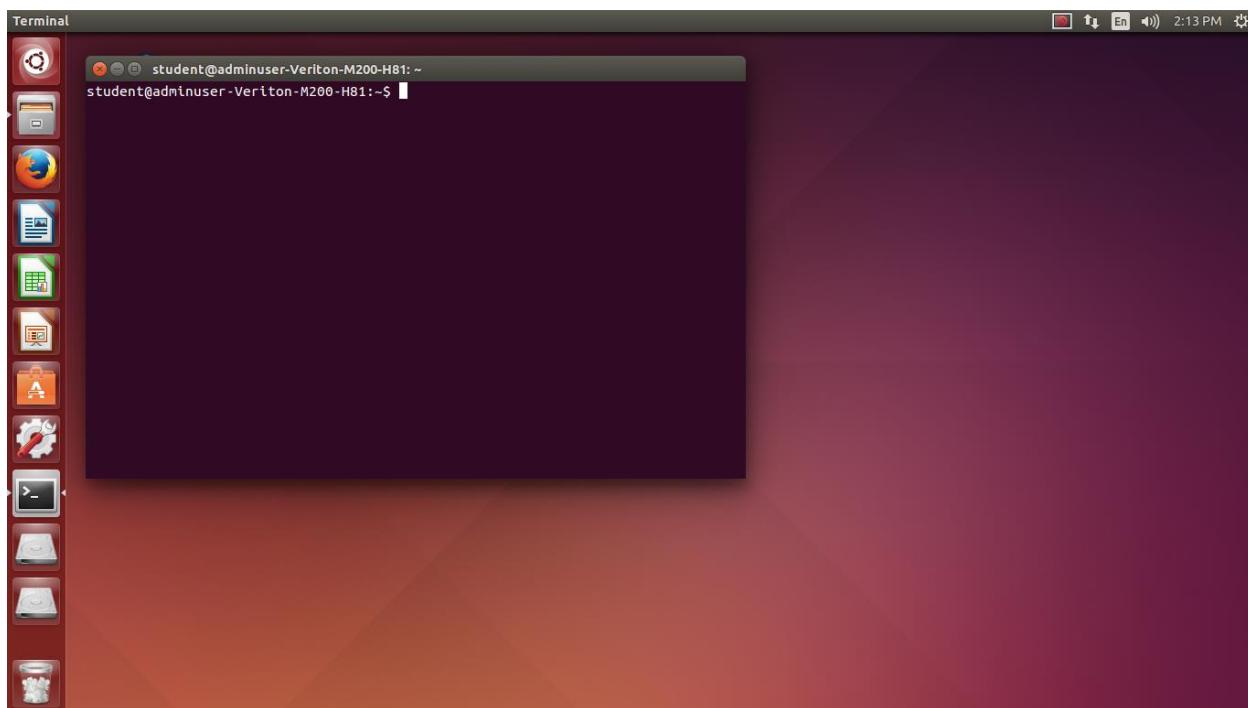


## STEPS TO IMPLEMENT THE DESIGN

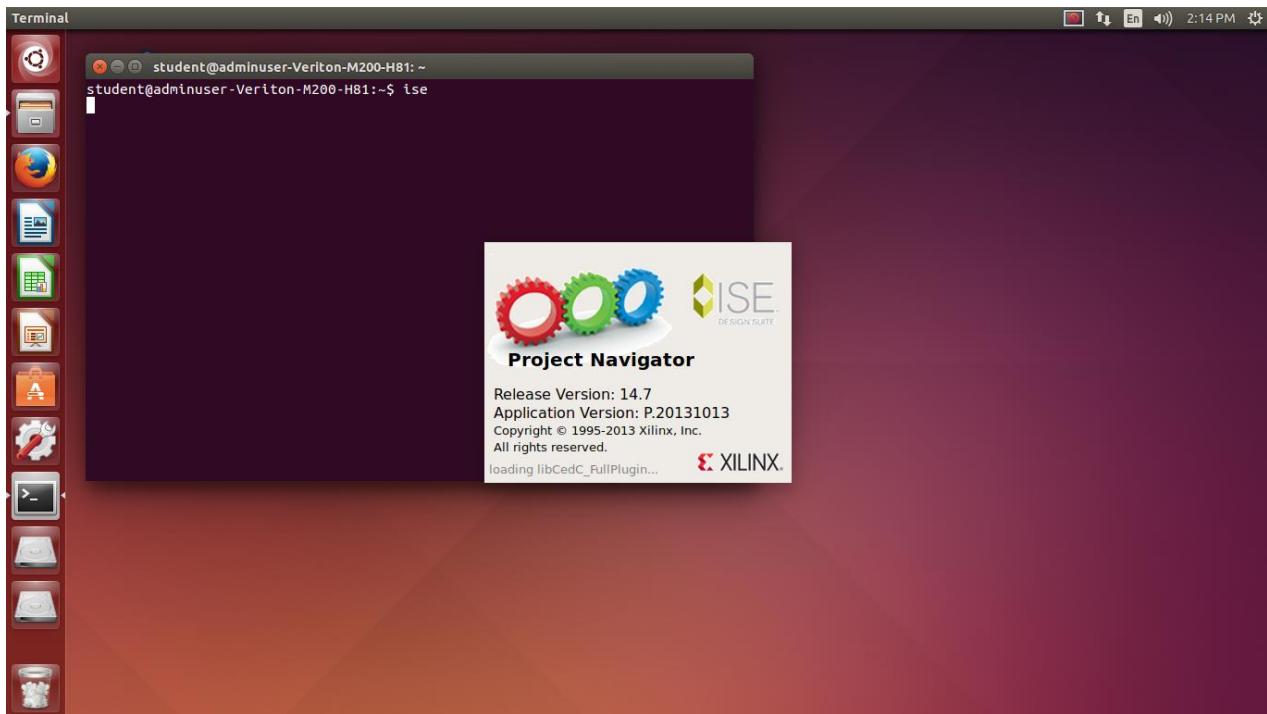
**Step 1:** Search for “terminal” in the search bar. Once terminal is shown in the results click on it.



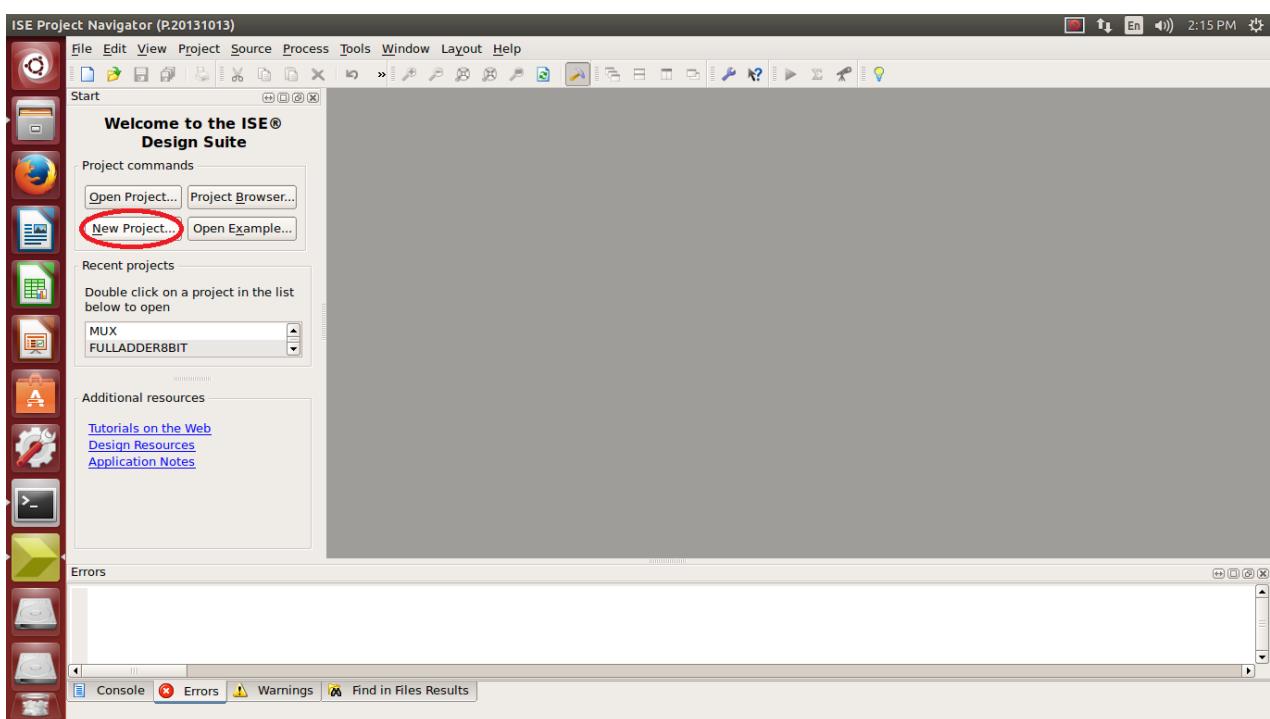
The terminal window opens.



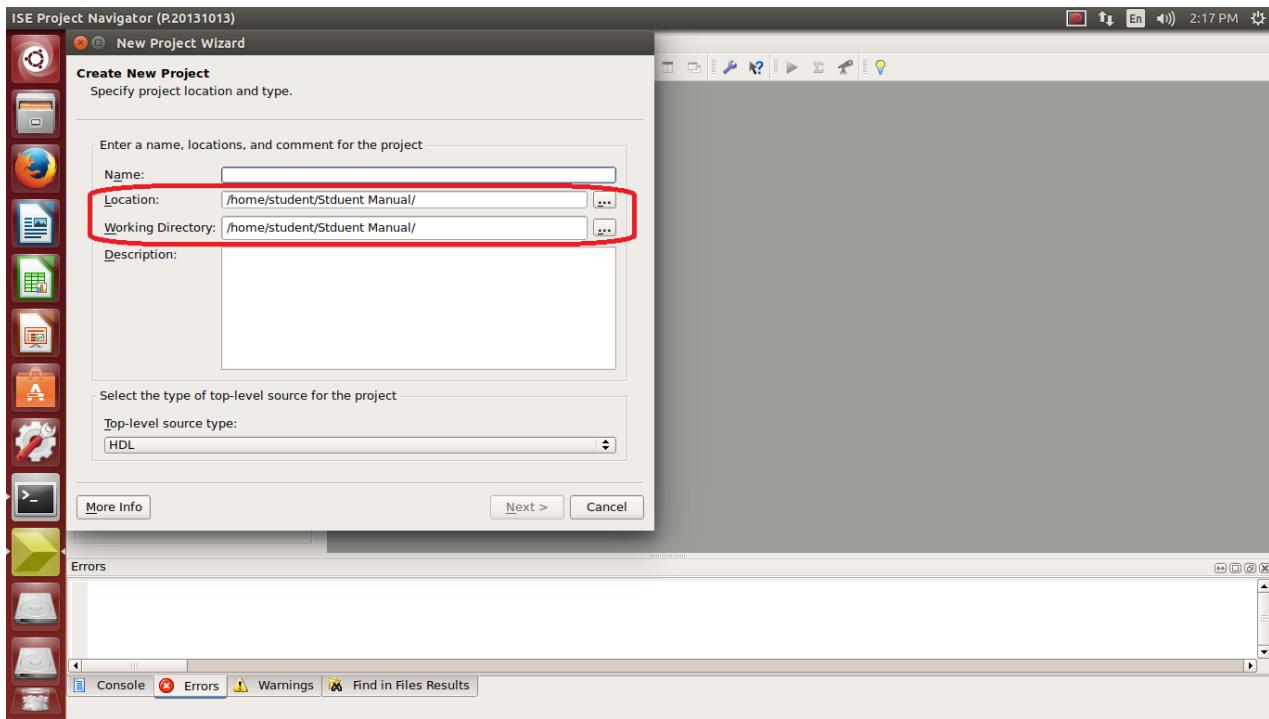
**Step 2:** Type “**ise**” in the terminal box and press enter. This will open the **Xilinx** window.



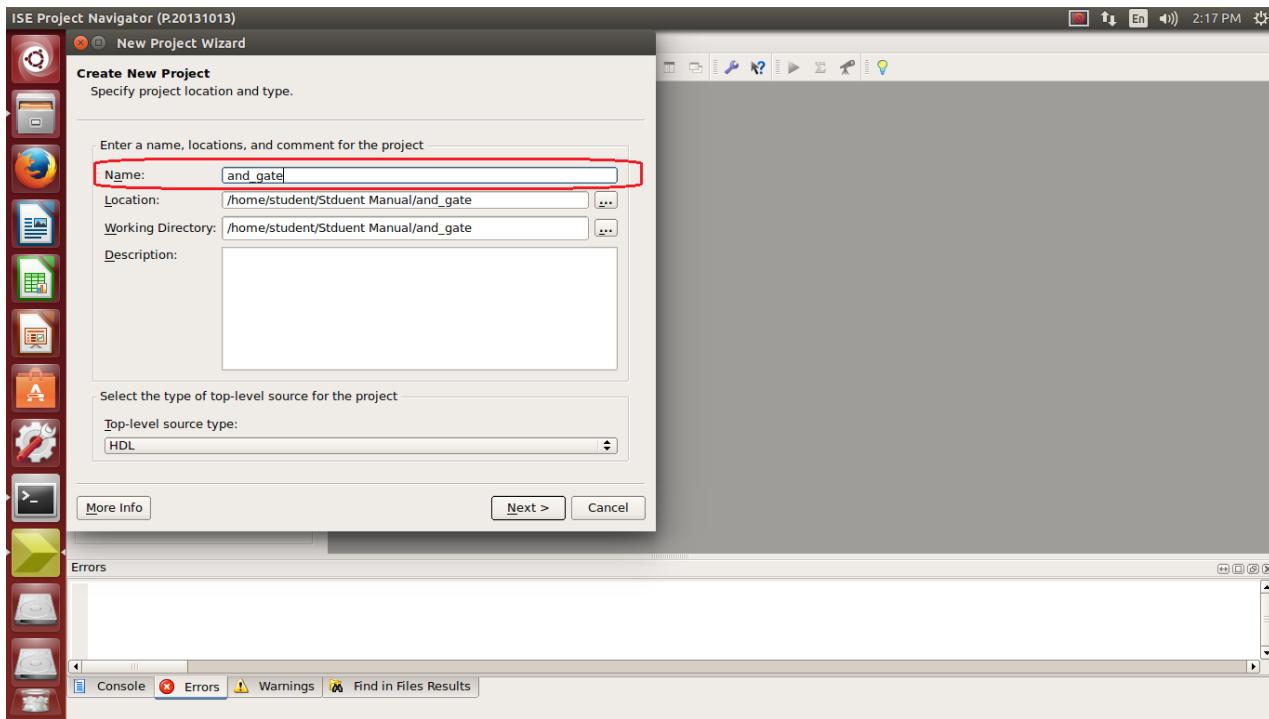
**Step 3:** Once the Xilinx window opens click on **New Project**.



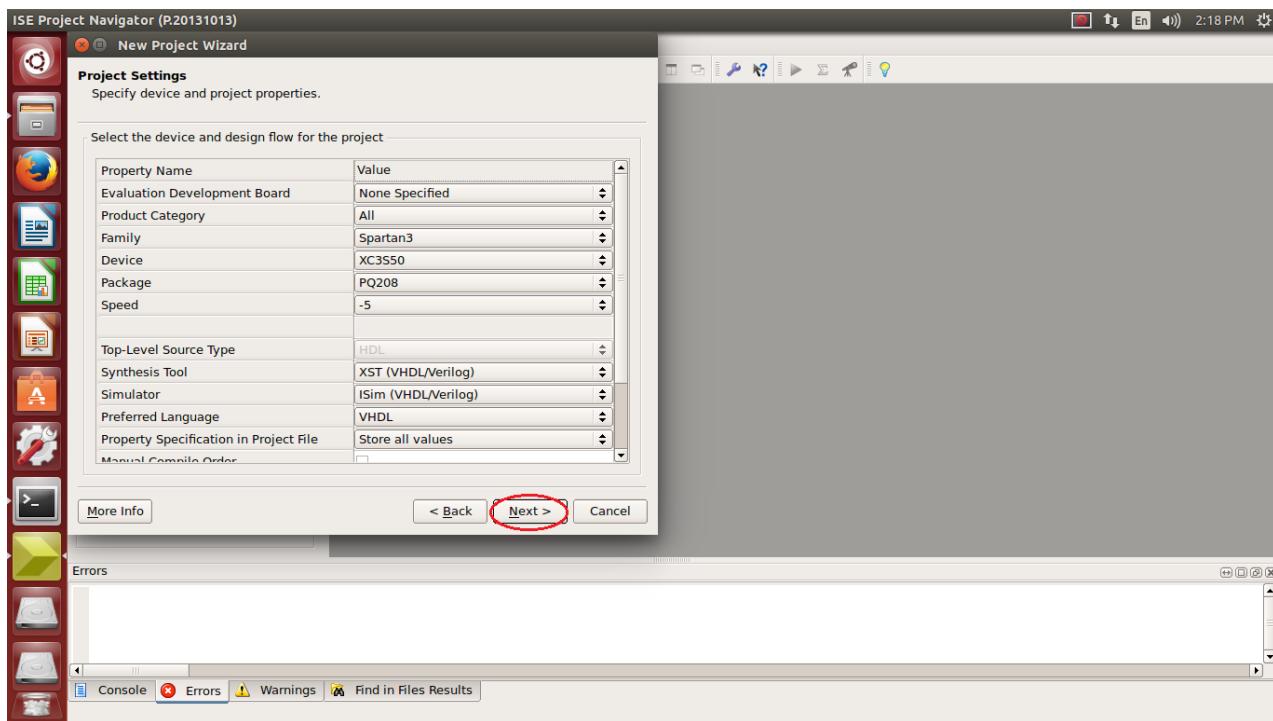
**Step 4:** The Create New Project window will appear. Select the *location* where the project should be saved.



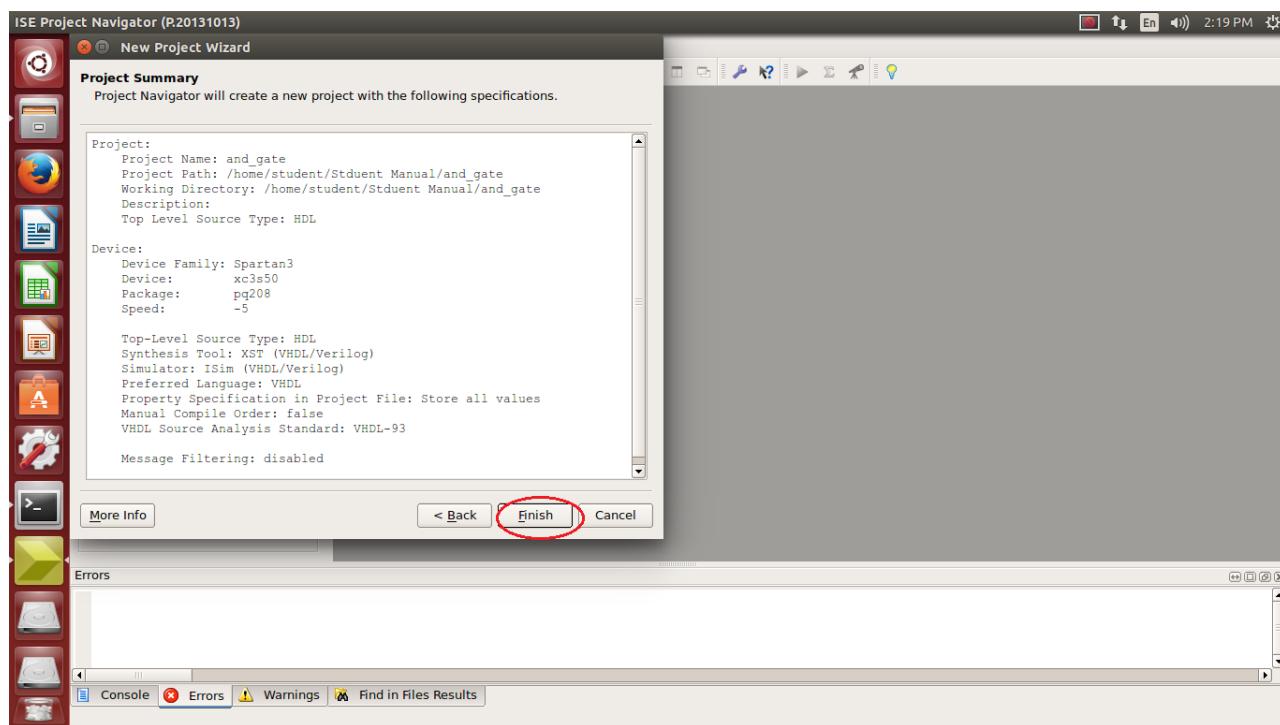
**Step 5:** Enter a name for the project and click on **Next**.



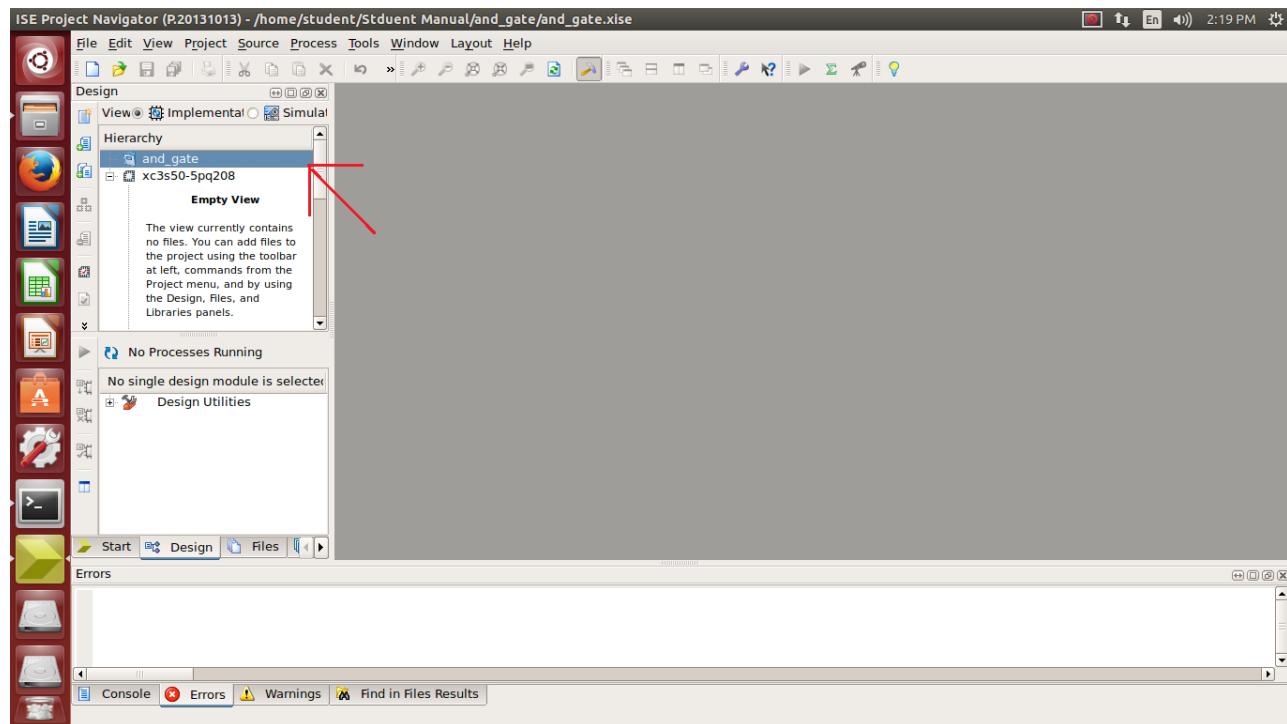
**Step 6:** The Project Settings window is shown. Make the necessary changes and click on **Next**.



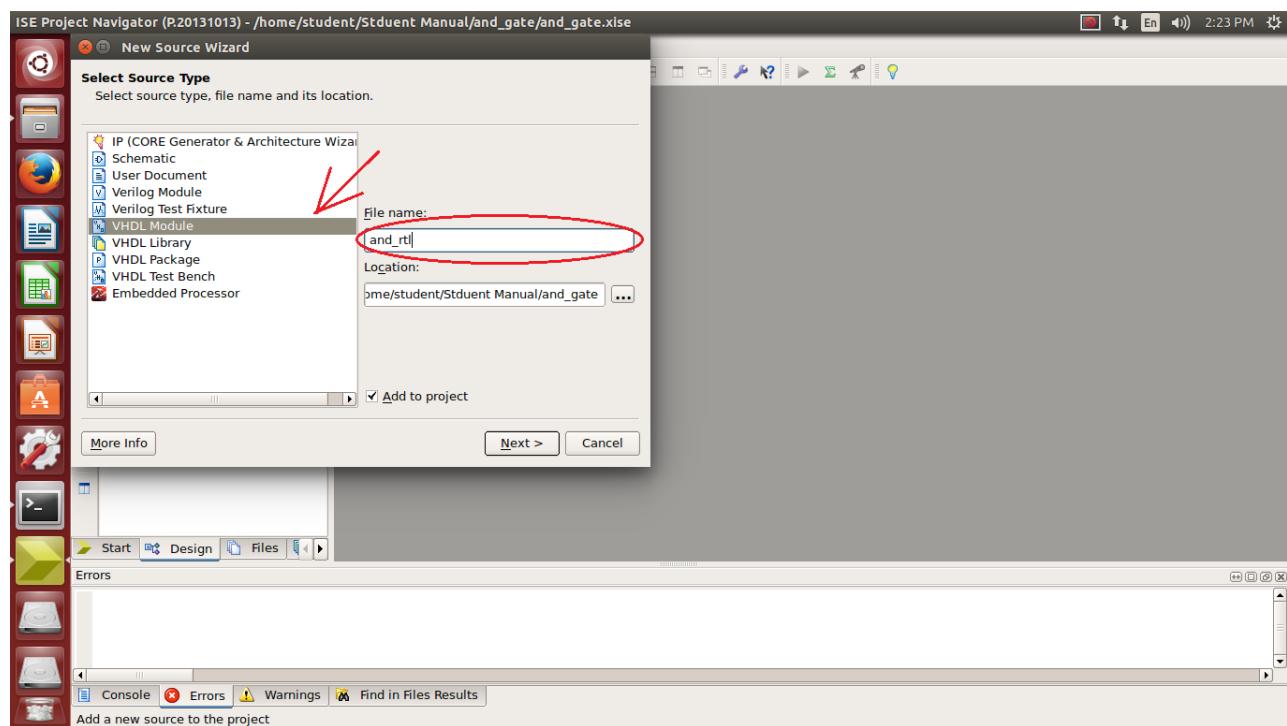
**Step 7:** The Project Summary window is shown. Click on **Finish**.



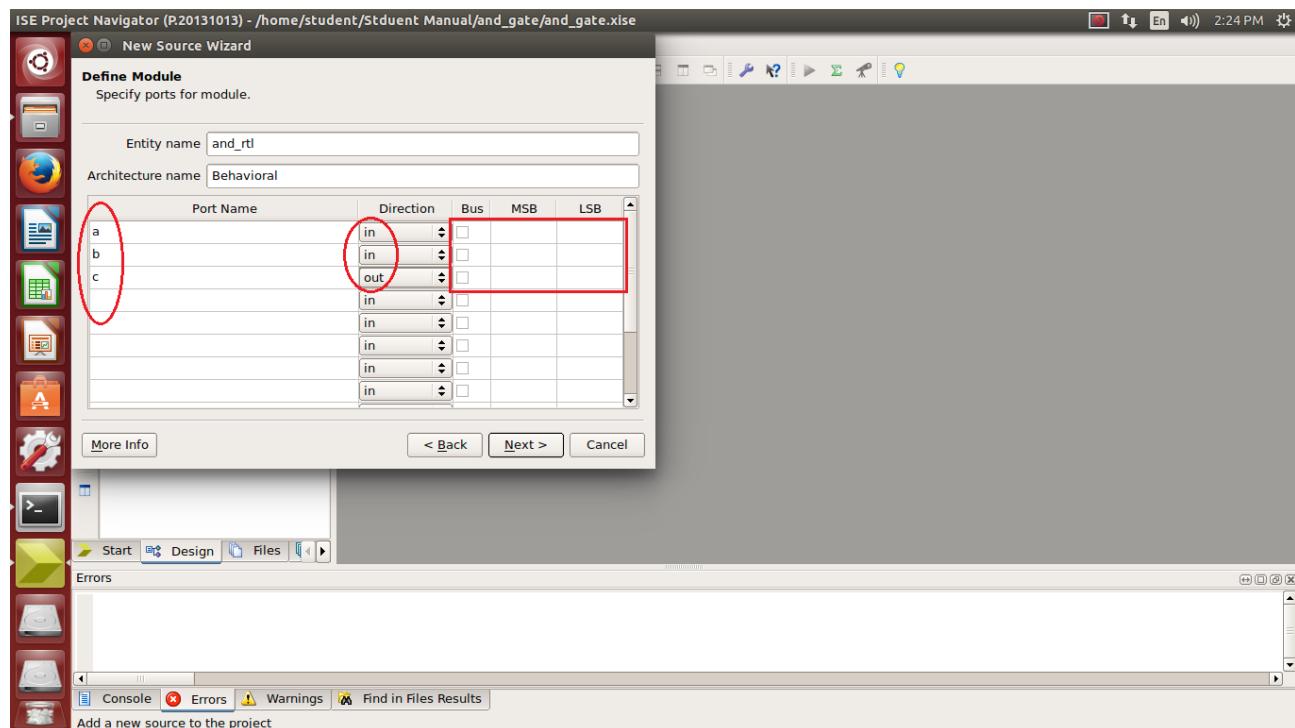
**Step 8:** A new project is created. It is seen that a **Design** panel is placed at the left-most side of the screen. Right click on the project name written at the top of the **Design** panel and select **New Source**.



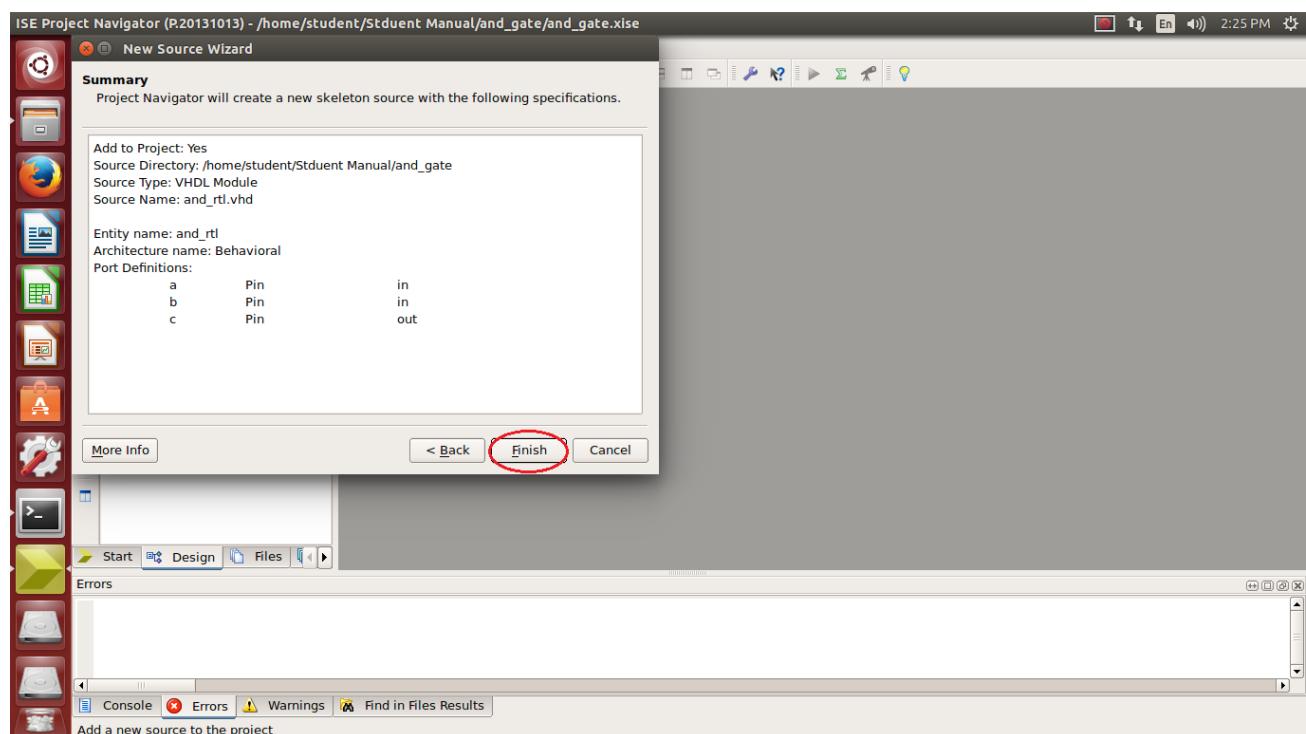
**Step 9:** The **New Source Wizard** opens. Click on **VHDL Module** and enter the **File name**. After this is done, click on **Next**.



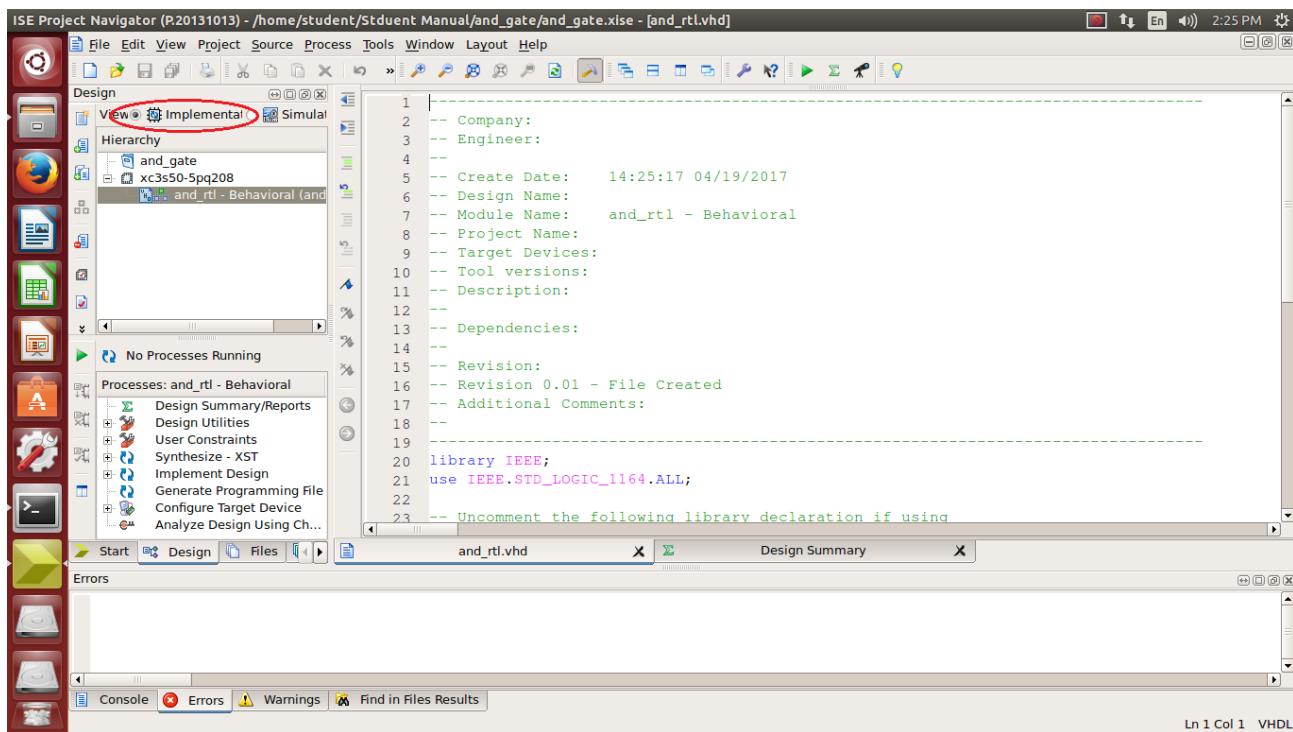
**Step 10:** The **Define Module** window opens to enable us to define the required *ports* and their *characteristics*. Enter the names of the port in the **Port Name** column, select their corresponding *input/output characteristic* in **Direction** column and *bus length* (first select the check box in the **Bus** column if you want it to be a bus then specify the **MSB** and **LSB**) where ever required.



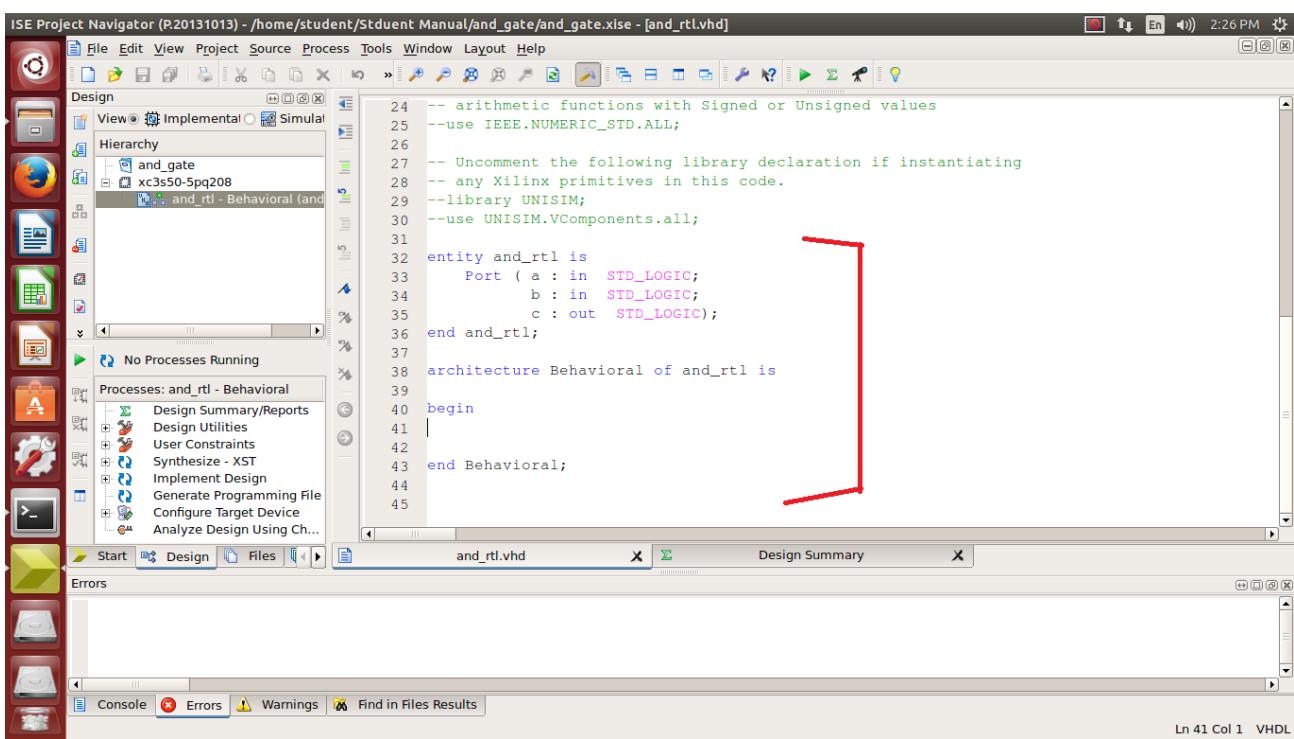
**Step 11:** The **Summary** window appears which displays the details of the defined module. Click on **Finish**.



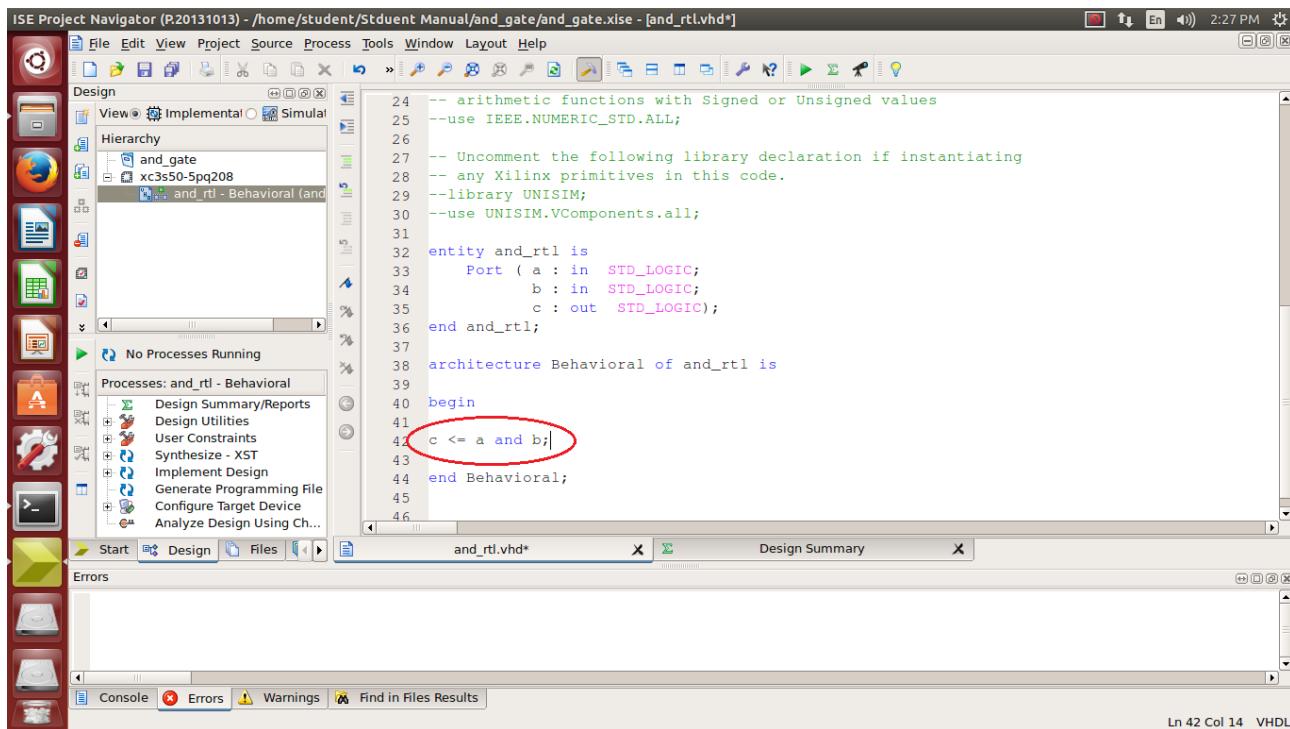
**Step 12:** The **RTL Behavioral** opens. By default, the **Implementation** viewing option is selected in the **Design** panel.



**Step 13:** Scroll down to the part where the **entity** block is defined.



**Step 14:** Write down the code after the entity block. In the given example, the required code has to be written in the **Behavioral** block. If any additional variables are needed then they are declared *before* the beginning of the **Behavioral** block.

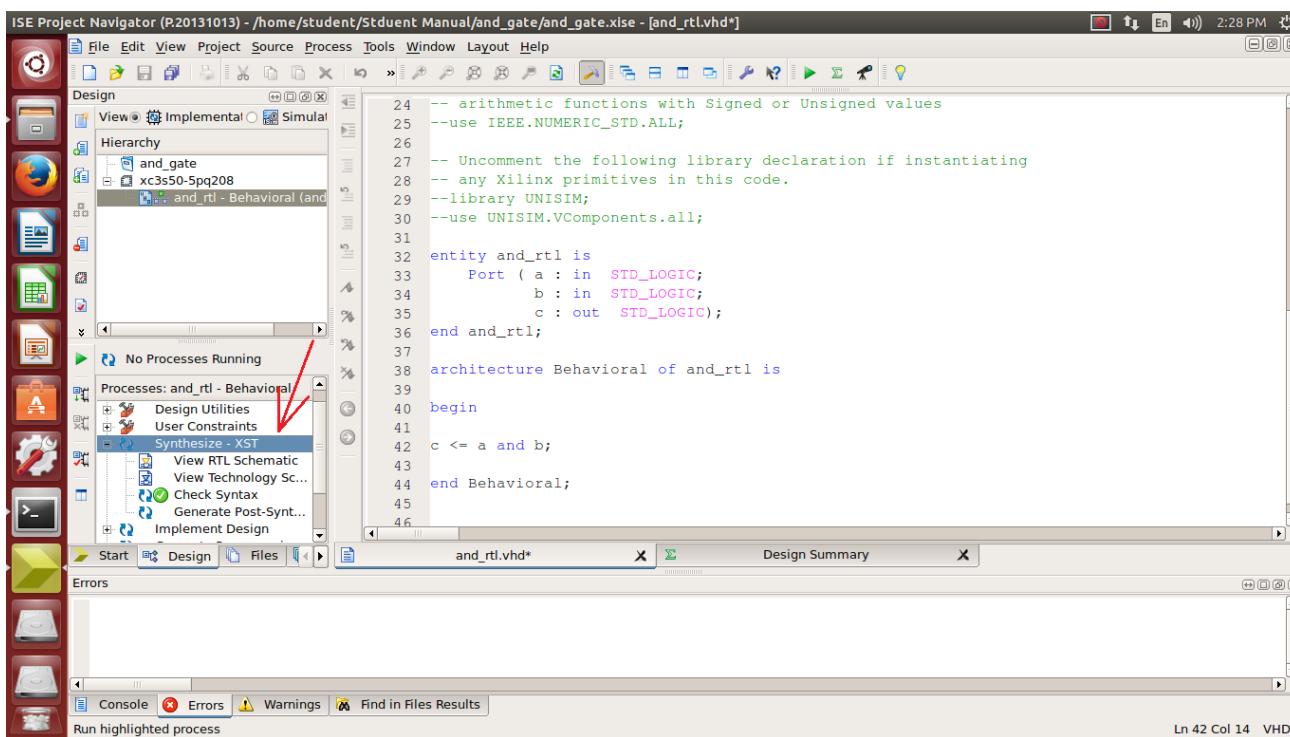


```

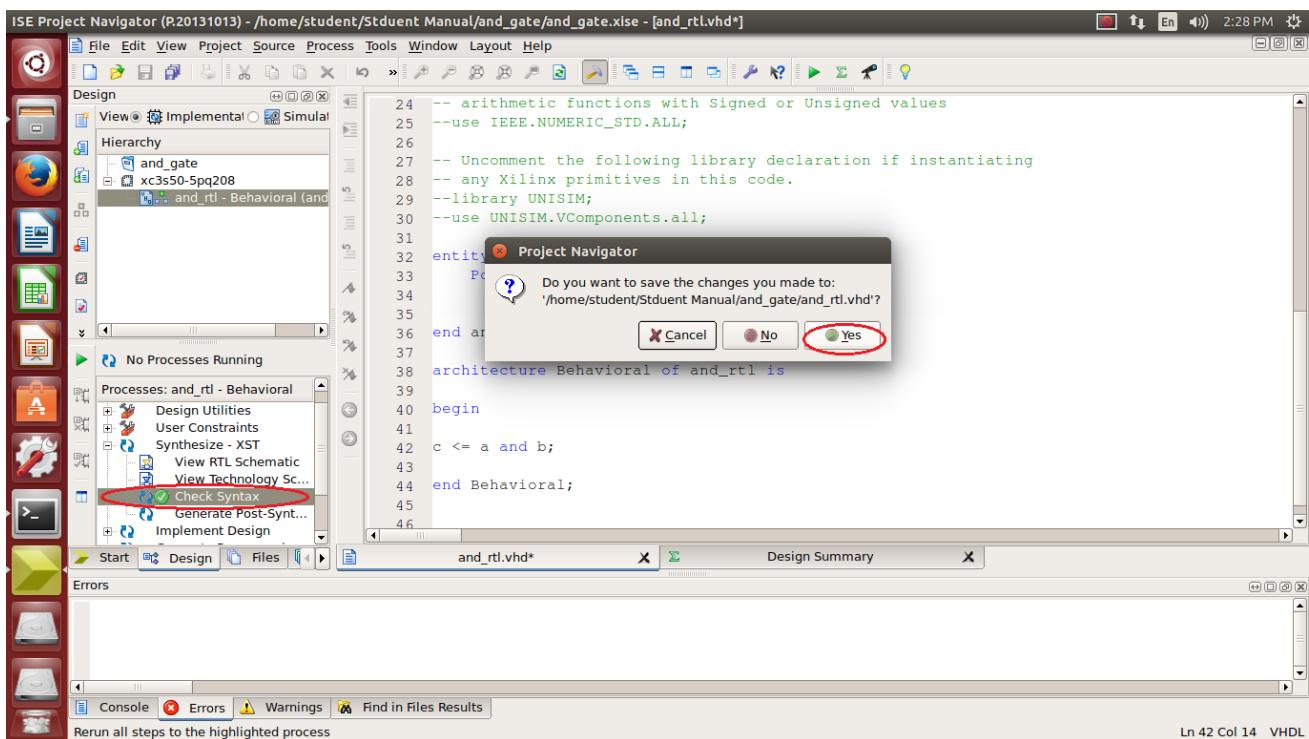
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
28 --library UNISIM;
29 --use UNISIM.VComponents.all;
30
31 entity and_rtl is
32     Port ( a : in STD_LOGIC;
33            b : in STD_LOGIC;
34            c : out STD_LOGIC);
35 end and_rtl;
36
37 architecture Behavioral of and_rtl is
38 begin
39
40 begin
41     c <= a and b;
42
43 end Behavioral;
44
45
46

```

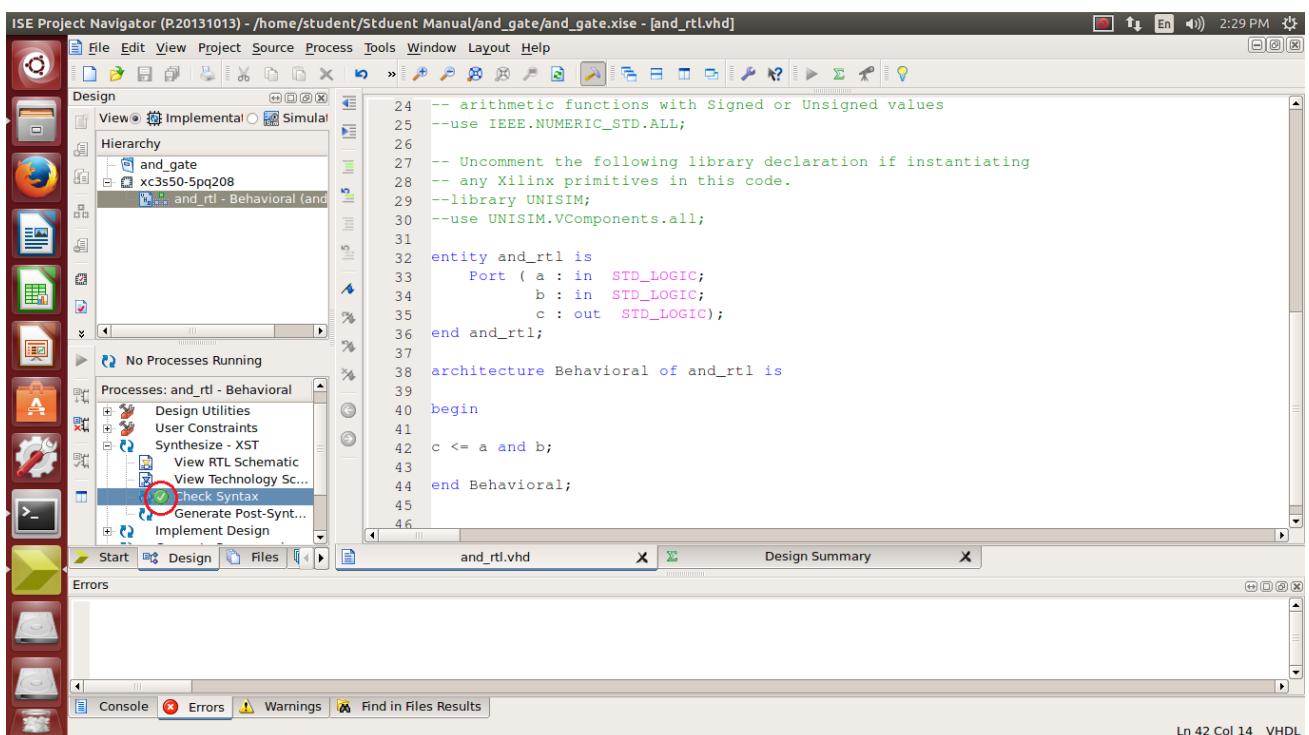
**Step 15:** After writing the code click and expand the **Synthesize-XST** option present in the Processes panel in the *left side* of the screen.



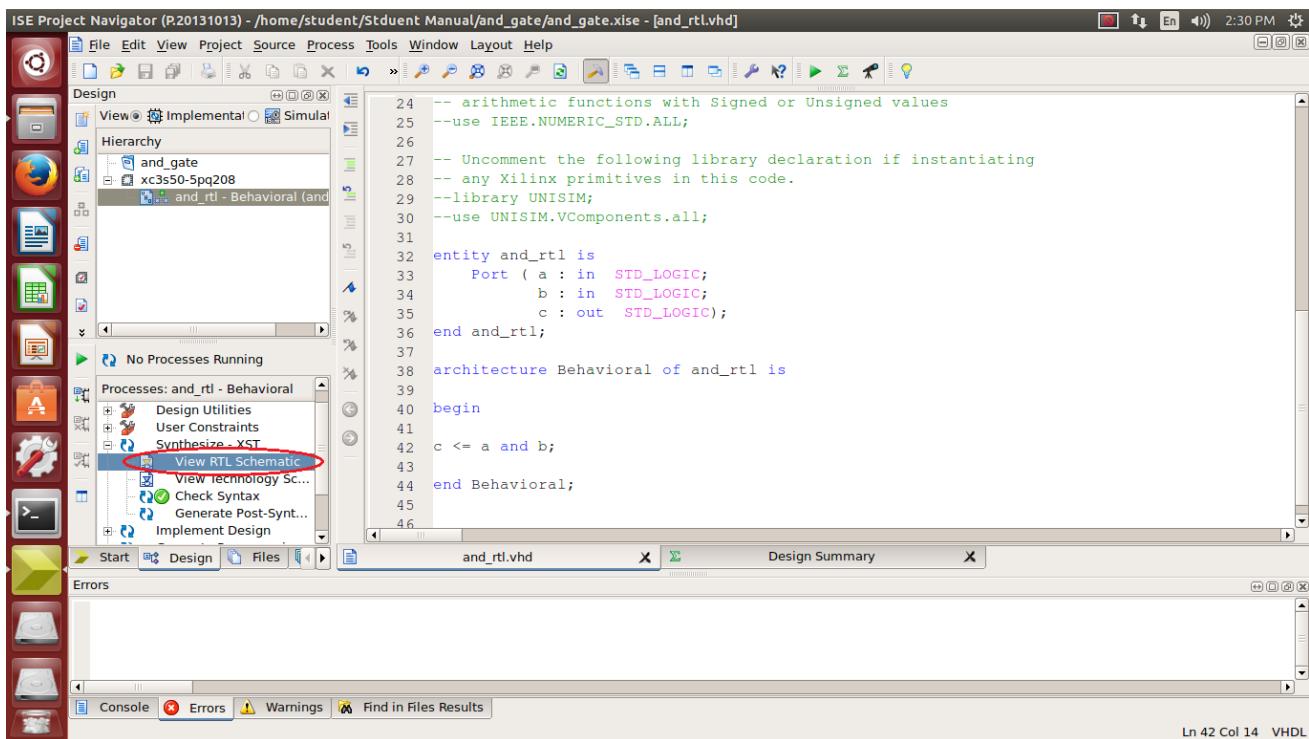
**Step 16:** Click on the **Check Syntax** option which appears under **Synthesize-XST**. On doing that a dialogue box appears asking whether you want to save the changes that have been made. Click **Yes**.



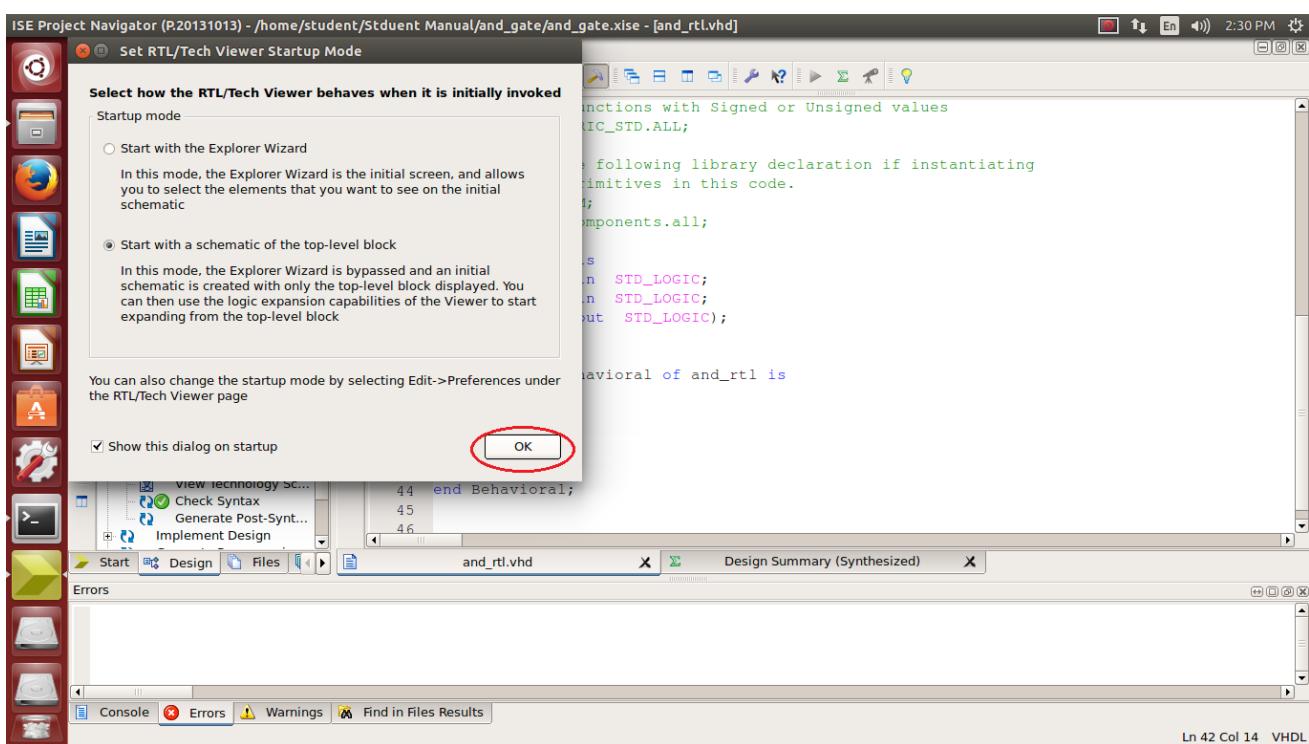
**Step 17:** The **Check Syntax** checks for *syntactical errors* in the code. If there is *no error* present then a **green tick** appears beside the **Check Syntax** option, or else a **red cross** appears indicating that the code has some *error*. If there are any errors, then make the necessary changes and right click **Check Syntax** and select **Rerun**.



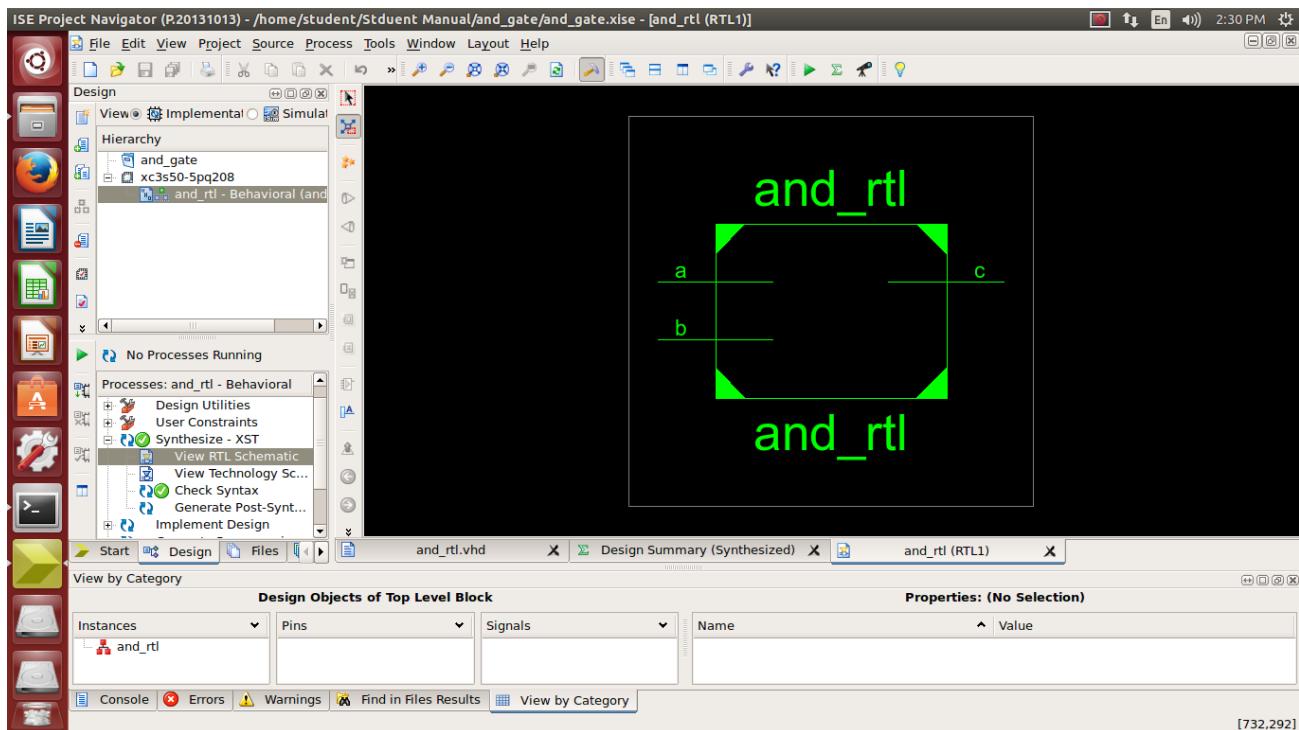
**Step 18:** Once the green tick appears click on **View RTL Schematic**.



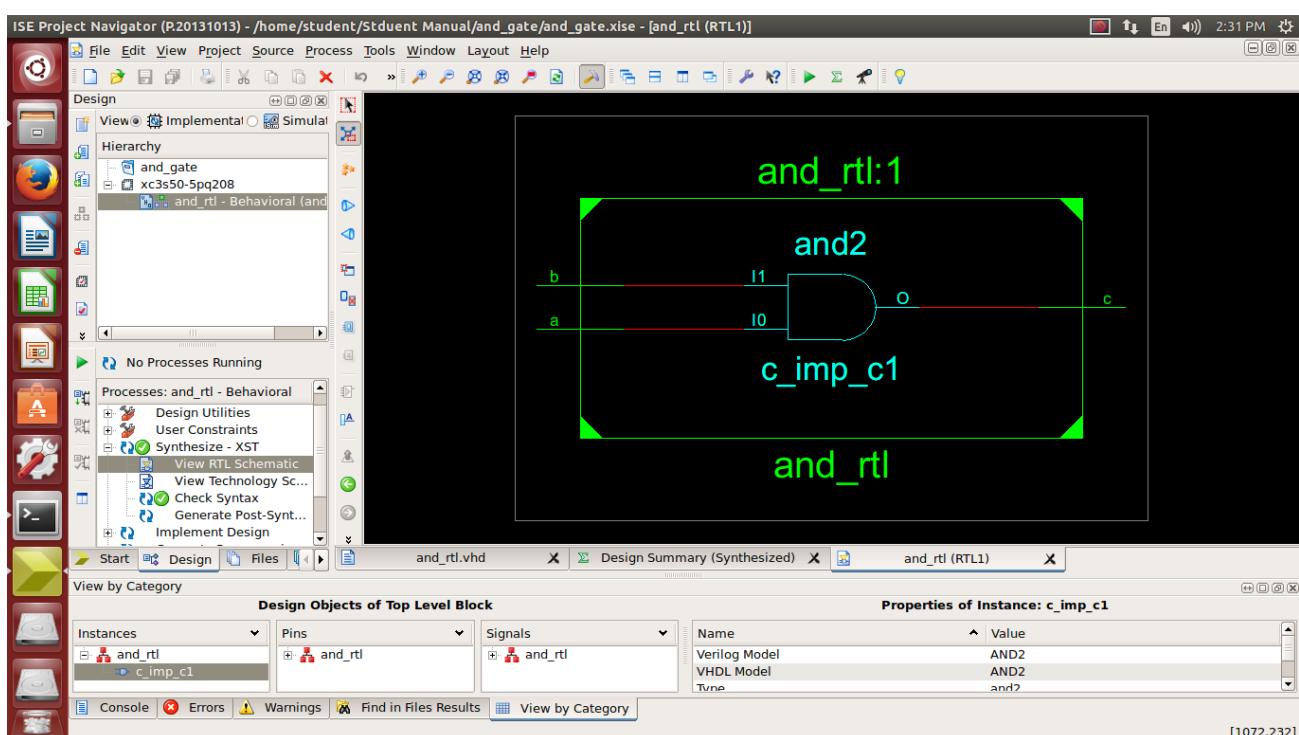
**Step 19:** Select the necessary viewing arrangement in the Startup Mode window and click on **OK** to proceed.



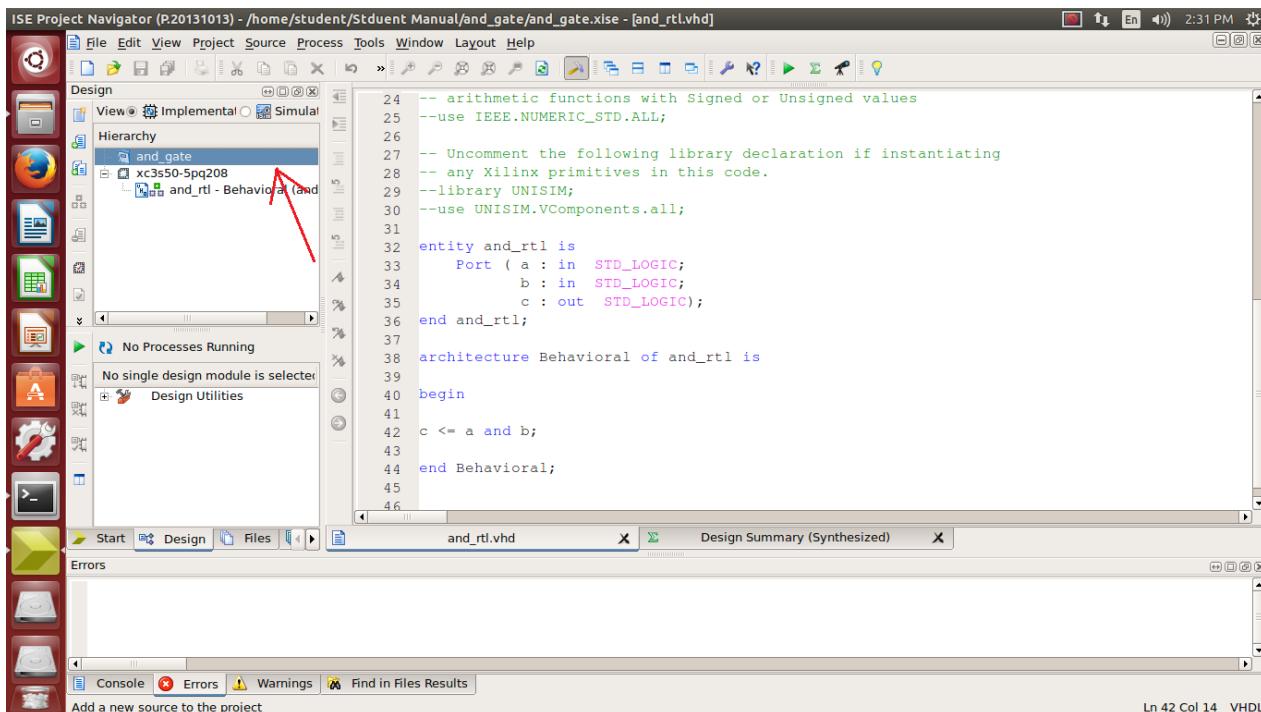
**Step 20:** The written code is compiled and the **RTL Schematic** is viewed. The window shows a *schematic diagram* of the written code along with the given input and output ports.



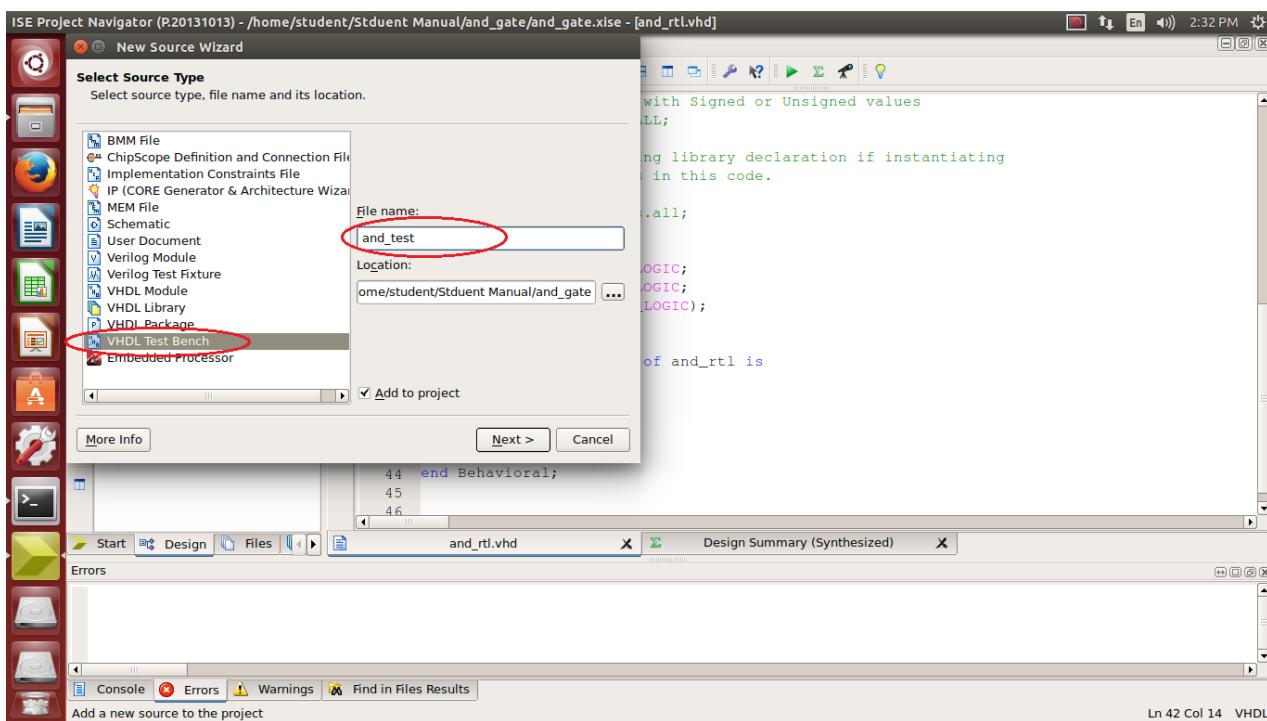
Double click on the schematic for a detailed view.



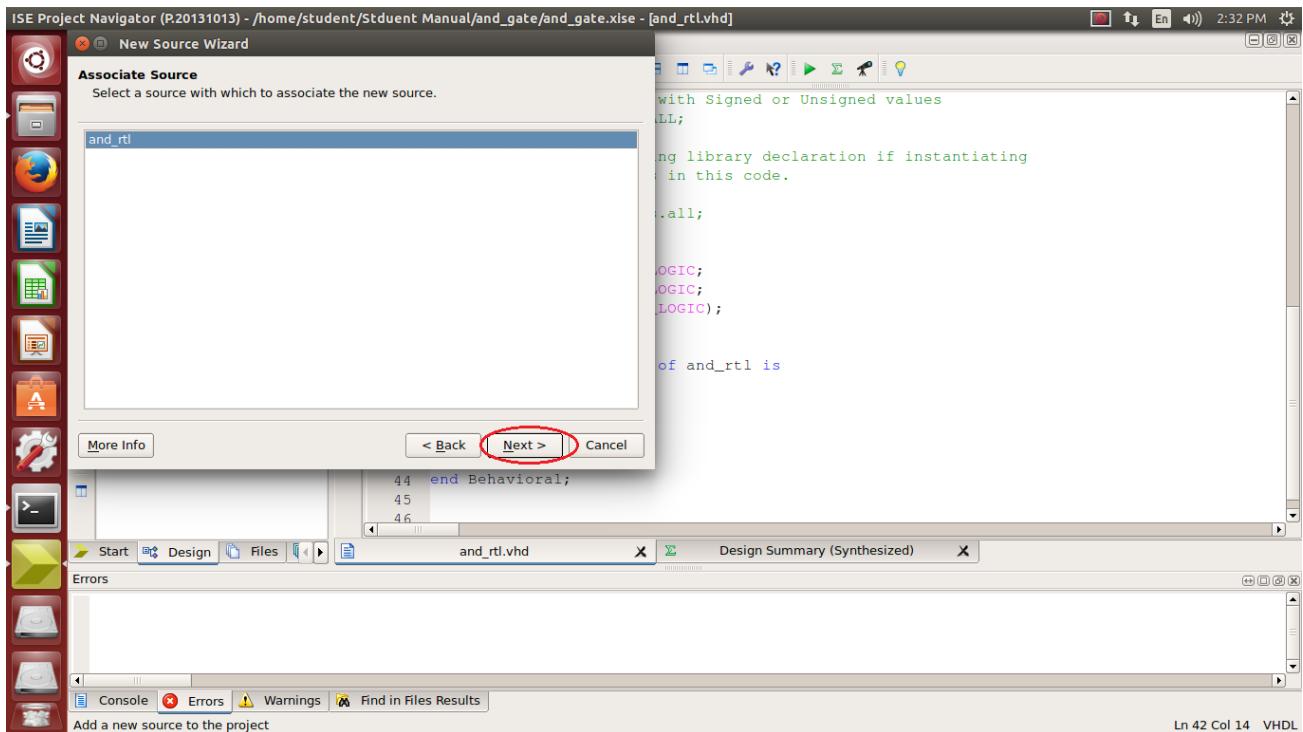
**Step 21:** After the RTL Schematic has been created, test cases have to be provided to the input ports to get the output waveform and check the whether the output is as desired. For this we need to create a **Test Bench**. Right click on the project name written at the top of the design panel and select **New Source**.



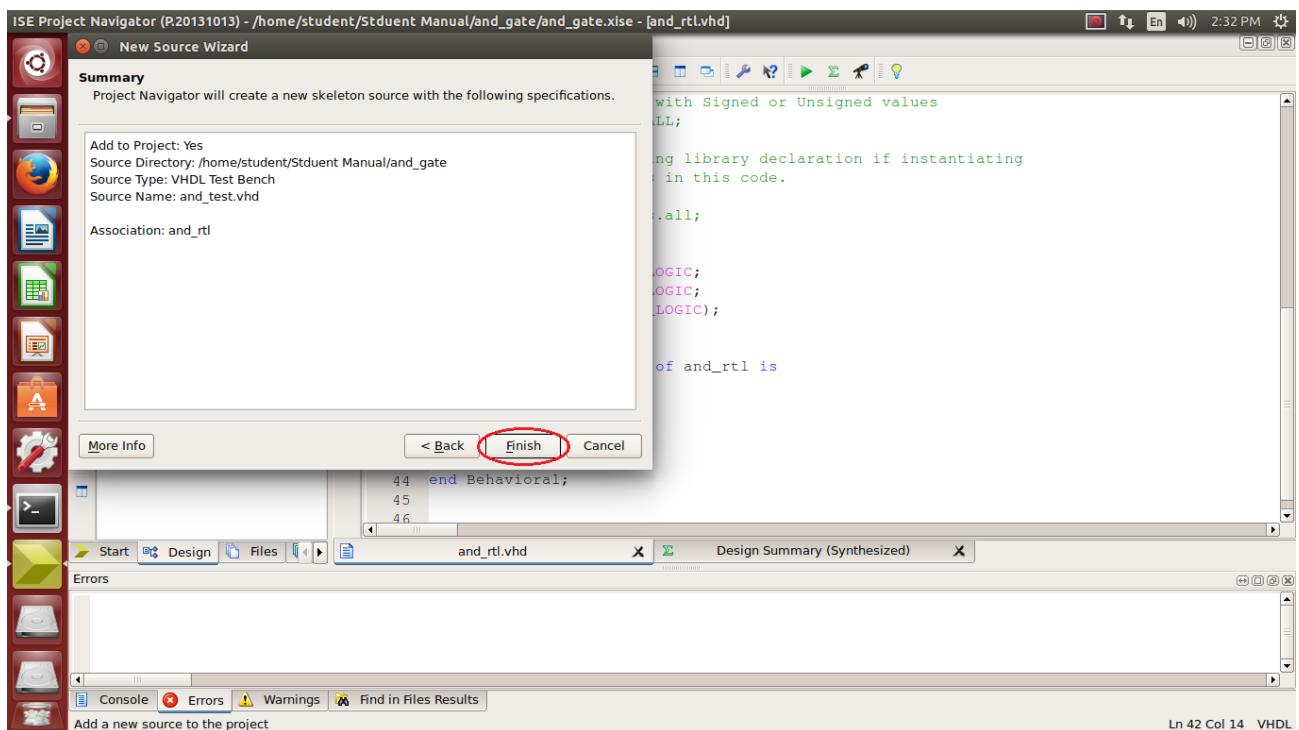
**Step 22:** The New Source Wizard opens. Select the **VHDL Test Bench** option and name the test bench module in the **File Name** box (*name should be different than the VHDL Module name created in step 9*). After completing this task click **Next** to proceed.



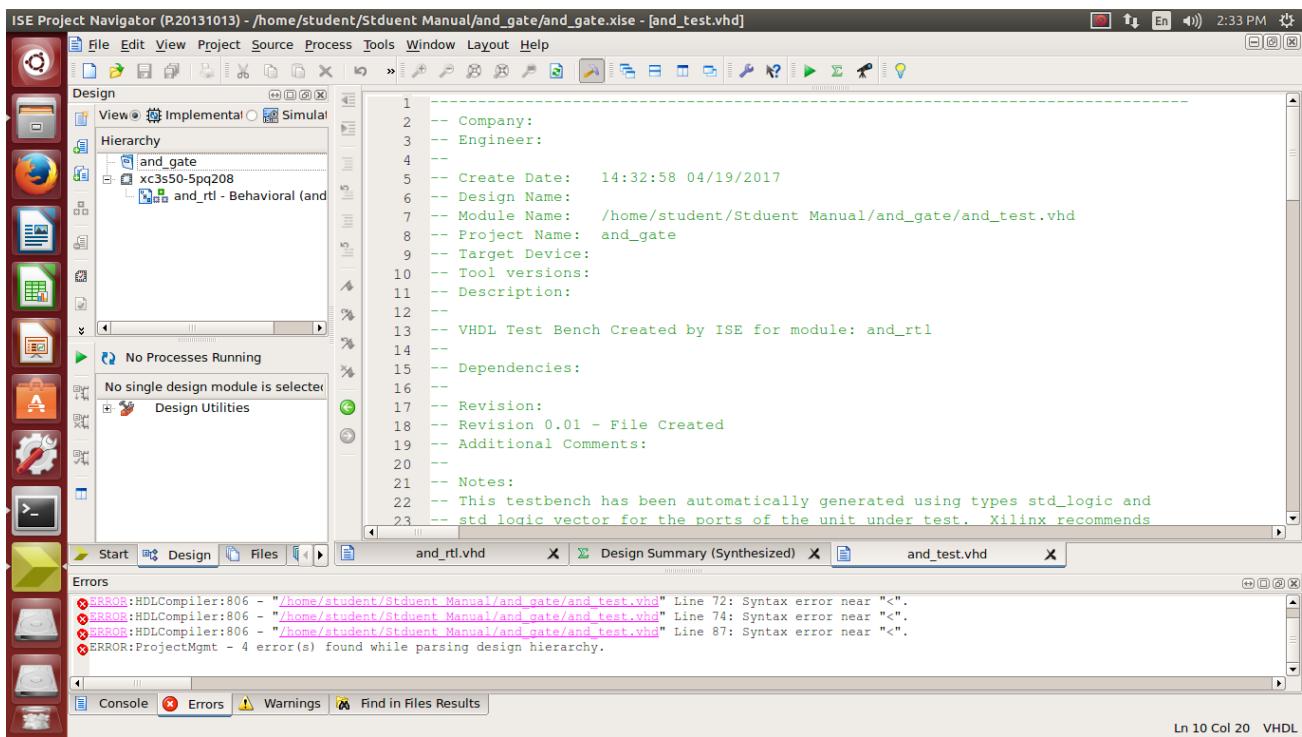
**Step 23:** Xilinx automatically associates this test bench file with the RTL file that had been created earlier. Click **Next** to proceed.



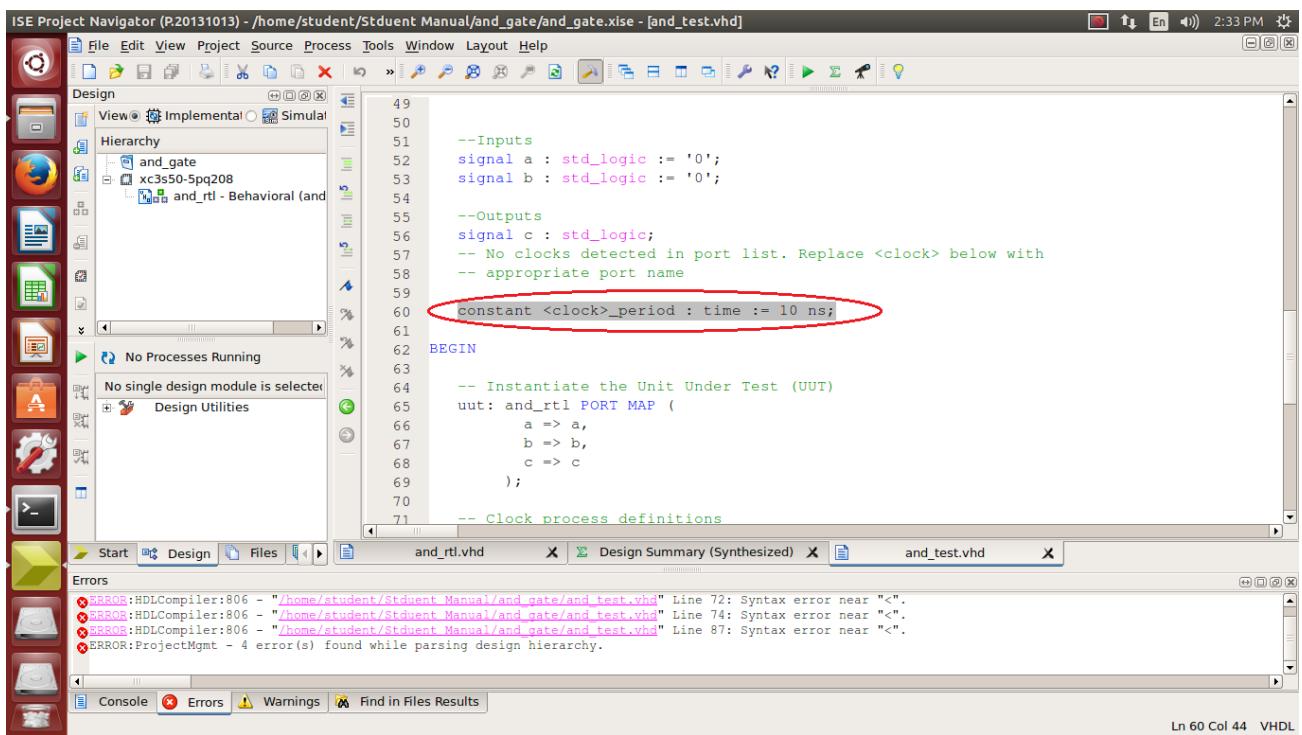
**Step 24:** Check the details of the test bench being created in the **Summary** window and click **Finish** to create the test bench.



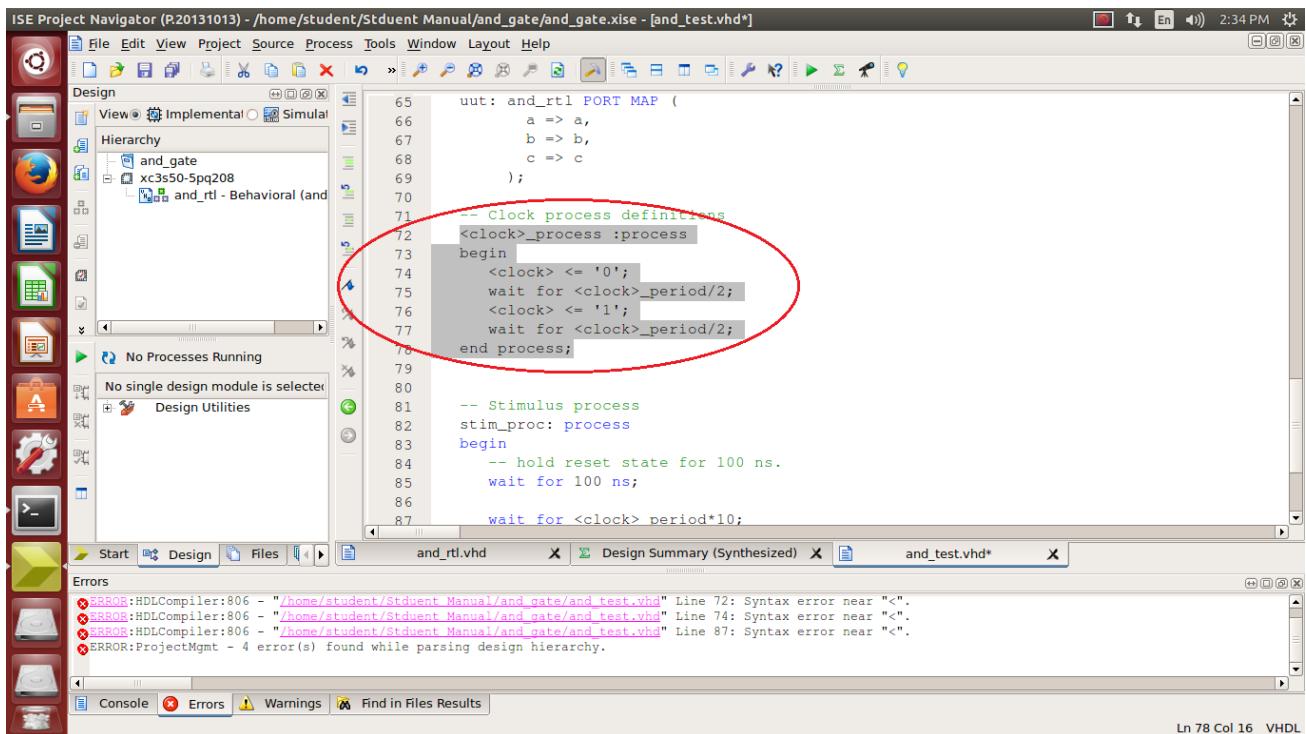
**Step 25:** The window for the test bench code opens.



**Step 26a:** Go through the already written part of the code (default blocks of code) and **remove the unwanted clock commands** (*for the assignments of sequential circuits which requires clock pulse, change it to 2ps.*)



**Step 26b:** Remove the circled part of the already written code for assignments of combinational circuits. For assignments of sequential circuits that require clock pulse, keep it intact.



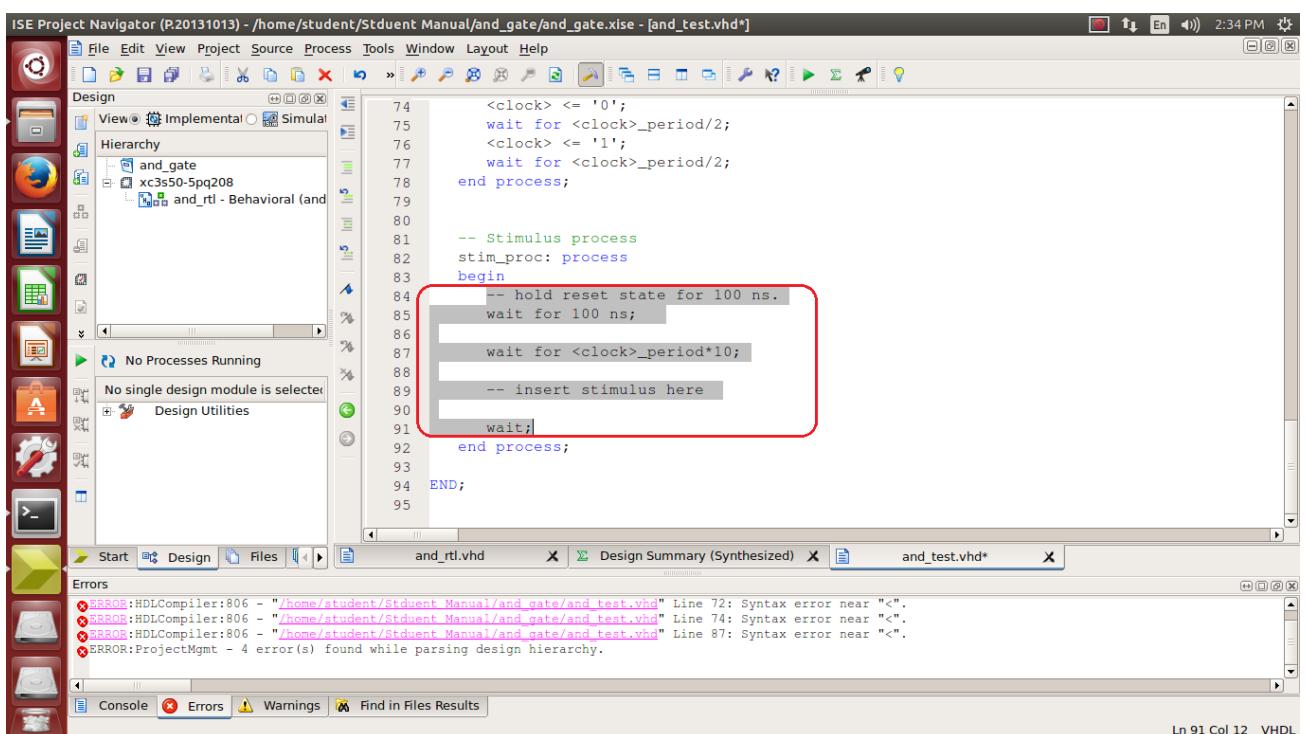
```

65      uut: and_rtl PORT MAP (
66        a => a,
67        b => b,
68        c => c
69      );
70
71      -- Clock process definitions
72      <clock>_process :process
73      begin
74        <clock> <= '0';
75        wait for <clock>_period/2;
76        <clock> <= '1';
77        wait for <clock>_period/2;
78      end process;
79
80
81      -- Stimulus process
82      stim_proc: process
83      begin
84        -- hold reset state for 100 ns.
85        wait for 100 ns;
86
87        wait for <clock> period*10;
88
89        -- insert stimulus here
90
91        wait;
92      end process;
93
94  END;
95

```

Ln 78 Col 16 VHDL

**Step 26c:** Remove the marked part of the code.



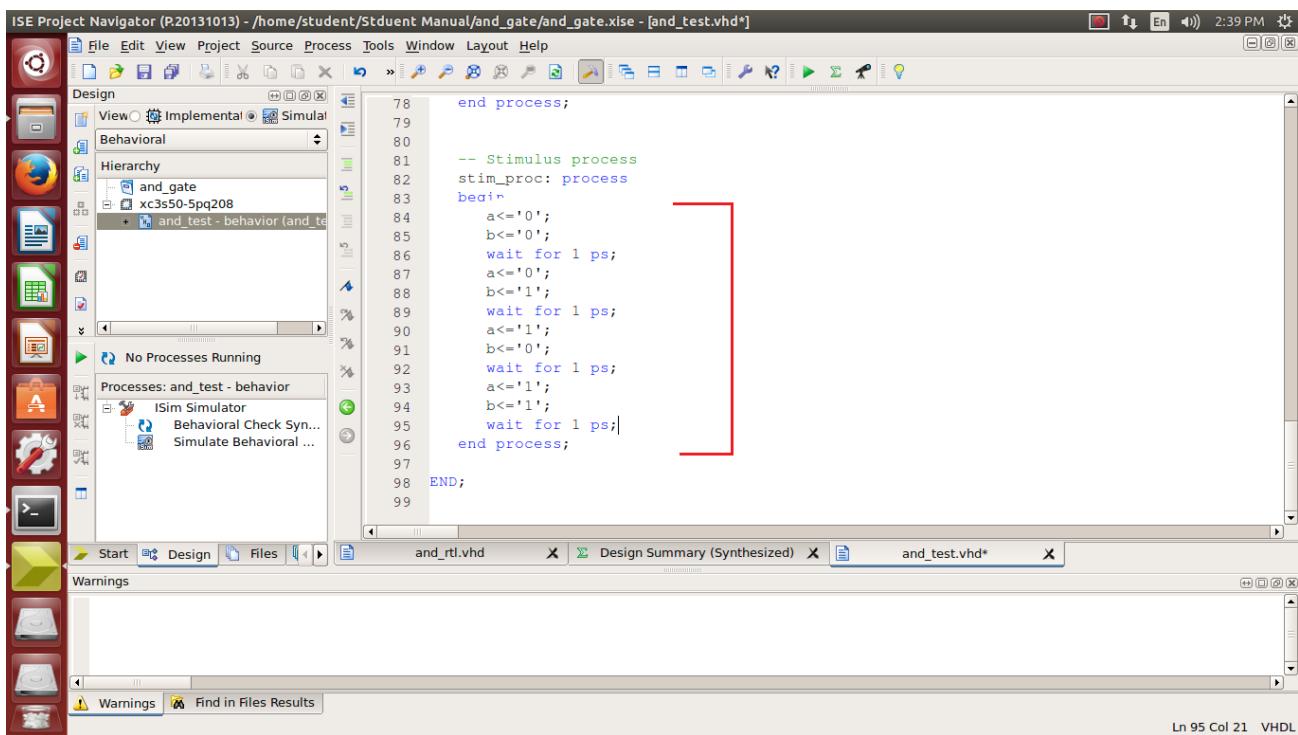
```

74      <clock> <= '0';
75      wait for <clock>_period/2;
76      <clock> <= '1';
77      wait for <clock>_period/2;
78    end process;
79
80
81    -- Stimulus process
82    stim_proc: process
83    begin
84      -- hold reset state for 100 ns.
85      wait for 100 ns;
86
87      wait for <clock>_period*10;
88
89      -- insert stimulus here
90
91      wait;
92    end process;
93
94  END;
95

```

Ln 91 Col 12 VHDL

**Step 27:** Write down the *test cases* for the input portals in in the **stim\_proc** block.



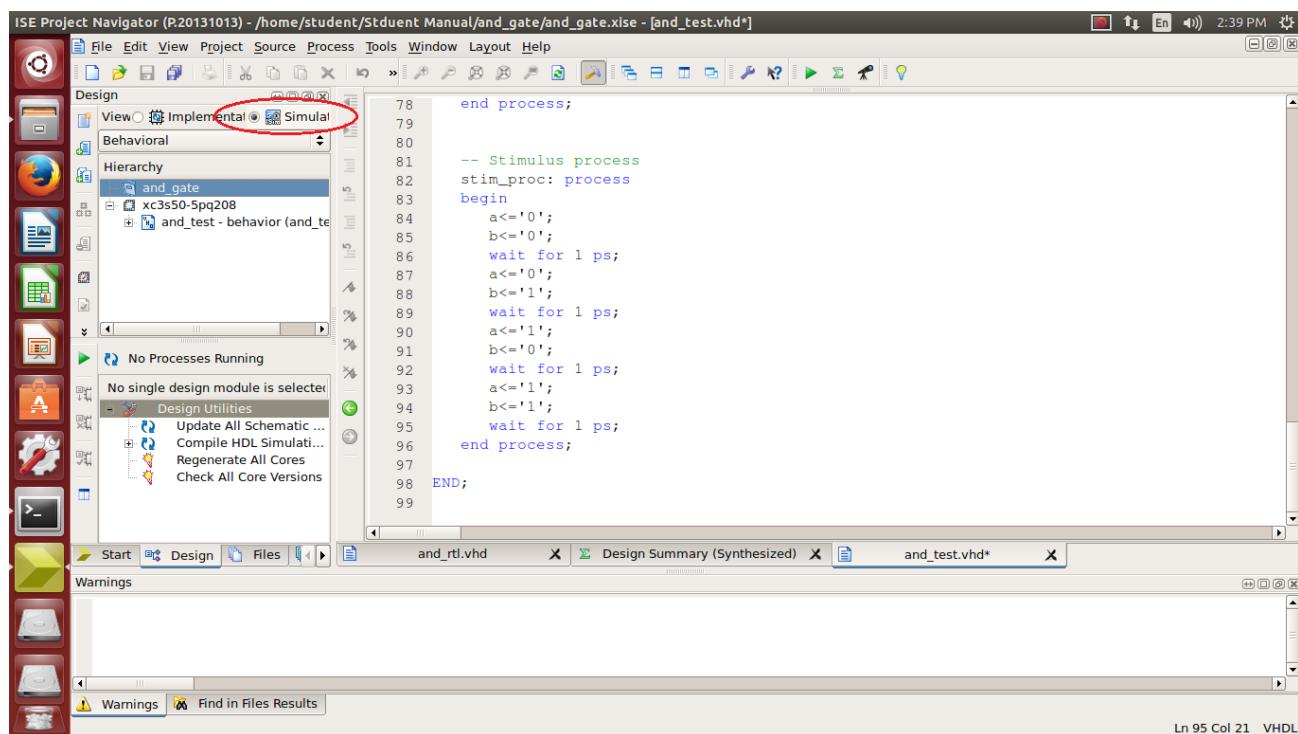
```

ISE Project Navigator (P.20131013) - /home/student/Stduent Manual/and_gate/and_gate.xise - [and_test.vhd*]
File Edit View Project Source Process Tools Window Layout Help
Design View Implemental Behavioral
Hierarchy and_gate xc3s50-5pq208 and_test - behavior (and_te...
No Processes Running
Processes: and_test - behavior
  iSim Simulator Behavioral Check Syn...
  Simulate Behavioral ...
Start Design Files and_rtl.vhd Design Summary (Synthesized) and_test.vhd*
Ln 95 Col 21 VHDL
    end process;

    -- Stimulus process
    stim_proc: process
    begin
        a<='0';
        b<='0';
        wait for 1 ps;
        a<='0';
        b<='1';
        wait for 1 ps;
        a<='1';
        b<='0';
        wait for 1 ps;
        a<='1';
        b<='1';
        wait for 1 ps;
    end process;
END;

```

**Step 28:** After writing the code for test bench, select the **Simulation** viewing option in the **Design** panel, which is on the left side of the screen.



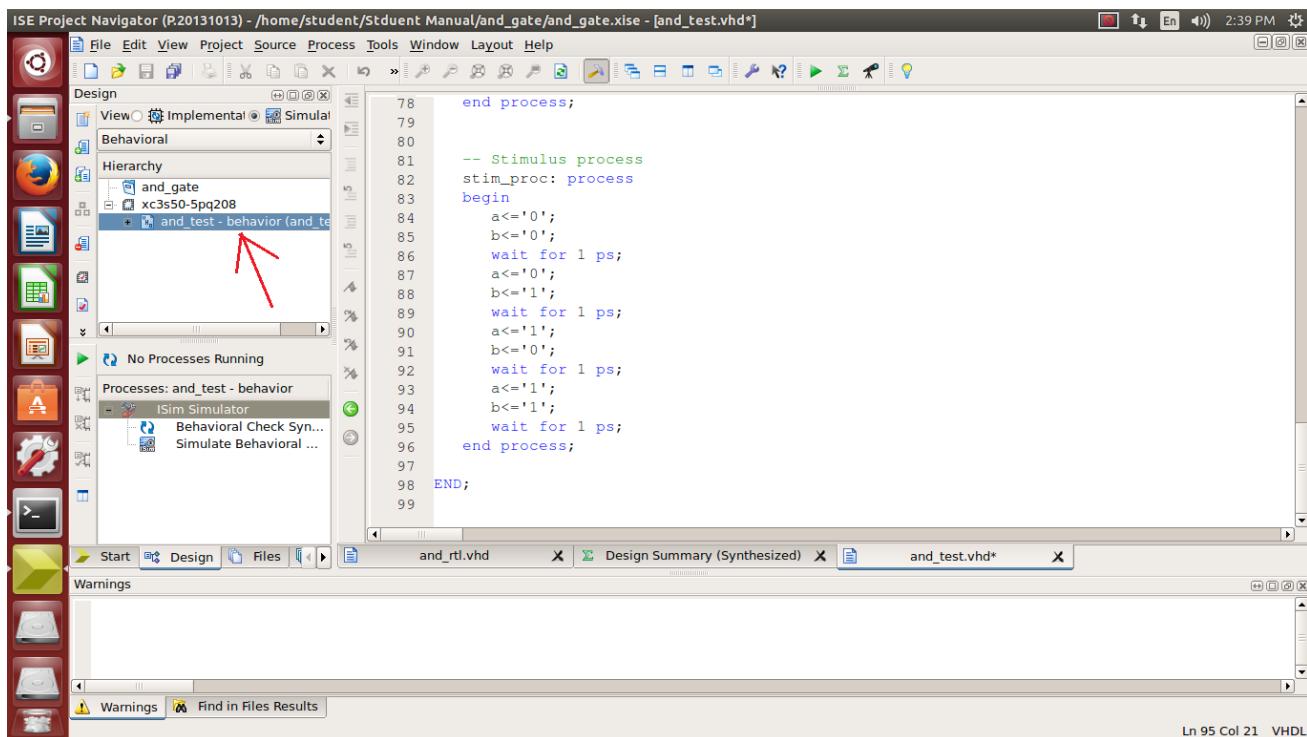
```

ISE Project Navigator (P.20131013) - /home/student/Stduent Manual/and_gate/and_gate.xise - [and_test.vhd*]
File Edit View Project Source Process Tools Window Layout Help
Design View Simulator Behavioral
Hierarchy and_gate xc3s50-5pq208 and_test - behavior (and_te...
No Processes Running
No single design module is selected
  Design Utilities
    Update All Schematic ...
    Compile HDL Simulat...
    Regenerate All Cores
    Check All Core Versions
Start Design Files and_rtl.vhd Design Summary (Synthesized) and_test.vhd*
Ln 95 Col 21 VHDL
    end process;

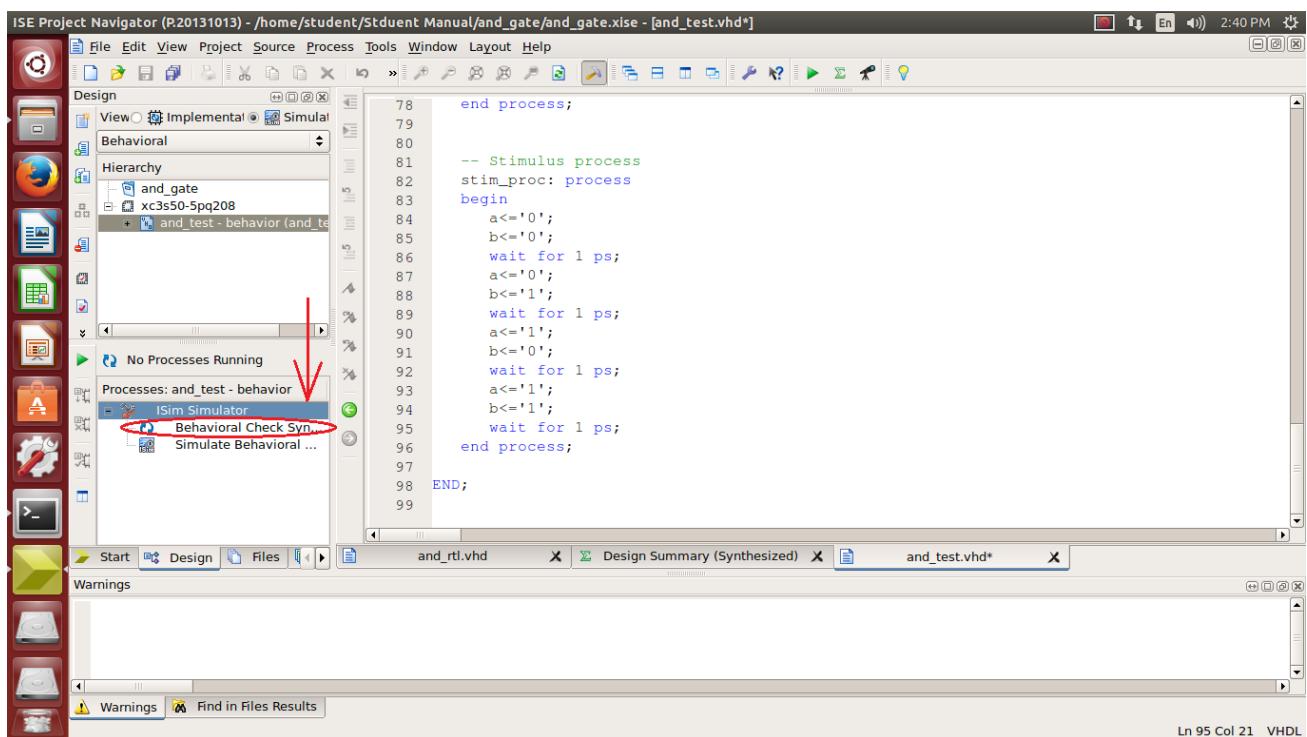
    -- Stimulus process
    stim_proc: process
    begin
        a<='0';
        b<='0';
        wait for 1 ps;
        a<='0';
        b<='1';
        wait for 1 ps;
        a<='1';
        b<='0';
        wait for 1 ps;
        a<='1';
        b<='1';
        wait for 1 ps;
    end process;
END;

```

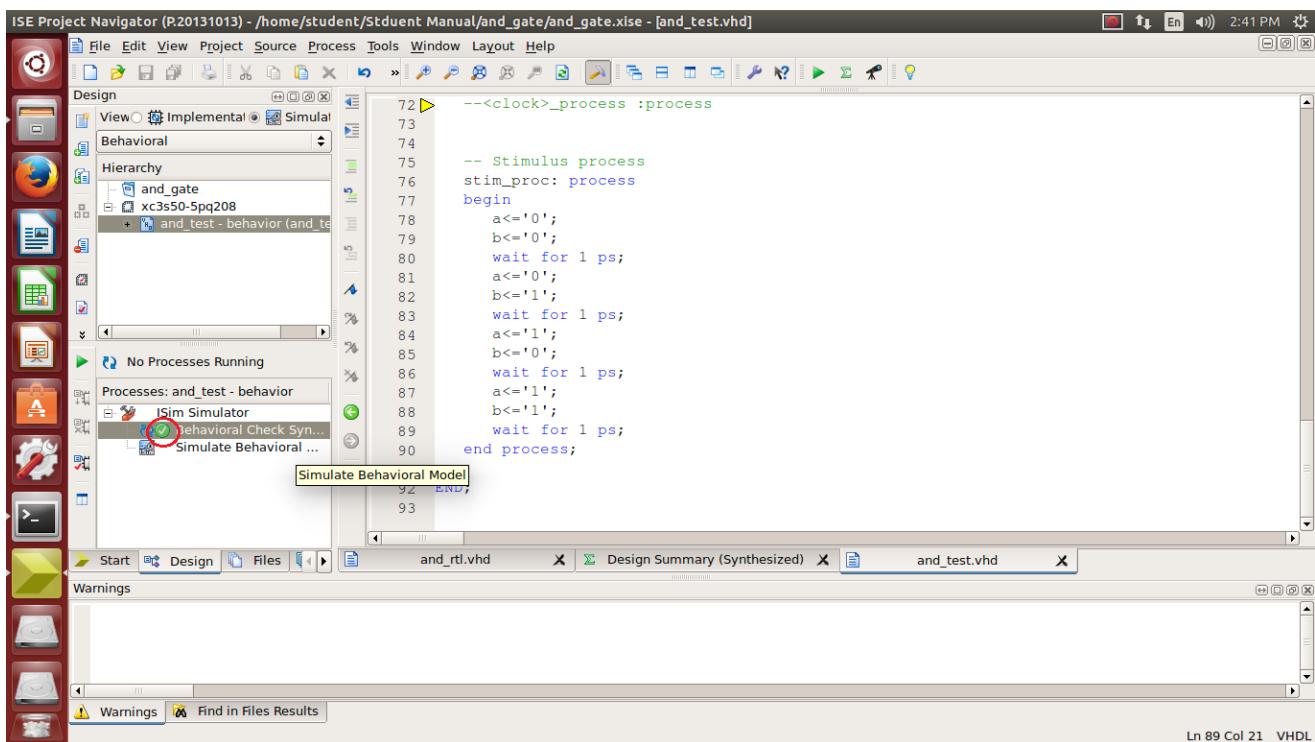
**Step 29:** Select the behavior option in the Design panel.



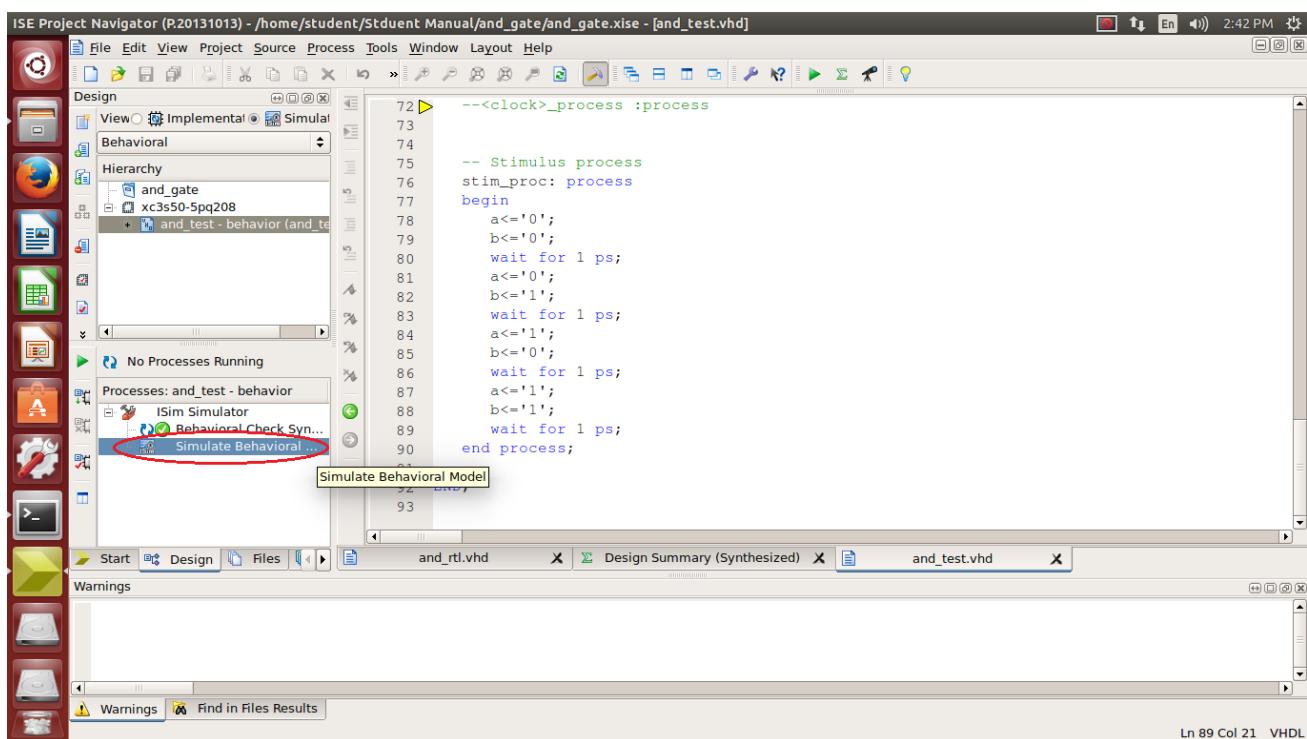
**Step 30:** Select and expand the **ISim Simulator** option in the Processes panel and click on **Behavioral Check Syntax**.



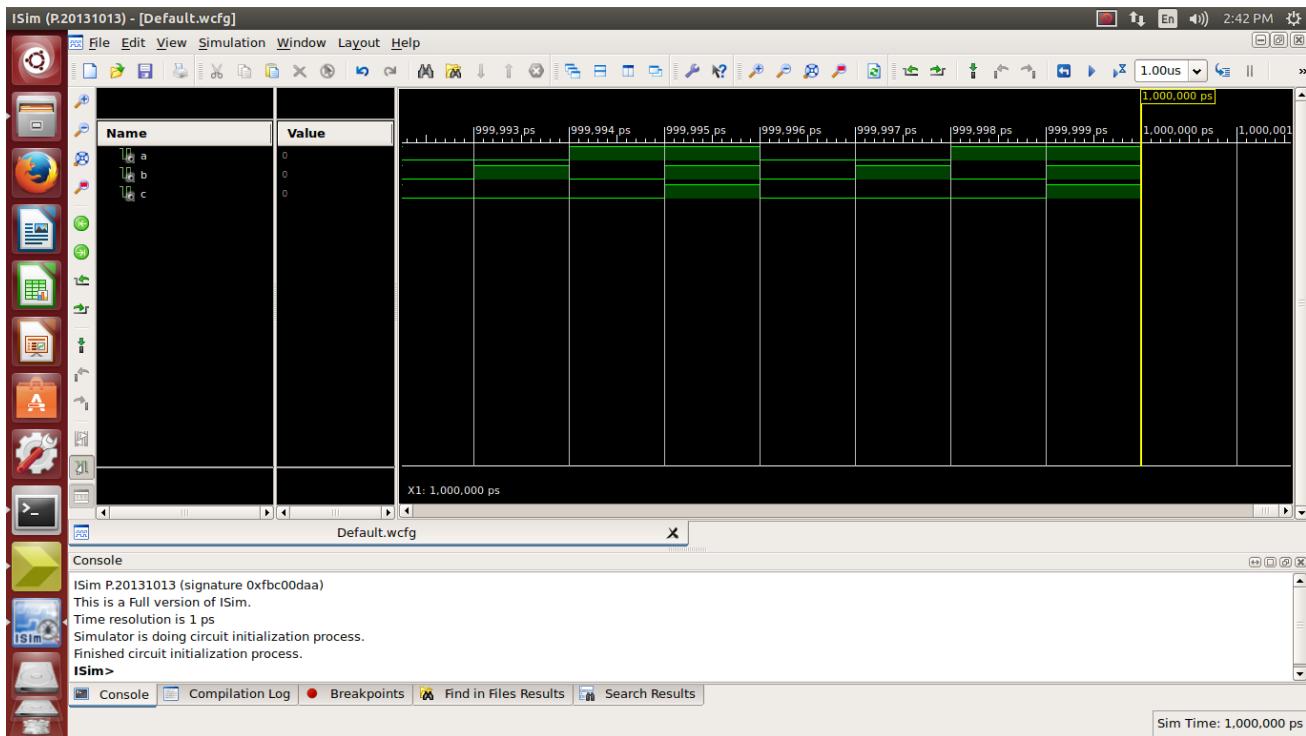
**Step 31:** If there are no *syntactical errors in the code* then a *green tick appears beside the Behavioral Check Syntax option*, or else a *red cross* appears *indicating that the program is not error-free.*



**Step 32:** Once the *green tick* apperas and it is clear that the code has no errors then click on the **Simulate Behavioral** option.



**Step 33:** The *output waveform* for the given test cases is displayed.



**Step 34:** Press **ctrl+home** to view the first input test cases and the corresponding output and then use the left and right arrow keys to navigate to the next or previous input.

