

# EGEC281 – FALL 2024

## LAB 1

### DESIGN AND SIMULATE GATES

Designing with VHDL  
Experiment 1A, and 1B

---

Rakesh Mahto, Ph.D.  
Office: E 314, California State University, Fullerton  
Office Hour: Monday and Wednesday 2:00 - 3:30 pm  
Or by appointment  
Zoom Meeting ID: 894 4126 5483  
Email: [ramahto@fullerton.edu](mailto:ramahto@fullerton.edu)  
Phone No: 657-278-7274

# LEARNING OBJECTIVES

---

- ✗ Implement logic gates.
- ✗ Implement logic design's based on DeMorgan's Law.
- ✗ Simulate your design to verify it works
- ✗ Introduction to ISE, and basic VHDL

# VHDL CODE INTRO

- ✗ VHDL designs have three parts: library, entity declaration, architecture declaration.
- ✗ Entity and Architecture are the textual descriptions of the circuit
- ✗ Signal assignment symbol `<=`
- ✗ End code with `;`
- ✗ Identifiers: Signal names (I/O ports) and labels (entity and architecture names).
- ✗ Not case sensitive. No convention for spaces and indentation.
- ✗ Label used in entity declaration must be used in architecture.
- ✗ Precedence: NOT, AND, OR

# START A NEW PROJECT

New Project@drmahto-rsch

**Project Name**  
Enter a name for your project and specify a directory where the project data files will be stored.

Project name:

Project location:

☒ Create project subdirectory

Project will be created at: /home/rmahto/project\_1

Enter  
AND\_3

Enter  
working directory on  
your computer or  
Flashdrive

When done entering data  
Click on Next>

# START A NEW PROJECT

New Project@drmahto-rsch

**Project Type**  
Specify the type of project to create.

☒ **RTL Project**  
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.  
☐ Do not specify sources at this time

☐ **Post-synthesis Project**  
You will be able to add sources, view device resources, run design analysis, planning and implementation.  
☐ Do not specify sources at this time

☐ **I/O Planning Project**  
Do not specify design sources. You will be able to view part/package resources.

☐ **Imported Project**  
Create a Vivado project from a Synplify, XST or ISE Project File.

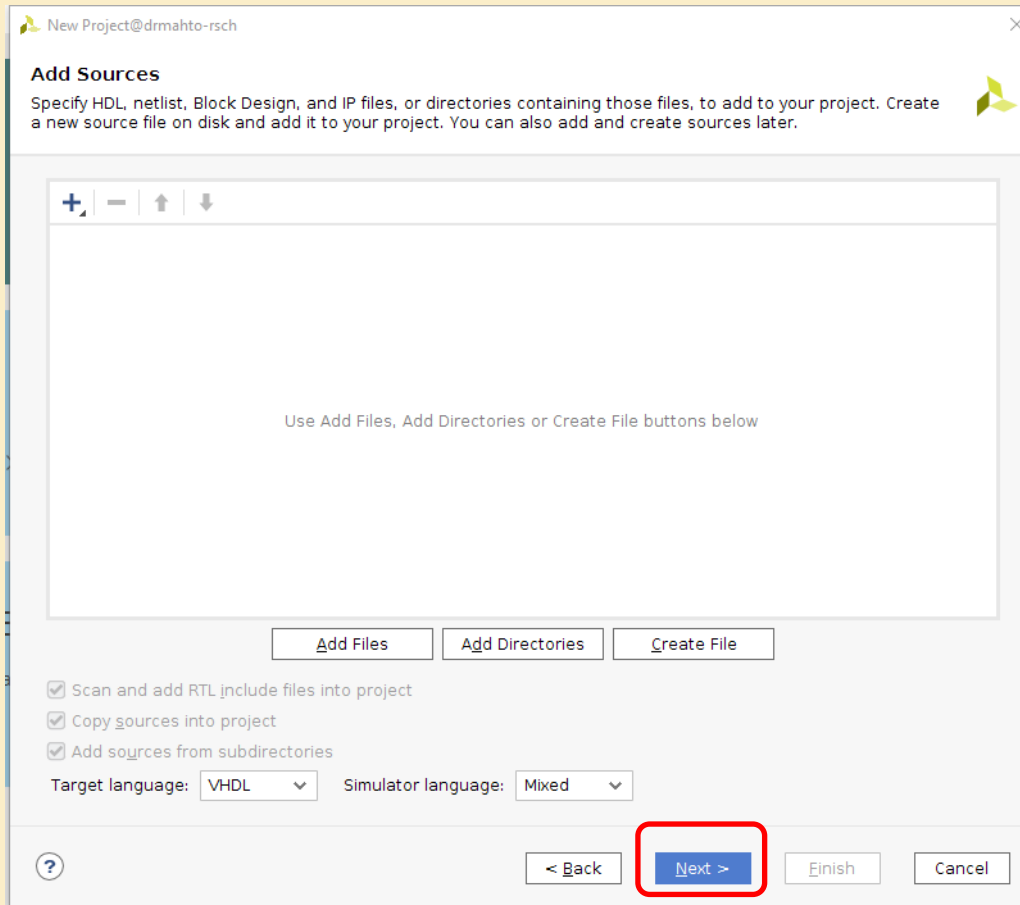
☐ **Example Project**  
Create a new Vivado project from a predefined template.

? < Back **Next >** Finish Cancel

Select RTL Project

When done entering data  
Click on Next>

# START A NEW PROJECT



New Project@drmahto-rsch

### Add Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

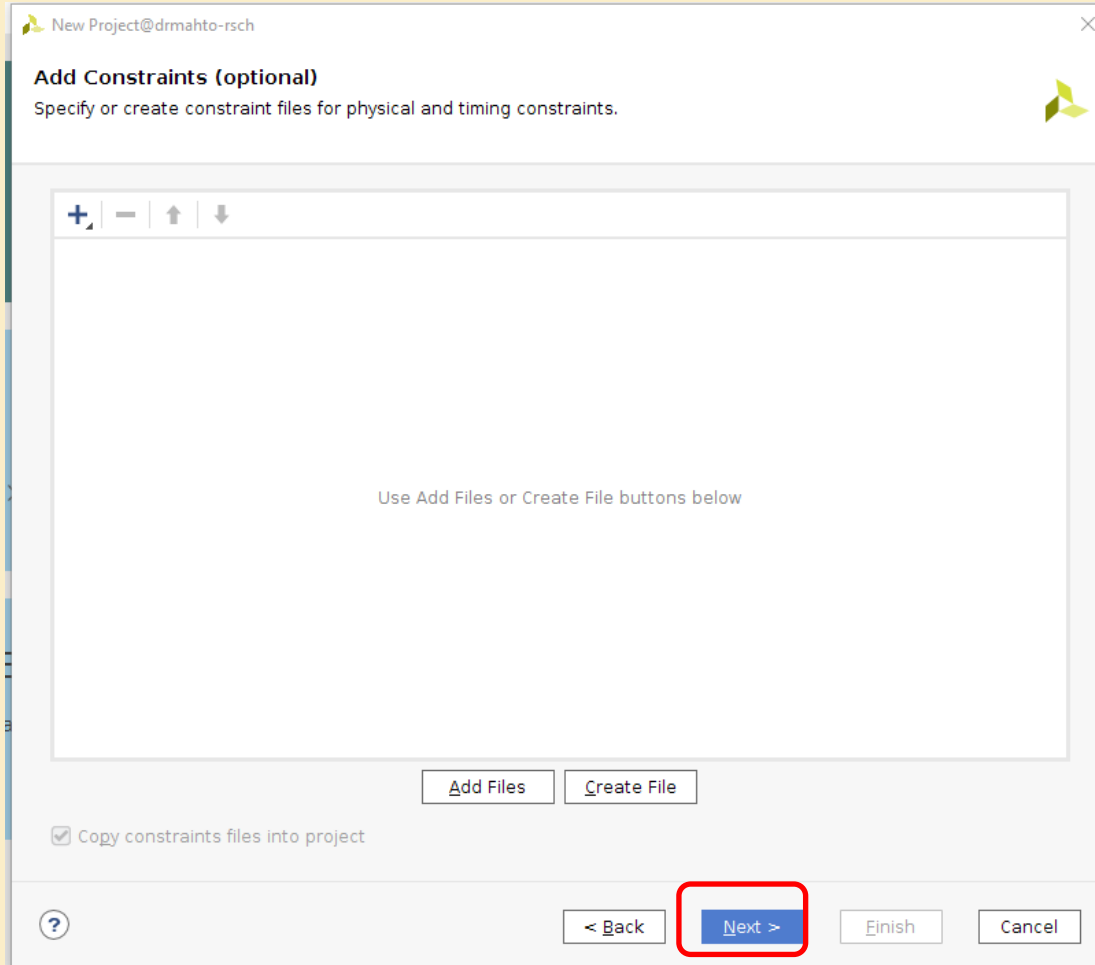
Use Add Files, Add Directories or Create File buttons below

☒ Scan and add RTL include files into project  
☒ Copy sources into project  
☒ Add sources from subdirectories

Target language: VHDL Simulator language: Mixed

Click Next

# START A NEW PROJECT



New Project@drmahto-rsch

**Add Constraints (optional)**  
Specify or create constraint files for physical and timing constraints.

Use Add Files or Create File buttons below

☐ Copy constraints files into project

Click Next

# SELECT BOARD AND LANGUAGE TYPE

New Project@drmahto-rsch

**Default Part**  
Choose a default Xilinx part or board for your project.

**Parts** | Boards

[Reset All Filters](#)

Category: All Package: csg324 Temperature: All Remaining  
Family: Artix-7 Low Voltage Speed: All Remaining Static power: All Remaining

Search: Q-

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs
xc7a15tlcsg324-2L	324	210	10400	20800	25	0	45
xc7a35tlcsg324-2L	324	210	20800	41600	50	0	90
xc7a50tlcsg324-2L	324	210	32600	65200	75	0	120
xc7a75tlcsg324-2L	324	210	47200	94400	105	0	180
xc7a100tlcsg324-2L	324	210	63400	126800	135	0	240

< Back Next > Finish Cancel

**ALWAYS Check These**

- Artix 7
- XC7A100T
- CSG324
- VHDL

**Debugging Tip:**

The fields marked in yellow are not always populated with data specific to the boards in the lab.

If these are not correct you will have errors when you attempt to generate the I/O Map.



# SELECT BOARD AND LANGUAGE TYPE

New Project Wizard

← Project Settings  
Specify device and project properties.

Select the device and design flow for the project

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Artix7
<b>Device</b>	XC7A100T
Package	CSG324
Speed	-2L
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

More Info < Back **Next >** Cancel

**ALWAYS Check These**

- Artix 7
- XC7A100T
- CSG324
- VHDL

Debugging Tip:

The fields marked in yellow are not always populated with data specific to the boards in the lab.

If these are not correct you will have errors when you attempt to generate the I/O Map.

Click on Next>

# COMPLETE THE PROJECT SETUP

New Project@drmahto-rsch

**VIVADO**  
HLS Editions

### New Project Summary

- A new RTL project name **and\_3** will be created.
- No source files or directories will be added. Use Add Sources to add them later.
- No constraints files will be added. Use Add Sources to add them later.
- The default part and product family for the new project:
  - Default Part: xc7a100tlcsg324-2L
  - Product: Artix-7
  - Family: Artix-7 Low Voltage
  - Package: csg324
  - Speed Grade: -2L

To create the project, click Finish

**XILINX**

? < Back Next > Finish Cancel

Project Name you Entered

Working directory on  
your computer or  
Flashdrive (Flashdrive  
is better)

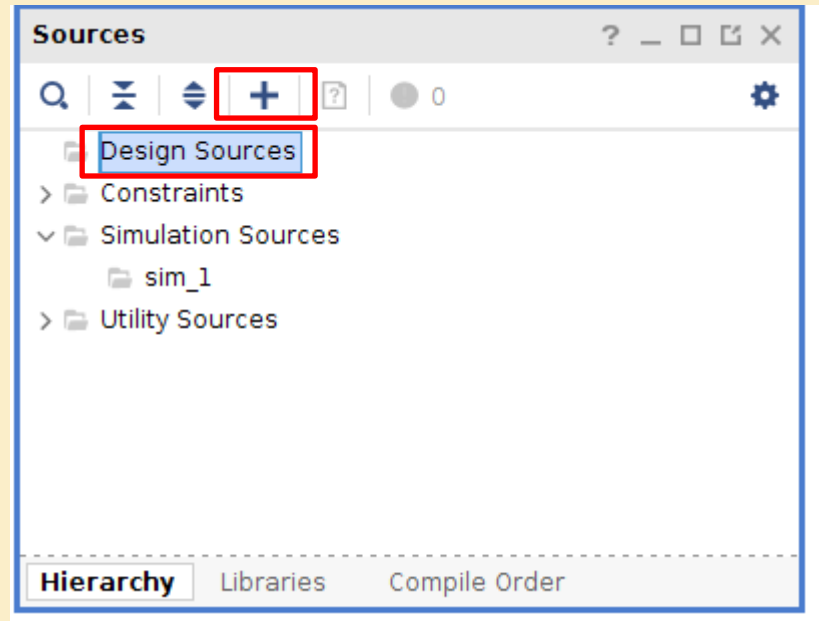
Check that these are  
setup correctly

If things do not look right  
Click on <Back and re-enter data

When done checking data  
Click on Finish

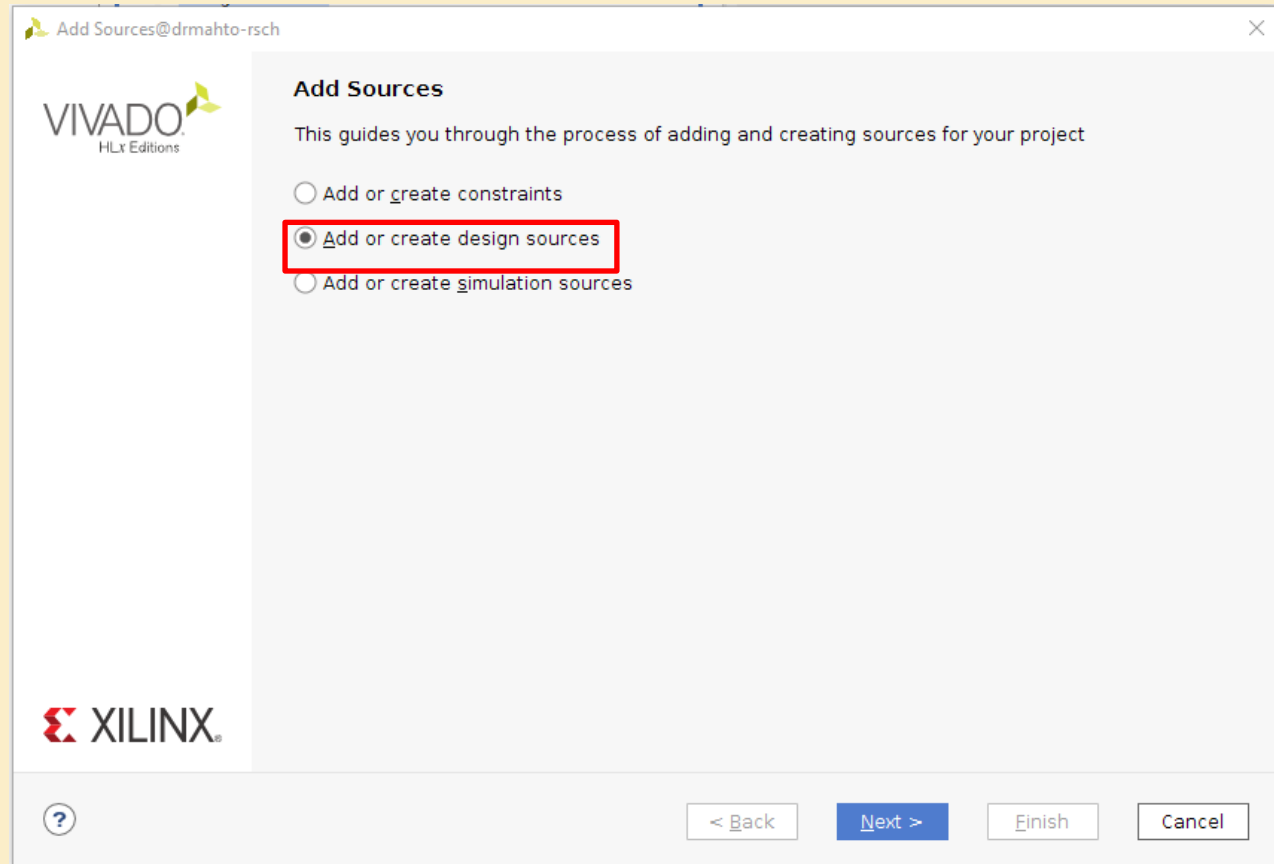
# ADD DESIGN SOURCE

1. Select Design Source
2. Click the “+” sign



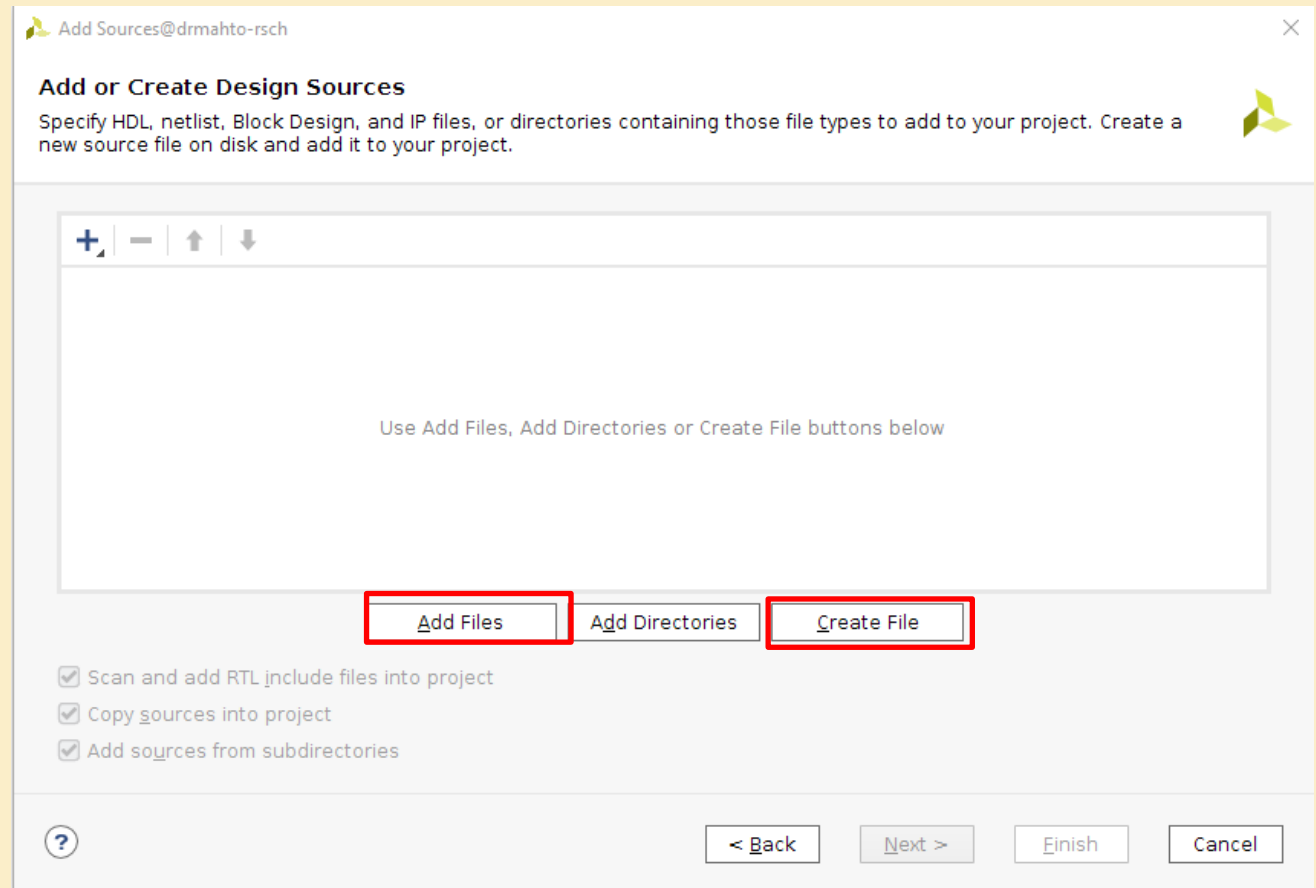
# ADD DESIGN SOURCE

3. Make sure  
Add or create  
design sources



# ADD DESIGN SOURCE

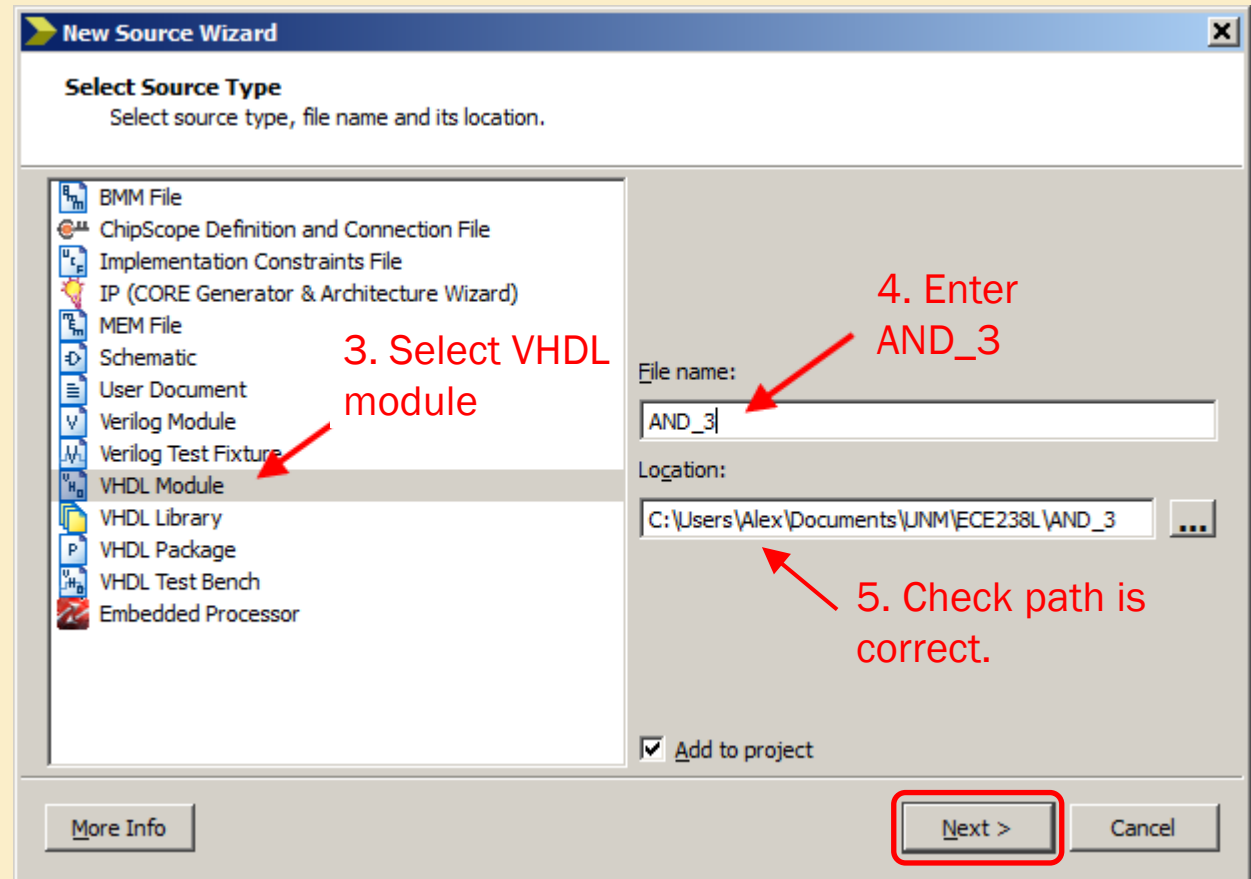
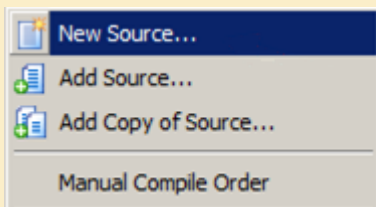
4. Either “Create File” for new file or “Add files” to add file in your project.



# ADD NEW SOURCE

1. Right-click in the Design Panel (Upper left hand side of screen.)

2. Choose New Source from pop-up menu



When done entering data  
Click on Next>

# ENTER SOURCE INFORMATION

1. Enter the entity name

2. Enter the architecture name

3. Enter the input names

4. Enter the output name

5. Update Direction to out for output

Define Module@drmahto-rsch

Define a module and specify I/O Ports to add to your source file.  
For each port specified:  
MSB and LSB values will be ignored unless its Bus column is checked.  
Ports with blank names will not be written.

**Module Definition**

Entity name: And\_3

Architecture name: Behavioral

**I/O Port Definitions**

Port Name	Direction	Bus	MSB	LSB
A, B, C	in	<input type="checkbox"/>	0	0
F	out	<input type="checkbox"/>	0	0

OK Cancel

When done entering data Click on OK

# ADD VHDL CODE FOR 3-BIT AND GATE

```
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 entity AND_3 is
24     Port ( A,B,C : in  STD_LOGIC;
25           F : out  STD_LOGIC);
26 end AND_3;
27
28 architecture Boolean_Function of AND_3 is
29
30 begin
31
32 F <= A and B and C;
33
34 end Boolean_Function;
```

Enter the logic function between the **begin** and **end Boolean\_Function** statements

Debugging tip:

The “assignment” VHDL deals with data flows so you need to be aware that

$F \leq A$


Is not the same as

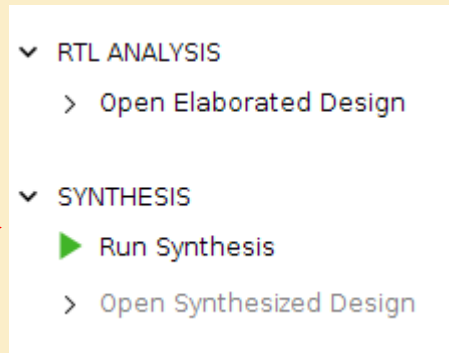
$A \leq F$

The first statement says move values in A into F, where the second statement moves values from F into A



# SYNTHESIZE TO CHECK CODE

Left click on the  symbol next to Synthesize to run a syntax check



Name	Constraints	Status
▼ ✓ synth_1	constrs_1	synth_design Complete!
▶ impl_1	constrs_1	Not started

If the syntax is correct you will get the following message in the bottom console panel,

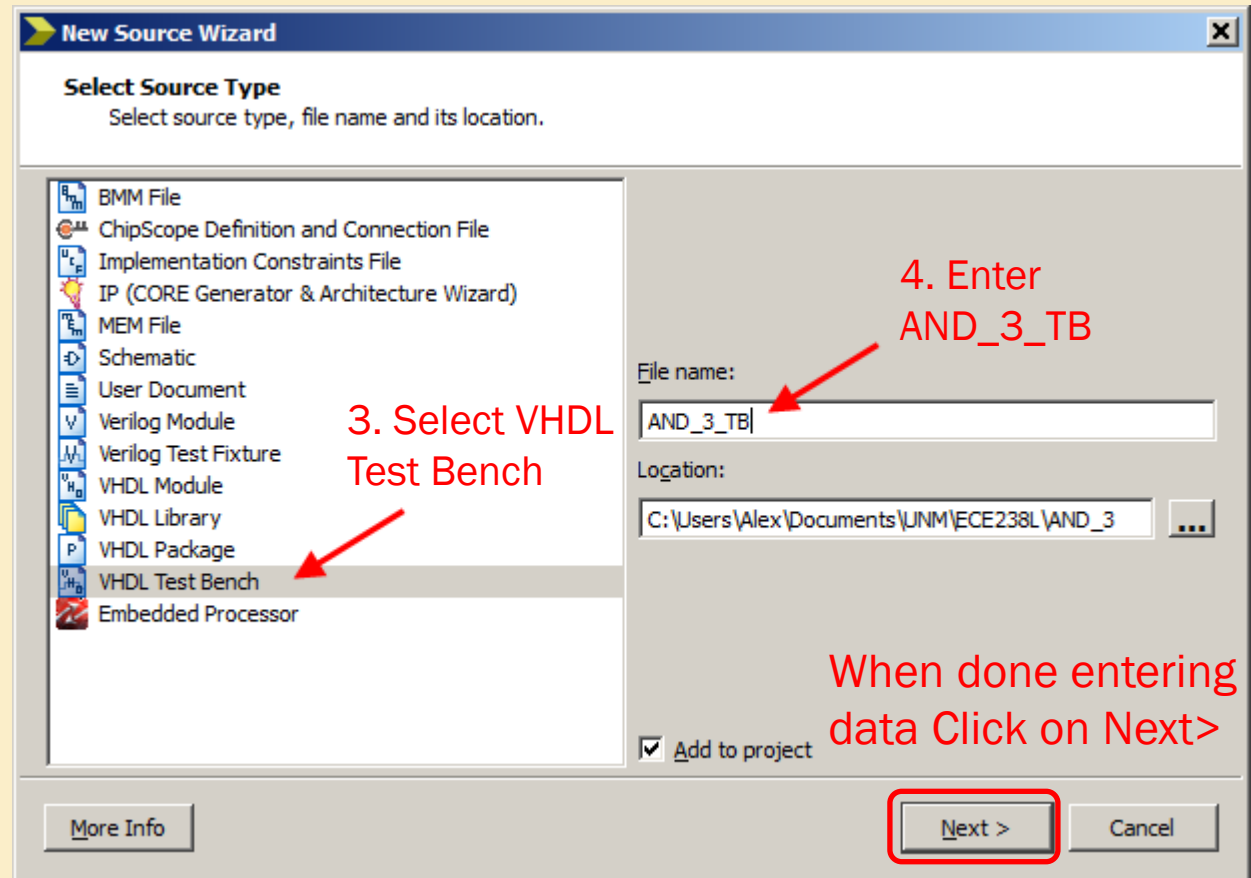
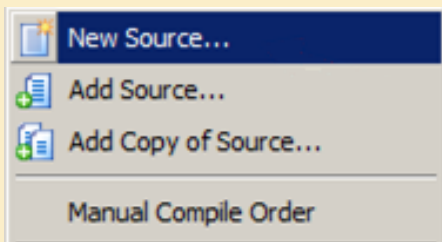
If something failed the check, a message indicating the line number and a “hint” as to what failed.

**Debugging Tip:** Always fix the first item on the list first. Some of the other messages may have been a result of the first lines error,

# CREATE TEST BENCH

1. Right-click in the Design Panel (Upper left hand side of screen.)

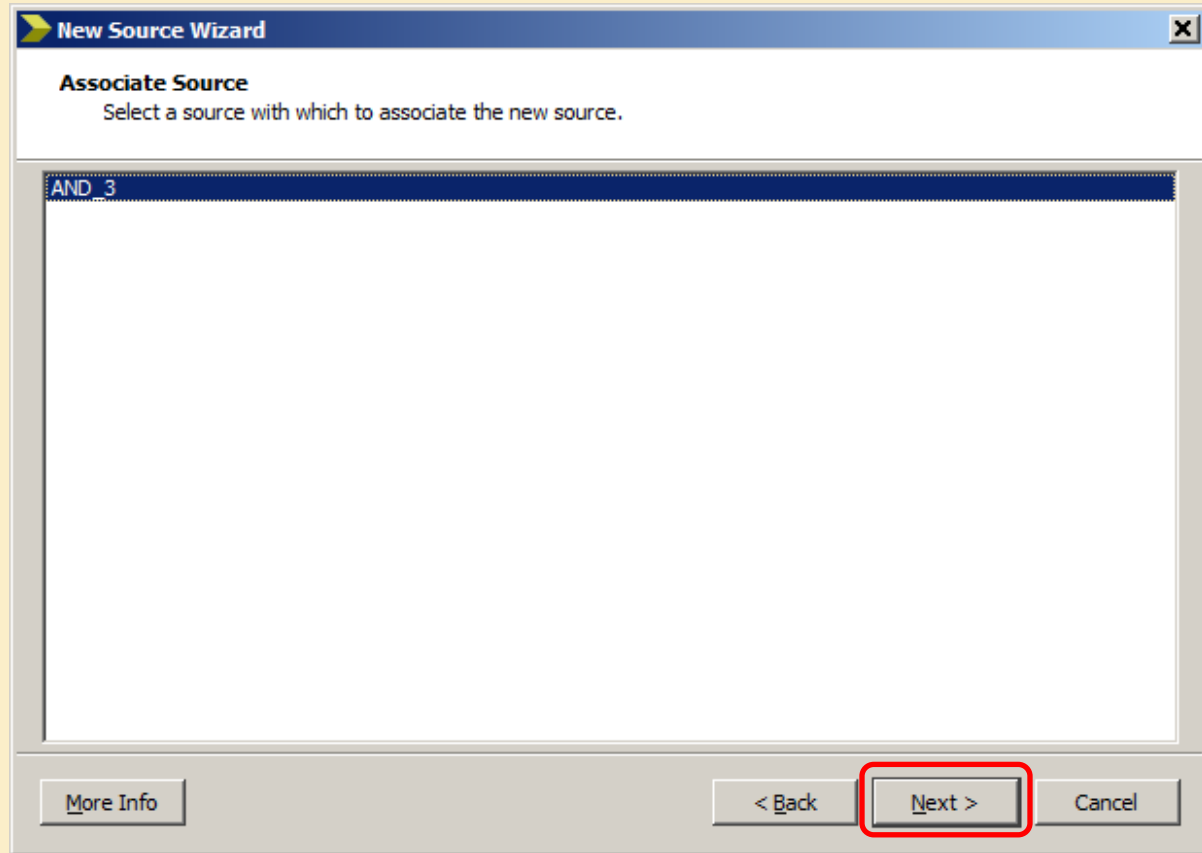
2. Choose New Source from pop-up menu



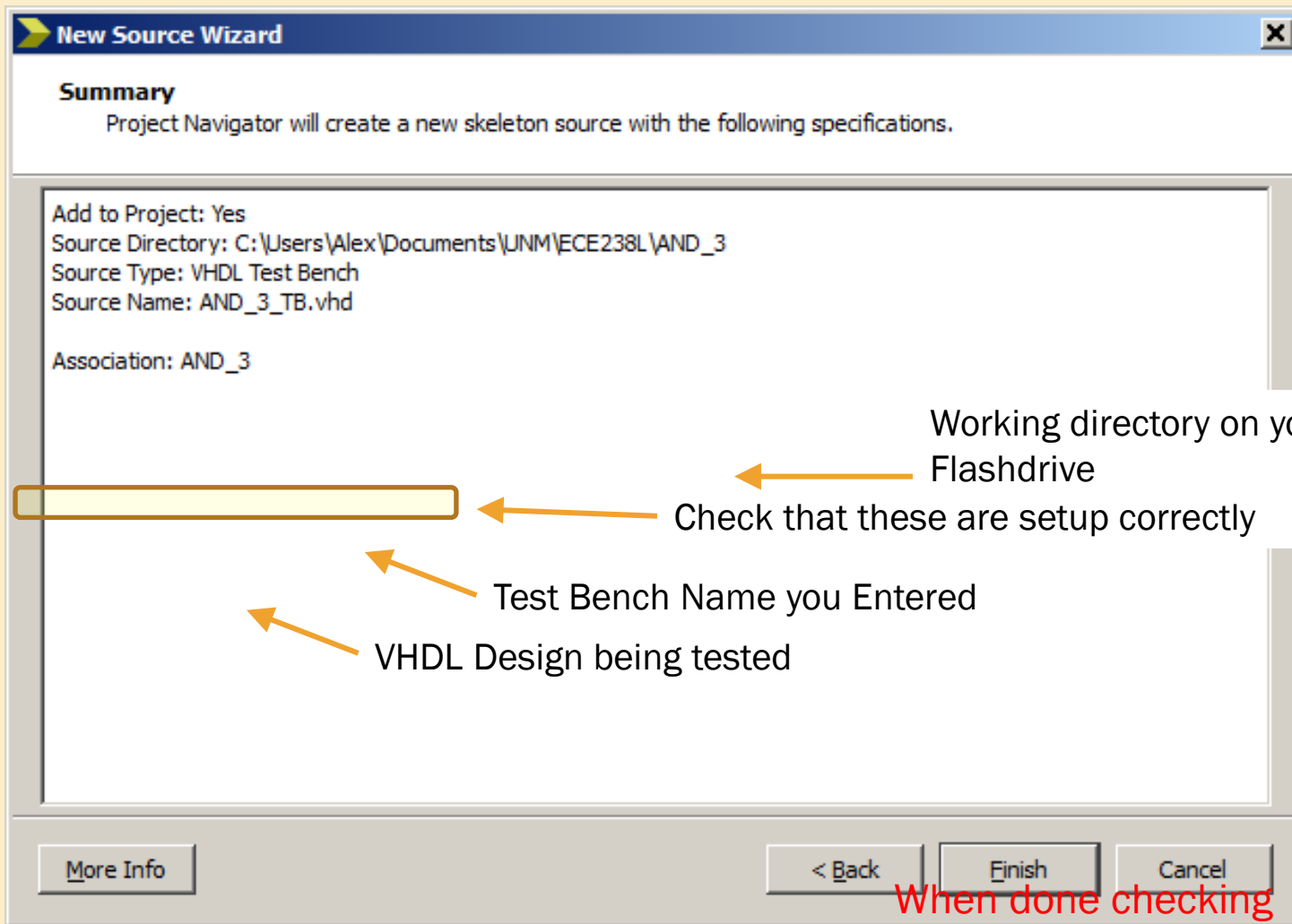
Debugging Tip: Do not name the test bench file exactly the same name as the module. You will over write the module and have to reenter it.

# ASSOCIATE SOURCE FOR TEST BENCH

1. Select the module that the test bench is for.



When done selecting the file Click on Next>



When done checking  
data  
Click on Finish



# COMBINATORIAL LOGIC HAS NO CLOCKS

```
55 -- No clocks detected in port list. Replace <clock> below with
56 -- appropriate port name
57
58 constant <clock>_period : time := 10 ns;
```



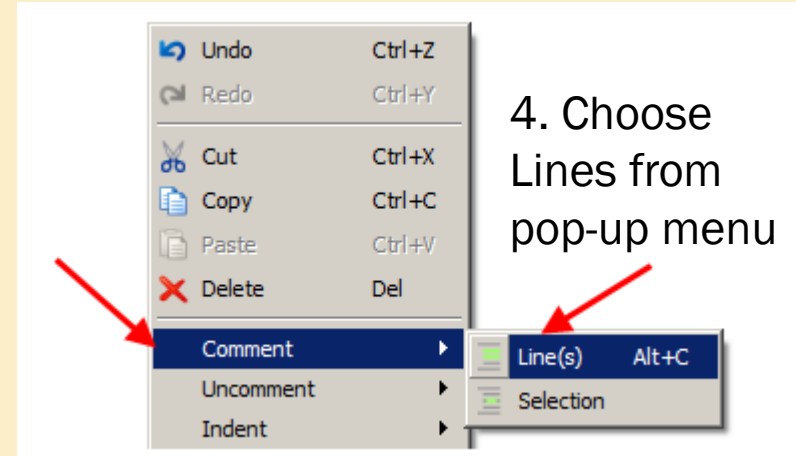
1. Comment out the clock period line (line 58) in the Test Bench File using double dash as shown below

```
55 -- No clocks detected in port list. Replace <clock> below with
56 -- appropriate port name
57
58 -- |constant <clock>_period : time := 10 ns;
```

2. Highlight lines 71 to 77  
and right click

```
68 );
69
70 -- Clock process definitions
71 <clock>_process :process
72 begin
73     <clock> <= '0';
74     wait for <clock>_period/2;
75     <clock> <= '1';
76     wait for <clock>_period/2;
77 end process;
78
79
80 -- Stimulus process
81 stim proc: process
```

3. Choose  
Comment from  
pop-up menu



4. Choose  
Lines from  
pop-up menu

# CREATE THE STIMULUS

```
80  -- Stimulus process
81  stim_proc: process
82  begin
83      -- hold reset state for 100 ns.
84      wait for 100 ns;
85
86      wait for <clock>_period*10;
87
88      -- insert stimulus here
89
90      wait;
91  end process;
```

The test bench file is an automated test of your design. The software will simulate what happens when each line in the truth table is applied to the inputs. So you need to have worked out the truth table in advance.



Replace the text indicated above with the following representation of the truth table.

```
80  -- Stimulus process
81  stim_proc: process
82  begin
83      -- stimulus inputs
84      A <= '0'; B <= '0'; C <= '0';      wait for 100 ns;
85      A <= '0'; B <= '0'; C <= '1';      wait for 100 ns;
86      A <= '0'; B <= '1'; C <= '0';      wait for 100 ns;
87      A <= '0'; B <= '1'; C <= '1';      wait for 100 ns;
88      A <= '1'; B <= '0'; C <= '0';      wait for 100 ns;
89      A <= '1'; B <= '0'; C <= '1';      wait for 100 ns;
90      A <= '1'; B <= '1'; C <= '0';      wait for 100 ns;
91      A <= '1'; B <= '1'; C <= '1';      wait for 100 ns;
92      A <= '0'; B <= '0'; C <= '0';      wait;
93  end process;
```

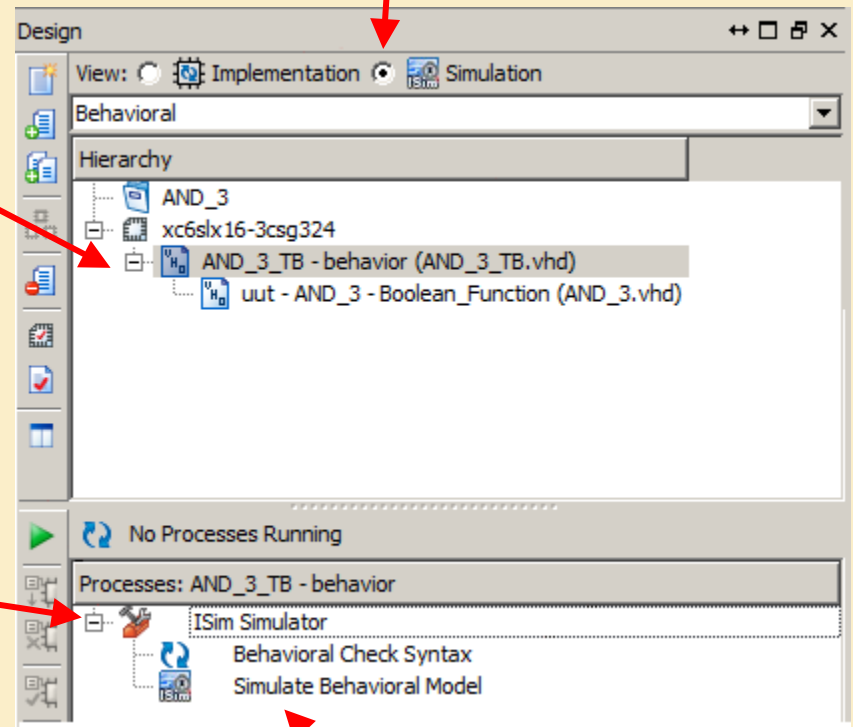
Note the wait times at the end of each line, this is how long the stimulus is active on the input pins of the FPGA chip.

# RUN SIMULATION

1. Select the simulation mode radio button in the upper left hand panel


2. Select test bench file to be used by the simulation.

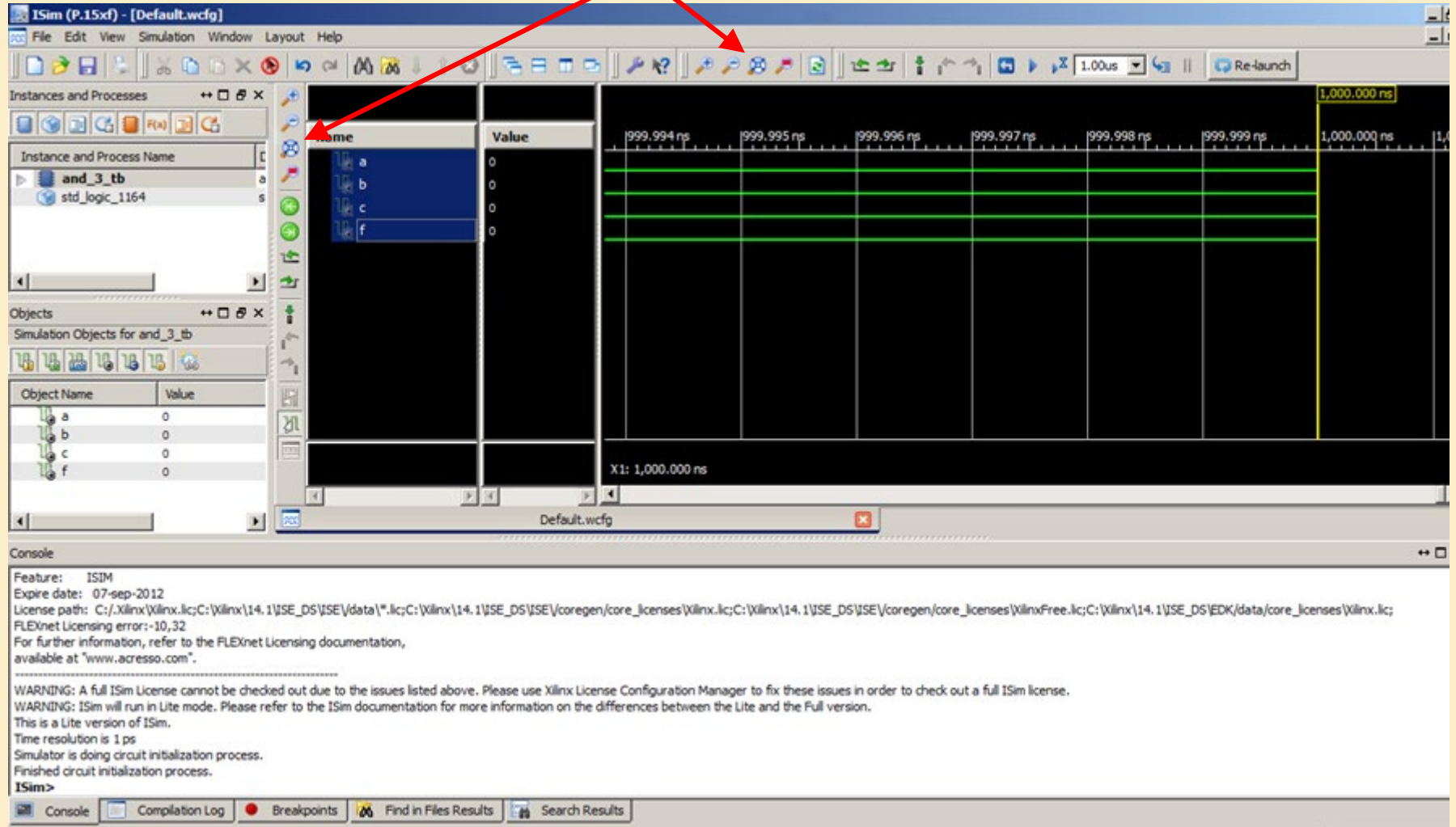
3. Expand the ISM Simulation dropdown menu



4. Select simulate behavioral model

# VIEWING SIMULATION RESULTS

Use the  icon to expand the view so that you can see the wave forms.



**Instances and Processes**

Instance and Process Name	Value
and_3_tb	0
std_logic_1164	0

**Objects**

Object Name	Value
a	0
b	0
c	0
f	0

**Waveform Viewer**

Time: 1,000.000 ns

Console

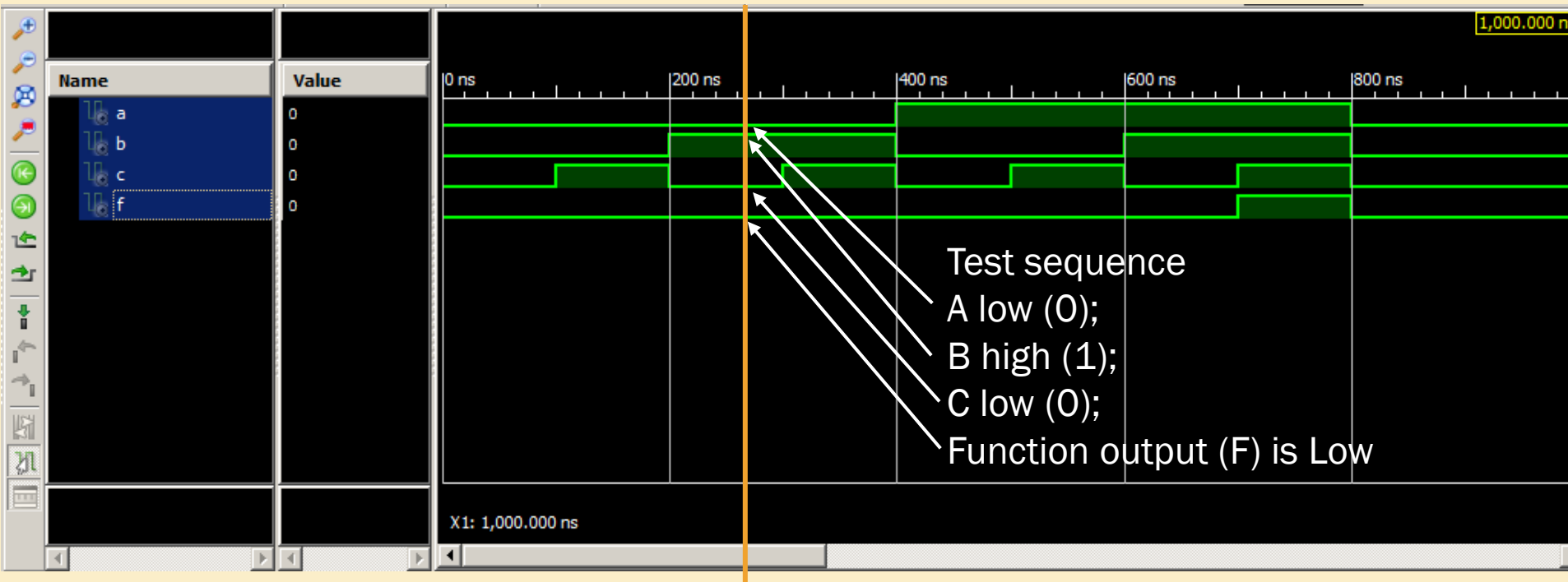
Feature: ISIM  
Expire date: 07-sep-2012  
License path: C:/Xilinx/Xilinx.lc;C:/Xilinx/14.1/ISE\_DS/ISE/data/\*.lc;C:/Xilinx/14.1/ISE\_DS/ISE/coregen/core\_licenses/Xilinx.lc;C:/Xilinx/14.1/ISE\_DS/ISE/coregen/core\_licenses/XilinxFree.lc;C:/Xilinx/14.1/ISE\_DS/EDK/data/core\_licenses/Xilinx.lc;  
FLEXnet Licensing error: -10,32  
For further information, refer to the FLEXnet Licensing documentation,  
available at "www.acresso.com".

WARNING: A full ISim License cannot be checked out due to the issues listed above. Please use Xilinx License Configuration Manager to fix these issues in order to check out a full ISim license.  
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.  
This is a Lite version of ISim.  
Time resolution is 1 ps  
Simulator is doing circuit initialization process.  
Finished circuit initialization process.  
ISim>

Console | Compilation Log | Breakpoints | Find in Files Results | Search Results



# WAVEFORM AFTER EXPANDING VIEW



This is one way of checking whether the design works or not.

Debugging Tip: This is a good tool for spotting timing errors.

**Please Close Simulation Window**

# LAB 1 PROJECT A – NOT, AND, OR

---

1. Create a new project. Name it Lab1A.
  - Write the VHDL that performs the NOT, AND, OR operations on two input values.
  - Input signals: A and B
  - Output signal: F\_NOT, F\_AND, F\_OR.
2. Create the truth table for these three operations.
3. Simulate the VHDL module. Make stimulus inputs count from 00 – 11, waiting 100 ns between each count.
4. Verify that your truth table corresponds with the simulation output.

# TRUTH TABLES FOR NOT, AND, OR GATES

Inputs		Output
	A	F_NOT
1	0	1
2	1	0

Inputs			Output
	A	B	F_AND
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

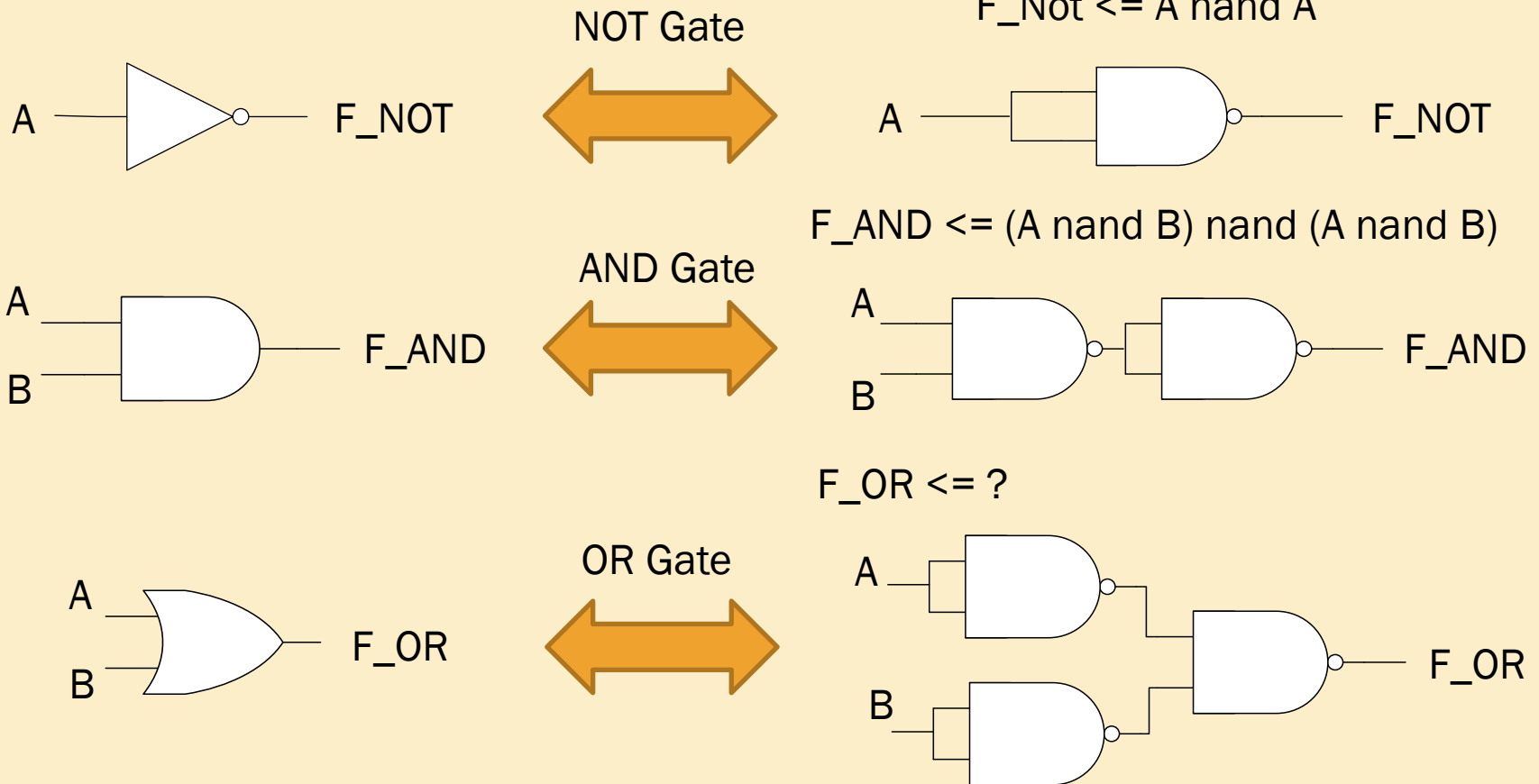
Inputs			Output
	A	B	F_OR
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

# LAB 1 PROJECT B – NAND, NOR, XOR, XNOR, BUFFER

---

1. Create a new project. Name it Lab1B.
  - Write the VHDL that performs the NAND, NOR, XOR, XNOR and Buffer operations on input values.
  - Input signals: A and B
  - Output signal: F\_NAND, F\_NOR, F\_XOR, F\_XNOR, F\_BUFF.
2. Create the truth table for these five operations.
3. Simulate the VHDL module. Make stimulus inputs count from 00 – 11, waiting 100 ns between each count.
4. Verify that your truth table corresponds with the simulation output.

# DEMORGAN LOGIC GATES



# LAB REPORT

---

- ✗ Cover page:
  - + Course Title
  - + Lab Number, Letter
  - + Team Names
- ✗ Project 1A:
  - + VHDL Module
  - + Schematic
  - + Truth Table
  - + Simulation Waveform
- ✗ Project 1B:
  - + VHDL Module
  - + Schematic
  - + Truth Table
  - + Simulation
- ✗ Summary paragraph
  - + Work completed
  - + Any problems
  - + Helpful Hints
  - + Suggested Improvements