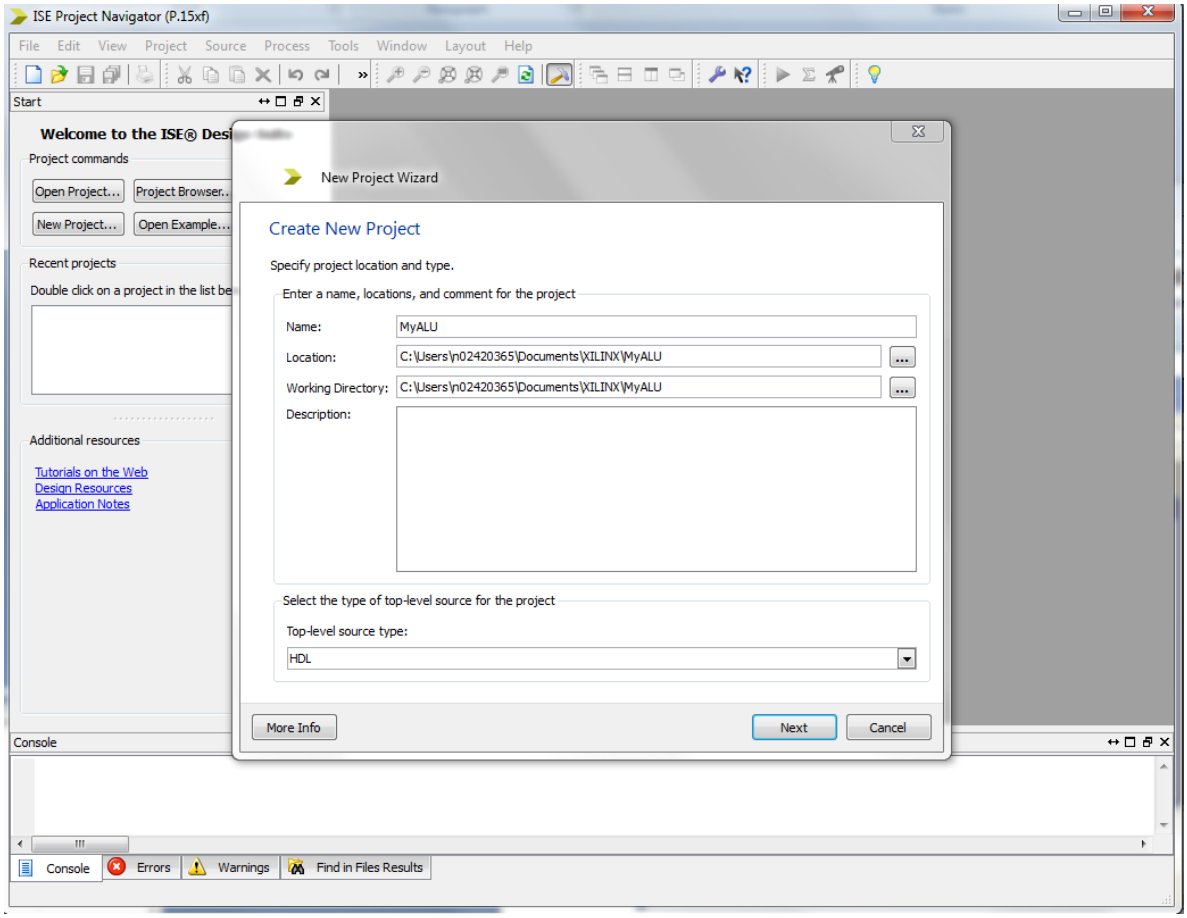
Xilinx VHDL

< Design Suite Version: 14.1>

Tutorial

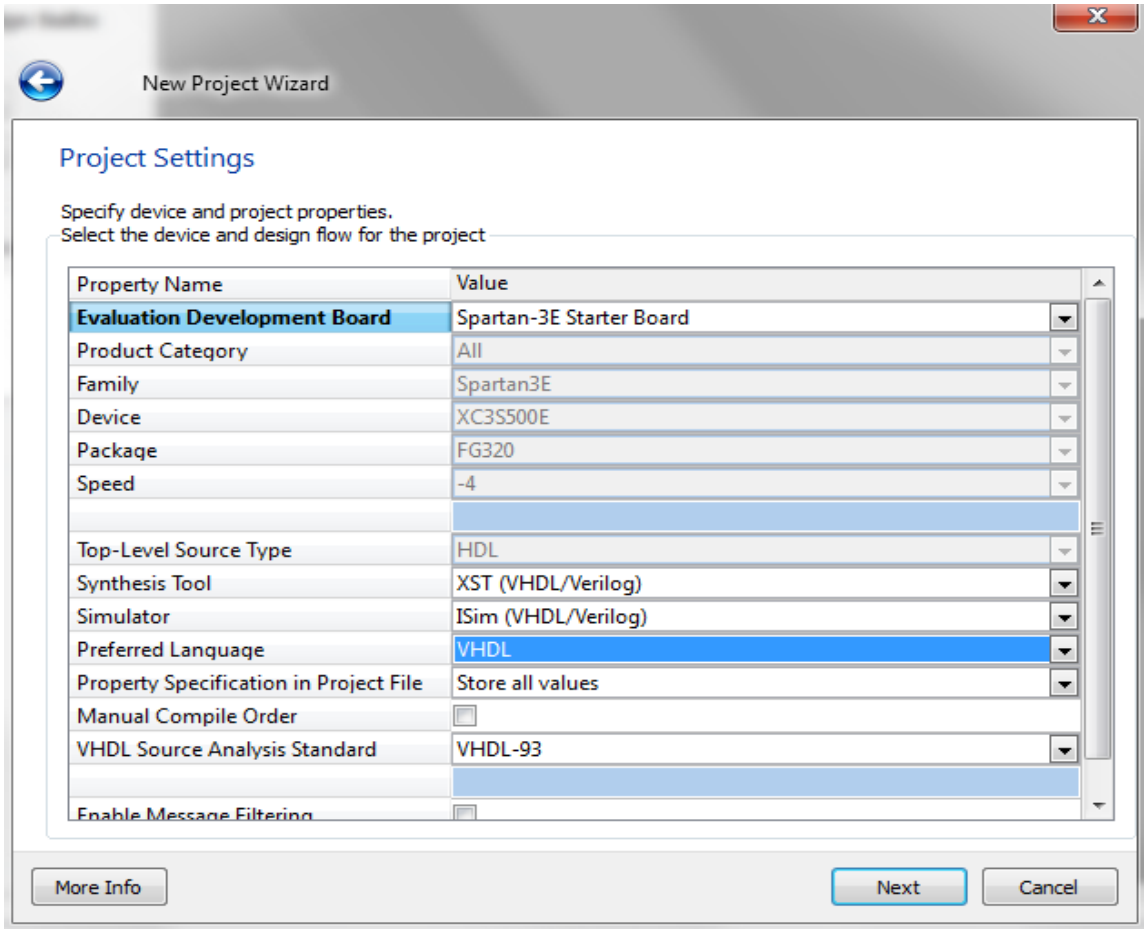
Department of Electrical and Computer Engineering
State University of New York – New Paltz

1. Start A New Project.



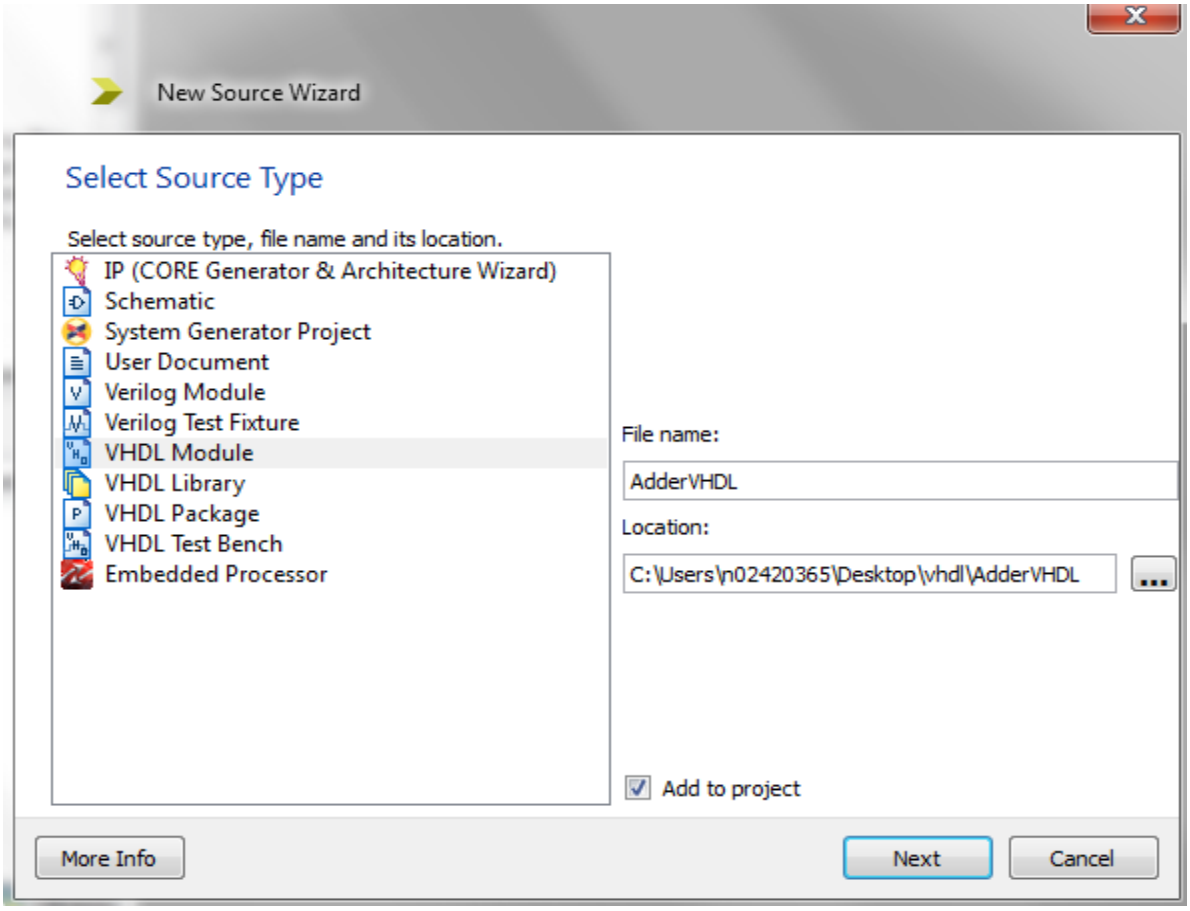
Click Next.

Make Sure the Device Properties are chosen as shown below.



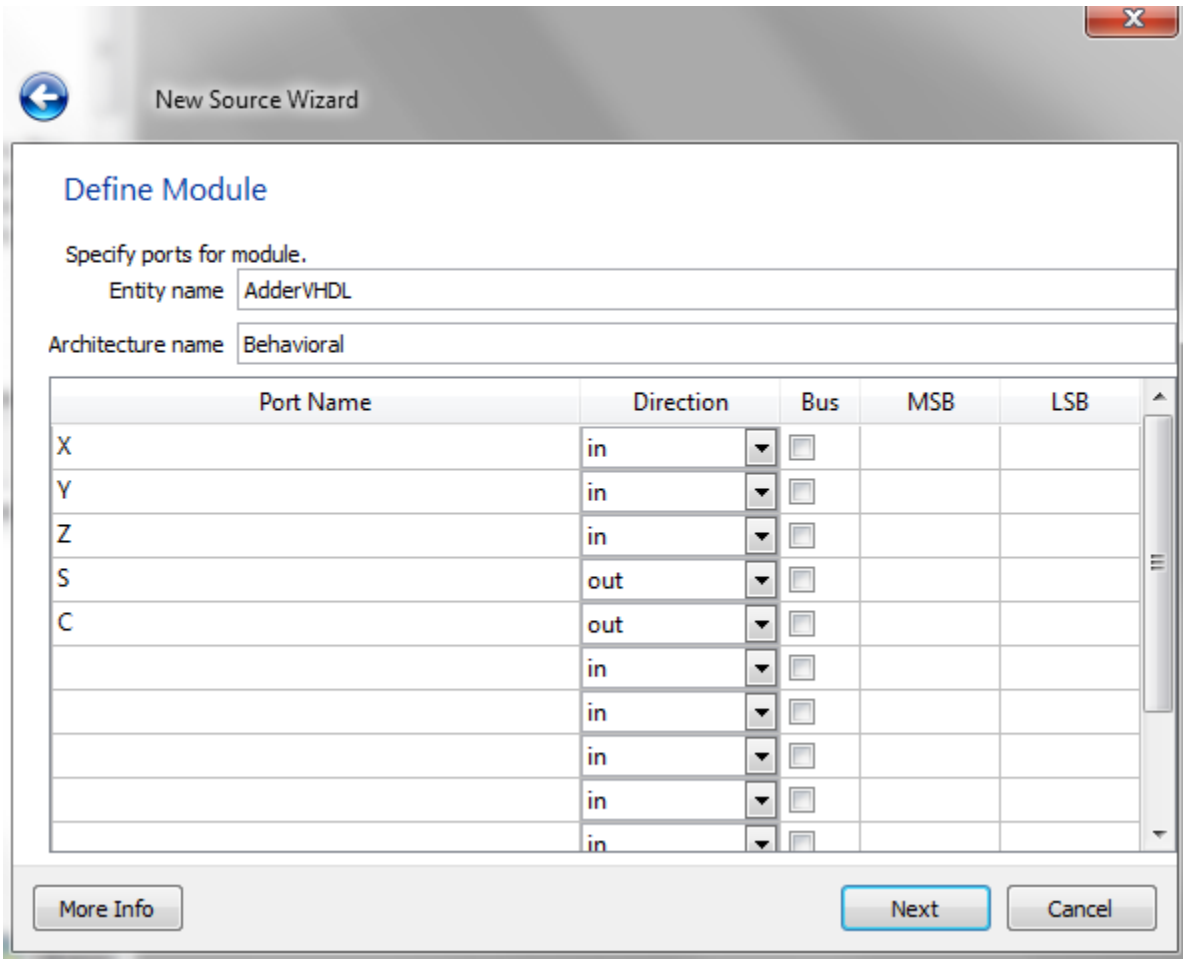
Click Next.
Click finish.

2. Go to **Project** <- **New Source**
On the New Source Wizard, click on **VHDL module** and type a filename.



Click "Next".
Click "Finish".

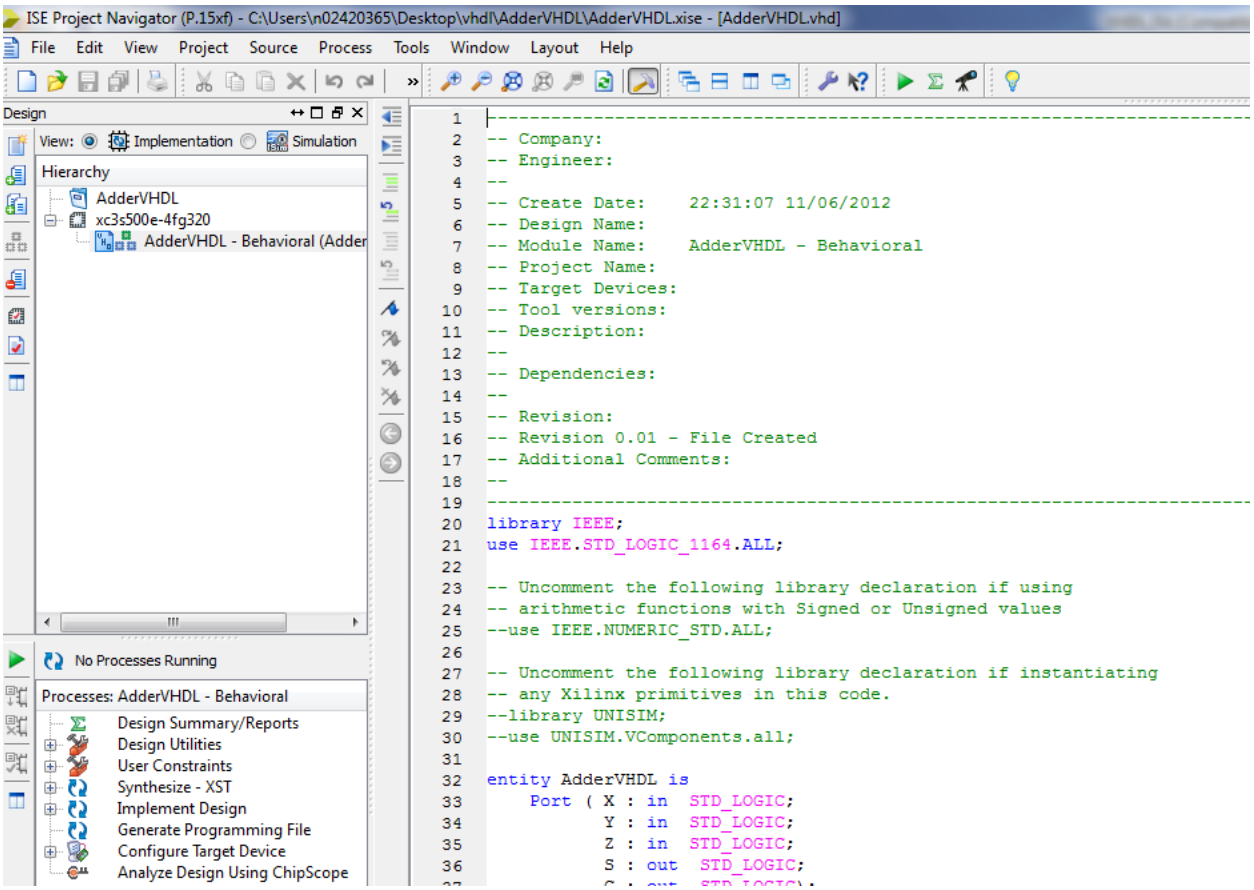
3. In this exercise, you are designing a full adder with X, Y, and Z as inputs and S and C as outputs. Hence, set the ports accordingly.



Click "**Next**"

Click "**Finish**"

4. This will open the editor where you can input your VHDL code.



Note that

$S = X \oplus Y \oplus Z$
$C = XY + YZ + XZ$

Hence,

```
----------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity AdderVHDL is
    Port ( X : in  STD_LOGIC;
           Y : in  STD_LOGIC;
           Z : in  STD_LOGIC;
           S : out  STD_LOGIC;
           C : out  STD_LOGIC);
end AdderVHDL;

architecture Behavioral of AdderVHDL is

begin
S <= X XOR Y XOR Z;
C <= (X AND Y) OR (Y AND Z) OR (X AND Z);

end Behavioral;
```

5. Save the file.

The project can be simulated using ISE simulator. For ISE simulator details refer the ISE Simulator tutorial.

Make sure that for the **testbench** in the auto generated **".vhd"** delete all the clock signal lines(or you can also make them as comments) and follow the same procedure as in the ISE simulator tutorial.

```
60     signal C : std_logic;
61     -- No clocks detected in port list. Replace <clock> below with
62     -- appropriate port name
63
64     constant <clock>_period : time := 10 ns;
65
66  BEGIN
67
68     -- Instantiate the Unit Under Test (UUT)
69     uut: AdderVHDL PORT MAP (
70            X => X,
71            Y => Y,
72            Z => Z,
73            S => S,
74            C => C
75          );
76
77     -- Clock process definitions
78     <clock>_process :process
79     begin
80        <clock> <= '0';
81        wait for <clock>_period/2;
82        <clock> <= '1';
83        wait for <clock>_period/2;
84     end process;
85
86
87     -- Stimulus process
88     stim_proc: process
89     begin
90        -- hold reset state for 100 ns.
91        wait for 100 ns;
92
93        wait for <clock>_period*10;
94
95        -- insert stimulus here
96
97        wait;
98     end process;
99
100 END;
101
```

```
   signal testvar : std_logic_vector (2 downto 0) := "000";
   --Outputs
   signal S : std_logic;
   signal C : std_logic;
   -- No clocks detected in port list. Replace <clock> below with
   -- appropriate port name

   --constant <clock>_period : time := 10 ns;
BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: AdderVHDL PORT MAP (
          X => X,
          Y => Y,
          Z => Z,
          S => S,
          C => C
        );

   -- 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 for 100 ns.

      --wait for <clock>_period*10;

      -- insert stimulus here

      wait;
   end process;
```

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.