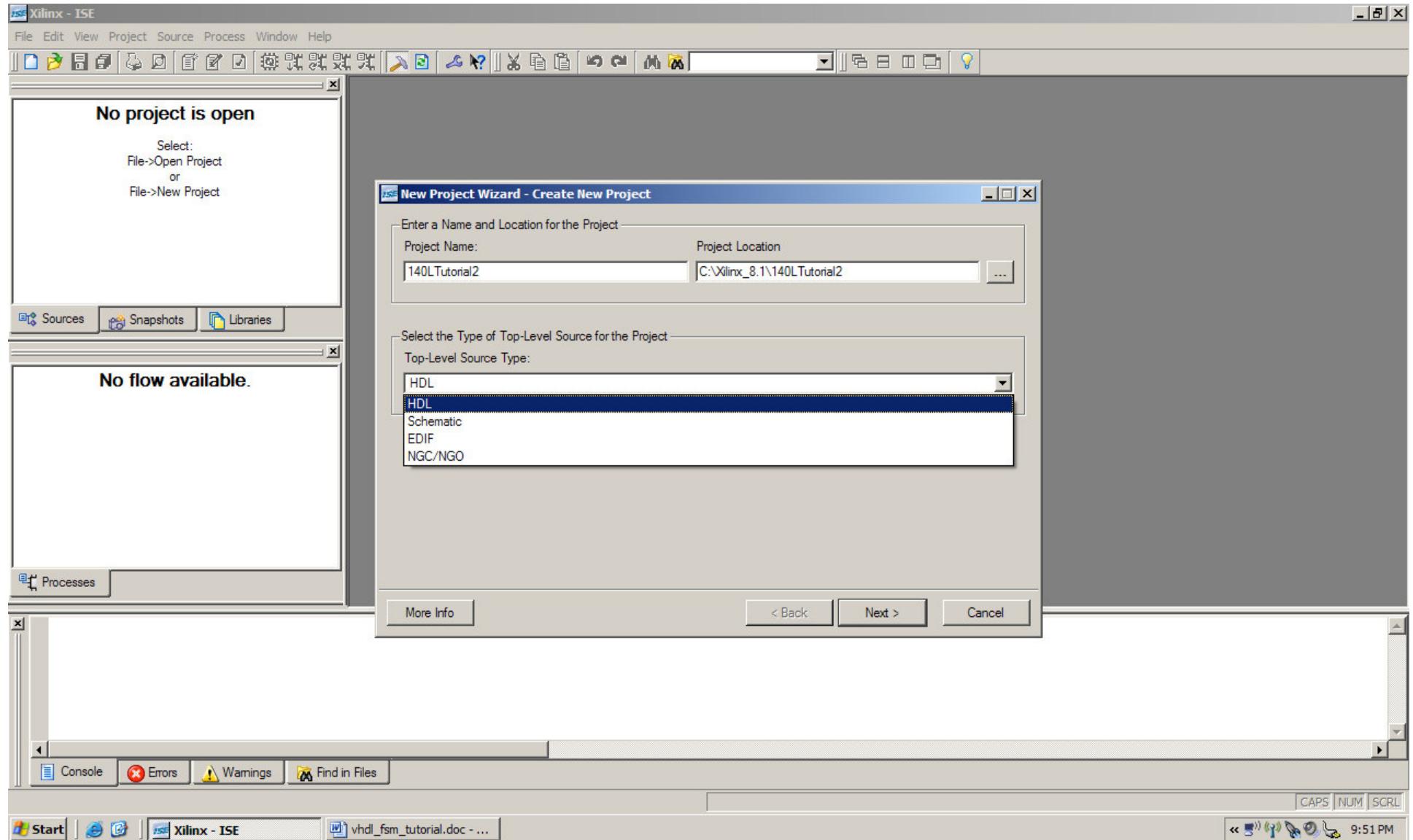
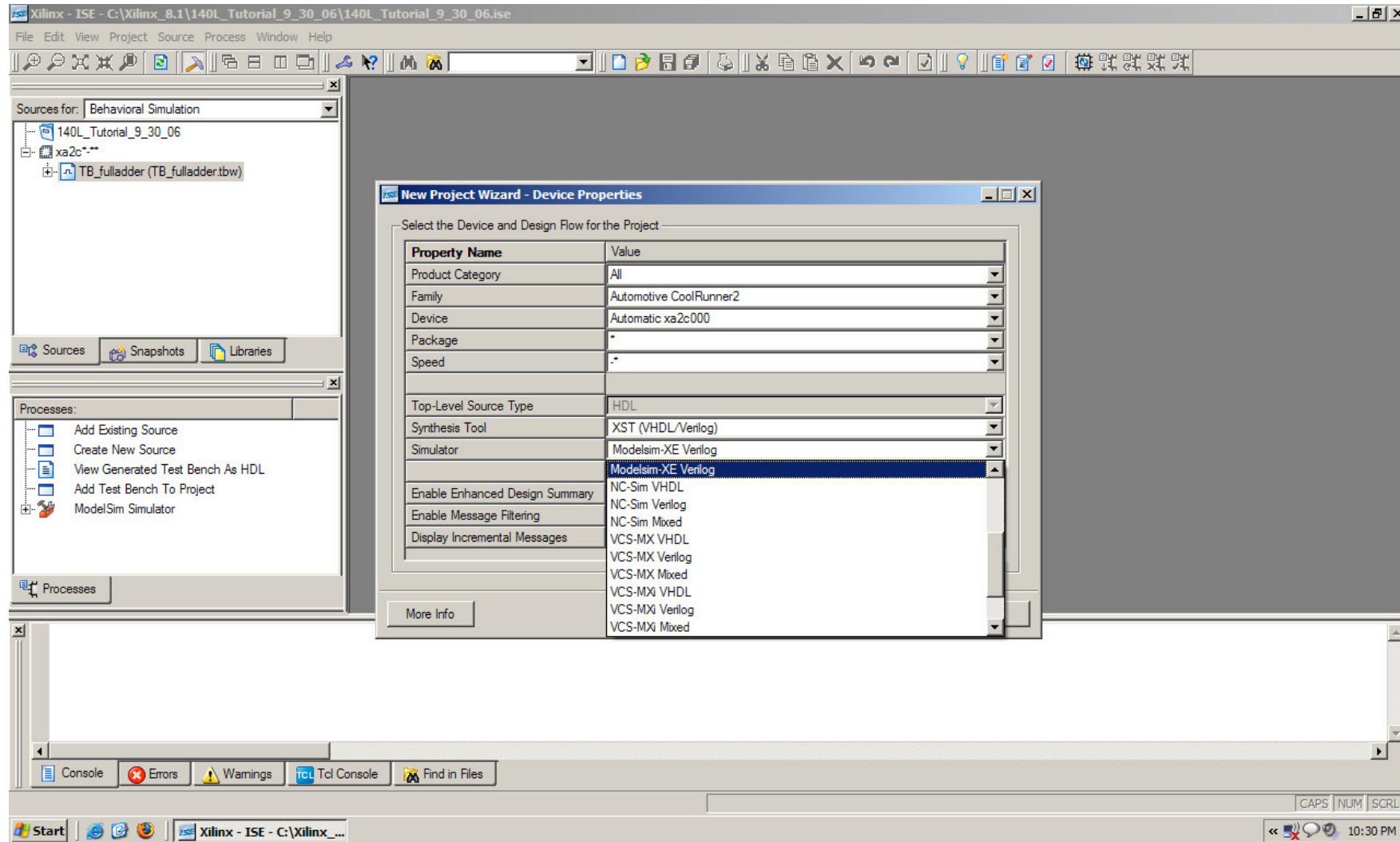


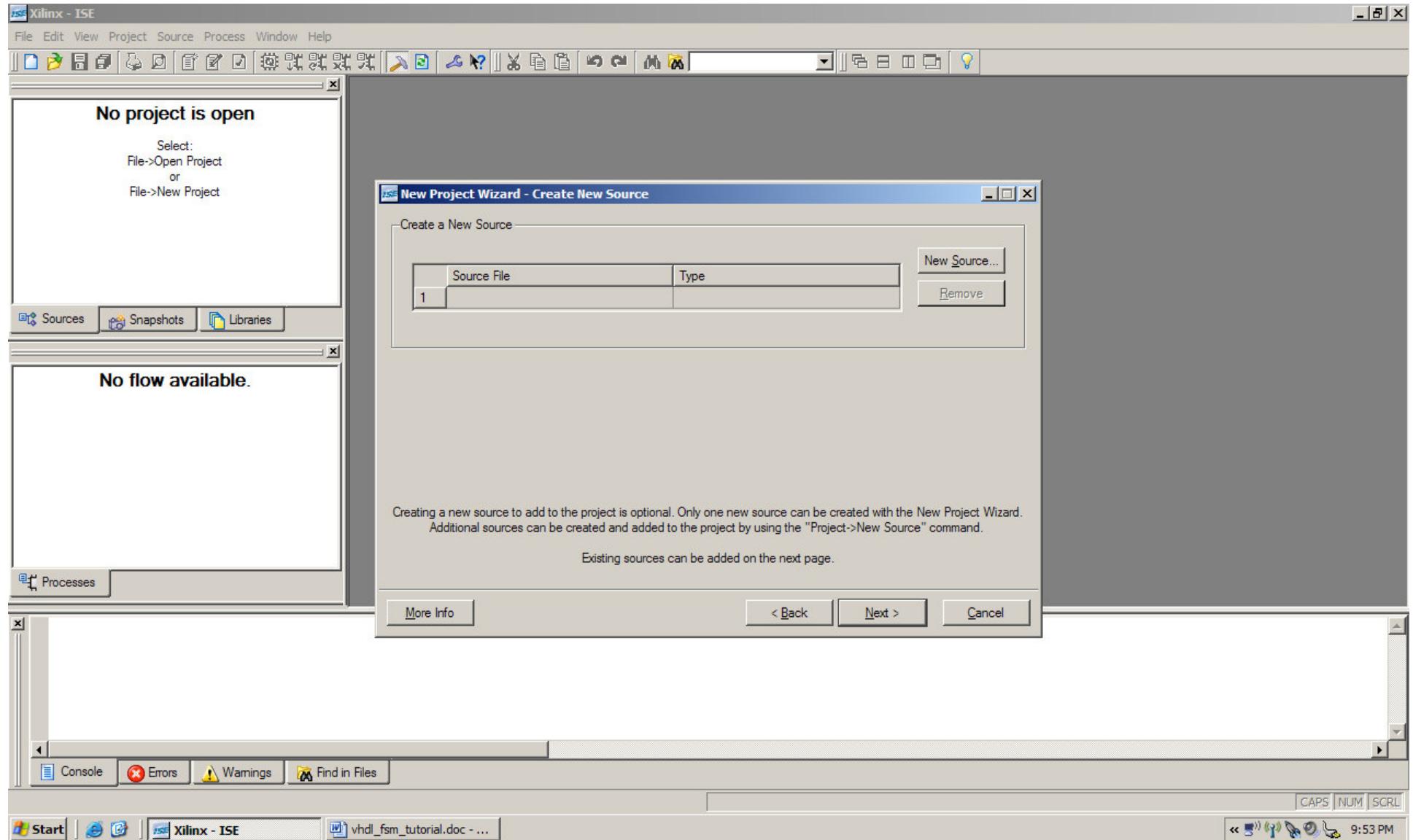
Start a new project called 140LTutorial2. In this tutorial, we will learn how to create a simple finite state machine using VHDL. The finite state machine will detect odd number of ones. There will be 3 inputs (data_in, rst, clk) and one output (odd_ind).



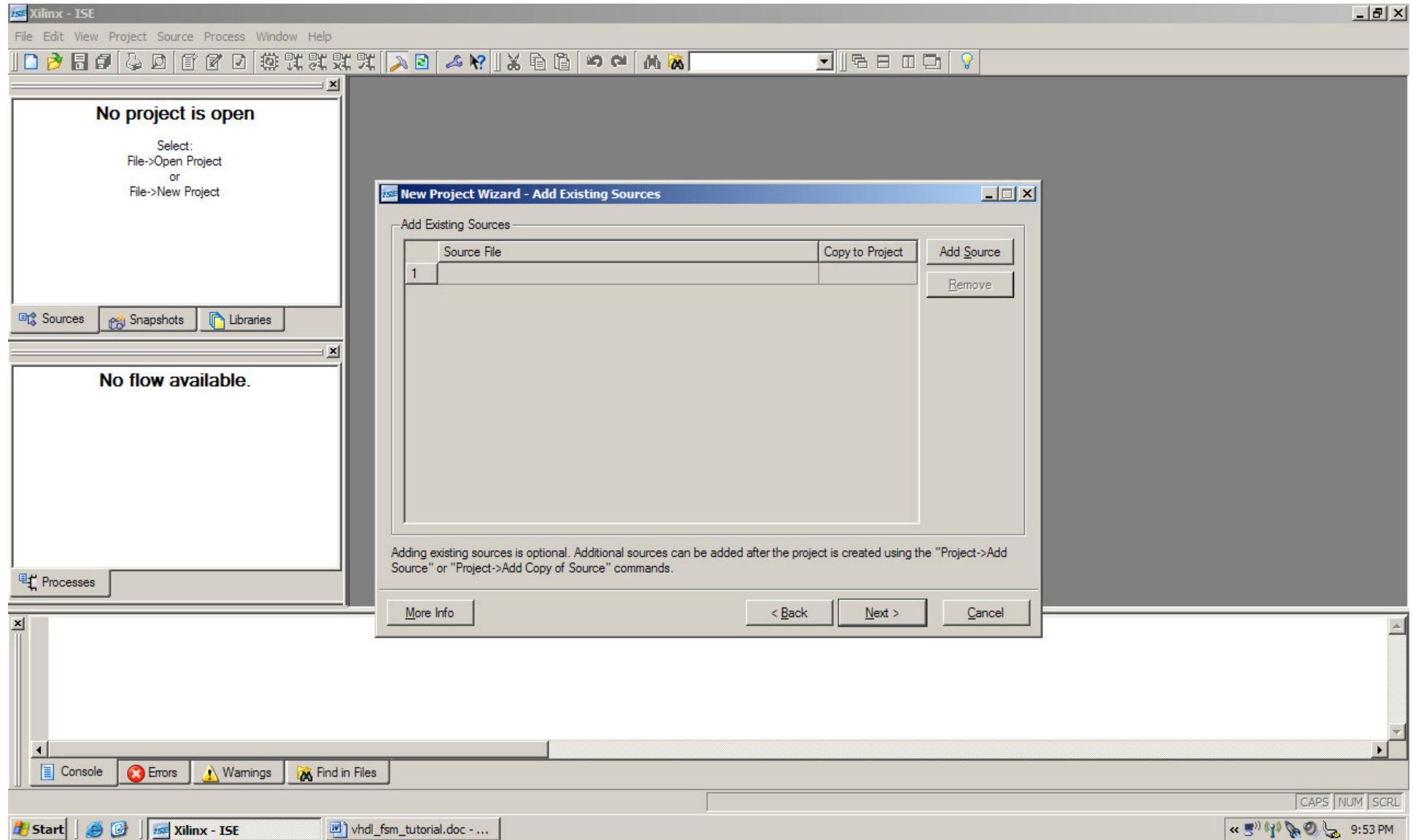
Enter the project name 140LTutorial2. Be sure to select HDL as the Top-Level Source Type.
Click Next to continue.



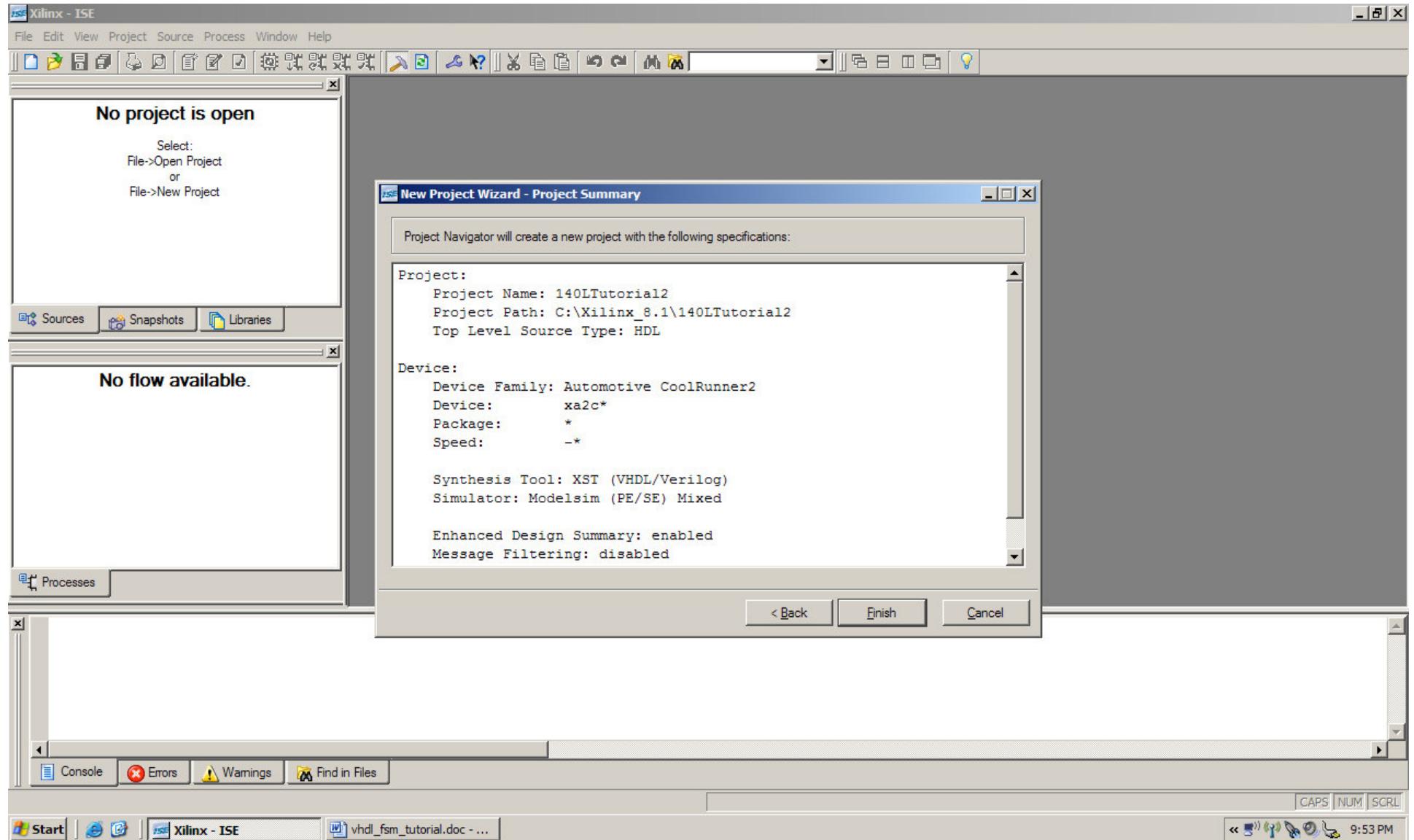
Leave everything as default... Your simulator selection should have been selected as Modelsim-XE Verilog in the previous labs already. Click Next to continue.



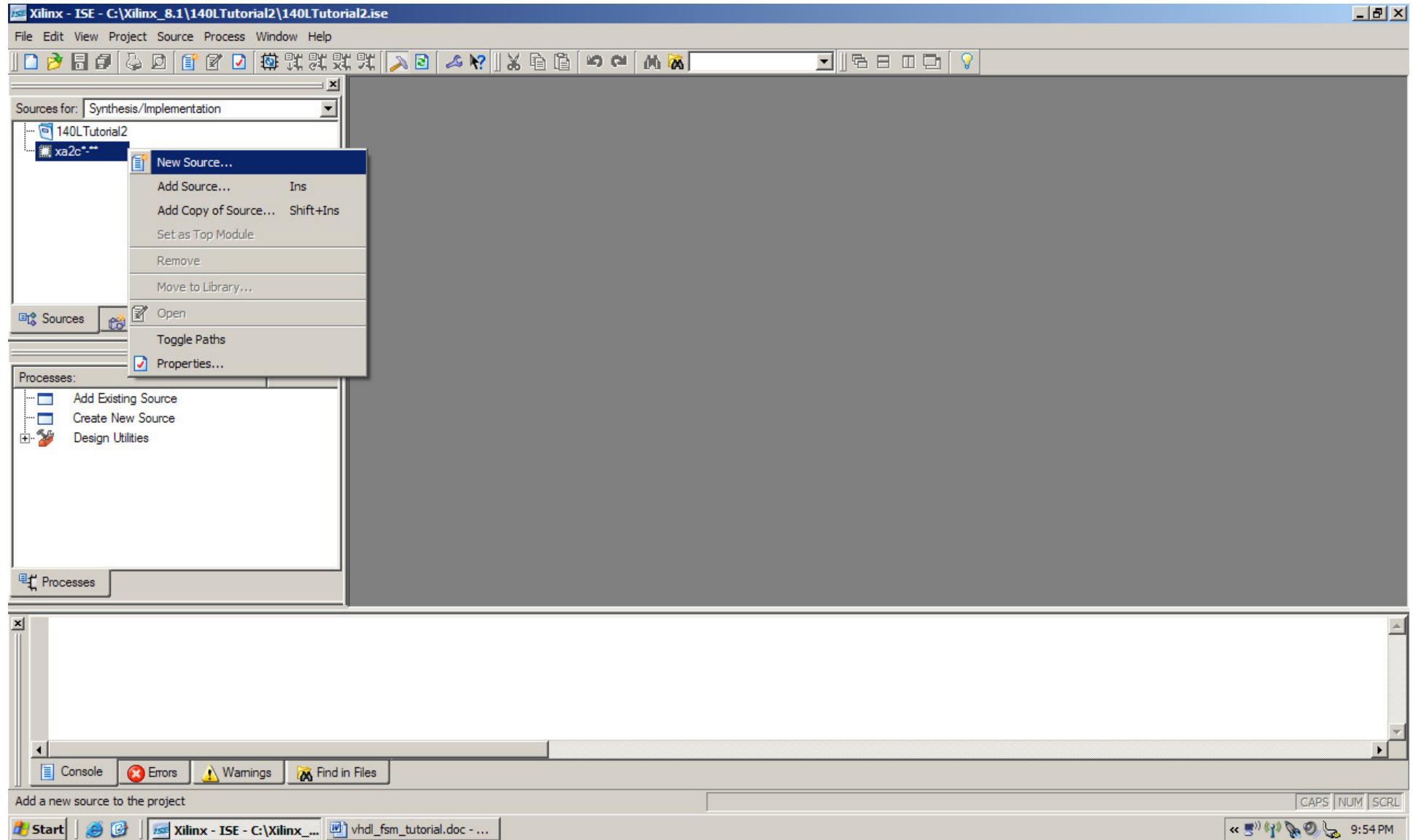
Leave everything as default. You can always add new source files later.
Click Next to continue.



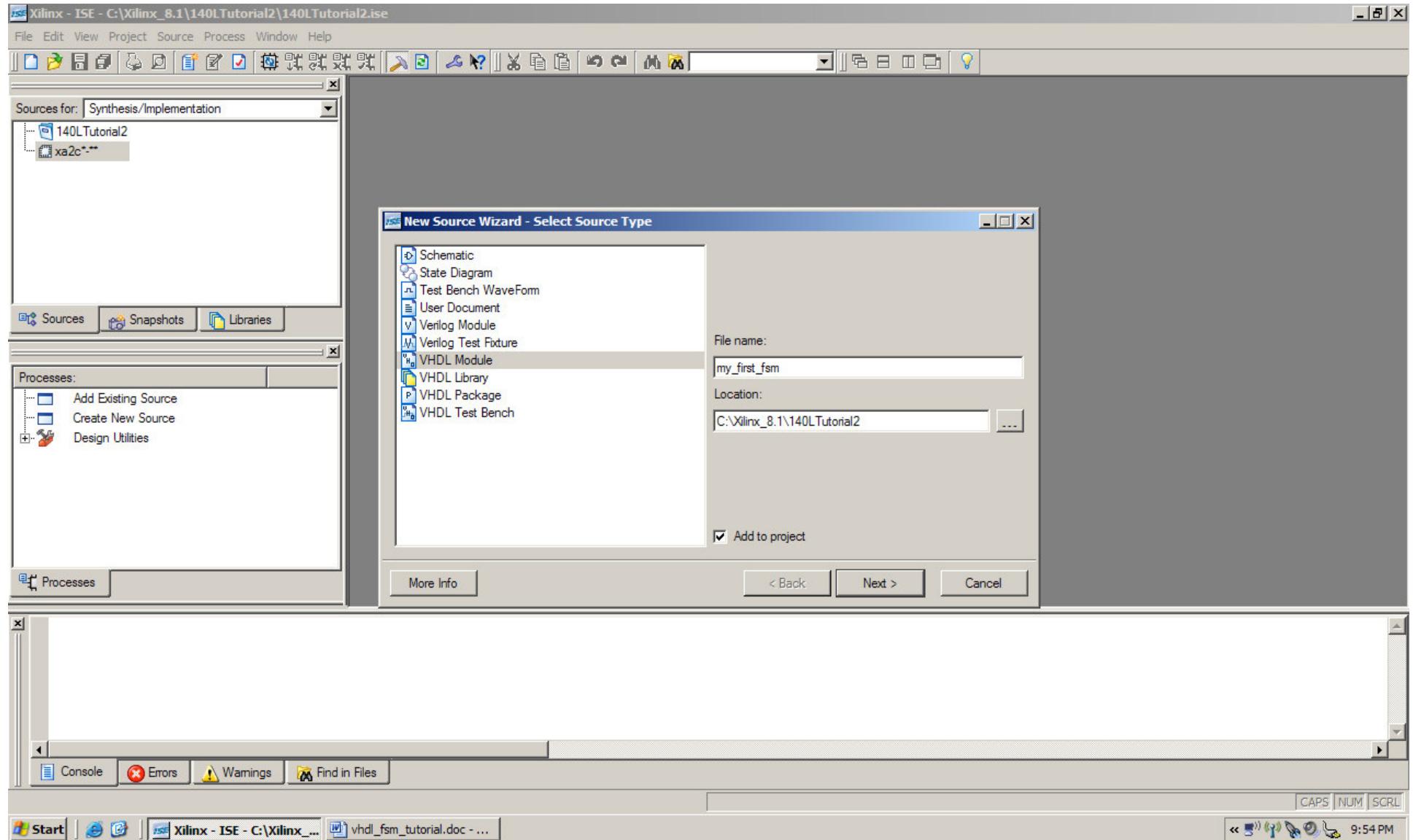
Since you don't have existing sources to add, you can skip this step also.
Click Next to continue.



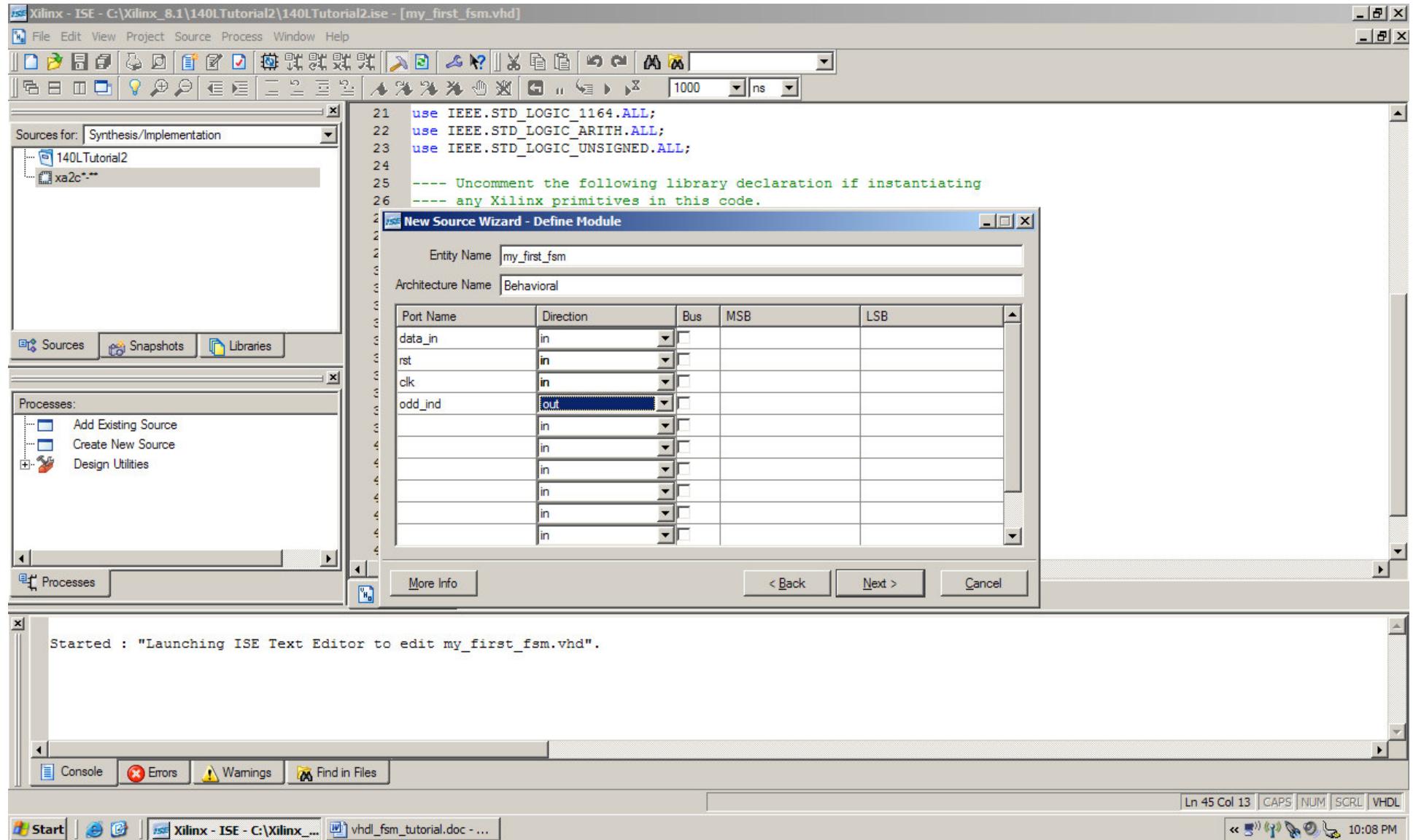
A quick summary... click Finish



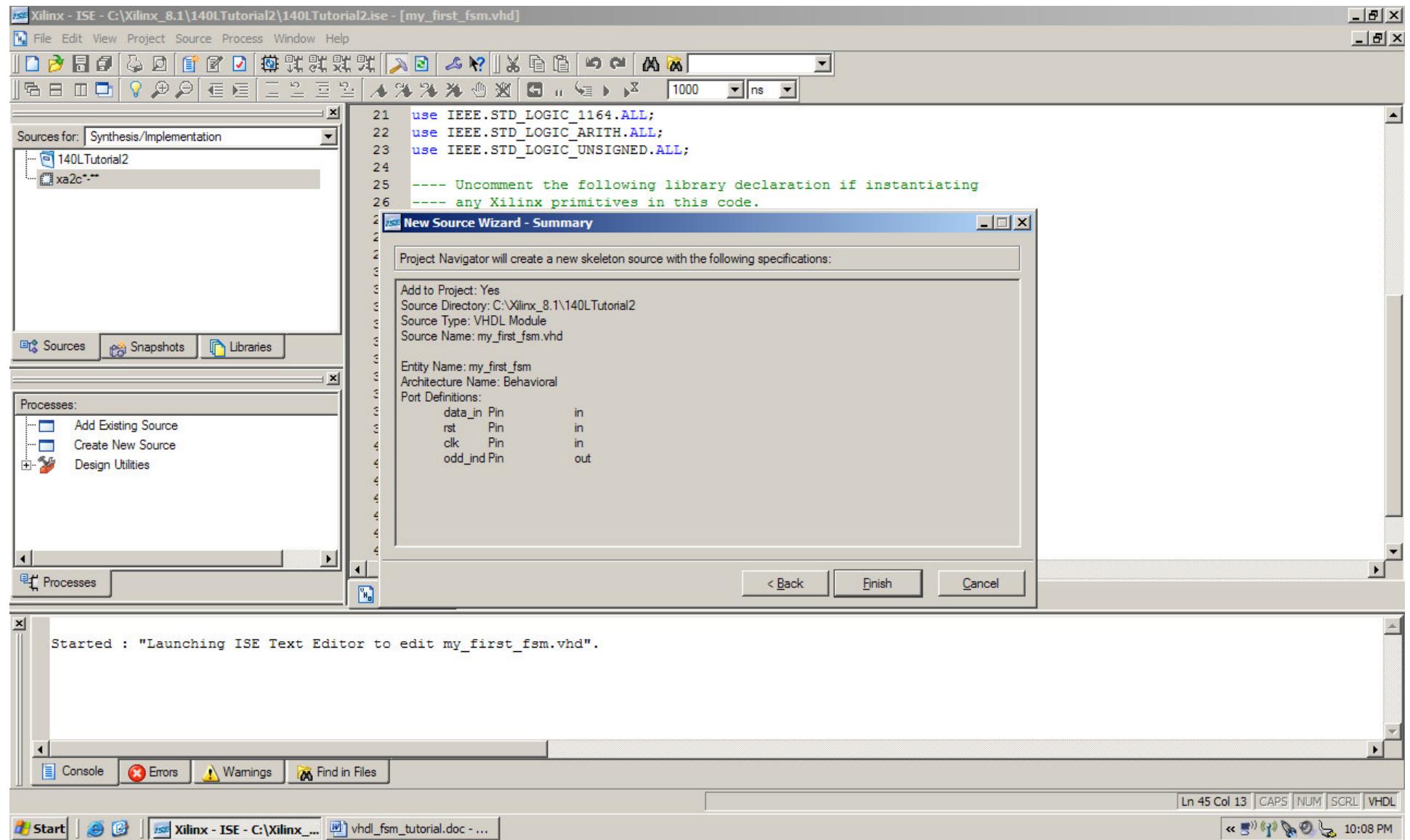
Now, let's create a new VHDL source.



Make sure you select VHDL Module on the left pane and enter the name “my_first_fsm”
Click Next to continue.



Enter three inputs (data_in, rst, clk) and one output (odd_ind) to indicate an even or odd number of ones detected from the data_in every clock cycle. Click Next to continue.



A quick summary... click Finish.

Xilinx - ISE - C:\Xilinx_8.1\140LTutorial2\140LTutorial2.ise - [my_first_fsm.vhd]

File Edit View Project Source Process Window Help

Sources for: Synthesis/Implementation

140LTutorial2
xa2c**
my_first_fsm - Behavioral (my_first_fsm.vhd)

Sources Snapshots Libraries

Processes:

- Add Existing Source
- Create New Source
- Design Utilities
- User Constraints
- Implement Design

my_first_fsm

```
1 --- Company:  
2 --- Engineer:  
3 ---  
4 --- Create Date: 22:08:48 05/05/2006  
5 --- Design Name:  
6 --- Module Name: my_first_fsm - Behavioral  
7 --- Project Name:  
8 --- Target Devices:  
9 --- Tool versions:  
10 --- Description:  
11 ---  
12 --- Dependencies:  
13 ---  
14 --- Revision:  
15 --- Revision 0.01 - File Created  
16 --- Additional Comments:  
17 ---  
18 ---  
19 library IEEE;  
20 use IEEE.STD_LOGIC_1164.ALL;  
21 use IEEE.STD_LOGIC_UNSIGNED.ALL;  
22 use IEEE.STD_LOGIC_ARITH.ALL;  
23  
24 ---- Uncomment the following library declaration if instantiating  
25 ---- any Xilinx primitives in this code.
```

Started : "Launching ISE Text Editor to edit my_first_fsm.vhd".

Console Errors Warnings Find in Files

Ln 1 Col 1 CAPS NUM SCRL VHDL

Start Xilinx - ISE - C:\Xilinx ... vhdl_fsm_tutorial.doc ... 10:08 PM

The screenshot shows the Xilinx ISE 8.1 software interface. The main window displays the VHDL source code for 'my_first_fsm.vhd'. The code includes company and engineer information, creation details, and library declarations for IEEE logic standards. A message at the bottom indicates the file is being edited. The interface features a toolbar, a sources browser, a processes palette, and a status bar at the bottom.

Notice the 3 libraries already included as default.

Xilinx - ISE - C:\Xilinx_8.1\140LTutorial2\140LTutorial2.ise - [my_first_fsm.vhd]

File Edit View Project Source Process Window Help

Sources for: Synthesis/Implementation

140LTutorial2
xa2c**
my_first_fsm - Behavioral (my_first_fsm.vhd)

Processes:

- Add Existing Source
- Create New Source
- Design Utilities
- User Constraints
- Implement Design

my_first_fsm

```
19 library IEEE;
20 use IEEE.STD_LOGIC_1164.ALL;
21 use IEEE.STD_LOGIC_UNSIGNED.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD.TEXTIO.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 my_first_fsm is
31     Port ( data_in : in STD_LOGIC;
32             rst : in STD_LOGIC;
33             clk : in STD_LOGIC;
34             odd_out : out STD_LOGIC);
35 end my_first_fsm;
36
37 architecture Behavioral of my_first_fsm is
38
39 begin
40
41 end Behavioral;
42
43
44
```

Started : "Launching ISE Text Editor to edit my_first_fsm.vhd".

Console Errors Warnings Find in Files

Ln 44 Col 1 CAPS NUM SCRL VHDL

Start Xilinx - ISE - C:\Xilinx... vhd_fsm_tutorial.doc ...

10:09 PM

The screenshot shows the Xilinx ISE 8.1 software interface. The main window displays a VHDL source code for an entity named 'my_first_fsm'. The code includes library declarations for IEEE, port definitions with inputs 'data_in', 'rst', and 'clk', and an output 'odd_out', and an architecture section labeled 'Behavioral'. On the left, there's a 'Sources' panel showing the project structure with 'my_first_fsm - Behavioral (my_first_fsm.vhd)' selected. Below it is a 'Processes' panel with options like 'Add Existing Source' and 'Create New Source'. At the bottom, a 'Console' window shows the message 'Started : "Launching ISE Text Editor to edit my_first_fsm.vhd".' The status bar at the bottom right indicates 'Ln 44 Col 1' and shows the date and time '10:09 PM'.

Scroll down and you'll see the skeleton code built for you. Entity is the definition and architecture is the actual logic implementation.

Xilinx - ISE - C:\Xilinx_8.1\140LTutorial2\140LTutorial2.ise - [my_first_fsm.vhd]

File Edit View Project Source Process Window Help

Sources for: Behavioral Simulation

140LTutorial2
xa2c**
my_first_fsm - Behavioral (my_first_fsm.vhd)

Sources Snapshots Libraries

Processes:

- Add Existing Source
- Create New Source
- ModelSim Simulator
- Simulate Behavioral Model

Processes my_first_fsm

```
37 architecture Behavioral of my_first_fsm is
38     --- Add the type definition...
39     TYPE state IS (even, odd);
40     SIGNAL current_state, next_state: state;
41 begin
42     -- Add the code
43     PROCESS (rst, clk)
44     BEGIN
45         IF (rst = '1') THEN
46             current_state <= even;
47         ELSIF (clk'EVENT AND clk='1') THEN
48             current_state <= next_state;
49         END IF;
50     END PROCESS;
51
52     PROCESS (data_in, current_state)
53     BEGIN
54         CASE current_state IS
55             WHEN even =>
56                 odd_ind <= '0';
57                 IF (data_in = '1') THEN next_state <= odd;
58                 ELSE next_state <= even;
59                 END IF;
60             WHEN odd =>
61                 odd_ind <= '1';
62                 IF (data_in = '1') THEN next_state <= even;
63                 ELSE next_state <= odd;
64                 END IF;
65         END CASE;
66     END PROCESS;
67 end Behavioral;
```

Started : "Launching ISE Text Editor to edit my first fsm.vhd".

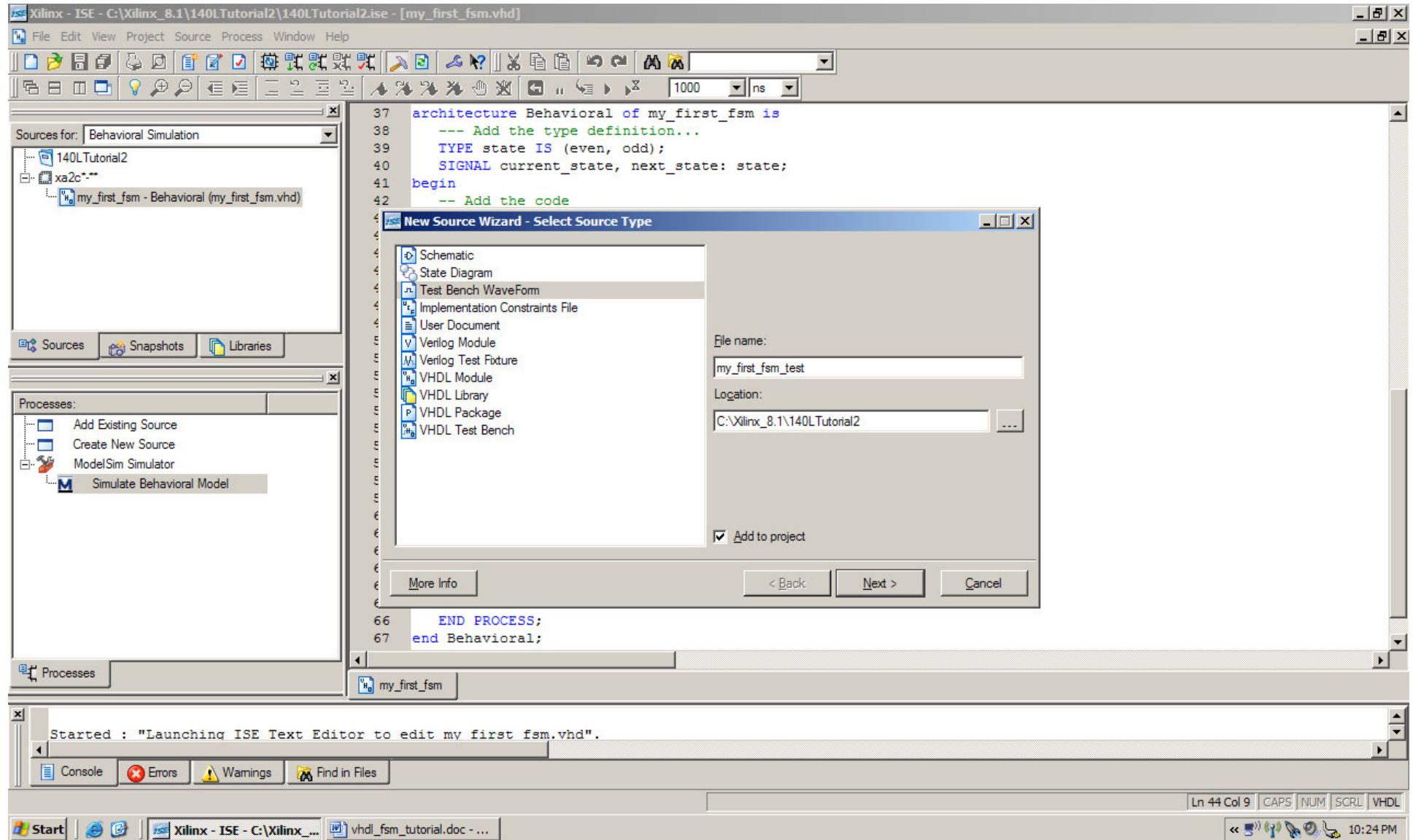
Console Errors Warnings Find in Files

Ln 44 Col 9 CAPS NUM SCRL VHDL

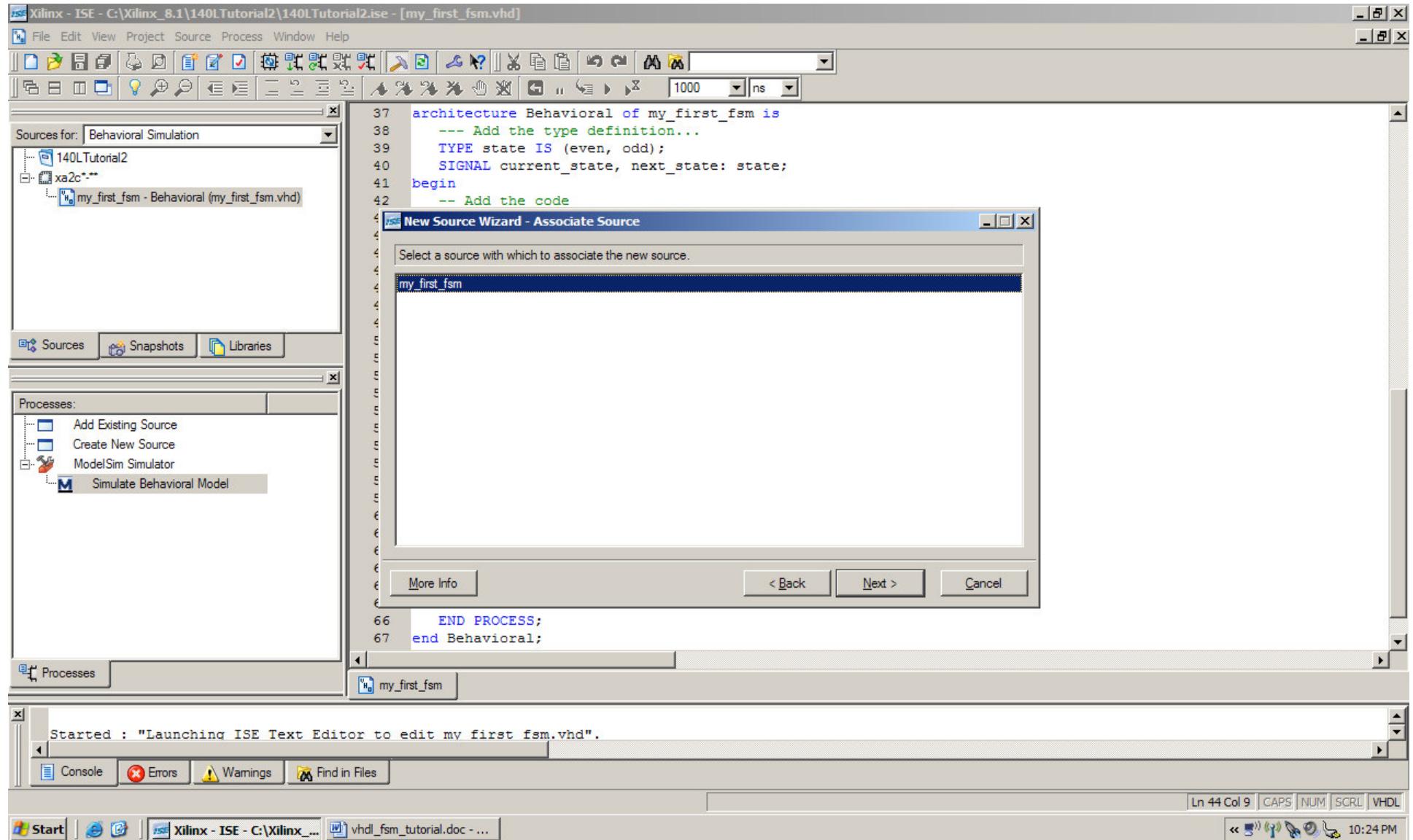
Start Xilinx - ISE - C:\Xilinx ... vhdl_fsm_tutorial.doc - ...

<< 10:23 PM

Add the code for the finite state machine.

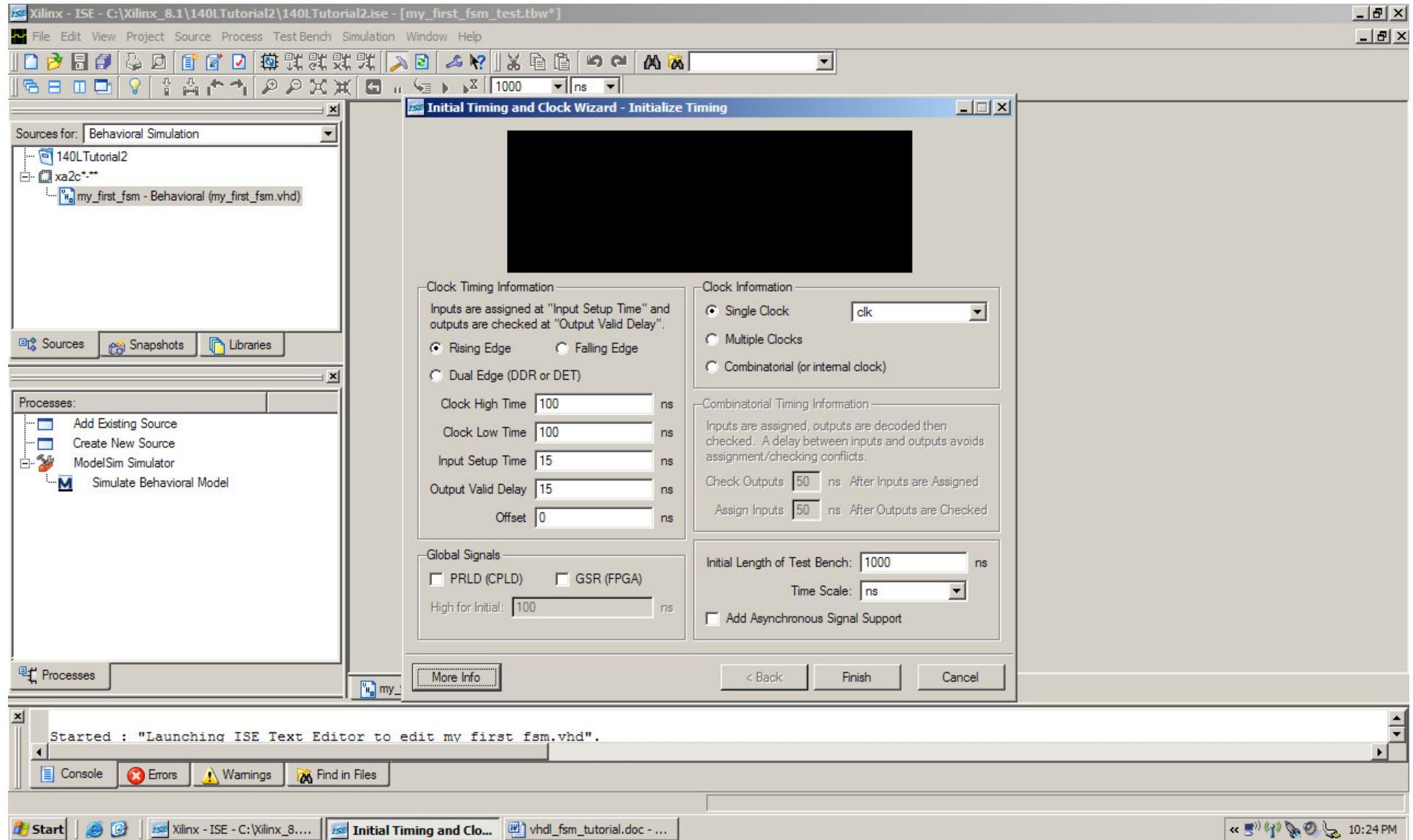


Now, add the test case for the state machine by right clicking on the file and choose “New Source” as you did in previous tutorial.

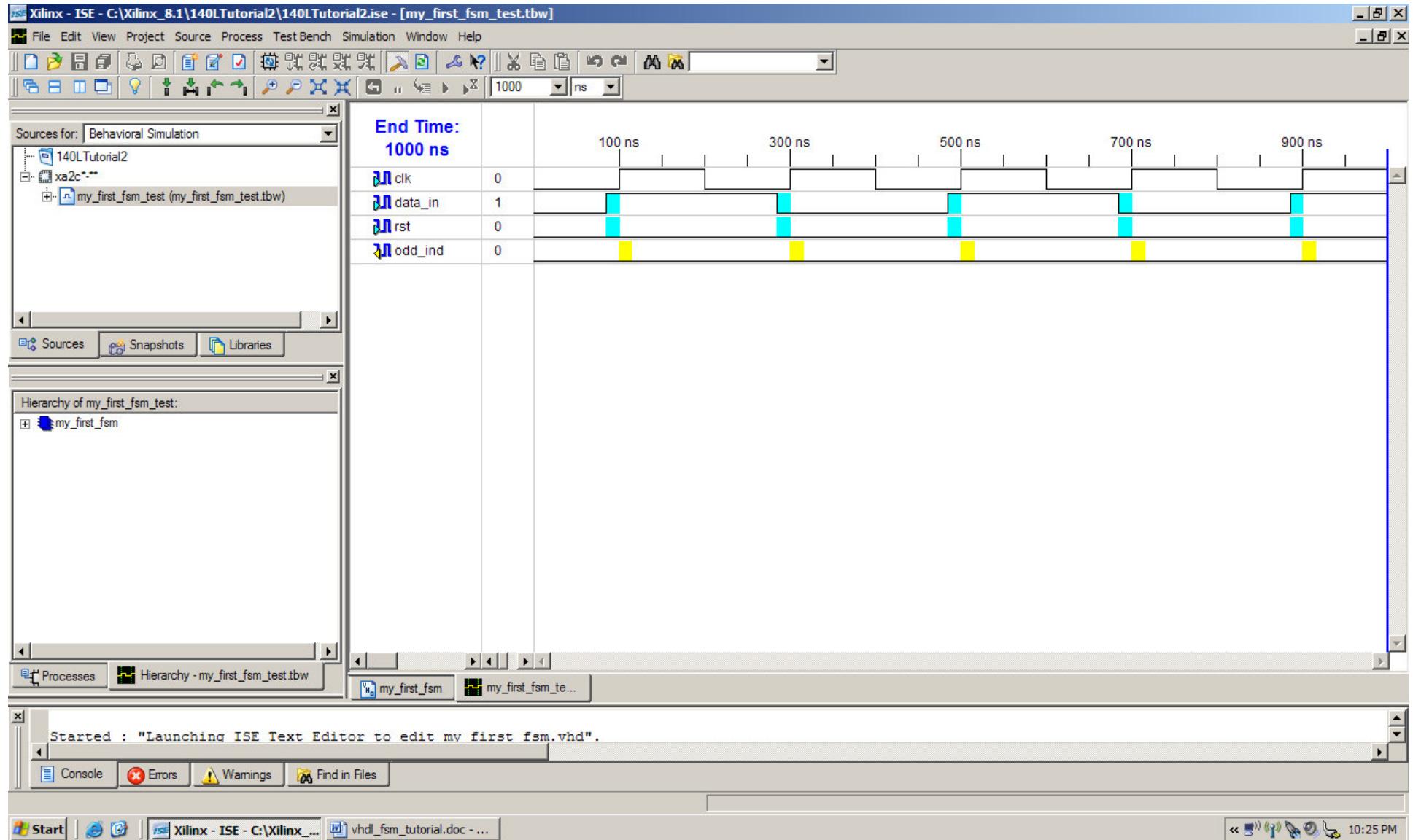


Choose the file you just created to be tested.

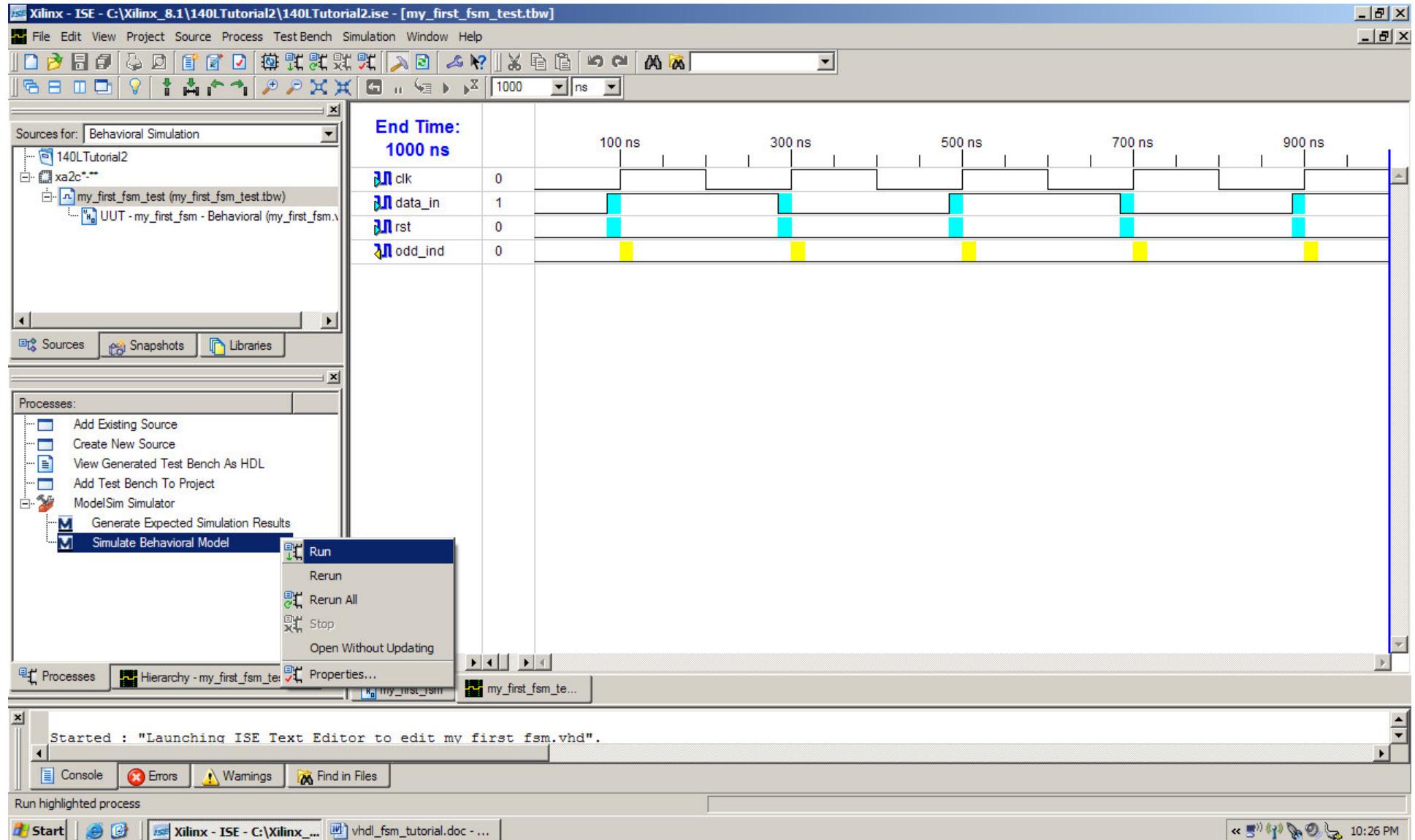
Click Next to continue.



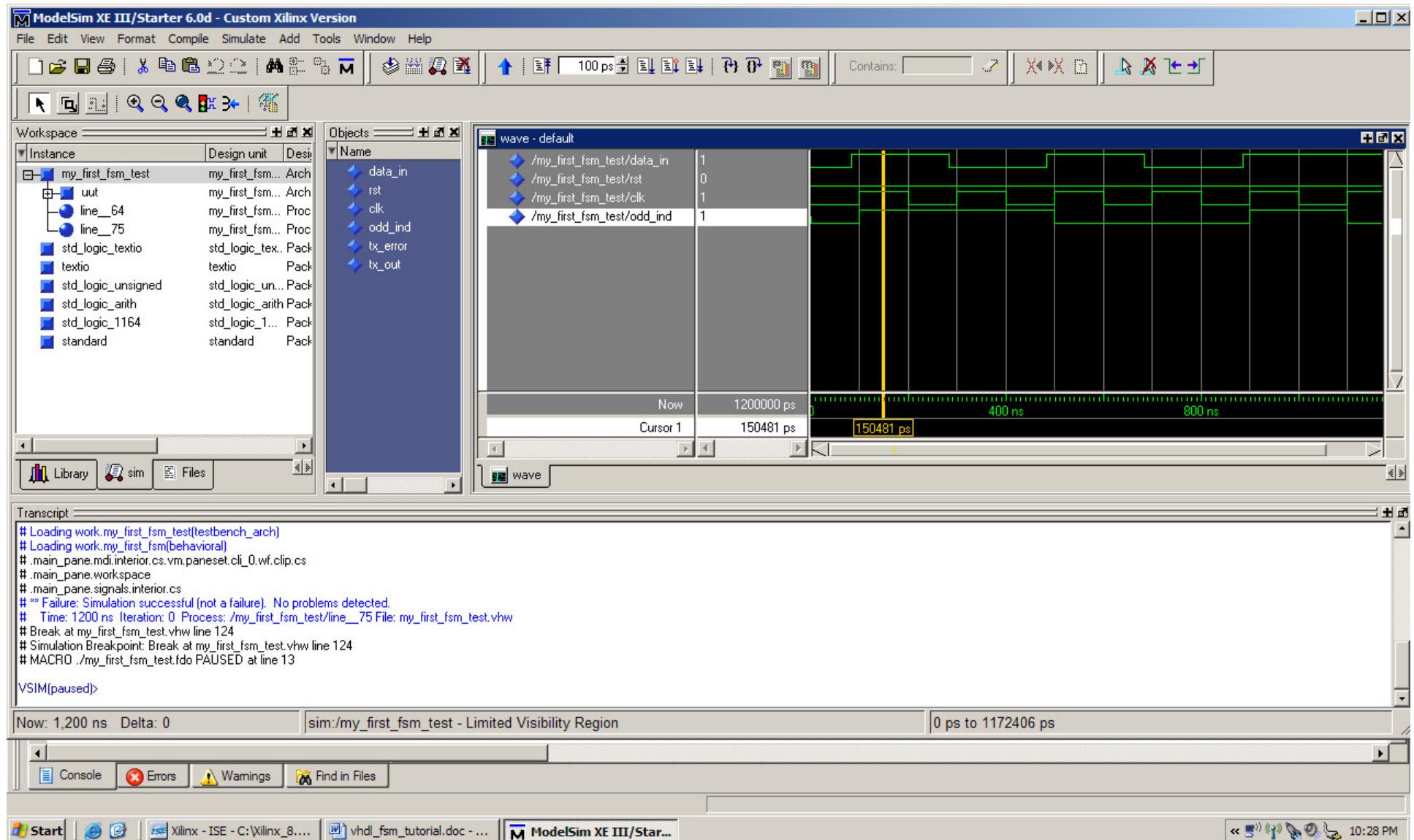
A new window will pop up. Be sure to select the Single Clock and select the “clk” as the clock line. Leave everything else as default.



Toggle the data_in line and set rst='0'.



Before running the simulation, you need to open the Simulation Behavioral Model's "Properties..." window and make sure "Testbench Model Target Language" is VHDL. Once you verify this, run the simulation.



If you typed everything in correctly, you will see the simulation result as expected.

Xilinx - ISE - C:\Xilinx_8.1\140LTutorial2\140LTutorial2.ise - [my_first_fsm.vhd]

File Edit View Project Source Process Window Help

Sources for: Behavioral Simulation

140LTutorial2
xa2c**
my_first_fsm_test (my_first_fsm_test.tbw)
UUT - my_first_fsm - Behavioral (my_first_fsm.vhd)

39 TYPE state IS (even, odd);
40 SIGNAL current_state, next_state: state;
41 begin
42 -- Add the code
43 PROCESS (rst, clk)
44 BEGIN
45 IF (rst = '1') THEN
46 current_state <= even;
47 ELSIF (clk'EVENT AND clk='1') THEN
48 current_state <= next_state;
49 END IF;
50 END PROCESS;
51
52 PROCESS (data_in, current_state)
53 BEGIN
54 CASE current_state IS
55 WHEN even =>
56 odd_ind <= '0';
57 IF (data_in = '1') THEN next_state <= odd;
58 ELSE next_state <= even;
59 END IF;
60 WHEN odd =>
61 odd_ind <= '1';
62 IF (data_in = '1') THEN next_state <= even;
63 ELSE next_state <= odd;
64 END IF;
65 -----END CASE;
66 END PROCESS;
67 end Behavioral;

Processes Hierarchy - my_first_fsm_test.tbw

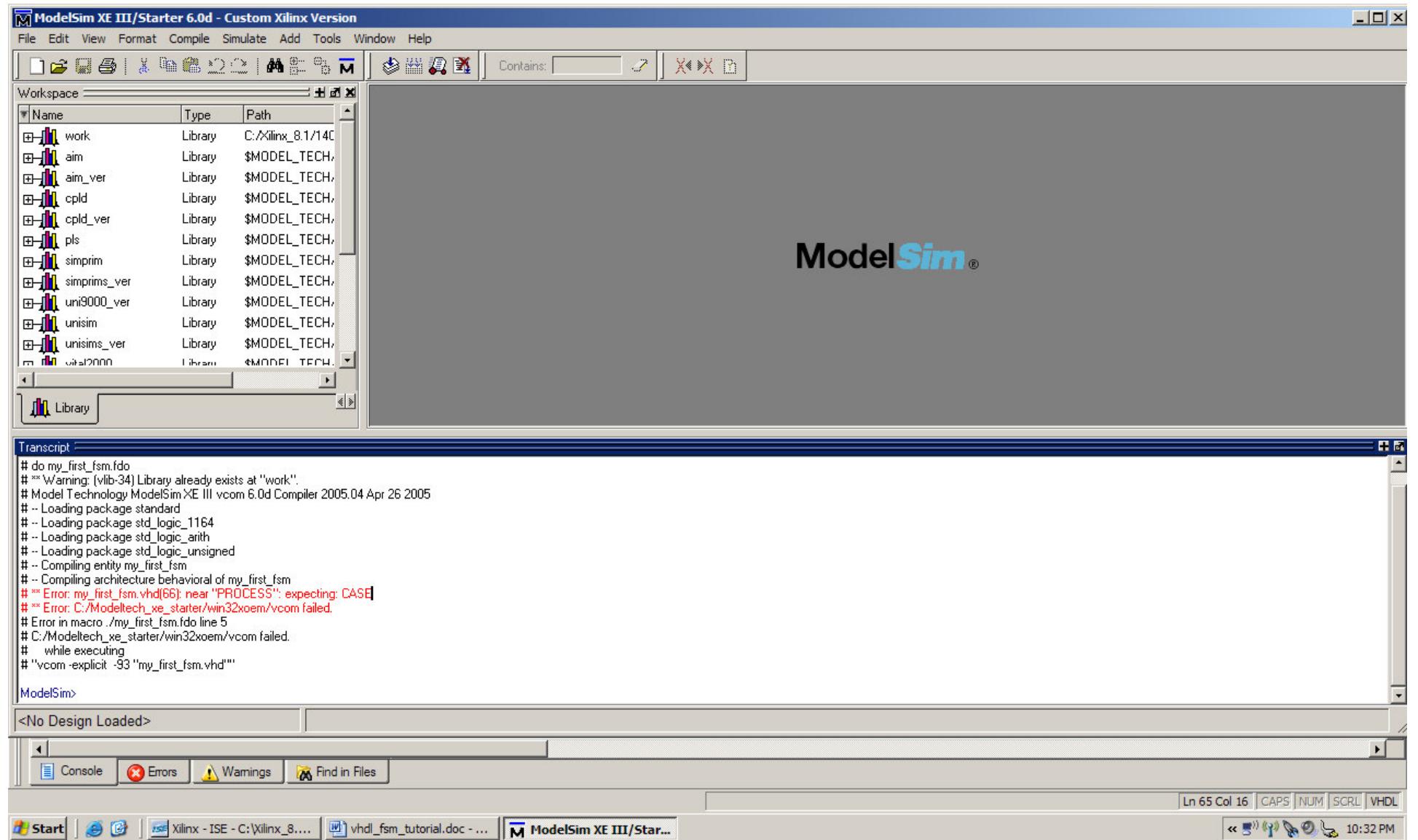
my_first_fsm my_first_fsm_te...

Console Errors Warnings Find in Files

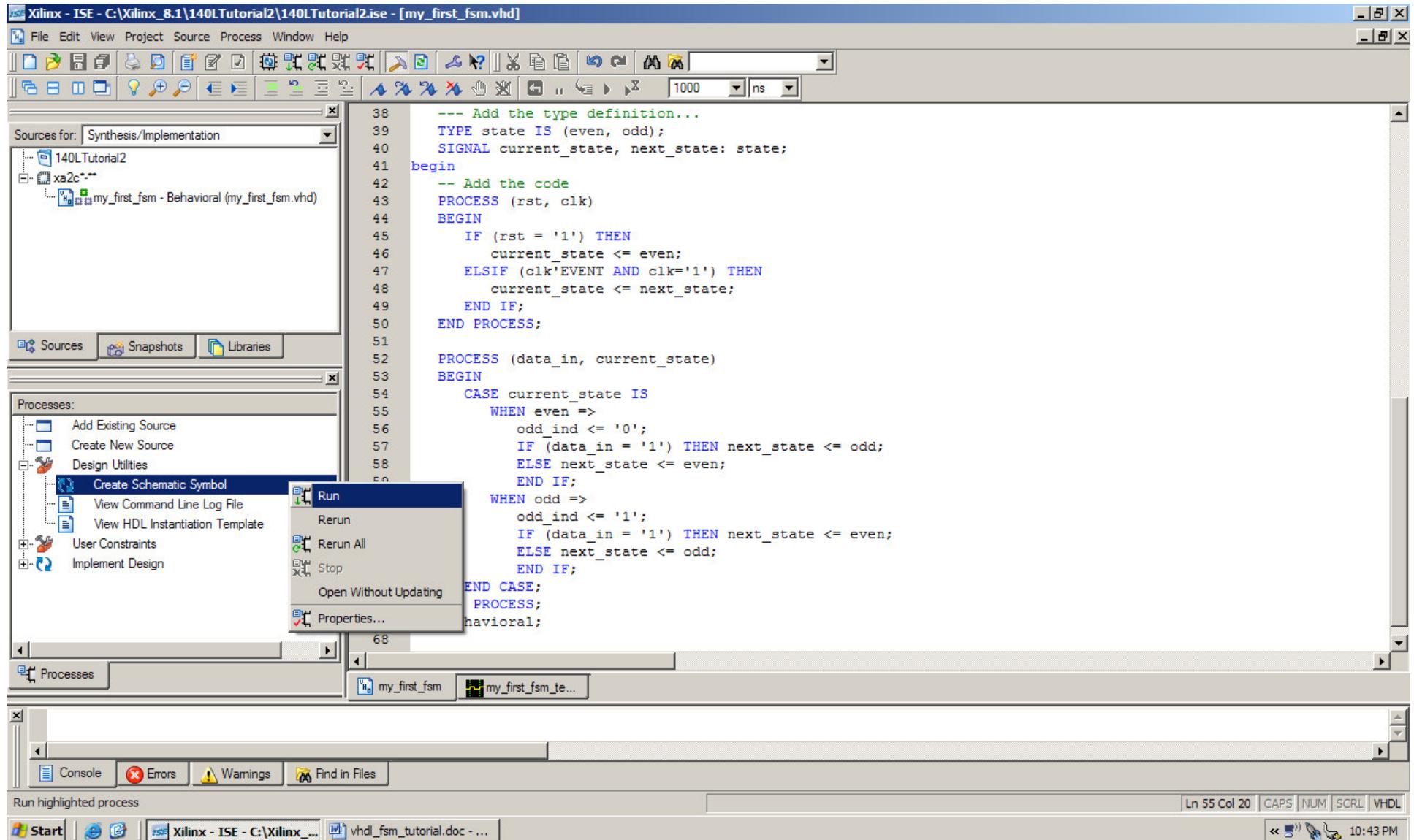
Ln 65 Col 16 CAPS NUM SCRL VHDL

Start Xilinx - ISE - C:\Xilinx_... vhdl_fsm_tutorial.doc - ... 10:30 PM

Let's force a mistake by commenting out the 'END CASE' statement and see what happens...



Notice the Modelsim will not run if there is an error.



After making the correction, let's see if we can see what the circuit will look like when we generate it.

Select the Sources for “Synthesis/Implementation” in the source upper left window.

In the Process window, drill down the “Design Utilities” and run the “Create Schematic Symbol”.

Also, right click on the “Implement Design” and click “Run”. This will translate your code to actual hardware.

Xilinx - ISE - C:\Xilinx_8.1\140LTutorial2\140LTutorial2.ise - [C:\Xilinx_8.1\140LTutorial2\my_first_fsm_html\fit\appletref.htm]

File Edit View Project Source Process Window Help

Sources for: Synthesis/Implementation

140LTutorial2
xa2c**
my_first_fsm - Behavioral (my_first_fsm.vhd)

Sources Snapshots Libraries

Processes:

- View HDL Instantiation Template
- User Constraints
- Implement Design**
 - Synthesize - XST
 - View Synthesis Report
 - View RTL Schematic
 - View Technology Schematic
 - Check Syntax
 - Translate
 - Ft
 - Generate Programming File
 - Optional Implementation Tools

Equation Display Style VHDL

XILINX CPLD Reports CoolRunner-II

Fitter Report | Timing Report

Fitter Report

Summary Errors/Warnings Logic Inputs Function Blocks Equations Pin List Compiler Options Text Report Help

Summary

Design Name	my_first_fsm
Fitting Status	Successful
Software Version	I.27
Device Used	XA2C32A-6-VQ44
Date	5-5-2006, 10:46PM

RESOURCES SUMMARY

Macrocells Used	Pterms Used	Registers Used	Pins Used	Function Block Inputs Used
1/32 (4%)	1/112 (1%)	1/32 (4%)	4/33 (13%)	1/80 (2%)

PIN RESOURCES

Signal Type	Required	Mapped	Pin Type	Used	Total
Input	1	1	I	0	1
Output	1	1	I/O	2	24

Process "Generate Programming File" completed successfully

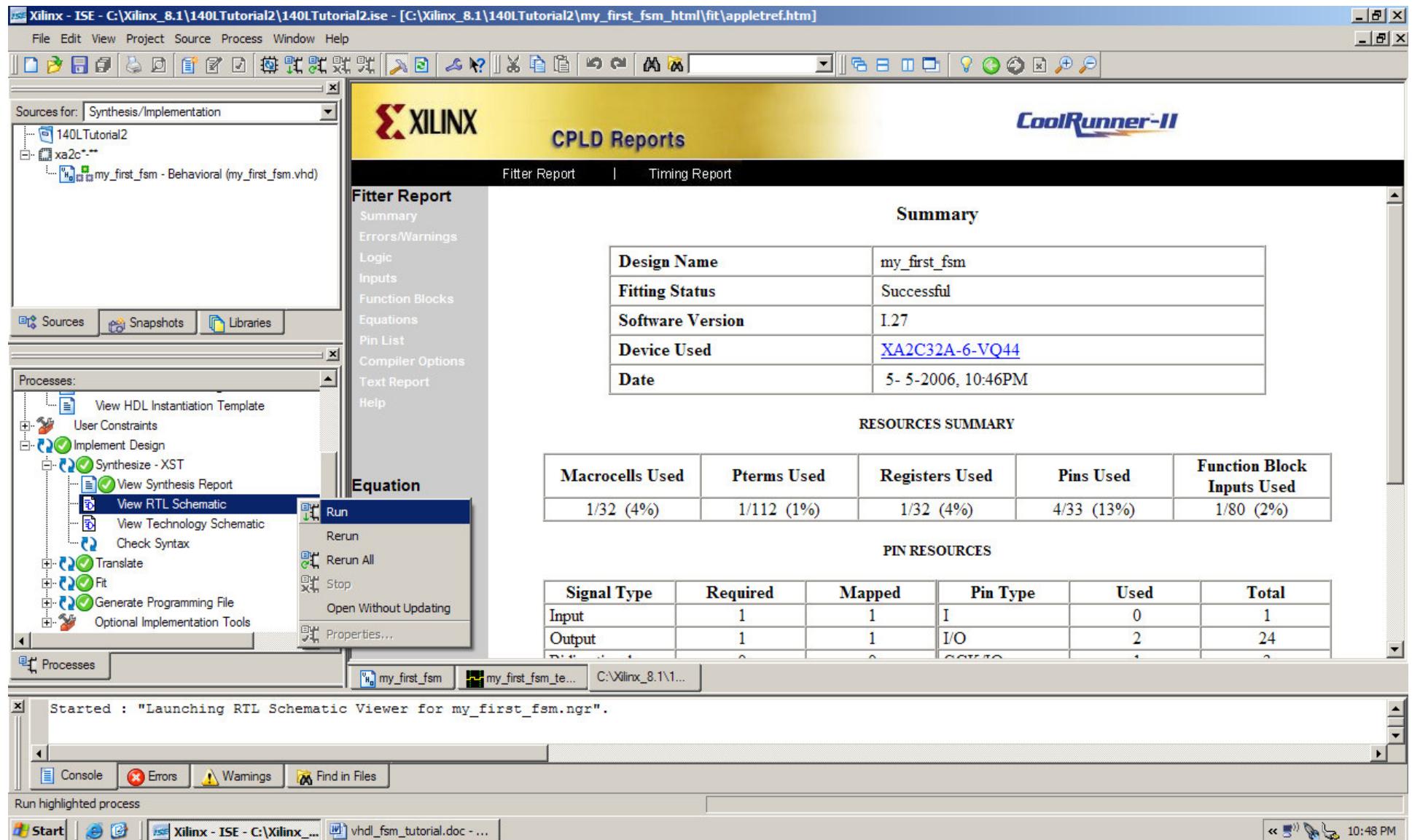
Console Errors Warnings Find in Files

Done

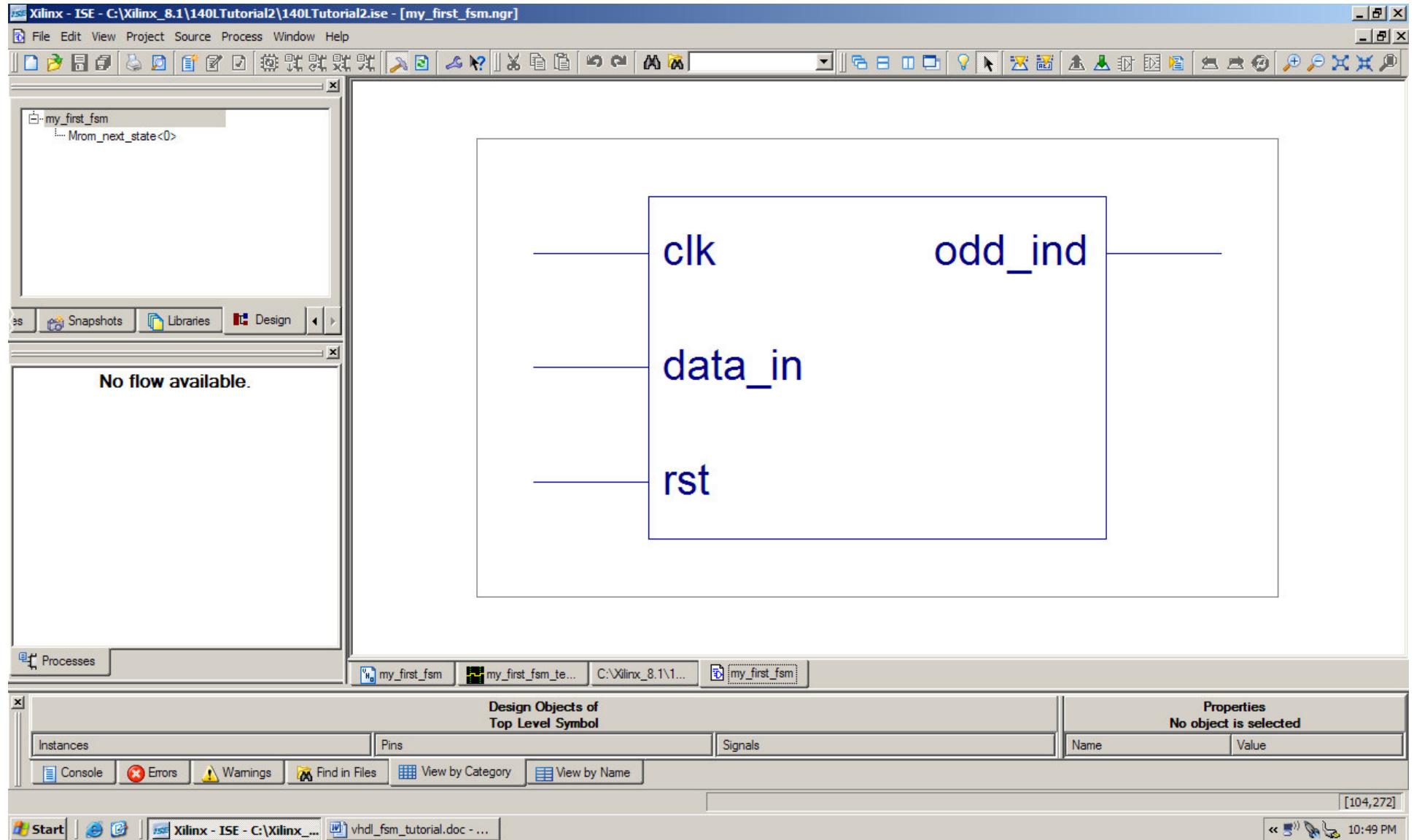
Start Xilinx - ISE - C:\Xilinx... vhdl_fsm_tutorial.doc - ... 10:47 PM

The screenshot shows the Xilinx ISE interface with the 'Fitter Report' selected. The main pane displays the 'Summary' section with details like Design Name (my_first_fsm), Fitting Status (Successful), Software Version (I.27), Device Used (XA2C32A-6-VQ44), and Date (5-5-2006, 10:46PM). Below the summary are 'RESOURCES SUMMARY' and 'PIN RESOURCES' tables. The 'PIN RESOURCES' table shows one input pin (I) and one output pin (I/O). At the bottom, a message indicates the 'Generate Programming File' process completed successfully.

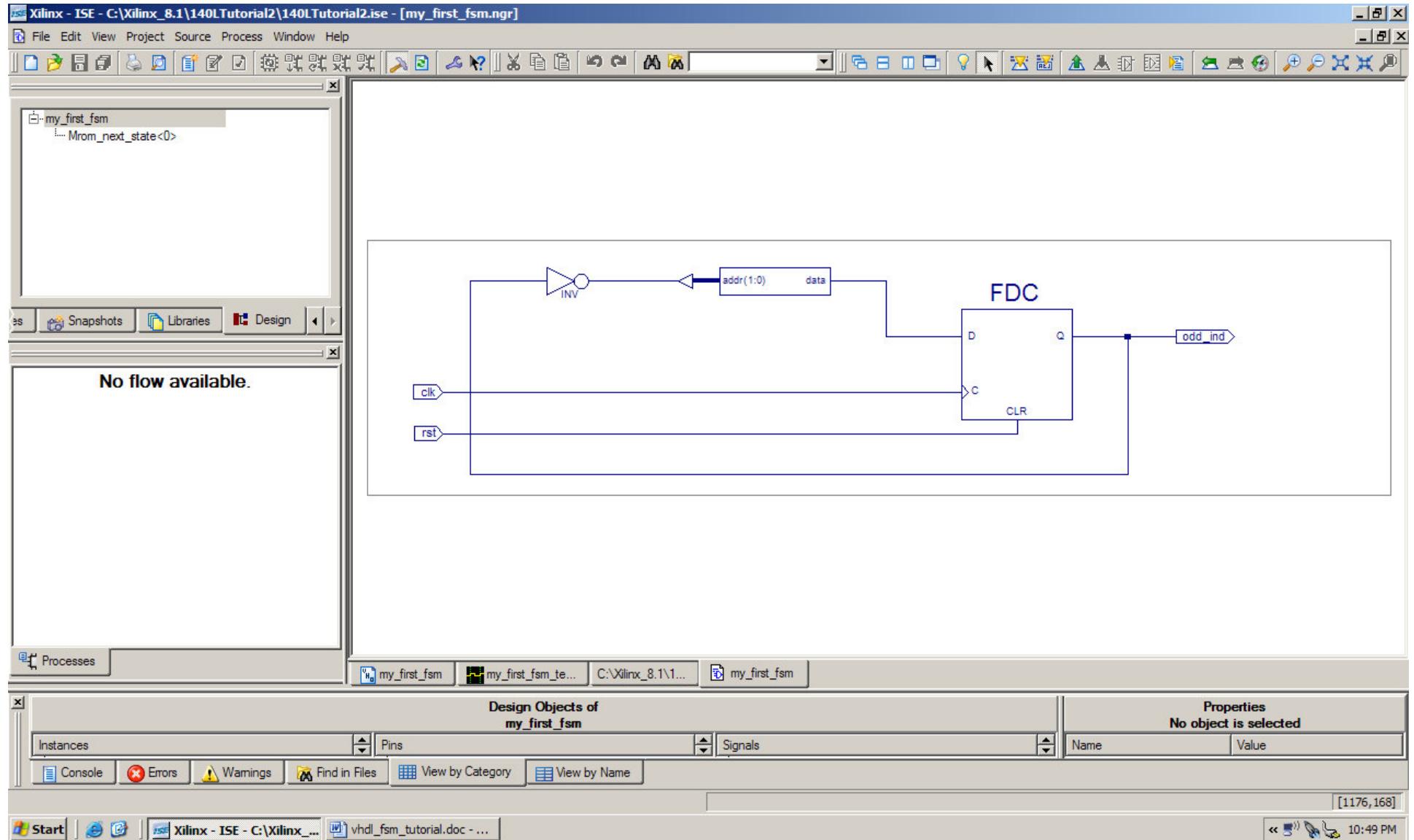
If all goes well, you can see all green check marks when you drill down the “Implementation Design”.



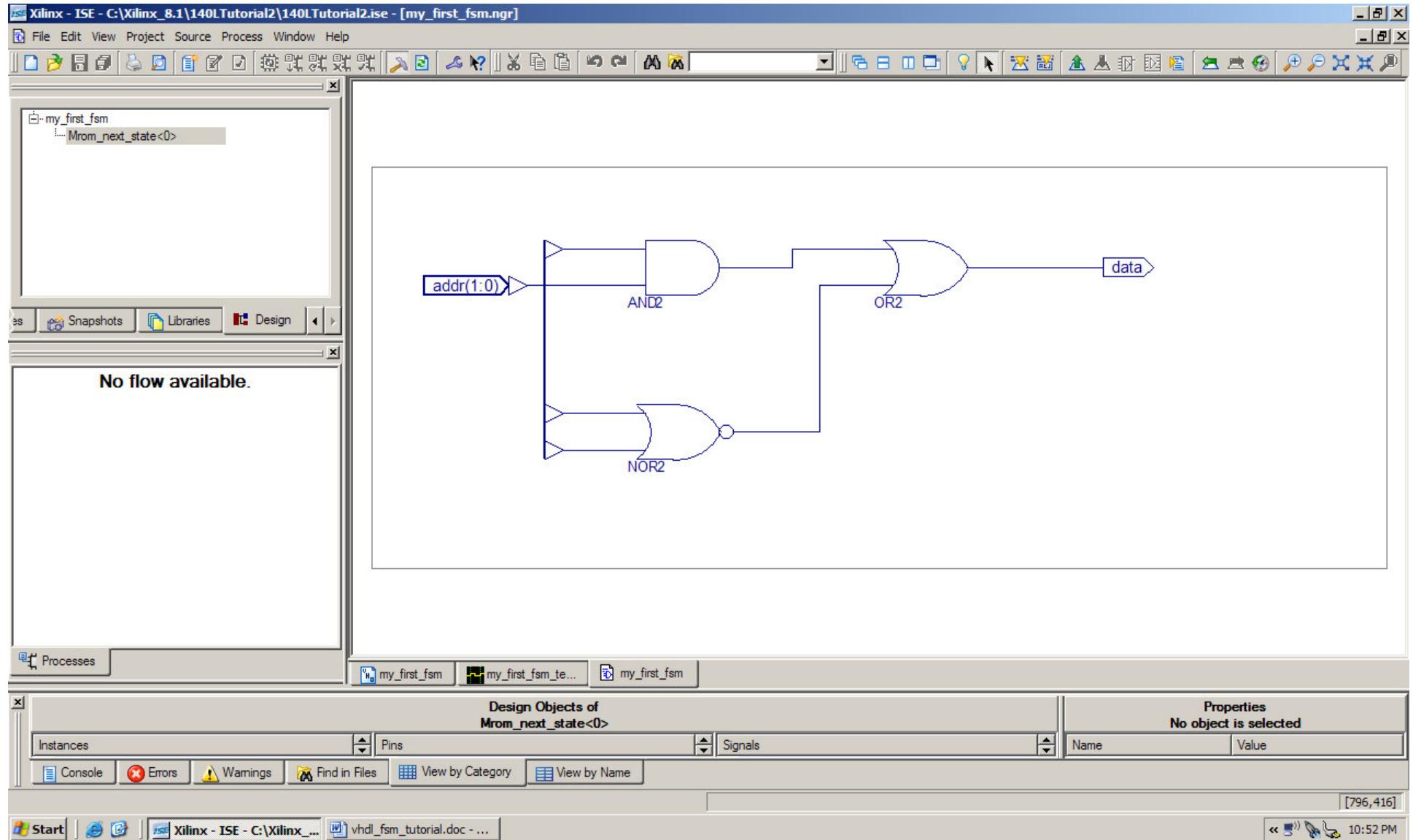
You can see the actual logic circuit by running the “View RTL Schematic”.



This is the top level view. Double click on the circuit itself to drill down the detail.



You can see the circuit actually created. Double click on the block next to the inverter.



You now see the logic gates which implemented the logic of this circuit.
Close this window...

Xilinx - ISE - C:\Xilinx_8.1\140LTutorial2\140LTutorial2.ise - [my_first_fsm.vhd]

File Edit View Project Source Process Window Help

Sources for: Synthesis/Implementation

140LTutorial2
xa2c**
my_first_fsm - Behavioral (my_first_fsm.vhd)

Processes:

- User Constraints
- Implement Design
- Synthesize - XST
 - View Synthesis Report
 - View RTL Schematic
 - View Technology Schematic
- Check Syntax
- Translate
- Fit
- Generate Programming File
- Optional Implementation Tools

Run

Rerun

Rerun All

Stop

Open Without Updating

Properties...

Started : "Launching RTL Schematic Viewer for my_fsm.ngr".

Console Errors Warnings Find in Files

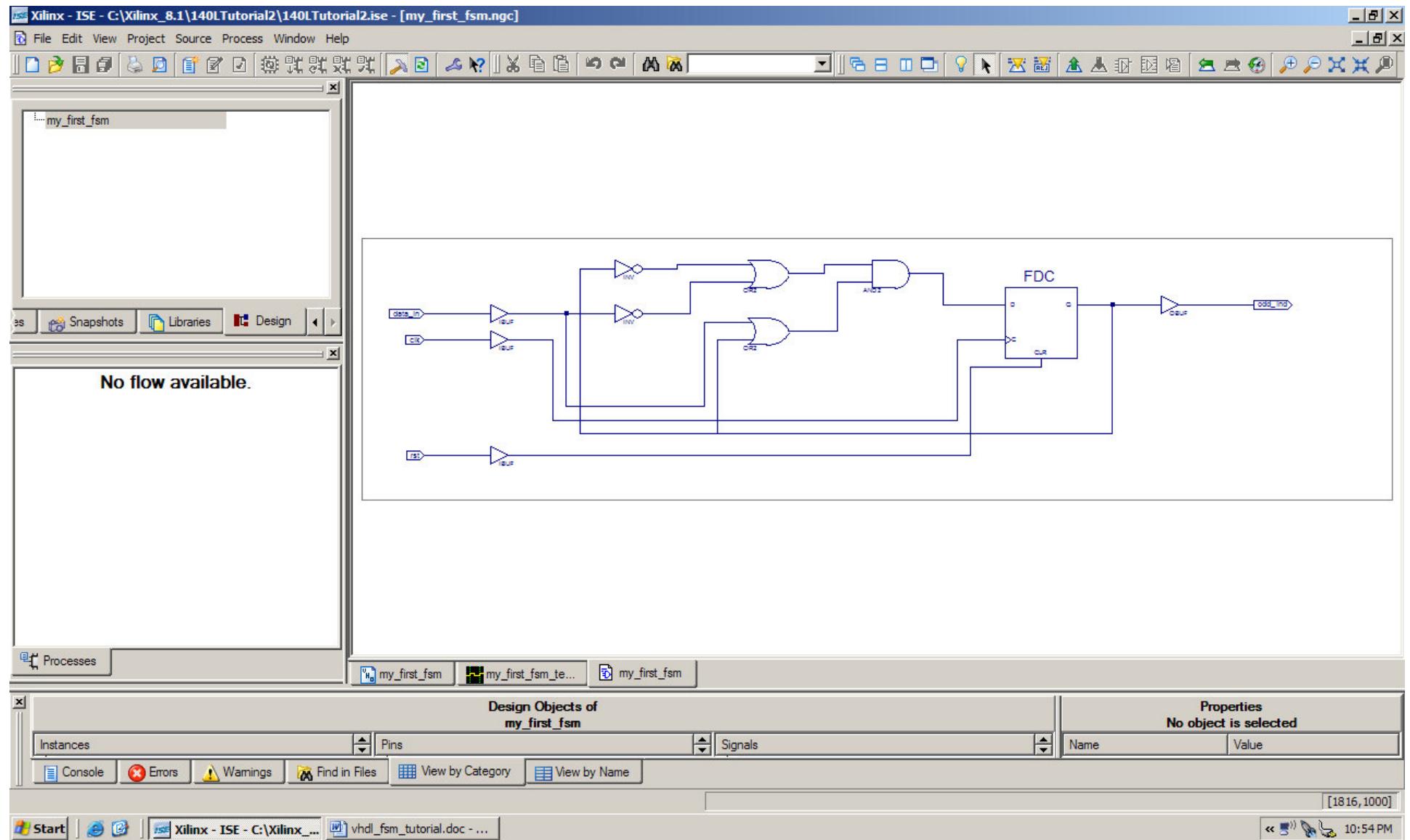
Run highlighted process [2112,-364] CAPS NUM SCRL VHDL

Start Xilinx - ISE - C:\Xilinx... vhd_fsm_tutorial.doc - ... 10:51 PM

```
38 --- Add the type definition...
39 TYPE state IS (even, odd);
40 SIGNAL current_state, next_state: state;
41 begin
42 -- Add the code
43 PROCESS (rst, clk)
44 BEGIN
45 IF (rst = '1') THEN
46     current_state <= even;
47 ELSIF (clk'EVENT AND clk='1') THEN
48     current_state <= next_state;
49 END IF;
50 END PROCESS;

51
52 PROCESS (data_in, current_state)
53 BEGIN
54 CASE current_state IS
55 WHEN even =>
56     odd_ind <= '0';
57     IF (data_in = '1') THEN next_state <= odd;
58     ELSE next_state <= even;
59     END IF;
56 WHEN odd =>
57     odd_ind <= '1';
58     IF (data_in = '1') THEN next_state <= even;
59     ELSE next_state <= odd;
60     END IF;
61 END CASE;
62 D PROCESS;
63 behavioral;
```

You can also see the entire design by running the “View Technology Schematic”.



After double click on the top level circuit, you can see the entire design.
 Now you are ready to do lab3.part3. Good luck!