Introduction to VHDL in Xilinx ISE 10.1

Objectives

- Learn VHDL with Xilinx package
- Create projects
    1. Create VHDL source
    2. Enter VHDL code
3. Synthesize VHDL code
- Simulate a Module Using ISE Simulator
    1. Create test bench for simulation
    2. Simulate behavioral model by using ISE Simulator

Xilinx ISE means Xilinx® Integrated Software Environment (ISE). This Xilinx® design software suite allows you to take your design from design entry through Xilinx device programming. The ISE Project Navigator manages and processes your design through several steps in the ISE design flow. These steps are Design Entry, Synthesis, Implementation, Simulation/Verification, and Device Configuration.

In this lab, you are going to see how to create a counter in the Xilinx ISE and how to simulate it with ISE Simulator. You will create a test bench waveform containing input stimulus you can use to simulate the counter module. This test bench waveform is a graphical view of a test bench.

Open Xilinx ISE 10.1

To open the Xilinx ISE 10.1, type ‘ise’ on terminal (without quotes).

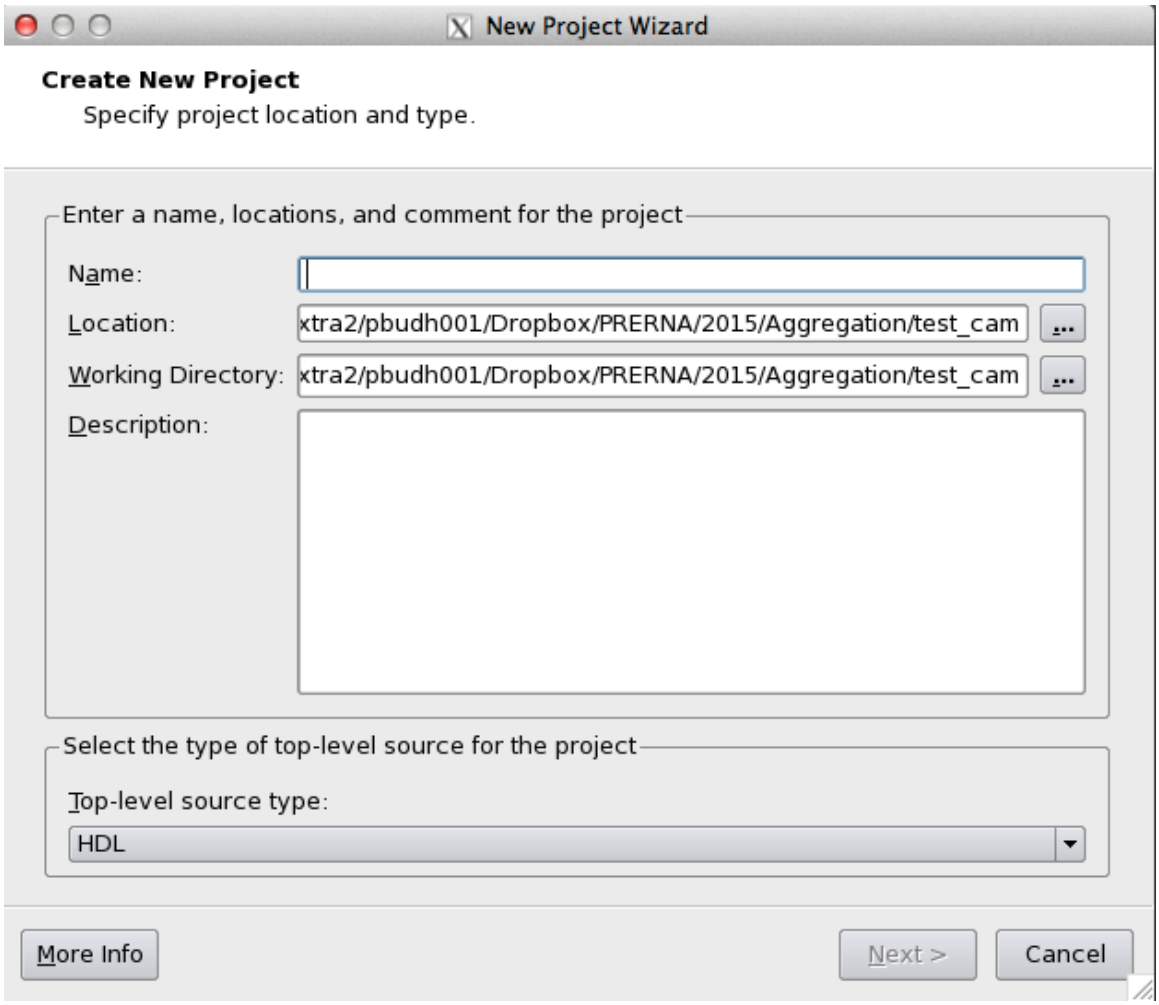
Create a project

In this section, you will create a new ISE project. A project is a collection of all files nesessary to create and to download a design to a selected FPGA or CPLD device.

1. Select File -> New Project. The New Project Wizard appears.

2. First, enter a location (directory path) for the new project, then give a name for the project. For example, we name it Counter01.

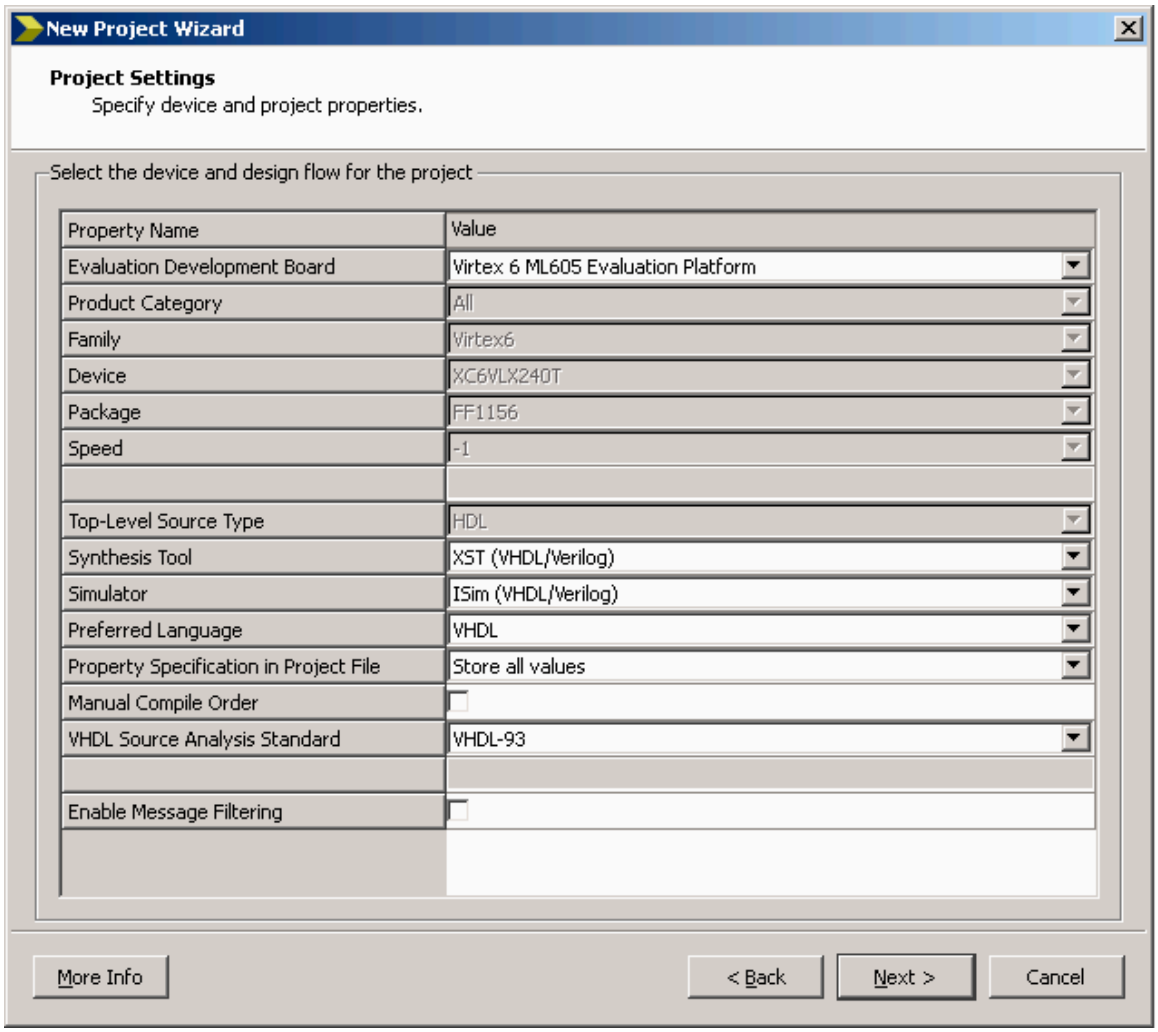
Here is a picture to illustrate the idea.



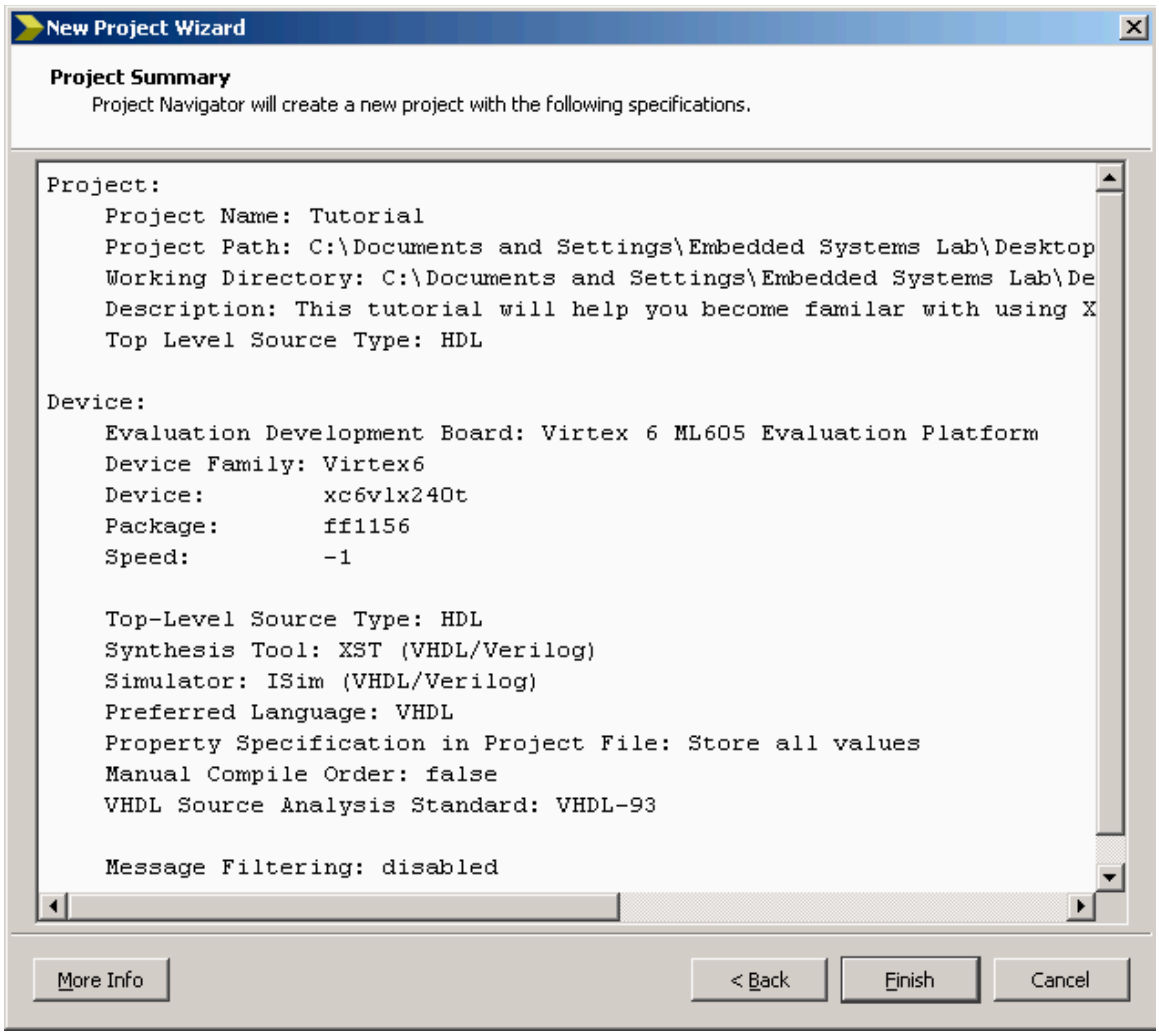
Select the Evaluation Development Board, if known.
(I.E. Spartan-3E Starter Board)
Otherwise, you will have to look at the FPGA physical board chip or product documentation for the Family, Device, Package, and Speed.

Other settings are shown below.
Synthesis Tool: XST (VHDL/Verilog)
Simulator: ISim (VHDL/Verilog)
Preferred Language: VHDL Property Specification in Project File: Store all Values
Manual Compile Order: Unselected
VHDL Source Analysis Standard: VHDL-93
Enable Message Filtering: Unselected

Select Next >

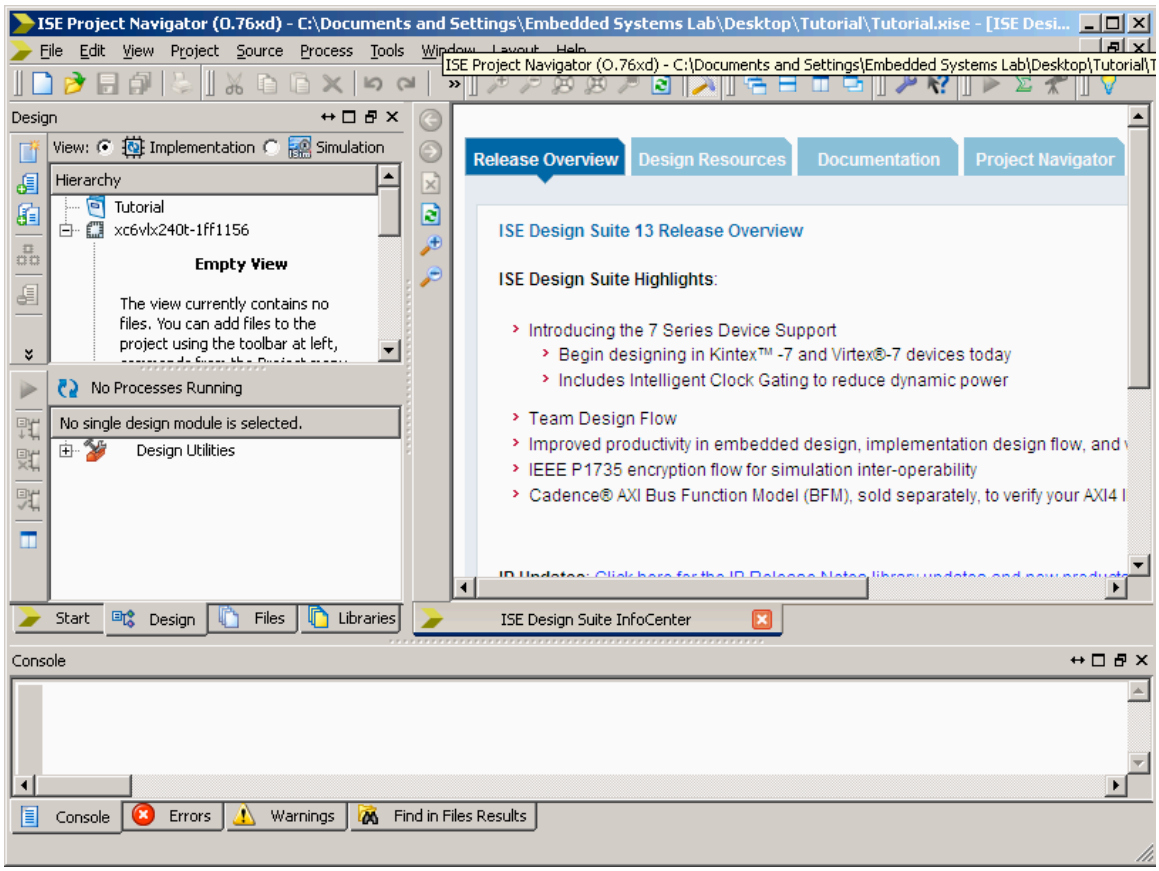


A Project Summary will then be shown. Review the content for correctness and select Finish if correct, otherwise repeat the previous steps.

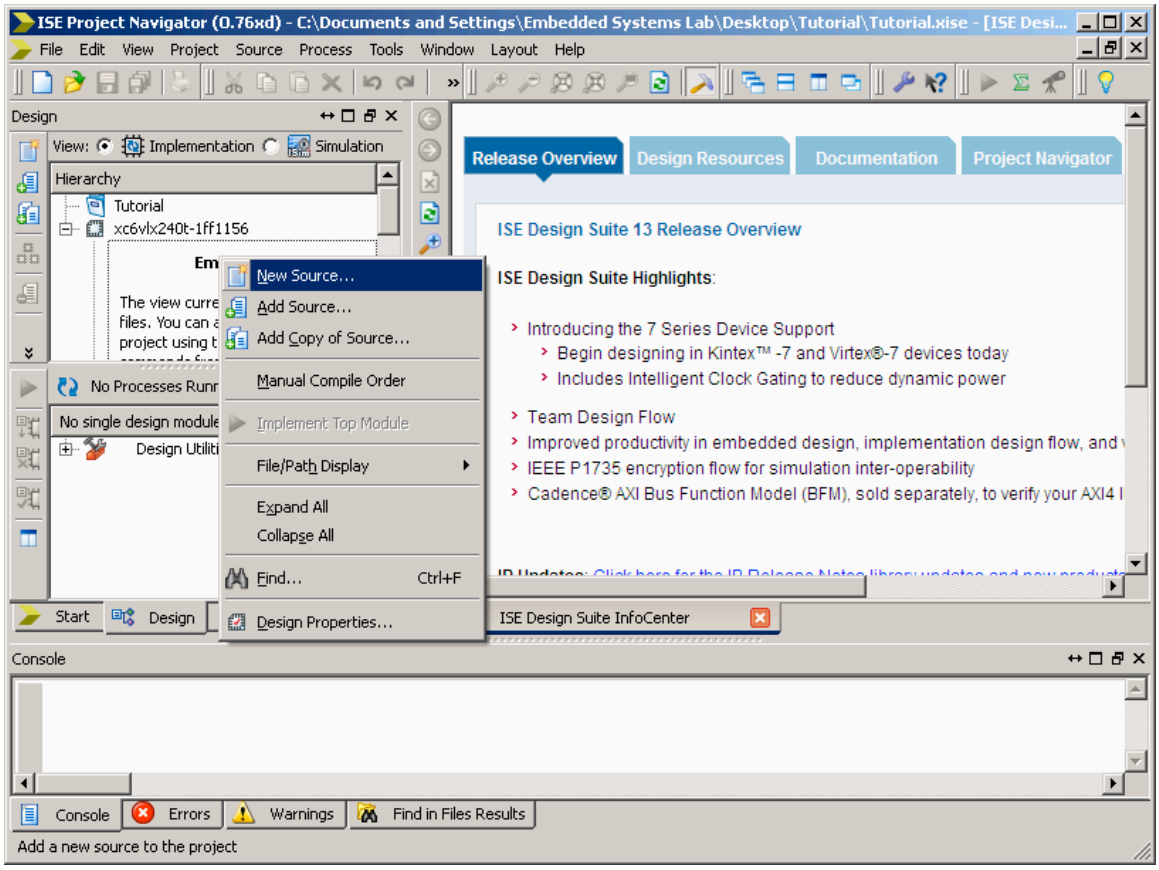


Step 2: Entity Setup

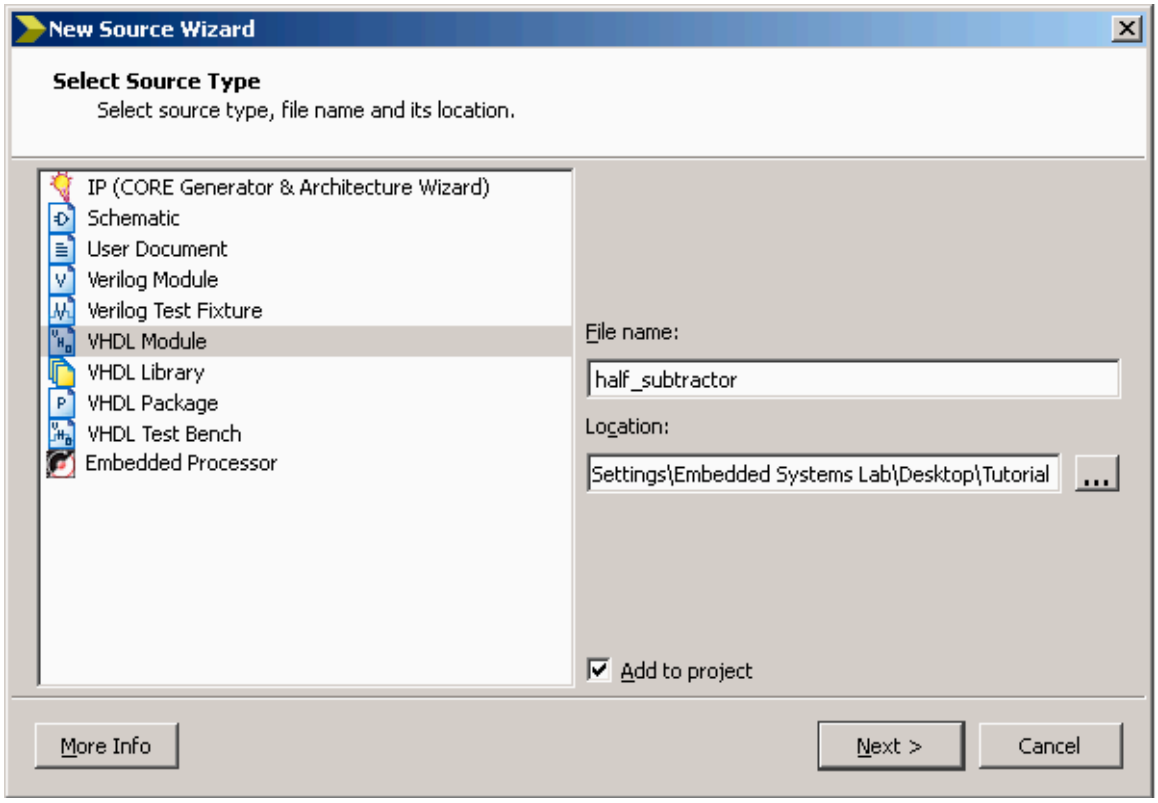
You are now viewing the Project Navigator. The Sources in the Project section will automatically organize your VHDL module tree (Top Left). The Processes for Source pane will allow you to perform various processes such as synthesis or device programming, view reports, and access useful tools (Middle Left). The bottom pane contains console output - notice the Console, Errors, Warnings tabs; these are useful in debugging (bottom section). The right are is used to display any files or documents you have opened.


ISE Project Navigator (0.76xd) - C:\Documents and Settings\Embedded Systems Lab\Desktop\Tutorial\Tutorial.xise - [ISE Desi...
File Edit View Project Source Process Tools Window Layout Help
ISE Project Navigator (O.76xd) - C:\Documents and Settings\Embedded Systems Lab\Desktop\Tutorial\T
Design
View: Implementation Simulation
Hierarchy
Tutorial
xc6vlx240t-1ff1156
Empty View
The view currently contains no files. You can add files to the project using the toolbar at left,
No Processes Running
No single design module is selected.
Design Utilities
Start Design Files Libraries
Release Overview
Design Resources
Documentation
Project Navigator
ISE Design Suite 13 Release Overview
ISE Design Suite Highlights:
Introducing the 7 Series Device Support
Begin designing in Kintex™ -7 and Virtex®-7 devices today
Includes Intelligent Clock Gating to reduce dynamic power
Team Design Flow
Improved productivity in embedded design, implementation design flow, and
IEEE P1735 encryption flow for simulation inter-operability
Cadence® AXI Bus Function Model (BFM), sold separately, to verify your AXI4
ISE Design Suite InfoCenter
Console
Console Errors Warnings Find in Files Results

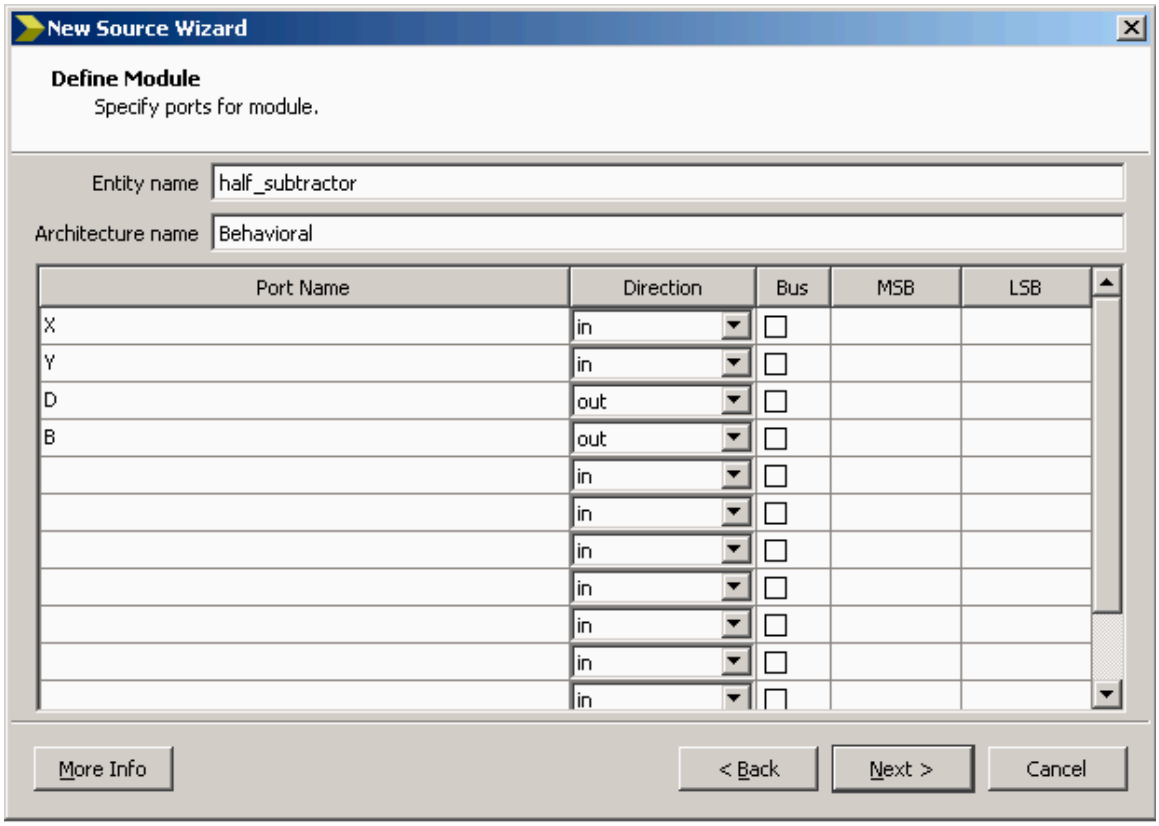
Now you will create a new VHDL entity. Right-Click on the chip icon and select New Source. Take note that you can add already created sources.



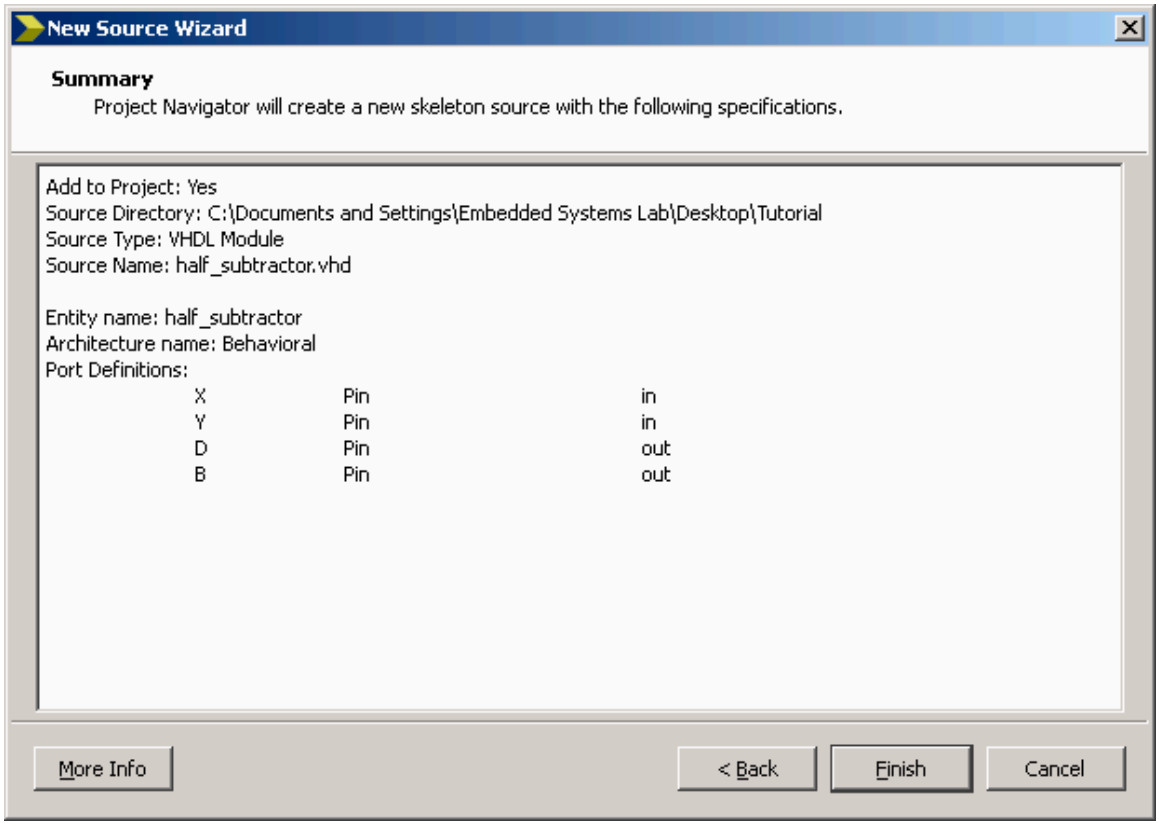
Select VHDL Module
Give the module a name.
(I.E. Pass_thru)
Select Next >



The Entity name will near in its text box. The Architecture name can also be changed in its text box as well, for this example we are describing a Behavioral.
In the first Port Name text box give the port a name.
Select **Next >**



- Review the summary of the entity then Select Finish.

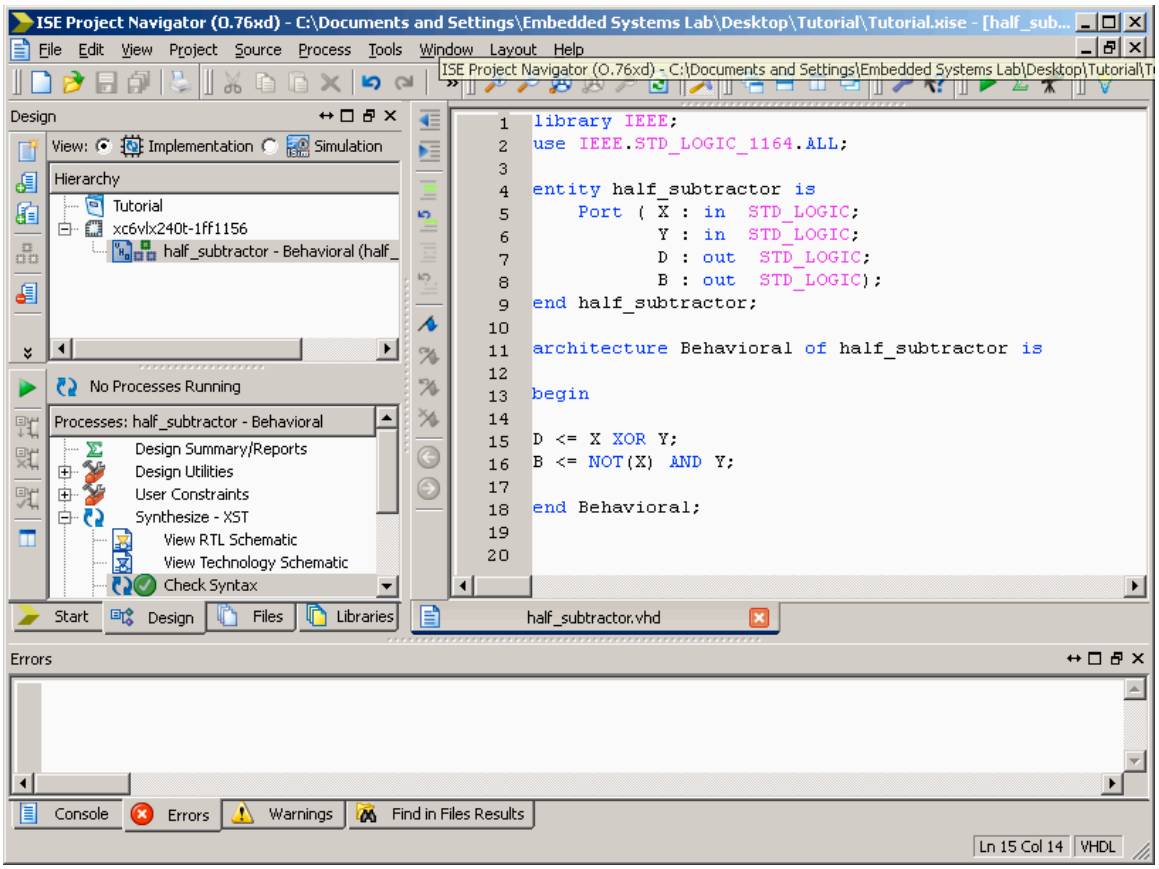


Step 3: Synthesis

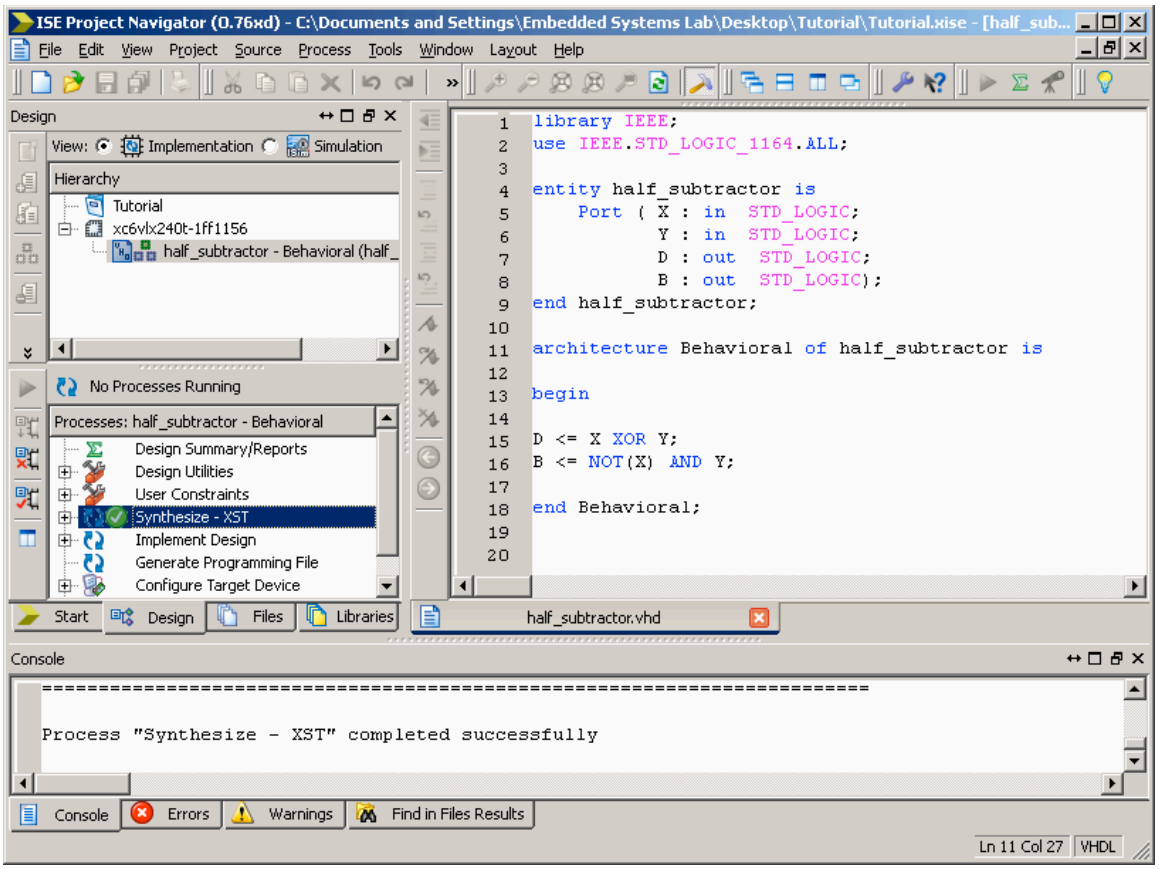
- Enter the following code.

```
library IEEE; use IEEE.STD_LOGIC_1164.ALL;
entity half_subtractor is
Port ( X : in  STD_LOGIC;
       Y : in  STD_LOGIC;
       D : out  STD_LOGIC;
       B : out  STD_LOGIC);
end half_subtractor;
architecture Behavioral of half_subtractor is
begin
  D <= X XOR Y;   B <= NOT(X) AND Y;
end Behavioral;
```

Once Entered, now we need to check if the VHDL syntax is correct. Expand the **Synthesize - XST** tree and double click **Check Syntax**.



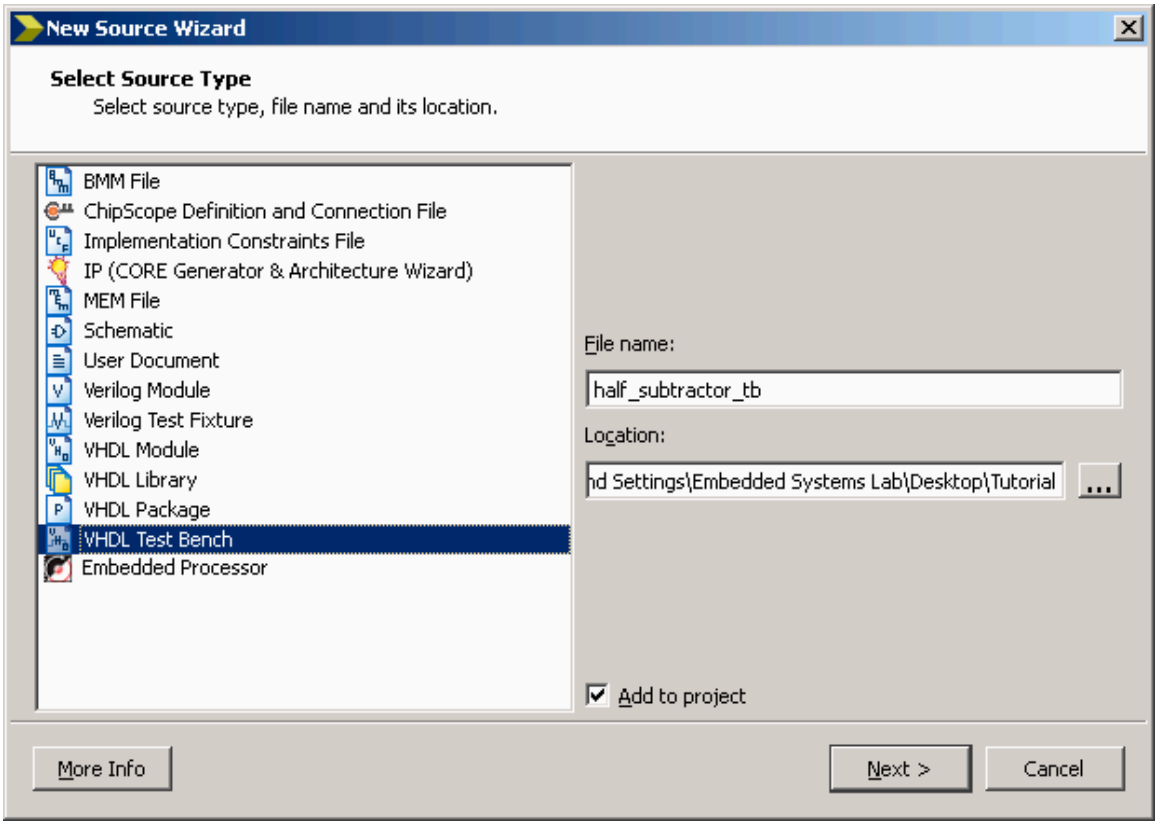
If correct, then close tree and select **Synthesize - XST** to check to make sure the code is synthesizable; there should not be warnings or errors.



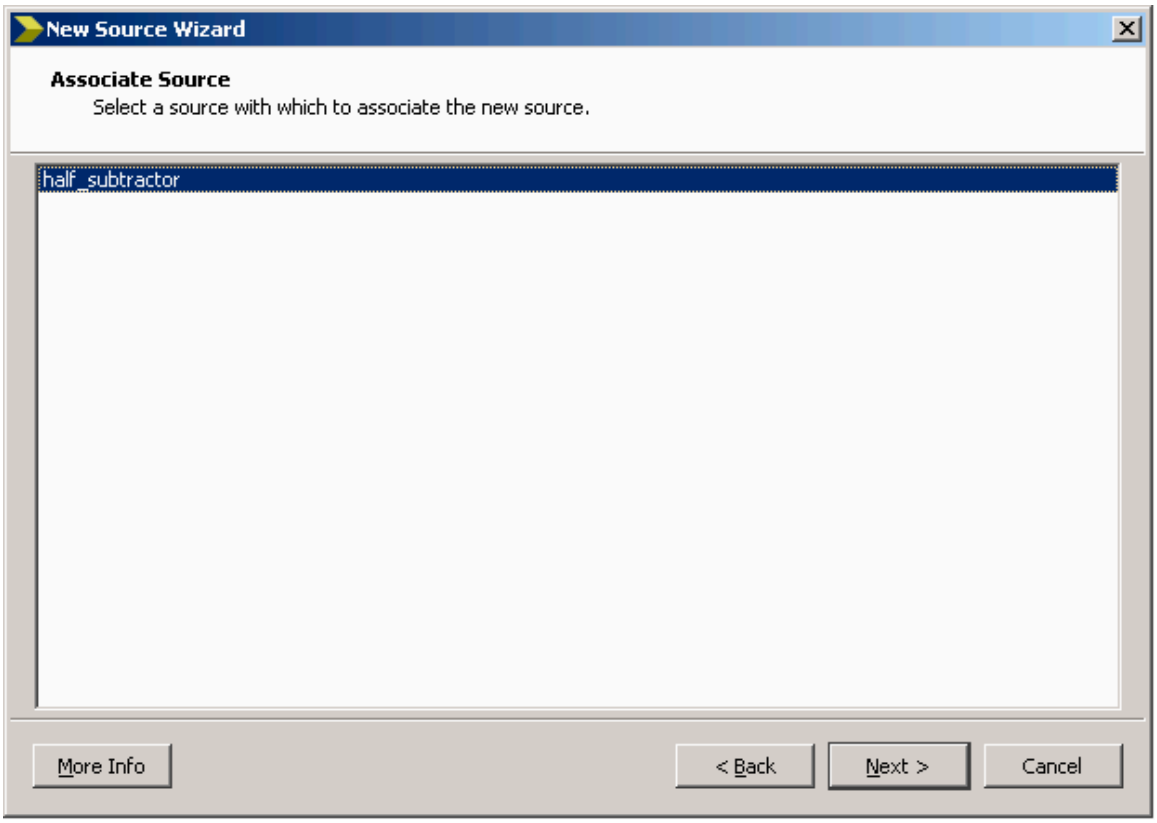
Step 4: Testbench

- Add a new source.

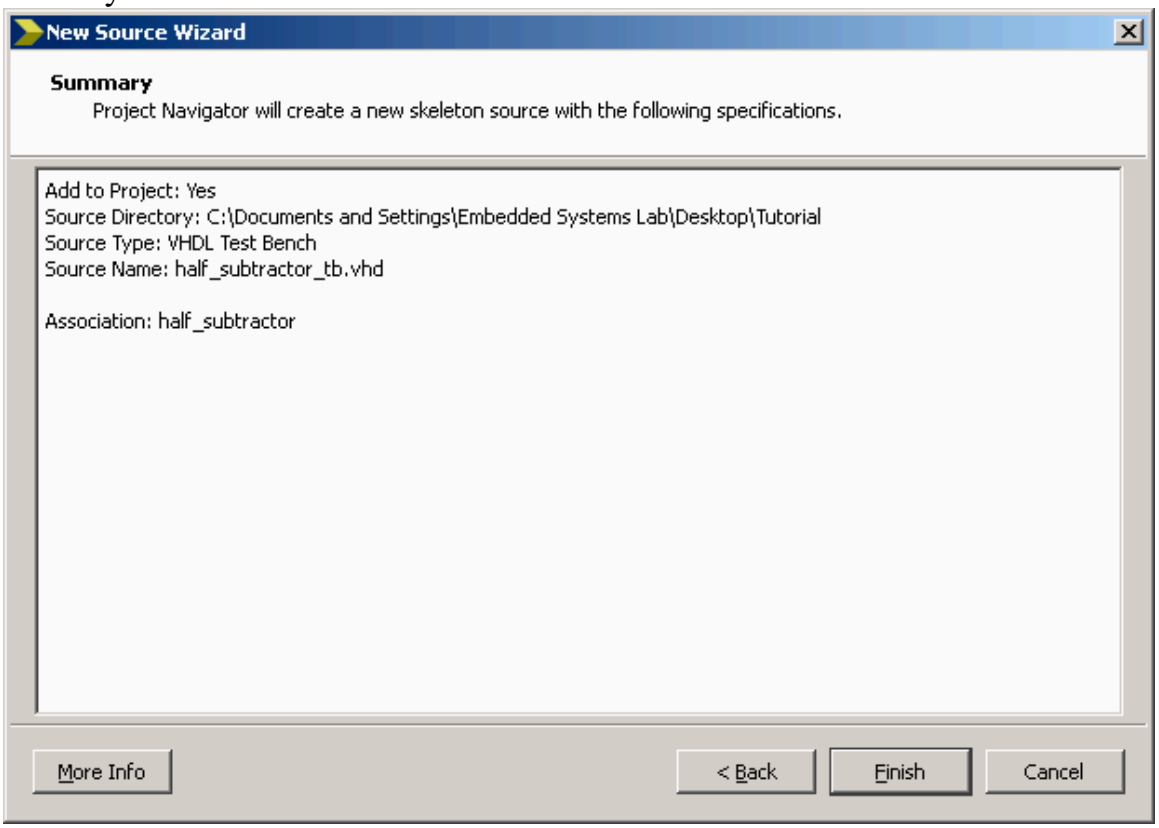
- Select VHDL Test Bench
  File name: pass_thru_tb
- Select **Next >**



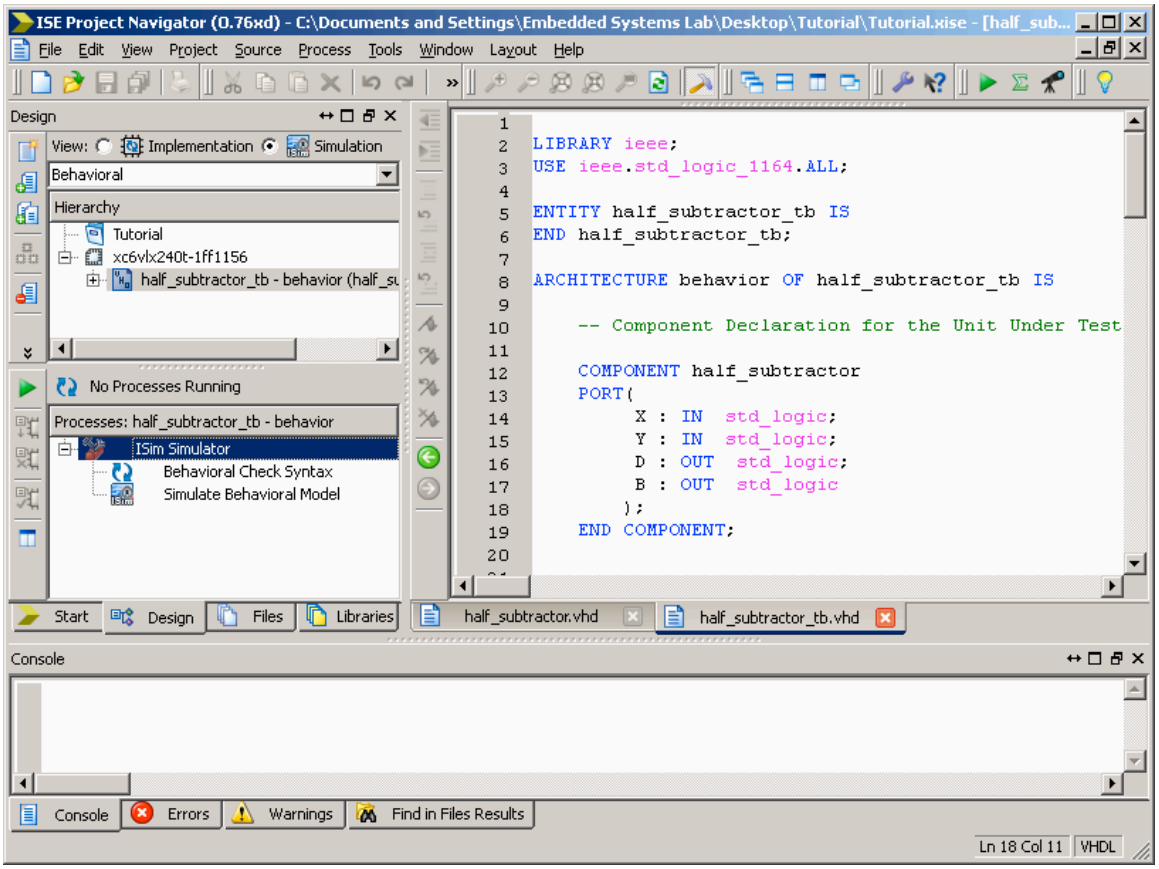
- Select pass_thru for the associate source.
- Select **Next >**



Review the Summary and select **Finish**.



Select the Simulation on the view button above the VHDL module tree to see the testbench and simulation tree.



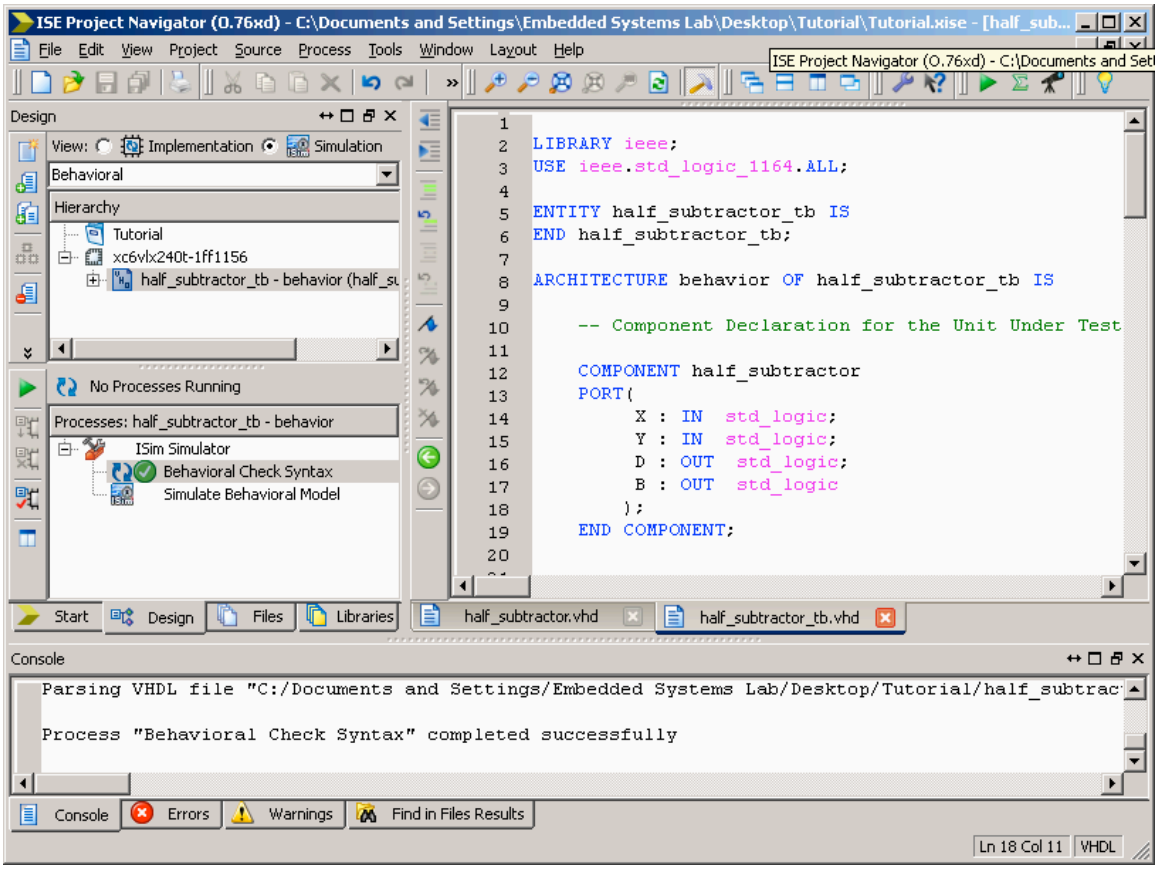
Enter the code

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY half_subtractor_tb IS
END half_subtractor_tb;

ARCHITECTURE behavior OF half_subtractor_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT half_subtractor
    PORT(
         X : IN  std_logic;
         Y : IN  std_logic;
         D : OUT  std_logic;
         B : OUT  std_logic
        );
    END COMPONENT;
```

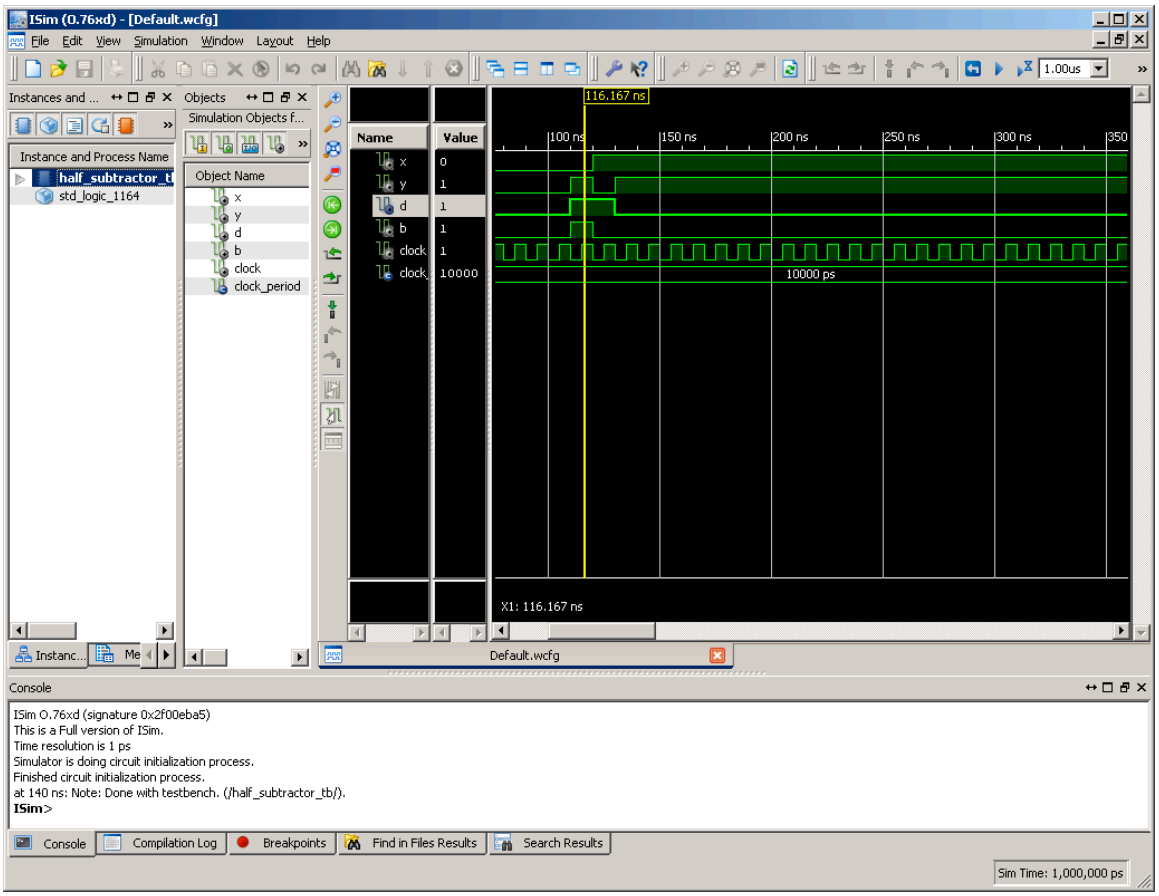
```
    --Inputs
    signal X : std_logic := '0';
    signal Y : std_logic := '0';

      --Outputs
    signal D : std_logic;
    signal B : std_logic;

      --Clock
      signal clock : std_logic;
    constant clock_period : time := 10 ns;

BEGIN

      -- Instantiate the Unit Under Test (UUT)
    uut: half_subtractor PORT MAP (
           X => X,
           Y => Y,
           D => D,
           B => B
         );

    -- Clock process definitions
    clock_process :process
    begin
           clock <= '0';
           wait for clock_period/2;
           clock <= '1';
           wait for clock_period/2;
    end process;


    -- Stimulus process
    stim_proc: process
    begin
           --Hold reset state
             X <= '0';
             Y <= '0';
       wait for clock_period*10;

       -- insert stimulus here
             --Cases
             X <= '0';
             Y <= '0';
             wait for clock_period;
             Assert ( D = '0' and B = '0' )
```

```
            Report "Case 0"
            Severity ERROR;

        X <= '0';
        Y <= '1';
        wait for clock_period;
        Assert ( D = '1' and B = '1' )
            Report "Case 1"
            Severity ERROR;

              X <= '1';
        Y <= '0';
        wait for clock_period;
        Assert ( D = '1' and B = '0' )
            Report "Case 2"
            Severity ERROR;

              X <= '1';
        Y <= '1';
        wait for clock_period;
        Assert ( D = '0' and B = '0' )
            Report "Case 3"
            Severity ERROR;

    Report "Done with testbench." severity NOTE;
  wait;
end process;

    END;
```

In the Processes for Source pane expand the ISim Simulator and double click the Behavioral Check Syntax. If correct then there will be no errors.



In the Processes for Source pane double click the Simulate Behavior Model. A new window will appear showing the simulation. If the test bench completes with no errors in the command prompt then simulation was successful for the tested cases. You might have to zoom out to view the signals in ns. Zoom out by pressing F7 and zoom in by pressing F8 or use the shortcuts on the top bar. **Note:** the testbench does not necessarily run to the final wait statement. You might need to run the test until you get the message "Note: Done with testbench".

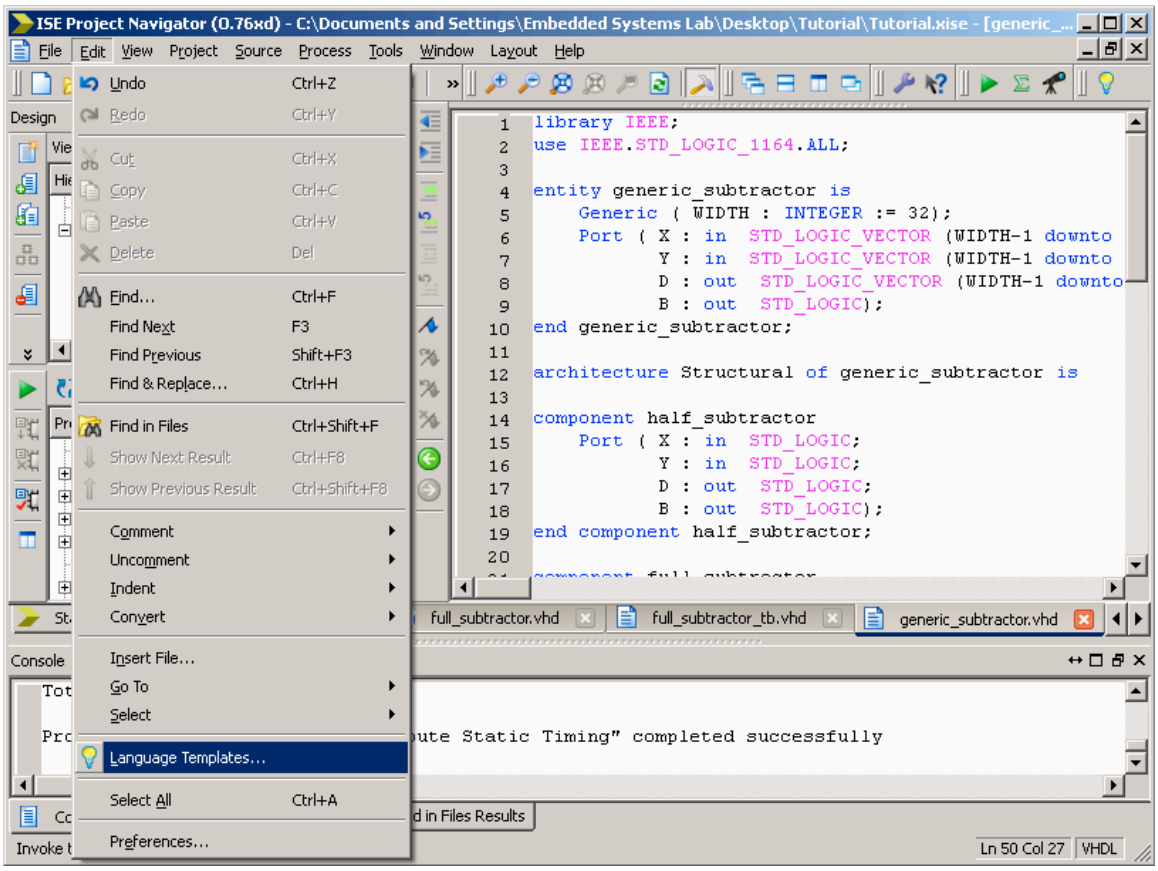


- You can also select the signals and change the display format from binary to hexadecimal by selecting radix hexadecimal. Other functionality includes: being able to add signals to the test bench from internal signals, by dragging signals/variables from the center pane to the right most pane. Other components can be explored by expanding and selecting object in the left most pane.

- Switch back to implementation view to continue editing.
- Repeat the processes above to add the full subtractor.

Step 6: VHDL templates

- Templates are a very useful resource for engineers. Unlike many other programming languages, only a subset of VHDL can be synthesized to hardware implementation. To access the language templates:
  go to Edit --> Language Templates.


ISE Project Navigator (0.76xd) - C:\Documents and Settings\Embedded Systems Lab\Desktop\Tutorial\Tutorial.xise - [generic_...
File Edit View Project Source Process Tools Window Layout Help
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Find... Ctrl+F
Find Next F3
Find Previous Shift+F3
Find & Replace... Ctrl+H
Find in Files Ctrl+Shift+F
Show Next Result Ctrl+F8
Show Previous Result Ctrl+Shift+F8
Comment
Uncomment
Indent
Convert
Insert File...
Go To
Select
Language Templates...
Select All Ctrl+A
Preferences...
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity generic_subtractor is
Generic (WIDTH : INTEGER := 32);
Port (X : in STD_LOGIC_VECTOR (WIDTH-1 downto
Y : in STD_LOGIC_VECTOR (WIDTH-1 downto
D : out STD_LOGIC_VECTOR (WIDTH-1 downto
B : out STD_LOGIC);
end generic_subtractor;
architecture Structural of generic_subtractor is
component half_subtractor
Port (X : in STD_LOGIC;
Y : in STD_LOGIC;
D : out STD_LOGIC;
B : out STD_LOGIC);
end component half_subtractor;
full_subtractor.vhd
full_subtractor_tb.vhd
generic_subtractor.vhd
Console
ute Static Timing" completed successfully
Ln 50 Col 27
VHDL

In viewing the template look only at VHDL > Synthesis Constructs > Coding Examples when design hardware. These templates will give you an intuition for how to create hardware structures.