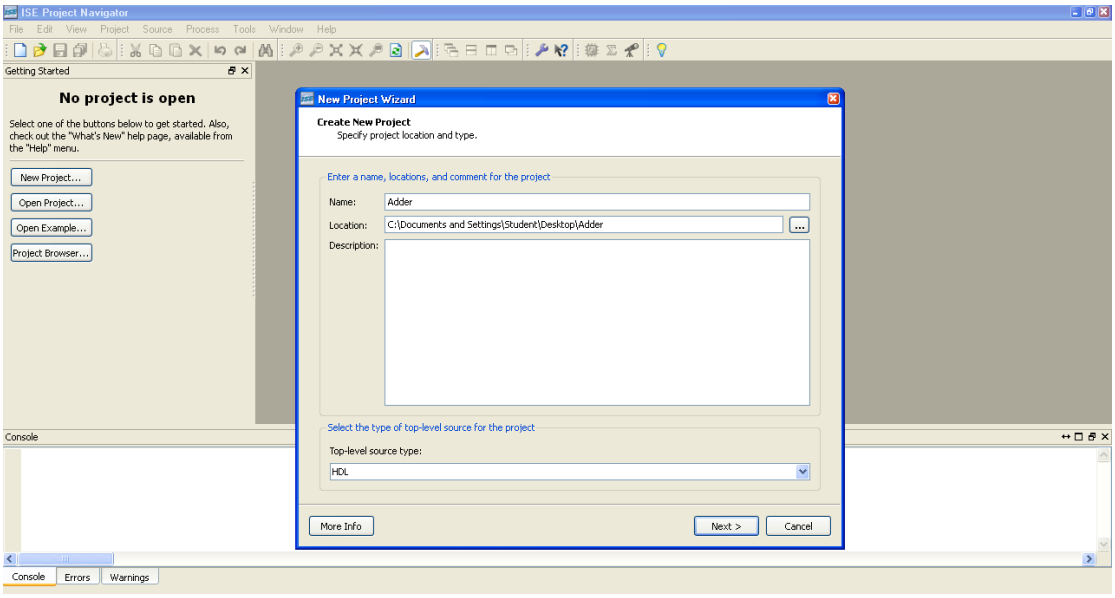
Xilinx ISE Tutorial:

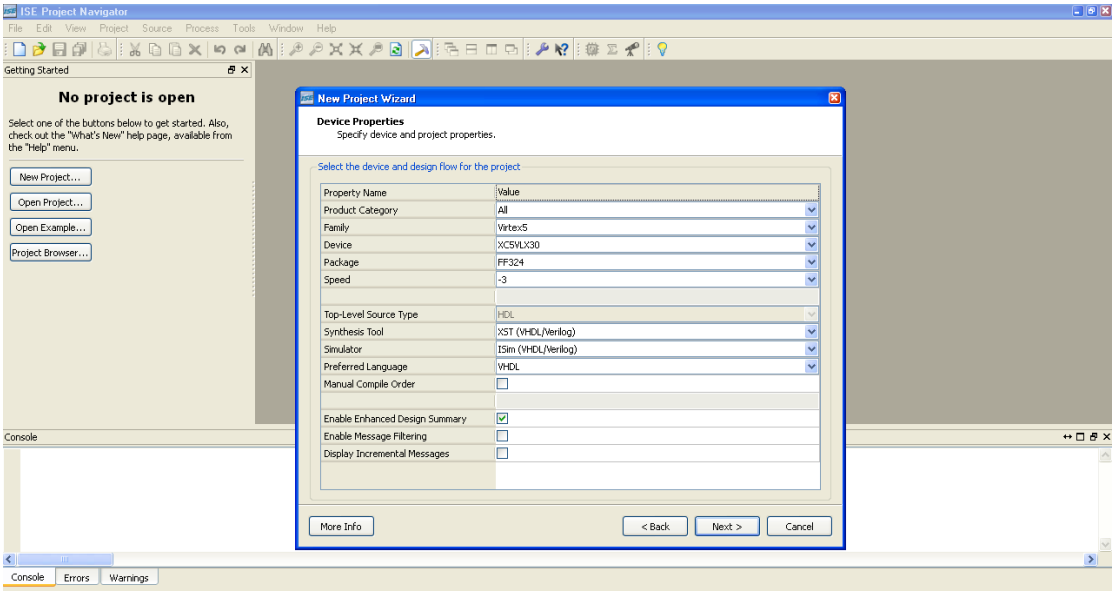
This tutorial provides a step-by-step guide to simulating a VHDL description of a *Full Adder* using Xilinx ISE 11.1 –11.4.

Step 1:

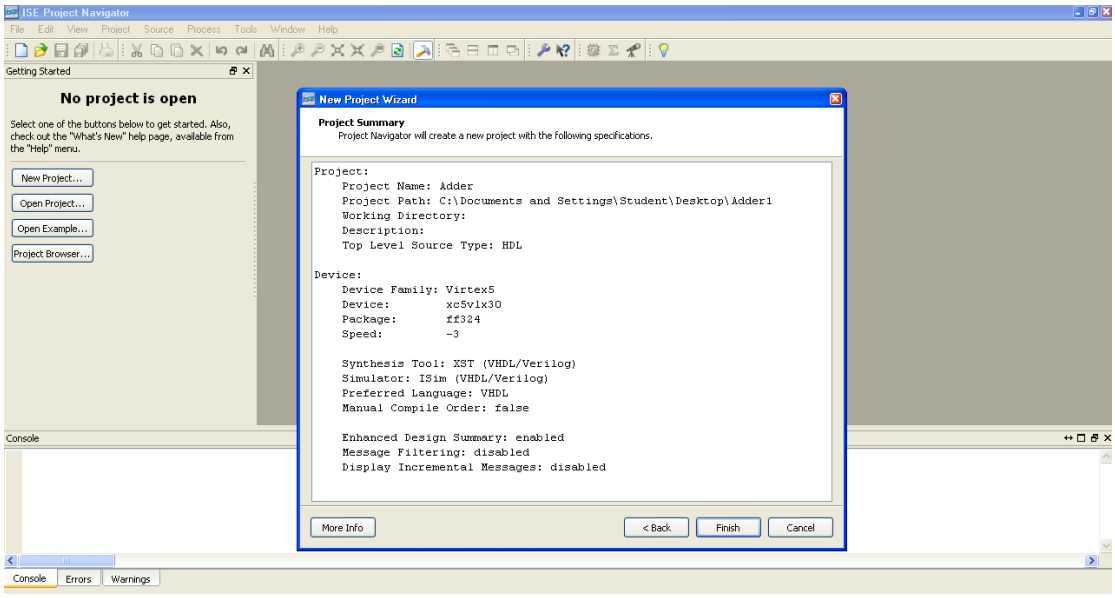
- Open ISE, select "new project"
- Specify your project name and location.
- Leave the working directory entry blank.
- Choose *HDL* as the source type from the *Top-Level Source Type* menu.
- Click *Next* button.



Step 2: Click "Next", use the default configuration:

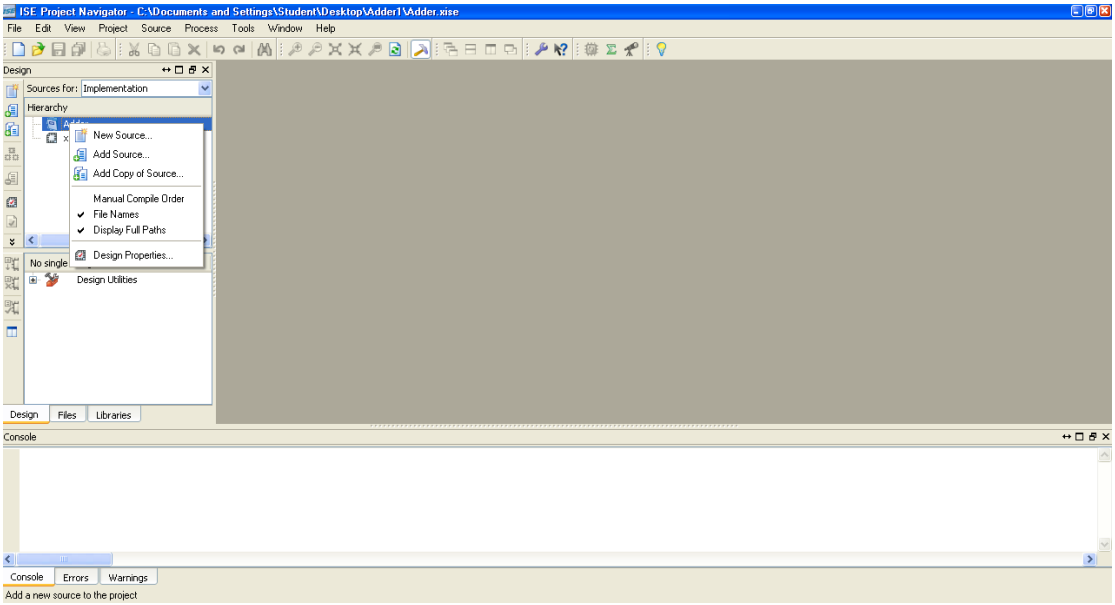


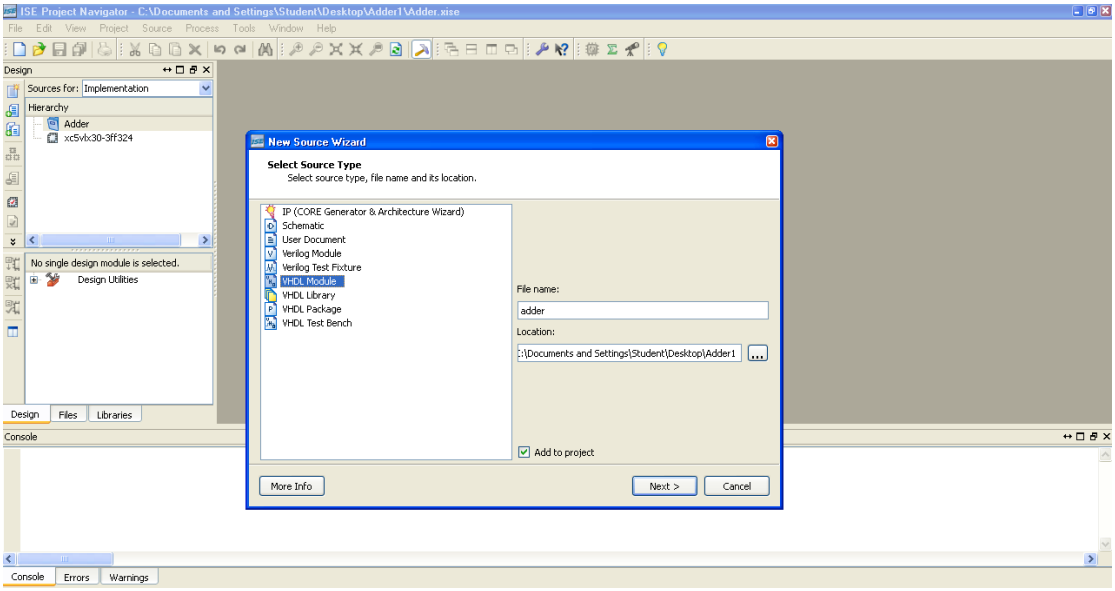
Step 3: Click "Next" and then a project summary will appear. Click on the *Finish* button.



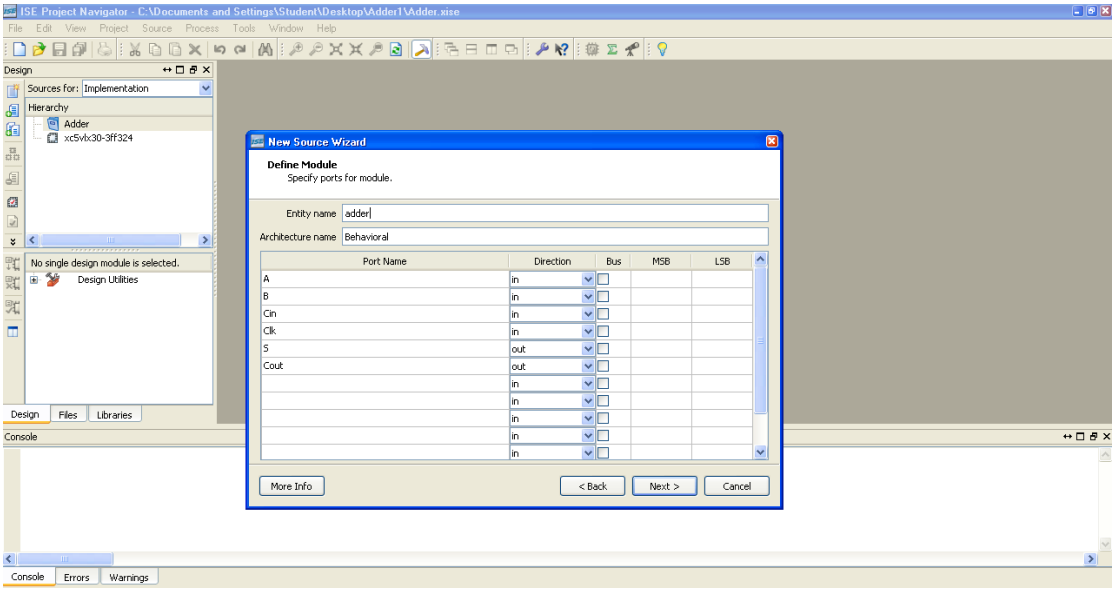
Step 4: Now we want to add a new file to our project.

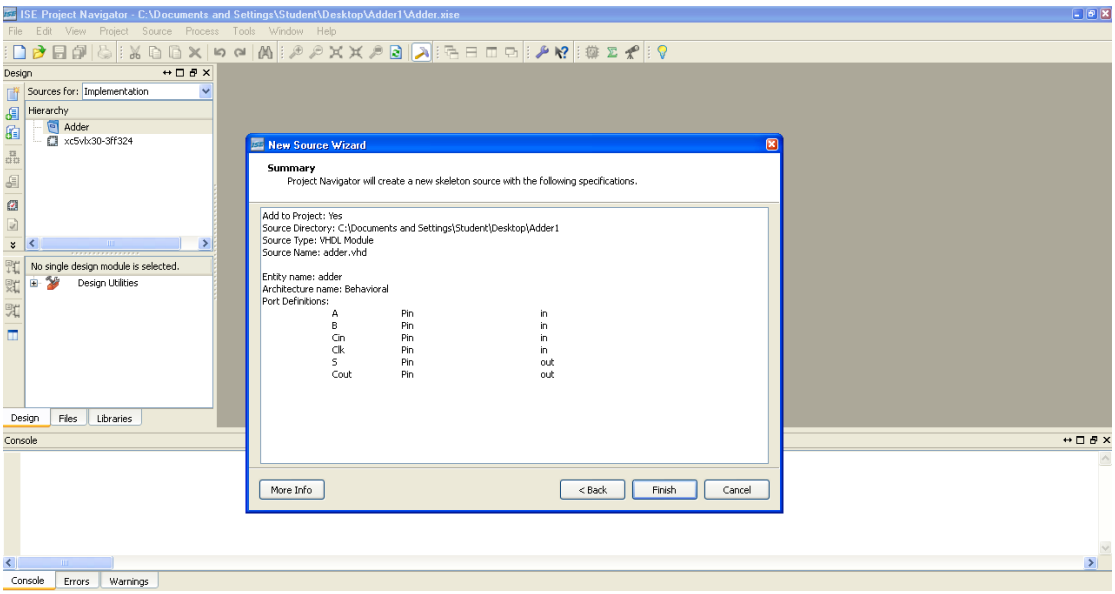
- Right Click on *your project name*, choose *New Source.*
- Choose *VHDL Module* as the file type.
- In the *File name:* box enter the desired file name, in this case the file is named "adder".
- Click on the *Next* button.





Step 5: Specify the *input* and *output.* Then Click *Fnish.*



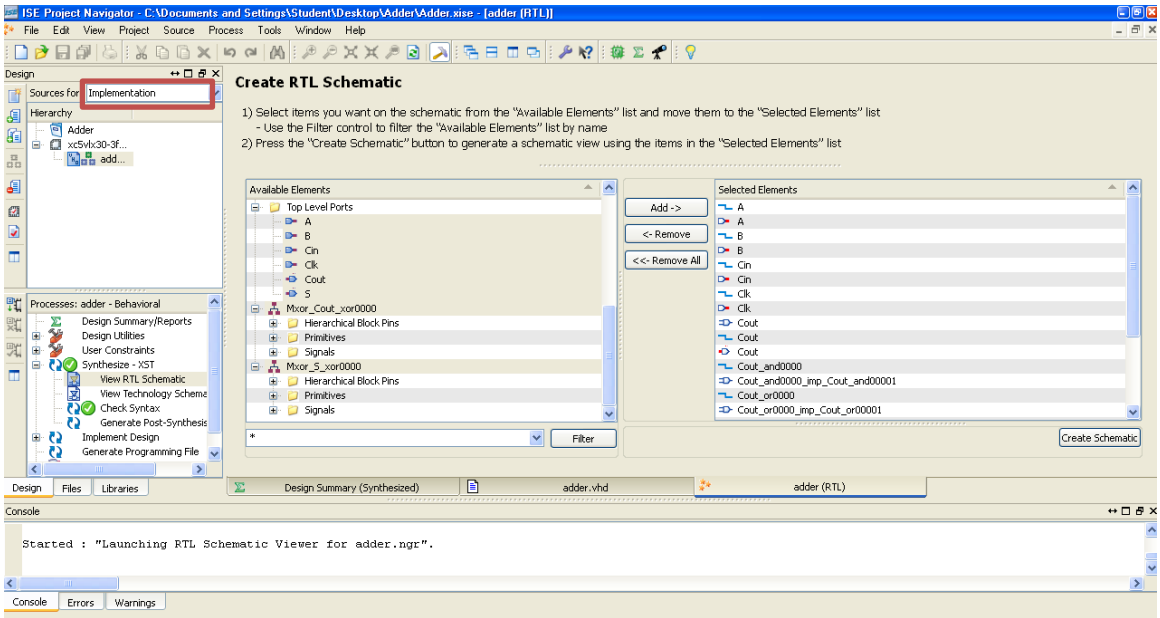


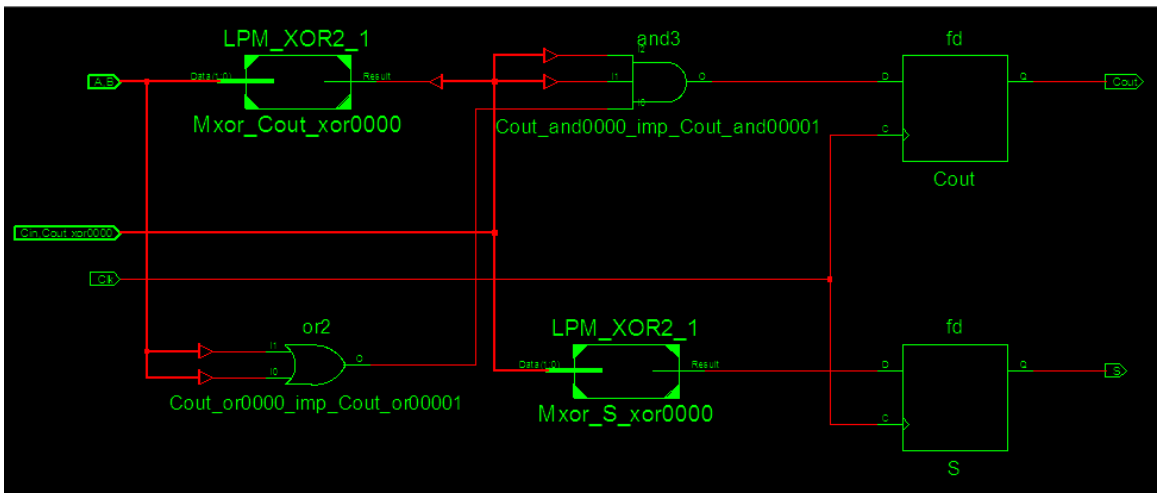
Step 6: Fill your *VHDL code:*

```
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
```

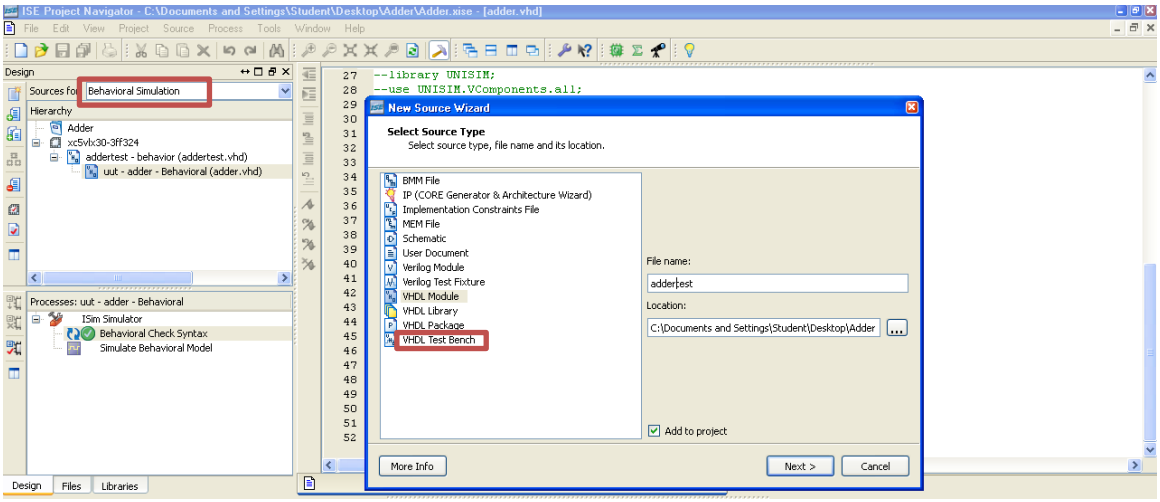
```
30  entity adder is
31      Port ( A : in  STD_LOGIC;
32             B : in  STD_LOGIC;
33             Cin : in  STD_LOGIC;
34             Clk : in  STD_LOGIC;
35             S : out  STD_LOGIC;
36             Cout : out  STD_LOGIC);
37  end adder;
38
39  architecture Behavioral of adder is
40
41  begin
42     Process(Clk)
43     begin
44        if rising_edge(Clk) then
45           S <= A xor B xor Cin;
46           Cout <= (A or B) and (Cin and (A xor B));
47        end if;
48     end Process;
49
50  end Behavioral;
51
52
```

Step 7: Choose *Implementation*, and *Synthesize – XST*. You can use View RTL Schematic to review the full adder's RTL Schematic (Right Click and choose run or double click). Then use shift to select all. After all, choose *Create Schematic*:





Step 8: Choose *Behavioral*, and right click and choose new source and select *VHDL Test Bench*. Specify the file name and the location.



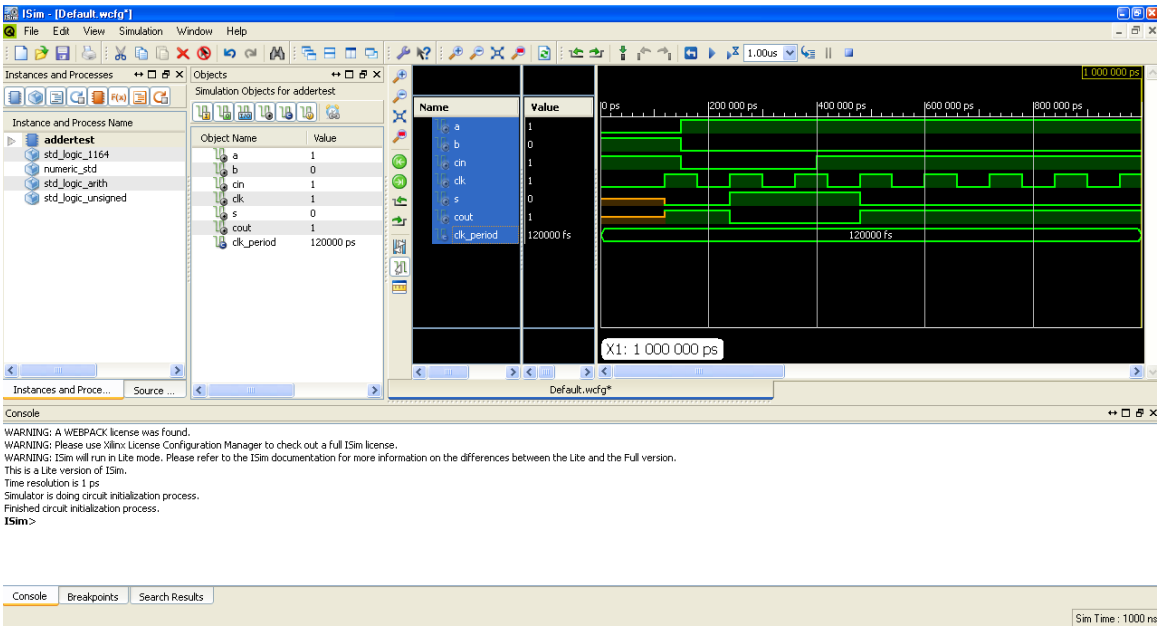
Step 10: Fill the code in the addertest.vhd:

```
28  LIBRARY ieee;
29  USE ieee.std_logic_1164.ALL;
30  USE ieee.std_logic_unsigned.all;
31  USE ieee.numeric_std.ALL;
32
33  ENTITY addertest IS
34  END addertest;
35
36  ARCHITECTURE behavior OF addertest IS
37
38      -- Component Declaration for the Unit Under Test (UUT)
39
40      COMPONENT adder
41      PORT(
42           A : IN  std_logic;
43           B : IN  std_logic;
44           Cin : IN  std_logic;
45           Clk : IN  std_logic;
46           S : OUT  std_logic;
47           Cout : OUT  std_logic
48          );
49      END COMPONENT;
50
51
52     --Inputs
53     signal A : std_logic := '0';
54     signal B : std_logic := '0';
55     signal Cin : std_logic := '0';
56     signal Clk : std_logic := '0';
57
58     --Outputs
59     signal S : std_logic;
60     signal Cout : std_logic;
61
62     constant clk_period : time := 120 ns;
63
64  BEGIN
65
66     -- Instantiate the Unit Under Test (UUT)
67     uut: adder PORT MAP (
68            A => A,
69            B => B,
70            Cin => Cin,
71            Clk => Clk,
72            S => S,
73            Cout => Cout
74          );
75
76     -- No clocks detected in port list. Replace <clock> below with
77     -- appropriate port name
```

```
78
79      clk_process : process
80      begin
81          wait for clk_period/2;
82          Clk <= '0';
83          wait for clk_period/2;
84          Clk <= '1';
85      end process;
86
87
88      -- Stimulus process
89      stim_proc: process
90      begin
91          A <= '0';
92          B <= '1';
93          Cin <= '1';
94          wait for 150ns;
95          A <= '1';
96          B <= '0';
97          Cin <= '0';
98          wait for 250ns;
99          A <= '1';
100         B <= '0';
101         Cin <= '1';
102         wait;
```

```
103     end process;
104
105 END;
106
```

Step 11: Choose *ISim Simulator*. Double click *Behavior Check Syntax* and *Simulate Behavioral Model*:



Note: Once the code in addertest.vhd (the test file in Behavioral Simulation) is **modified**, you should **Synthesize** all the code again so that the test file can output the correct waveform.