

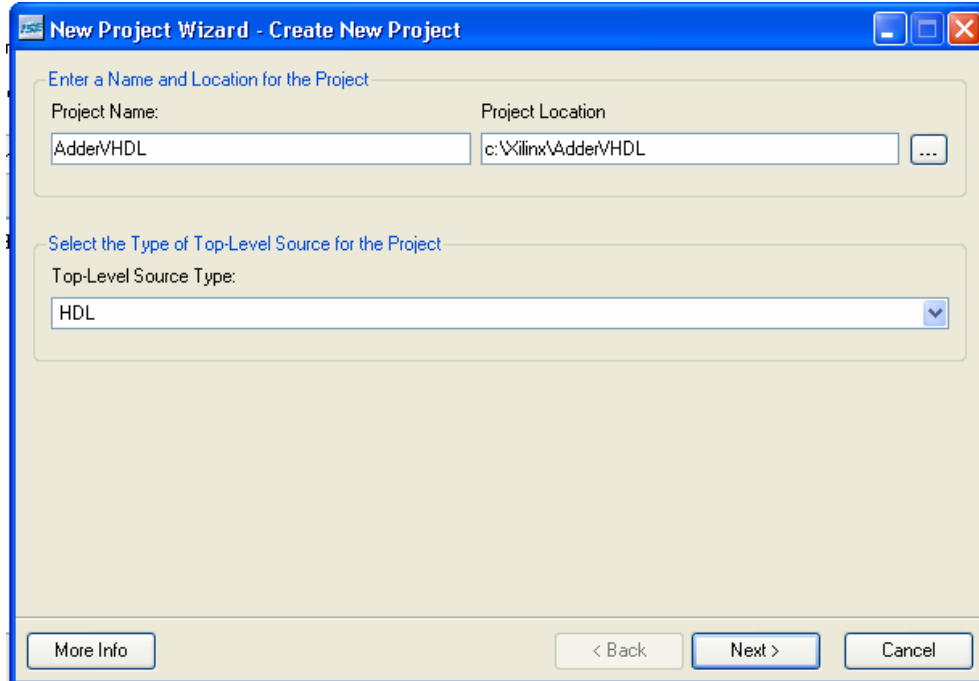
Xilinx VHDL

<Release Version: 8.2i>

Tutorial

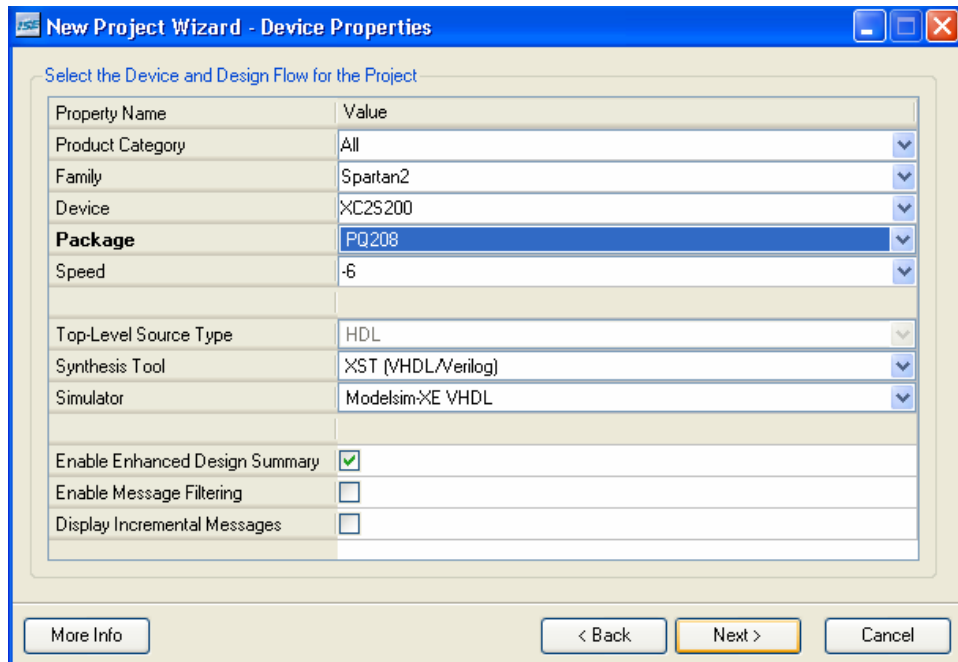
Department of Electrical and Computer Engineering
State University of New York – New Paltz

Start A New Project.



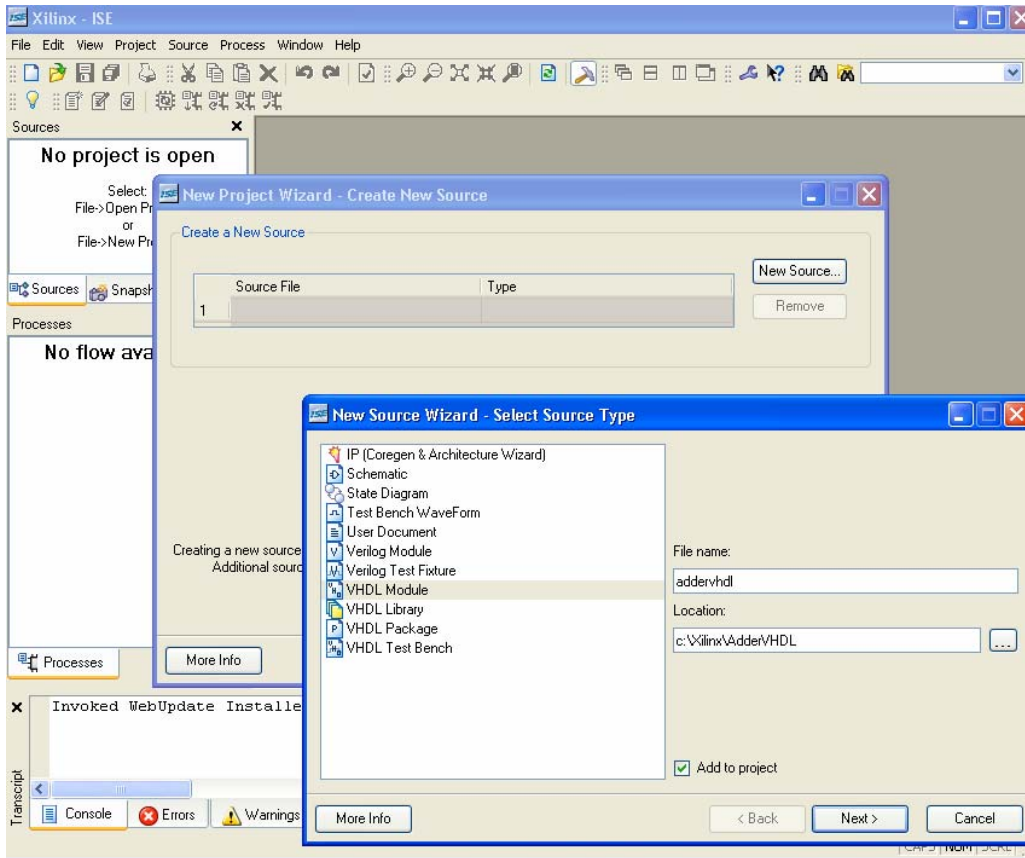
Click Next.

Make Sure the Device Properties are chosen as shown below.

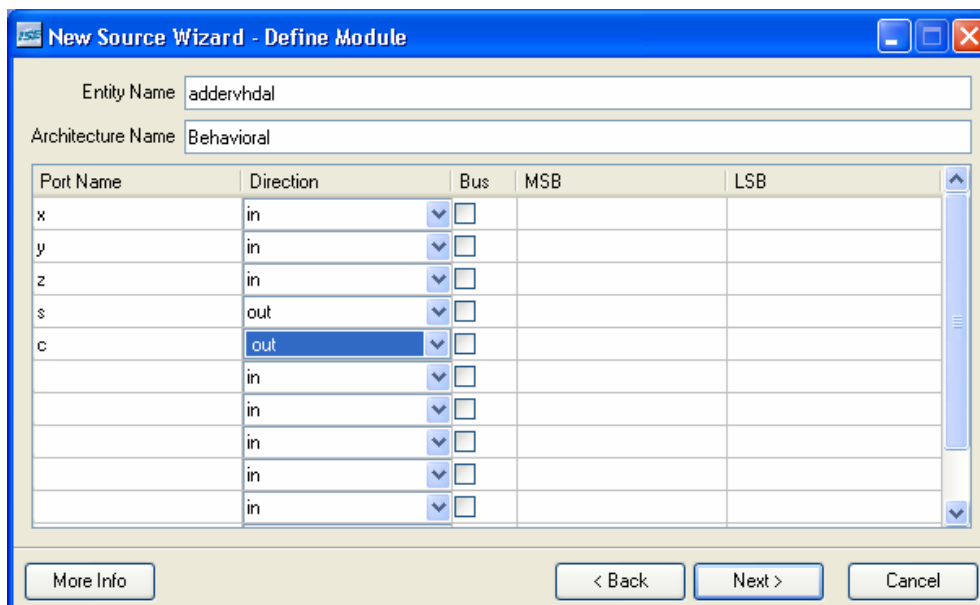


Click Next.

On the New Project Wizard, click on New Source. In New Source Wizard, select VHDL module and type in a filename.



Click "Next". In this exercise, you are designing a full adder with x, y, and z as inputs and s and c as outputs. Hence, set the ports accordingly and click Next.



Click “Next”, “Next” and “Finish”. This will open a editor where you can input your VHDL code.

```
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    19:52:49 11/07/2006
6  -- Design Name:
7  -- Module Name:   addervhdl - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library declaration if instantiating
26 ---- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity addervhdl is
31     Port ( x : in  STD_LOGIC;
32           y : in  STD_LOGIC;
33           z : in  STD_LOGIC;
34           s : out STD_LOGIC;
35           c : out STD_LOGIC);
36 end addervhdl;
37
38 architecture Behavioral of addervhdl is
39
40 begin
41
42
43 end Behavioral;
44
45
```

Note that

$$S = X \oplus Y \oplus Z$$

$$C = XY + YZ + XZ$$

Hence,

```

19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library declaration if instantiating
26 ---- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity addervhdal is
31     Port ( x : in  STD_LOGIC;
32           y : in  STD_LOGIC;
33           z : in  STD_LOGIC;
34           s : out STD_LOGIC;
35           c : out STD_LOGIC);
36 end addervhdal;
37
38 architecture Behavioral of addervhdal is
39
40 begin
41
42     s <= x xor y xor z;
43     c <= (x and y) or (y and z) or (x and z);
44
45 end Behavioral;
46
47

```

Save the file.

The project can be simulated using ModelSim or ISE simulator. For ISE simulator details refer the ISE Simulator tutorial. ModelSim simulator follows a similar procedure as outlined in ModelSim tutorial. Try to use ModelSim to simulate your design.

When the design is completed, open the User Constraints Editor and assign the pins to the correct inputs and outputs. Follow the steps in the Download Tutorial to complete the process. Don't forget to connect the Enable pin to Vcc.