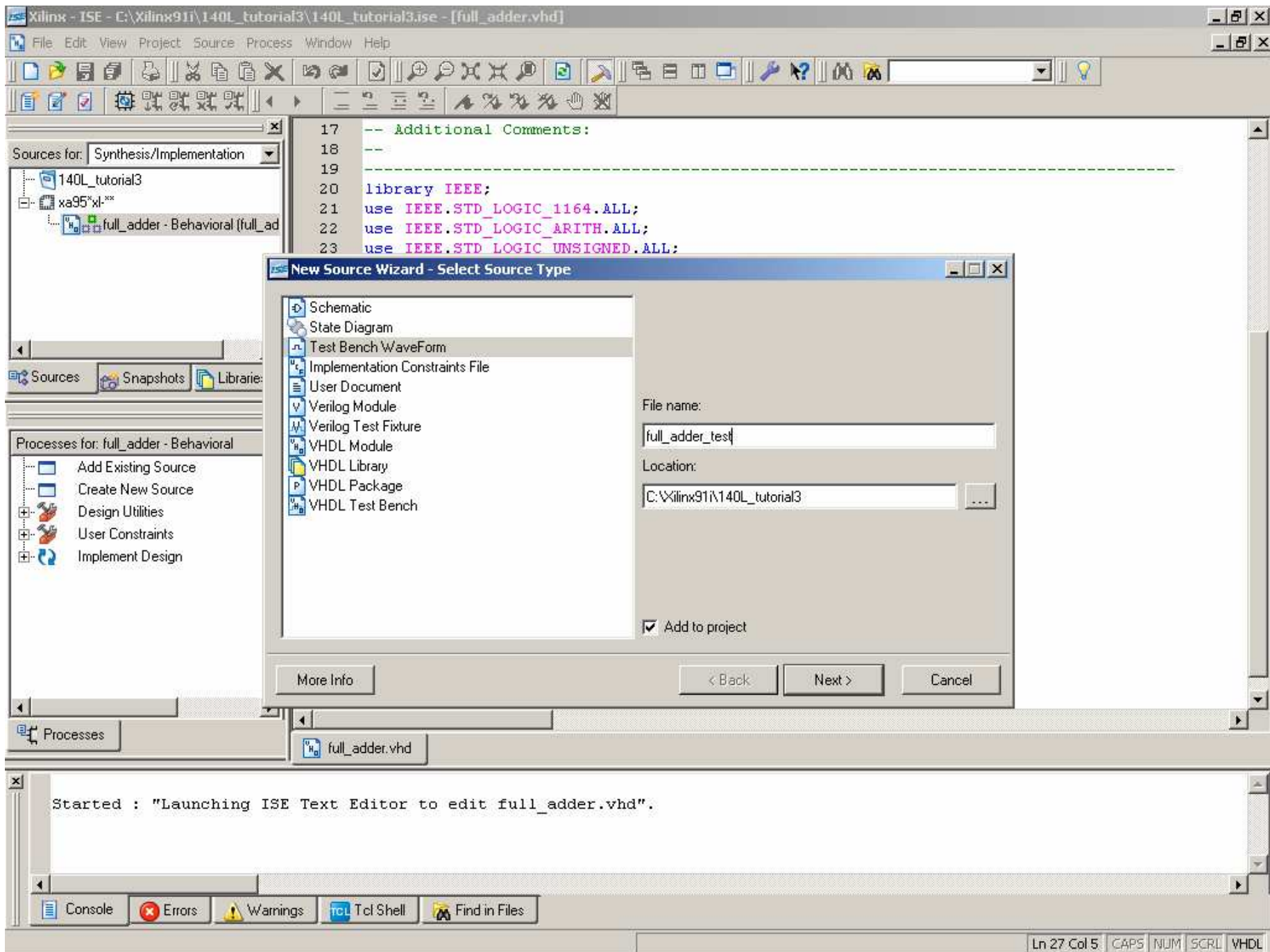
A quick summary page is shown. Click Finish.



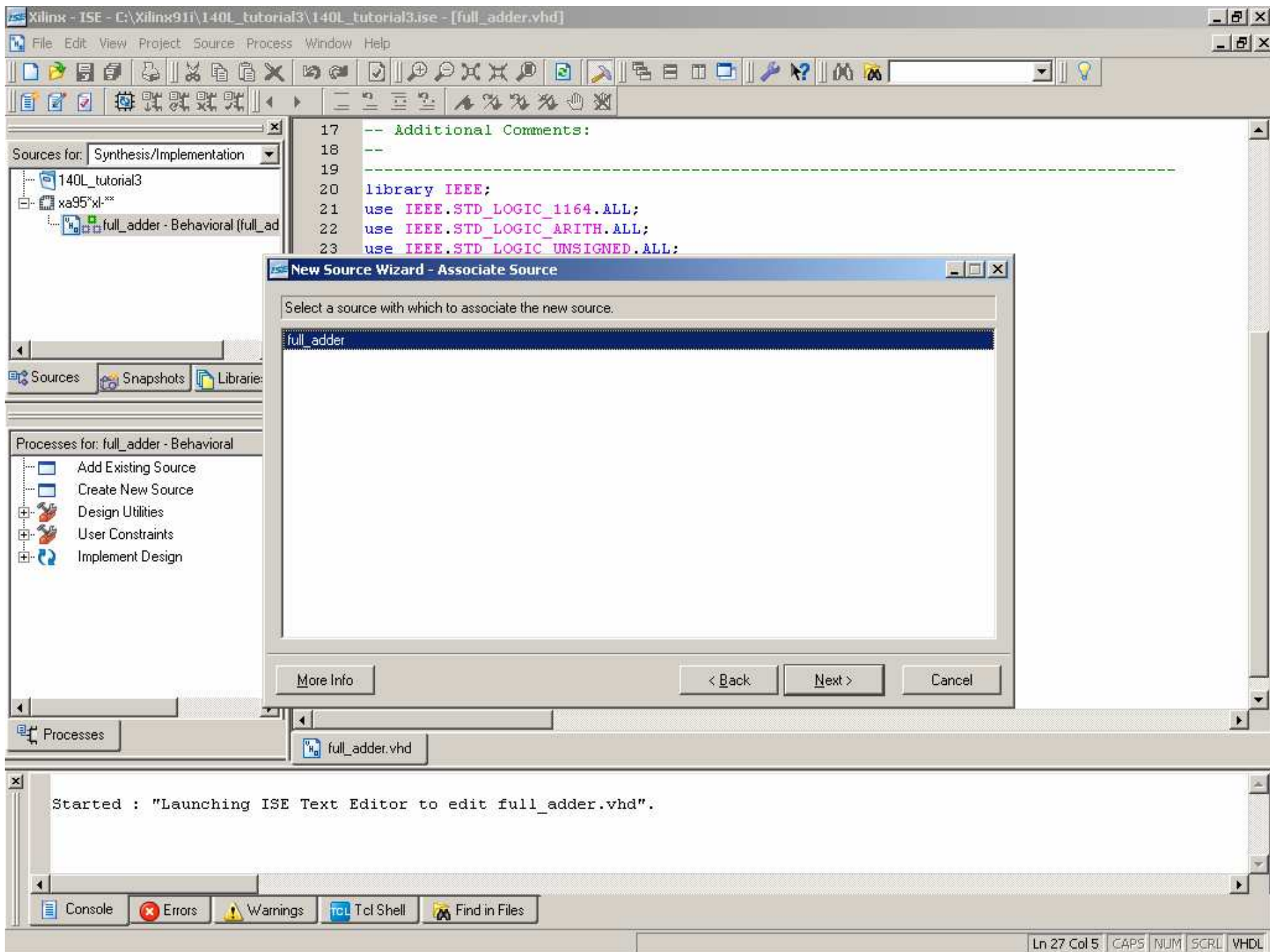
Scroll down to architecture section of the code. Enter the code highlighted (SUM and COUT logic). Save your work!



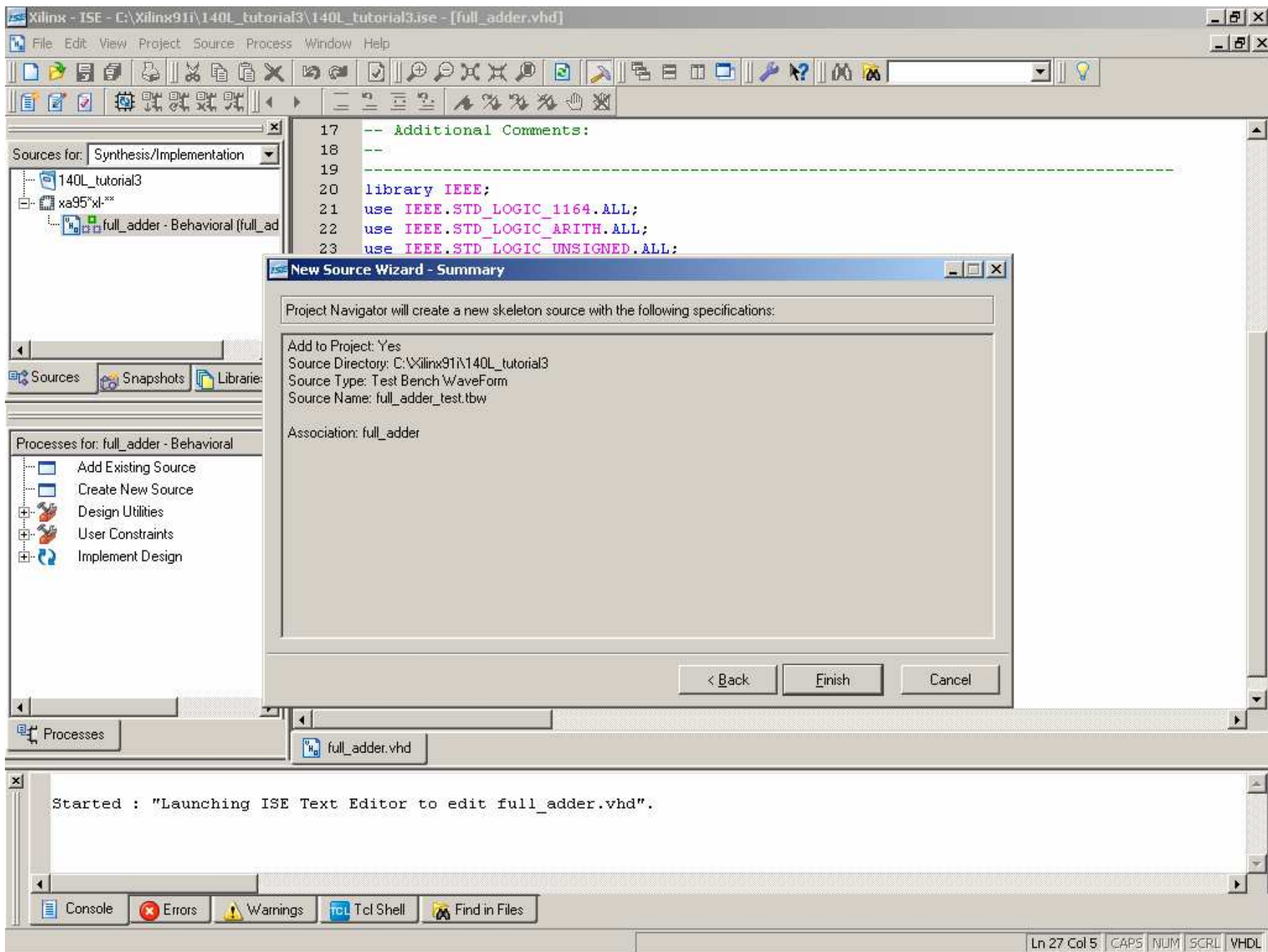
Add a test-bench for this full-adder.



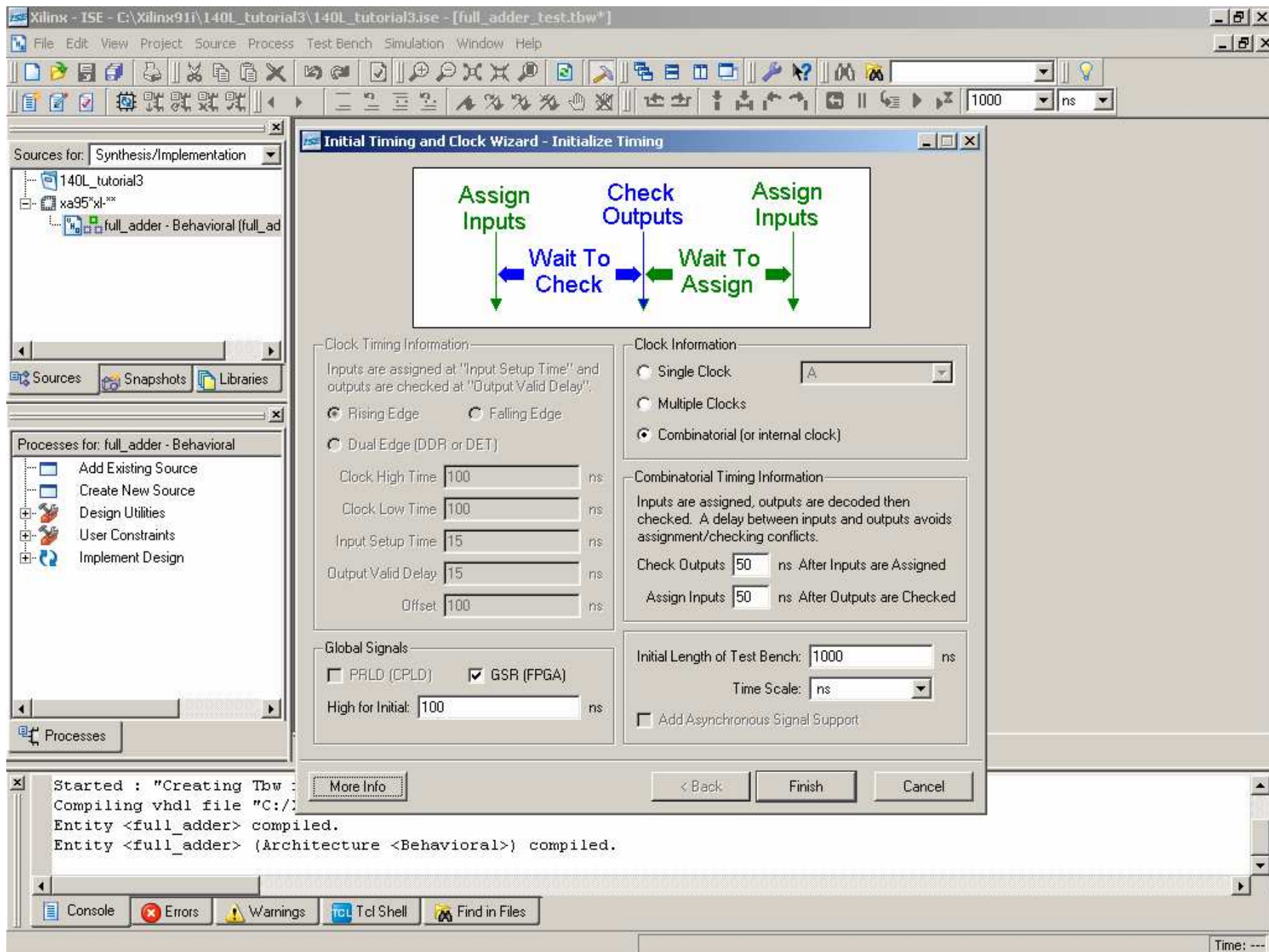
After you select the “Test Bench Waveform” and enter the file name “full_adder_test”, click Next.



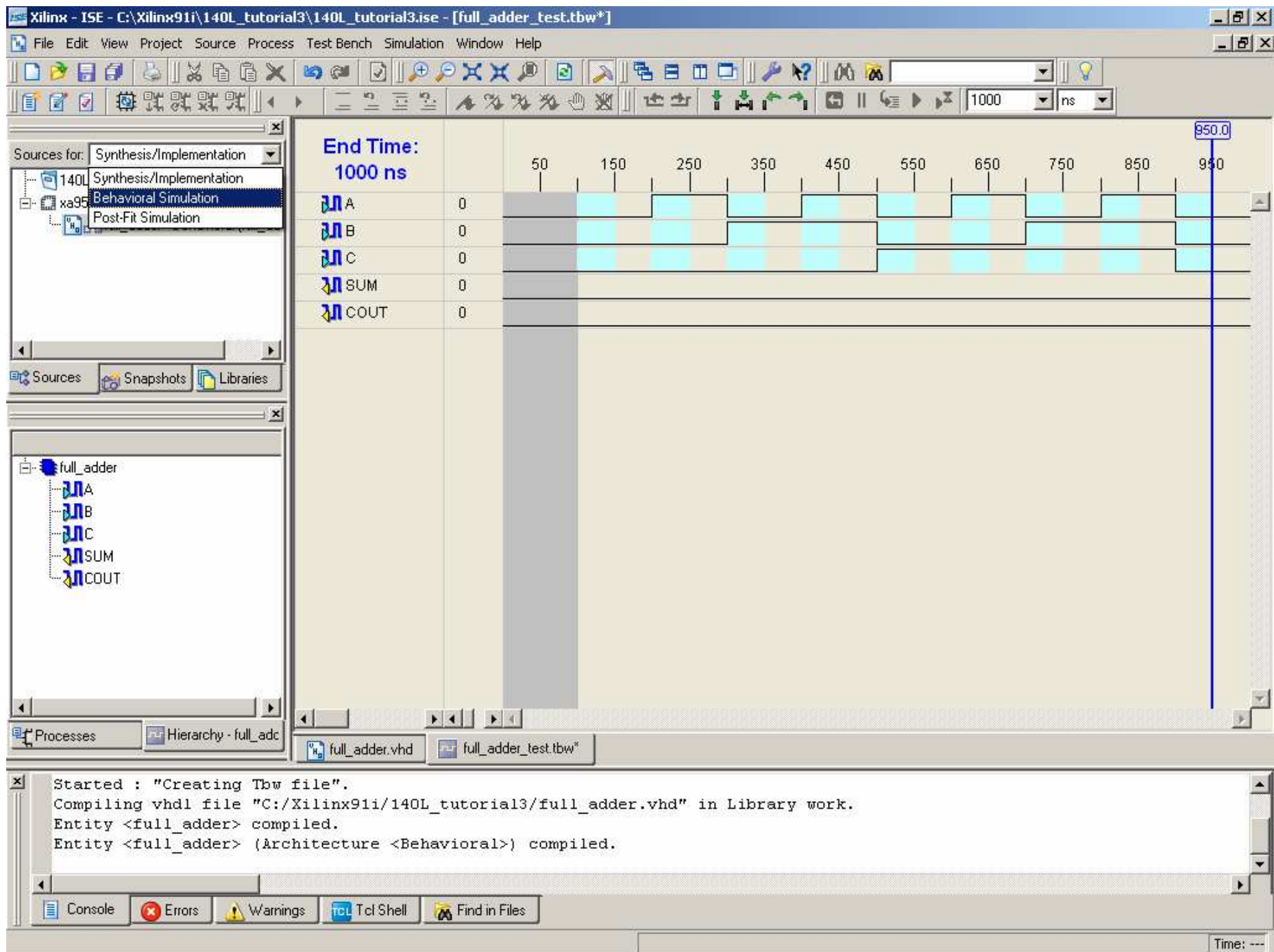
Select the module you want to test (only one here) and click Next.



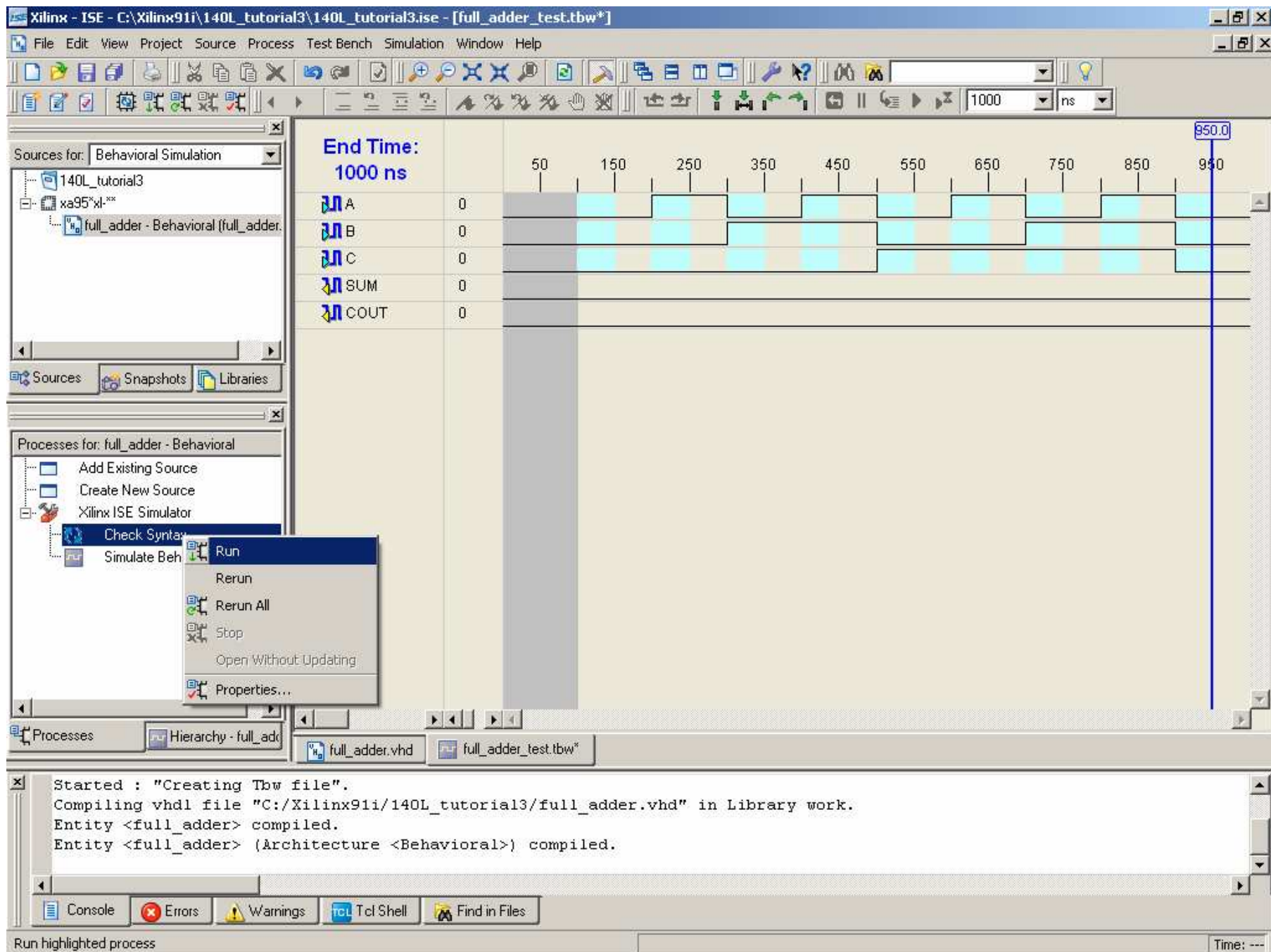
A quick summary page is shown. Click Finish.



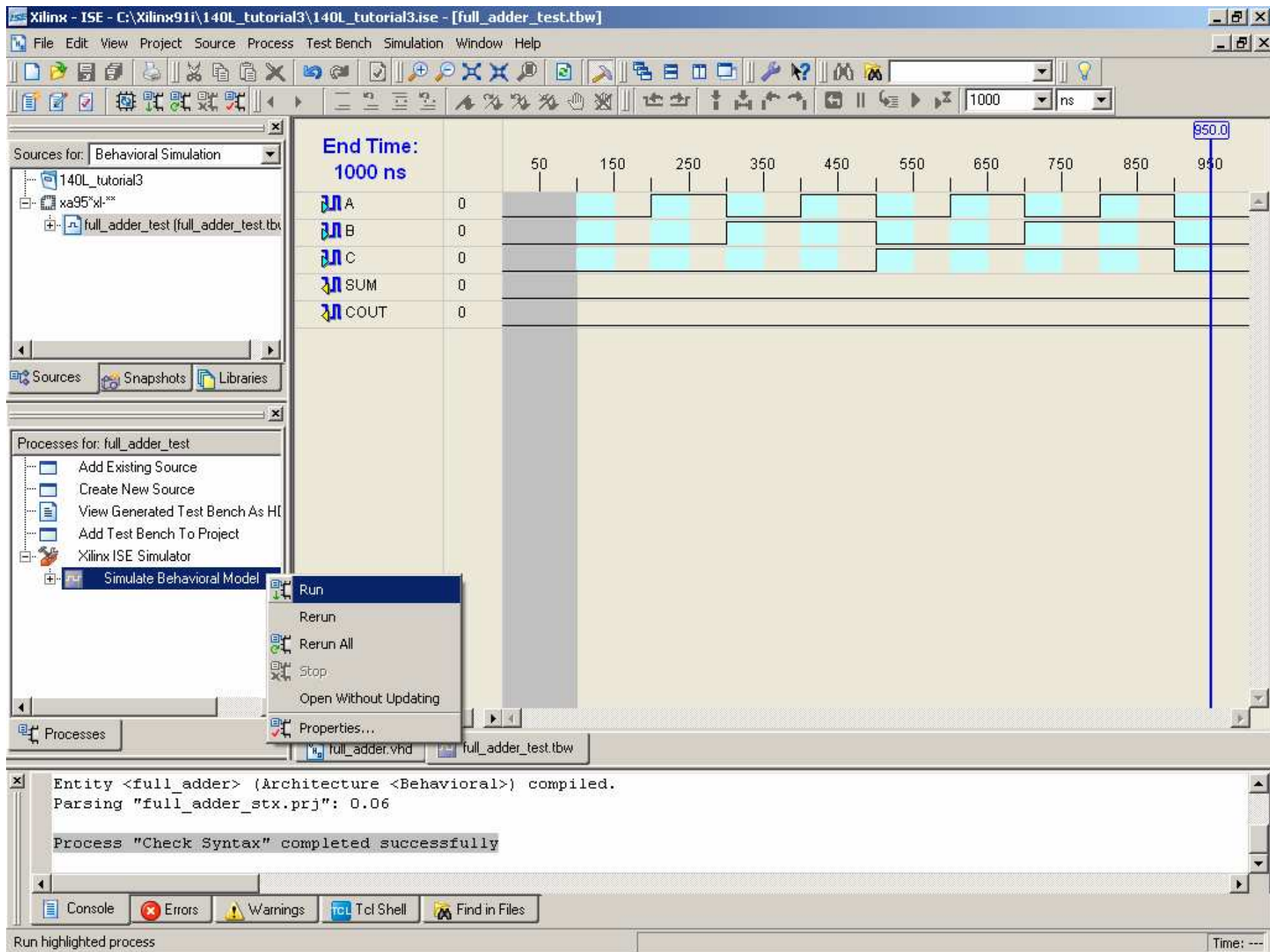
Select the "Combinational (or internal clock)" option in the Clock Information and click Finish.



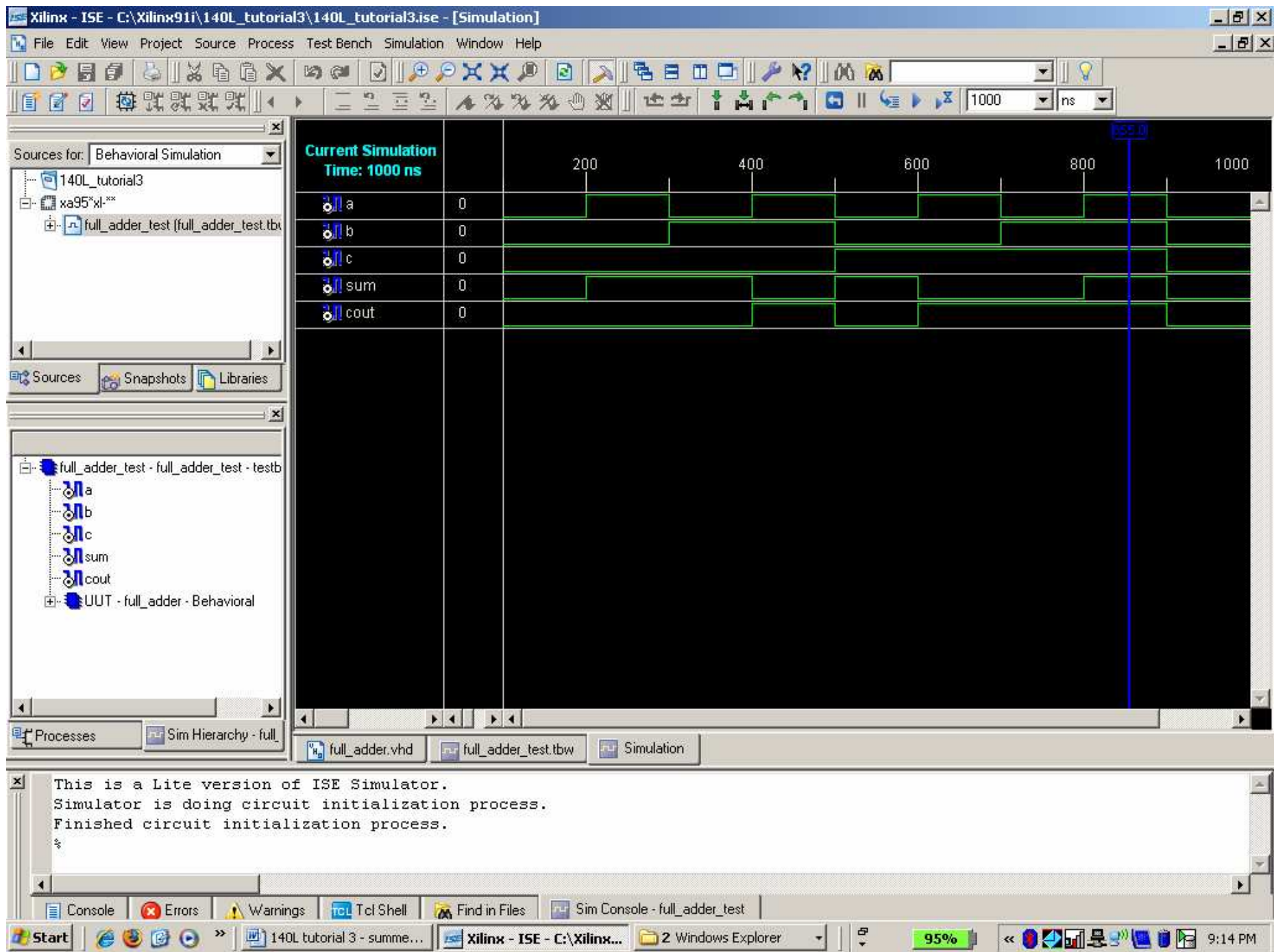
Assign the value for A, B, C and save your work. Then, select the “Behavioral Simulation” in the upper left pane.



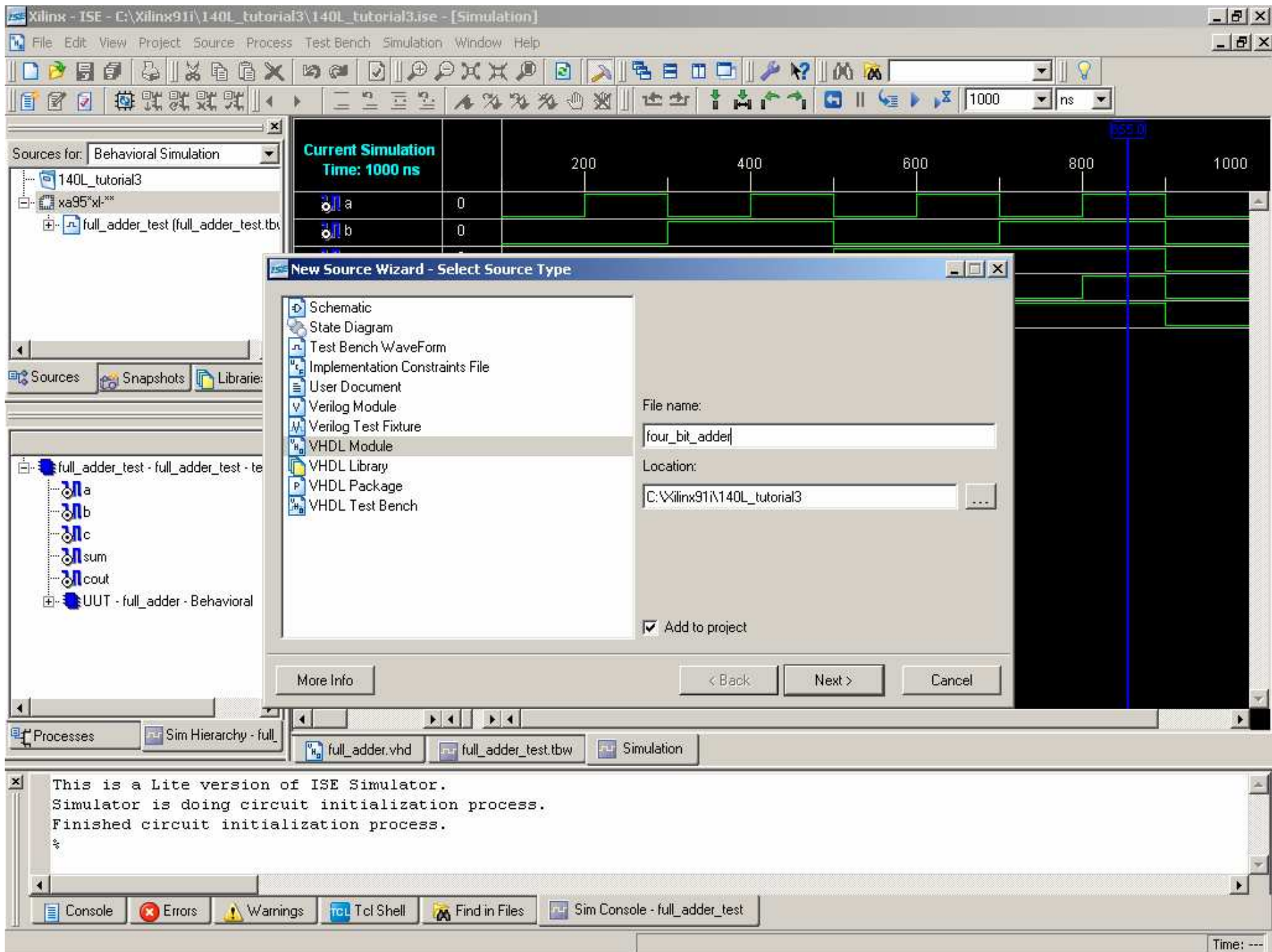
Select the “Processes” pane on the bottom left window and drill down to “Check Syntax” and run this check.



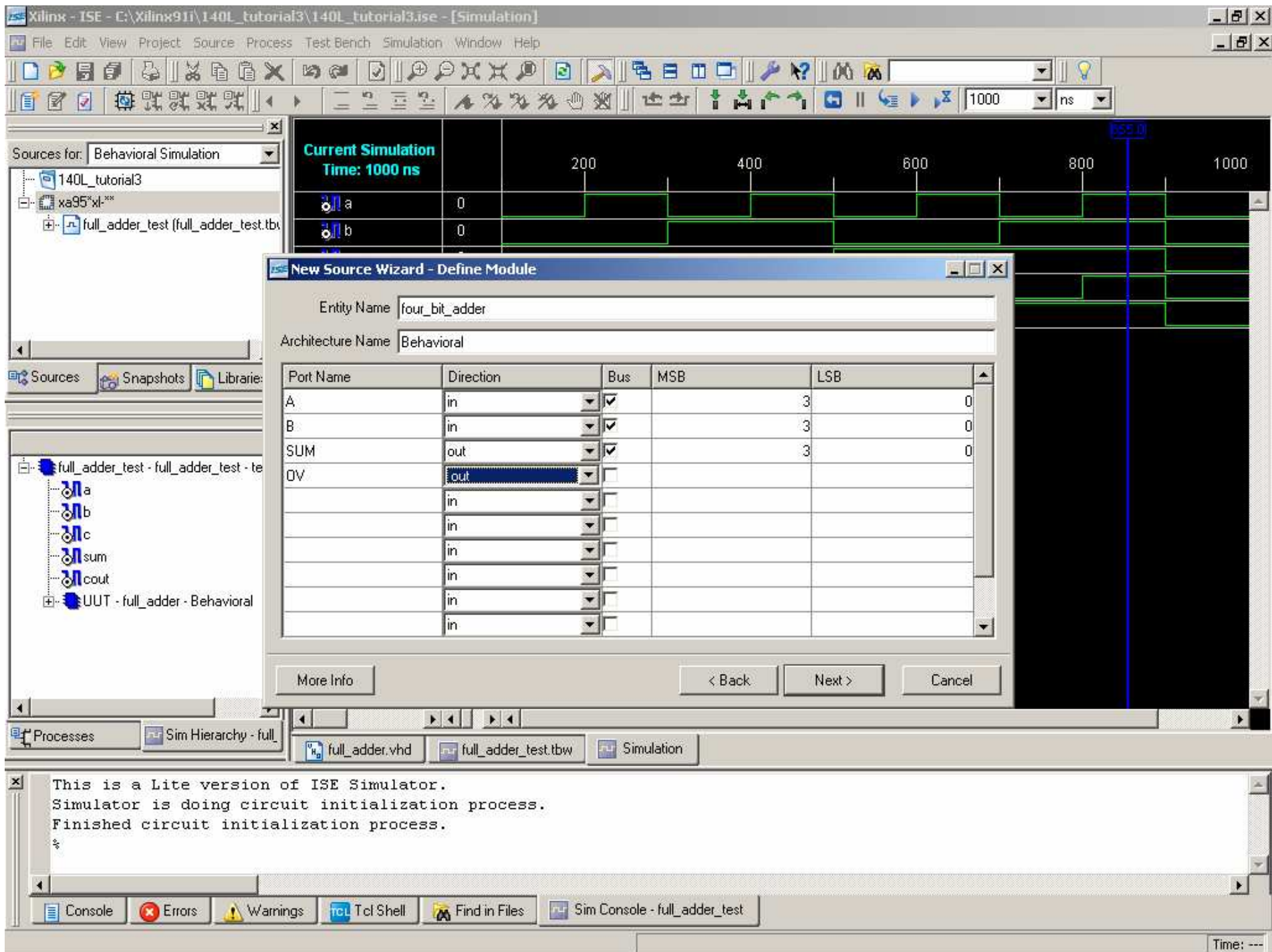
If you typed everything correctly, you will see the “completed successfully” message in the bottom console window. Now, you are ready to run the simulation.



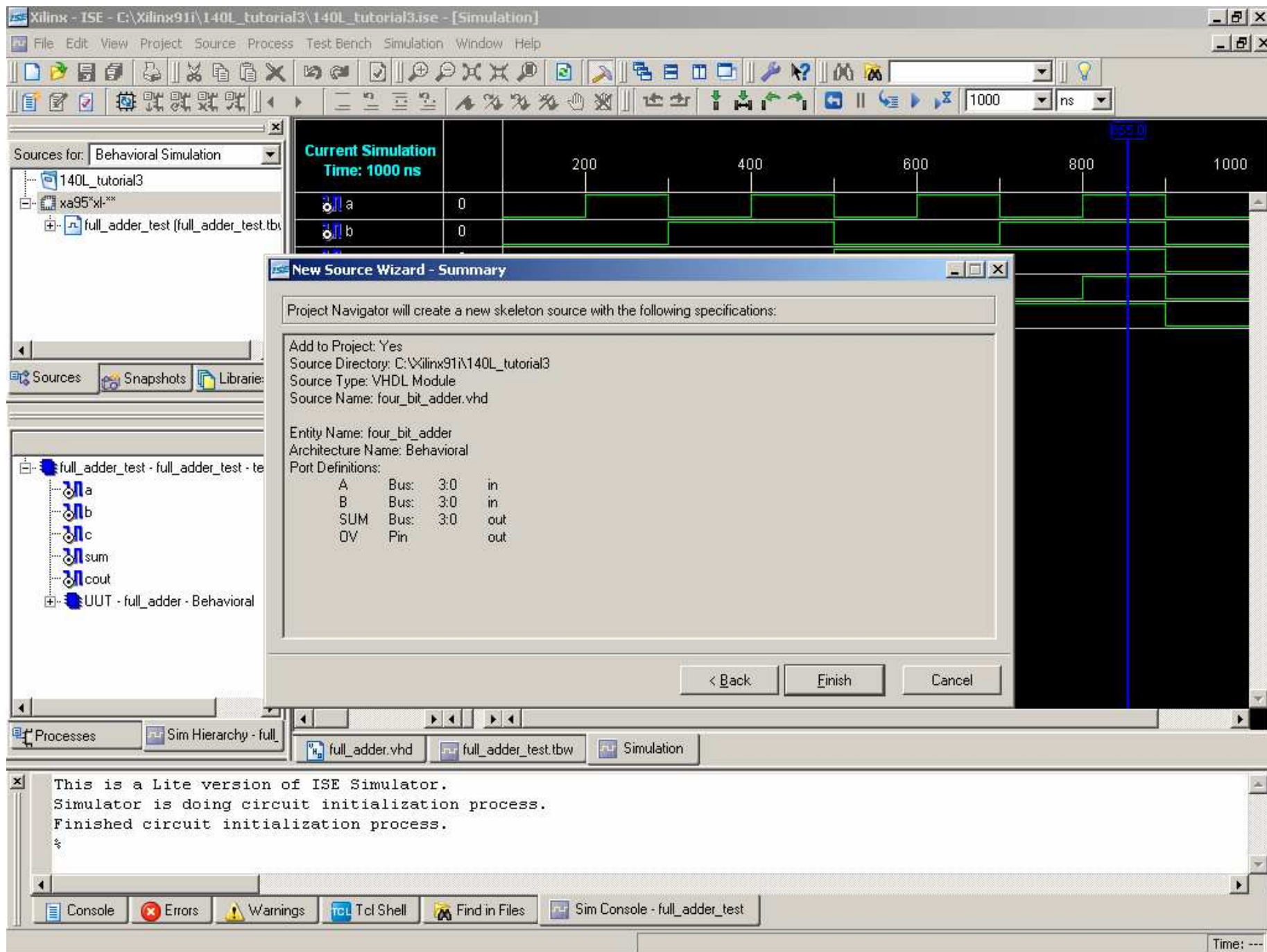
Here is the output of the simulation (using ISE Simulator)



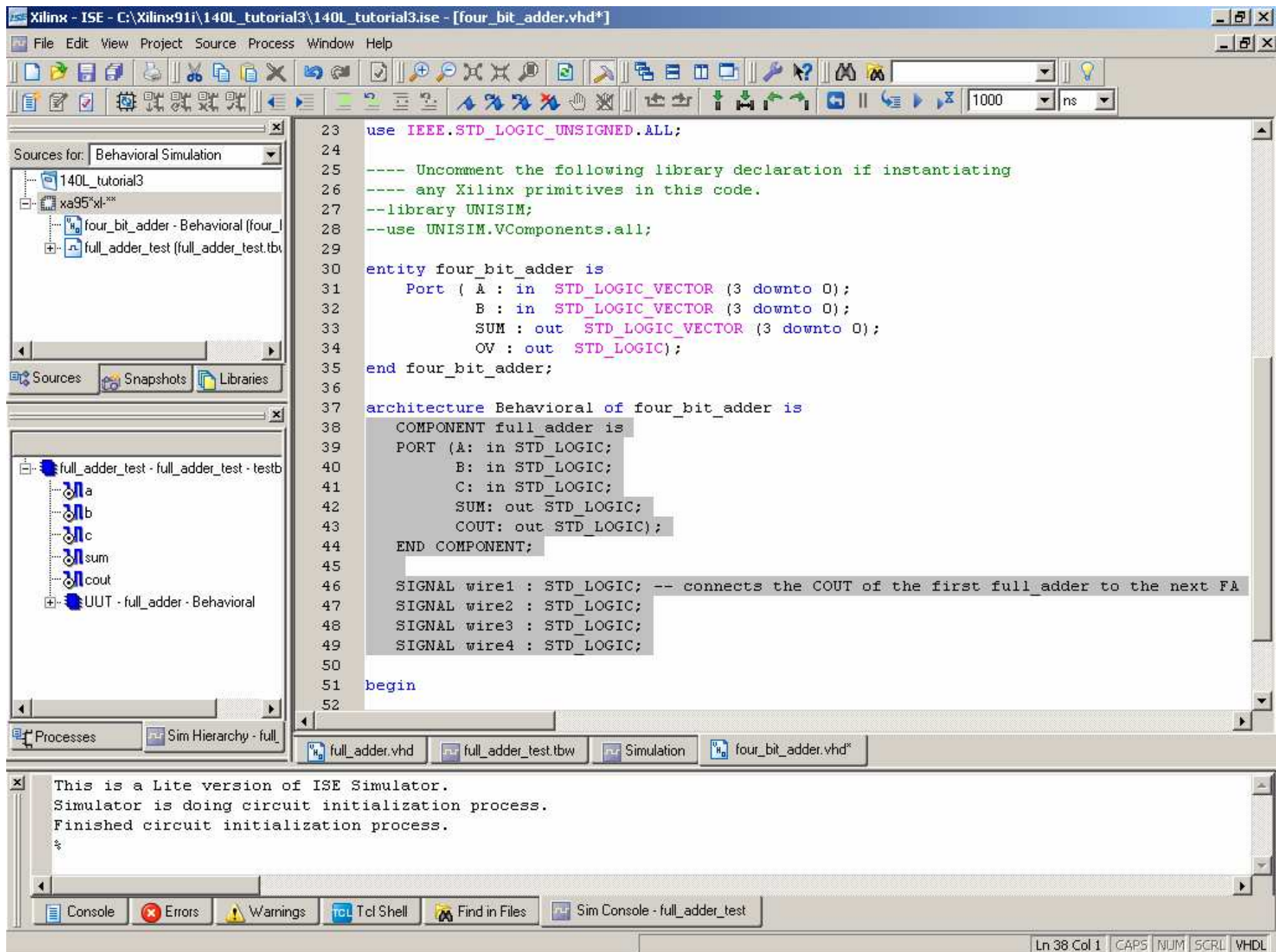
Now, let's create a four-bit adder using the full_adder we just created in VHDL as shown above.



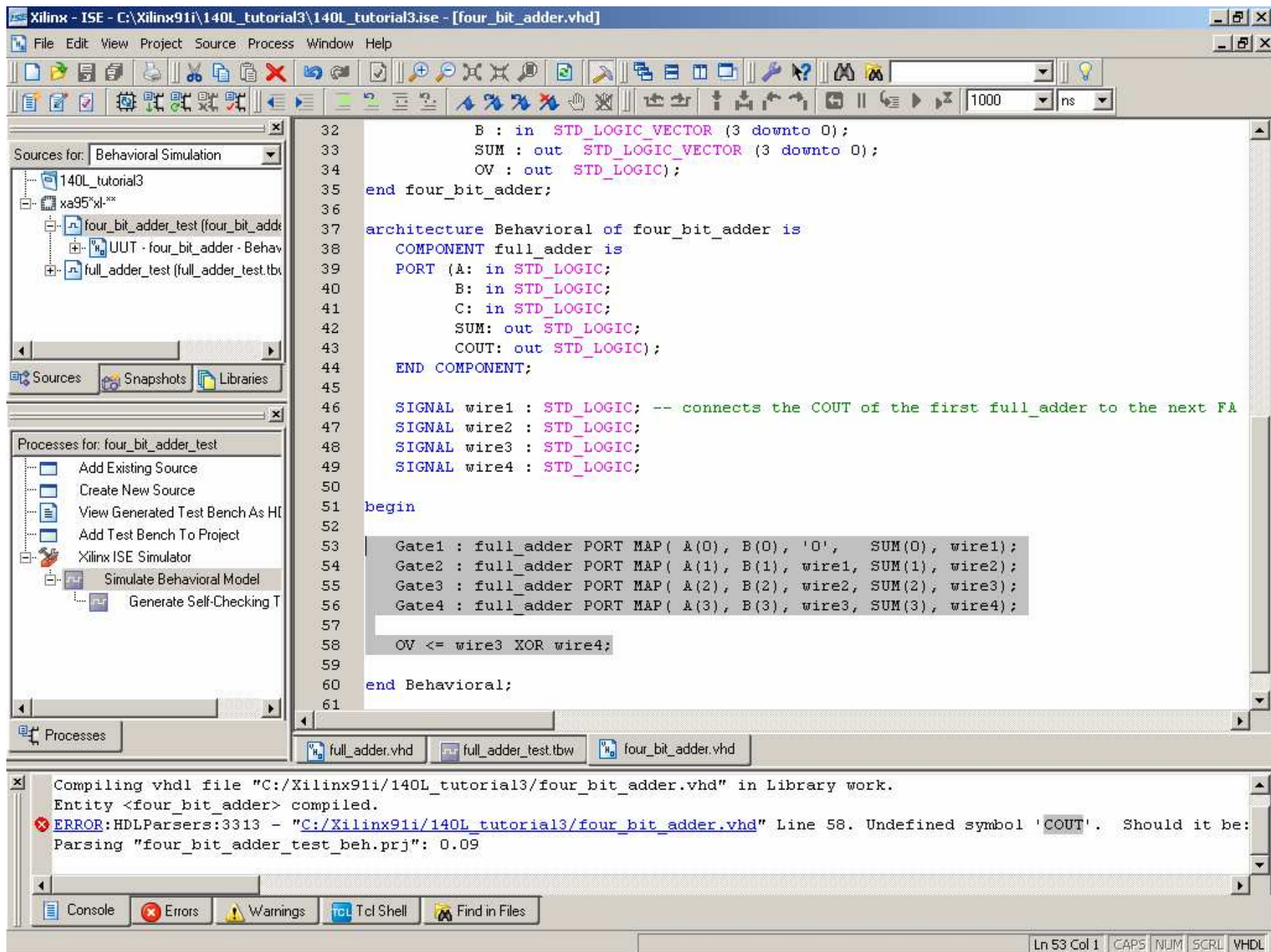
For input and outputs, click the “Bus” checkbox for A, B, and SUM. Enter the proper MSB value for the four-bit adder (3 down to 0) and Direction as shown above. Click Next



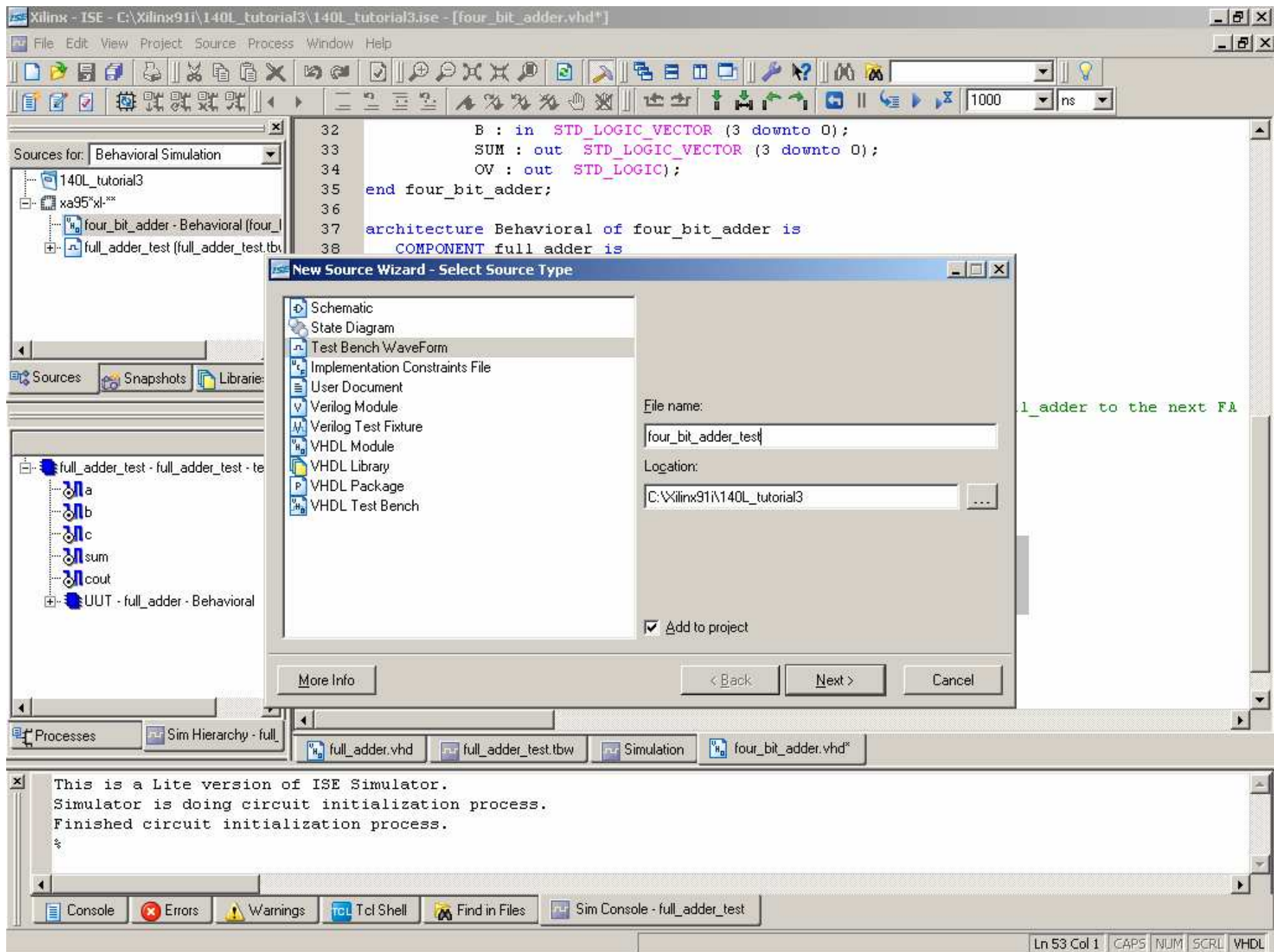
Quick summary page is shown. Click Finish.



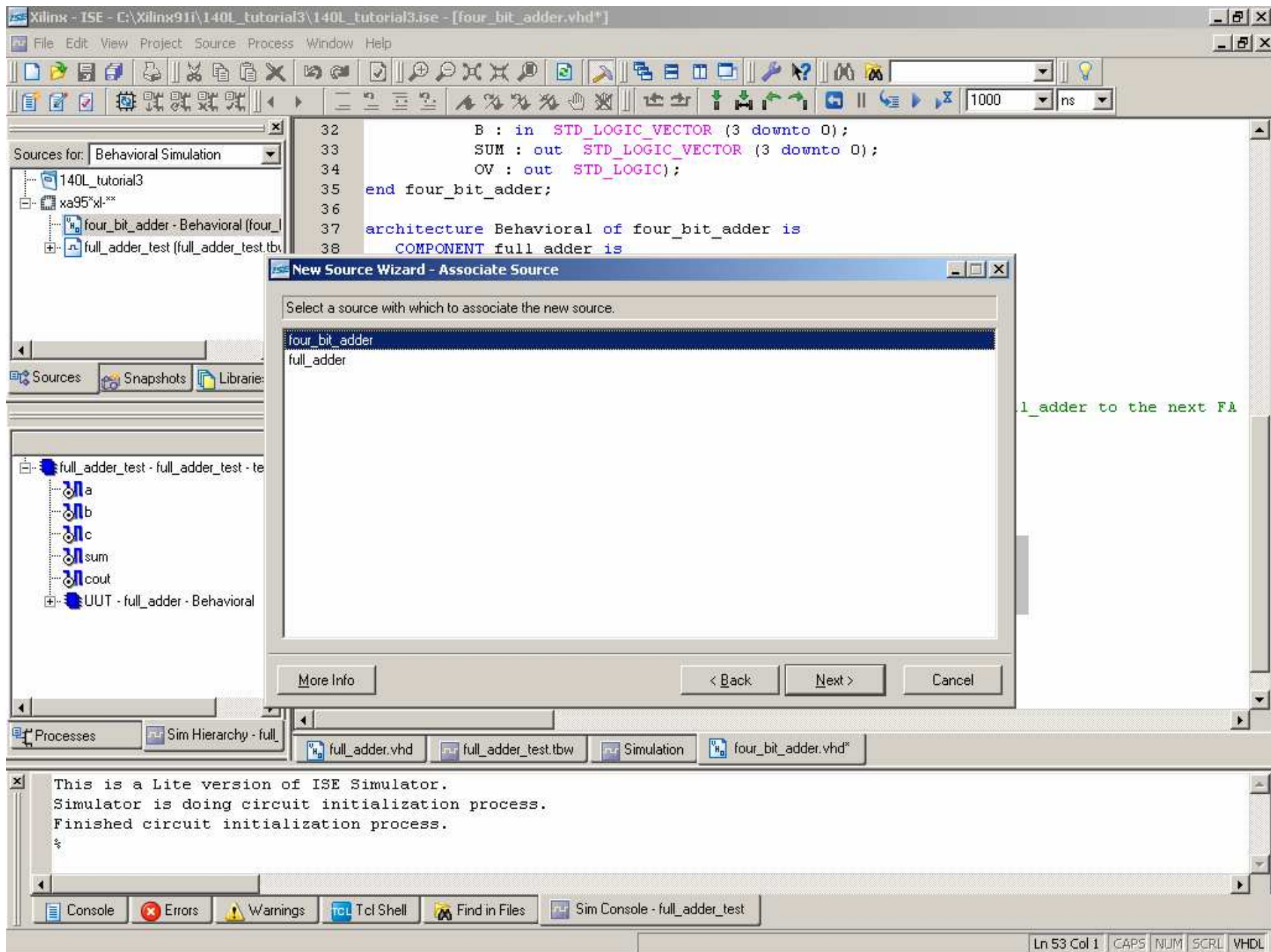
Type in the highlighted code shown above. This is similar to declaring a function and local variable in C language.



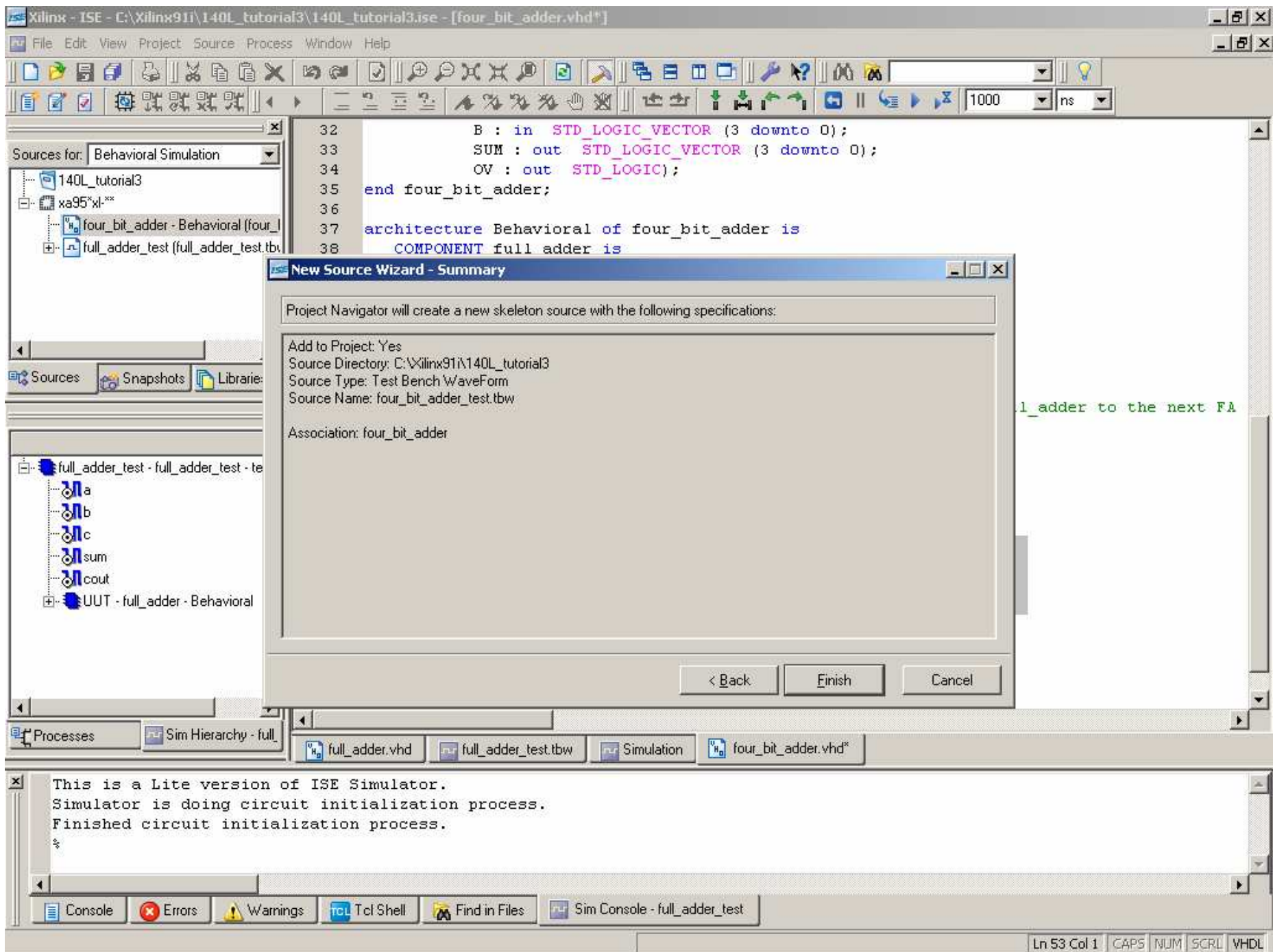
Now, add in the highlighted code for connecting four full_adder together. Save your work! Next, let's create a test-bench for the four bit adder.



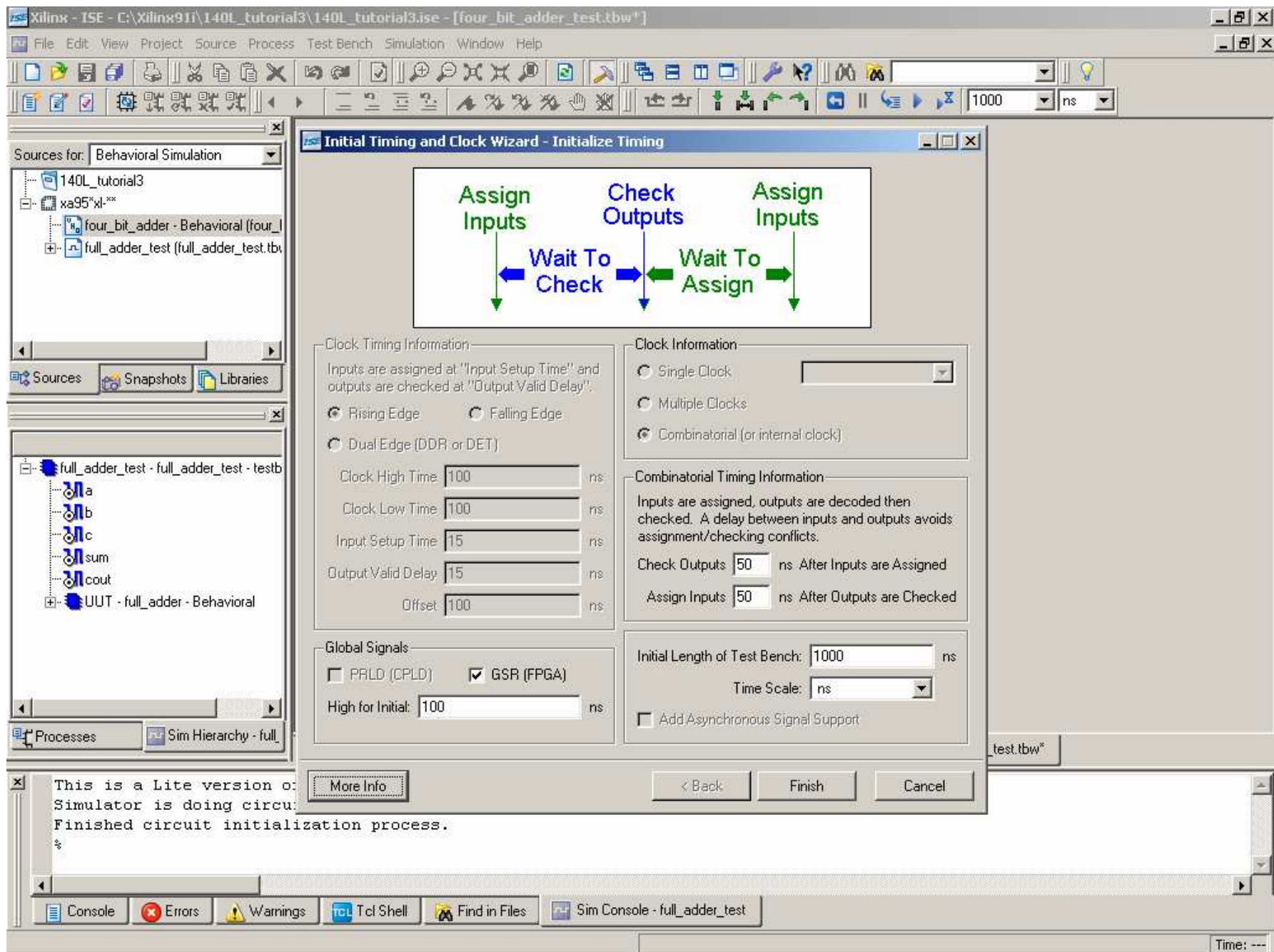
Be sure to select "Test Bench Waveform" and enter the file name "four_bit_adder_test". Click Next.



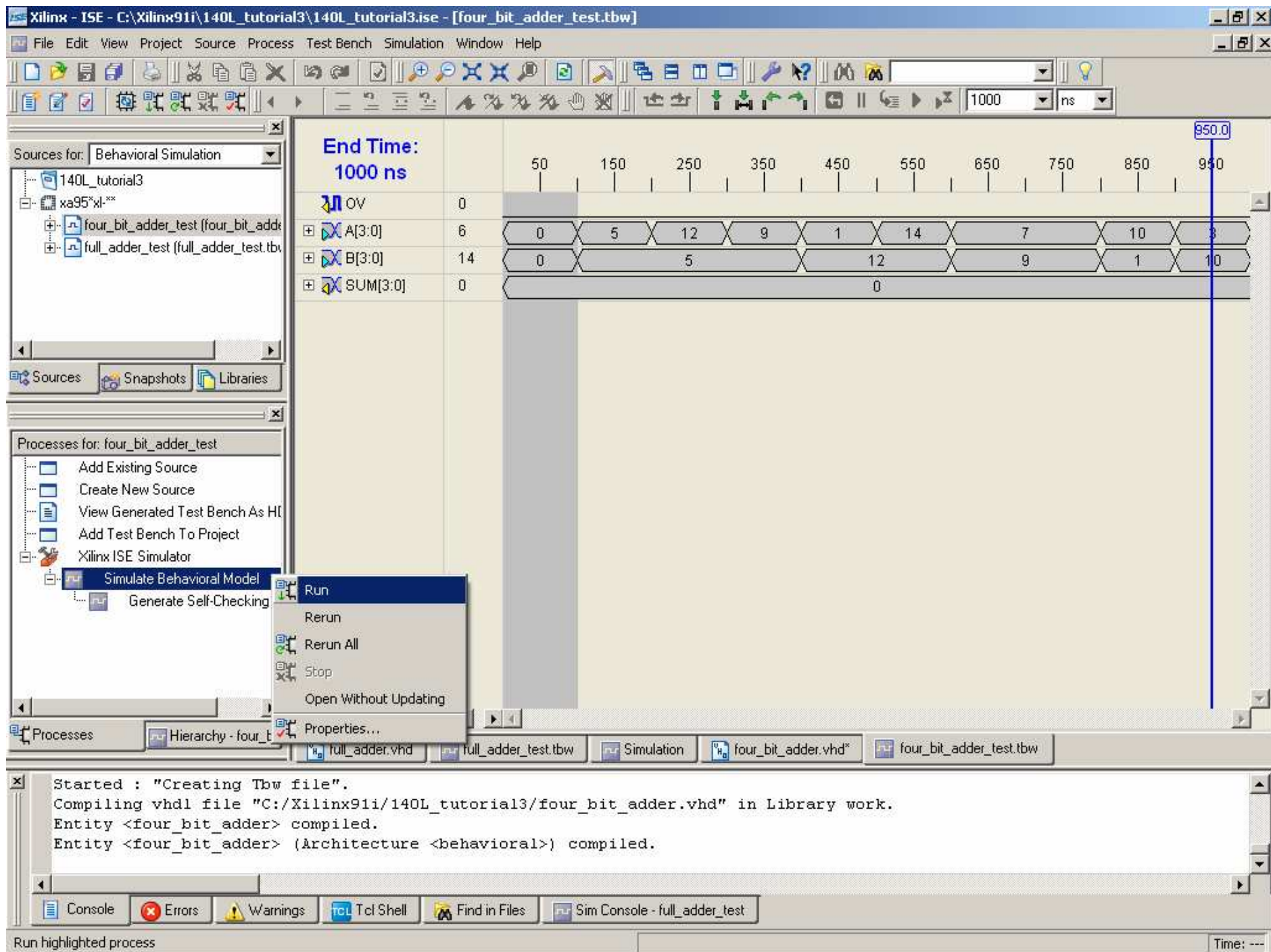
Highlight the “four_bit_adder” and click Next.



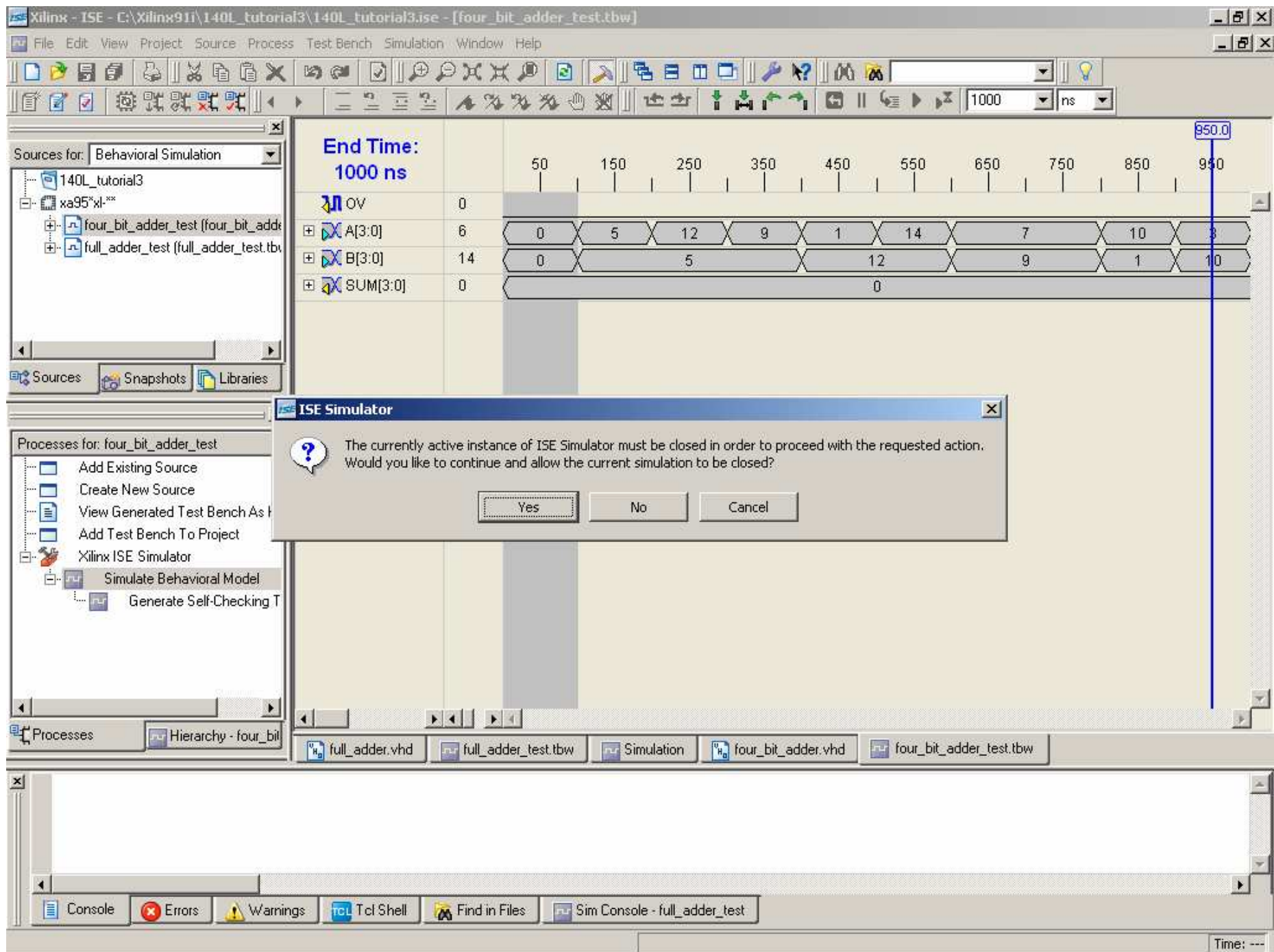
A quick summary page is shown. Click Finish.



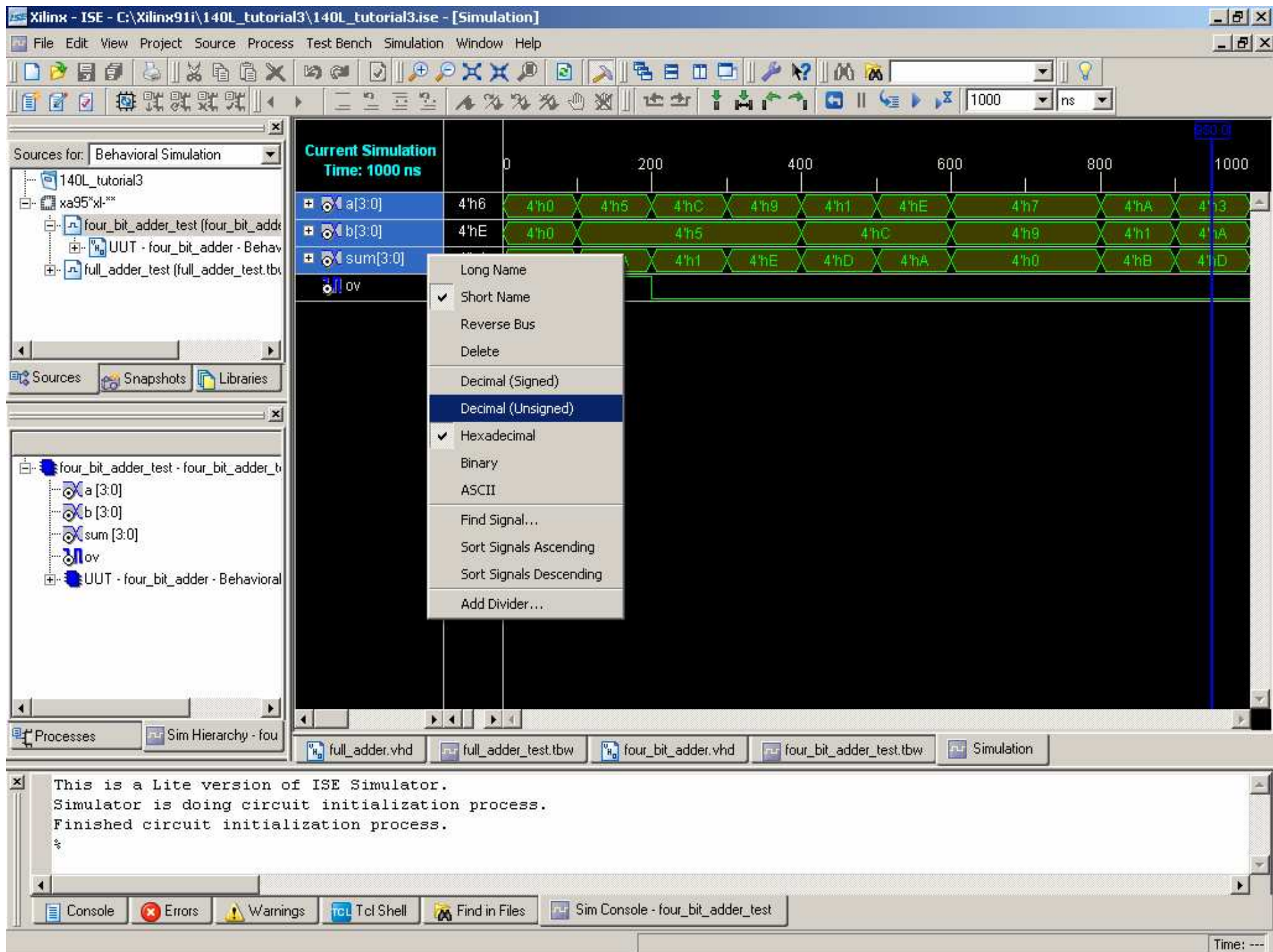
Click Finish. (Note that you can't select Single Clock or Multiple Clocks option because the inputs did not contain a single-wire signal to represent a clock signal.)



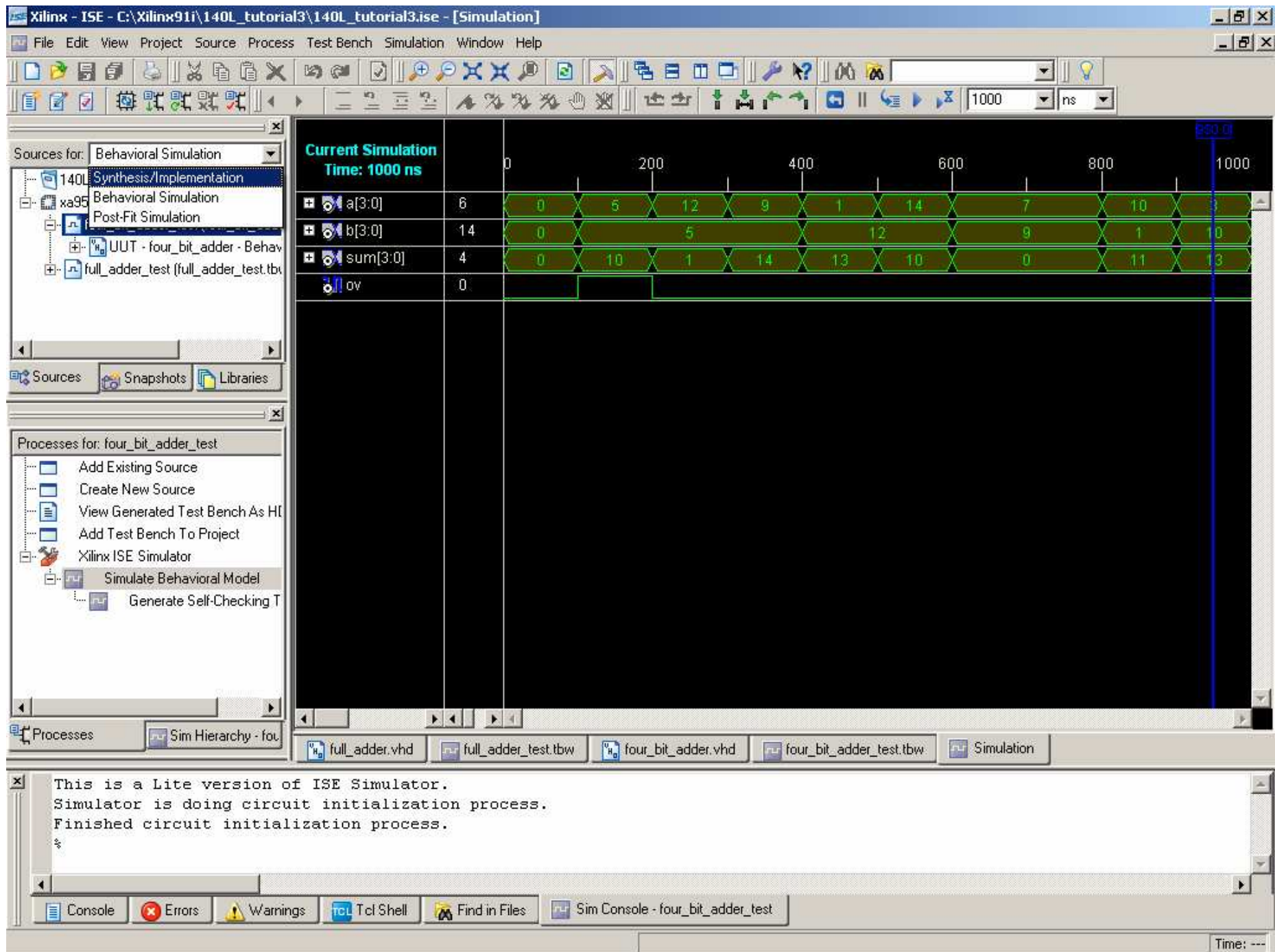
Set some random value for A[3:0] and B[3:0] as shown above. Save your work. Then, click on the “Processes” pane on the bottom-left window and Run the simulation.



Oops! We have an active Simulator. Click Yes to close the current one so we can run the new one.



Here is the simulation result. Take a look in Decimal (Unsigned) format.



There it is! Now, the last part is synthesis (converting VHDL code to actual circuit). Select “Synthesis/Implementation” drop down on the upper-left window.

Xilinx - ISE - C:\Xilinx91i\140L_tutorial3\140L_tutorial3.ise - [Simulation]

File Edit View Project Source Process Test Bench Simulation Window Help

Sources for: Synthesis/Implementation

- 140L_tutorial3
 - xa95*.xli**
 - four_bit_adder - Behavioral (fo

Processes for: four_bit_adder - Behavioral

- Add Existing Source
- Create New Source
- Design Utilities
- User Constraints
- Implement Design
 - Synthesize - XST
 - Translate
 - Fit
 - Generate Programm
 - Optional Implement

Run

Rerun

Rerun All

Stop

Open Without Updating

Properties...

Current Simulation Time: 1000 ns

		0	200	400	600	800	1000
a[3:0]	6	0	5	12	9	1	14
b[3:0]	14	0	5	12	9	1	10
sum[3:0]	4	0	10	1	14	13	10
ov	0						

Simulation

This is a Lite version of ISE Simulator.
 Simulator is doing circuit initialization process.
 Finished circuit initialization process.
 %

Console Errors Warnings Tcl Shell Find in Files Sim Console - four_bit_adder_test

Run highlighted process

Time: ---

Click Run on the Implementation Design Option. This may take a little time.

Xilinx - ISE - C:\Xilinx91i\140L_tutorial3\140L_tutorial3.ise - [C:\Xilinx91i\140L_tutorial3\four_bit_adder_html\fit\appletref.htm]

File Edit View Project Source Process Window Help

Sources for: Synthesis/Implementation

140L_tutorial3

xa95*.xli

four_bit_adder - Behavioral (fo

Sources Snapshots Libraries

Processes for: four_bit_adder - Behavioral

User Constraints

Implement Design

Synthesize - XST

View Synthesis Report

View RTL Schematic

View Technology S

Check Syntax

Translate

Fit

Generate Programming

Optional Implementation

Run

Rerun

Rerun All

Stop

Open Without Updating

Properties...

Sim Hierarchy - fou

full_adder.vhd

full_adder_test.tbw

four_bit_adder.vhd

four_bit_adder_test.tb

Simulation

C:\Xilinx91i\140L_tutorial3\fo

XILINX CPLD Reports

XC9500XL

Fitter Report | Timing Report

Fitter Report

Summary

Errors/Warnings

Logic

Inputs

Function Blocks

Equations

Pin List

Compiler Options

Text Report

Help

Summary

Design Name	four_bit_adder
Fitting Status	Successful
Software Version	J.33
Device Used	XA9536XL-15-VQ44
Date	6- 4-2007, 10:40PM

RESOURCES SUMMARY

Macrocells Used	Pterms Used	Registers Used	Pins Used	Function Block Inputs Used
7/36 (20%)	24/180 (14%)	0/36 (0%)	13/34 (39%)	14/108 (13%)

PIN RESOURCES

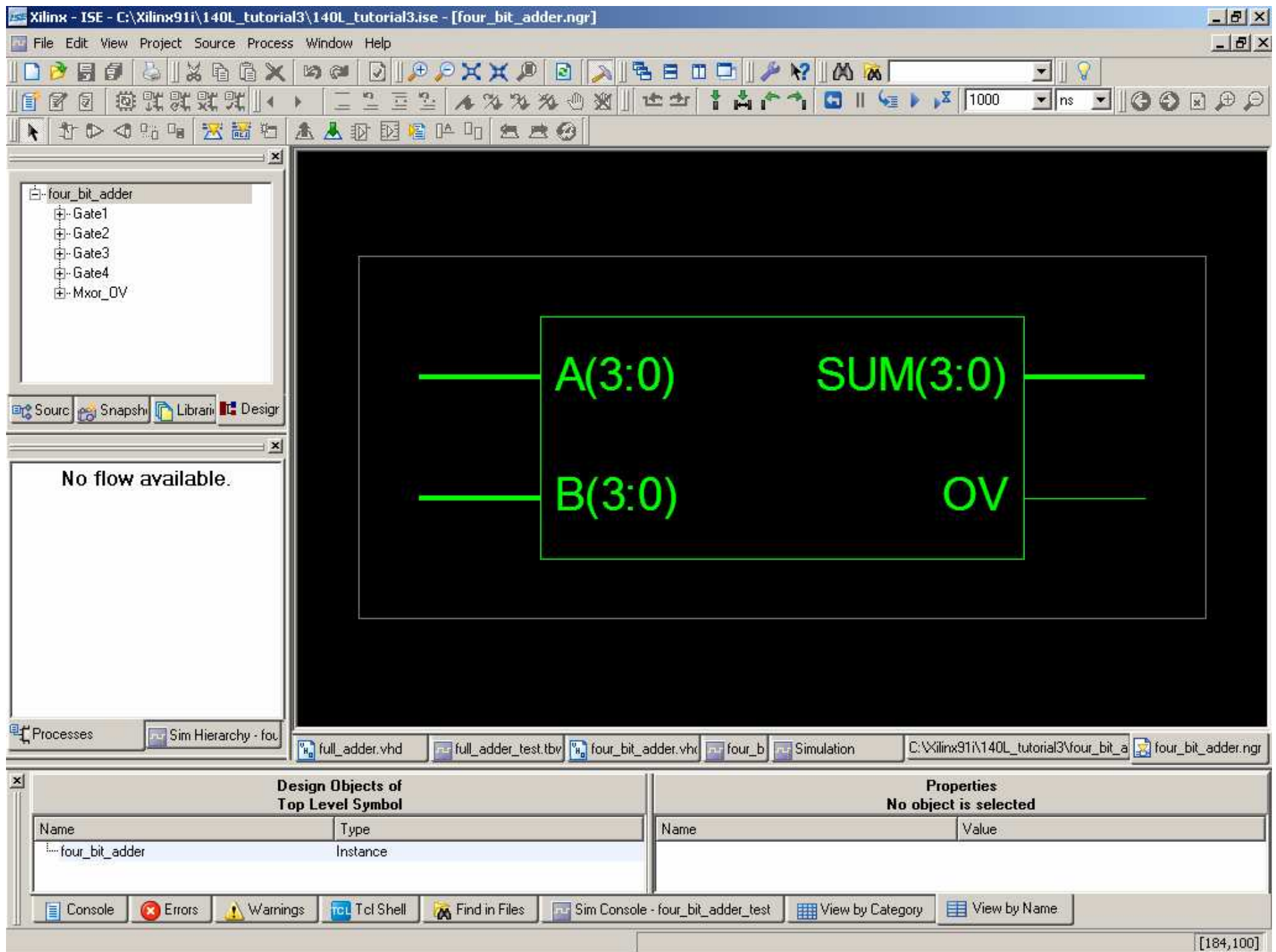
Signal Type	Required	Mapped	Pin Type	Used	Total
Input	8	8	I/O	13	28

This is a Lite version of ISE Simulator.
 Simulator is doing circuit initialization process.
 Finished circuit initialization process.

Console Errors Warnings TCL Tcl Shell Find in Files Sim Console - four_bit_adder_test

Run highlighted process

If all is well, you can see the CPLD Reports. Drill down on the Synthesis –XST and view the RTL Schematic.



Double click on the circuit to drill down to the detail implementation.

Xilinx - ISE - C:\Xilinx91i\140L_tutorial3\140L_tutorial3.ise - [four_bit_adder.ngr]

File Edit View Project Source Process Window Help

1000 ns

four_bit_adder

- Gate1
- Gate2
- Gate3
- Gate4
- Mxor_OV

Source Snapshot Library Design

No flow available.

Processes Sim Hierarchy - four_bit_adder

full_adder.vhd full_adder_test.tbv four_bit_adder.vhd four_b Simulation C:\Xilinx91i\140L_tutorial3\four_bit_a four_bit_adder.ngr

Design Objects of four_bit_adder

Name	Type
wire3,wire2,wire1,xv0	Net
wire4,wire3,wire2,wire1	Net
XST_GND	Instance

Properties
No object is selected

Name	Value

Console Errors Warnings Tcl Shell Find in Files Sim Console - four_bit_adder_test View by Category View by Name

[1132,444]

Double click on the full-adder to see one more level of detail.

Xilinx - ISE - C:\Xilinx91i\140L_tutorial3\140L_tutorial3.ise - [four_bit_adder.ngr]

File Edit View Project Source Process Window Help

1000 ns

four_bit_adder

- Gate1
- Gate2
- Gate3
- Gate4
- Mxor_OV

No flow available.

Processes Sim Hierarchy - four_bit_adder

full_adder.vhd full_adder_test.tbv four_bit_adder.vhd four_b Simulation C:\Xilinx91i\140L_tutorial3\four_bit_a four_bit_adder.ngr

Design Objects of Gate4

Name	Type
Mxor_Sum	Instance
SUM	Net
SUM	Pin

Properties

No object is selected

Name	Value
------	-------

Console Errors Warnings Tcl Shell Find in Files Sim Console - four_bit_adder_test View by Category View by Name

[1076,384]

Now you have learned enough about the tool to work on lab 4! Be sure you can synthesize lab 4 to get full credit. Good luck!