

# Xilinx™ Schematic Entry Tutorial

Xilinx™ ISE Schematic Entry  
&  
Modelsim Simulation

By  
Bilal Zafar and Gandhi Puvvada

Rev: 2.0  
September 1, 2008

# Overview

- What is circuit simulation and why is it important?
  - Complex designs, short design cycle
  - Simultaneous system design and hardware design
- Design Entry Options
  - Schematics
  - HDL
- Types of Simulation
  - Behavioral Simulation
  - Timing Simulation (post place & route)
- Tool or Design Flow that we will use
  - Create schematics using Xilinx Schematic Editor
  - Run simulations using a Verilog Test Fixture in Modelsim Simulator
- Implement and download to Nexys2
  - TOP design to connect to I/O resources on Nexys2 board
  - Adept tool to download

# Outline

- Introduction to Xilinx ISE Project Navigator
- Introduction to Xilinx ISE Schematics Entry Tool
- Simulation Using Verilog Test Fixture in ModelSim XE
- Creating Hierarchical Designs
- Creating TOP Design to interface to Nexys2 board, implementing the design using ISE, and downloading the bit file using Adept

# Introduction to Xilinx ISE Project Navigator

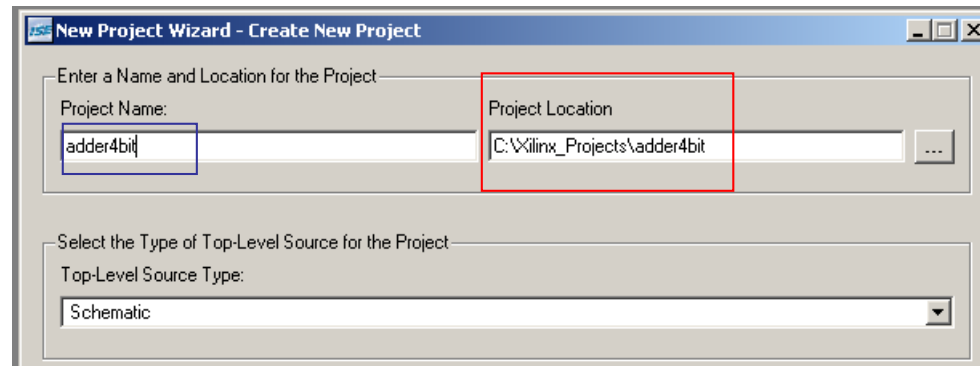
- Project Navigator: an integrated environment
  - create a project with many design files, etc.
  - enter our design (schematics and Verilog)
  - write test bench for the design
  - launch ModelSim XE simulator to run simulations
  - synthesize the design for FPGA board
  - program the FPGA
  - debug the FPGA-based design using ChipScope

# Setting up the Xilinx Tools

- Make sure you have installed and tested the latest versions of:  
Refer to the installation and testing procedure documents posted on the Blackboard.
  - Xilinx ISE Webpack (version 10.1i in Aug. 2008)
  - ModelSim Xilinx Edition (MXE) (version 6.3c in Aug. 2008)
  - Xilinx ChipScope (version 10.1i in Aug. 2008)
  
- Create a Xilinx Projects folder
  - Create a new folder in **C:\xilinx\_projects** for all your Xilinx Projects
  - We will refer to this folder as the Projects folder

# Creating a New Project

- Launch Xilinx Project Navigator
- Click File → New Project to start the new project wizard
- Enter the name, location and top-level source type for the project
  - Project Name: **adder4bit**
  - Project Location: C:\xilinx\_projects\**adder4bit**
  - Top-level Source Type: Schematics



- Click Next
- This creates a new sub-folder (**adder4bit**) where all your design files related to this project (plus many auxiliary files that Xilinx creates) are stored. The main project file necessary to open the project again at a later point of time is also stored here with the name **adder4bit.isc**.

# Expand the pull-down choices to get to know

New Project Wizard - Device Properties

Select the device and design flow for the project

| Property Name                  | Value                               |
|--------------------------------|-------------------------------------|
| Product Category               | All                                 |
| Family                         | Spartan3E                           |
| Device                         | XC3S500E                            |
| Package                        | FG320                               |
| Speed                          | -4                                  |
| Top-Level Source Type          | Schematic                           |
| Synthesis Tool                 | XST (VHDL/Verilog)                  |
| Simulator                      | Modelsim-XE Verilog                 |
| Preferred Language             | Verilog                             |
| Enable Enhanced Design Summary | <input checked="" type="checkbox"/> |
| Enable Message Filtering       | <input type="checkbox"/>            |
| Display Incremental Messages   | <input type="checkbox"/>            |

More Info < Back Next > Cancel

**FPGA on the Nexys2 board**

**Schematic/HDL**

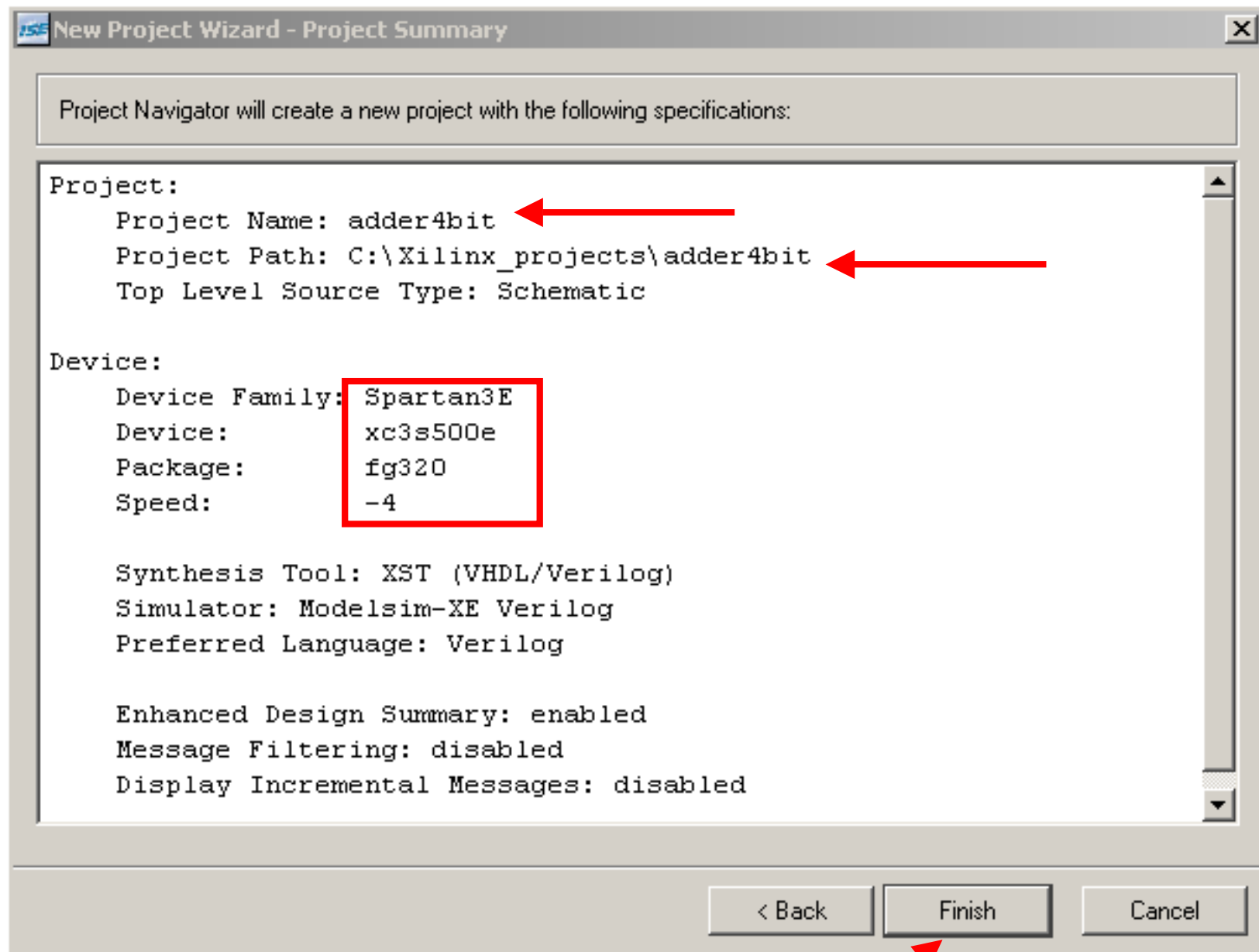
# Choosing Device Properties

- In “Device Properties” dialog box select the following values (see the previous slide)
- Click Next
- Click Next for the Create New Source dialogue box
- Click Next for the Add Existing Source dialogue box

|                       |                    |
|-----------------------|--------------------|
| Product Category      | All                |
| Family                | Spartan3E          |
| Device                | XC3S500E           |
| Package               | FG320              |
| Speed                 | -4                 |
| Top-Level Source Type | Schematic          |
| Synthesis Tool        | XST (VHDL/Verilog) |
| Simulator             | ModelsimXE Verilog |
| Preferred Language    | Verilog            |

**Explain what each of these values means**

# New Project Wizard – Project Summary



Click Finish

# Getting to Know the Project Navigator

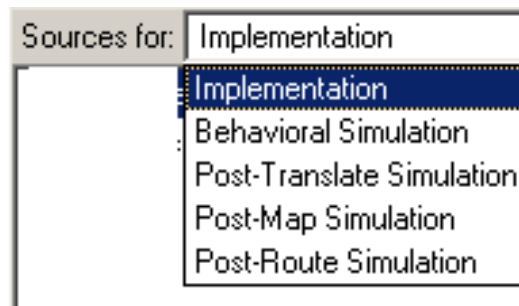
## 1. Toolbar

- Basic file management tools
- Process specific tools for the selected process

## 2. Sources Window

- Shows the design hierarchy with all source files
- Different views available (see 3)

## 3. Sources View (in the Sources Window)



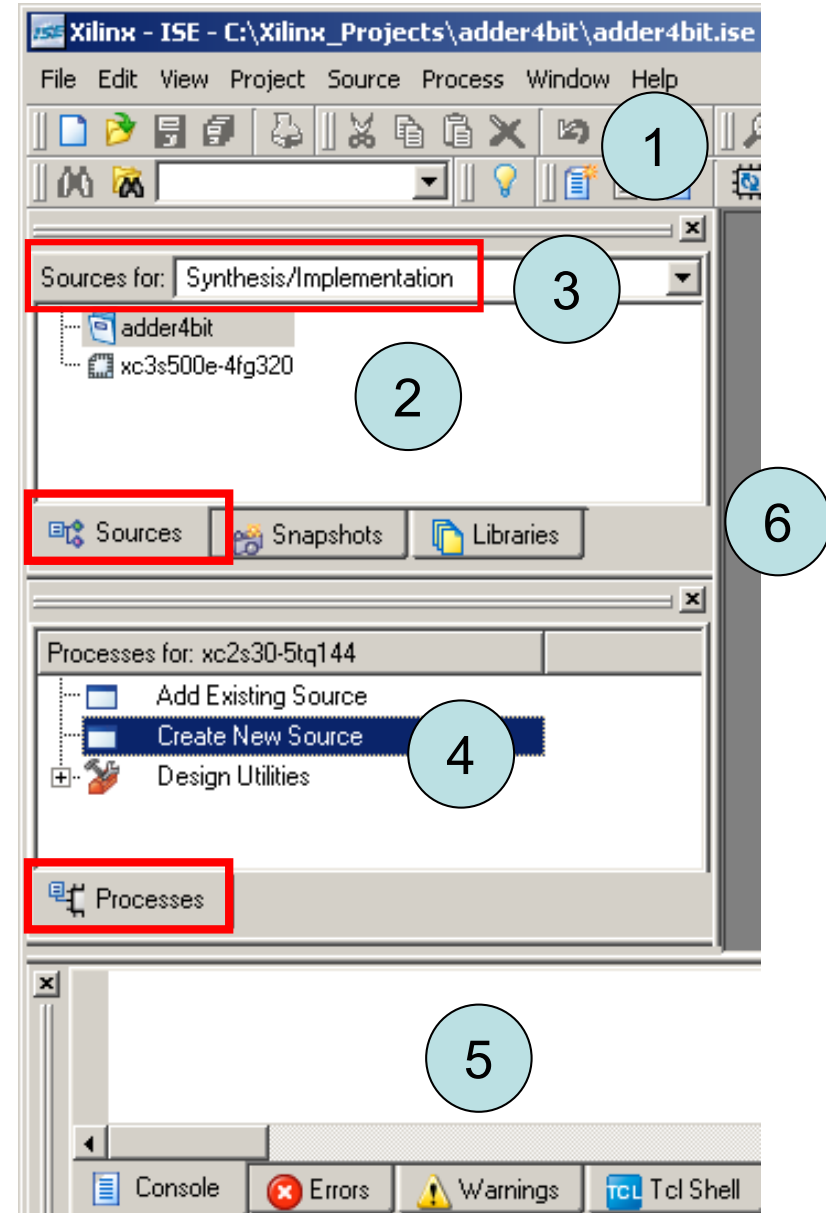
## 4. Processes Window

- Show available processes for the selected source

## 5. Transcript window:

- Space where warnings and error messages are reported

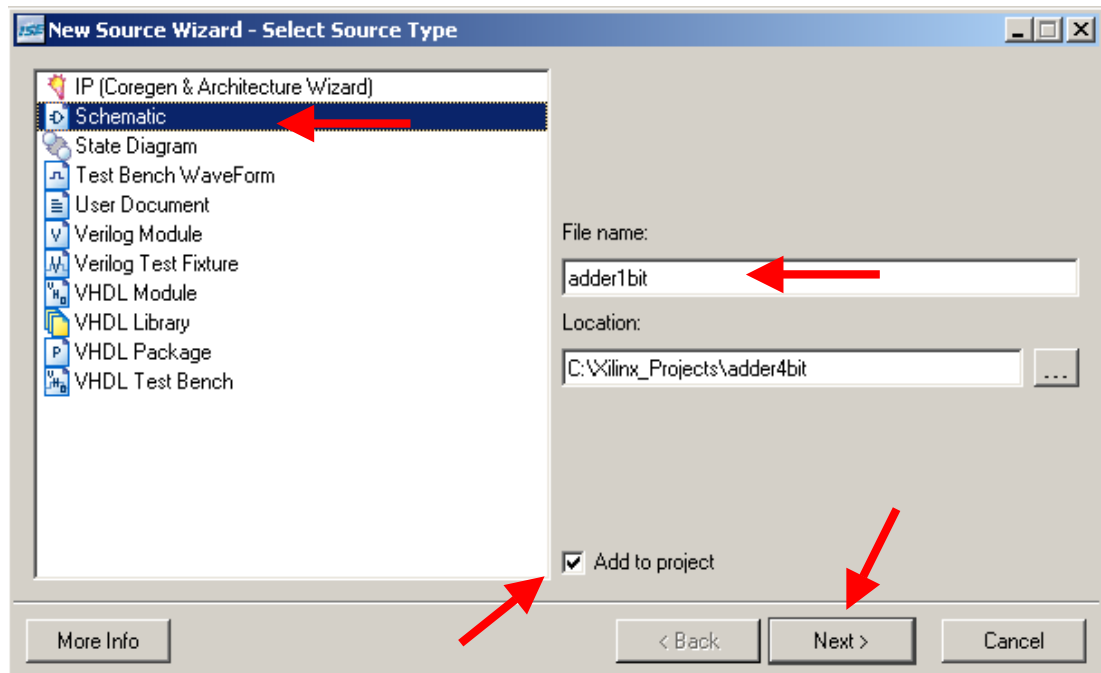
## 6. Workspace: the playing field!



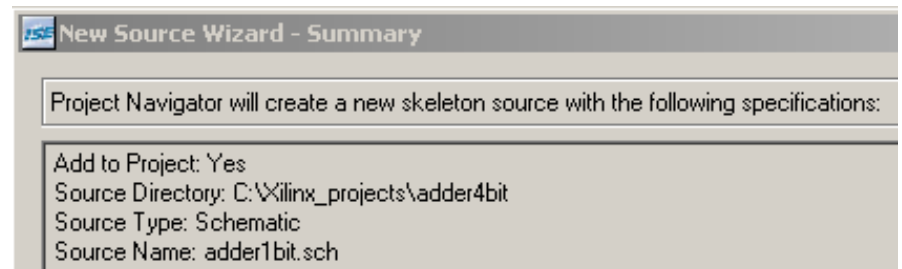
# Create a New Source File

- Double click Create New Source in the Processes window
- In the New Source Wizard dialogue box
  - select the type of source file: Schematics
  - enter the file name: adder1bit

- Click Next



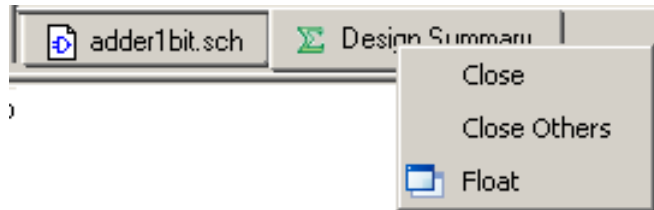
- Then click Finish in the Summary window



# Enhanced Design Summary

Close the Design Summary tab

- Enhanced Design Summary appears



- Switch to the adder1bit.sch tab to bring the a blank schematic sheet to the foreground

**add4bit Project Status**

|                         |                           |                            |     |
|-------------------------|---------------------------|----------------------------|-----|
| <b>Project File:</b>    | adder4bit.isc             | <b>Current State:</b>      | New |
| <b>Module Name:</b>     | adder1bit                 | <b>Errors:</b>             |     |
| <b>Target Device:</b>   | xc3s500e-4fg320           | <b>Warnings:</b>           |     |
| <b>Product Version:</b> | ISE 10.1.02 - WebPACK     | <b>Routing Results:</b>    |     |
| <b>Design Goal:</b>     | Balanced                  | <b>Timing Constraints:</b> |     |
| <b>Design Strategy:</b> | Xilinx Default (unlocked) | <b>Final Timing Score:</b> |     |

**add4bit Partition Summary**

No partition information was found.

**Detailed Reports**

| Report Name        | Status | Generated | Errors | Warnings | Info |
|--------------------|--------|-----------|--------|----------|------|
| Synthesis Report   |        |           |        |          |      |
| Translation Report |        |           |        |          |      |

# Getting to Know the Xilinx Schematic Editor

## 1. Toolbar

- new buttons for schematic entry

## 2. Symbols Tab

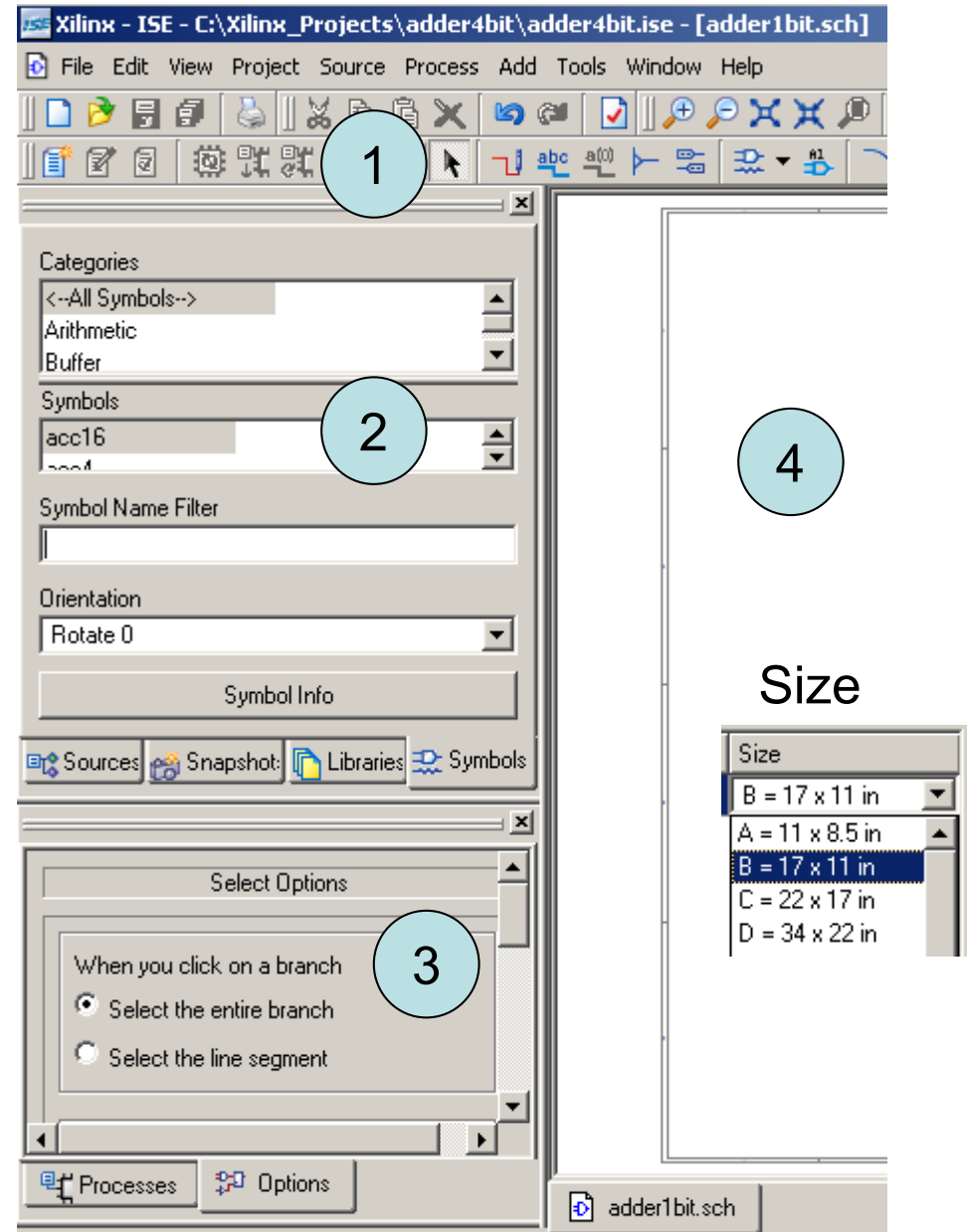
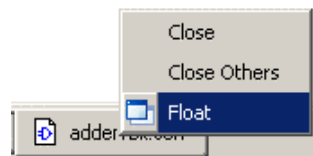
- Categories
- Symbols
- Symbol Name Filter
- Orientation
- Symbol Info (links to Xilinx Libraries Guide to show component's data sheet)

## 3. Options Tab

- Shows options related to the selected task

## 4. Schematic Editor Workspace

- You can detach the Schematic from the Project Navigator by right-clicking at the schematic file name tab (adder1bit.sch) and selecting Float

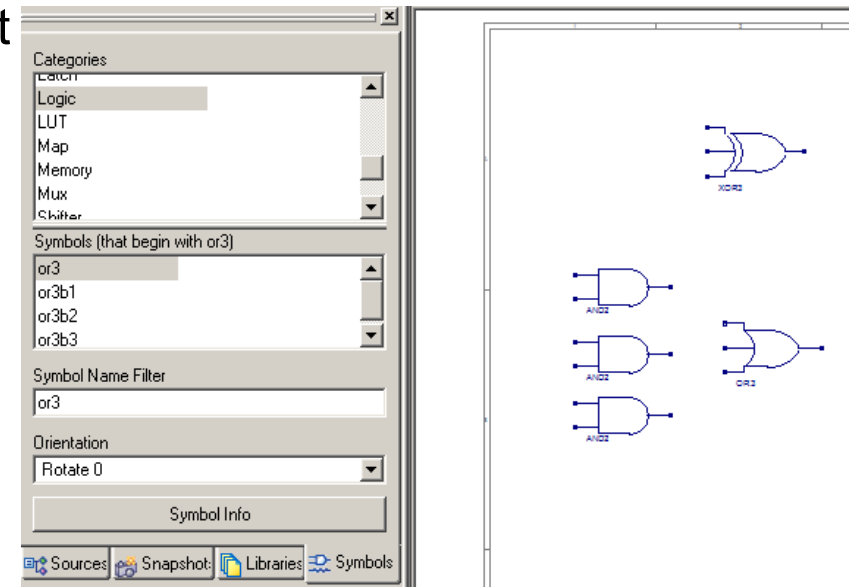
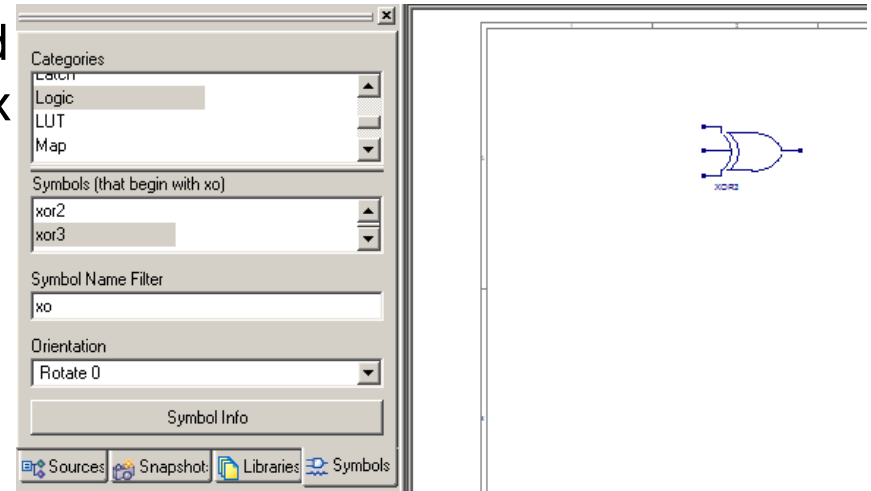


# Outline

- Introduction to Xilinx ISE Project Navigator
- **Introduction to Xilinx ISE Schematics Entry Tool**
- Simulation Using Verilog Test Fixture in ModelSim XE
- Creating Hierarchical Designs
- Creating TOP Design to interface to Nexys2 board, implementing the design using ISE, and downloading the bit file using Adept

# Inserting Logic Gates

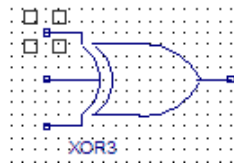
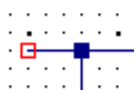
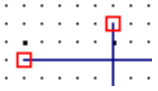
- In the Categories pane, scroll down and select Logic. Logic gates from the Xilinx Spartan 3 library appear in the Symbols pane.
  - OR, leave All Symbols selected in the Categories pane and type xor3 in the Symbol Name Filter
- Select xor3 in the Symbol pane
- Move the mouse over the schematic sheet and left click to drop the gate
- You can drop multiple copies of the gate (we don't need it here)
- Drop three and2 gates and one or3 gate, as shown
- Change the orientation of the gate using the Orientation drop-down menu

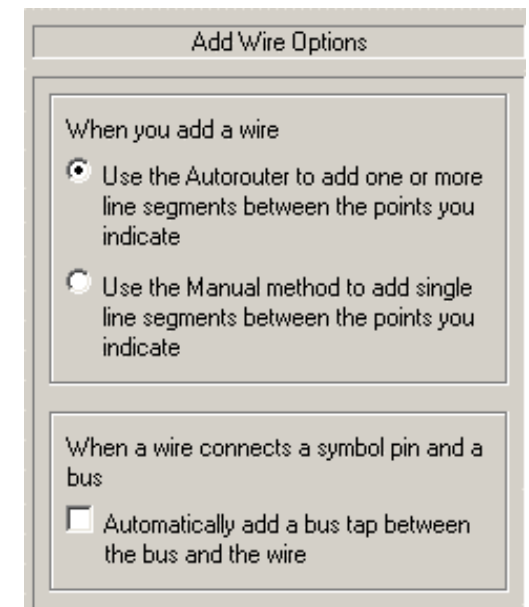


# Schematic Editing Toolbar

|                                                                                     |                                                                                                         |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
|    | Select tool (default)                                                                                   |
|    | Add wire                                                                                                |
|    | Add Net Name                                                                                            |
|    | Rename Selected Bus                                                                                     |
|    | Add bus tap                                                                                             |
|    | Add Input/Output marker                                                                                 |
|    | Add Symbol                                                                                              |
|    | Add Instance Name                                                                                       |
|    | Check Schematic                                                                                         |
|   | Add drawing & text  |
|  | Push into a symbol                                                                                      |
|  | Pop to the next higher level                                                                            |
|  | Zoom controls                                                                                           |
|  | Zoom back and forth                                                                                     |

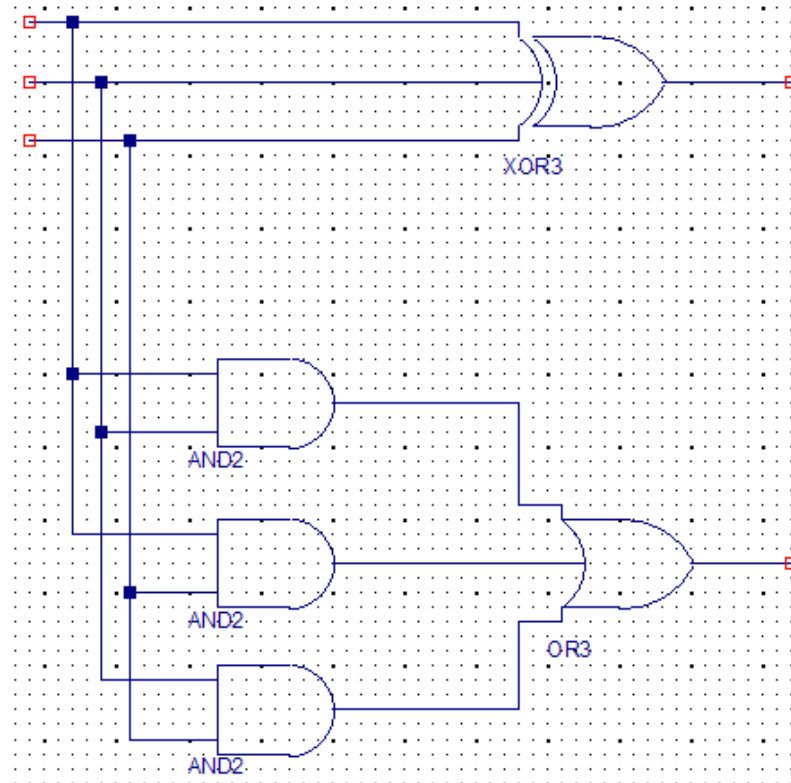
# Adding Wires and Buses

- To add a net (wire or bus)
  - Click on the Add wire toolbar button 
  - Or, press Ctrl+w
- To connect a net to a pin, bring the mouse pointer close to the pin
- The four square boxes next to the pin  indicate that the added net will snap to this pin
- Use “press-drag-release” to draw a net
- To create branching, place the pointer’s centre on the net and press-drag-release
- When connecting two nets, a solid blue dot at the intersection indicates that nets are connected
  -  = connected;  = not connected
- Add Wire Options appear under the Options tab



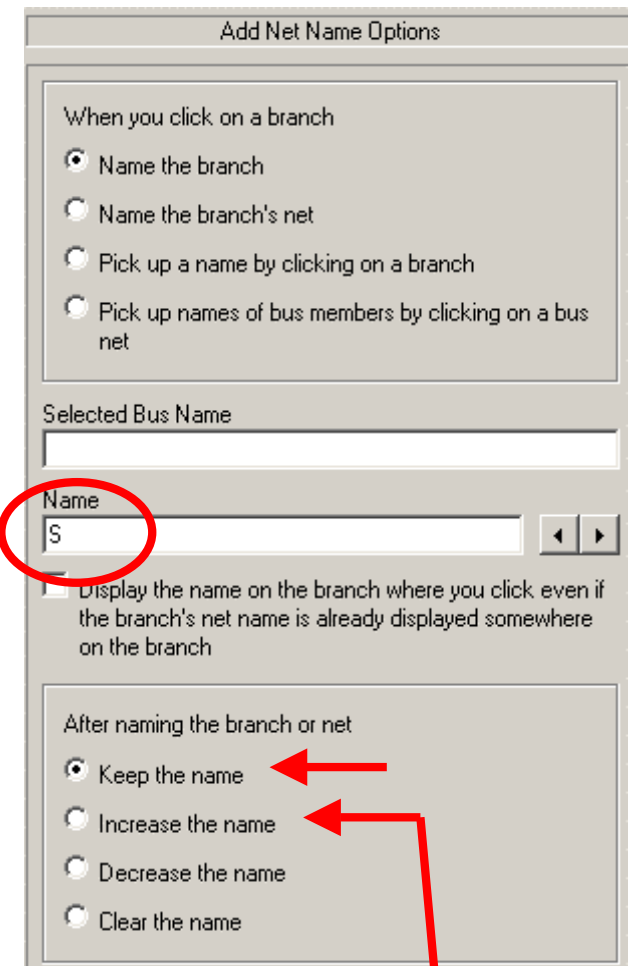
# Complete Wiring of adder1bit

- Complete the wiring as shown here



# Naming Wires and Buses

- To name a net (wire or bus)
  - click on the Add Net Name toolbar button 
  - or, press Ctrl+d
- Add Net Name Options appear in the Options tab; Type the net name under **Name**
- Centre the pointer (crosshair) at the net to which you want the name assigned and click. Net name will appear on the net
- To view a net's properties (including name):
  - select the net
  - right click and select **Object Properties** or press **Alt+Enter** or click on  (Object Properties Button)
  - net name appears in the **Object Properties** dialogue box



Add Net Name Options

When you click on a branch

- ☒ Name the branch
- ☐ Name the branch's net
- ☐ Pick up a name by clicking on a branch
- ☐ Pick up names of bus members by clicking on a bus net

Selected Bus Name

Name

S

☐ Display the name on the branch where you click even if the branch's net name is already displayed somewhere on the branch

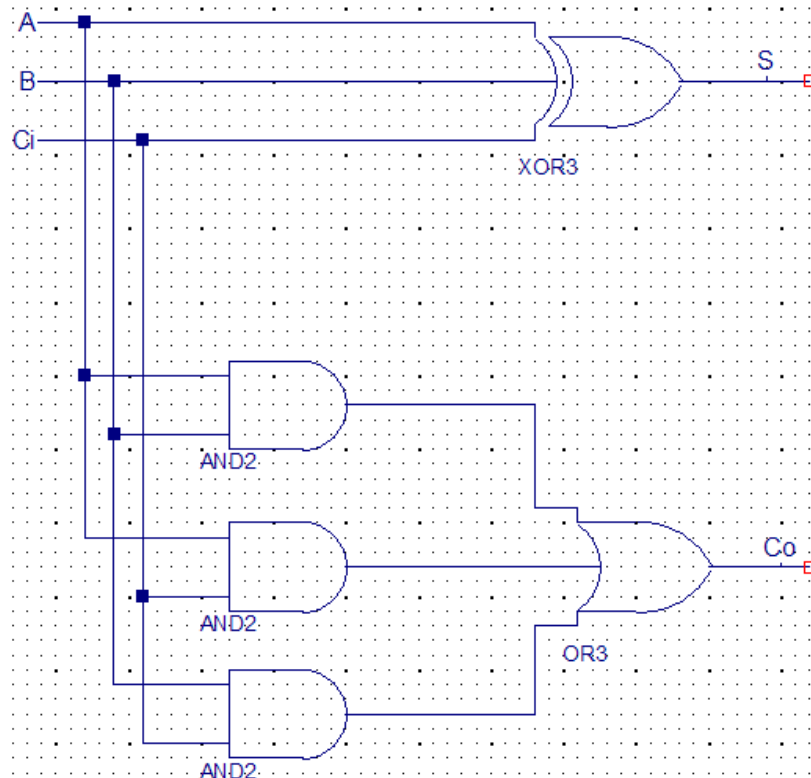
After naming the branch or net

- ☒ Keep the name
- ☐ Increase the name
- ☐ Decrease the name
- ☐ Clear the name

Useful in naming a series of wires with A(0), A(1), A(2), ..

# Complete the net naming of adder1bit

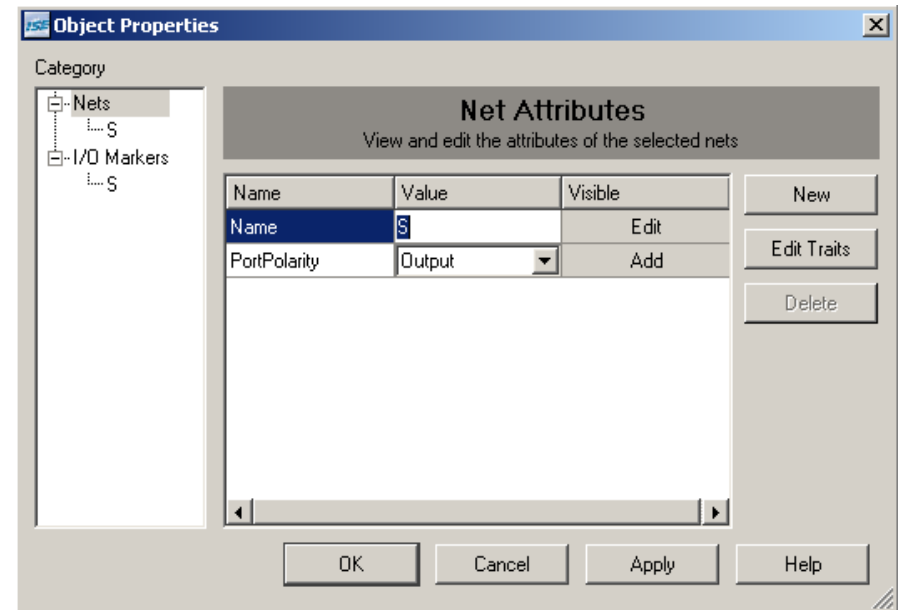
- Name the nets as shown here



- Note that deleting the label (text) from the schematics DOES NOT delete the net name. Delete net name by selecting the net and going to Object Properties

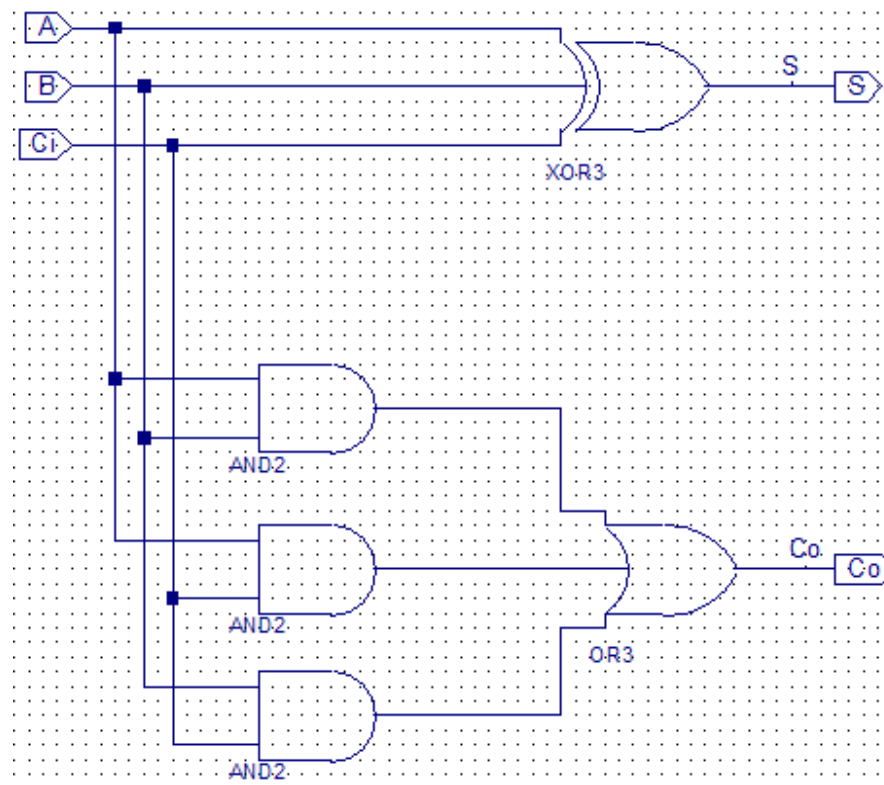
# Adding Input/Output Markers

- I/O marker is attached to a net to:
  - Identify the net as having an input, output or bidirectional signal; that is, establish net polarity (direction of signal flow)
  - Show that the net is externally accessible.
- All primary inputs and outputs must be marked with I/O markers
- To add an I/O marker to a net:
  - Click on the Add I/O Marker toolbar button 
  - Click on the end of a net
- I/O markers of named nets automatically take the net's name
- Xilinx Schematic Editor smartly identifies polarity of nets. To check (and change) the polarity:
  - Select the I/O marker
  - Right click and select **Object Properties**; or, press Alt+Enter, or 



# Adding I/O markers to the adder1bit design

- Add I/O markers as shown here



The schematics of the **adder1bit** design is now complete!

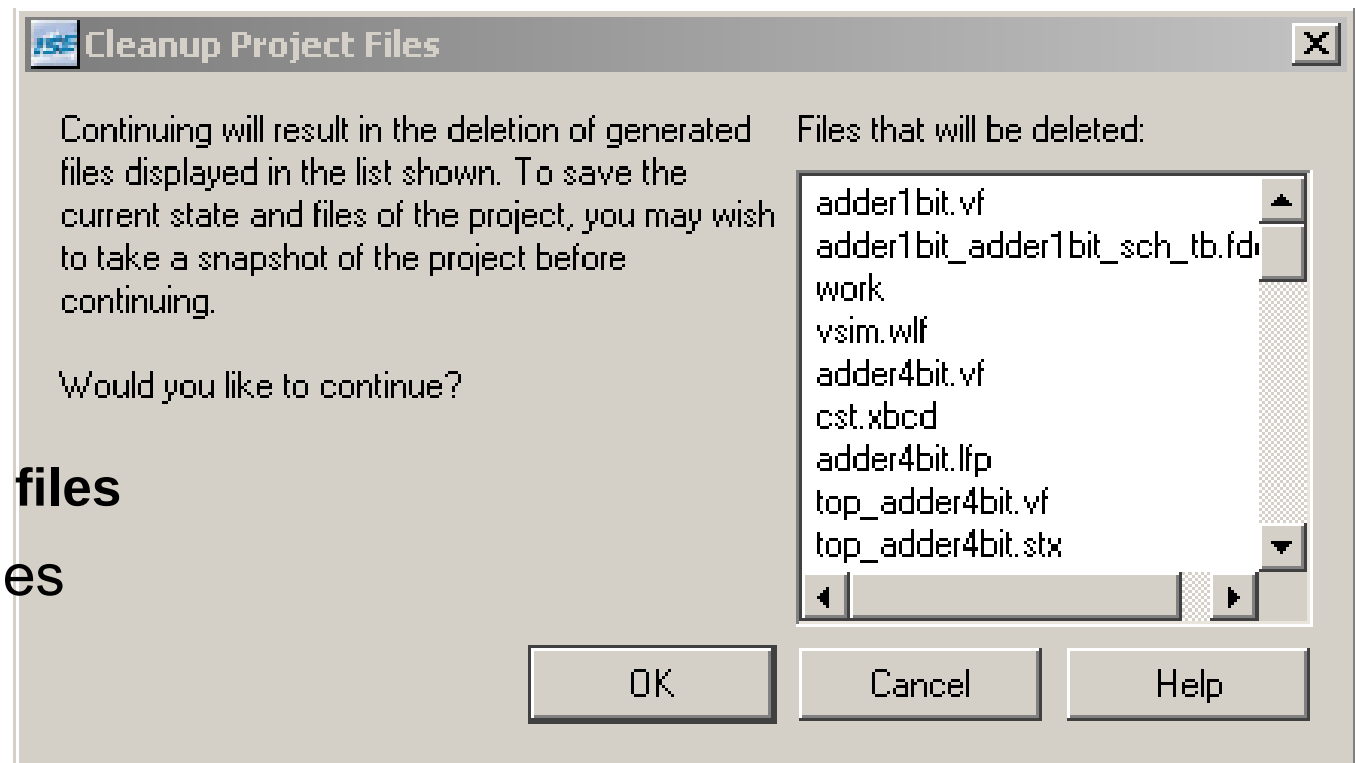
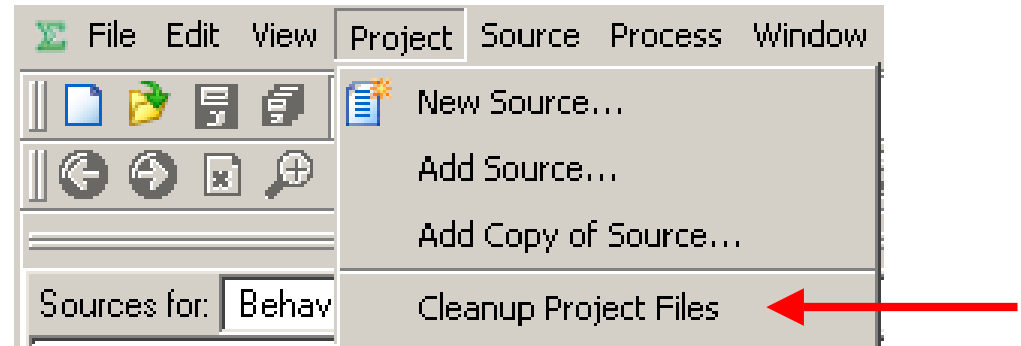
- SAVE the file and close the Schematic Editor. Click on the Check Schematic toolbar button  to make sure there are no errors.



# Project Cleanup

## Source files

- Schematic files
- Symbol files
- Testbenches
- Verilog files
- UCF file(s)



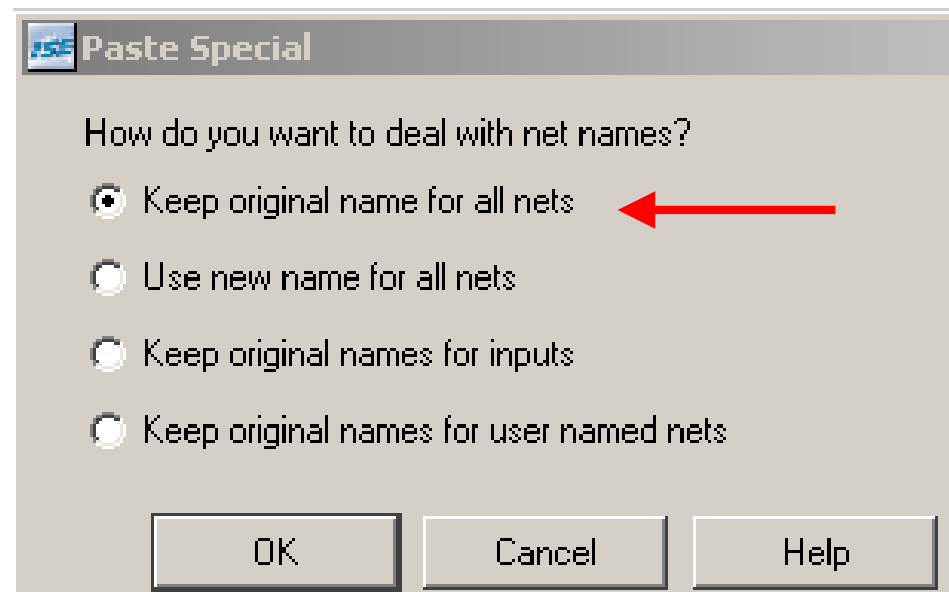
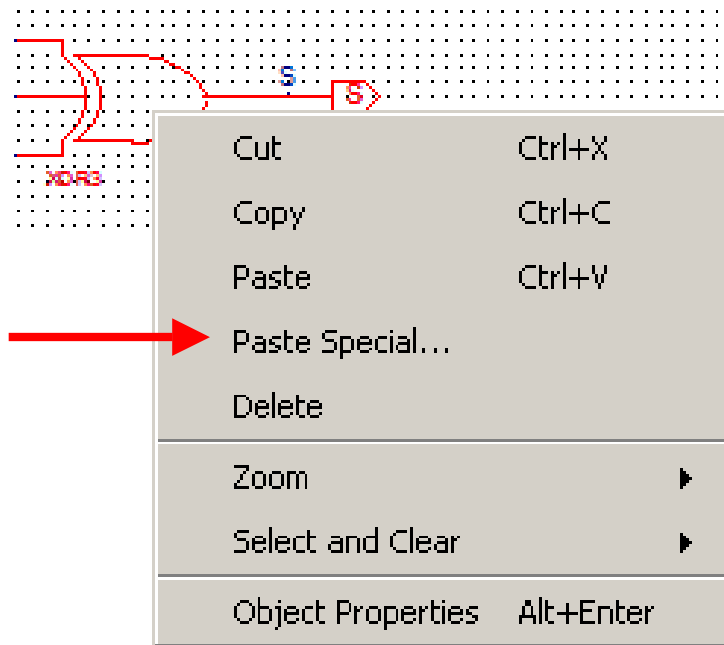
## Tool generated (derived) files

- Implementation files
- Report files
- .bit file

**It is a good idea to perform project cleanup periodically, especially when things are not working the way you expect them to work!**

# Copy and "Paste Special"

If you want to retain the net names and marker names when you copy and paste, then choose Edit => Paste Special (after copying the desired portion of the schematic).



# "Rename the Branch's Net" VS. "Rename the Branch"

If you named some wire as S and then you later realize that it should have been named S1.

If you select that piece of wire (called Branch in Xilinx terminology), right-click, and select "Rename Net", then you have two choices:

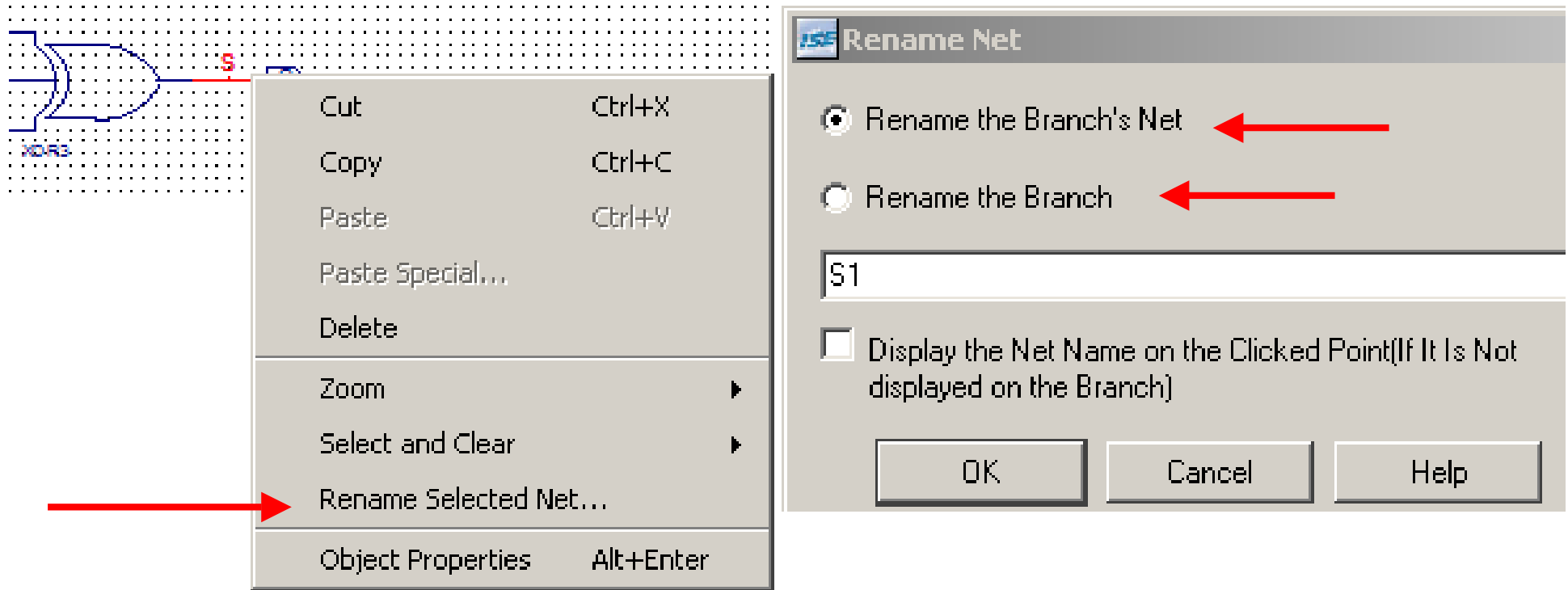
- Rename the Branch's Net

- Rename the Branch

If you select, "Rename the Branch's Net" you are asking all wires in the entire (multi-sheet) schematic, which were previously named as S to be renamed as S1.

If that is not your intent, and if you just wanted to correct that single piece (called Branch), then you should select, "Rename the Branch".

# "Rename the Branch's Net" vs. "Rename the Branch"

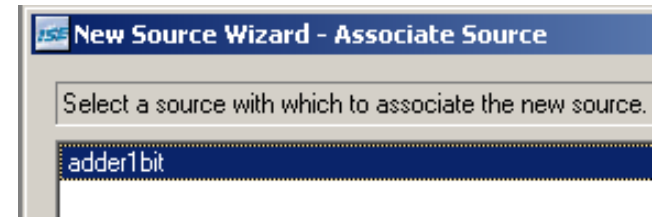


# Outline

- Introduction to Xilinx ISE Project Navigator
- Introduction to Xilinx ISE Schematics Entry Tool
- **Simulation Using Verilog Test Fixture in ModelSim XE**
- Creating Hierarchical Designs
- Creating TOP Design to interface to Nexys2 board, implementing the design using ISE, and downloading the bit file using Adept

# Simulating adder1bit

- We will simulate our schematic design using a Verilog Test Fixture and the Modelsim Simulator
  - Only nets with I/O marker attached as “Input” can be driven
  - ALL nets can be observed
  - Project Navigator automatically creates a Test Fixture; we only need to add test patterns to code
- To create a Verilog Test Fixture & simulate your design:
  - In the Processes window, double click Create New Source
  - Select Source Type: **Verilog Test Fixture**; File name: **adder1bit\_tb**
  - Click Next
  - Associate **adder1bit** to this test fixture
  - Click Next & then Finish



- A new Verilog Test Fixture source file appears in the workspace

# Auto-generated Verilog Test Fixture

- ``timescale 1ns / 1ps`
  - time unit is 1 nano second
  - resolution is 1 pico second
- Module name
- Input/Output declarations
  - Test fixture code will drive inputs
- Instantiating UUT
  - UUT = unit under test = adder1bit
  - format `.port_name(signal_name)`
  - port\_name is the name of the i/o port of the design under test
  - signal\_name is the name of the test fixture signal connected to this port
- Delete lines 27-32
- Do NOT delete line 33: the “endmodule” statement
- Replace lines 26-32 with test patterns given next ...

```
1 // Verilog test fixture created from
2
3 `timescale 1ns / 1ps
4
5 module adder1bit_adder1bit_sch_tb();
6
7 // Inputs
8 reg Ci;
9 reg A;
10 reg B;
11
12 // Output
13 wire S;
14 wire Co;
15
16 // Bidirs
17
18 // Instantiate the UUT
19 adder1bit UUT (
20     .Ci(Ci),
21     .A(A),
22     .B(B),
23     .S(S),
24     .Co(Co)
25 );
26 // Initialize Inputs
27 `ifdef auto_init
28     initial begin
29         Ci = 0;
30         A = 0;
31         B = 0;
32     `endif
33 endmodule
34
```

**Inputs to DUT**

**Outputs from DUT**

**Instantiation of DUT**

**Stimulus of DUT**

**delete**

**keep<sup>e</sup>**

# Adding Test Patterns

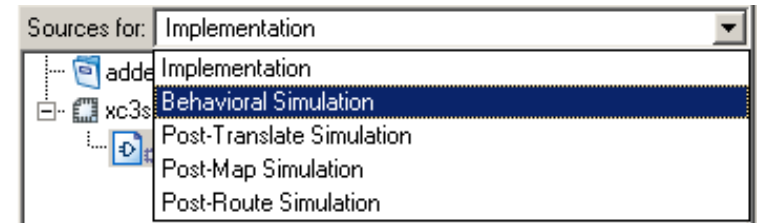
- Code shown here adds 4 test patterns to the test fixture
- Enter this code and complete it by adding the remaining 4 test patterns (total  $2 \times 3 = 8$ )
- #10 => wait for 10 time units => 10ns
- Once complete, SAVE the Verilog Test Fixture

```
26 // Initialize Inputs
27 initial
28     begin
29         Ci = 0;
30         A = 0;
31         B = 0;
32         #10;
33
34         Ci = 1;
35         A = 0;
36         B = 0;
37         #10;
38
39         Ci = 0;
40         A = 1;
41         B = 0;
42         #10;
43
44         Ci = 1;
45         A = 1;
46         B = 0;
47         #10;
48
49         // Add more
50         // test patterns
51         // here.
52     end
53
54 endmodule
```

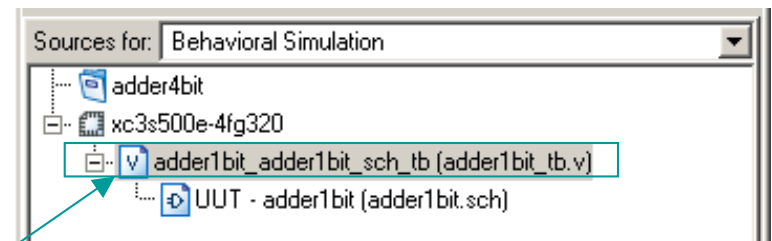
# Simulating the Test Fixture

- To simulate the test fixture, change the Sources view

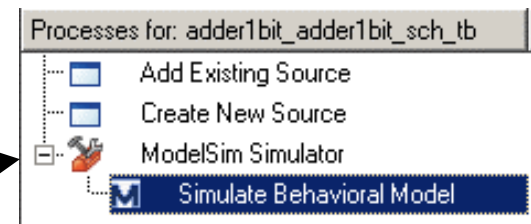
- In the **Sources for** selections box, switch from **Synthesis/Implementation** to **Behavioral Simulation**



- Sources view will change to show the test fixture

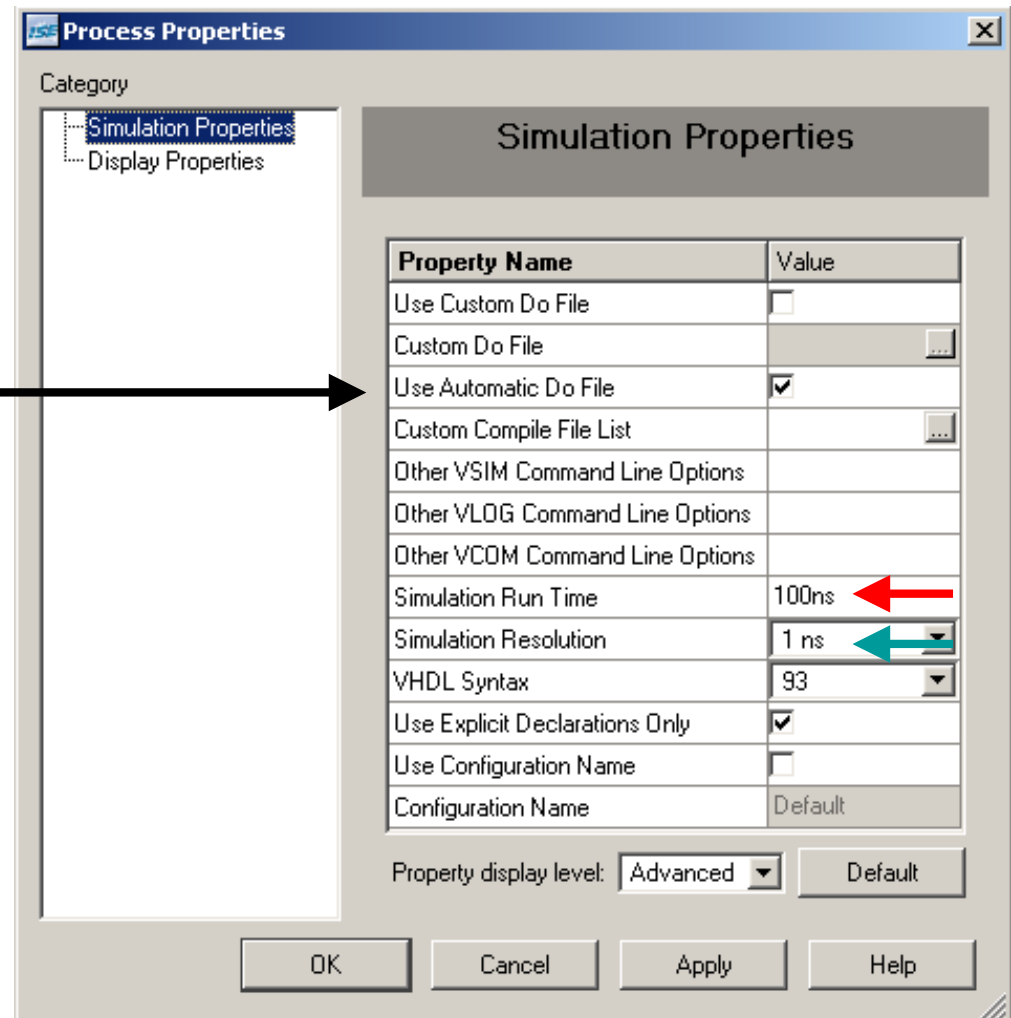


- Select the Verilog test fixture `adder1bit_tb`
- Expand Modelsim Simulator in the Processes window. You see "Simulate Behavioral Model".



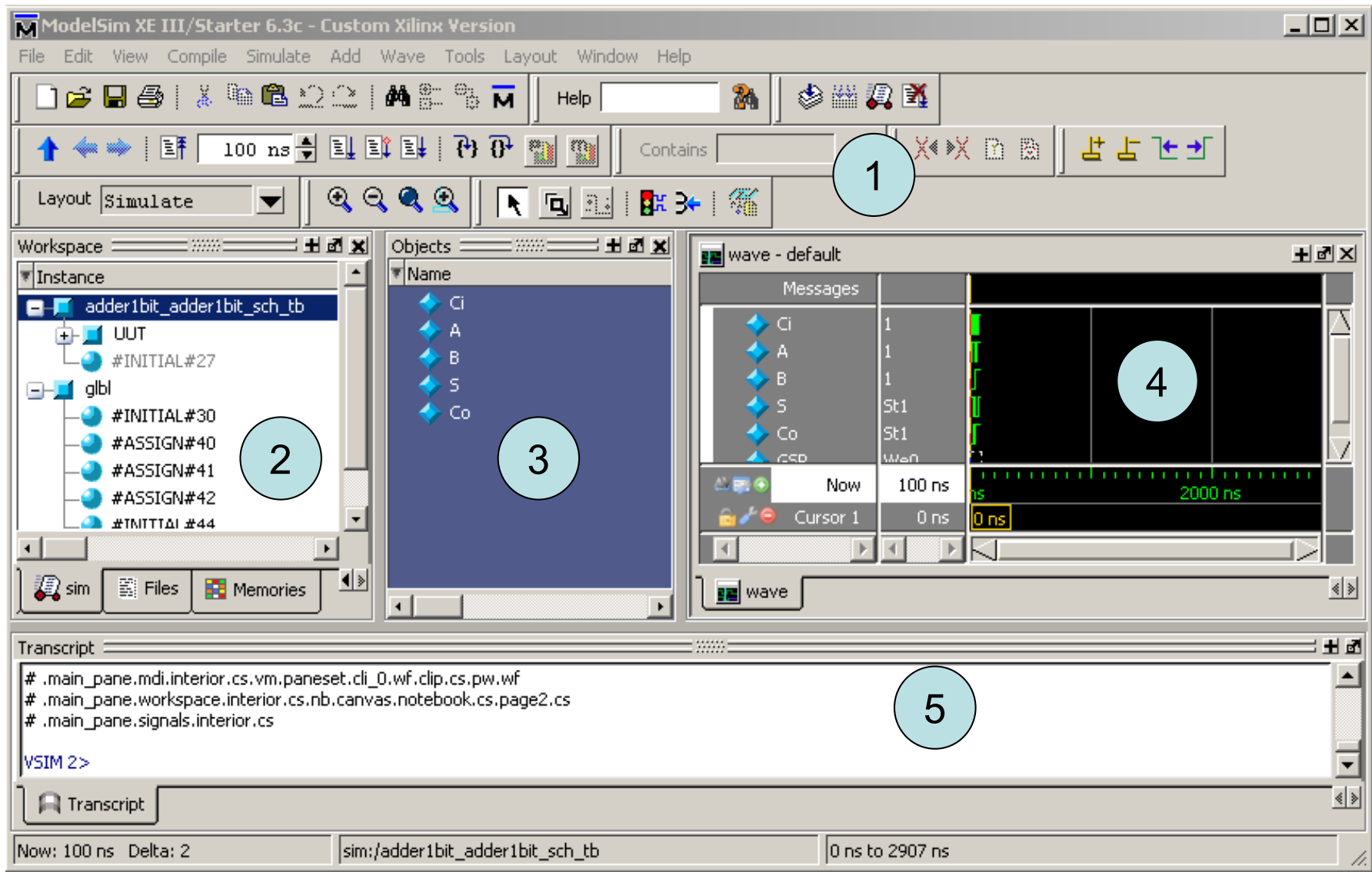
# Modelsim Properties

- Expand Modelsim Simulator in the Processes window
- Right click  Simulate Behavioral Model and select Properties
- “.do” files contain detailed commands for Modelsim. We allow Xilinx to automatically create a .do file for this design
- Change Simulation Run Time from 1000ns to 100ns
- Change Simulation Resolution from 1ps to 1ns
- Click Apply and then OK



# Launching Modelsim Simulator

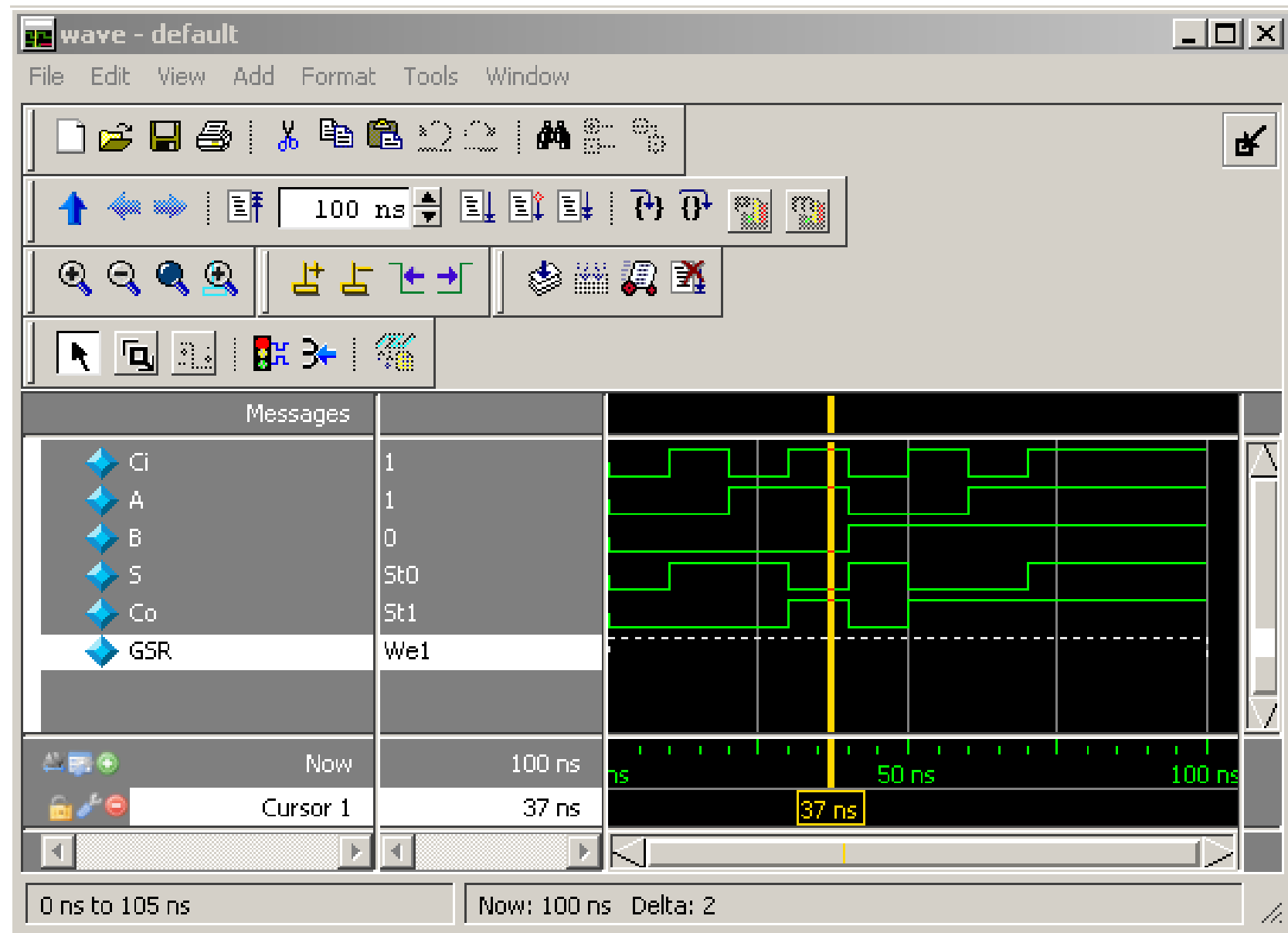
In the Processes window, double click on  Simulate Behavioral Model



# Modelsim Simulator

- Modelsim Simulator is launched (refer to the window on previous page):
- **1. Toolbar**
- **2. Instances window**  
You can see hierarchy of the design.
- **3. Objects window**  
You can see signals associated with the selected item in the hierarchy. You can add selected to signals to wave window.
- **4. Waveform window**  
You can change radix of signals, make time measurements, etc.
- **5. Transcript window**  
You can type in commands such as “run 100ns”

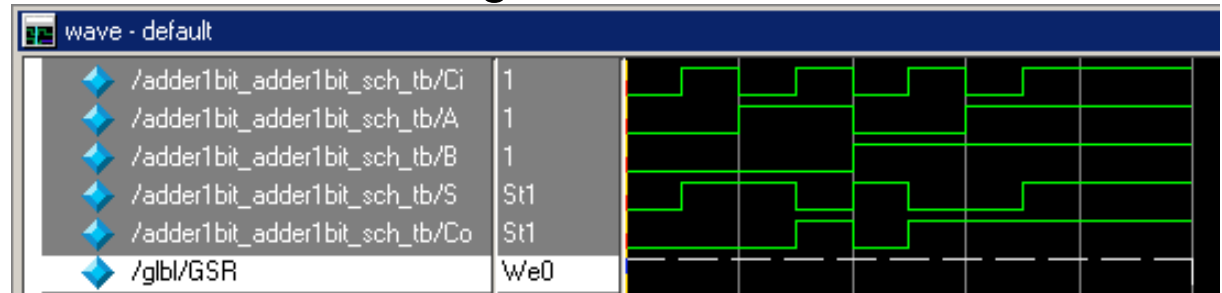
# Waveform



# Reading the Waveform

- Carefully inspect the waveform. It is best to dock out the waveform window. GSR stands for Global Set Reset control. ← Ignore for the time.

With the help of your TA, try changing radix, adding cursers, making time measurements, etc.



Find what is the difference between radix Decimal and radix Unsigned.

- Is the design functionally correct?
- Fill the table to the right to check
- Ask your TA how to produce the listing using ModelSim.

View => List (to display the list window)

Add to List => Signals in Region

| Input<br>{A,B,Ci} | Expected<br>{Co,S} | Observed<br>{Co, S} |
|-------------------|--------------------|---------------------|
| 000               | 00                 |                     |
| 001               | 01                 |                     |
| 010               | 01                 |                     |
| 011               | 10                 |                     |
| 100               | 01                 |                     |
| 101               | 10                 |                     |
| 110               | 10                 |                     |
| 111               | 11                 |                     |

# Listing window

```

ns↵ /adderlbit_adderlbit_sch_tb/Ci↵
delta↵ /adderlbit_adderlbit_sch_tb/A↵
      /adderlbit_adderlbit_sch_tb/B↵
      /adderlbit_adderlbit_sch_tb/S↵
      /adderlbit_adderlbit_sch_tb/Co↵

    0  +0                0 0 0 StX StX
    0  +2                0 0 0 St0 StX
    0  +3                0 0 0 St0 St0
   10  +0                1 0 0 St0 St0
   10  +2                1 0 0 St1 St0
   20  +0                0 1 0 St1 St0
   30  +0                1 1 0 St1 St0
   30  +2                1 1 0 St0 St0
   30  +3                1 1 0 St0 St1
   40  +0                0 0 1 St0 St1
   40  +2                0 0 1 St1 St1
   40  +3                0 0 1 St1 St0
   50  +0                1 0 1 St1 St0
   50  +2                1 0 1 St0 St0
   50  +3                1 0 1 St0 St1
   60  +0                0 1 1 St0 St1
   70  +0                1 1 1 St0 St1
   70  +2                1 1 1 St1 St1

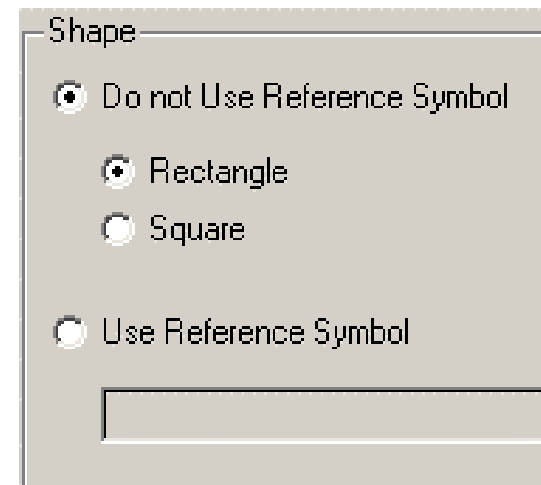
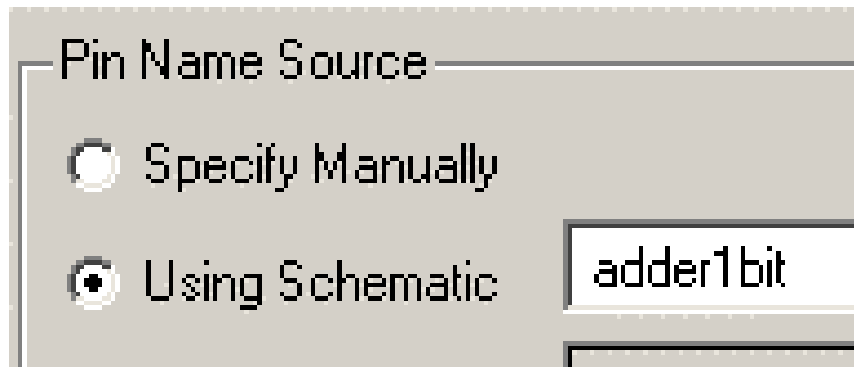
```

# Outline

- Introduction to Xilinx ISE Project Navigator
- Introduction to Xilinx ISE Schematics Entry Tool
- Simulation Using Verilog Test Fixture in ModelSim XE
- **Creating Hierarchical Designs**
- Creating TOP Design to interface to Nexys2 board, implementing the design using ISE, and downloading the bit file using Adept

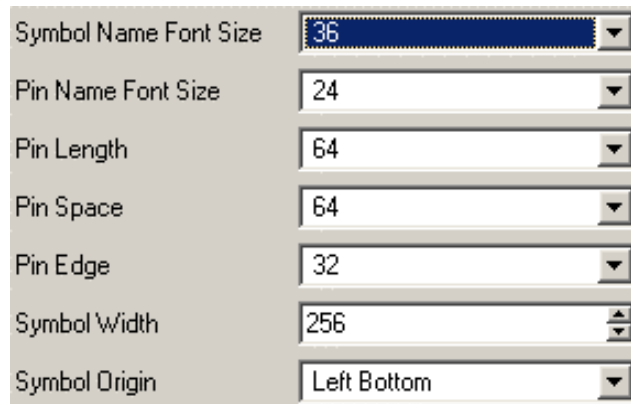
# Creating a Symbol

- To create hierarchical design, basic building blocks have to be encapsulated into user-defined symbols that can be instantiated at the next higher (hierarchical) level
- To create a symbol for adder1bit:
  - Open the schematic `adder1bit.sch` by double clicking it in the Sources window
  - Click Tools → Symbol Wizard
  - On the Symbol Wizard – Source Page, under the Pin Name Source choose **Using Schematics: adder1bit**
  - Leave shape set to Rectangle and click Next



# Creating a Symbol

- On the Symbol Wizard – Pin Page, check the order in which pins appear, the polarity of the pins and the side of symbol where the pin will appear.
- Click Next
- On the Symbol Wizard – Options Page set Symbol Name Font Size to 36



Symbol Name Font Size: 36

Pin Name Font Size: 24

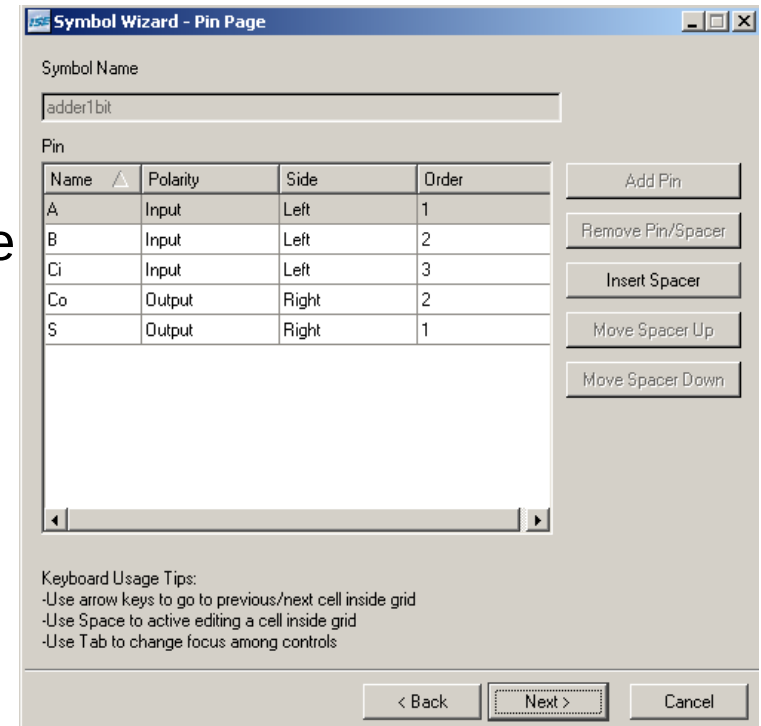
Pin Length: 64

Pin Space: 64

Pin Edge: 32

Symbol Width: 256

Symbol Origin: Left Bottom



Symbol Name: adder1bit

| Name | Polarity | Side  | Order |
|------|----------|-------|-------|
| A    | Input    | Left  | 1     |
| B    | Input    | Left  | 2     |
| Ci   | Input    | Left  | 3     |
| Co   | Output   | Right | 2     |
| S    | Output   | Right | 1     |

Buttons: Add Pin, Remove Pin/Spacer, Insert Spacer, Move Spacer Up, Move Spacer Down

Keyboard Usage Tips:  
-Use arrow keys to go to previous/next cell inside grid  
-Use Space to active editing a cell inside grid  
-Use Tab to change focus among controls

Buttons: < Back, Next >, Cancel

- Leave the rest as default and click Next
- Click Finish at the Symbol Wizard – Preview Page

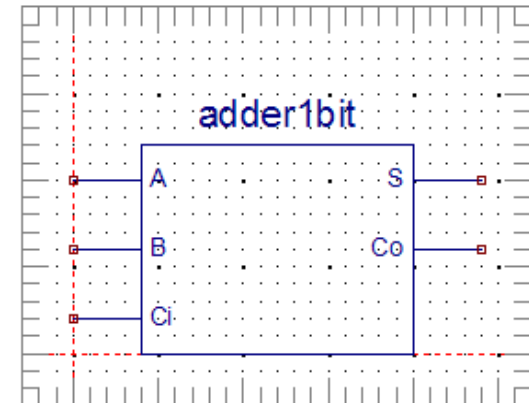
# Editing a Symbol

- `adder1bit.sym` appears in the Schematic Editor workspace

- You can edit any symbol using the drawing tools

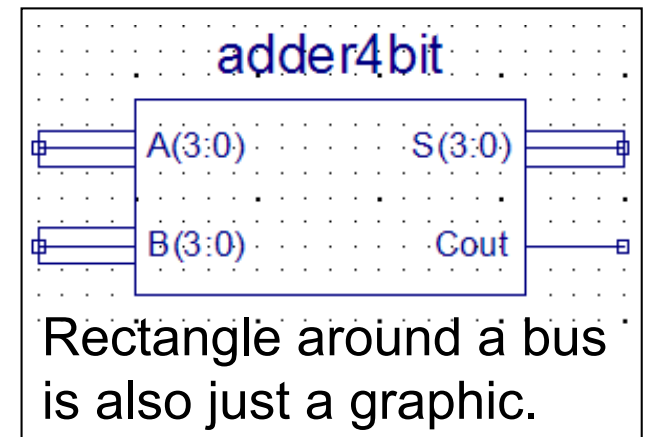


- We will not edit this symbol in this tutorial. However, note that, the small red squares representing the pins, and their labels are the only significant items in the symbol. The rest of the symbol is just a graphic and can be mended as you like.



- Close the Symbol and Schematic files.

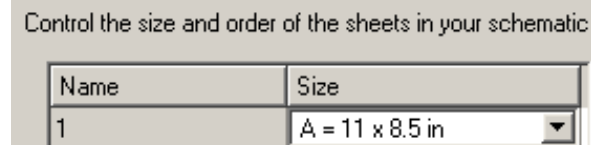
- Next, we create a 4-bit adder (`adder4bit`) using four 1-bit adder symbols (`adder1bit`)



# Creating adder4bit

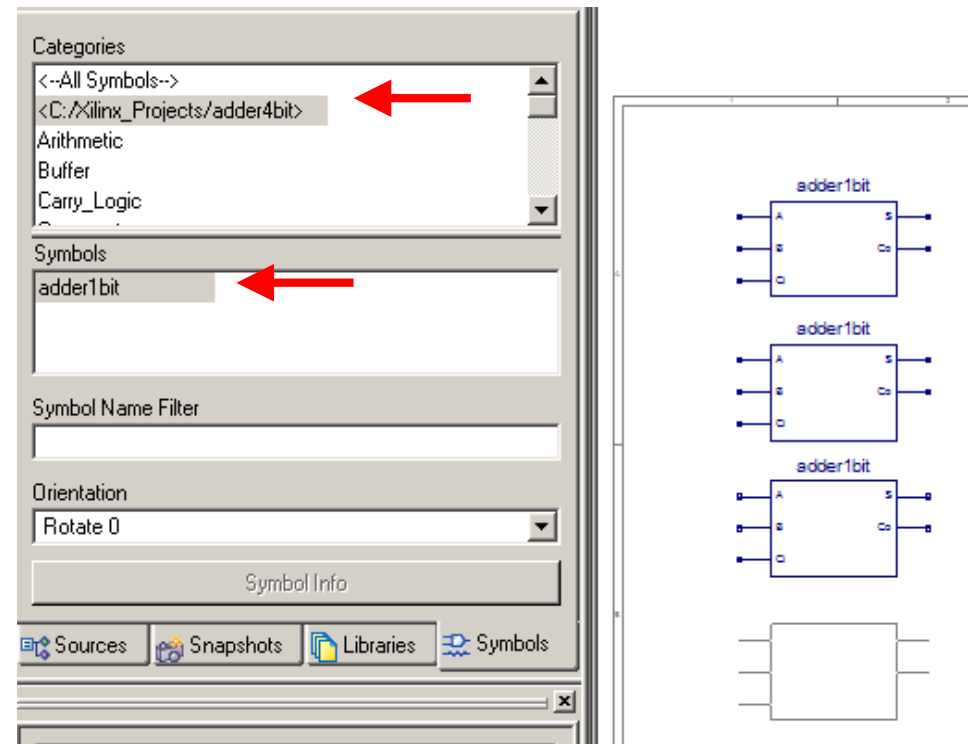
- Create a new schematics file adder4bit using the Create New Source in the Processes window.

☒ Add to project



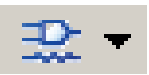
- A new category of symbols appears in the Symbols window <C:/xilinx\_projects/adder4bit>, indicating the project directory. All user-defined symbols are stored in their respective project directories.

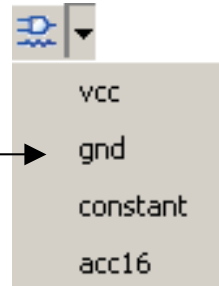
- Select adder1bit symbol and drop four copies on the workspace



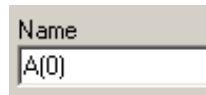
# Creating adder4bit

- Connect adder1bit symbols to create a correct 4-bit adder

- Connect Ci of the first adder1bit to the gnd symbol. Use the button 

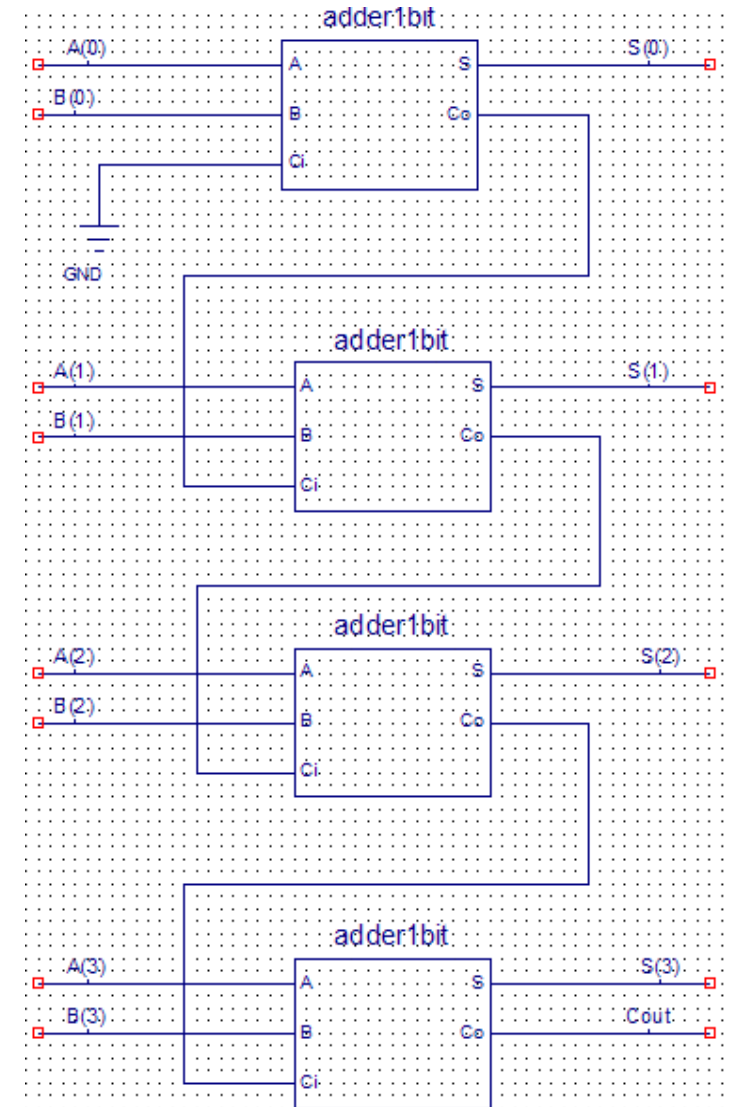


- Name the A and B inputs of each adder1bit as A(0) through A(3) and B(0) through B(3), respectively



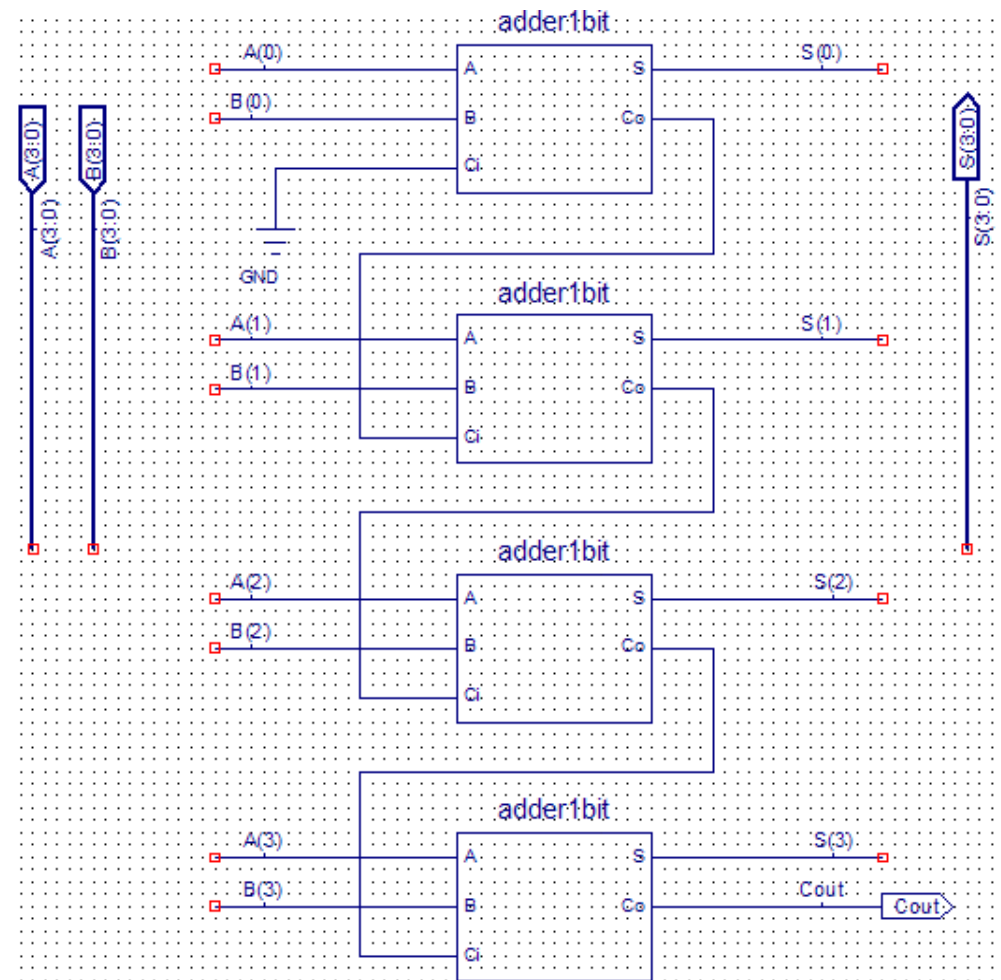
- to name multiple nets with names such as **A(0)**, **A(1)**, **A(2)**, ..., select **Increase the name** from the **After naming the branch or net** option in the Options window

- Complete schematics as shown



# Adding Bus and Bus Taps

- To create signals A, B and S as buses, add three dangling nets (neither end connected to any pin) and name them A(3:0), B(3:0) and S(3:0). This creates three buses
- These three nets will appear as thicker lines than ordinary (1-bit) wires as they are buses
- Then, connect I/O markers to each of these buses and one to Cout
- Completed schematics is shown here



# Adding Bus Taps

- It is not necessary to physically connect a bus to the various nets forming a bus. The tool associates the nets S(0), S(1), S(2) and S(3) with the bus S(3:0) automatically.

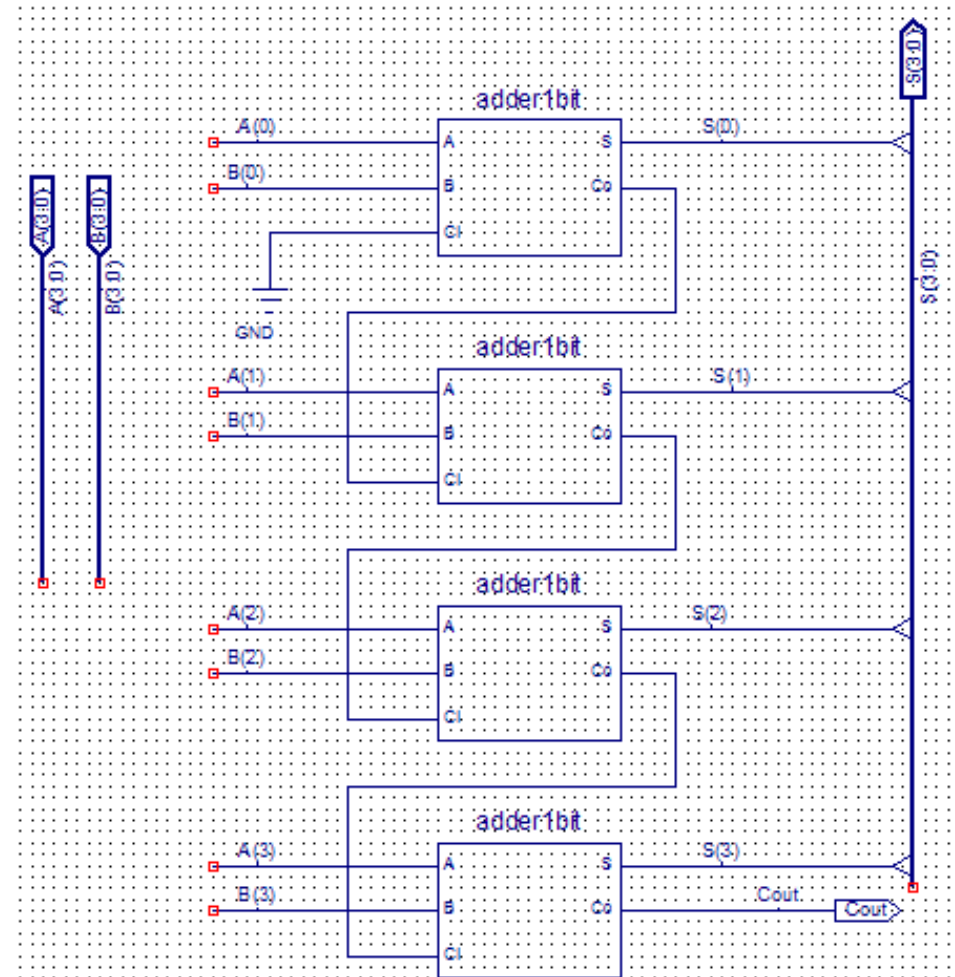
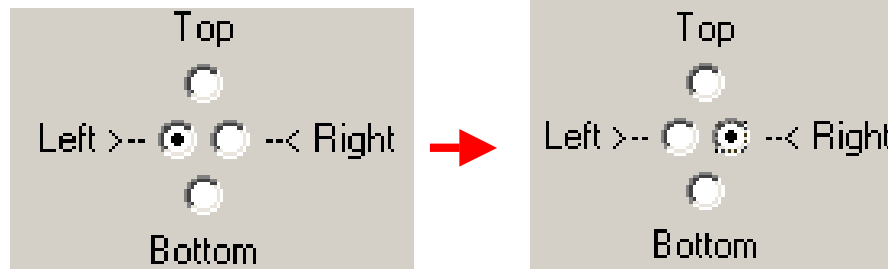
- Let us add bus taps to S(3:0) bus.



- Click on the Bus Tap tool

Add Bus Tap Options

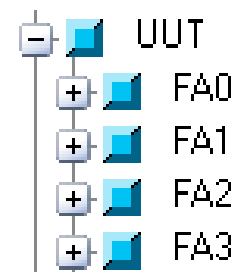
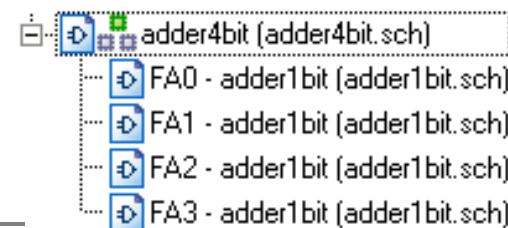
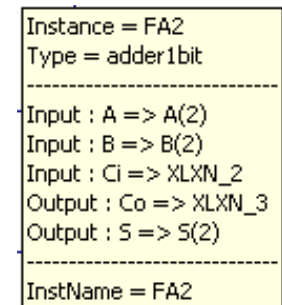
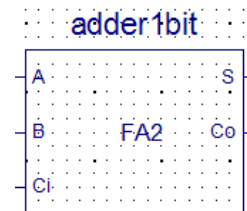
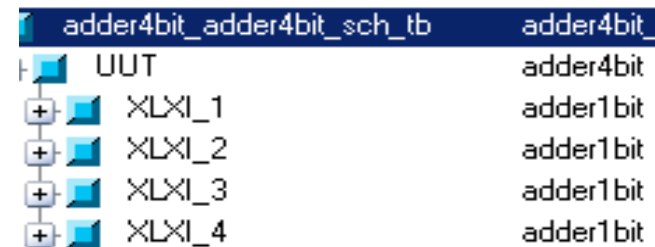
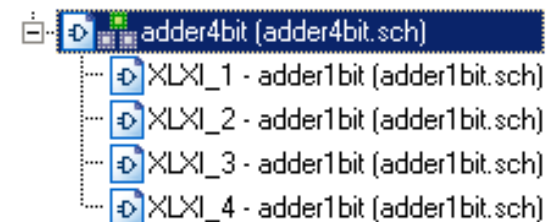
- Select Orientation



# Labeling instances may be very beneficial

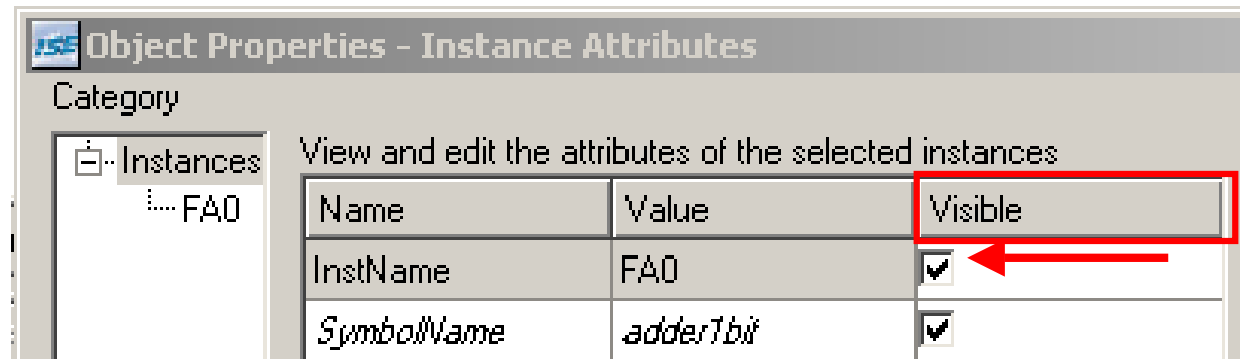


- The four instances of the adder1bit in the adder4bit design appear with machine generated names as seen on the side.
- Also, if you simulate the adder4bit design in ModelSim, the four instances of the adder1bit appear with machine generated names as seen on the side.
- The hierarchical names of nodes inside an instance in ModelSim look like  /adder4bit\_adder4bit\_sch\_tb/UUT/XLXI\_2/Co
- After labeling the instances as FA0, FA1, FA2, and FA3, the same looks like  /adder4bit\_adder4bit\_sch\_tb/UUT/FA1/Co

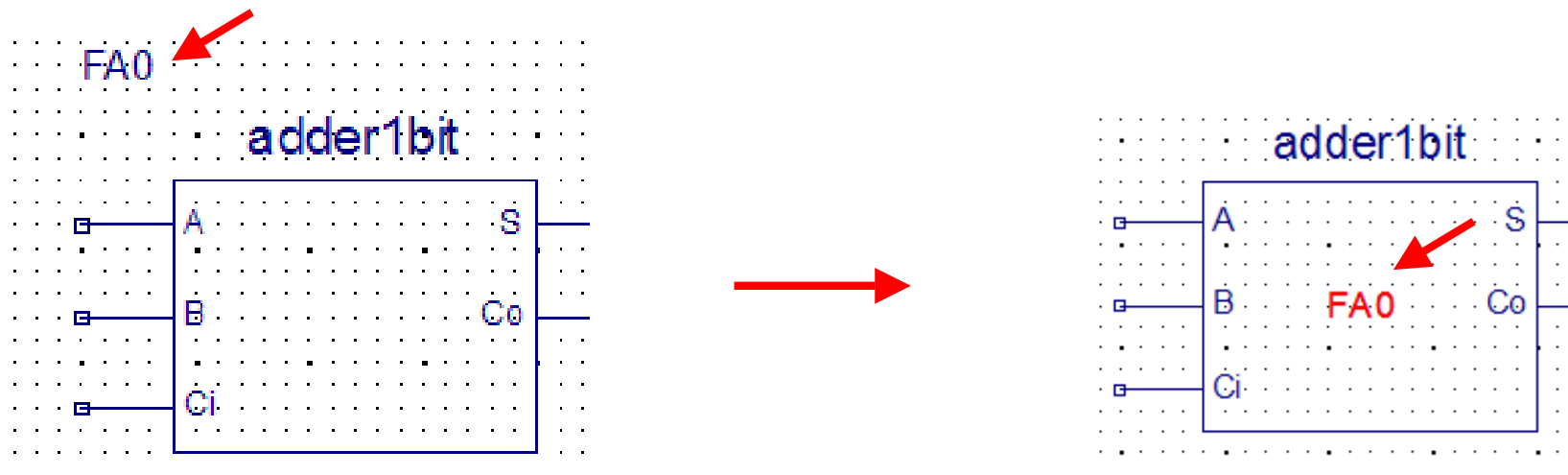


# Make Instance labels visible

Right click on an instance and select object properties.

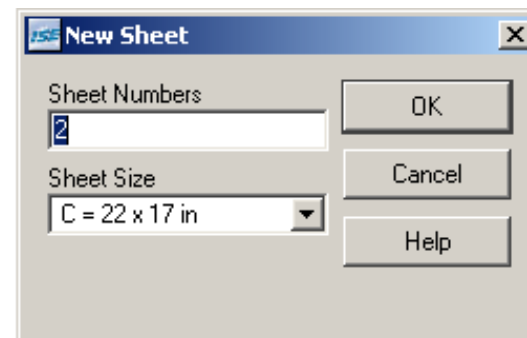
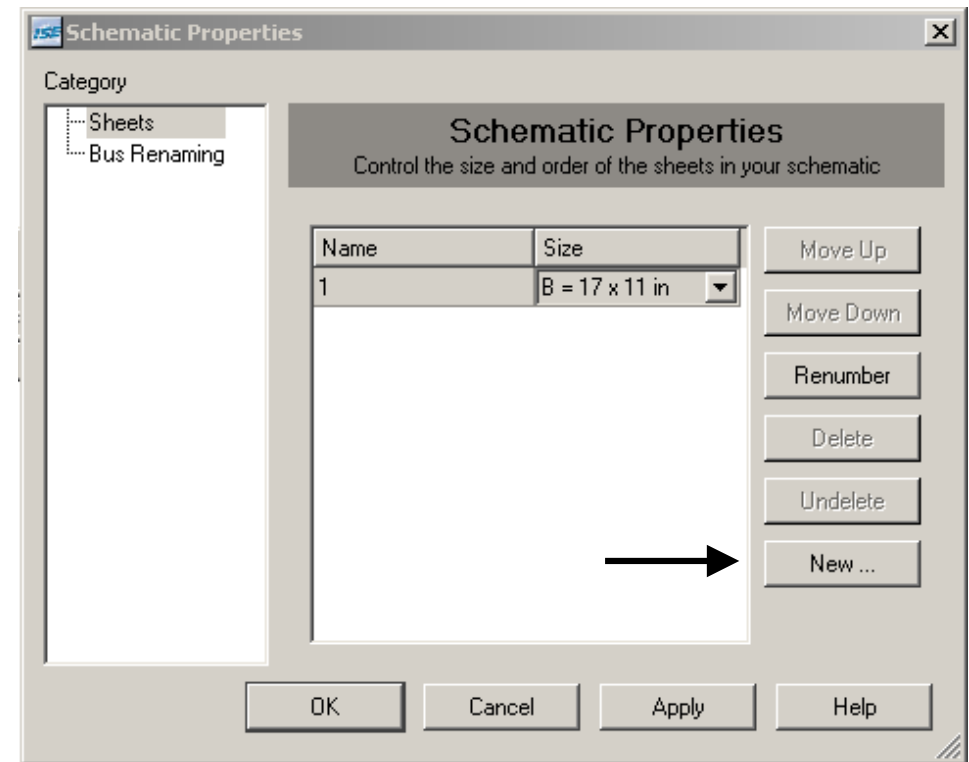


Drag and drop the instance label.

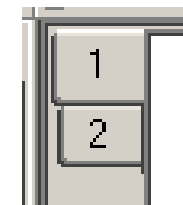


# Hierarchical vs. Flat design

- For real designs, you often require multiple sheets. You may not be able to fit everything in one sheet.
- While hierarchical design can reduce the number of sheets needed at a particular level of hierarchy, a judicious mix of hierarchical and flat design (using multiple sheets) is the best choice.
- To add additional sheets to a design, simply right-click on the schematic => Object Properties => New



| Name | Size           |
|------|----------------|
| 1    | B = 17 x 11 in |
| 2    | B = 17 x 11 in |



# Outline

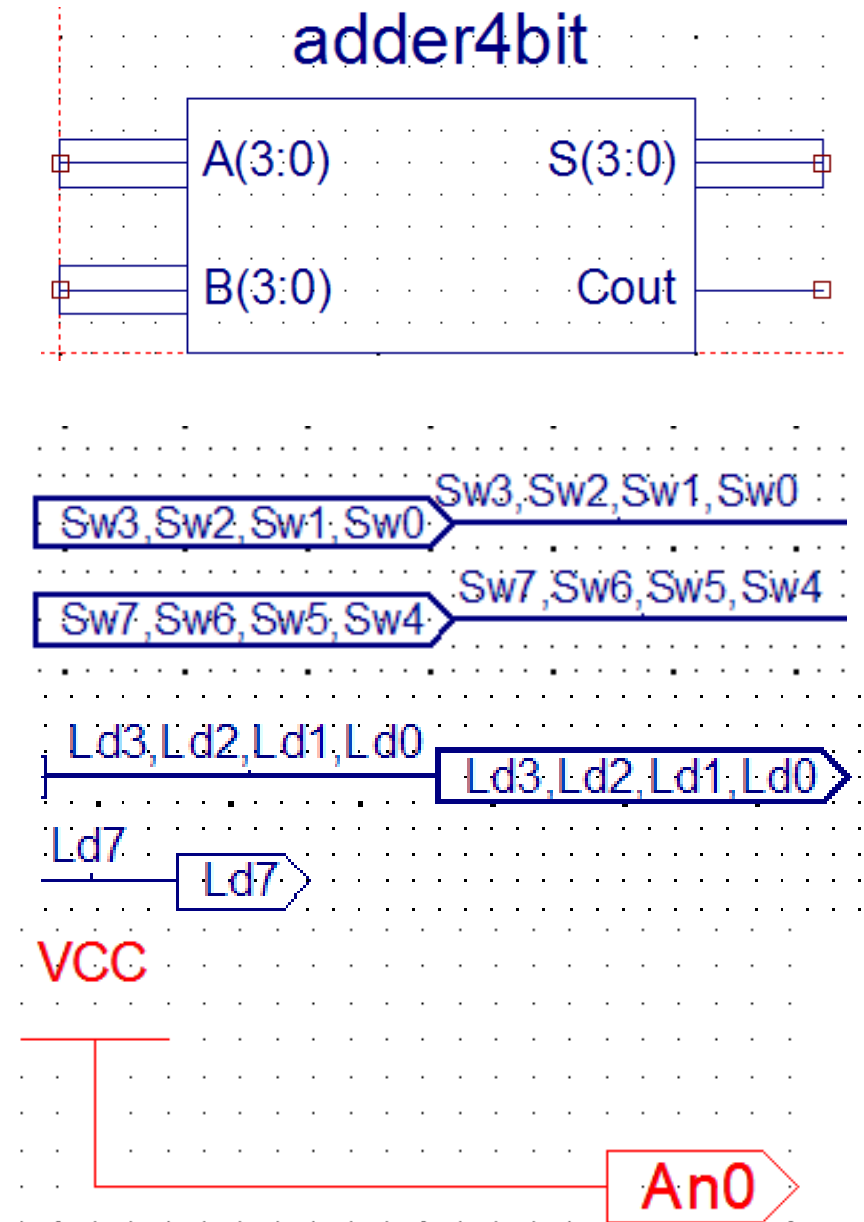
- Introduction to Xilinx ISE Project Navigator
- Introduction to Xilinx ISE Schematics Entry Tool
- Simulation Using Verilog Test Fixture in ModelSim XE
- **Creating Hierarchical Designs**
- Creating TOP Design to interface to Nexys2 board, implementing the design using ISE, and downloading the bit file using Adept

**Very briefly!**

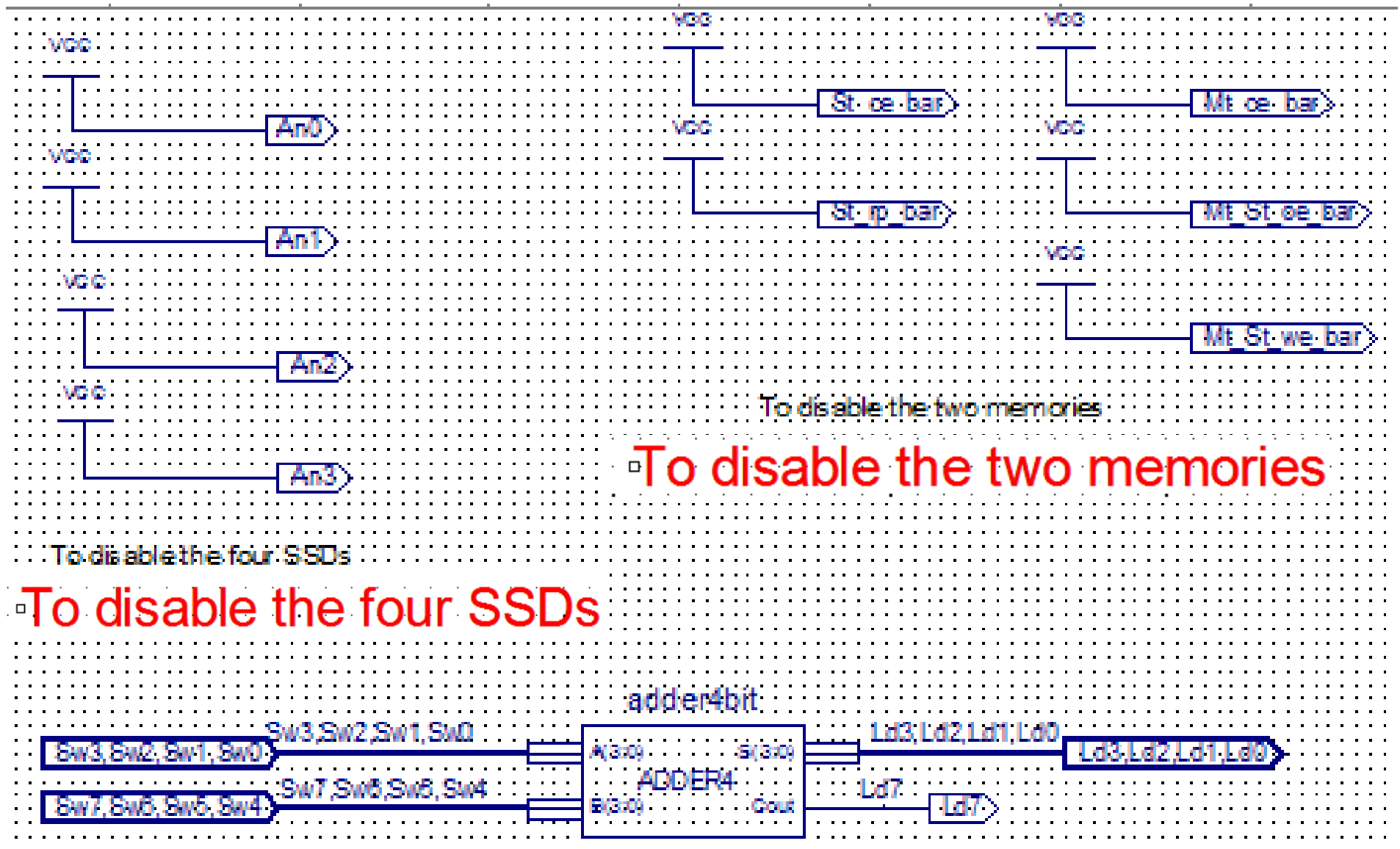


# TOP design top\_adder4bit.sch

- To interface the adder4bit design to the Nexys2 board, it is best to create a symbol for the adder4bit and interface it to the I/O resources of the board.
- It is best to use signal names such as Sw1, Ld1, etc. for connections to the resources on the board and use these signal names uniformly in the .ucf files (user constraint files) of all our designs.
- In the top schematic, we will also disable unused resources, particularly memories, and also the 7-segment displays



# top\_adder4bit.sch



# Pin Numbers in the UCF file

```
NET St_ce_bar LOC = R5;  
NET St_rp_bar LOC = T5;  
NET Mt_ce_bar LOC = R6;  
NET Mt_St_oe_bar LOC = T2;  
NET Mt_St_we_bar LOC = N7;
```

## Memories

```
NET Sw0 LOC = G18;  
NET Sw1 LOC = H18;  
NET Sw2 LOC = K18;  
NET Sw3 LOC = K17;  
NET Sw4 LOC = L14;  
NET Sw5 LOC = L13;  
NET Sw6 LOC = N17;  
NET Sw7 LOC = R17;
```

## Switches

## Pin number

```
NET Ld0 LOC = J14;  
NET Ld1 LOC = J15;  
NET Ld2 LOC = K15;  
NET Ld3 LOC = K14;  
# NET Ld4 LOC = E17;  
# NET Ld5 LOC = P15;  
# NET Ld6 LOC = F4;  
NET Ld7 LOC = R4;
```

## LEDs

```
NET An0 LOC = F17;  
NET An1 LOC = H17;  
NET An2 LOC = C18;  
NET An3 LOC = F15;
```

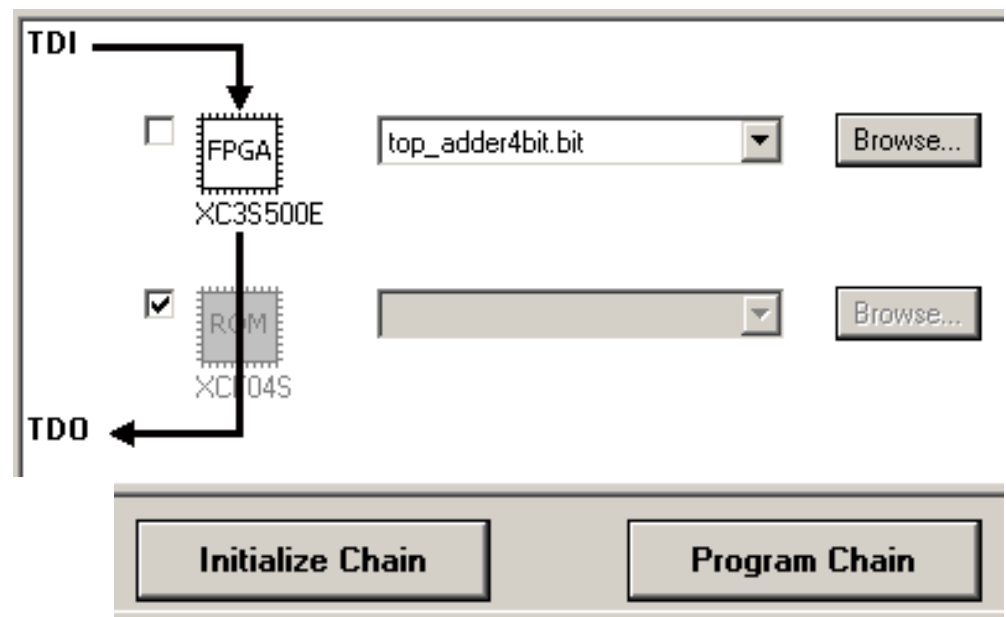
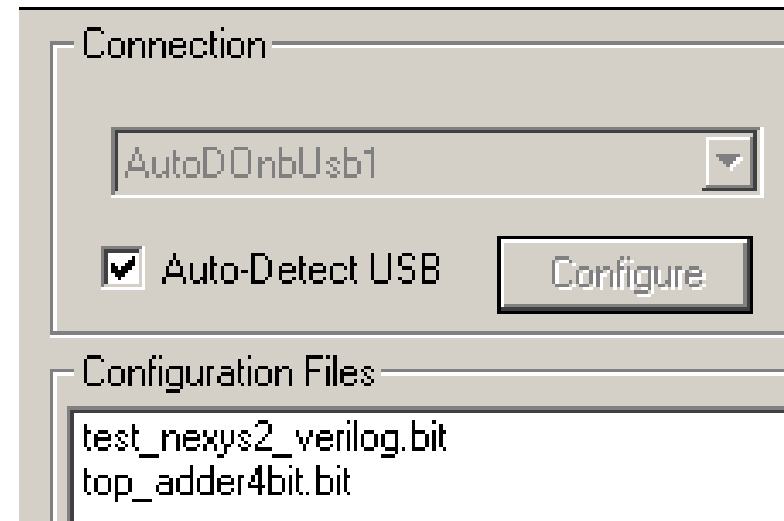
## Anodes for seven seg. displays

# I/O on the Nexys2 board used in this design



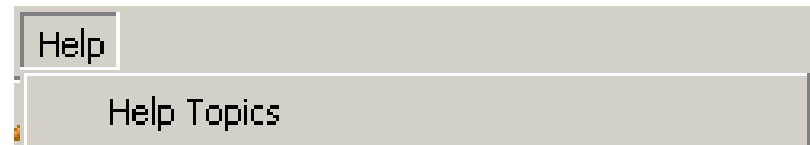
# Download the .bit file (top\_adder4bit.bit) to Nexys2

- Setup jumpers on Nexys2 board for USB-powering arrangement.
- Connect Nexys2 to the PC using USB cable.
- Invoke Adept Exporter.
- Initialize Chain. Bypass the flash memory chip.
- Add the top\_adder4bit.bit file.
- Program.



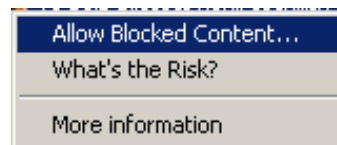
# References

- For help on ISE topics,  
Help => Help topics

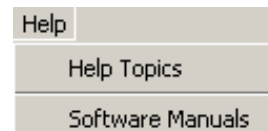


To help protect your security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your computer.

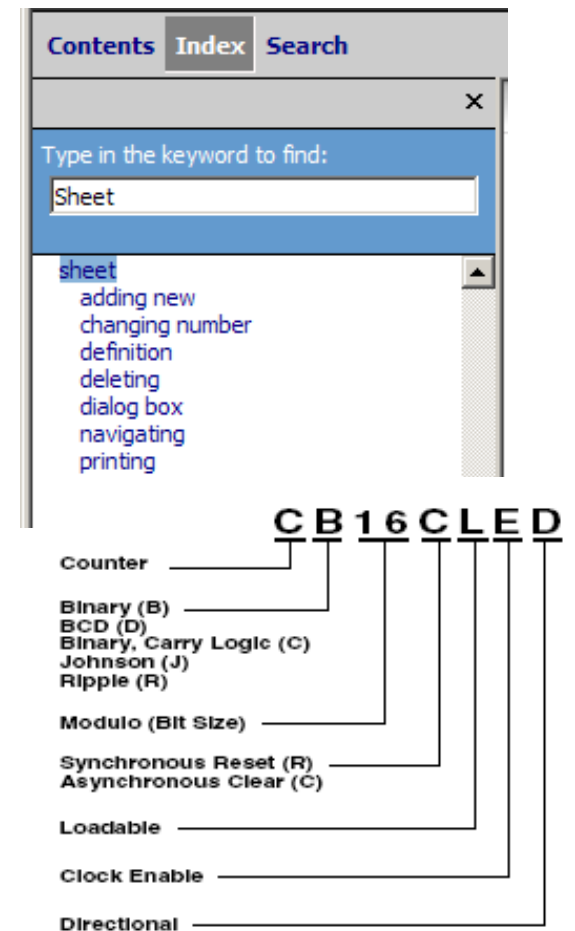
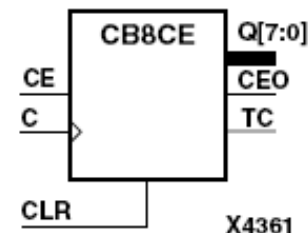
**Allow blocked content.**  
**Use Index search to**  
**get help.**



- For information on functional elements available in the symbol library such as counter cb8ce



**Help => Software Manuals =>**  
**libraries guide on the left panel =>**  
**Spartan-3 Libraries Guide for**  
**Schematic Designs => Functional**  
**Categories => Counter => CB8CE**



# CONGRATULATIONS ON YOUR FINISH!

