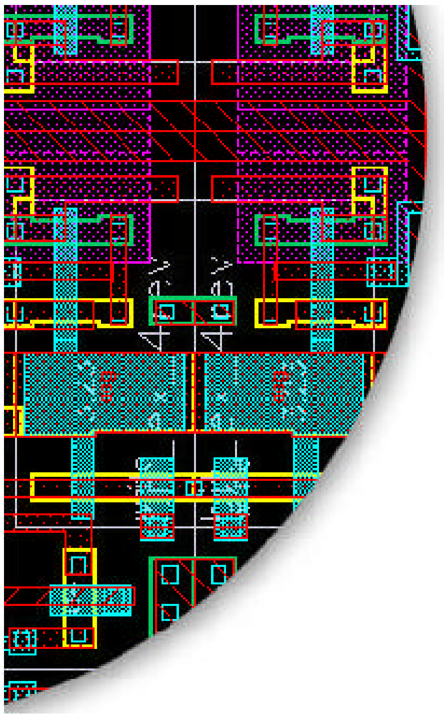


ispLEVER™ 1.0 Tutorial Manual

This online manual contains three CPLD tutorials designed to benefit all ispLEVER software users. These tutorials are hands-on projects that provide a fun and simple way to become familiar with the ispLEVER development software.

Tutorials covered in this manual include:

- Schematic and ABEL-HDL Design
- HDL Synthesis Design with Synplify
- HDL Synthesis Design with LeonardoSpectrum

Contents

Tutorial 1: Schematic and ABEL-HDL Design

The ispLEVER software supports mixed-mode design entry: a design with at least one schematic module as the top project source, and one or more sources of the same language. The language sources are mutually exclusive, so you must choose one of the three types when you begin a new project. For example, in addition to the schematic, you must choose an ABEL-HDL source, a Verilog HDL source, or a VHDL source.

This tutorial leads you through all the basic steps to design, simulate, implement, and verify a counter circuit targeted to a CPLD device. The design consists of a top-level schematic and two lower-level ABEL-HDL modules. A top-down, bottom-up design methodology is used.

Organization

This tutorial is divided into three modules, with Module 1 divided into six lessons. You should start with Module 1 and continue to the end of Module 3.

- Module 1: Design Entry and Simulation
- Module 2: Design Implementation
- Module 3: Design Verification

Prerequisites

None

Learning Objectives

When you have completed this tutorial, you should be able to:

- Setup an ispLEVER schematic project
- Create a top-level schematic source
- Create and import ABEL-HDL sources into the project
- Navigate the design
- Import a test vector file and run functional simulation
- Set pin constraints, fit the design, and read the Fitter report
- Run timing analysis and simulation and analyze the results
- Correlate simulation results using cross-probing
- Export the netlist and delays for timing simulation
- Build board-level Stamp models of a design

Time to Complete This Tutorial

The time to complete this tutorial is about 90 minutes.

Module 1: Design Entry and Simulation

The schematic design entry environment is a set of tools that allows you to capture the structure of a design as either a flat description or a hierarchical set of components, and the connectivity between these components. Then you can use this description to drive the Fitter and verification tools. Designs can be single-level (flat) or multi-level (hierarchical). Schematics can be drawn on multiple "sheets" and any size.

ABEL-HDL is a hierarchical hardware description language that supports a variety of behavioral input forms, including high-level equations, state diagrams, and truth tables.

Functional simulation is the process of simulating the functionality of your RTL design before it is compiled, thus letting you find and correct basic design errors sooner. While functional simulation will verify your Boolean equations, it does not indicate timing problems.

This module leads you through the steps to design and simulate a counter circuit targeted to an ispMACH 5000VG CPLD device. The design consists of a top-level schematic and two lower-level ABEL-HDL modules. A top-down, bottom-up design methodology is used.

Prerequisites

None

Learning Objectives

When you have completed this module, you should be able to:

- Add various schematic elements to create a top-level schematic source
- Check the schematic for errors
- Use the Text Editor to create a new ABEL-HDL source
- Import an ABEL-HDL source into the project
- Use the Hierarchy Navigator to navigate through the design and try "debug" methods

Time to Complete This Module

The time to complete this module is about 50 minutes.

Lesson 1: Setting Up a Project

In Lesson 1 you will set up the tutorial project. The ispLEVER Software employs the concept of a project. A project is a design. Each project has its own directory in which all source files, intermediate data files, and resulting files are stored.

Tasks covered in this lesson are:

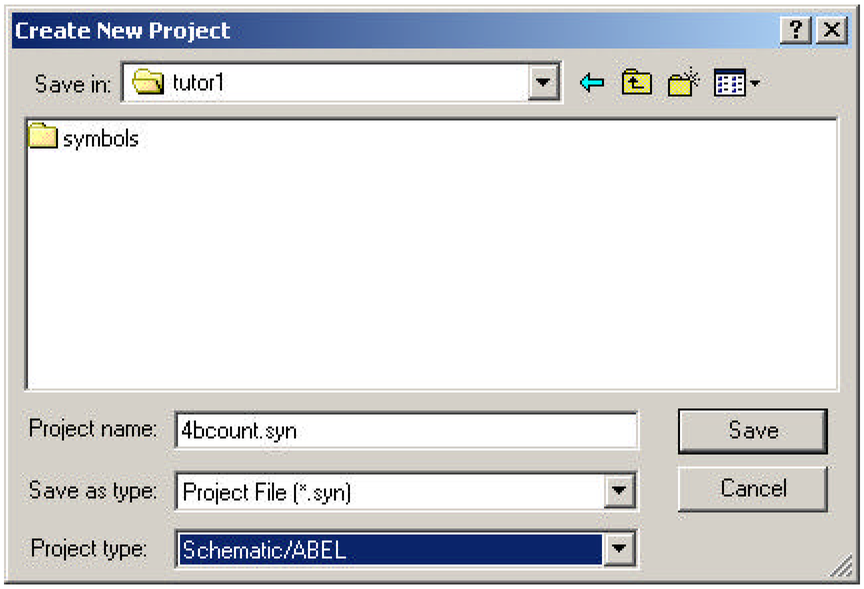
- Task 1: Create a New Project
- Task 2: Select a Device
- Task 3: Copy Schematic Symbols to the Project Directory

Task 1: Create a New Project

To begin a new project, you need to create a project directory. Then you need to give the project file a name (`.syn`). The Project Navigator will use this file name later to reload the project.

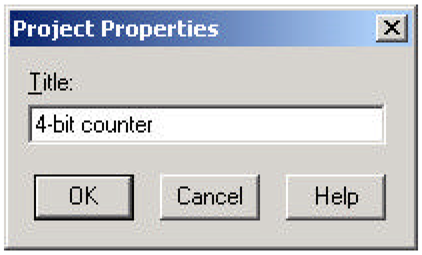
To create a new project:

1. Start the ispLEVER system, if it is not already running.
2. In the Project Navigator, choose **File > New Project** to open the Create New Project dialog box.
3. In the dialog box, change to the directory:
   `<`*`install path`*`>\ispcpld\examples\tutorial\`**`tutor1`**.
   - In the Project name box, type **`4bcount.syn`**.
   - In the Project type box, select **Schematic/ABEL**.
   - Click **Save**.



Project name: 4bcount.syn
Save as type: Project File (*.syn)
Project type: Schematic/ABEL

4. The default project title, `Untitled`, appears in the Sources window of the Project Navigator. Double-click the project title (**`Untitled`**) to open the Project Properties dialog box.
5. Type **`4-bit counter`** as your project title and click **OK**.



Title: 4-bit counter

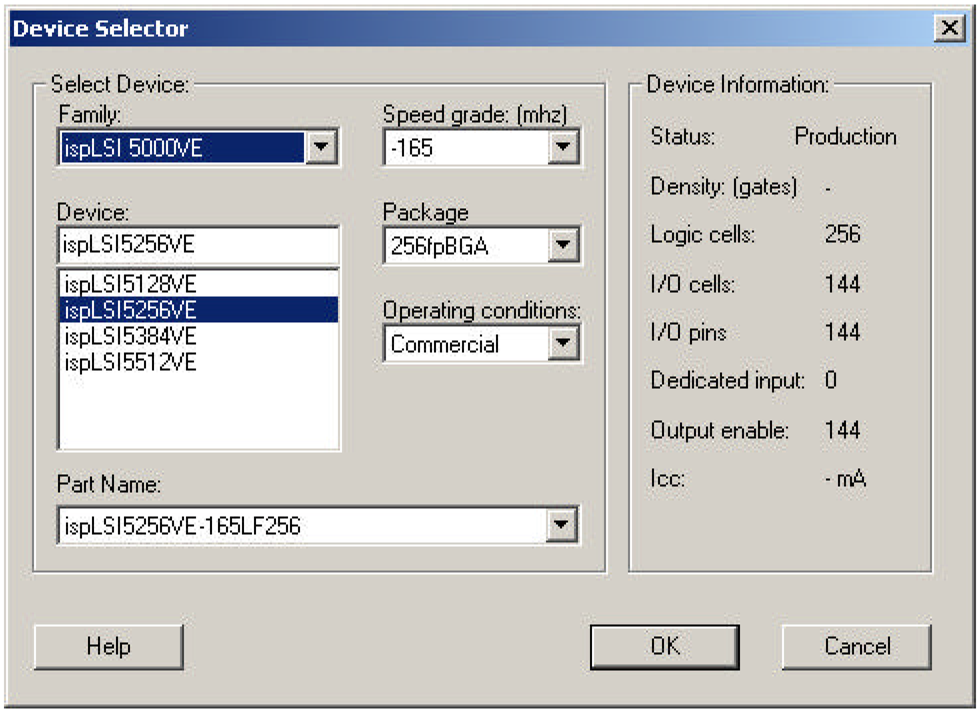
The new project title appears in the Sources window.

Task 2: Target a Device

In the Project Navigator Sources window is the device icon next to the target device for the project. The Project Navigator lets you target a design to a specific Lattice device at any time during the design process. The default device is the ispLSI5256VE-165LF256. For this project, you will target a different device.

To view the list of available devices and to change the target device:

1. In the Sources window, double-click the part name to open the Device Selector dialog box. The dialog box shows the default device.



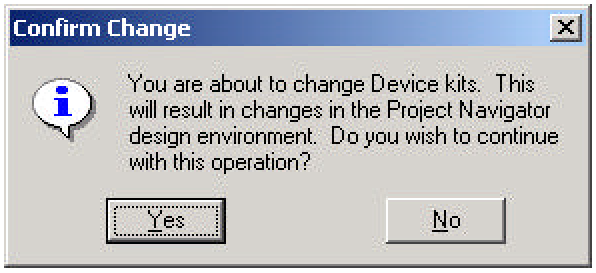
Family: ispLSI 5000VE
Speed grade: (mhz) -165
Device: ispLSI5256VE
Package 256fpBGA
Operating conditions: Commercial
Part Name: ispLSI5256VE-165LF256

Status: Production
Density: (gates) -
Logic cells: 256
I/O cells: 144
I/O pins 144
Dedicated input: 0
Output enable: 144
Icc: - mA

- Under Family, select **ispMACH 5000VG** from the drop-down list.

- Accept the default settings and click **OK**.

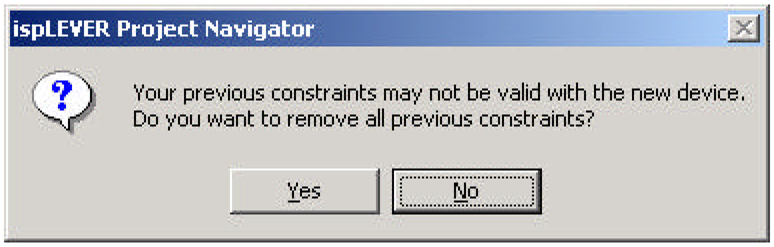
2. In the Confirm Change dialog box, click **Yes** to confirm that you wish to change device kits.



Confirm Change

You are about to change Device kits. This will result in changes in the Project Navigator design environment. Do you wish to continue with this operation?

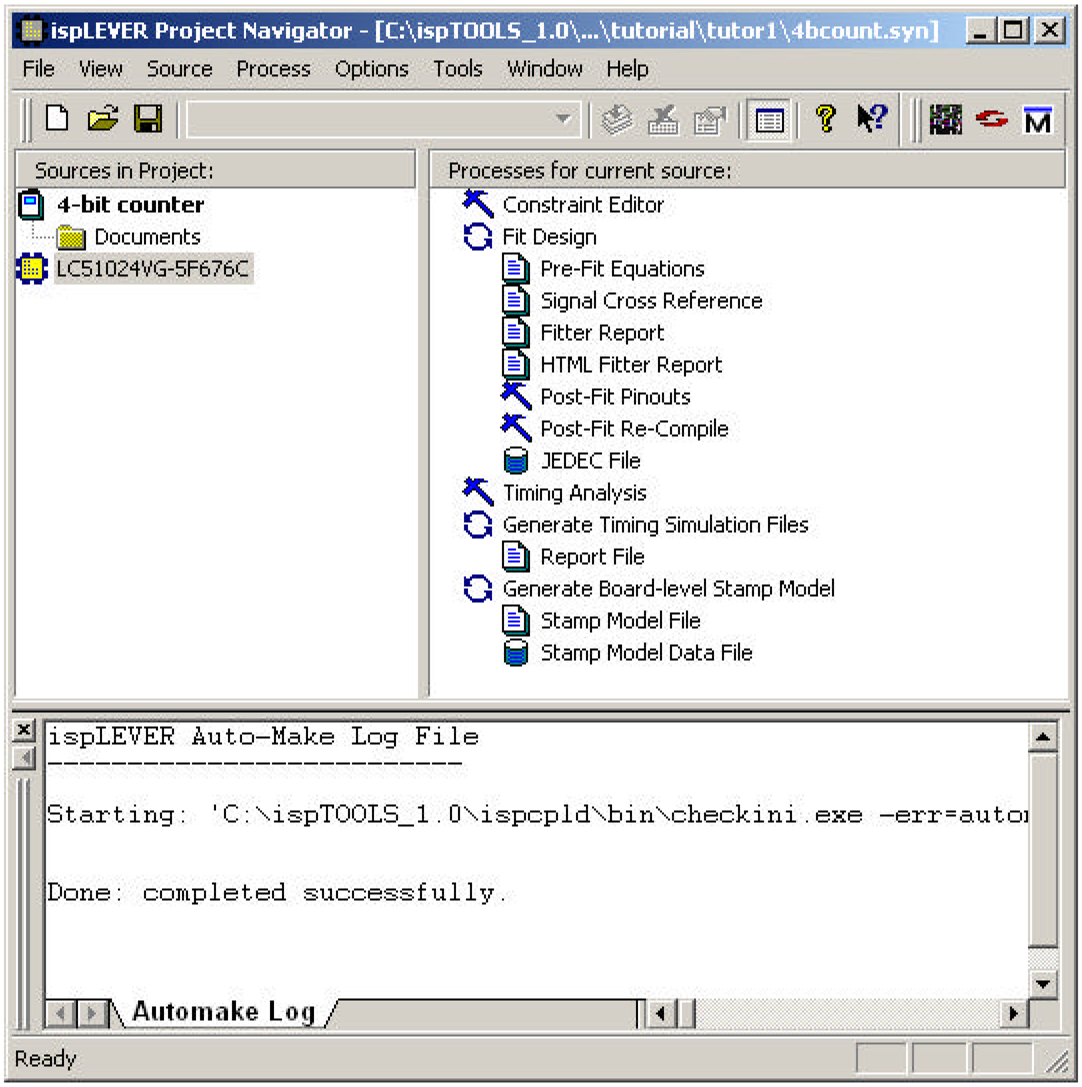
3. In the next dialog box, click **No**.



ispLEVER Project Navigator

Your previous constraints may not be valid with the new device. Do you want to remove all previous constraints?

4. Your Project Navigator should look like this:



```
ispLEVER Auto-Make Log File
--------------------------

Starting: 'C:\ispTOOLS_1.0\ispcpld\bin\checkini.exe -err=auto

Done: completed successfully.
```

Task 3: Copy Schematic Symbols to the Project Directory

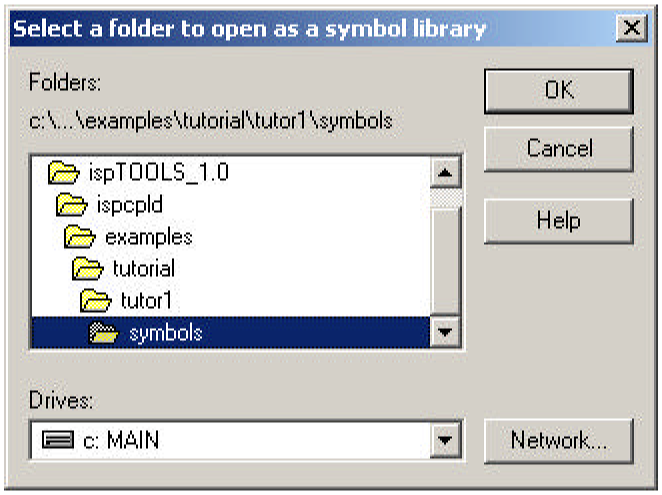
A schematic is composed of symbols, wires, I/O markers, graphics, and text. Symbols are graphic representations of components. The term "symbol" usually refers to an electrical symbol, such as a gate or a sub-circuit. You can draw graphic-only symbols (such as title blocks) with the Symbol Editor, but these have no electrical meaning.

Symbols are the most basic elements of a schematic. Symbols represent primitive design elements, whether those elements are individual transistors, complete gates, or a complex IC. A symbol can also be the hierarchical representation of a sub-circuit (a "Block" symbol).

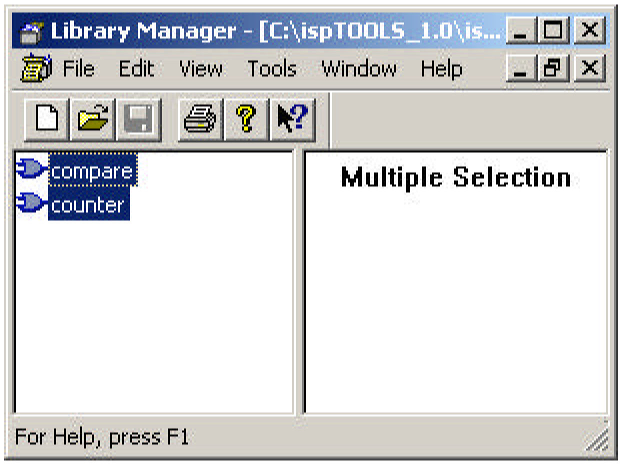
In this task you will copy two symbol files to your project directory so that you can use these pre-made symbols in your design.

To copy schematic symbols to the project directory:

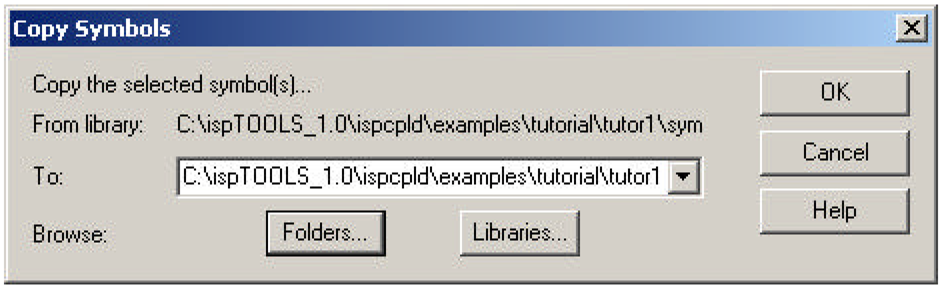
1. In the Project Navigator, choose **Window** > **Library Manager**.
2. In the Library Manager, choose **File** > **Open Folder** to open the dialog box.
3. Change to the directory `\tutor1\symbols`.



4. Click **OK** to open the library in the Library Manager.
5. Select the two symbols (**compare** and **counter**).


Library Manager - [C:\ispTOOLS_1.0\is...
File Edit View Tools Window Help
compare
counter
Multiple Selection
For Help, press F1

6. Choose **Edit** > **Copy Symbol(s)** to open the Copy Symbols dialog box.
7. Click **Folders**, then change to the directory **`\tutor1`** and click **OK**.


Copy Symbols
Copy the selected symbol(s)...
From library: C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor1\sym
To: C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor1
Browse:
Folders...
Libraries...
OK
Cancel
Help

8. Click **OK** to close the Copy Symbols dialog box.
9. Choose **File > Exit** to close the Library Manager.

Lesson 2: Creating a Top-Level Schematic Source

The schematic design entry environment is a set of tools that allows you to capture the structure of a design as either a flat description or a hierarchical set of components, and the connectivity between these components. Then you can use this description to drive the Fitter and verification tools.

You create schematic designs using the Schematic Editor. Schematics can be single-level (flat) or multi-level (hierarchical). Schematics can be drawn on multiple "sheets" and be of any size. The Schematic Editor can work in conjunction with the Hierarchy Navigator, Symbol Editor, and Library Manager programs.

In this lesson you will begin creating a top-level schematic source for the project.

***Note**: If you want to skip this lesson on creating a schematic, you can go directly to Lesson 3.*

Steps covered in this lesson are:

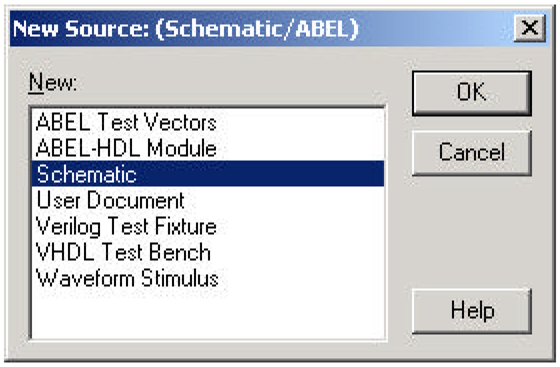
- Task 4: Add a New Schematic to the Project
- Task 5: Resize the Schematic Sheet
- Task 6: Place Two Block Symbols from the Local Symbol Library
- Task 7: Place a Symbol from the REGS Generic Symbol Library
- Task 8: Place Three Symbols from the IOPAD Generic Symbol Library
- Task 9: Add Wires and Buses
- Task 10: Edit the Schematic
- Task 11: Add Wires to Connect the Symbols
- Task 12: Duplicate the Input Pad and Wire Stub
- Task 13: Name the Buses
- Task 14: Add Bus Taps with Signal Names
- Task 15: Add Input Net Names
- Task 16: Add Data Input Net Names
- Task 17: Create Iterated Instances of the Flip-Flop
- Task 18: Add Input Markers

Task 4: Add a New Schematic to the Project

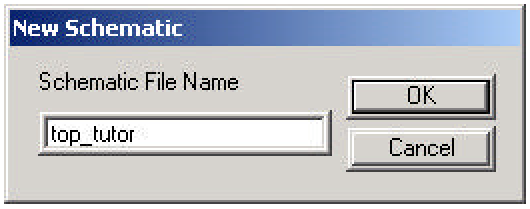
Designing top-down, you'll create the top-level source for the project. Because this design is mixed schematic and ABEL-HDL, you can use either as a source at the top-level design file. However, for this tutorial you'll use a schematic.

To add a new schematic source to the project:

1. In the Project Navigator, choose **Source** > **New** to open the New Source dialog box.



2. Select **Schematic** and click **OK**. The Schematic Editor opens and prompts you to enter a file name for the schematic.



3. Type the name **`top_tutor`** and click **OK**.

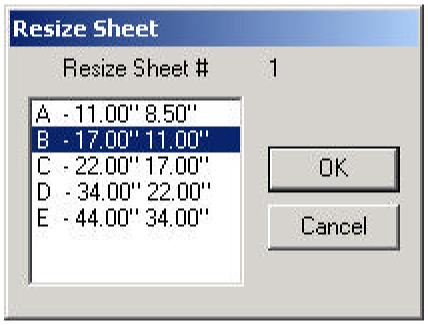
   The Schematic Editor names the current schematic sheet `top_tutor`, and the software imports the schematic into the Project Navigator as a new source file, `top_tutor.sch`.

Task 5: Resize the Schematic Sheet

You can resize a schematic sheet using the **Resize** command. The **Resize** command takes effect immediately and is applied to the sheet selected in the Sheets dialog box, not the active sheet or the sheet currently being worked on.

To resize the schematic sheet:

1. In the Schematic Editor, choose **File** > **Sheets** to open the Sheets dialog box. Because there is only one sheet in the schematic, you cannot select another sheet.
2. Click **Resize** to open the Resize Sheet dialog box. The current size of the selected sheet is highlighted. Other available sheet size choices are listed.



A - 11.00" 8.50"
B - 17.00" 11.00"
C - 22.00" 17.00"
D - 34.00" 22.00"
E - 44.00" 34.00"

3. Select **B**, and then click **OK** to close the Resize Sheet dialog box.
4. Click **Open**.

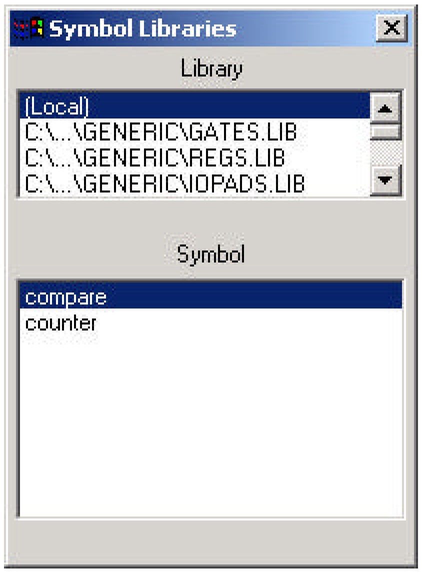
   The software resizes the sheet in the Schematic Editor.

Task 6: Place Two Block Symbols from the Local Symbol Library

The first step in this top-down design is to create block symbols to represent lower-level modules in the design. Later, you will design the I/O ports of the lower-level ABEL-HDL sources to match the names of the pins on the corresponding block symbols.

To place the block symbols in the schematic:

1. In the Schematic Editor, choose **Add** > **Symbol** to open the Symbol Libraries dialog box.
2. In the dialog box, under Library, select **Local**. This library is in the project folder you created. You copied two block symbols into it at the beginning of this tutorial in Task 3.



(Local)
C:\...\GENERIC\GATES.LIB
C:\...\GENERIC\REGS.LIB
C:\...\GENERIC\IOPADS.LIB

compare
counter

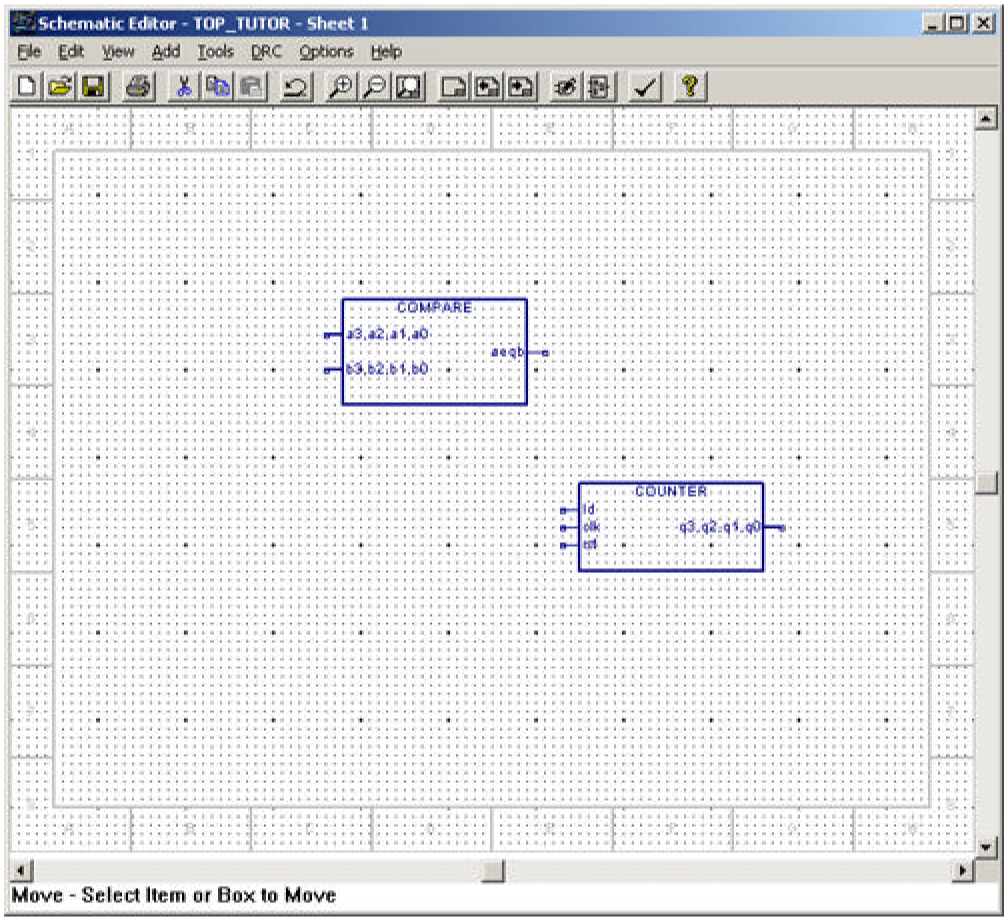
3. Under Symbol, select **compare**. The symbol is attached to the cursor.

Tip*: If you do not see a grid, choose* ***Options > Preferences*** *and select* ***Display Grid****.*

4. Position the cursor slightly above the upper middle area of the schematic, and then click to place the symbol.
5. Right-click to remove the symbol from the cursor.

***Tip**: If you make a mistake placing a symbol, you can choose **Edit > Cut** or **Move**, and then click the symbol that is to receive the action.*

6. Now select the **counter** symbol.
7. Position the cursor below and to the right of the COMPARE symbol, and then click to place the symbol.



Move - Select Item or Box to Move

8. Again, right-click to remove the symbol from the cursor.
9. Leave the dialog box open. You will continue to use it in the next step.

***Note**: It is not critical that your schematic look exactly like the example. Yours may look different depending on unimportant factors that do not affect the proper use of this tutorial.*

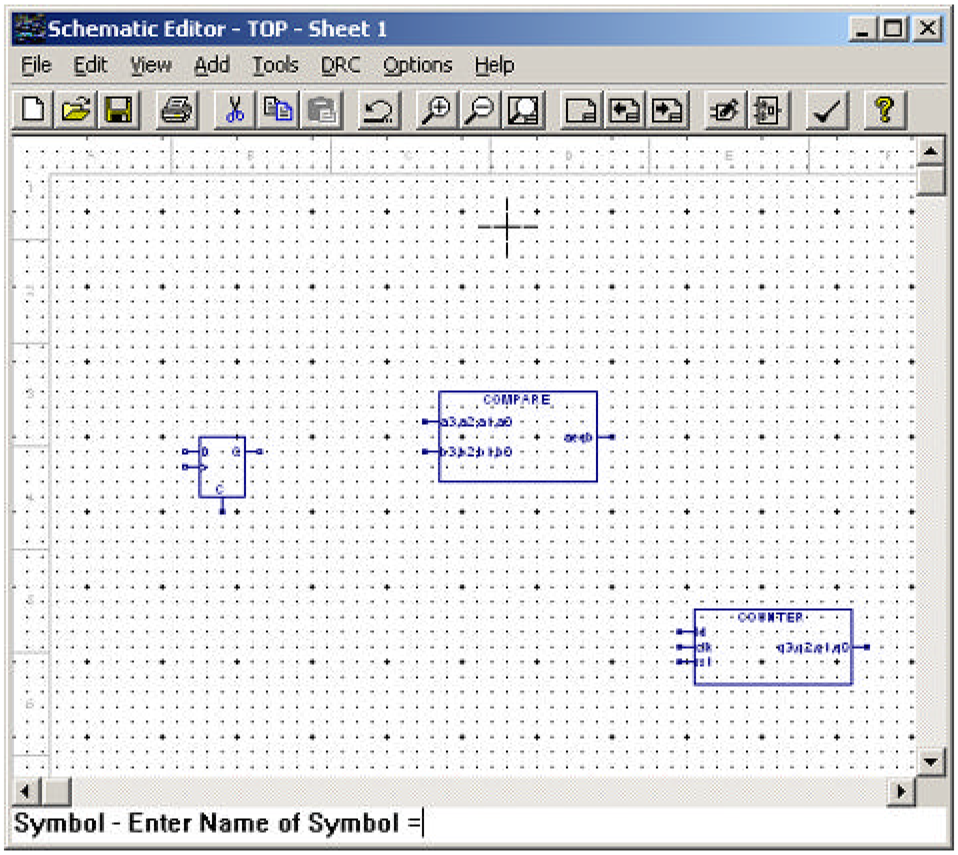
Task 7: Place a Symbol from the REGS Generic Symbol Library

The target device determines which symbol libraries are available. If you use symbols from the Generic Symbol Library, you can migrate designs to different devices without having to redraw the schematic.

***Note**: You can use the Drawing Toolbar to add symbols and other schematic drawing functions. To display the Drawing Toolbar, choose **View > Drawing Toolbar**.*

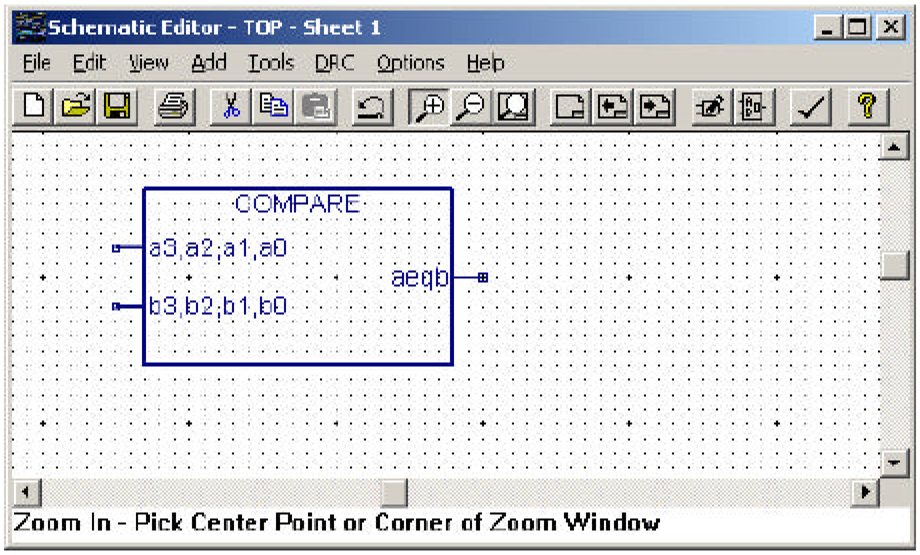
To place a symbol from the REGS generic symbol library:

1. In the Symbol Libraries dialog box, under Library, select **GENERIC\REGS.LIB**.
2. Under Symbol, select **G_DC**.
3. Place the symbol so that the **G_DC** symbol's output is aligned with the **b** input pin on the COMPARE block symbol. (Ignore the title block in the schematic.)


Schematic Editor - TOP - Sheet 1
File Edit View Add Tools DRC Options Help
COMPARE
a3a2a1a0
b3b2b1b0
aeqb
D
Q
C
COUNTER
d
clk
q3q2q1q0

Symbol - Enter Name of Symbol =

4. Choose **View** > **Zoom In**. The cursor changes to a large Z.
5. Click a portion of the schematic to enlarge it to a readable size. Or drag an area to zoom in to a view similar to the following one.


Schematic Editor - TOP - Sheet 1
File Edit View Add Tools DRC Options Help
COMPARE
a3,a2,a1,a0
b3,b2,b1,b0
aeqb

Zoom In - Pick Center Point or Corner of Zoom Window

6. Right-click to cancel the zoom command.

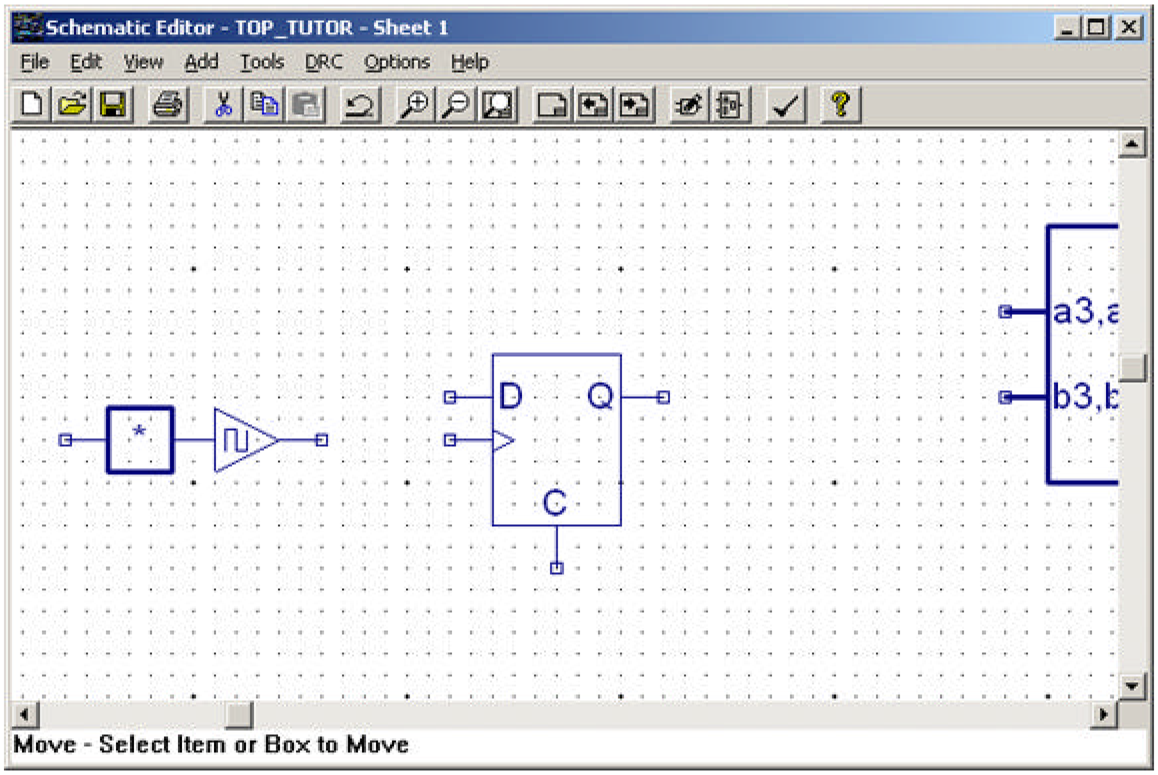
Task 8: Place Symbols from the IOPAD Generic Symbol Library

In this step you will continue adding symbols to the schematic. However, you will use the IOPADS generic symbol library.

Tip*: You can turn grid display off by choosing* ***Options > Preferences****.*

To add a clock buffer symbol:

1. In the Symbol Libraries dialog box, under Library, select **GENERIC\IOPADS.LIB**.
2. Under Symbol, select **G_CLKBUF**.
3. Place the symbol so that the output is aligned with the **clk** input pin on the **G_DC** symbol.


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
D
Q
C
a3,a
b3,b

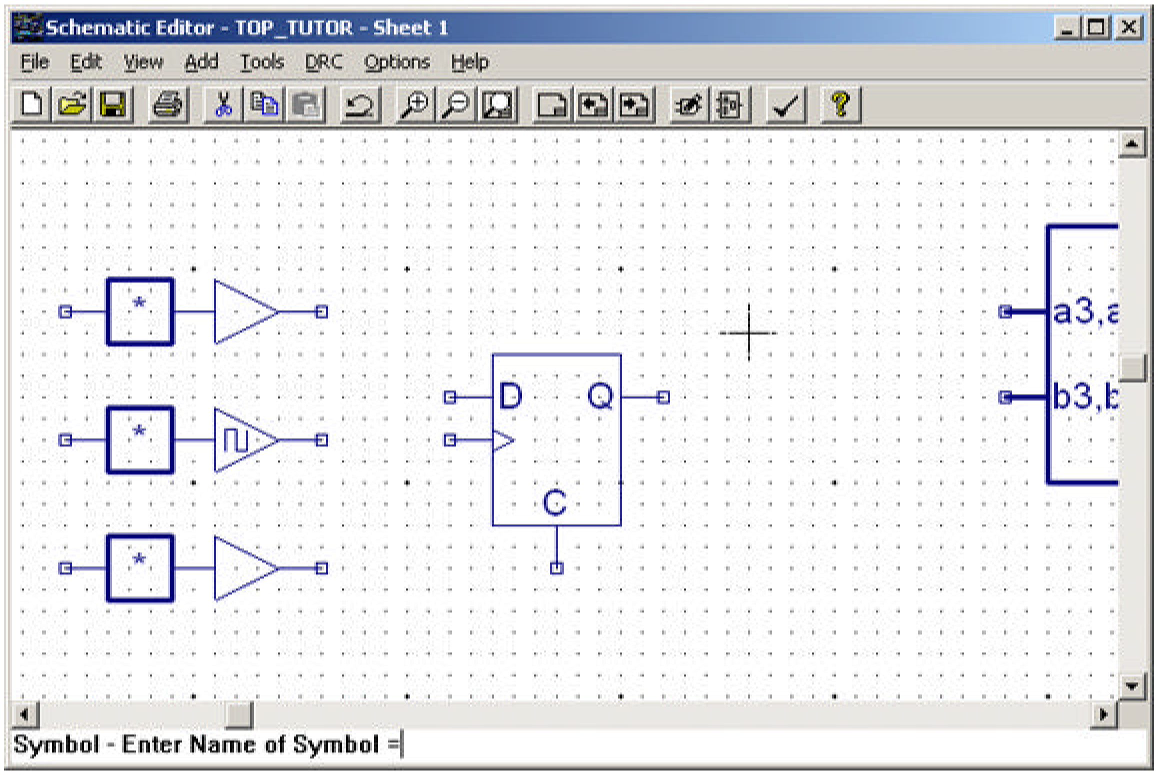
Move - Select Item or Box to Move

***Note**: The spacing of symbols in the schematic is very important. If symbols are placed too close to each other, you will not be able to add wires, bus taps, etc. between the symbols. In the following step, make sure the horizontal distance between the **G_CLKBUF** symbol and **G_DC** symbol is equal to at least the width of the **G_CLKBUF** symbol (see the above picture).*

***Tip**: If you make a mistake placing a symbol, you can choose **Edit > Cut** or **Move**, and then click the symbol that is to receive the action.*

Now you will add two input buffer symbols.

4. Under Symbol, select **G_INPUT**.
5. Click once to place a symbol instance. (Try to align symbols vertically as shown in the picture.)


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
D
Q
C
a3,a
b3,b

Symbol - Enter Name of Symbol =

6. Click once again to place another symbol instance.

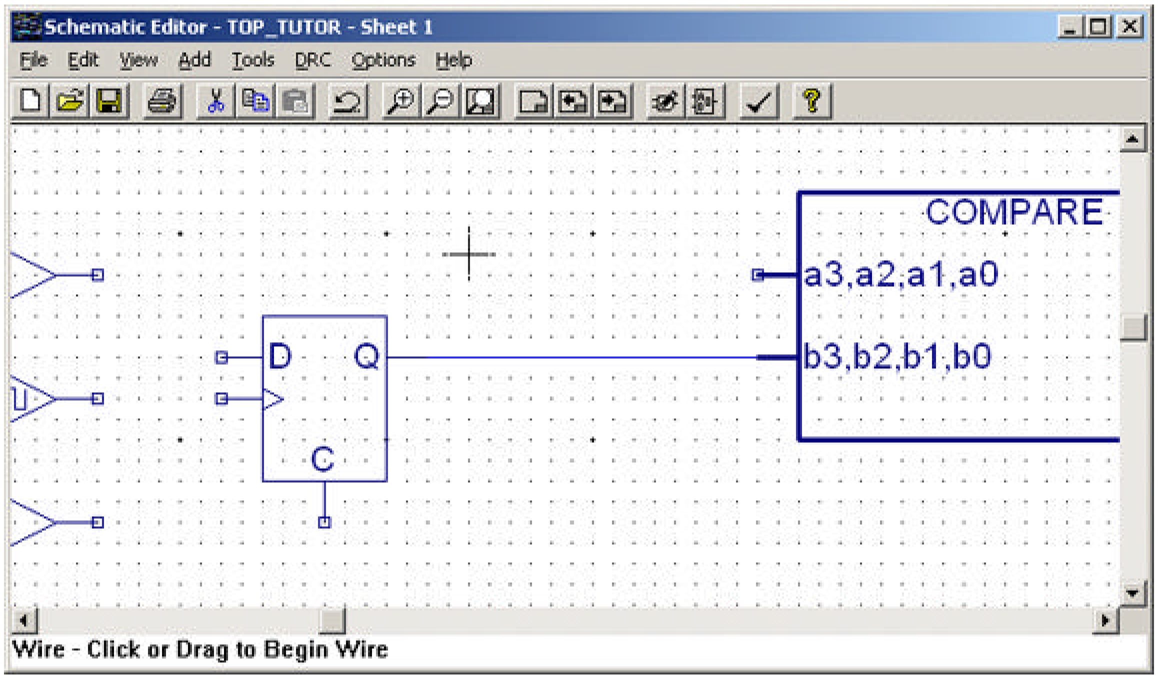
Task 9: Add Wires and Buses

The next step is adding wires and buses to interconnect the symbols. You draw wires and buses the same way; the Schematic Editor knows if a wire is a bus or a single net by the wire's name. In this task, you will draw all of the wires. Later, you will add names to the wires.

***Note**: The **Add Symbol** command from the previous task is canceled when you select the command for this new task.*

To connect the flip-flop output to the COMPARE symbol:

1. Choose Add > Wire.
2. Click the **Q** output of the flip-flop.
3. Drag the wire across to **COMPARE** and click the **b** pin to end the wire segment.


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
COMPARE
a3,a2,a1,a0
b3,b2,b1,b0
D
Q
C
Wire - Click or Drag to Begin Wire

Task 10: Edit the Schematic

It's easy to make changes to the schematic, when you need to erase or reposition any of the wires or symbols.

To erase a symbol or wire:

1. In the Schematic Editor, choose **Edit** > **Delete**.
2. Click one of the input buffer symbols to erase it. That item disappears.
3. Now choose **Edit > Undo**. The buffer symbol reappears.

Note*: If you erase or reposition the wrong item, you can always undo your action by choosing* ***Edit > Undo****.*

To reposition a symbol:

1. Choose Edit > Move.
2. Click the **COMPARE** block. The symbol is attached to the cursor.
3. Move the cursor toward the bottom of the schematic and then click to place the block.
4. Choose **Edit > Undo** to return the block to its previous position.

To reposition several items at once:

1. Choose Edit > Drag.
2. Drag a square around both the COMPARE block and the flip-flop. Both items are attached to the cursor.
3. Move the cursor to the bottom portion of the schematic and click to place the items.
4. Choose **Edit > Undo** to return the items to their previous positions.

***Note**: Almost all commands remain in effect until you select a different command. For example, if the Add Wire command is active, you can continue to draw wires until you select a different command.*

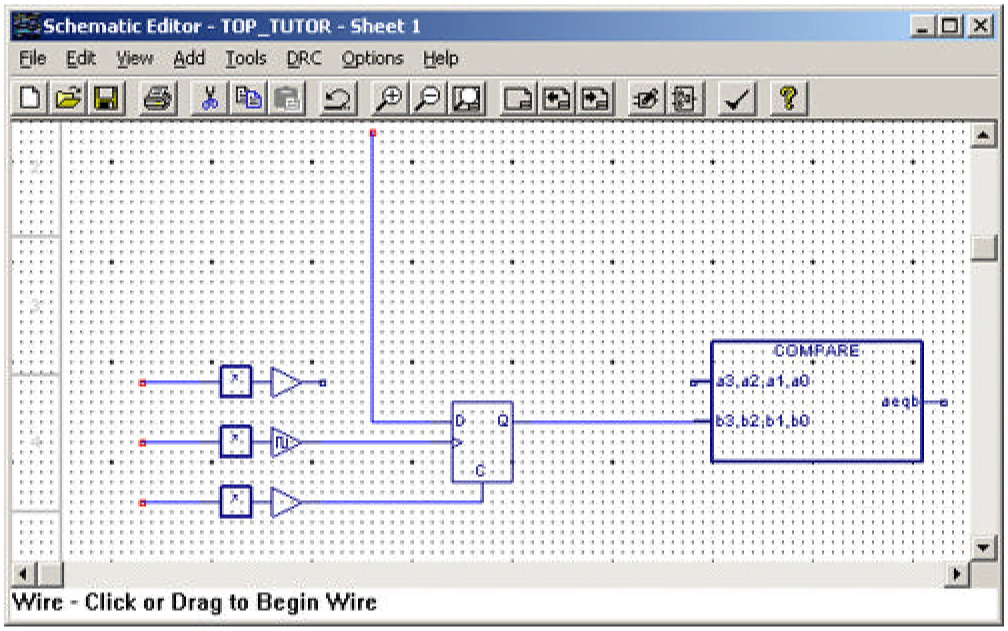
Task 11: Add Wires to Connect the Symbols

In this step, you will add three wires to the flip-flop. Also, you will add wire stubs to the buffers.

The length of the horizontal and vertical segments of the D input wire (Step 7 below) is important. The wire segments must be long enough so that you have room to add a bus name to the horizontal wire segment and add bus taps between the vertical wire segment and input buffers.

To add a wire from D input:

1. Choose Add > Wire.
2. Click the **D input**. Move the cursor to the left to create a horizontal segment with a length equal to about half the horizontal distance between the input buffer and flip-flop. Click once to end the segment.
3. Move the cursor up until the vertical height of the wire is about equal to 3 times the height of the flip-flop.
4. Right-click to end the wire.


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
COMPARE
a3,a2,a1,a0
b3,b2,b1,b0
aeqb
D
Q
C
Wire - Click or Drag to Begin Wire

5. Add a wire from the **C input** to the **input buffer**.
6. Add wire from the **clock input** to the **clock buffer**.
7. Add wire stubs (short wires) to the **input buffers** and the **clock buffer**. (Double-click to end a wire in space, or right-click to cancel the command.)

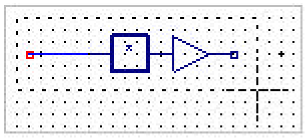
Task 12: Duplicate the Input Pad and Wire Stub

This step shows you how to use the **Duplicate** command to quickly add the rest of the input pads.

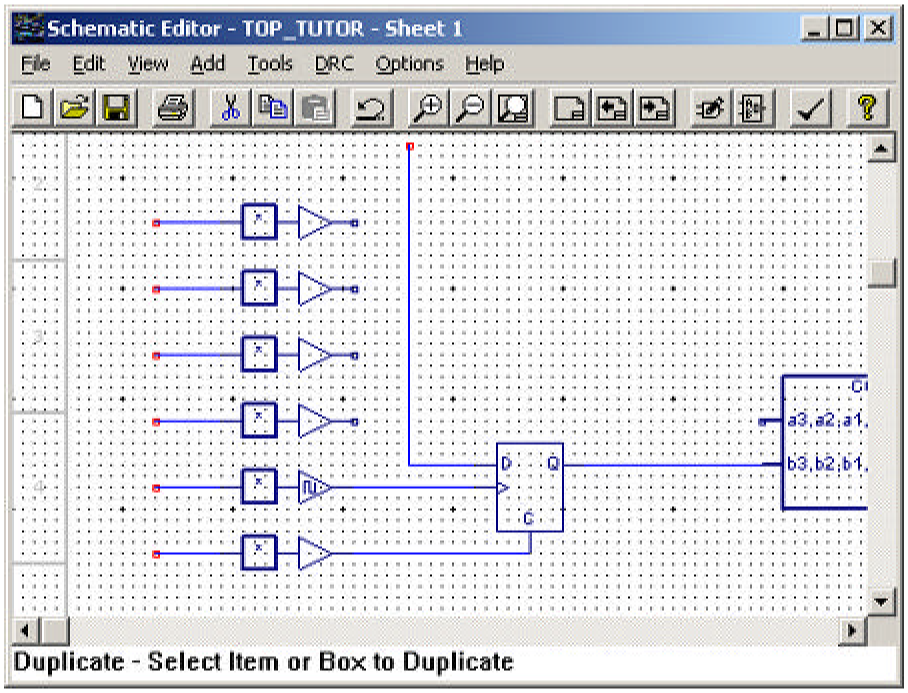
The **Duplicate** command lets you copy one or more elements, and then place them at different locations within the same symbol or schematic. You can place the duplicated item as many times as you want until you select another command. **Duplicate** differs from the **Copy / Paste** command sequence only in that it does not change the contents of the clipboard.

To duplicate the input buffer:

1. Choose Edit > Duplicate.
2. Hold down the mouse button and drag a region around the input buffer and wire stub.



3. Click to place three of the duplicated buffer and wires as shown.


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
D Q
C
a3,a2,a1,
b3,b2,b1,
Duplicate - Select Item or Box to Duplicate

4. Right-click to cancel the command.

Task 13: Name the Buses

Any single- or multi-wire connection between pins is called a network, or net. A bus is a combination of two or more signals into a single wire. Buses are a convenient way to group related signals. This grouping can produce a less cluttered, functionally clearer drawing and clarify the connection between the main circuit and a Block symbol.

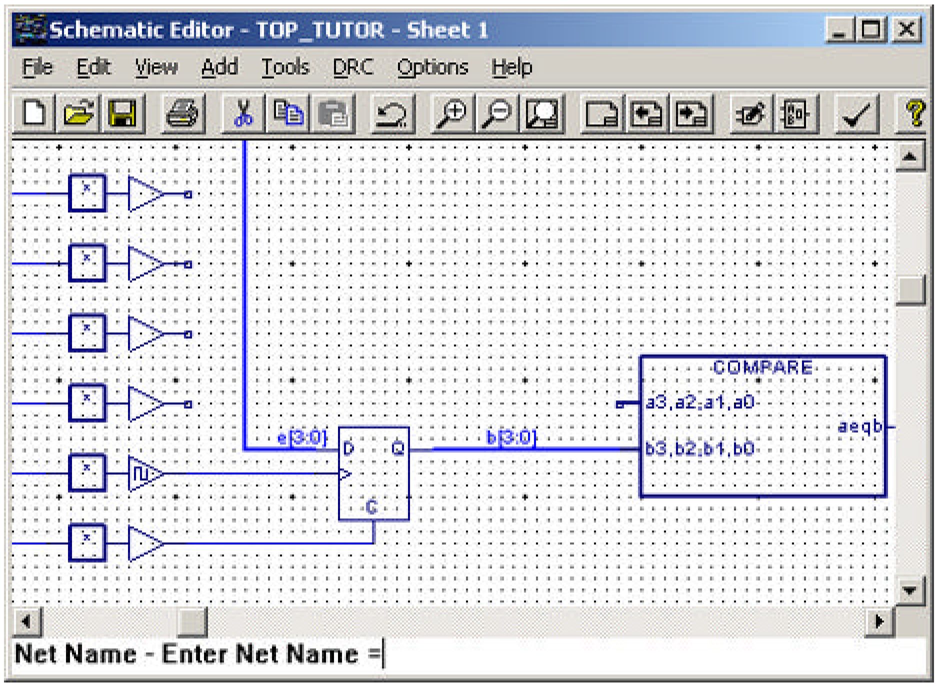
There are two types of buses: *ordered* and *unordered*. An ordered bus has a compound name consisting of the names of the signals that comprise the bus. Any signals can be combined into an ordered bus, whether they are related or not.

A net becomes an ordered bus when it is given a *compound name*. You form a compound name by adding a sequence of numbers to the name. The sequence is specified as a starting number, an ending number, and an optional increment (default = 1). The numbers are positive integers, and are delimited by commas (**,**), dashes (**–**), or colons (**:**). The sequence is enclosed in brackets **[ ]**, parentheses **( )**, or curly braces **{ }**.

In this step, you will name two buses.

To name the buses:

1. Choose Add > Net Name.
2. On the Prompt Line at the bottom of the Schematic Editor, type the name of the bus, **`b[3:0]`**, and then press **Enter**. The name is attached to the cursor.
3. Click the wire between the flip-flop and the COMPARE symbol to place the name on the wire. Notices that the wire thickens once you've named it a bus.


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
COMPARE
a3,a2,a1,a0
b3,b2,b1,b0
aeqb
e[3:0]
b[3:0]
D
Q
C
Net Name - Enter Net Name =

4. Now name another bus by typing on the Prompt Line `e[3:0]`, and then press **Enter**.
5. Click the horizontal segment on the left side of the flip-flop. Again, the wire thickens and now carries the name of the bus.

Task 14: Add Bus Taps with Signal Names

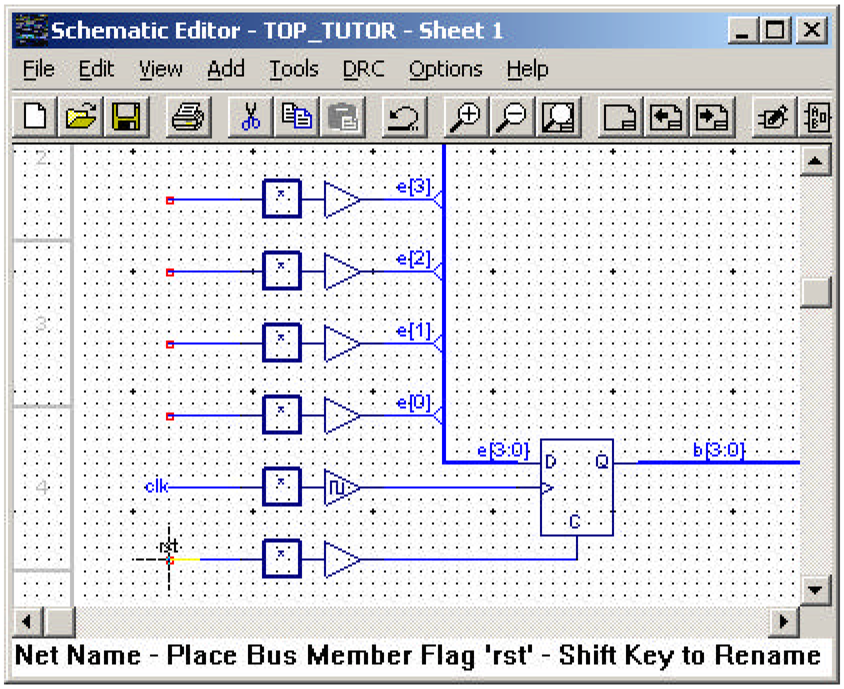
Signals enter and exit a bus at points called *bus taps*. A bus tap can be added to any existing bus, net, or wire. If a net or wire is not already a bus, adding the tap automatically promotes it to a bus.

You can add bus taps only on vertical or horizontal sections of a bus. Tap connections are shown with two diagonal lines, rather than a solder dot.

There are several ways to add bus taps. The following procedure shows you how to create a tap, the connecting wire, and the net name in one simple step using the **Net Name** command.

To add bus taps with signal names:

1. Choose Add > Net Name.
2. Click the bus labeled `e[3:0]`. The cursor picks up the name of the bus.
3. Right-click once to split the bus name into its individual signal names. The signal name `e[3]` is attached to the cursor.
4. In one action, click the pin of the top input pad, drag a wire to the bus, and then release the mouse button. The software adds a bus tap, wire, and signal label. Notice that the signal name decrements.


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
e[3]
e[2]
e[1]
e[0]
e[3:0]
b[3:0]
clk
rst
D
Q
C
Net Name - Place Bus Member Flag 'rst' - Shift Key to Rename

5. Add the remaining taps (in descending order) just by clicking the pins of the buffers.

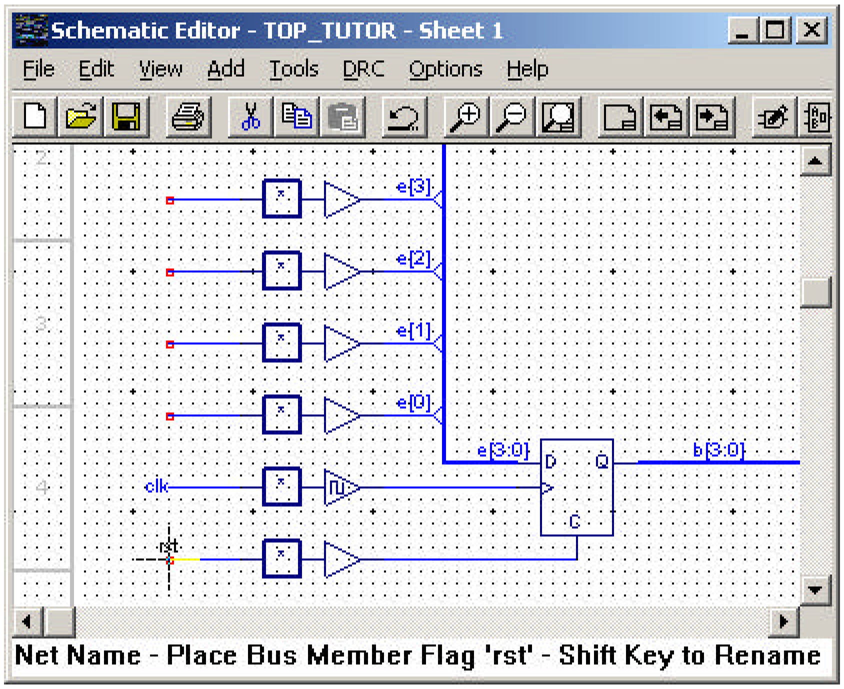
Task 15: Add Input Net Names

Every net has a name, either assigned by you or by the Schematic Editor. You can override any name assigned by the Schematic Editor by assigning one of your own using the **Net Name** command.

You can name nets one at a time. A faster way is to create a compound name (in this example, a group of unique names), and then sequentially attach individual names of a compound name to different nets.

To add a compound net name:

1. Choose Add > Net Name.
2. On the Prompt Line, type: **`clk,rst`**, and then press **Enter**. (Don't forget the comma separating the names.)
3. Notice that both names are attached to the cursor. Right-click once to separate the names. Now only `clk` is showing.
4. Click the end of the input wire for clock buffer. The software adds the first name (`clk`) to the wire. Also notice that the next name in the list (`rst`) appears on the cursor.


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
e[3]
e[2]
e[1]
e[0]
e[3:0]
b[3:0]
clk
rst
D
Q
C
Net Name - Place Bus Member Flag 'rst' - Shift Key to Rename

5. Click the end of the **`rst`** input buffer to add the final name (`rst`) to the wire.

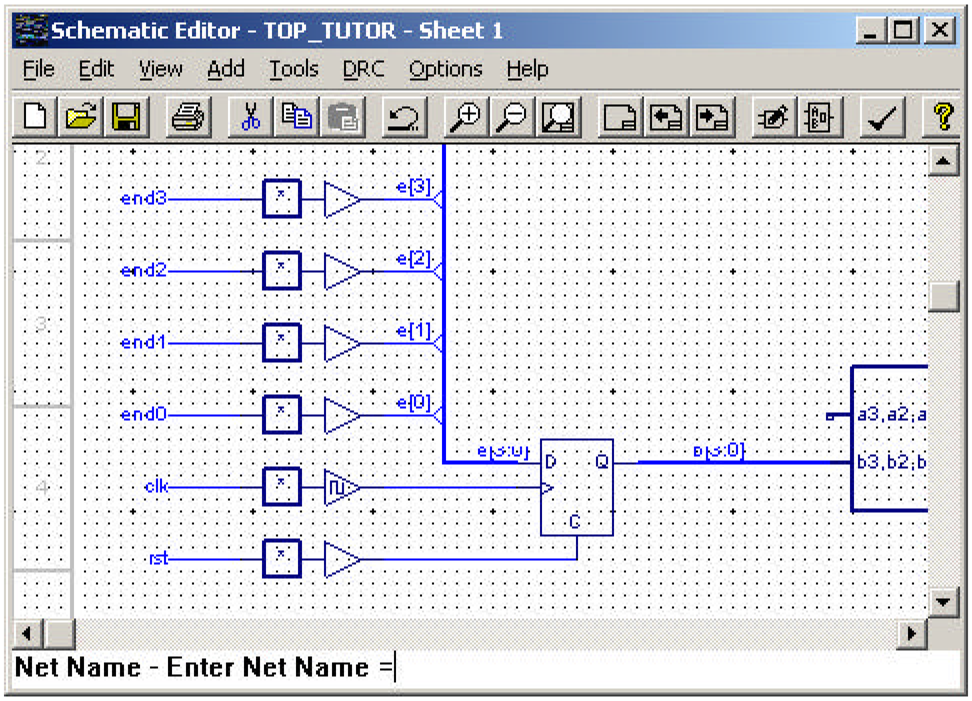
Task 16: Add Data Input Net Names

Another net naming feature makes it easy to deal with buses by automatically incrementing the net name as you place it. In this step, you will add net names to the end segments, and the Schematic Editor will automatically increment the name.

To add sequential net names:

1. Choose Add > Net Name.
2. On the Prompt Line, type: **`end0+`**, and then press **Enter**. The name **`end0`** is attached to the cursor.

   This tells the Schematic Editor to name the signal **`end`**, to start numbering at **`0`**, and to increment (**`+`**) the numbers as the names are placed.
3. Click the bottom data input wire stub. The Schematic Editor places the name **`end0`** and automatically increments the net name.


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
end3
end2
end1
end0
clk
rst
e[3]
e[2]
e[1]
e[0]
e[3:0]
b[3:0]
D
Q
C
a3,a2,a
b3,b2,b

Net Name - Enter Net Name =

4. Click the next wire stub. Repeat this step until all data input wire stubs have net names.
5. Right-click when **`end4`** appears on the cursor to cancel the command.

Task 17: Create Iterated Instances of the Flip-Flop

A powerful feature in the schematic is its capability of using an iterated instance. Iterated instances allow a single symbol to represent multiple instances connected in parallel.

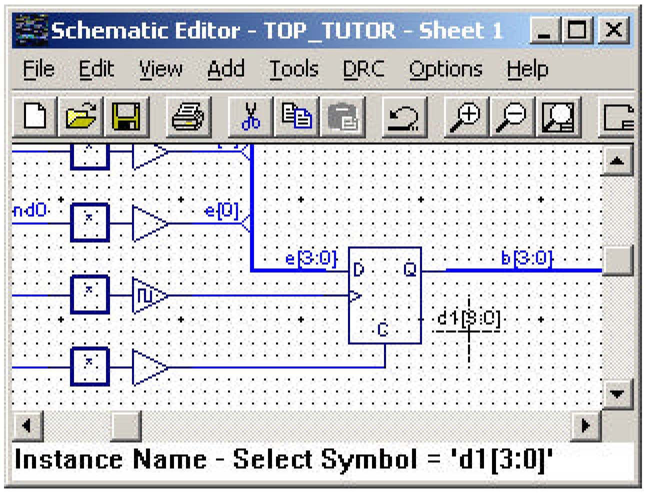
You can convert a single instance into an iterated instance by giving it a compound instance name of the form:

```
INV[3-10]
```

In this case, eight instances of the symbol you've named INV are created, but the symbol appears only once in the schematic.

To create iterated instances of the flip-flop:

1. Choose Add > Instance Name.
2. In the Prompt Line, type: **`d1[3:0]`**, and then press **Enter**. The name is attached to the cursor.


Schematic Editor - TOP_TUTOR - Sheet 1
File Edit View Add Tools DRC Options Help
e[3:0]
b[3:0]
d1[3:0]

Instance Name - Select Symbol = 'd1[3:0]'

3. Click the flip-flop once. The Schematic Editor places the label on the symbol.

   The flip-flop is now really four flip-flops, with signal `e[3]` feeding `d[3]`, `e[2]` feeding `d[2]`, etc. The common signals are connected in parallel to the common pins (such as `clk` and `rst`).

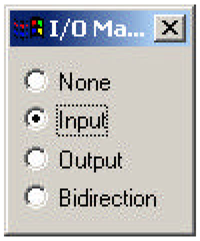
Task 18: Add Input Markers

An I/O marker is a special indicator that identifies a net name as a device input, output, or bidirectional signal. This establishes net polarity (direction of signal flow) and indicates that the net is externally accessible.

The Schematic Editor **Consistency Check** command uses I/O markers to flag any discrepancies in the polarity of marked signals and the symbol pins. Discrepancies in polarity are also flagged each time you run the Hierarchy Navigator.

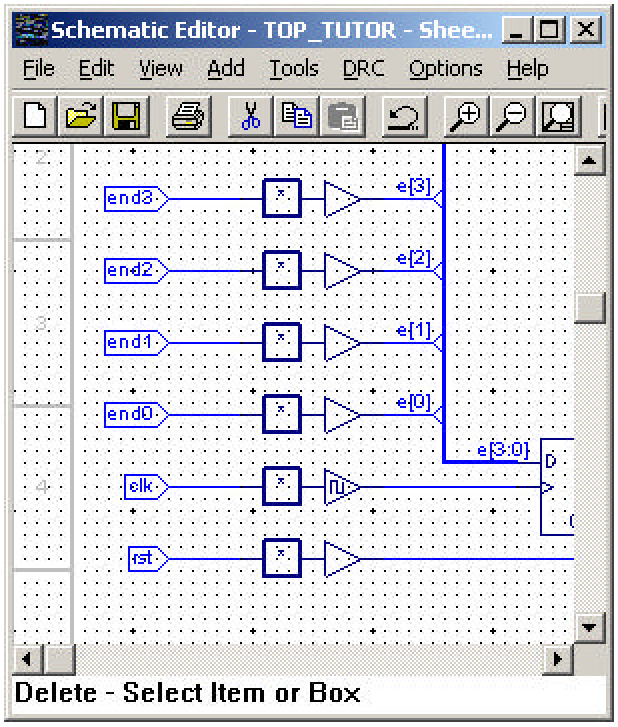
To add input markers:

1. Choose **Add** > **I/O Marker** to open the dialog box.
2. Select **Input**.



None
[✓ Input]
Output
Bidirection

3. You can add a marker by clicking at the point where the I/O marker touches the end of a horizontal or vertical wire segment or bus. However, if you have a group of nets that you want to add markers to, there is a faster way. Select all of the input wires at once by dragging a region around them. The Schematic Editor adds markers to all the nets at once.


Schematic Editor - TOP_TUTOR - Shee...
File Edit View Add Tools DRC Options Help
end3
end2
end1
end0
clk
rst
e[3]
e[2]
e[1]
e[0]
e[3:0]
D
Delete - Select Item or Box

4. Close the I/O Marker dialog box.
5. Choose **File** > **Save** to save the schematic.
6. Close the Schematic Editor.

Note*: To remove an I/O marker, select* ***None****.*

Lesson 3: Finishing the Schematic

To save time, we have completed the schematic for you.

In this lesson you will remove the schematic source you were building and import a completed source. You will learn how to automatically create a symbol for the currently loaded schematic. You will also learn how to check your schematic for design rule violations.

Steps covered in this lesson are:

- Task 19: Import the Completed Schematic
- Task 20: Create a Matching Symbol
- Task 21: Check the Schematic for Consistency Errors

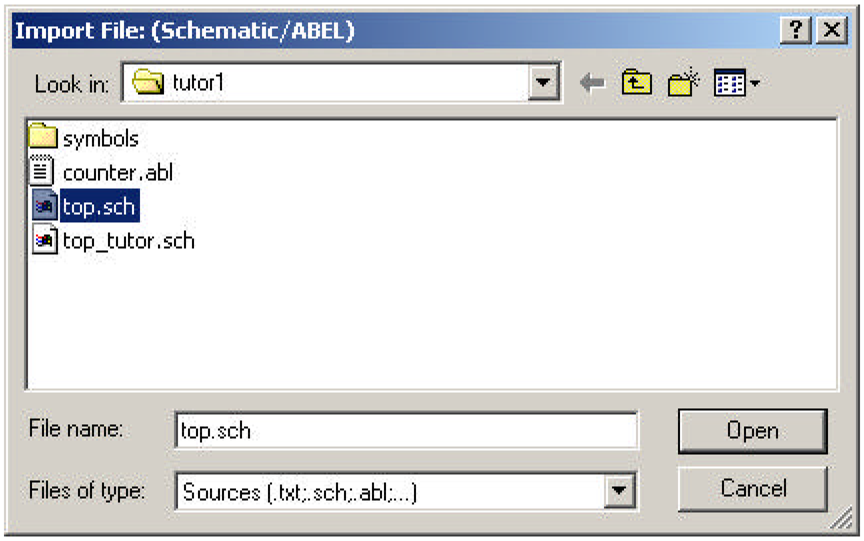
Task 19: Import the Completed Schematic

In this step, you will remove the schematic source you have been working on and import the completed schematic of the same name.

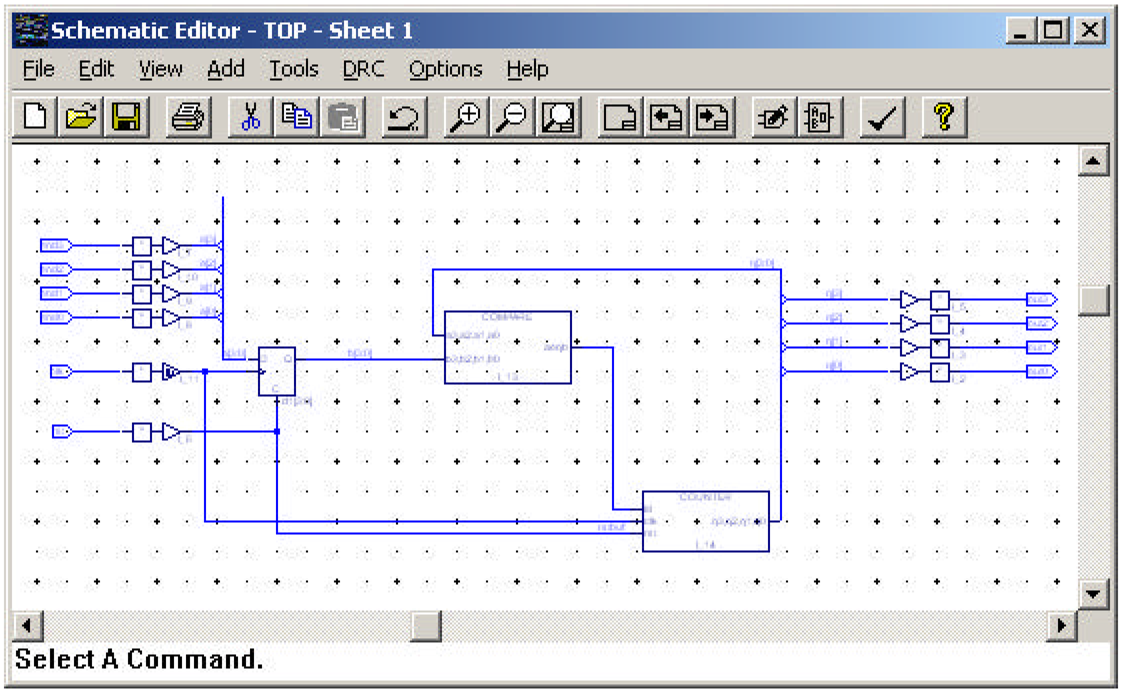
***Note**: If you skipped Lesson 2, you do not have to complete Steps 1 and 2.*

To import the completed schematic:

1. In the Sources window of the Project Navigator, select the **top_tutor** schematic source.
2. Choose **Source** > **Remove** to delete the source from the project.
3. Choose **Source** > **Import** to open the Import File dialog box.



4. Select **`top.sch`** and click **Open**.
5. Double-click the **top** source to open the schematic in the Schematic Editor.


Schematic Editor - TOP - Sheet 1
File Edit View Add Tools DRC Options Help
Select A Command.

The Schematic Editor opens, showing the completed schematic.

Task 20: Create a Matching Symbol

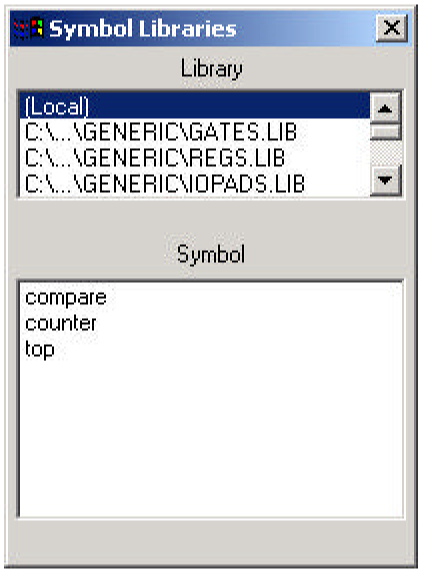
You can use the **Matching Symbol** command to create a symbol file (`*.sym`) for the schematic currently loaded, with the same base name. The input and output pins on the symbol have the same signal names and polarities as the I/O markers in the schematic.

The Schematic Editor creates the symbol in the same directory as the schematic. You can use the **Add Symbol** command to insert the symbol into any other schematic.

In this step, you will create a symbol for the **top** schematic. The symbol will be saved in the Local symbol library in the project directory.

To create a matching symbol for the Top schematic:

1. Choose **File** > **Matching Symbol**. The Schematic Editor automatically creates a symbol.
2. Choose **Add** > **Symbol** to open the Symbol Libraries dialog box.
3. Under Library, scroll to the top and select **(Local)**. Notice, without selecting it, the symbol is named **top**.



(Local)
C:\...\GENERIC\GATES.LIB
C:\...\GENERIC\REGS.LIB
C:\...\GENERIC\IOPADS.LIB

compare
counter
top

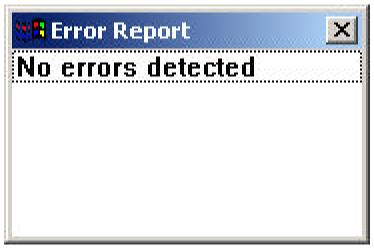
4. Close the Symbol Libraries dialog box.

Task 21: Check the Schematic for Consistency Errors

The Schematic Editor continually checks for errors, such as closed loops and shorted nets, while you're drawing your schematic. You can also check your schematic for other errors such as unconnected wires or pins or an unnamed signal tapped from a bus. Errors found are shown in a "hot" list box. Clicking an error causes the cursor to jump to the offending location.

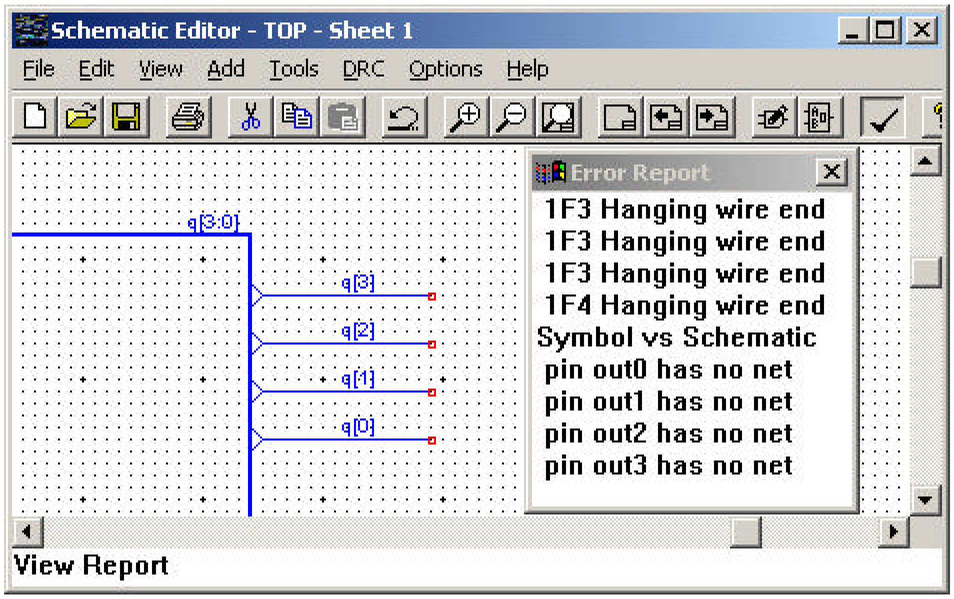
To check for consistency errors:

1. Choose **DRC > Consistency Check** to open the Error Report. There should be no errors in the report.



No errors detected

2. Close the Error Report.
3. Choose Edit > Delete.
4. Drag the area around the circuitry on the four gates on the far right side of the schematic, and then release the mouse button.
5. Choose DRC > Consistency Check again.


Schematic Editor - TOP - Sheet 1
File Edit View Add Tools DRC Options Help
q[3:0]
q[3]
q[2]
q[1]
q[0]
Error Report
View Report

1F3 Hanging wire end
1F3 Hanging wire end
1F3 Hanging wire end
1F4 Hanging wire end
Symbol vs Schematic
pin out0 has no net
pin out1 has no net
pin out2 has no net
pin out3 has no net

6. In the Error Report, select an error. Notice that the schematic view shifts to the location of the selected error.
7. Close the Error Report.
8. Choose **File > Exit** to exit the schematic. When asked to save your changes, click **NO**.

Congratulations! You have just created and checked a top-level schematic. Now let’s create an ABEL-HDL source and add it to the project.

Lesson 4: Adding ABEL-HDL Sources to the Project

The Project Navigator now lists the schematic, top, and the two block symbols referenced in the schematic, counter and compare. These sources have the undefined icon next to them because they do not exist yet. Remember that this is a top-down design example.

This lesson leads you through the steps necessary to add two lower-level ABEL-HDL modules to the project. One you will create from scratch. The other you will import.

Steps covered in this lesson are:

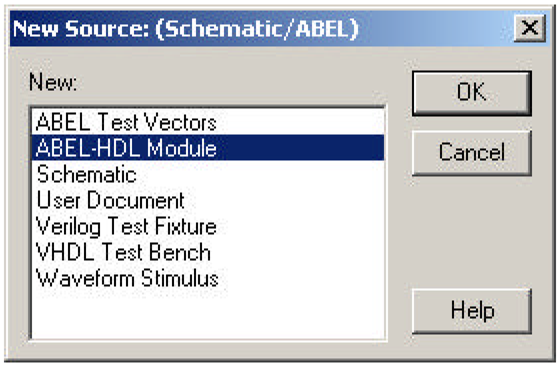
- Task 22: Create a New ABEL-HDL Source File Template
- Task 23: Enter the ABEL-HDL Source Description
- Task 24: Import an Existing ABEL-HDL Source File

Task 22: Create a New ABEL-HDL Source File Template

There are two steps to creating a new ABEL-HDL source file. First, you create the template. Then you enter the source description. In this task, you will create a template for the compare ABEL-HDL module.

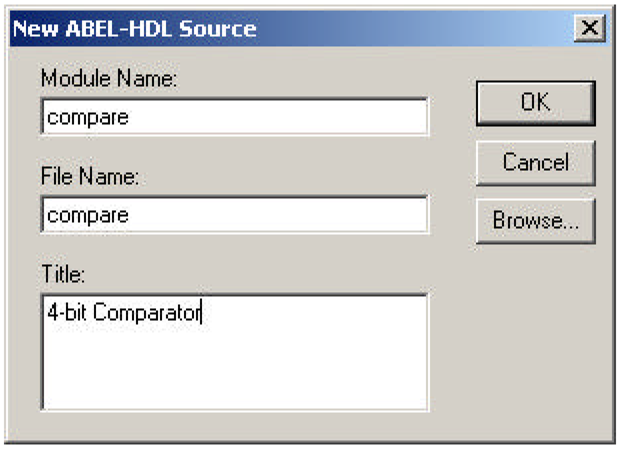
To create an ABEL-HDL template:

1. In the Project Navigator Sources window, double-click **compare** to open the New Source dialog box.
2. Select **ABEL-HDL Module** and click **OK**.



The Text Editor opens, plus the **New ABEL-HDL Source** dialog box that prompts you for module information.

3. In the dialog box, type the information shown below.



Module Name: compare

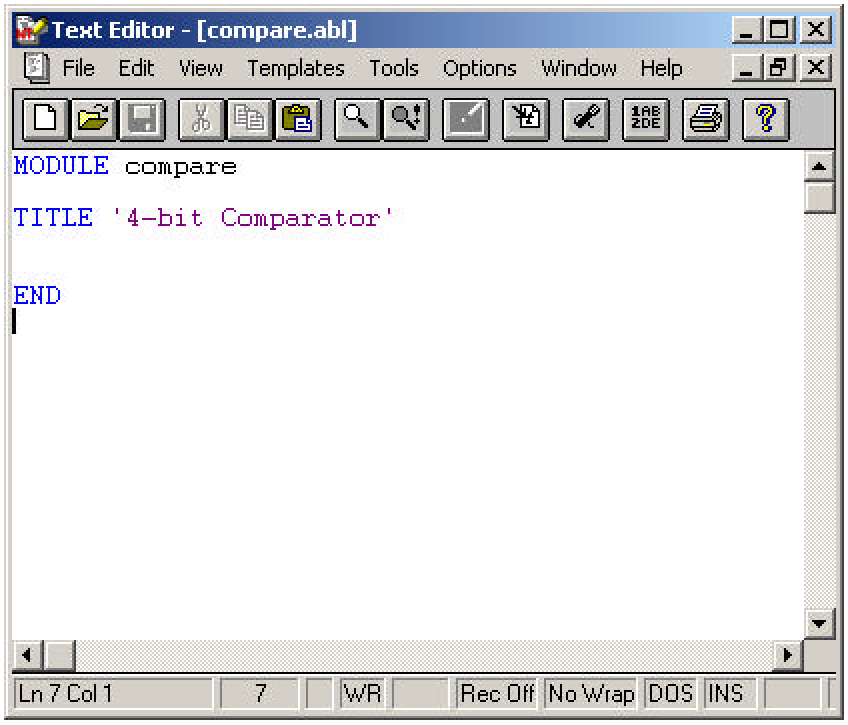
File Name: compare

Title: 4-bit Comparator

The text is case sensitive. Make sure that you enter the text as shown. Note that the Module Name matches the corresponding block symbol name. The module and file names don't have to be the same, but it makes things simpler.

4. Click **OK** to close the dialog box.

   The Text Editor appears with an ABEL-HDL template. The template shows the module and file names you entered.


Text Editor - [compare.abl]
File Edit View Templates Tools Options Window Help

```
MODULE compare

TITLE '4-bit Comparator'


END
```

Ln 7 Col 1 | 7 | WR | Rec Off | No Wrap | DOS | INS

***Note**: In the Project Navigator Sources window there are red check marks beside the sub-module names because they have not yet been defined.*

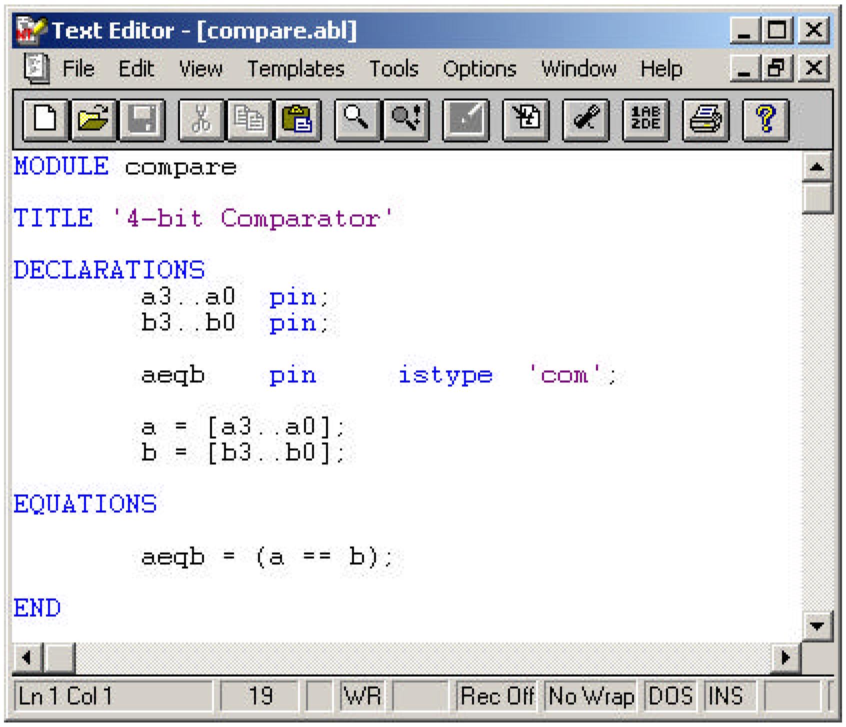
Task 23: Enter the ABEL-HDL Source Description

In this step, you will complete the ABEL-HDL module by entering its description.

To describe an ABEL-HDL module:

1. In the Text Editor, type the text description as shown in the picture below.

   Notice that the ABEL-HDL pin declarations must match the name and the case of the I/O pins on the corresponding block symbol.


Text Editor - [compare.abl]
File Edit View Templates Tools Options Window Help

```
MODULE compare

TITLE '4-bit Comparator'

DECLARATIONS
        a3..a0  pin;
        b3..b0  pin;

        aeqb    pin     istype  'com';

        a = [a3..a0];
        b = [b3..b0];

EQUATIONS

        aeqb = (a == b);

END
```

```
Ln 1 Col 1   19   WR   Rec Off  No Wrap  DOS  INS
```

2. When you are through, choose **File > Save As** to save the file. Save the file with the same name and click **OK**.

3. Close the Text Editor.

   In the Project Navigator Sources window, notice that the icon next to the compare file has changed, and that the file name (`compare.abl`) appears next to the name. This indicates that the file is now a *defined* module.

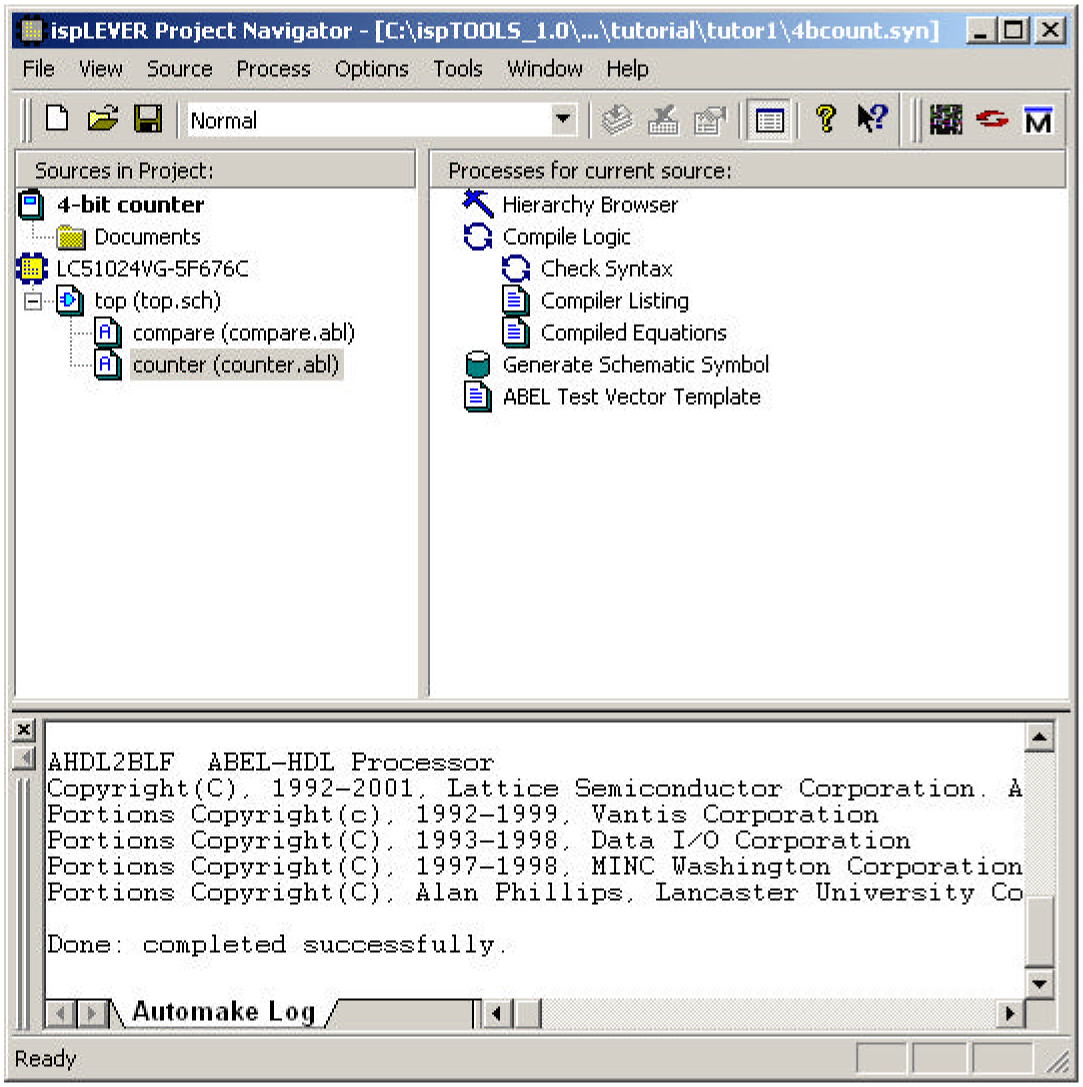
Task 24: Import an Existing ABEL-HDL Source File

As you continue working with the ispLEVER software, you will want to import existing ABEL-HDL source files, in addition to creating new ones. To save time in this tutorial, you will import the remaining ABEL-HDL source file.

To import an existing ABEL-HDL source:

1. In the Project Navigator, choose **Source** > **Import** to open the Import File dialog box.
2. Select **`counter.abl`** and click **Open**.

   The Project Navigator should now look like the image shown below. You'll take a look at the "insides" of this module in the following lesson.



```
AHDL2BLF  ABEL-HDL Processor
Copyright(C), 1992-2001, Lattice Semiconductor Corporation. A
Portions Copyright(c), 1992-1999, Vantis Corporation
Portions Copyright(C), 1993-1998, Data I/O Corporation
Portions Copyright(C), 1997-1998, MINC Washington Corporation
Portions Copyright(C), Alan Phillips, Lancaster University Co

Done: completed successfully.
```

Lesson 5: Navigating the Design

The Hierarchy Navigator program allows you to navigate through a schematic design that consists of a top-level schematic and lower-level schematics and HDL modules. The Hierarchy Navigator loads a full hierarchical design all at once so that you can view it in its complete form, rather than as individual sources. Every schematic sheet and behavioral file at all levels of hierarchy is included.

This lesson shows you how to use the Hierarchy Navigator to perform several useful functions.

Steps covered in this lesson are:

- Task 25: Open the Hierarchy Navigator
- Task 26: Push into the COUNTER Block Symbol
- Task 27: Access Connectivity Information for the COUNTER Block
- Task 28: Query a Net

Task 25: Open the Hierarchy Navigator

You can open the Hierarchy Navigator from within the Project Navigator. The Hierarchy Navigator performs several important functions:

- It verifies the correctness and consistency of a design's wiring. Verification occurs at each level in the design, and across all the levels, from top to bottom.
- It provides the environment in which you can analyze and optimize the circuit's performance.
- It prepares the design data for later steps in the design process, for example creating netlists.

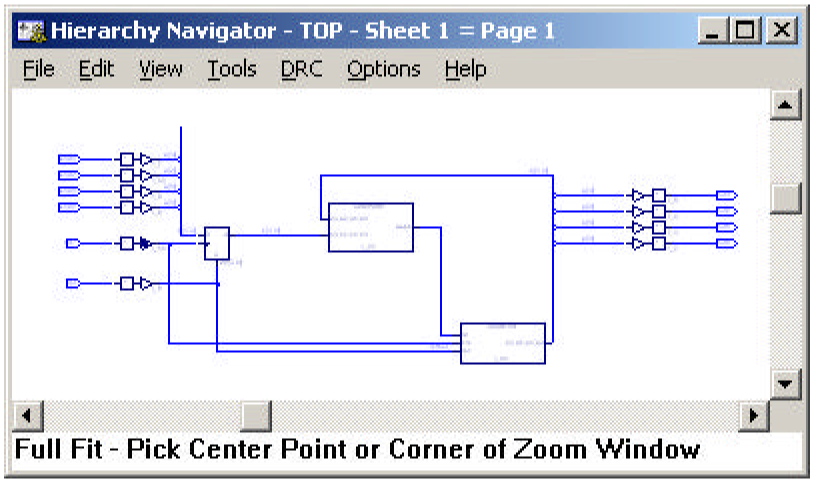
To open the Hierarchy Navigator:

1. In the Sources window, select the **top** schematic source (`top.sch`).

   Notice that the Navigate Hierarchy process is displayed at the top of the Processes window. Also notice that Navigate Hierarchy is visible *only* when a schematic source is selected.

2. In the Processes window, double-click the **Navigate Hierarchy** icon.

   The Hierarchy Navigator opens with sheet 1 of the selected schematic source loaded.


Hierarchy Navigator - TOP - Sheet 1 = Page 1
File Edit View Tools DRC Options Help
Full Fit - Pick Center Point or Corner of Zoom Window

Note*: Remember, this is not the Schematic Editor. You cannot edit the schematic or one of its symbols in the Hierarchy Navigator. However, you can open editors from the Hierarchy Navigator to make changes in a specific schematic element.*

Task 26: Push into the Counter Block Symbol

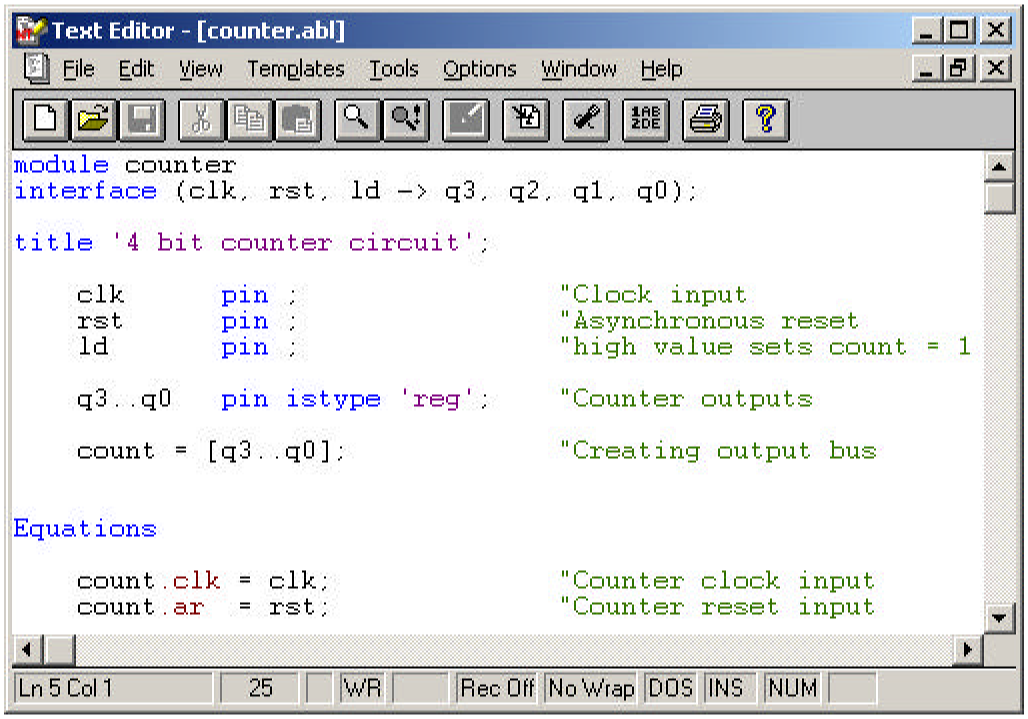
You can use the **Push/Pop** command on the View menu to move down and up (respectively) through the hierarchical levels of a design. This command works on both schematics and ABEL-HDL modules.

You may want to use the **Zoom In** command to view the COUNTER module before starting this step. Right-click to cancel the zoom command.

To push into the COUNTER block symbol:

1. Choose **View** > **Push/Pop**. The cursor changes to a cross ha ir.
2. To move down a level, into the COUNTER module, click *inside* the **COUNTER** schematic symbol.

   The Text Editor opens with the ABEL-HDL description of the module.



Text Editor - [counter.abl]
File Edit View Templates Tools Options Window Help

```
module counter
interface (clk, rst, ld -> q3, q2, q1, q0);

title '4 bit counter circuit';

    clk         pin ;               "Clock input
    rst         pin ;               "Asynchronous reset
    ld          pin ;               "high value sets count = 1

    q3..q0      pin istype 'reg';   "Counter outputs

    count = [q3..q0];               "Creating output bus


Equations

    count.clk = clk;                "Counter clock input
    count.ar  = rst;                "Counter reset input
```

Ln 5 Col 1
25
WR
Rec Off
No Wrap
DOS
INS
NUM

3. View the contents of the COUNTER module. When you finish, close the Text Editor, keeping the Hierarchy Navigator open.

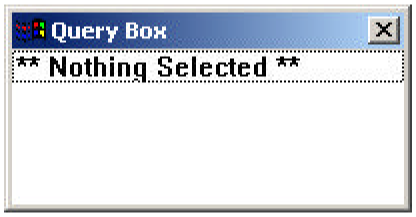
Task 27: Access Connectivity Information for the Counter Block

You can use the **Query** command to display additional information about circuit elements. The information appears in a text box that pops up when the first element is selected. The box is updated when another element is selected.

In this step, you will use the **Query** command to query the Counter ABEL-HDL module for information.

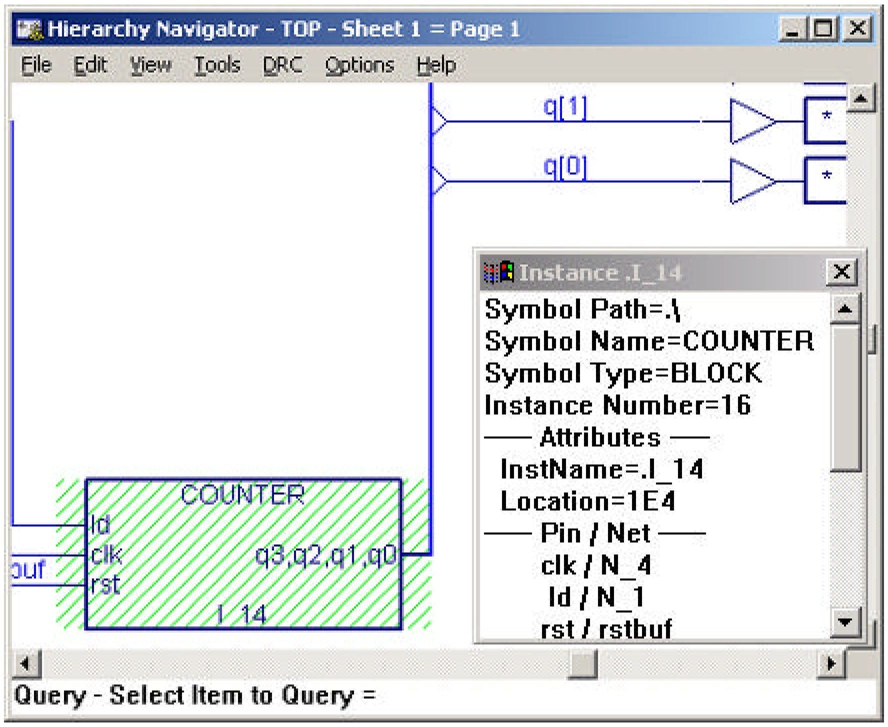
To query the COUNTER block symbol:

1. In the Hierarchy Navigator, choose **DRC** > **Query**. The Query text box opens with the message "Nothing Selected." Move the Query Box so that you can see the COUNTER block.



** Nothing Selected **

2. Click the **COUNTER** block. The Query Box is no longer titled "Query." Instead, it now reflects the name of the instance you've selected and shows various items of information about the COUNTER block.


Hierarchy Navigator - TOP - Sheet 1 = Page 1
File Edit View Tools DRC Options Help
q[1]
q[0]
COUNTER
ld
clk
rst
q3,q2,q1,q0
I_14
buf
Instance .I_14

Symbol Path=.\
Symbol Name=COUNTER
Symbol Type=BLOCK
Instance Number=16
—— Attributes ——
InstName=.I_14
Location=1E4
—— Pin / Net ——
clk / N_4
ld / N_1
rst / rstbuf

Query - Select Item to Query =

3. Look at the **Pin/Net** section.

   The Hierarchy Navigator knows which nets are connected to which pins, and displays instance names, reference designators, or symbol names that have been assigned to that symbol.

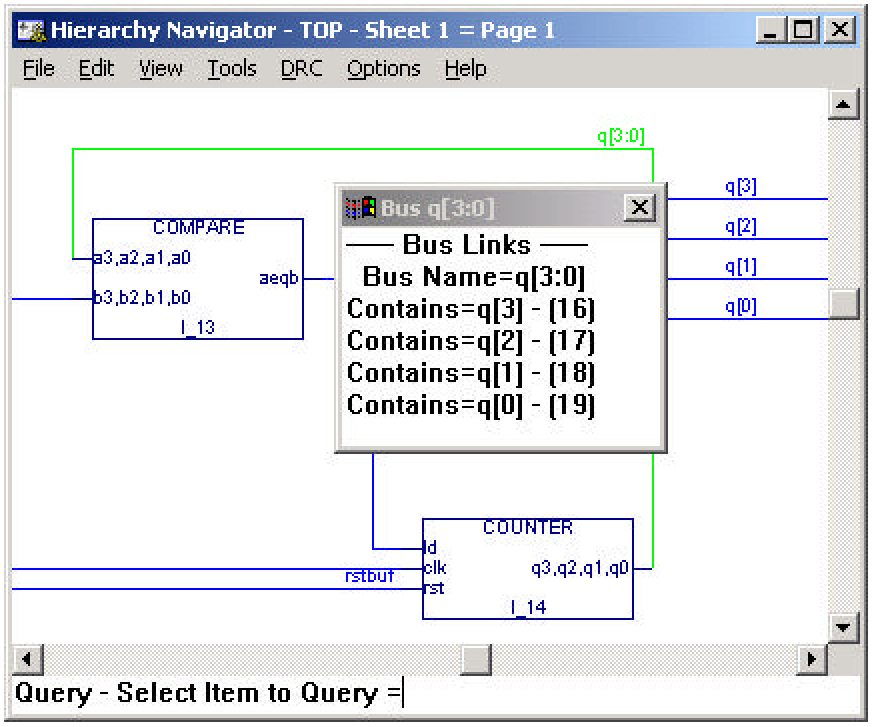
4. Keep the Query Box open for the next task.

Task 28: Query a Net

Although design problems are usually observed at the top level, the source of the problems is often at a lower level. Tracing signals from the primary outputs down through the hierarchy can greatly aid debugging. For instance, to determine if a net needs more buffers, use the **Query** command to determine what components are attached to the net and what will be affected by changes to the net.

To query a net:

1. With the **Query** command still active, click the bus labeled **`q[3:0]`** in the schematic. A list of its content signals appears in the Query Box.



—— Bus Links ——
Bus Name=q[3:0]
Contains=q[3] - (16)
Contains=q[2] - (17)
Contains=q[1] - (18)
Contains=q[0] - (19)

Query - Select Item to Query =

Observe that the signals displayed are links (pointers) to the individual nets.

2. In the Hierarchy Navigator, click **`q[1]`**. The Query Box shows new information for the net. (You will not see information about the Bus and Net at the same time. When one window opens, the other window closes.)

3. In the Query Box, click one of the connections to a net. The cursor automatically moves to the component where the connecting pin is located. The cursor automatically moves to any page in the hierarchy, no matter which level, if necessary to display the selected pin.

4. Close the Query Box and the Hierarchy Navigator. If prompted **Data has been modified**, click **NO**.

Lesson 6: Running Functional Simulation

Functional simulation is the process of simulating the functionality of your RTL design before it is compiled, thus letting you find and correct basic design errors sooner. While functional simulation will verify your Boolean equatio ns, it does not indicate timing problems.

The ispLEVER software supports functional simulation for any Lattice Semiconductor device using the Lattice Logic Simulator or ModelSim™ from Model Technology. These simulators operate in both stand-alone and integrated environments.

This lesson shows you how to use the Lattice Logic Simulator to run integrated functional simulation.

Steps covered in this lesson are:

- Task 29: Import the Test Vector File
- Task 39: Run Functional Simulation

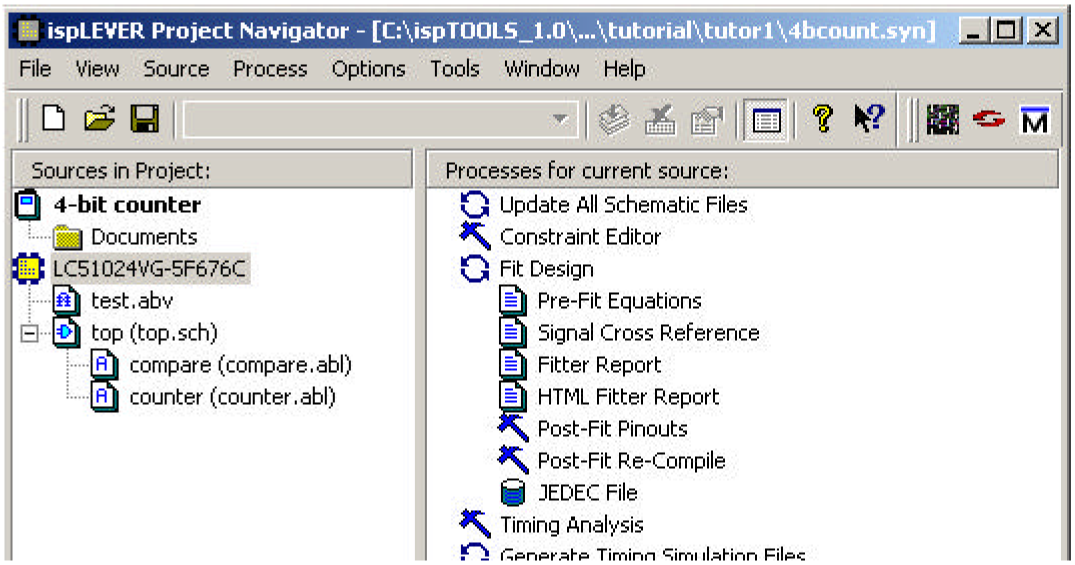
Task 29: Import the Test Vector File

You can create a test vector file in a text editor using proper keywords. Test vectors are sets of input stimulus values and corresponding expected outputs that can be used with both functional and timing simulators. Test vectors can be specified either in a top-level ABEL-HDL source or in a separate ABEL-HDL test vector format file called an `.abv` file. The `.abv` file is considered a text document and is kept above the device level in the Sources window. Whether the test vectors are part of a top-level ABEL-HDL source (`.abl`) or are in a separate file, they will be compiled and passed to the simulator.

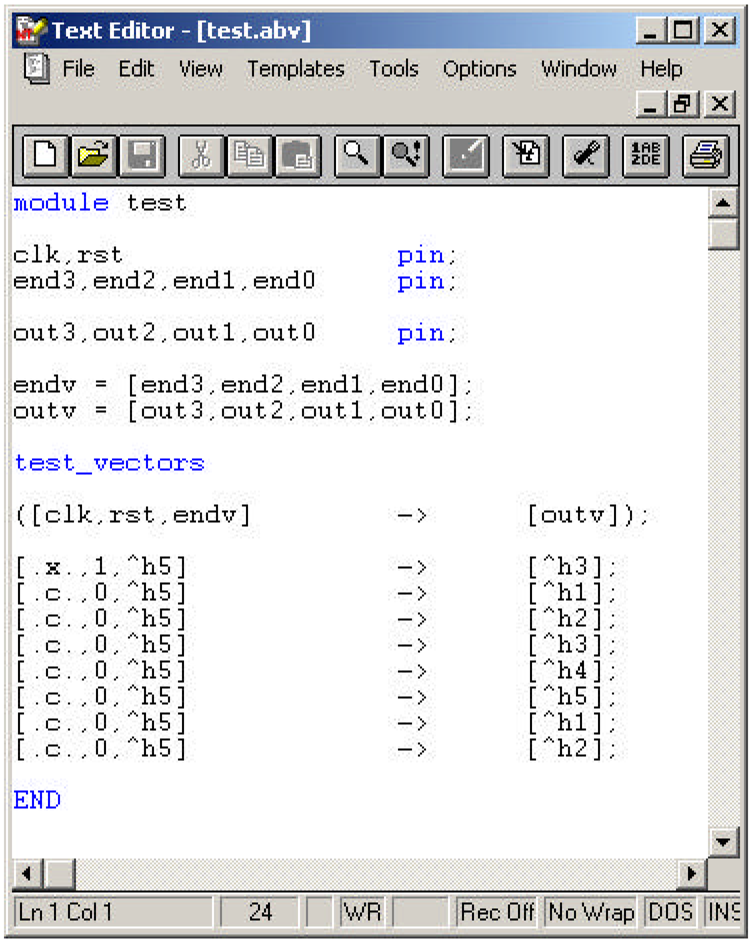
In this task you will import a test vector that has already been created for you.

To import a test vector source file:

1. In the Project Navigator, choose **Source** > **Import** to open the Import File dialog box.
2. Select the **`test.abv`** file and click **Open**. The file appears just below the device in the Sources window.



3. View the contents of the test vector file in the Text Editor by double clicking the file name in the Sources window.



```
module test

clk,rst                 pin;
end3,end2,end1,end0     pin;

out3,out2,out1,out0     pin;

endv = [end3,end2,end1,end0];
outv = [out3,out2,out1,out0];

test_vectors

([clk,rst,endv]         ->       [outv]);

[.x.,1,^h5]             ->       [^h3];
[.c.,0,^h5]             ->       [^h1];
[.c.,0,^h5]             ->       [^h2];
[.c.,0,^h5]             ->       [^h3];
[.c.,0,^h5]             ->       [^h4];
[.c.,0,^h5]             ->       [^h5];
[.c.,0,^h5]             ->       [^h1];
[.c.,0,^h5]             ->       [^h2];

END
```

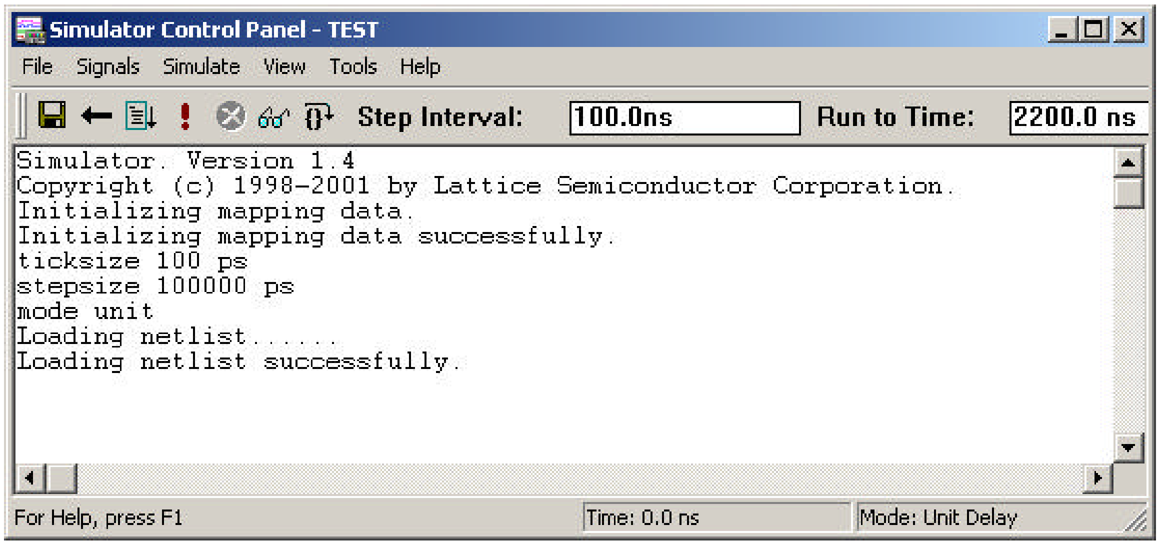
4. Close the Text Editor.

Task 30: Run Functional Simulation

After creating a test vector file, you can verify the behavior of your design by performing functional simulation. You can run integrated functional simulation using the Functional Simulation process in the design flow. This process opens the Lattice Logic Simulator and loads the functional netlist.

To run functional/timing simulation:

1. In the Project Navigator Sources window, select the ABEL-HDL test vector file **`test.abv`**.
2. In the Processes window, double-click the **Functional Simulation** process to launch the Lattice Logic Simulator Control Panel and load the netlist.

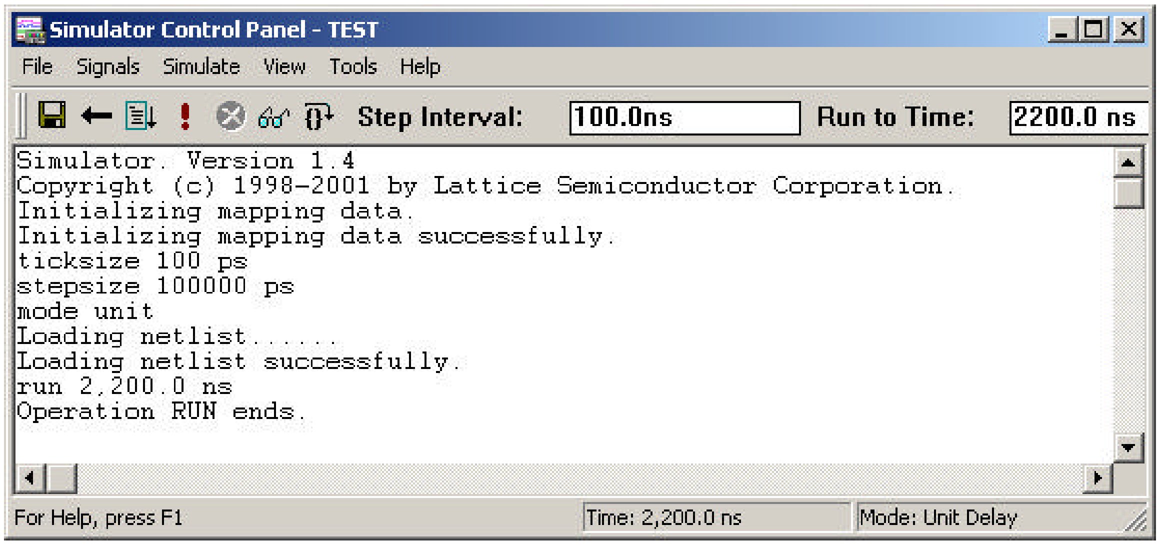

Simulator Control Panel - TEST
File Signals Simulate View Tools Help
Step Interval:
Run to Time:

Step Interval: 100.0ns
Run to Time: 2200.0 ns

```
Simulator. Version 1.4
Copyright (c) 1998-2001 by Lattice Semiconductor Corporation.
Initializing mapping data.
Initializing mapping data successfully.
ticksize 100 ps
stepsize 100000 ps
mode unit
Loading netlist......
Loading netlist successfully.
```

For Help, press F1 | Time: 0.0 ns | Mode: Unit Delay

3. In the Simulator Control Panel, choose **Simulate > Run** to start the simulation.

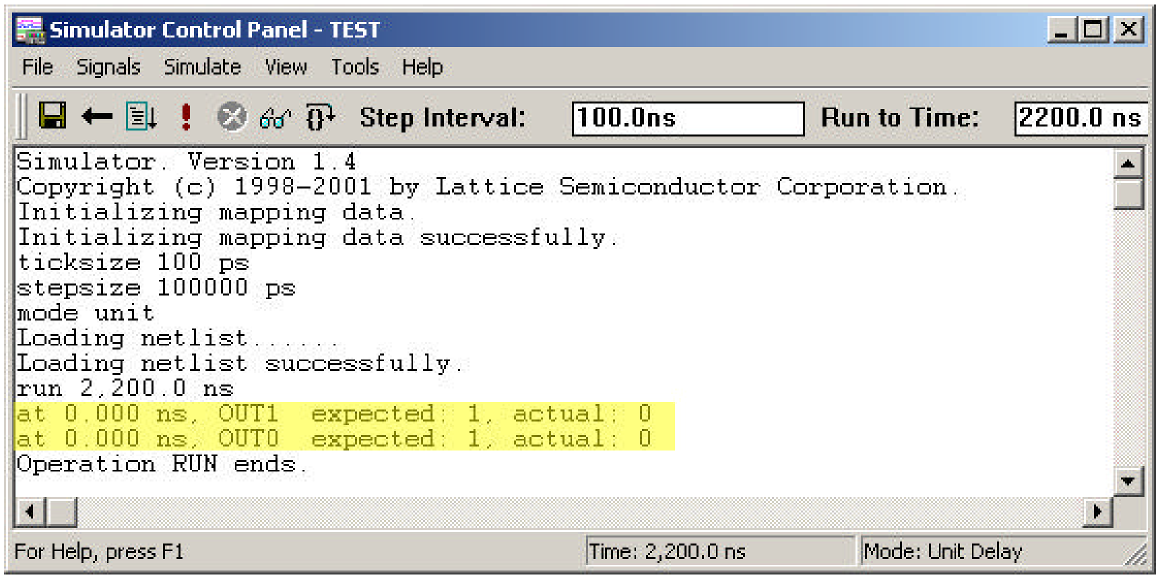

Simulator Control Panel - TEST
File Signals Simulate View Tools Help
Step Interval:
Run to Time:

Step Interval: 100.0ns
Run to Time: 2200.0 ns

```
Simulator. Version 1.4
Copyright (c) 1998-2001 by Lattice Semiconductor Corporation.
Initializing mapping data.
Initializing mapping data successfully.
ticksize 100 ps
stepsize 100000 ps
mode unit
Loading netlist......
Loading netlist successfully.
run 2,200.0 ns
Operation RUN ends.
```

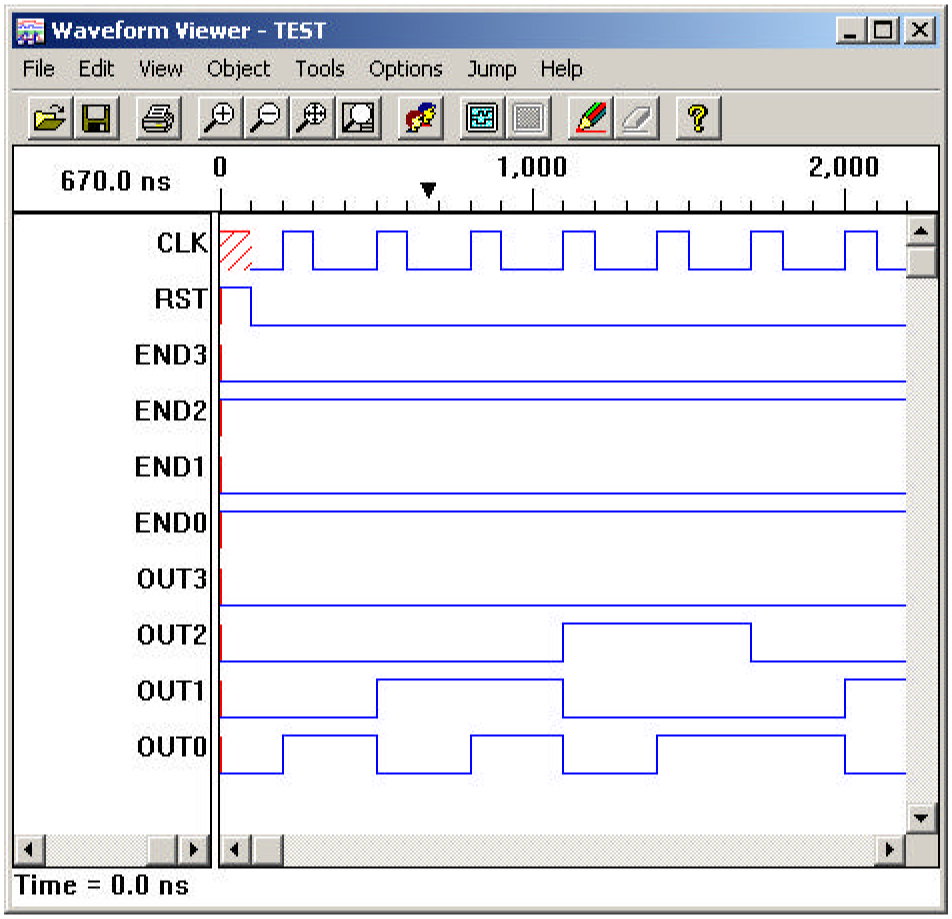
For Help, press F1 | Time: 2,200.0 ns | Mode: Unit Delay

***Note**: If there are errors in the simulation, the control panel displays a message similar to the example highlighted below:*



```
Simulator. Version 1.4
Copyright (c) 1998-2001 by Lattice Semiconductor Corporation.
Initializing mapping data.
Initializing mapping data successfully.
ticksize 100 ps
stepsize 100000 ps
mode unit
Loading netlist......
Loading netlist successfully.
run 2,200.0 ns
at 0.000 ns, OUT1  expected: 1, actual: 0
at 0.000 ns, OUT0  expected: 1, actual: 0
Operation RUN ends.
```

4. After functional simulation is complete, the simulator automatically opens the Waveform Viewer, which is the primary tool for viewing simulation results. The viewer graphically depicts the activity on any node in the simulation database. The Viewer automatically updates as simulation progresses.



5. Close the Waveform Viewer and the Lattice Logic Simulator Control Panel.
6. In the Project Navigator, choose **File** > **Save** to save the design before going to the next module.

Module 2: Design Implementation

For ispLEVER CPLD designs, the design implementation process consists of several steps, which generally include applying user constraints, and compiling, optimizing, and fitting the design. These individual steps happen automatically and in sequence when you run the Fit Design process in the Project Navigator.

This module shows you how to set constraints, and then leads you through the processes of compiling, optimizing, and fitting your CPLD design. Also, you will learn how to run multiple passes of your design using different combinations of Fitter/Optimizer settings and critical timing constraints, re-run your design with the new settings, and view the various report files generated by the ispLEVER software.

Prerequisites

Module 1.

Learning Objectives

When you have completed this module, you should be able to:

- Use the Constraint Editor to pre-assign pin locations
- Use the Constraint Editor Package View window to graphically view the actual pin assignments in the target device, and assign pin locations using the drag-and-drop technique.
- Run the ispLEVER Fitter
- Set Fitter report viewing options and read the Fitter report

Time to Complete This Module

The time to complete this module is about 20 minutes.

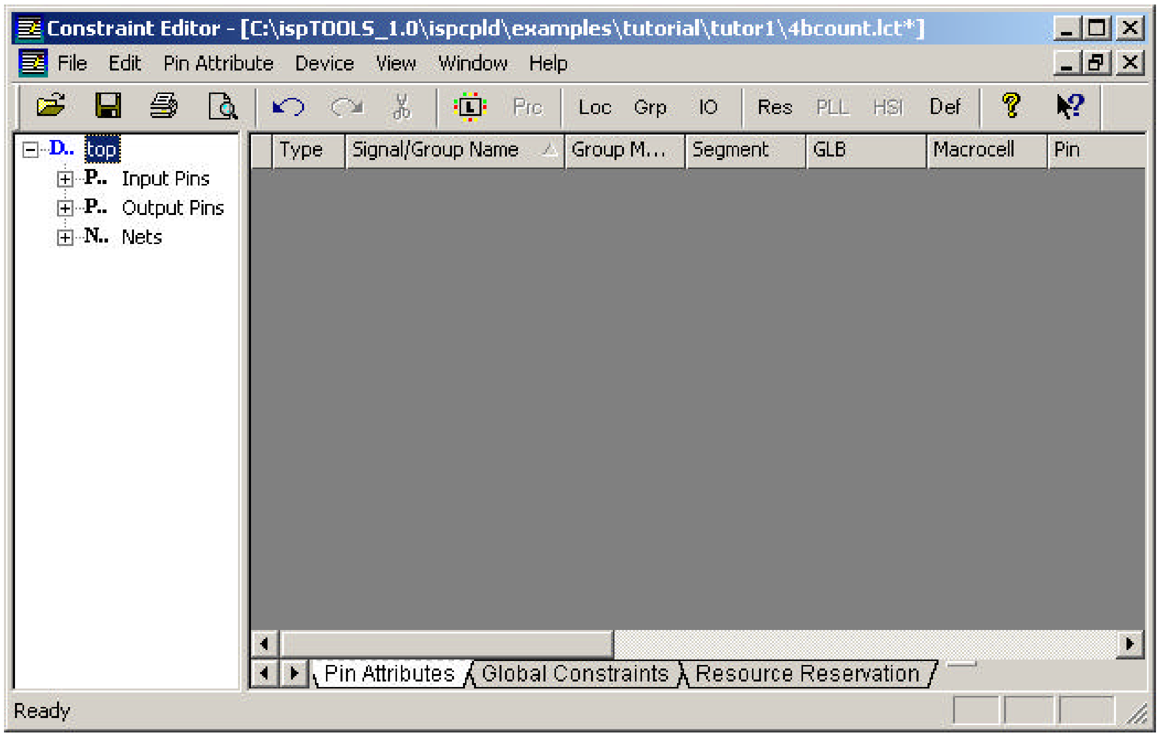
Task 1: Pre-Assign Pin Locations

In some cases you may want to assign constraints before running the Fitter. For example, you may want to pre-assign pin and node locations. The Constraint Editor lets you specify various constraints using the graphical interface such as pin and node assignments, group assignments, pin reservations, power level settings, output slew-rates, and JEDEC file options. The Constraint Editor reads the constraint file and displays the constraint settings in the main window. You can modify the constraint file using the function dialogs available from the toolbar. However, some constraints can be modified directly in the main window.

The Constraint Editor implements simple error checking to ensure that the user assignments or constraints are applicable to the selected device and that there are no conflicting assignments. If the user constraints do not apply to the selected device, or are conflicting with the selected device, the Constraint Editor displays these constraints in red.

To assign constraints to the design:

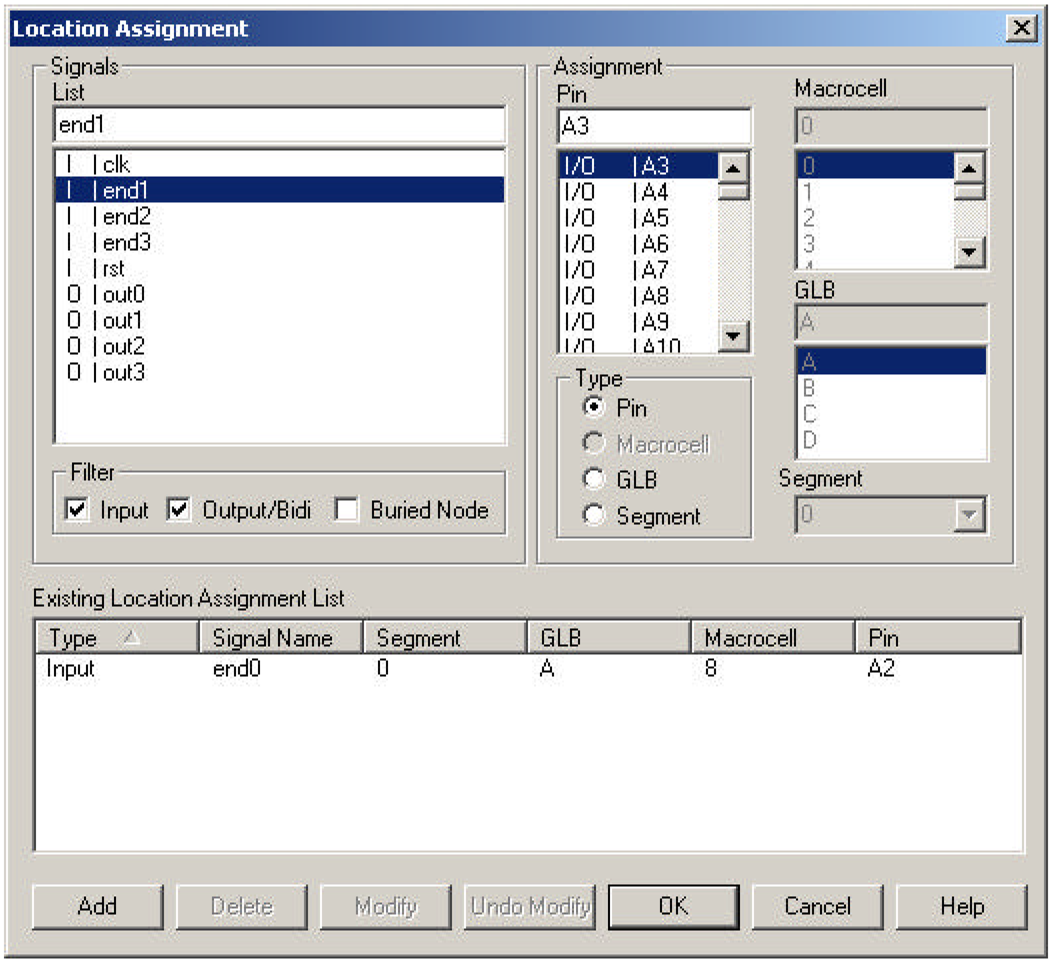
1. In the Sources window, select the target device. In the Processes window, double-click **Constraint Editor** to open it.



2. Choose **Pin Attribute > Location Assignment** to open the dialog box and do the following:

- Under Signals List, select **end0**.
- Under Pin Assignment, select **A2**.

- Click **Add**.



Location Assignment

Signals

List

end1

```
I  | clk
I  | end1
I  | end2
I  | end3
I  | rst
O  | out0
O  | out1
O  | out2
O  | out3
```

Filter: [✓ Input] [✓ Output/Bidi] Buried Node

Assignment

Pin: A3

```
I/O   | A3
I/O   | A4
I/O   | A5
I/O   | A6
I/O   | A7
I/O   | A8
I/O   | A9
I/O   | A10
```

Type: [✓ Pin] Macrocell GLB Segment

Macrocell: 0 (0, 1, 2, 3)

GLB: A (A, B, C, D)

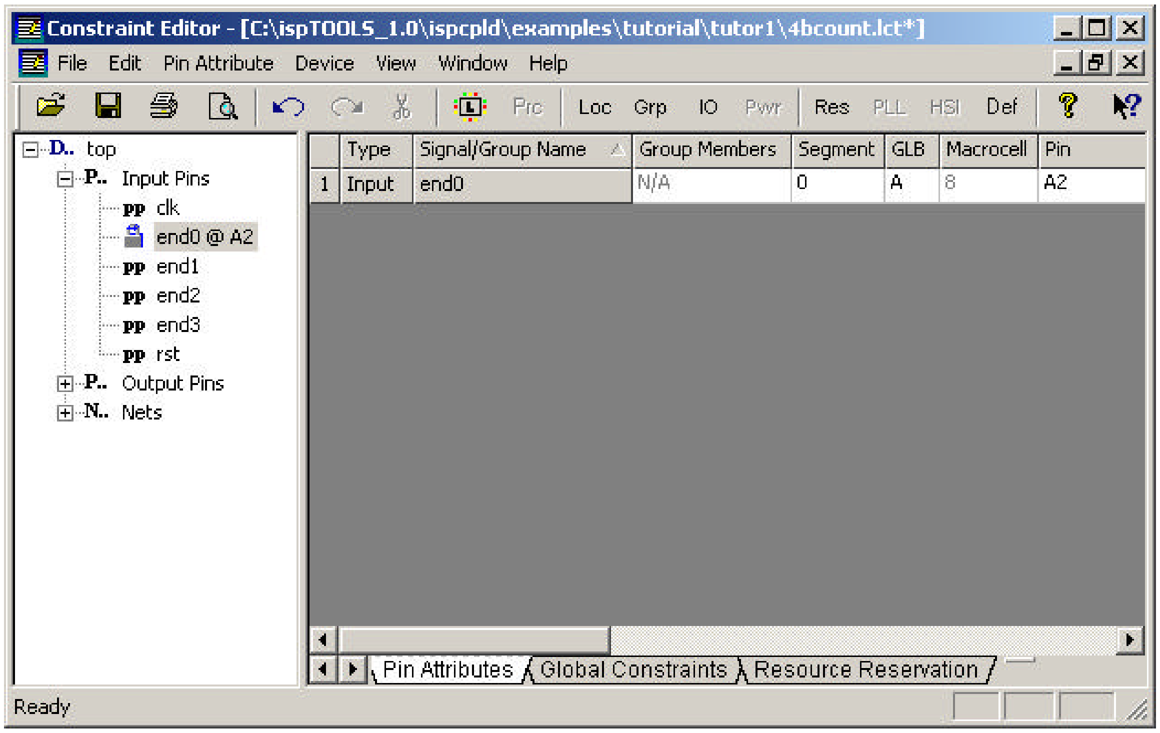
Segment: 0

Existing Location Assignment List

Type	Signal Name	Segment	GLB	Macrocell	Pin
Input	end0	0	A	8	A2

3. Click **OK** to close the dialog.

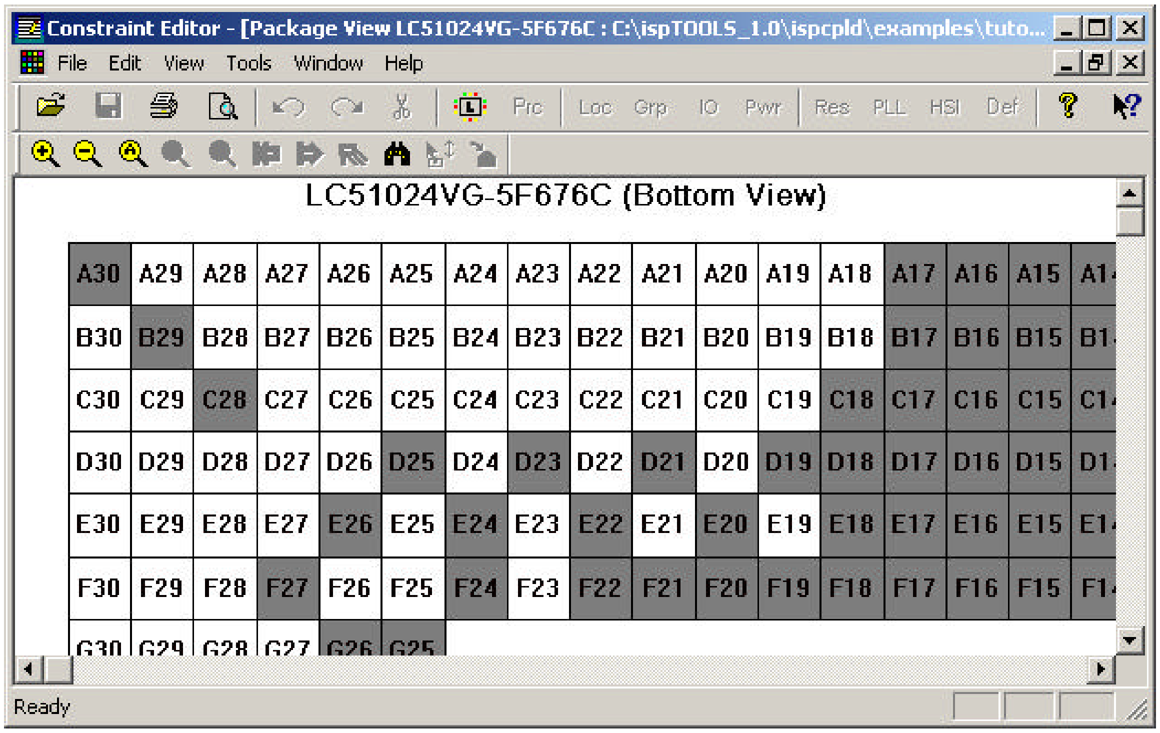
4. In the Constraint Editor, notice the pin location information in the Pin Attributes sheet. In the Signal List Pane on the left side of the main window, click the plus sign [+] in front of **Input Pins** to expand the tree view. Notice that the end0 pin has a "locked" icon next to it indicating that the pin location has been assigned.


Constraint Editor - [C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor1\4bcount.lct*]
File Edit Pin Attribute Device View Window Help
Prc Loc Grp IO Pwr Res PLL HSI Def
D.. top
P.. Input Pins
clk
end0 @ A2
end1
end2
end3
rst
P.. Output Pins
N.. Nets
Pin Attributes
Global Constraints
Resource Reservation
Ready

	Type	Signal/Group Name	Group Members	Segment	GLB	Macrocell	Pin
1	Input	end0	N/A	0	A	8	A2

5. You can also use the Constraint Editor Package View window to graphically view the actual pin assignments in the target device, and assign pin locations using the drag-and-drop technique. By default, the system (non-user) pins are highlighted in gray, reserved pins are in lime, assigned input pins are in blue, output pins in yellow, and bi-directional pins in magenta. Unused pins are blank.

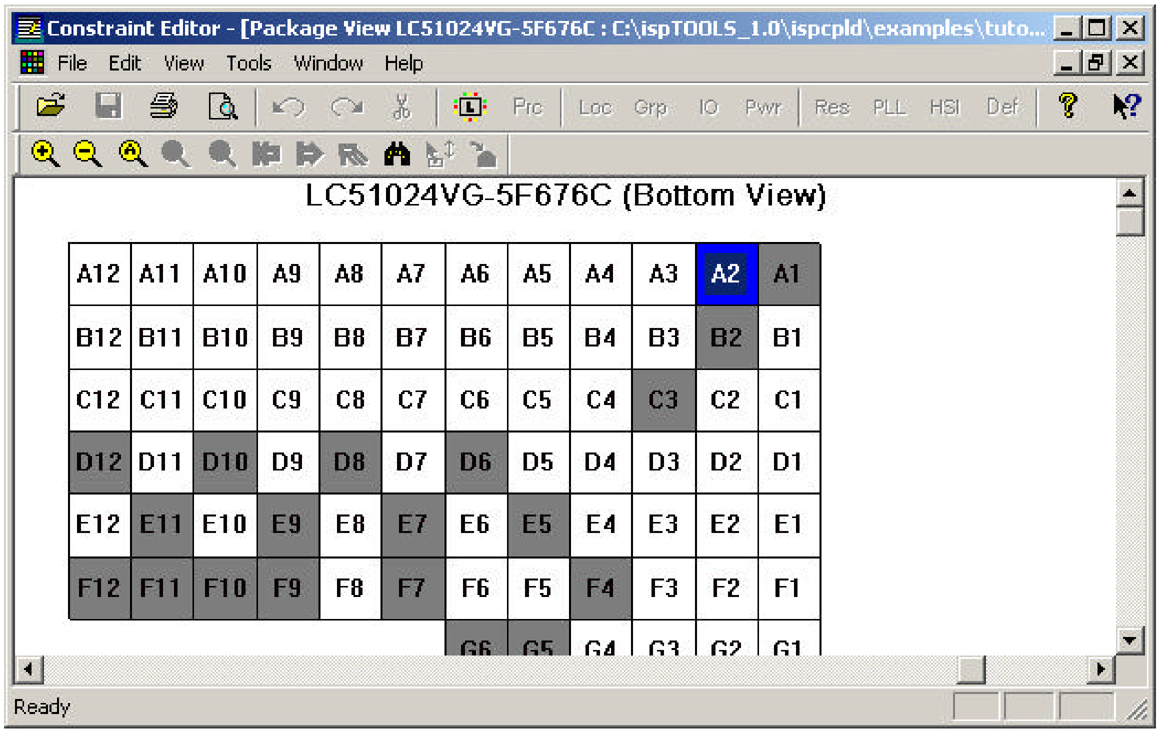
   Choose **Device > Package View** to open the Package View window.


Constraint Editor - [Package View LC51024VG-5F676C : C:\ispTOOLS_1.0\ispcpld\examples\tuto...
File Edit View Tools Window Help
Prc Loc Grp IO Pwr Res PLL HSI Def
Ready

LC51024VG-5F676C (Bottom View)

A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A1
B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16	B15	B1
C30	C29	C28	C27	C26	C25	C24	C23	C22	C21	C20	C19	C18	C17	C16	C15	C1
D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D1
E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E1
F30	F29	F28	F27	F26	F25	F24	F23	F22	F21	F20	F19	F18	F17	F16	F15	F1
G30	G29	G28	G27	G26	G25											

6. To easily locate a previously assigned pin, on the toolbar click the Find icon (🔍). In the Find dialog box, type **A2**, select **Device Pins**, and then click **Find**.

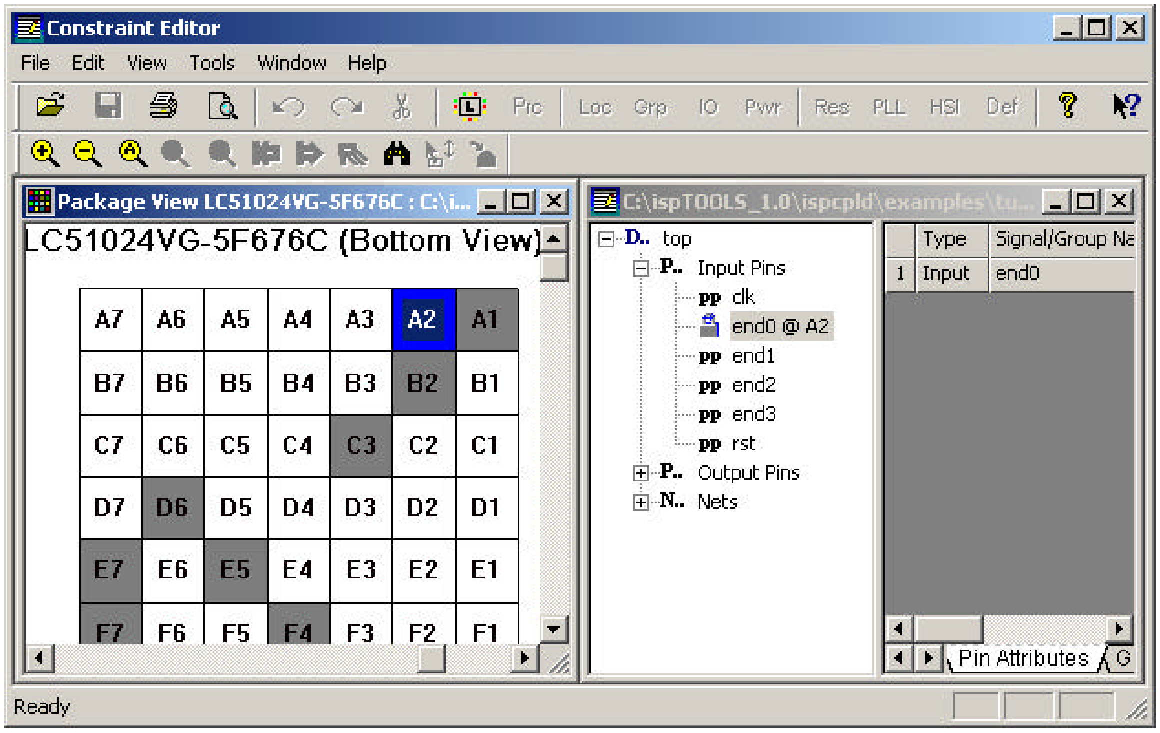

Constraint Editor - [Package View LC51024VG-5F676C : C:\ispTOOLS_1.0\ispcpld\examples\tuto...
File Edit View Tools Window Help
Prc Loc Grp IO Pwr Res PLL HSI Def
Ready

LC51024VG-5F676C (Bottom View)

A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1
C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1
D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1
E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1
F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1
						G6	G5	G4	G3	G2	G1

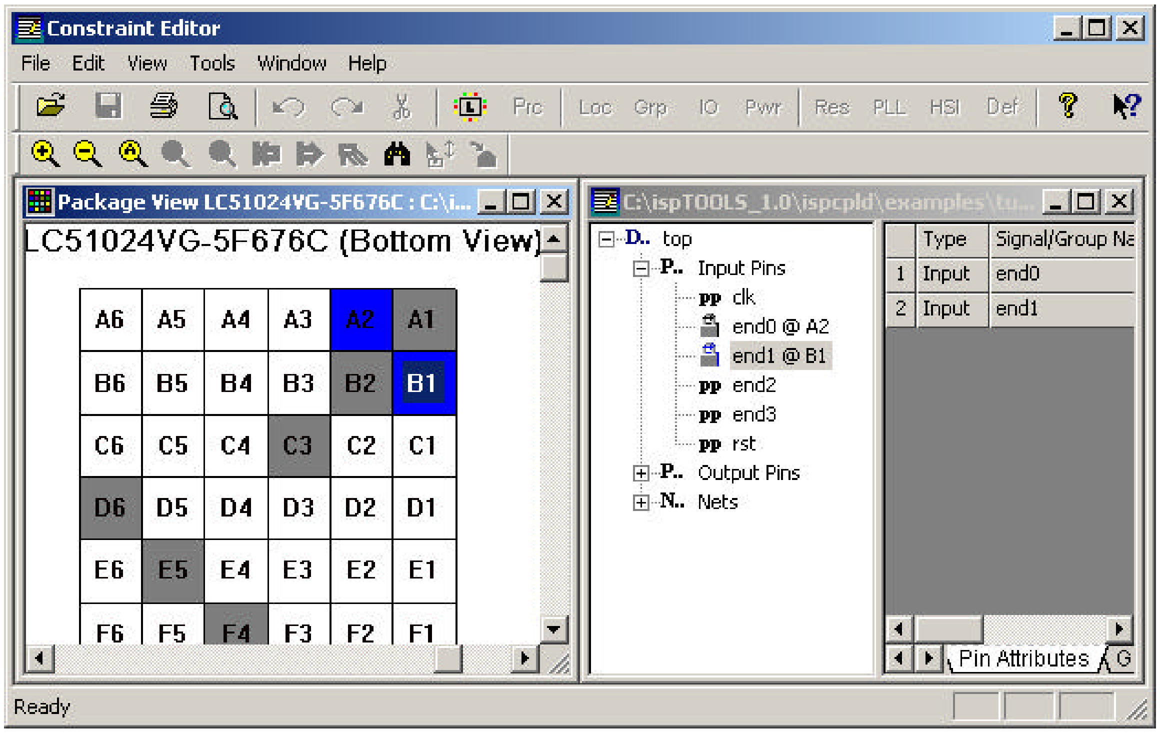
The view shifts to the specified location.

7. You can also assign pin locations using the Package View window. Choose **Window > Tile Vertically**. Make sure the Package View window is showing the A2 pin and the Signal List is expanded to show the Input Pins.



	Type	Signal/Group Na
1	Input	end0

8. Select **end1** from the Signals List and drag it to the **B1** location in the Package View window. Notice the B1 location is blue (assigned input pins), the "locked" icon appears next to end1 in the Signals List, and end1 has been added to the Pin Attributes sheet.



	Type	Signal/Group Na
1	Input	end0
2	Input	end1

9. Close the Package View window. Choose **File** > **Save** to save the pin assignment. Close the Constraint Editor.

Task 2: Fit the Design

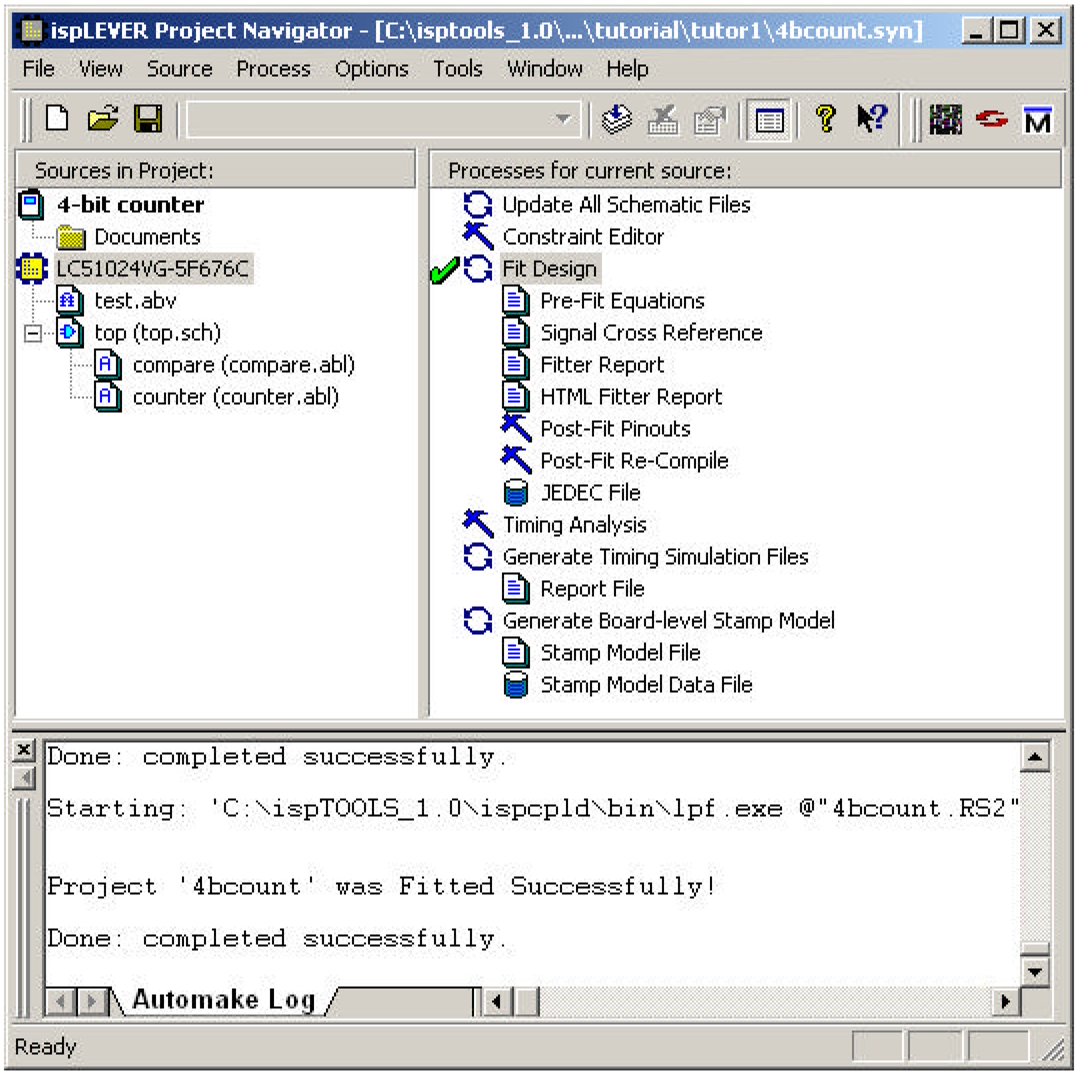
The ispLEVER software has a single user interface with all options preset to deliver the highest possible push-button performance. At the end of a successful fitter run, the ispLEVER software generates a JEDEC file, as well as a fitter report, so that you can see how the ispLEVER software has routed the design and utilized resources on the part.

Fitting is an integrated process that includes these steps:

- **Compiling** - Changing your design entry format into Boolean equations, which serve as input to simulation and device implementation programs.
- **Optimizing -** Running a set of options that let you achieve the highest possible performance in the smallest possible device, for most designs.
- **Partitioning** - After optimization, the design is partitioned into individual blocks on the specified device. Partitioning is achieved by assigning logic to specific blocks, based on several considerations.
- **Fitting** - Running the device fitter. A fitter report is generated regardless of whether the process is successful or not. The Fitter also generates the JEDEC file.

To fit a design:

1. In the Sources window, select the target device.
2. In the Processes window, double-click **Fit Design**. A message appears in the Output Panel telling you that the fit process completed successfully. Also, a green check mark appears to the left of the Fit Design process.



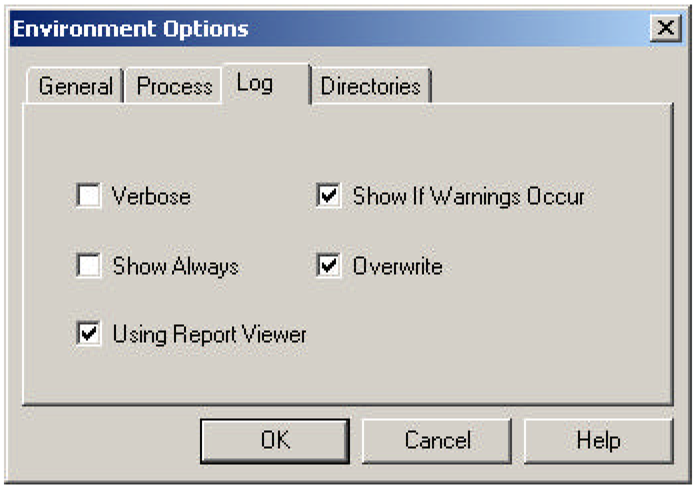
```
Done: completed successfully.
Starting: 'C:\ispTOOLS_1.0\ispcpld\bin\lpf.exe @"4bcount.RS2"

Project '4bcount' was Fitted Successfully!
Done: completed successfully.
```

Task 3: Set Report Viewing Options

By default, the ispLEVER software opens report files in the Project Navigator Output Panel. However, some reports may be too large to allow easy viewing in the Output Panel.

To open a report in the Report Viewer:

- Choose **Options > Environment** to open the dialog box. Then click the **Log** tab and select **Using Report Viewer**. Click **OK** to close the dialog box.



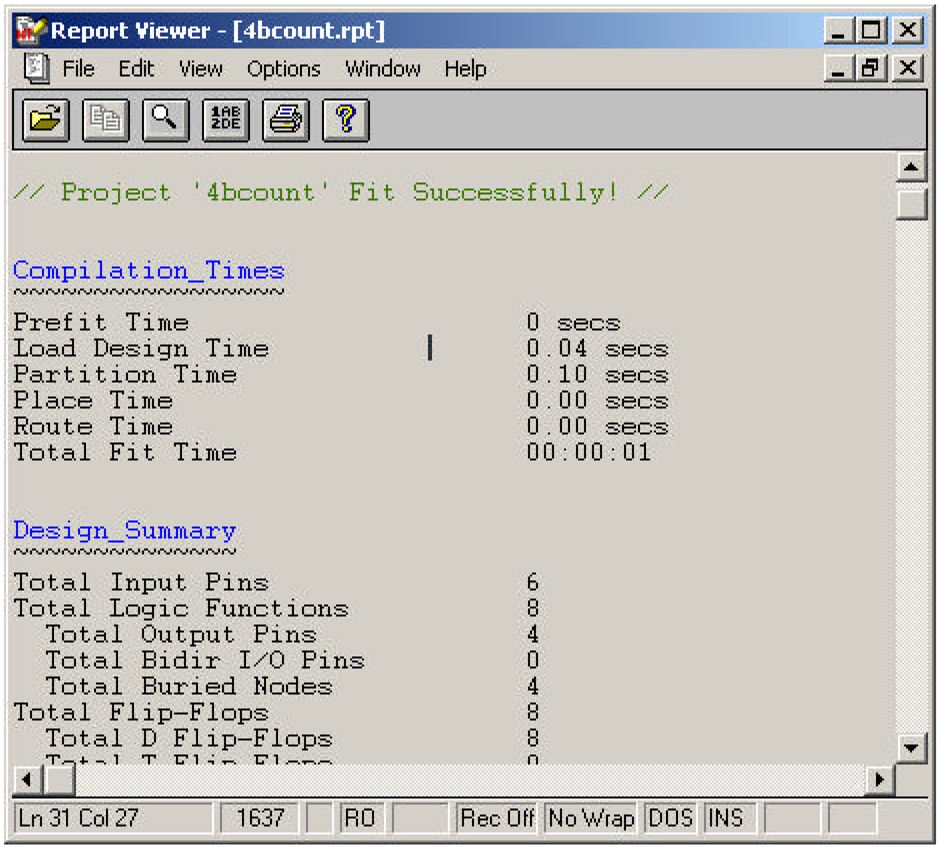
☐ Verbose ☑ Show If Warnings Occur
☐ Show Always ☑ Overwrite
☑ Using Report Viewer

Task 4: Read the Fitter Report

The Fitter Report displays statistics and informatio n on the fitting process of your design, including utilization numbers, pin assignments, etc.

To view the Fitter Report:

1. In the Processes window, double-click the **Fitter Report** process to open the Fitter Report in the Report Viewer. View the report and then close the Report Viewer.



```
// Project '4bcount' Fit Successfully! //


Compilation_Times
~~~~~~~~~~~~~~~~~
Prefit Time                     0 secs
Load Design Time                0.04 secs
Partition Time                  0.10 secs
Place Time                      0.00 secs
Route Time                      0.00 secs
Total Fit Time                  00:00:01


Design_Summary
~~~~~~~~~~~~~~
Total Input Pins                6
Total Logic Functions           8
  Total Output Pins             4
  Total Bidir I/O Pins          0
  Total Buried Nodes            4
Total Flip-Flops                8
  Total D Flip-Flops            8
```

2. As you probably noticed, the Fitter Report is divided into several sections and can be quite long and difficult to find the information that you are looking for. As an alternative, you can view the Fitter Report using your HTML browser. This offers the benefit of easy browsing using a navigation panel with links to each major section in the report.

   In the Processes window, double-click the **HTML Fitter Report** process to open the report in your browser.

3. Click the **Pinout Listing** navigation link. Find pin number **A2**. Look all the way to the right and notice that the signal is **end0**. Remember that this is the pin location constraint that you set in a previous task in this module.

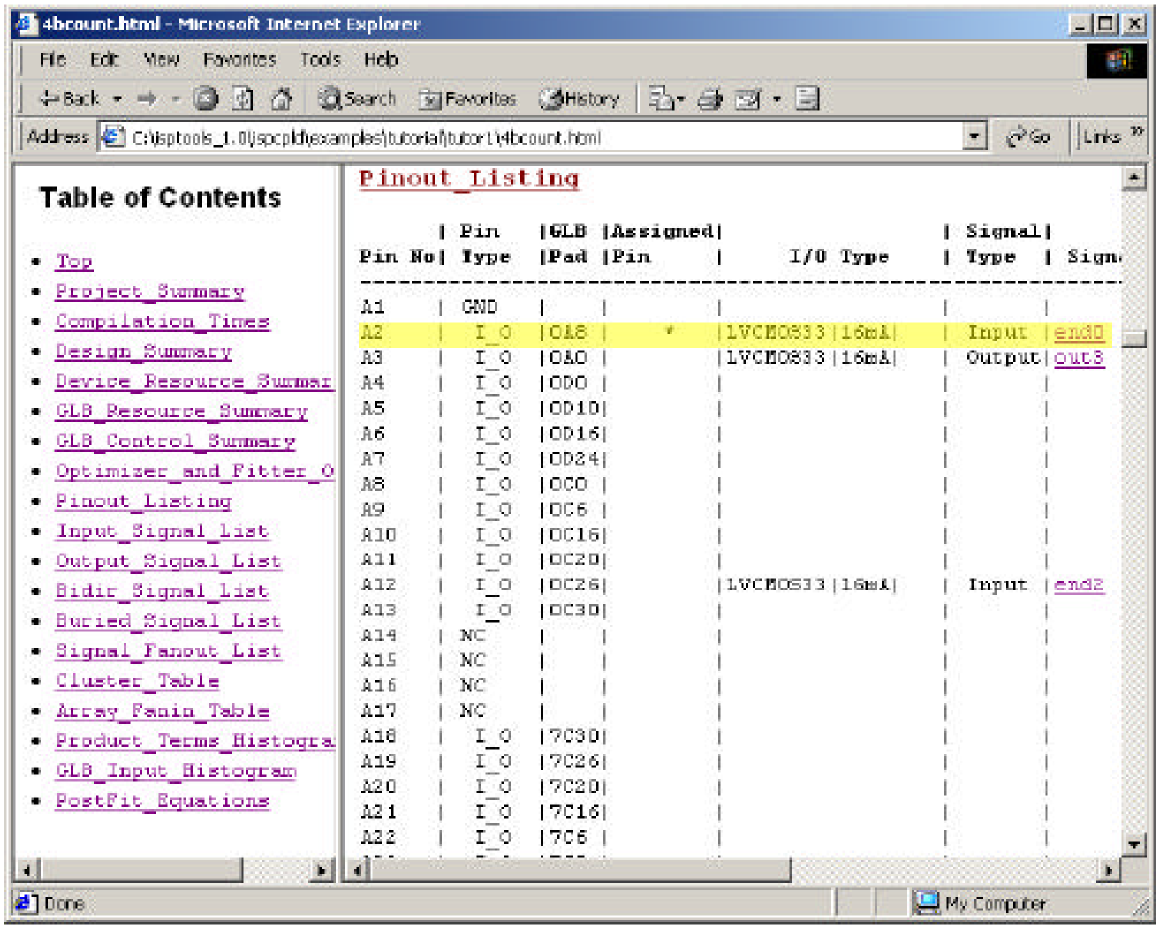

4bcount.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\isptools_1.0\ispcpld\examples\tutorial\tutor1\4bcount.html
Go Links
Done
My Computer

Table of Contents

- Top
- Project_Summary
- Compilation_Times
- Design_Summary
- Device_Resource_Summar
- GLB_Resource_Summary
- GLB_Control_Summary
- Optimizer_and_Fitter_O
- Pinout_Listing
- Input_Signal_List
- Output_Signal_List
- Bidir_Signal_List
- Buried_Signal_List
- Signal_Fanout_List
- Cluster_Table
- Array_Fanin_Table
- Product_Terms_Histogra
- GLB_Input_Histogram
- PostFit_Equations

Pinout_Listing

```
       | Pin    |GLB |Assigned|                  | Signal|
Pin No|  Type   |Pad |Pin     |     I/O Type     | Type  | Sign
---------------------------------------------------------------
A1     | GND    |    |        |                  |       |
A2     |  I_O   |0A8 |    *   |LVCMOS33|16mA|   | Input |end0
A3     |  I_O   |0A0 |        |LVCMOS33|16mA|   | Output|out3
A4     |  I_O   |0D0 |        |                  |       |
A5     |  I_O   |0D10|        |                  |       |
A6     |  I_O   |0D16|        |                  |       |
A7     |  I_O   |0D24|        |                  |       |
A8     |  I_O   |0C0 |        |                  |       |
A9     |  I_O   |0C6 |        |                  |       |
A10    |  I_O   |0C16|        |                  |       |
A11    |  I_O   |0C20|        |                  |       |
A12    |  I_O   |0C26|        |LVCMOS33|16mA|   | Input |end2
A13    |  I_O   |0C30|        |                  |       |
A14    | NC     |    |        |                  |       |
A15    | NC     |    |        |                  |       |
A16    | NC     |    |        |                  |       |
A17    | NC     |    |        |                  |       |
A18    |  I_O   |7C30|        |                  |       |
A19    |  I_O   |7C26|        |                  |       |
A20    |  I_O   |7C20|        |                  |       |
A21    |  I_O   |7C16|        |                  |       |
A22    |  I_O   |7C6 |        |                  |       |
```

4. When you are finished looking at the Fitter report close the browser.

5. Choose **File > Save** to save the design.

Module 3: Design Verification

This module provides an overview of the features and operation of the ispLEVER software, focusing on the tasks and tools needed to verify a MACH device design.

Prerequisites

Module 2.

Learning Objectives

When you have completed this module, you should be able to:

- Run timing analysis and simulation and analyze the results
- Correlate simulation results using cross-probing
- Export the netlist and delays for timing simulation
- Build board-level Stamp models of a design

Time to Complete This Module

The time to complete this module is about 20 minutes.

Task 1: Run Static Timing Analysis

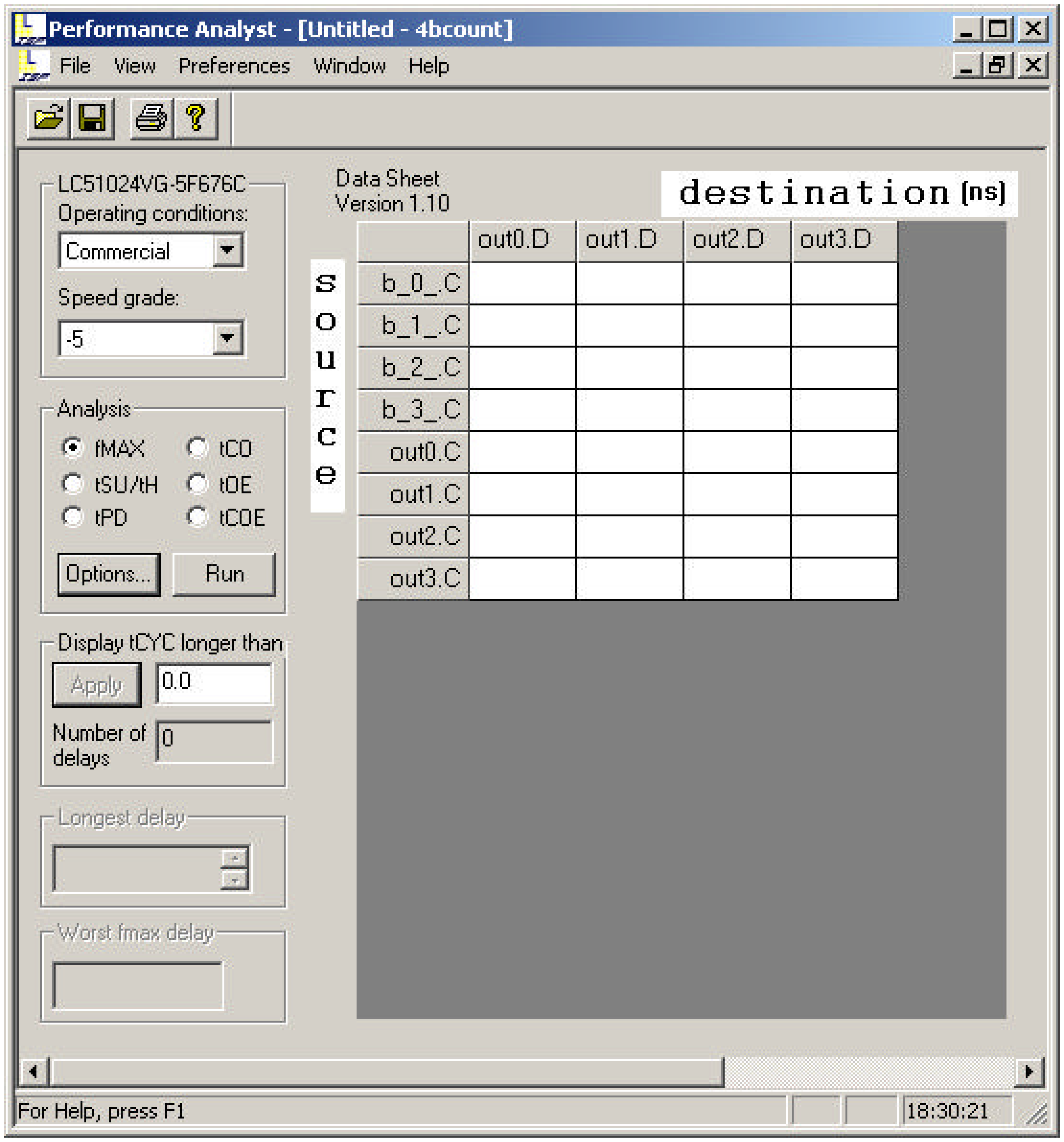
Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data such as the critical path, setup/hold time requirements, and the maximum frequency.

The Performance Analyst traces each logical path in the design and calculates the path delays using the device's timing model and worst-case AC specs supplied in the device data sheet.

The timing analysis results are displayed in a graphical spreadsheet with source signals displayed on the vertical axis and destination signals displayed on the horizontal axis. The worst-case delay value is displayed in a spreadsheet cell if there is at least one delay path between the source and destination. To more easily identify performance bottlenecks, you can double-click a cell to view the path delay details.

To run timing analysis:

1. In the Project Navigator Sources window, select the target device. In the Processes window, double-click the **Timing Analysis** process to open the Performance Analyst.


Performance Analyst - [Untitled - 4bcount]

LC51024VG-5F676C

Operating conditions: Commercial

Speed grade: -5

Analysis: [✓ fMAX] tCO tSU/tH tOE tPD tCOE

Display tCYC longer than: 0.0

Number of delays: 0

Data Sheet Version 1.10

destination (ns)

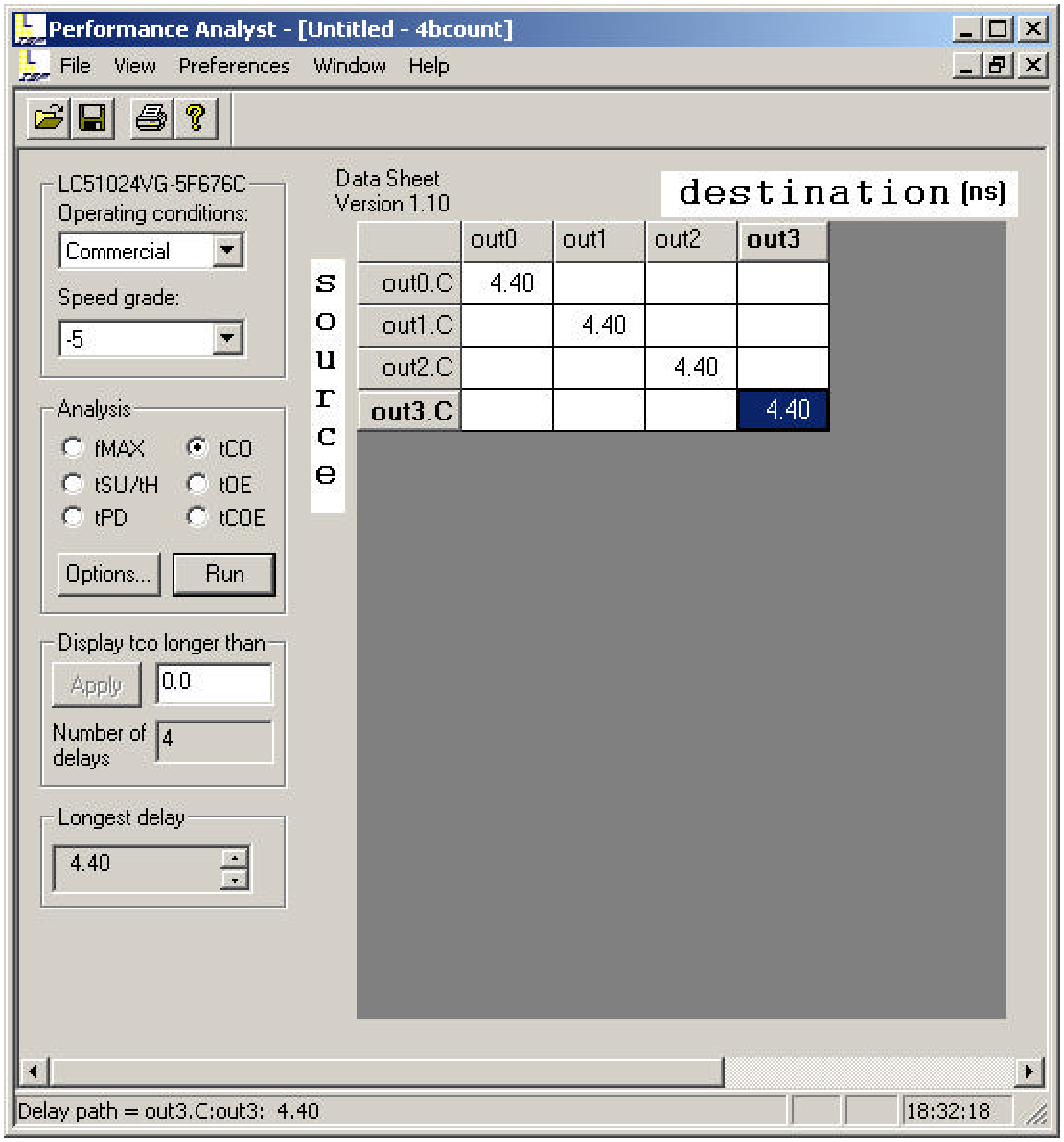
source	out0.D	out1.D	out2.D	out3.D
b_0_C				
b_1_C				
b_2_C				
b_3_C				
out0.C				
out1.C				
out2.C				
out3.C				

For Help, press F1

18:30:21

2. The Performance Analyst performs six distinct analysis types: fMAX, tSU/tH, tPD, tCO, tOE, and tCOE. The first type, fMAX, is an internal register-to-register delay analysis. fMAX measures the maximum clock operating frequency, limited by worst-case register-to-register delay. The remaining five types are external pin-to-pin delay analysis. Timing threshold filters, source and destination filters, and path filters can be used to independently fine-tune each analysis.

   Under Analysis, select **tCO** and then click **Run**. The tCO path trace analysis reports clock-to-out delay starting from the primary input, going through the clock of flip-flops or gate of latches, and ending at the primary output. In this case it is 4.40ns.



LC51024VG-5F676C

Operating conditions: Commercial

Speed grade: -5

Analysis: fMAX ○ tSU/tH ○ tPD ○ [✓ tCO] tOE ○ tCOE ○

Display tco longer than: 0.0

Number of delays: 4

Longest delay: 4.40

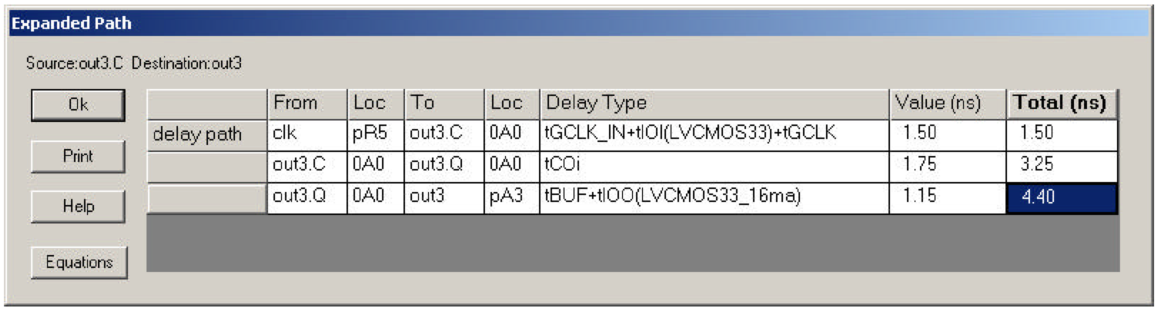
Data Sheet Version 1.10

	out0	out1	out2	out3
out0.C	4.40			
out1.C		4.40		
out2.C			4.40	
out3.C				4.40

Delay path = out3.C:out3: 4.40 | 18:32:18

There are several things to notice. The lower-left corner shows the Longest Delay (4.40ns). Also, in the Spreadsheet window, you can see the delays in a source-destination matrix.

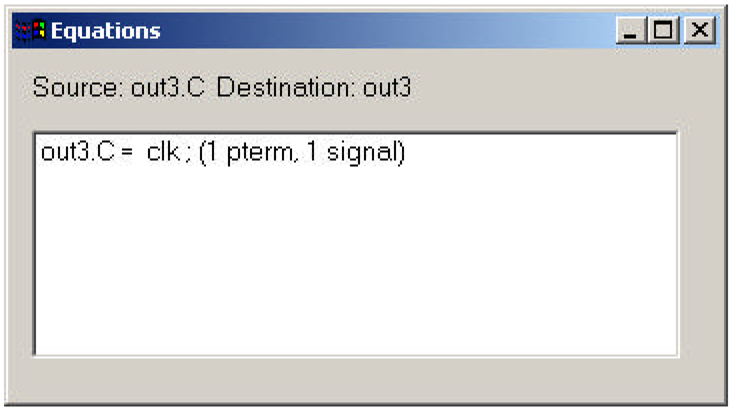
3. In the Spreadsheet window, double-click inside the selected (blue) cell to open the Expanded Path dialog box. This dialog lets you analyze individual timing components used to calculate the timing path. There is a source pin (From) and a destination pin (To). Also shown are the delay type, the delay of that path (Value ns), and the cumulative delay of all the signals.



Source:out3.C Destination:out3

	From	Loc	To	Loc	Delay Type	Value (ns)	Total (ns)
delay path	clk	pR5	out3.C	0A0	tGCLK_IN+tIOI(LVCMOS33)+tGCLK	1.50	1.50
	out3.C	0A0	out3.Q	0A0	tCOi	1.75	3.25
	out3.Q	0A0	out3	pA3	tBUF+tIOO(LVCMOS33_16ma)	1.15	4.40

4. Click **Equations** to open the Equations dialog box, which shows the functional relationship between the selected source/destination.



Source: out3.C Destination: out3

out3.C = clk ; (1 pterm, 1 signal)

5. Close the Performance Analyst without saving.

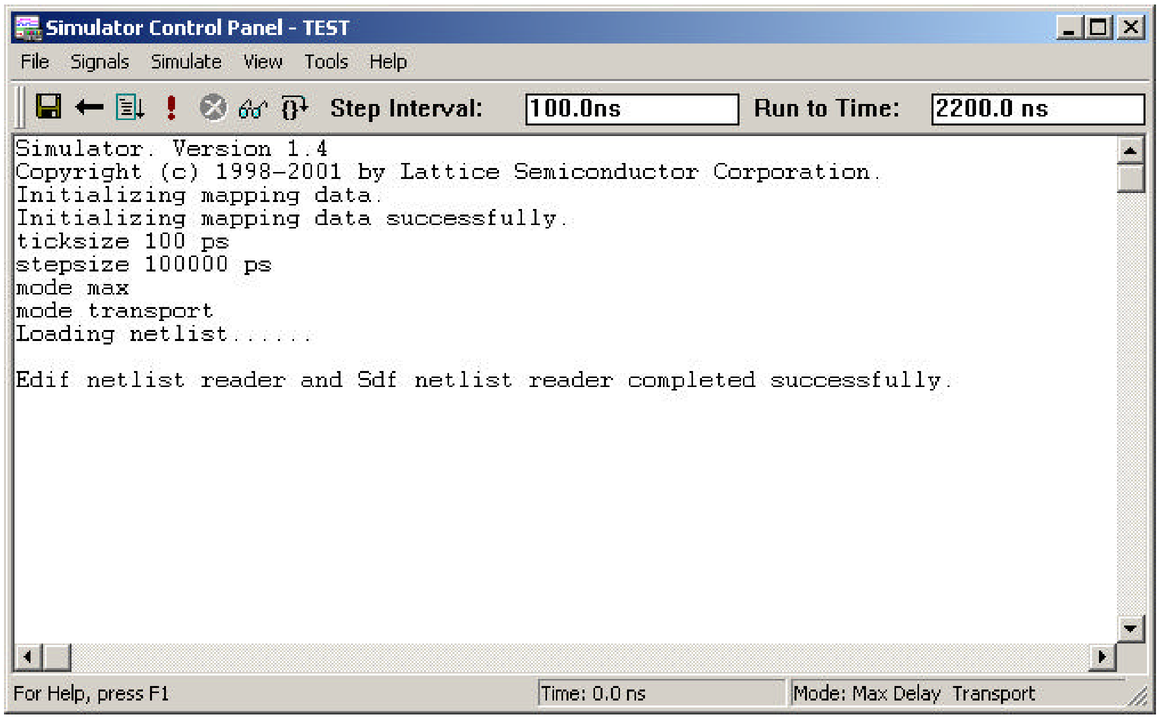
Task 2: Run Timing Simulation

In a previous module you used the Lattice Logic Simulator for functional simulation. Now you will use the simulator for timing simulation.

Timing simulation differs from timing analysis in a couple of ways. For example, you need a test vector file for simulation. And while both tools provide timing information based on an implemented design, only the Lattice Logic Simulator simulates the design logic as well. Also, the simulator provides graphical waveform display and debugging capabilities. In the end, which tool you use will depend on your specific requirements. You can use the Performance Analyst for a quick critical path analysis, and use the Lattice Logic Simulator for a more detailed and thorough simulation and analysis.

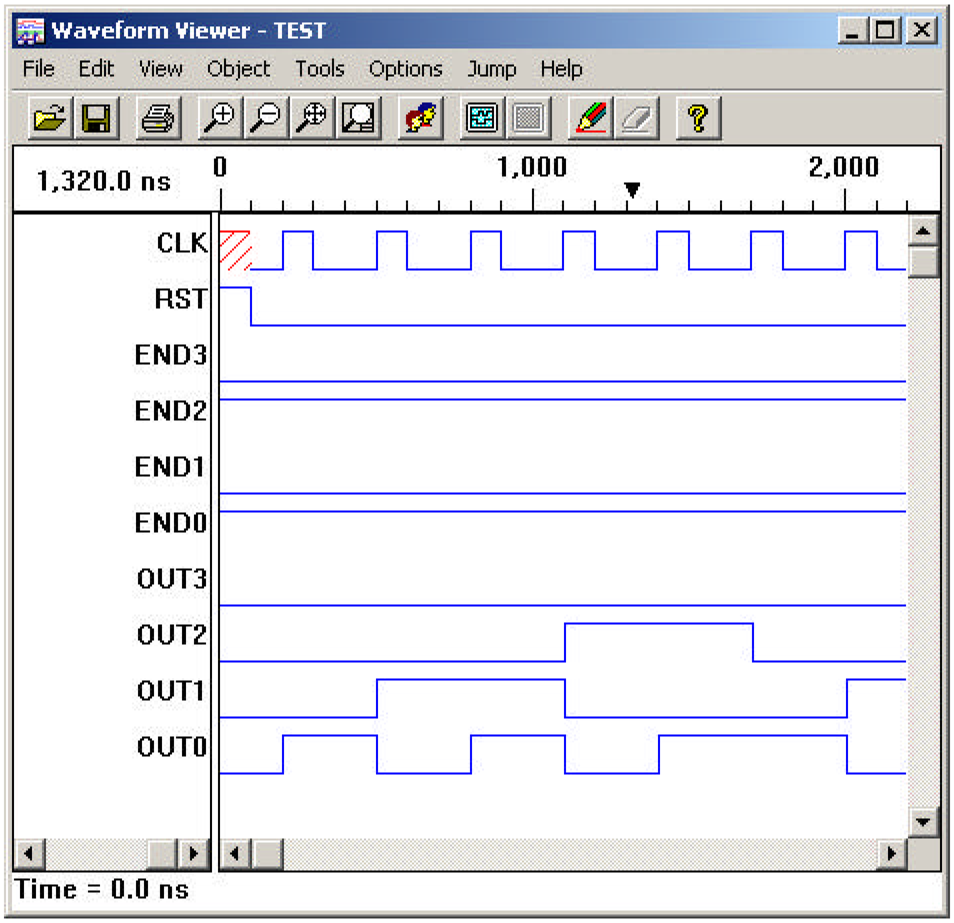
To run timing simulation:

1. In the Project Navigator Sources window, select the test vector file **`test.abv`**.
2. In the Processes window, double-click the **Timing Simulation** process to open the Lattice Logic Simulator Control Panel. In the toolbar, make sure that the Step Interval is set to **100.0ns** and Run to Time is set to **2200.0ns**.


Simulator Control Panel - TEST
File Signals Simulate View Tools Help
Step Interval: 100.0ns
Run to Time: 2200.0 ns
For Help, press F1
Time: 0.0 ns
Mode: Max Delay Transport

```
Simulator. Version 1.4
Copyright (c) 1998-2001 by Lattice Semiconductor Corporation.
Initializing mapping data.
Initializing mapping data successfully.
ticksize 100 ps
stepsize 100000 ps
mode max
mode transport
Loading netlist......

Edif netlist reader and Sdf netlist reader completed successfully.
```

3. Choose **Simulate** > **Run** to start the timing simulation and open the Waveform Viewer.

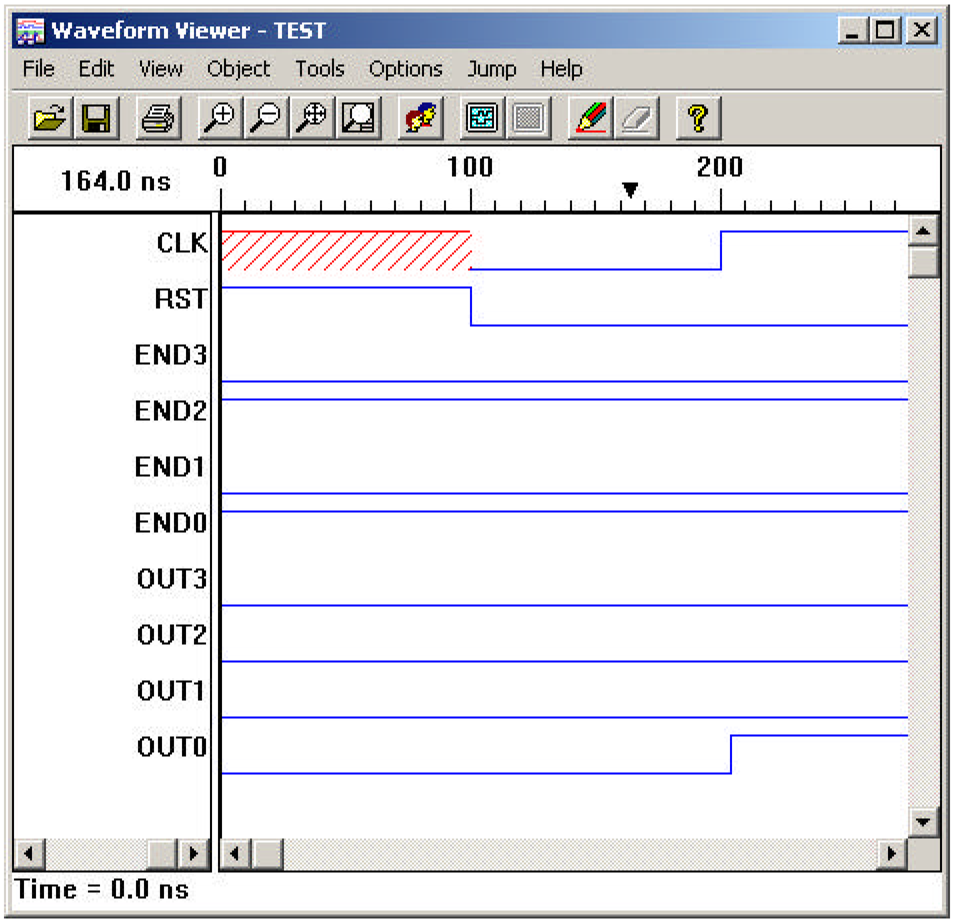


Task 3: Analyze the Simulation Results

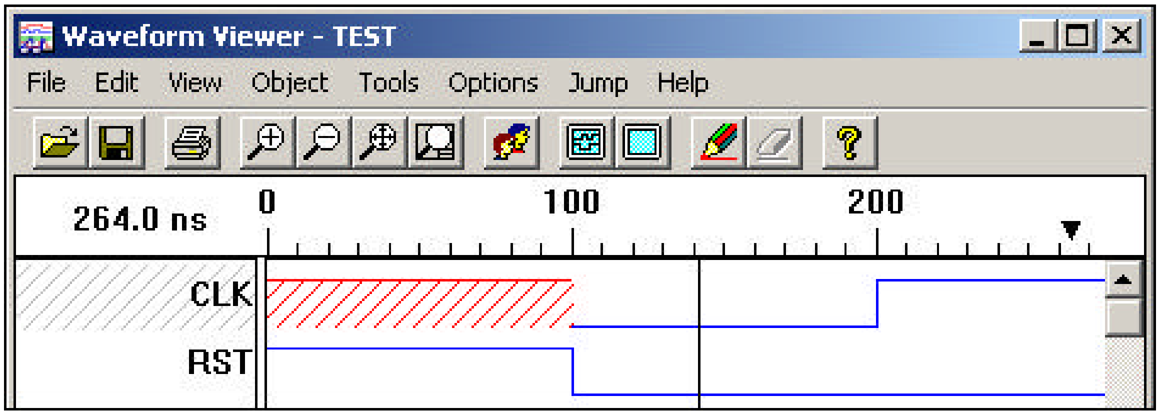
You can use the Waveform to measure the time difference between two events. In this task you will measure the tCO and see how it compares with the results of the Performance Analyst in a previous task.

To measure the difference between two events:

1. On the toolbar, click the **Zoom In** icon (🔍). The cursor switches to a large "Z." Click in the waveform display area until the major scale is in 100ns increments, as shown in the horizontal timeline below the toolbar. Right-click to return to the regular cursor.
2. Choose Jump > Time=0.

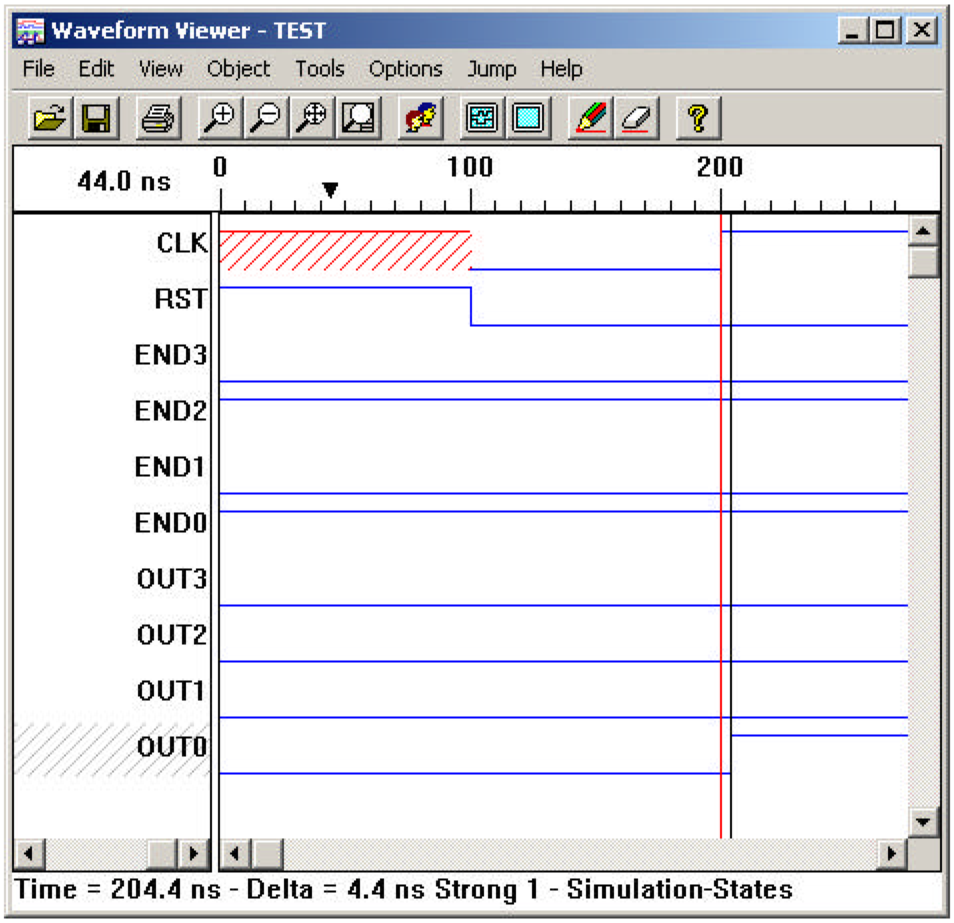


3. Select the **CLK** signal by clicking the label in the waveform name area. Click the Query cursor at about time **150ns**. This sets a vertical marker at that point.



4. Choose **Jump > Next Change**. The marker jumps to the time 200ns.
5. Choose **Object > Place Marker** to set a "permanent " marker at this point.
6. Now select the **OUT0** signal.
7. Choose **Jump > Next Change**. The temporary marker jumps to time 204.4ns. In the status bar, notice **Delta = 4.4ns**. Remember that tCO measured in the Performance

Analyst was also 4.40ns.



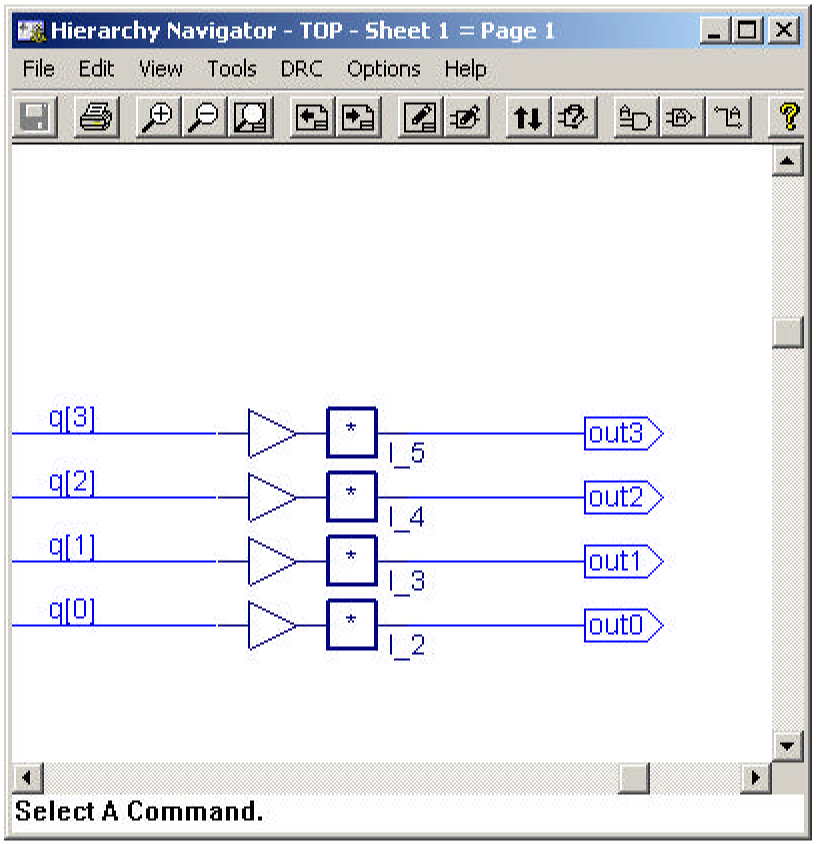
Time = 204.4 ns - Delta = 4.4 ns Strong 1 - Simulation-States

Task 4: Correlate Simulation Results using Cross-probing

Cross-probing lets you display simulation results on schematics, and lets you add waveforms to the Waveform Viewer from a schematic. Crossing-probing also makes it easier to correlate simulation results with the sources in the design.

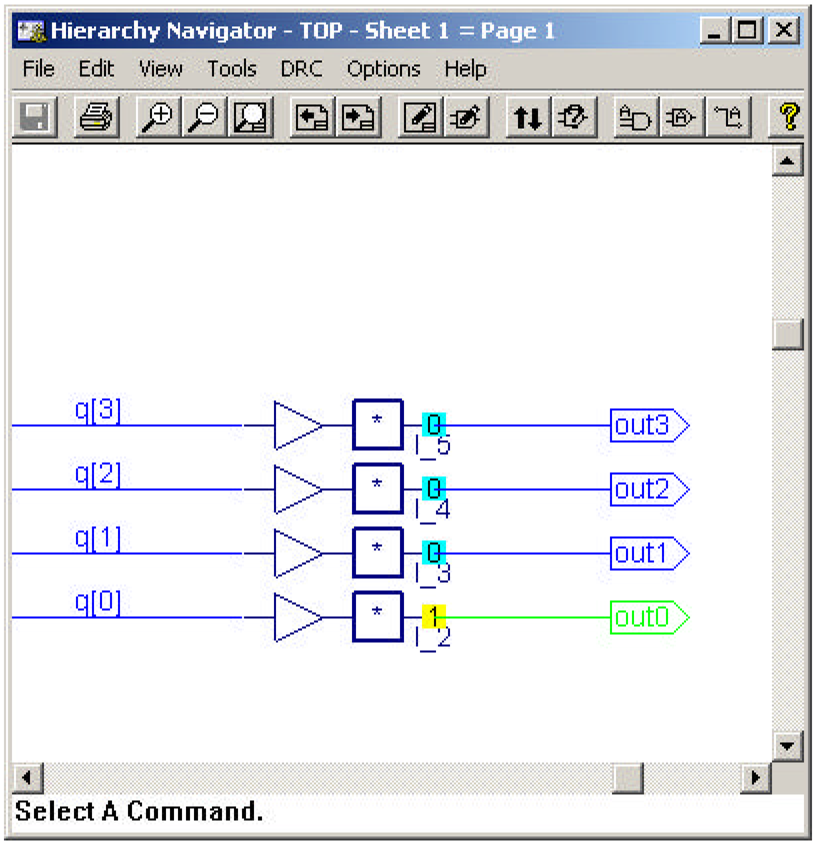
To cross-probe:

1. While the Waveform Viewer is running, select the **top** schematic source (`top.sch`) in the Project Navigator.
2. In the Processes window, double-click **Navigate Hierarchy** to open the Hierarchy Navigator. Position it beside the Waveform Viewer.
3. In the Hierarchy Navigator, choose **View** > **Zoom In** and enlarge the view of the output pins on the far right of the schematic (out0-3).


Hierarchy Navigator - TOP - Sheet 1 = Page 1
File Edit View Tools DRC Options Help
q[3]
q[2]
q[1]
q[0]
I_5
I_4
I_3
I_2
out3
out2
out1
out0
Select A Command.

4. In the Waveform Viewer, choose **Object** > **Hide Marker**.
5. Choose Jump > Time=0.
6. Select the output pins **out0-3** one at a time. As you do, notice that the corresponding pin in the Hierarchy Navigator changes color.
7. In the Waveform Viewer, select the **OUT0** signal name. Then choose **Jump** > **Next Change**. The time marker moves to Time=204.4ns.

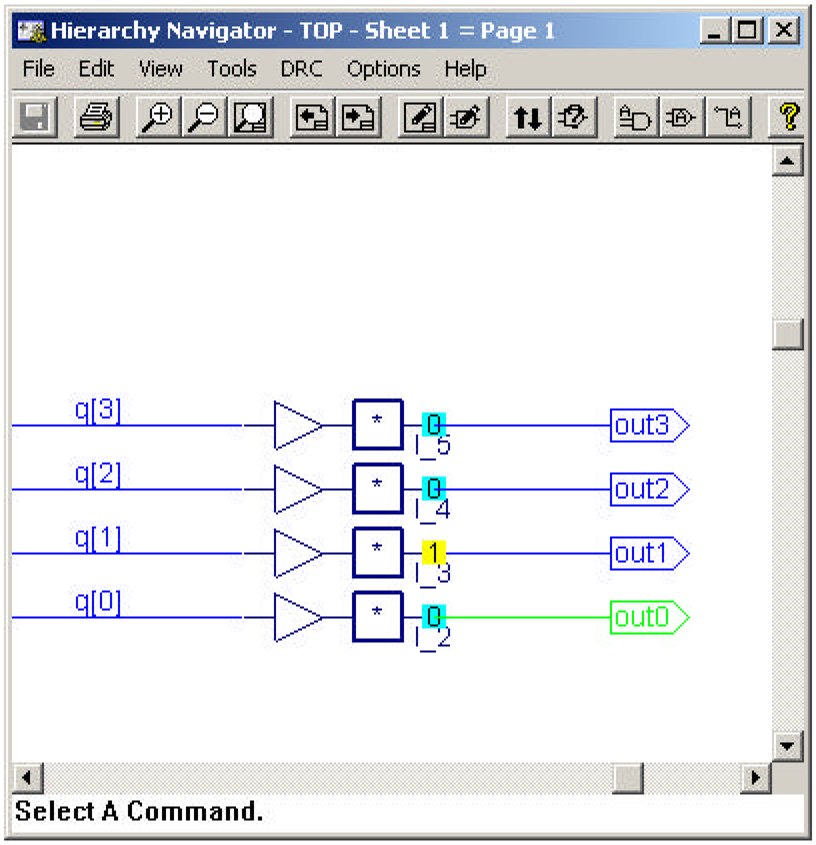
   The logic values determined during simulation are displayed on the schematic loaded in the Hierarchy Navigator. Because you were at time = 0, all the outputs were low (0). However, after you jumped the time marker to the next transition, signal OUT0 changed to high (1).


Hierarchy Navigator - TOP - Sheet 1 = Page 1
File Edit View Tools DRC Options Help
q[3]
q[2]
q[1]
q[0]
0
0
0
1
I_5
I_4
I_3
I_2
out3
out2
out1
out0

Select A Command.

8. Choose **Jump > Next Change** again to move the time marker to time 504.4.

   Now you are on the transition where OUT0 goes back low and OUT1 goes high. As the cursor jumps to different points along the time line, the logic values on the schematic change to those for that simulation time.


Hierarchy Navigator - TOP - Sheet 1 = Page 1
File Edit View Tools DRC Options Help
q[3]
q[2]
q[1]
q[0]
0
I_5
0
I_4
1
I_3
0
I_2
out3
out2
out1
out0
Select A Command.

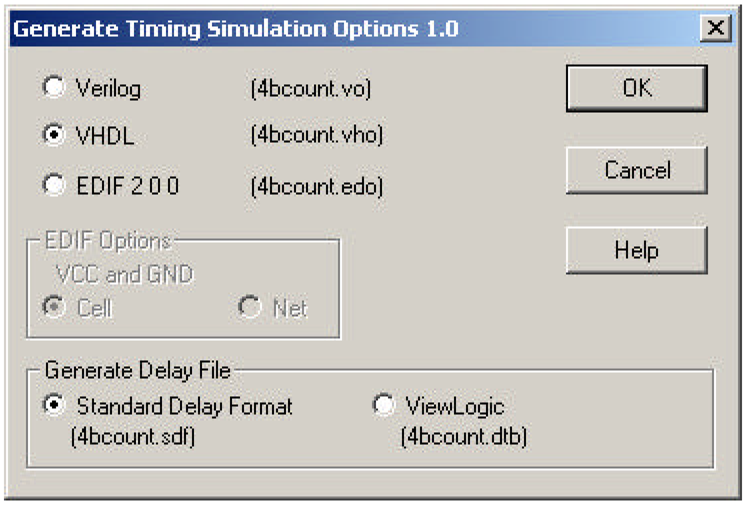
9. Close the Hierarchy Navigator, Waveform Viewer, and Lattice Logic Simulator.

Task 5: Timing Simulation for 3rd Party Simulators

After you have fit the design, the ispLEVER software lets you export the netlist and delays for timing simulation. For netlist files, ispLEVER supports VHDL, EDIF, and Verilog formats. For timing delay files, ispLEVER supports the standard SDF and Viewlogic DTB timing formats.

To generate a 3rd party simulation netlist:

1. In the Project Navigator, choose **Tools > Generate Timing Simulation Options** to open the dialog box.


Generate Timing Simulation Options 1.0

( ) Verilog (4bcount.vo)
(●) VHDL (4bcount.vho)
( ) EDIF 2 0 0 (4bcount.edo)

EDIF Options
VCC and GND
(●) Cell ( ) Net

Generate Delay File
(●) Standard Delay Format (4bcount.sdf)
( ) ViewLogic (4bcount.dtb)

2. Depending on the timing simulator you are using, the ispLEVER software lets you choose a specific format for the output netlist and the timing delay file. Accept the format settings and click **OK** to close the dialog box.
3. In the Project Navigator Sources window, select the target device.
4. In the Project Navigator Processes window, double-click **Generate Timing Simulation Files**.

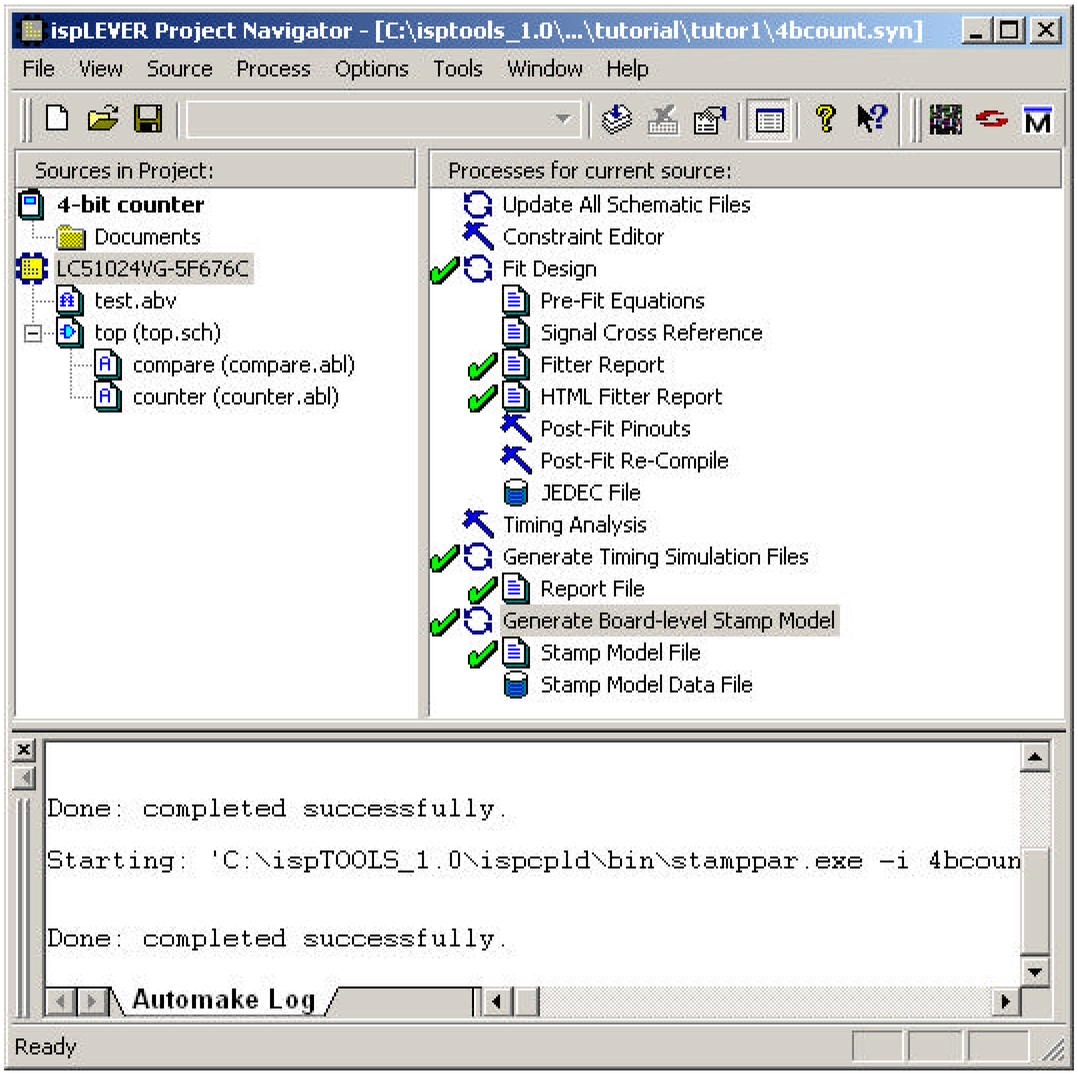
   The ispLEVER software generates a report file, which contains a summary of the design as well as the name and type of netlist file and delay file. Your timing simulator can read these files.
5. Right-click on the **Report File** process and choose **View** to view the results generated from the Generate Timing Simulation Files process in the Report Viewer.
6. Close the Report Viewer when you are finished viewing the report.

Task 6: Board-level Static Timing Analysis

You can use the Lattice Stamp Model Generator to build board-level Stamp Models of a design using any Lattice Semiconductor device for 3rd party board-level static timing analysis tools. A board-level static timing analysis tool enables you to manage large, high performance designs while minimizing development time. With the Stamp models, you accelerate board-level design.

To generate Stamp Model files:

1. In the Project Navigator, select the target device, and then double-click the **Generate Board-level Stamp Model** process. The software generates two files: Stamp Model File and Stamp Model Data File.



```
Done: completed successfully.
Starting: 'C:\ispTOOLS_1.0\ispcpld\bin\stamppar.exe -i 4bcoun

Done: completed successfully.
```

2. Right-click on these processes and choose **View** to open the files in the Report Viewer.
3. Close the Report Viewer when you are finished viewing the files.

Congratulations

You have finished the Schematic and ABEL-HDL Design tutorial. In this tutorial, you have learned how to:

- Setup an ispLEVER schematic project
- Create a top-level schematic source
- Create and import ABEL-HDL sources into the project
- Navigate the design
- Import a test vector file and run functional simulation
- Set pin constraints, fit the design, and read the Fitter report
- Run timing analysis and simulation and analyze the results

- Correlate simulation results using cross-probing
- Export the netlist and delays for timing simulation
- Build board-level Stamp models of a design

Tutorial 2: HDL Synthesis Design with Synplify

This tutorial shows you how to use Synplify from within ispLEVER to synthesize a VHDL design and generate an EDIF file for a Lattice CPLD device.

***Note**: If you want to learn how to use Synplify in standalone mode, or understand more about its advanced features, please see the Third-Party Manuals online documentation by choosing **Help > Manuals** from the ispLEVER Project Navigator.*

Learning Objectives

When you have completed this tutorial, you should be able to:

- Create a new EDIF project in the ispLEVER system and target a device.
- Launch Synplify from within ispLEVER and generate an EDIF netlist file.
- Import the EDIF file into the ispLEVER system, fit the design, generate a JEDEC file, and view the Fitter report.
- Run static timing analysis using the Performance Analyst and view the results.

Time to Complete This Tutorial

The time to complete this tutorial is about 20 minutes.

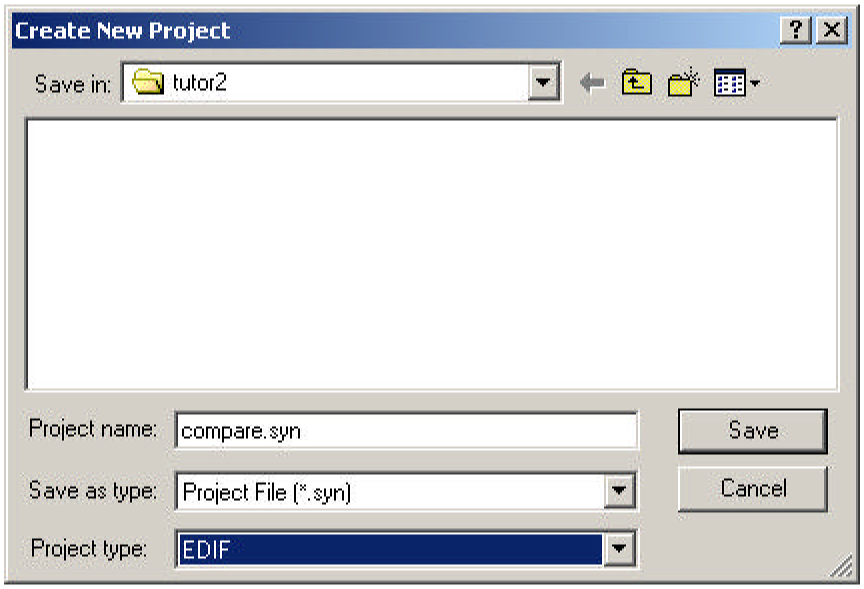
Task 1: Create a New Project

To begin a new project, you must create a project directory. Then you need to give the project file a name (`.syn`) and declare the project type (EDIF).

The ispLEVER software saves an initial design file with the `.syn` file extension in the directory you specify. All project files are copied to or created in this directory. The project type specifies that all design sources will be of this type.

To create a new project:

1. Start the ispLEVER system, if it is not already running.
2. In the Project Navigator, choose **File** > **New Project** to open the Create New Project dialog box.
3. In the dialog box,
   - Change to the directory: `<`*`install path`*`>\ispcpld\examples\tutorial\`**`tutor2.`**
   - In the Project name box, type **`compare.`**
   - In the Project type box, select **EDIF**.
   - Click **Save**.

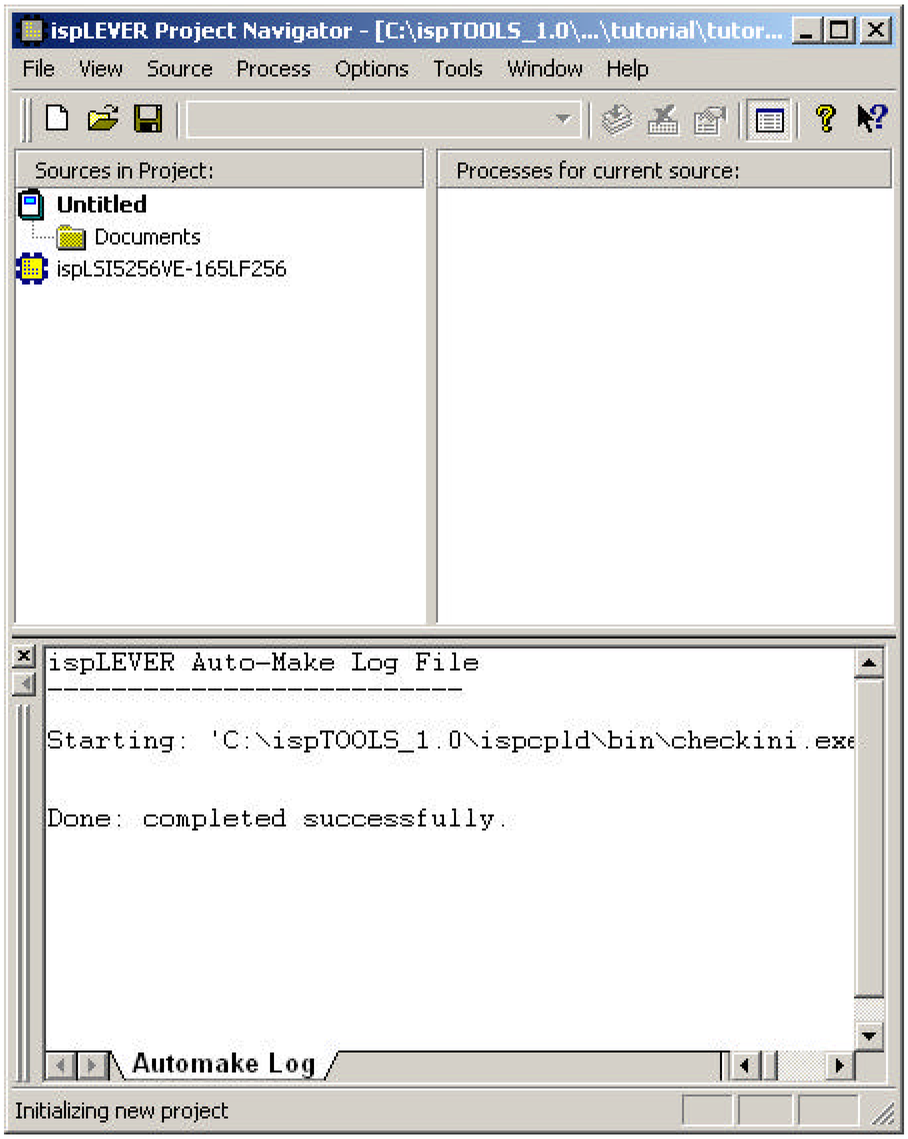

Create New Project

Save in: tutor2

Project name: compare.syn

Save as type: Project File (*.syn)

Project type: EDIF


ispLEVER Project Navigator - [C:\ispTOOLS_1.0\...\tutorial\tutor...
File View Source Process Options Tools Window Help
Sources in Project:
Untitled
Documents
ispLSI5256VE-165LF256
Processes for current source:
Automake Log

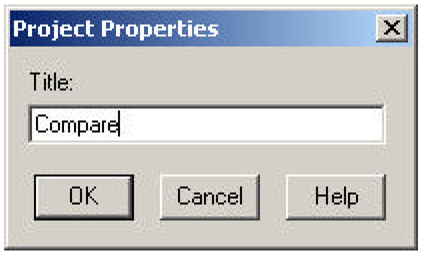
```
ispLEVER Auto-Make Log File
---------------------------

Starting: 'C:\ispTOOLS_1.0\ispcpld\bin\checkini.exe

Done: completed successfully.
```

Initializing new project

4. The default project title, `Untitled`, appears in the Sources window of the Project Navigator. Double-click the project title (**`Untitled`**) to open the Project Properties dialog box.

   The default title for a new project is "Untitled." You can create a title for the project with as many characters as you want. The title can contain spaces and any other keyboard character except tabs and returns.

5. Type **`Compare`** as your project title and click **OK**.



Title:

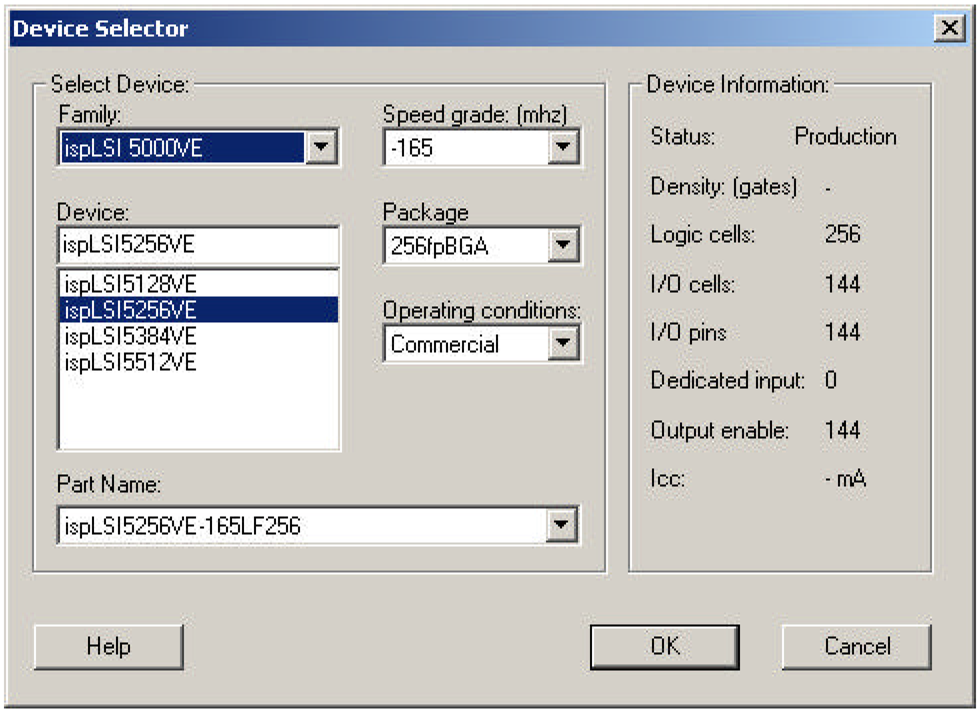
Compare

Task 2: Target a Device

In the Project Navigator Sources window is the device icon next to the target device for the project. The Project Navigator lets you target a design to a specific Lattice device at any time during the design process. The default device is the ispLSI5256VE-165LF256. For this project, you will target a different device.

To view the list of available devices and to change the target device:

1. In the Sources window, double-click the device name to open the Device Selector dialog box. The dialog box shows the default device as well as all available devices and their options.



Device Selector

Select Device:

Family: ispLSI 5000VE

Speed grade: (mhz) -165

Device: ispLSI5256VE

ispLSI5128VE
ispLSI5256VE
ispLSI5384VE
ispLSI5512VE

Package 256fpBGA

Operating conditions: Commercial

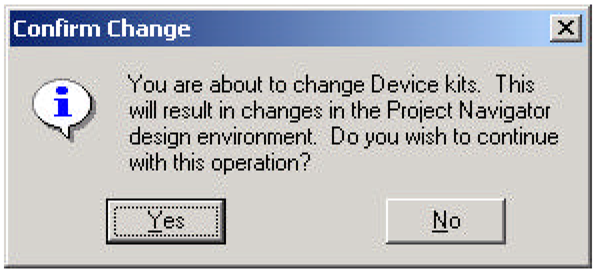
Part Name: ispLSI5256VE-165LF256

Device Information:

Status:	Production
Density: (gates)	-
Logic cells:	256
I/O cells:	144
I/O pins	144
Dedicated input:	0
Output enable:	144
Icc:	- mA

- Under Family, select **ispMACH 5000VG** from the drop-down list.
- Accept the default settings and click **OK**.

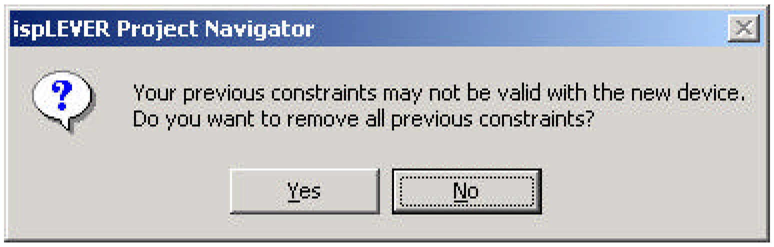
2. In the Confirm Change dialog box, click **Yes** to confirm that you wish to change device kits.



Confirm Change

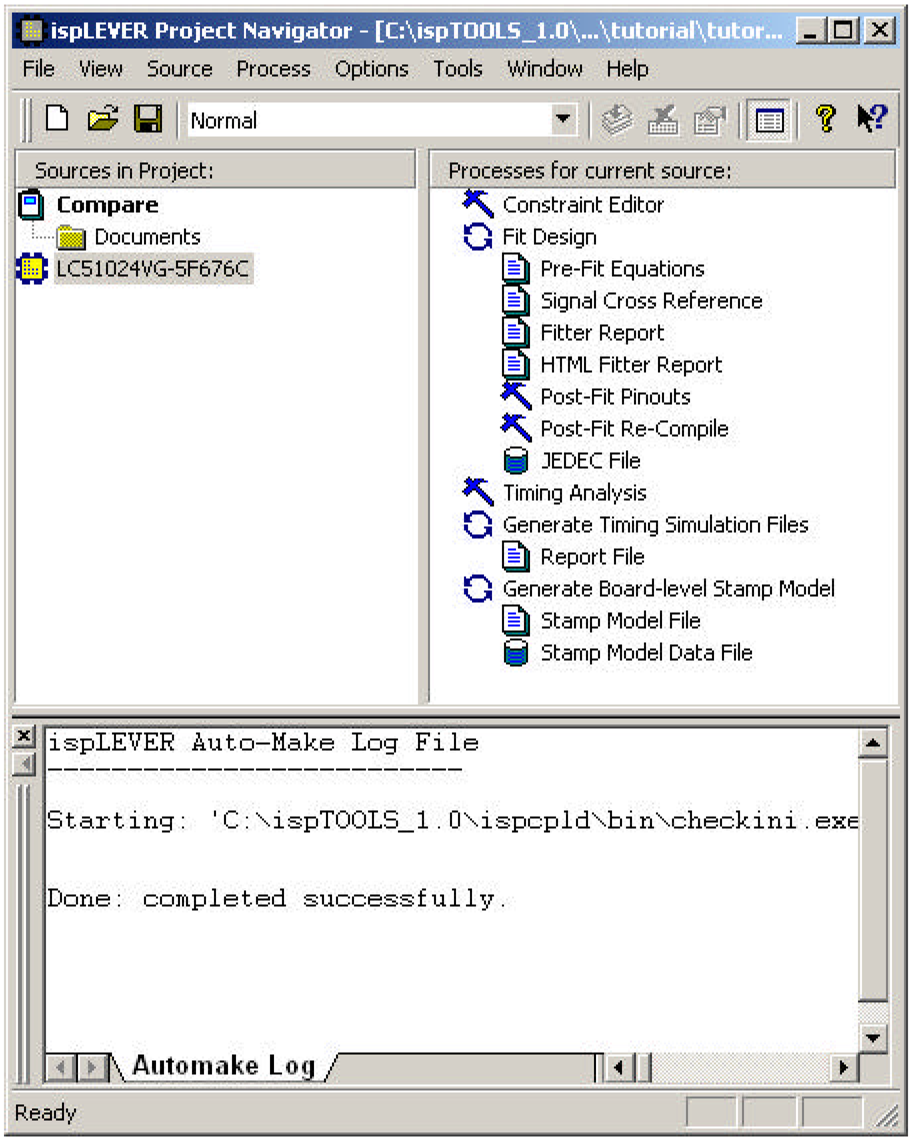
You are about to change Device kits. This will result in changes in the Project Navigator design environment. Do you wish to continue with this operation?

3. In the next dialog box, click **No**.



Your previous constraints may not be valid with the new device.
Do you want to remove all previous constraints?

4. Your Project Navigator should look like this:



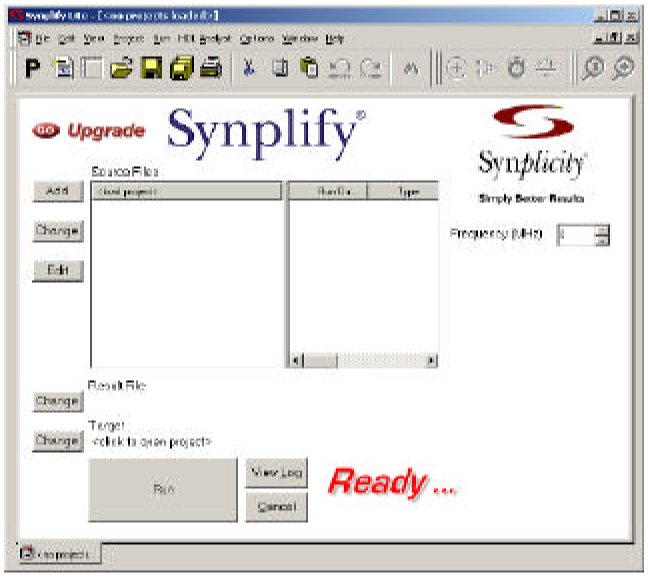
```
ispLEVER Auto-Make Log File
---------------------------

Starting: 'C:\ispTOOLS_1.0\ispcpld\bin\checkini.exe

Done: completed successfully.
```

Task 3: Create a Synplify Project

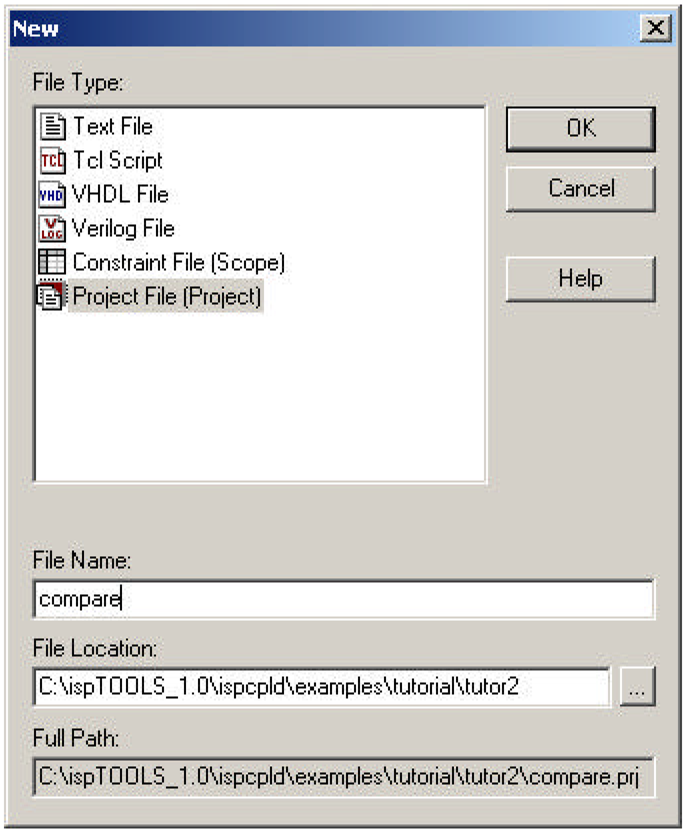
Synplify for Lattice is a logic synthesis tool for CPLDs, developed by Synplicity. Synplify starts with high-level designs written in Verilog or VHDL hardware description languages (HDLs). Using proprietary Behavior Extracting Synthesis Technology (B.E.S.T.) the tool converts the HDL into small, high-performance, design netlists that are optimized for Lattice devices.

To start Synplify:

1. In the Project Navigator, choose **Tools** > **Synplify Synthesis** to open the Synplify synthesis tool in the Project view.



2. Choose **File** > **New** to open the dialog box. In the dialog box, do the following:
   - Under File Type, select **Project File**
   - Under File Name, type **`compare`**
   - Under File Location, make sure you are in the **tutor2** folder
   - Click **OK**



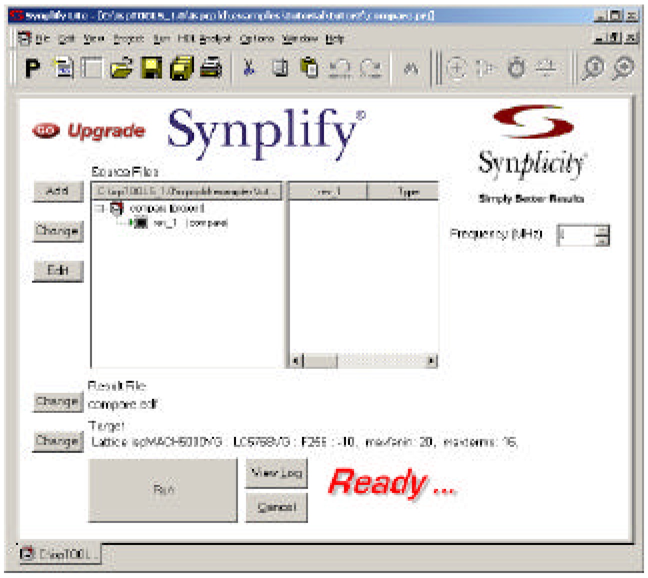
File Type:
- Text File
- Tcl Script
- VHDL File
- Verilog File
- Constraint File (Scope)
- Project File (Project)

File Name: compare

File Location: C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor2

Full Path: C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor2\compare.prj

3. The Synplify screen should look like this:



Source Files

compare (project)

rev_1 (compare)

Frequency (MHz)

Result File: compare.edf

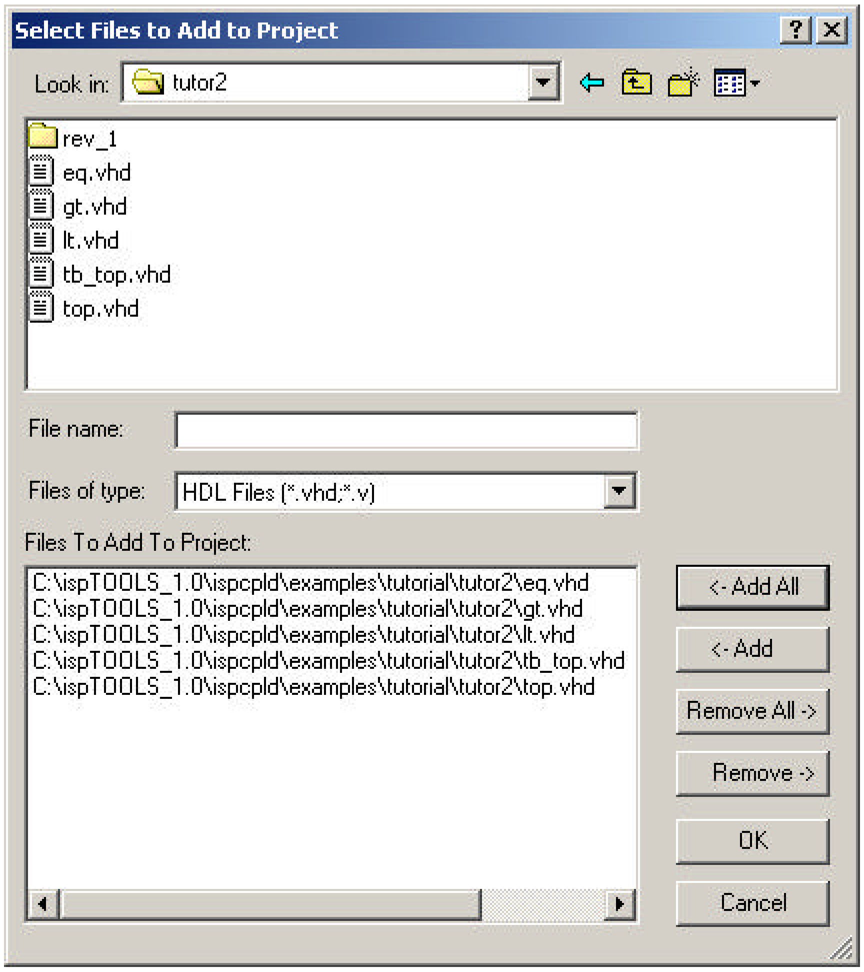
Target: Lattice ispMACH5000VG : LC5768VG : F256 : -10, maxfanin: 20, maxterms: 16

Task 4: Add the VHDL Source Files

In this task you will add VHDL source files to the project.

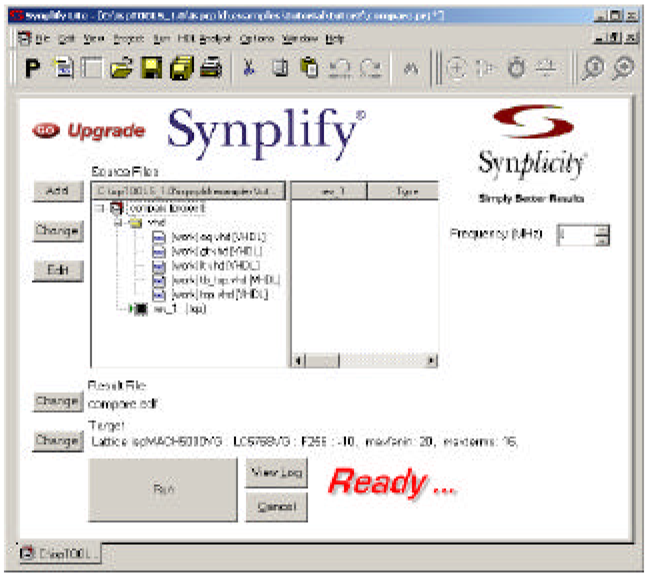
To add source files to the project:

1. On the Synplify main window, click **Add** to open the dialog box. You should see four VHDL files (`.vhd`).
2. Click **Add All** to add all of the VHDL files to the project (shown in the Files To Add To Project field).



```
C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor2\eq.vhd
C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor2\gt.vhd
C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor2\lt.vhd
C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor2\tb_top.vhd
C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor2\top.vhd
```

3. Click **OK**. The Synplify screen should look like this:



Source Files

compare (project)
vhd
[work] eq.vhd [VHDL]
[work] gt.vhd [VHDL]
[work] lt.vhd [VHDL]
[work] tb_top.vhd [VHDL]
[work] top.vhd [VHDL]
rev_1 (top)

Frequency (MHz)

Result File
compare.edf

Target
Lattice ispMACH5000VG : LC5768VG : F256 : -10, maxfanin: 20, maxterms: 16,

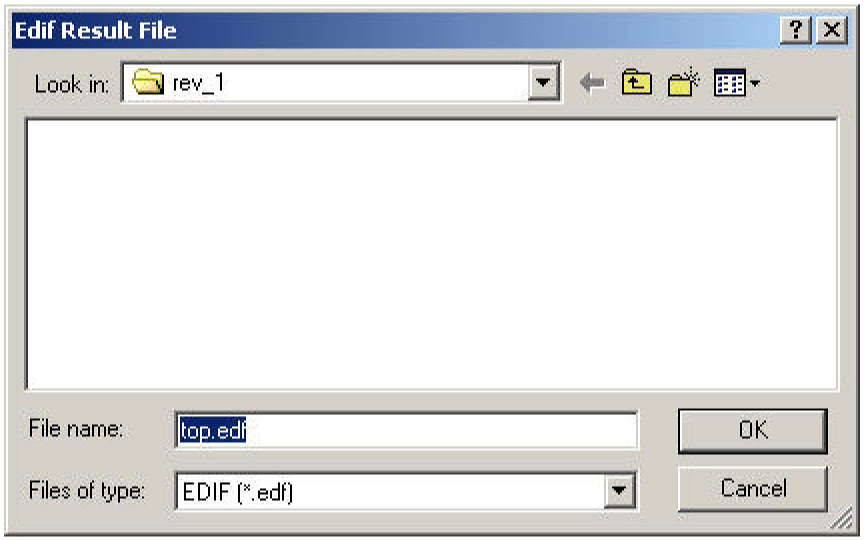
***Note**: `top.vhd` must be at the bottom of the list of VHDL files for the project in the Synplicity GUI because Synplicity only compiles the module at the bottom of the list. For example, if `eq.vhd` were at the bottom of the list, then the EDIF file will only contain the logic for `eq.vhd` because it does not call any other modules.*

Task 5: Set Implementation Options

Synplify has probably set your implementation options correctly. However, it is good practice to check them.

To set the implementation options:

1. Click **Change** (**Results File)** to open the dialog box. It should look like this:

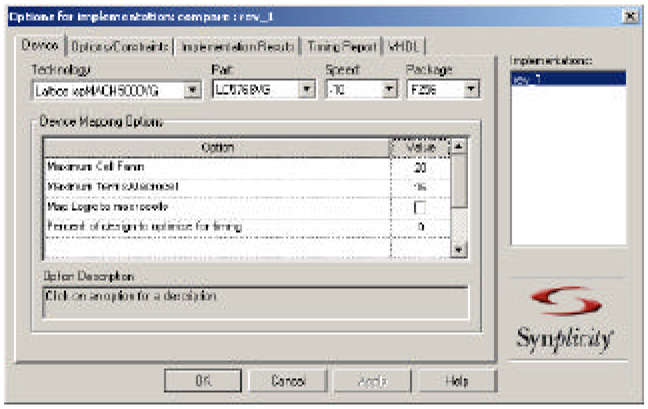


Look in: rev_1

File name: top.edf

Files of type: EDIF (*.edf)

2. Click **Cancel** to close the dialog box.
3. Click **Change** (**Target)** to open the dialog box. The Device tab should look like this:



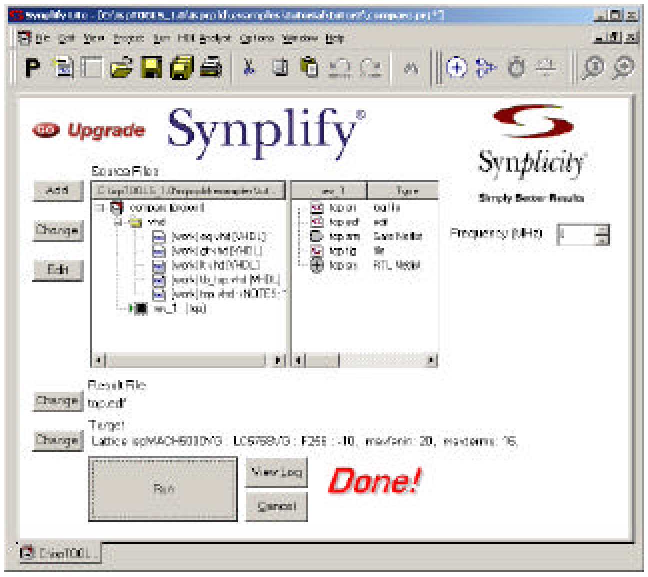
Technology: Lattice ispMACH5000VG
Part: LC5768VG
Speed: -10
Package: F256

Device Mapping Options

Option	Value
Maximum Cell Fanin	20
Maximum Terms/Macrocell	16
Map Logic to macrocells	☐
Percent of design to optimize for timing	0

Option Description
Click on an option for a description

4. Click **Cancel** to close the dialog box.
5. Click **Run** to start the synthesis process. Synplify synthesizes the VHDL design and creates an EDIF file, as well as several other files, and displays them in the window on the right. If you like, you can double-click the different files and view them. When you are finished, close the files.



Source Files
compare (project)
vhd
[work] eq.vhd (VHDL)
[work] gt.vhd (VHDL)
[work] lt.vhd (VHDL)
[work] tb_top.vhd (VHDL)
[work] top.vhd (NOTES)
rev_1 (top)

rev_1	Type
top.srr	log file
top.edf	edif
top.srm	Srs Netlist
top.tlg	file
top.srs	RTL Model

Frequency (MHz)

Result File
top.edf

Target
Lattice ispMACH5000VG : LC5768VG : F256 : -10, maxfanin: 20, maxterms: 16,

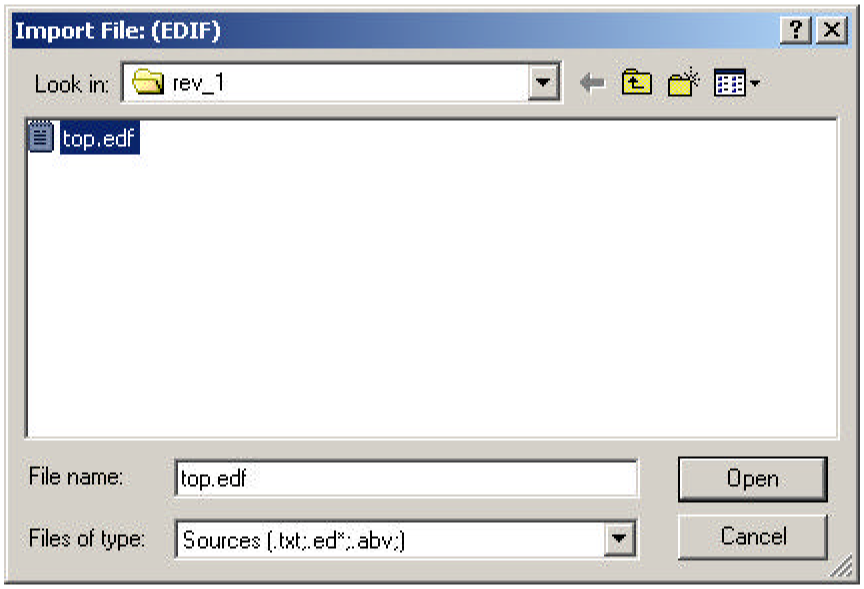
6. Choose **File** > **Save** to save the Synplify project.
7. Now you are ready to import the EDIF file into ispLEVER project. You can either minimize Synplify or close it.

Task 6: Import the EDIF File into Your Project

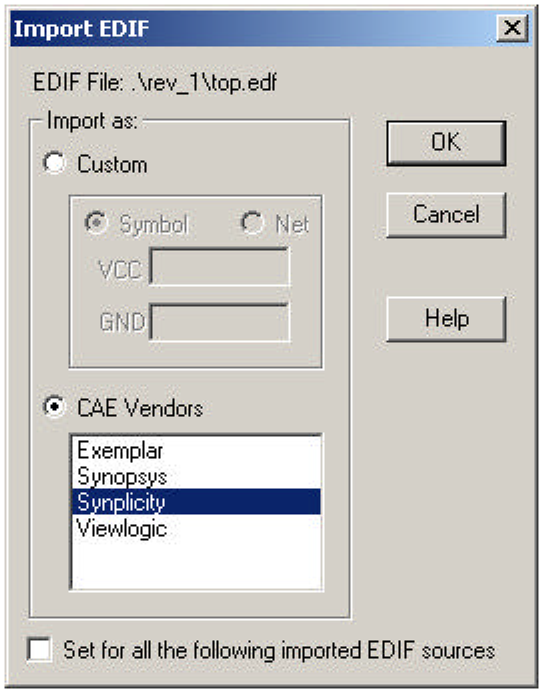
You can import EDIF 2.0.0 netlists from third-party synthesis tools, such as Synplicity Synplify, into ispLEVER.

To import an EDIF netlist into your project:

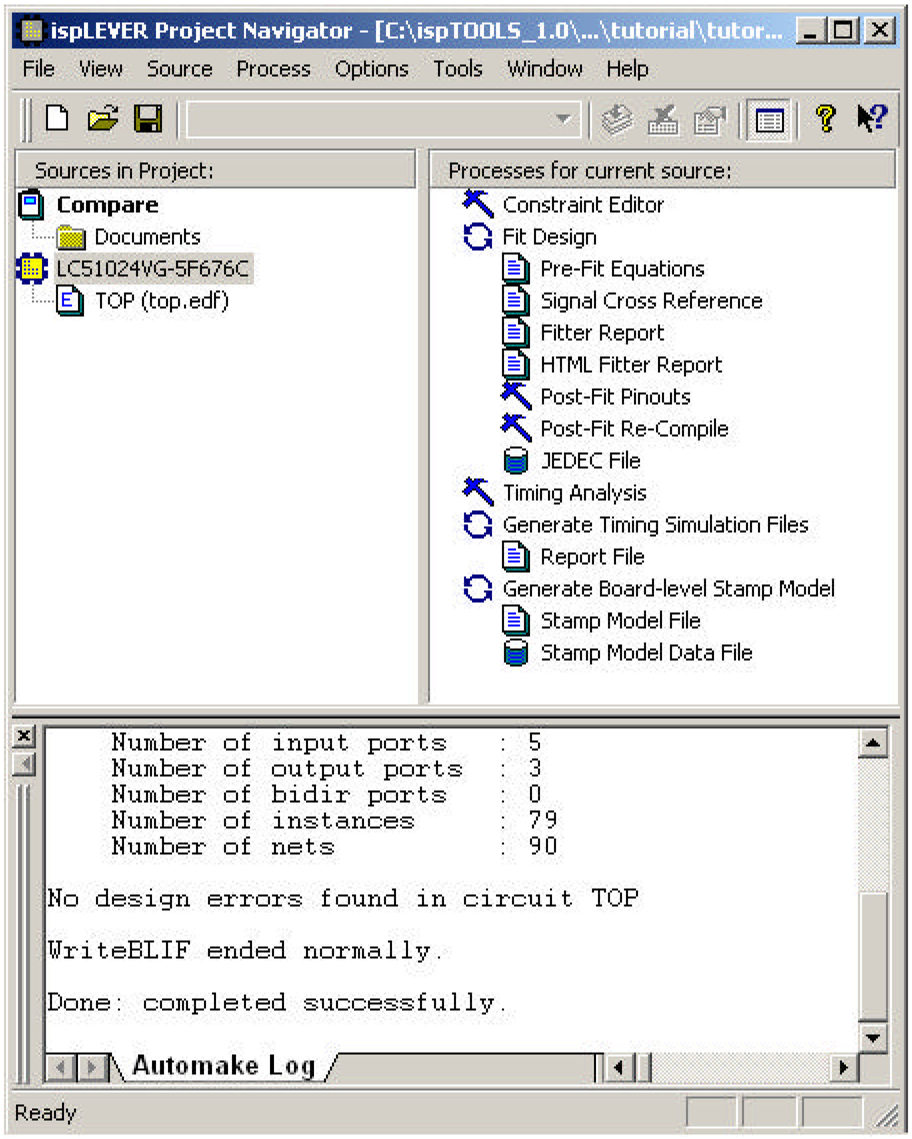
1. In the ispLEVER Project Navigator, choose **Source** > **Import** to open the Import File dialog box.
2. Go to the **rev_1** folder, select **`top.edf`**, and then click **Open** to open the Import EDIF dialog box.



3. Under CAE Vendors, select **Synplicity** and click **OK**.



The software adds the selected EDIF file (`top.edf`) to the project sources.



```
    Number of input ports    : 5
    Number of output ports   : 3
    Number of bidir ports    : 0
    Number of instances      : 79
    Number of nets           : 90

No design errors found in circuit TOP

WriteBLIF ended normally.

Done: completed successfully.
```

***Note**: After you import an EDIF file into the ispLEVER project, it is always linked to the Project Navigator. Therefore, if you make changes and recompile your HDL file to create a new EDIF file, your project is automatically updated as well.*

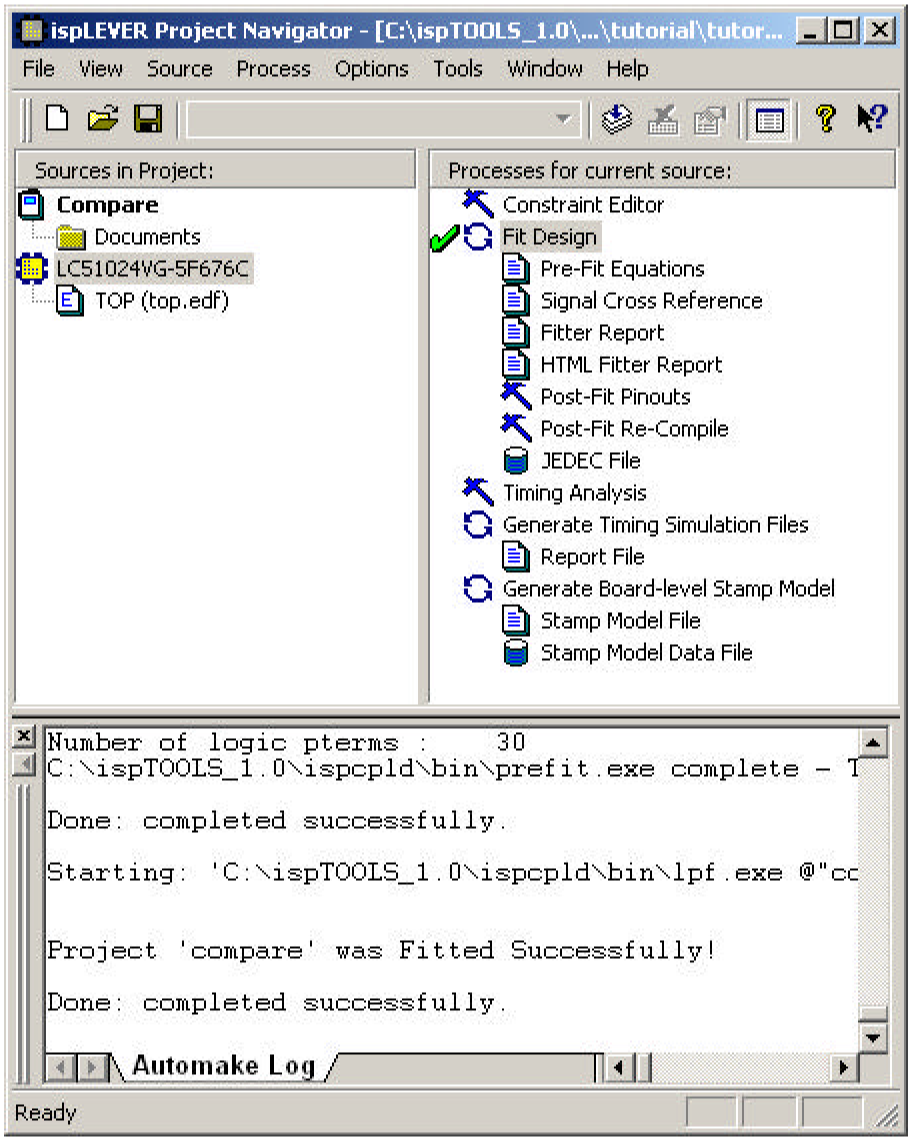
Task 7: Fit the Design and View the Report

The ispLEVER software has a single user interface with all options preset to deliver the highest possible push-button performance for most devices. At the end of a successful fitter run, the ispLEVER software generates a JEDEC file, as well as a fitter report so that you can see how the ispLEVER software has utilized and routed the part.

To run the Fitter and view the report:

1. With the target device selected in the Sources window, double-click **Fit Design** in the Processes window to run the Fitter. The ispLEVER software successfully fits the design in the specified device and generates a JEDEC file.

Optional*: If you like, you can right-click on the* ***JEDEC File*** *process and select* ***View*** *to look at the contents of the JEDEC file. Close the file when you are through.*



```
Number of logic pterms :    30
C:\ispTOOLS_1.0\ispcpld\bin\prefit.exe complete - T

Done: completed successfully.

Starting: 'C:\ispTOOLS_1.0\ispcpld\bin\lpf.exe @"cc


Project 'compare' was Fitted Successfully!

Done: completed successfully.
```

2. Double-click on the **HTML Fitter Report** process to open the report in your browser. View the contents and then close the report.

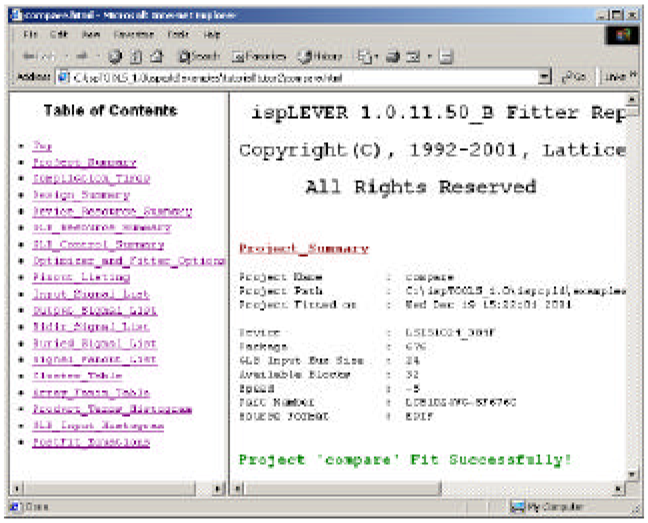


Table of Contents

- Top
- Project_Summary
- Compilation_Times
- Design_Summary
- Device_Resource_Summary
- GLB_Resource_Summary
- GLB_Control_Summary
- Optimizer_and_Fitter_Options
- Pinout_Listing
- Input_Signal_List
- Output_Signal_List
- Bidir_Signal_List
- Buried_Signal_List
- Signal_Fanout_List
- Cluster_Table
- Array_Terms_Table
- Product_Terms_Histogram
- GLB_Input_Histogram
- PostFit_Equations

```
ispLEVER 1.0.11.50_B Fitter Rep
Copyright(C), 1992-2001, Lattice
All Rights Reserved

Project_Summary

Project Name          :  compare
Project Path          :  C:\ispTOOLS_1.0\ispcpld\examples
Project Fitted on     :  Wed Dec 19 15:22:04 2001

Device                :  LSI5I024_384F
Package               :  676
GLB Input Mux Size    :  24
Available Blocks      :  32
Speed                 :  -5
Part Number           :  LC5I024VG-5F676C
Source Format         :  EDIF

Project 'compare' Fit Successfully!
```

Task 8: Run Static Timing Analysis

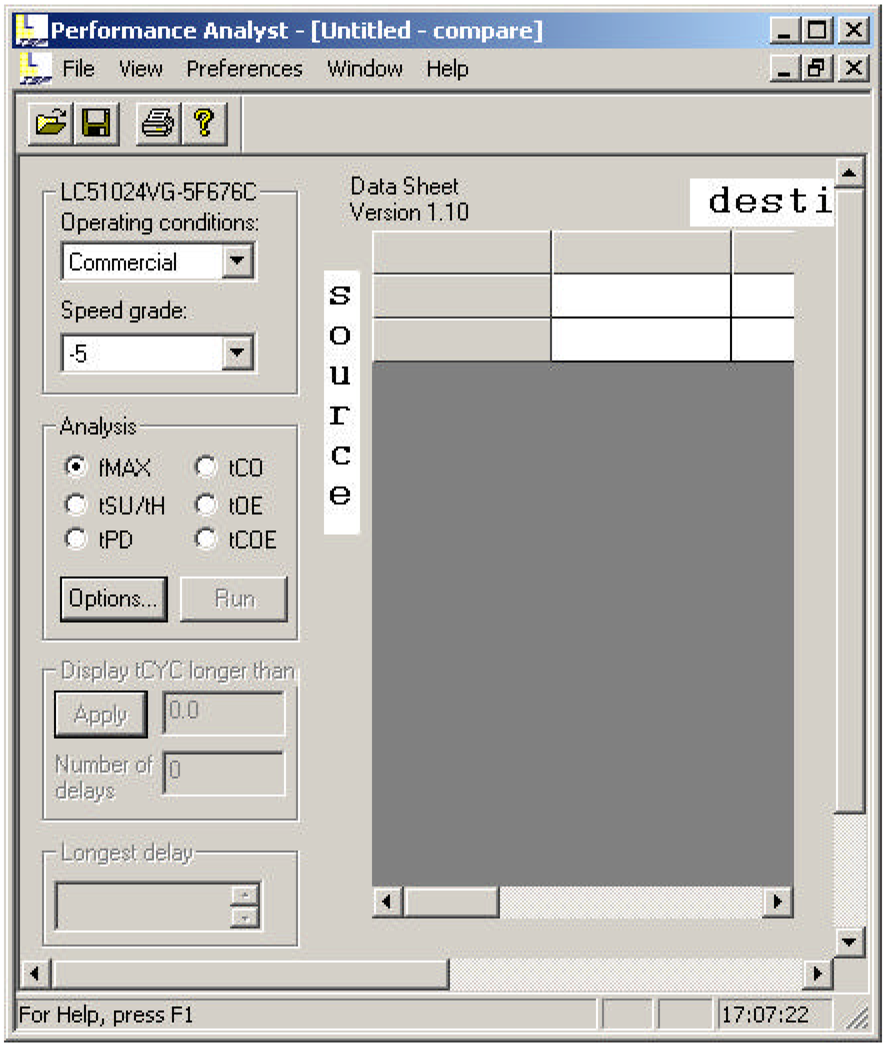
Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data such as the critical path, setup/hold time requirements, and the maximum frequency.

The Performance Analyst traces each logical path in the design and calculates the path delays using the device's timing model and worst-case AC specs supplied in the device data sheet.

The timing analysis results are displayed in a graphical spreadsheet with source signals displayed on the vertical axis and destination signals displayed on the horizontal axis. The worst-case delay value is displayed in a spreadsheet cell if there is at least one delay path between the source and destination. To more easily identify performance bottlenecks, you can double-click a cell to view the path delay details.

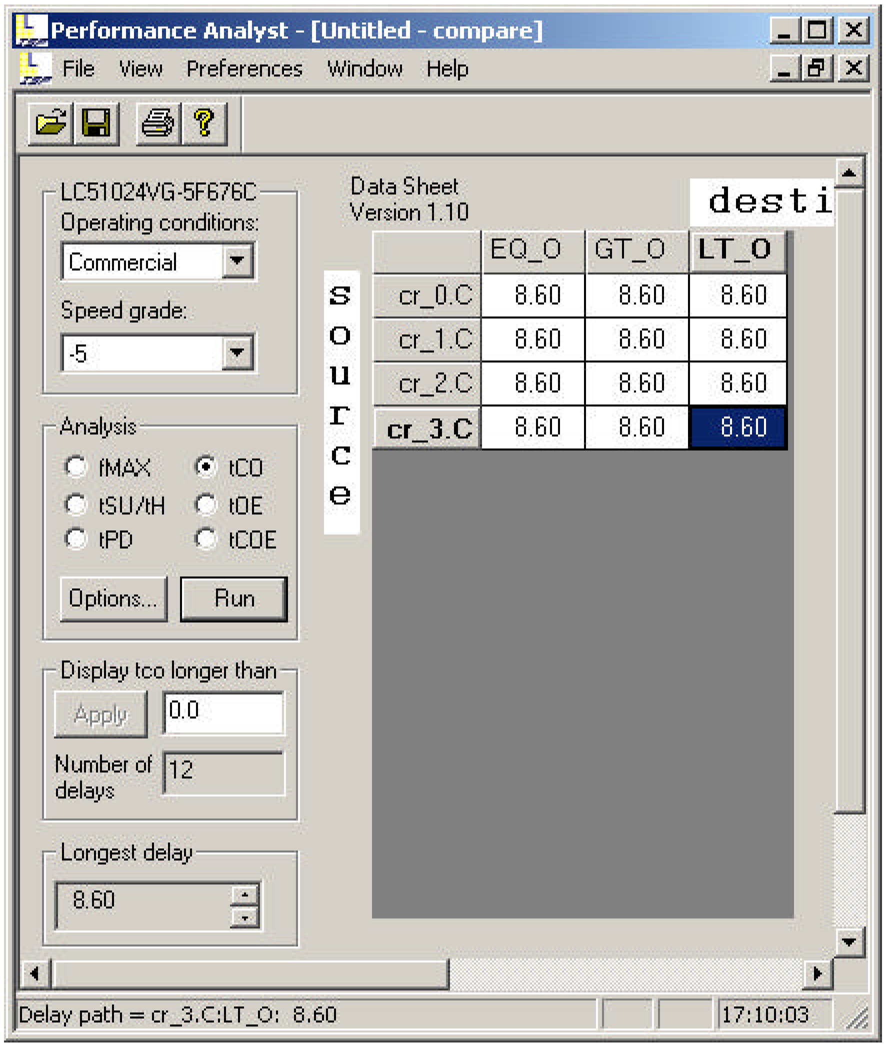
To run timing analysis:

1. In the Project Navigator Sources window, select the **target device**. In the Processes window, double-click the **Timing Analysis** process to open the Performance Analyst.



2. The Performance Analyst performs six distinct analysis types: fMAX, tSU/tH, tPD, tCO, tOE, and tCOE. The first type, fMAX, is an internal register-to-register delay analysis. fMAX measures the maximum clock operating frequency, limited by worst-case register-to-register delay. The remaining five types are external pin-to-pin delay analysis. Timing threshold filters, source and destination filters, and path filters can be used to independently fine-tune each analysis.

   Under Analysis, select **tCO** and then click **Run**. The tCO path trace analysis reports clock-to-out delay starting from the primary input, going through the clock of flip-flops or gate of latches, and ending at the primary output. In this case it is 8.60ns.


Performance Analyst - [Untitled - compare]
File View Preferences Window Help
Options...
Run
Apply

LC51024VG-5F676C
Operating conditions: Commercial
Speed grade: -5

Analysis
( ) fMAX [✓ tCO]
( ) tSU/tH ( ) tOE
( ) tPD ( ) tCOE

Display tco longer than: 0.0
Number of delays: 12

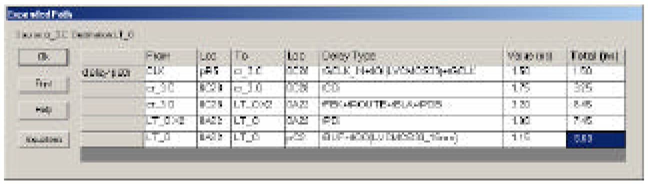
Longest delay: 8.60

Data Sheet
Version 1.10

source \ desti	EQ_O	GT_O	LT_O
cr_0.C	8.60	8.60	8.60
cr_1.C	8.60	8.60	8.60
cr_2.C	8.60	8.60	8.60
cr_3.C	8.60	8.60	8.60

Delay path = cr_3.C:LT_O: 8.60 17:10:03

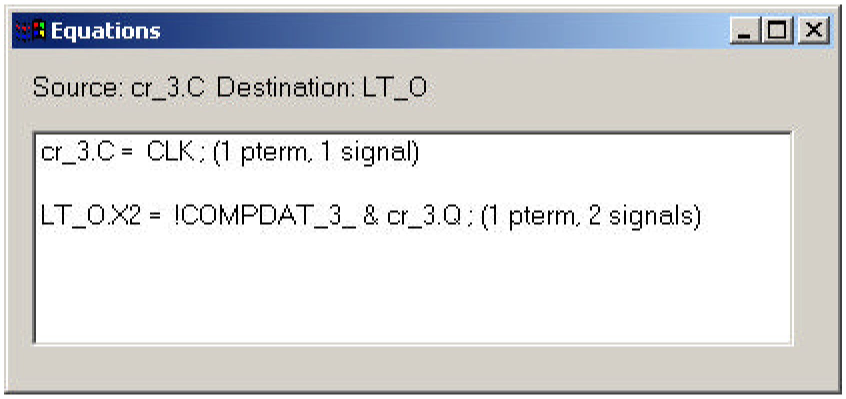
3. Double-click the highlighted cell (**8.60**) in the spreadsheet window to open the Expanded Path dialog box. This dialog lets you analyze individual timing components used to calculate the timing path. There is a source pin (From) and a destination pin (To). Also shown is the delay type, the delay of that path (Value ns), and the cumulative delay of all the signals.



Source cr_3.C Destination LT_O

	From	Loc	To	Loc	Delay Type	Value (ns)	Total (ns)
delay-path	CLK	[illegible]	cr_3.C	[illegible]	[illegible]	1.50	1.50
	cr_3.C	[illegible]	cr_3.Q	[illegible]	tCO	1.75	3.25
	cr_3.Q	[illegible]	LT_OX2	[illegible]	[illegible]	3.20	6.45
	LT_OX2	[illegible]	LT_O	[illegible]	[illegible]	1.00	7.45
	LT_O	[illegible]	LT_O	[illegible]	[illegible]	1.15	8.60

4. Click **Equations** to open the Equations dialog box, which shows the functional relationship between the selected source/destination.



Source: cr_3.C Destination: LT_O

```
cr_3.C = CLK ; (1 pterm, 1 signal)

LT_O.X2 = !COMPDAT_3_ & cr_3.Q ; (1 pterm, 2 signals)
```

5. Close the Performance Analyst without saving.
6. Close Synplify, and then close ispLEVER without saving.

Congratulations

You have completed the HDL Synthesis Design with Synplify tutorial. In this tutorial you have learned how to:

- Create a new EDIF project in the ispLEVER system and target a device.
- Launch Synplify from within ispLEVER and generate an EDIF netlist file.
- Import the EDIF file into the ispLEVER system, fit the design, generate a JEDEC file, and view the Fitter report.
- Run static timing analysis using the Performance Analyst and view the results.

Tutorial 3: HDL Synthesis Design with LeonardoSpectrum

This tutorial shows you how to use LeonardoSpectrum from within ispLEVER to synthesize a Verilog design and generate an EDIF file for a Lattice CPLD device.

***Note**: If you want to learn how to use LeonardoSpectrum in standalone mode, or understand more about its advanced features, please see the Third-Party Manuals online documentation by choosing **Help > Manuals** from the ispLEVER Project Navigator.*

Learning Objectives

When you have completed this tutorial, you should be able to:

- Create a new EDIF project in the ispLEVER system and target a device.
- Launch LeonardoSpectrum from within ispLEVER and generate an EDIF netlist file using the Quick Setup tab flow.
- Import the EDIF file into the ispLEVER system, fit the design, generate a JEDEC file, and view the Fitter report.
- Run static timing analysis using the Performance Analyst and view the results.

Time to Complete This Tutorial

The time to complete this tutorial is about 20 minutes.

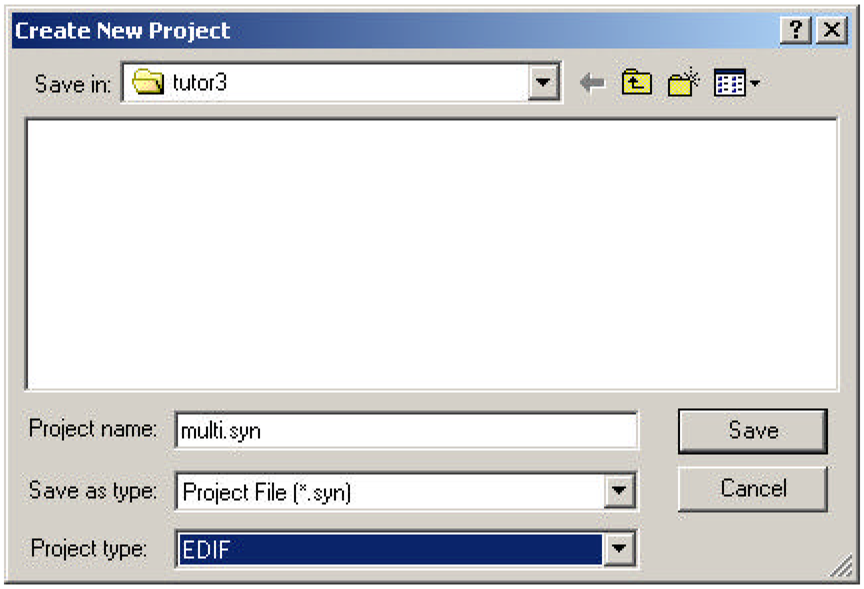
Task 1: Create a New Project

To begin a new project, you must create a project directory. Then you need to give the project file a name (`.syn`) and declare the project type (EDIF).

The ispLEVER software saves an initial design file with the `.syn` file extension in the directory you specify. All project files are copied to or created in this directory. The project type specifies that all design sources will be of this type.

To create a new project:

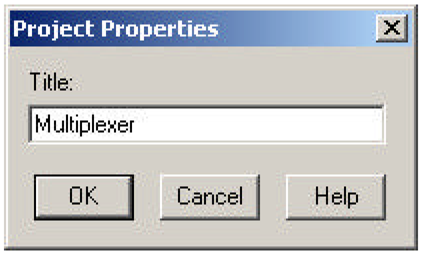
1. Start the ispLEVER system, if it is not already running.
2. In the Project Navigator, choose **File** > **New Project** to open the Create New Project dialog box.
3. In the dialog box,
   - Change to the directory: `<`*`install path`*`>\ispcpld\examples\tutorial\`**`tutor3`**.
   - In the Project name box, type **`multi`**.
   - In the Project type box, select **EDIF**.
   - Click **Save**.



4. The default project name, `Untitled`, appears in the Sources window of the Project Navigator. Double-click the project title (**`Untitled`**) to open the Project Properties dialog box.

   The default title for a new project is "Untitled." You can create a title for the project with as many characters as you want. The title can contain spaces and any other keyboard character except tabs and returns.

5. Type `Multiplexer` as your project title and click **OK**.



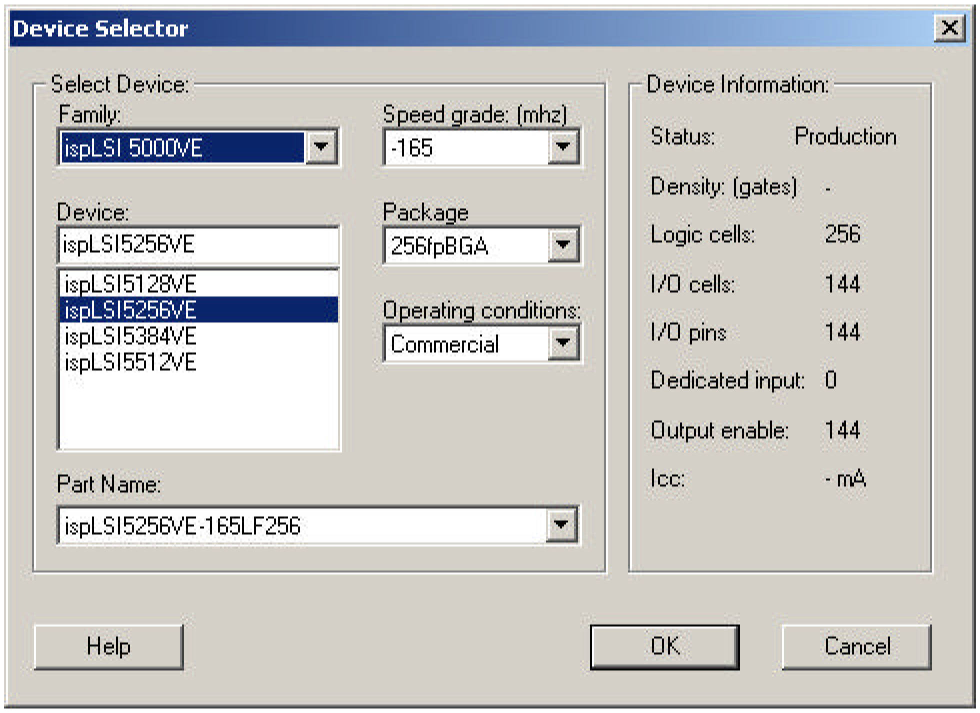
Title: Multiplexer

Task 2: Target a Device

In the Project Navigator Sources window is the device icon next to the target device for the project. The Project Navigator lets you target a design to a specific Lattice device at any time during the design process. The default device is the ispLSI 5000VE. For this project, you will target a different device.

To view the list of available devices and to change the target device:

1. In the Sources window, double-click the part name to open the Device Selector dialog box. The dialog box shows the default device as well as all available devices and their options.

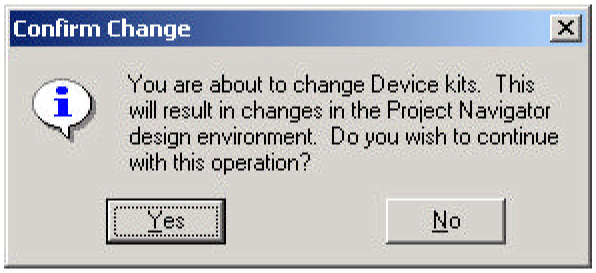


Family: ispLSI 5000VE
Speed grade: (mhz) -165
Device: ispLSI5256VE
ispLSI5128VE
ispLSI5256VE
ispLSI5384VE
ispLSI5512VE
Package 256fpBGA
Operating conditions: Commercial
Part Name: ispLSI5256VE-165LF256

Status:	Production
Density: (gates)	-
Logic cells:	256
I/O cells:	144
I/O pins	144
Dedicated input:	0
Output enable:	144
Icc:	- mA

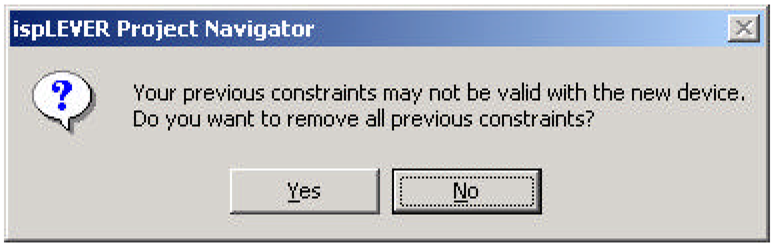
- Under Family, select **ispMACH 5000VG** from the drop-down list.
- Accept the default settings and click **OK**.

2. In the Confirm Change dialog box, click **Yes** to confirm that you wish to change device kits.


Confirm Change

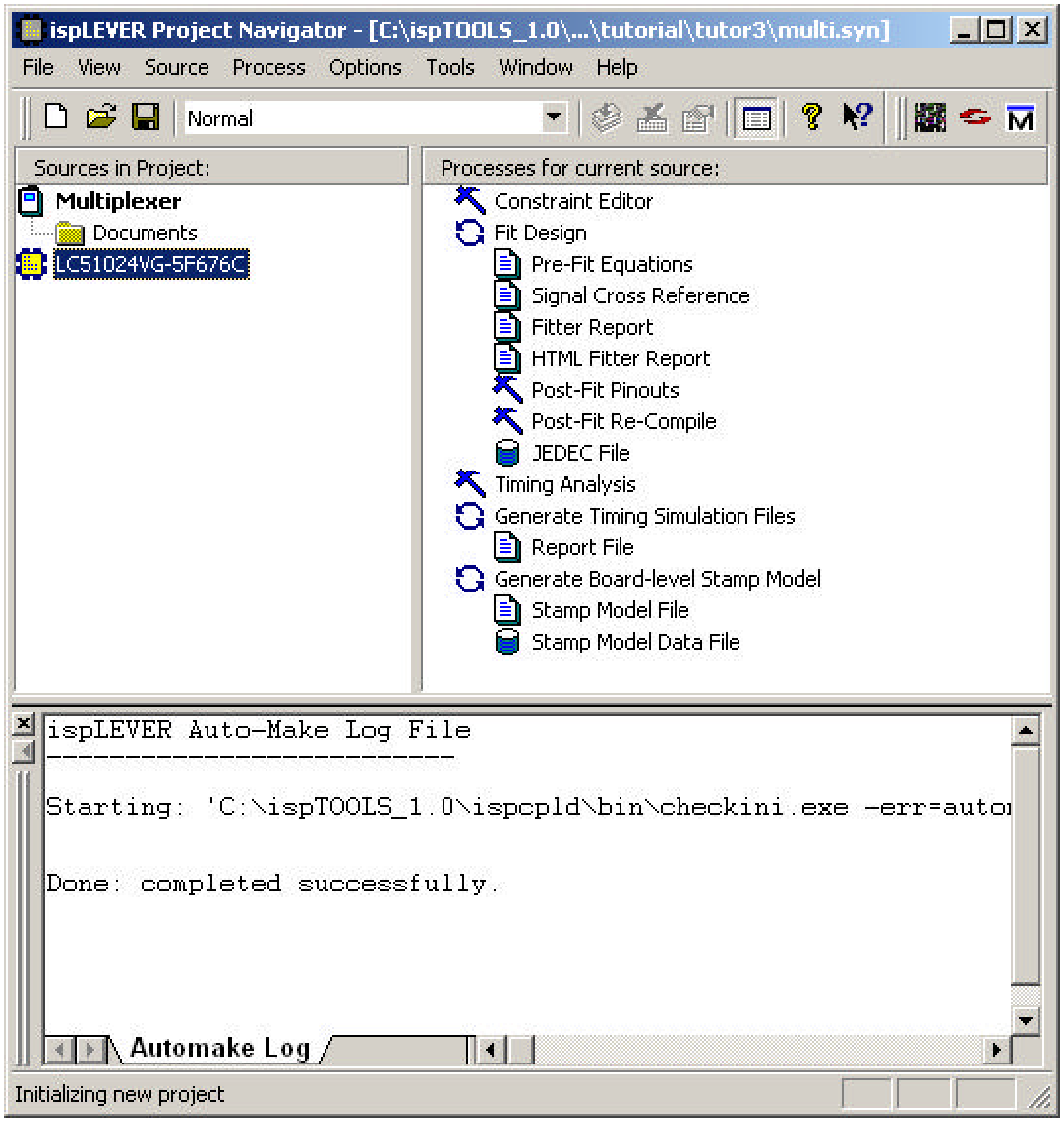
You are about to change Device kits. This will result in changes in the Project Navigator design environment. Do you wish to continue with this operation?

3. In the next dialog box, click **No**.


ispLEVER Project Navigator

Your previous constraints may not be valid with the new device. Do you want to remove all previous constraints?

4. Your Project Navigator should look like this:



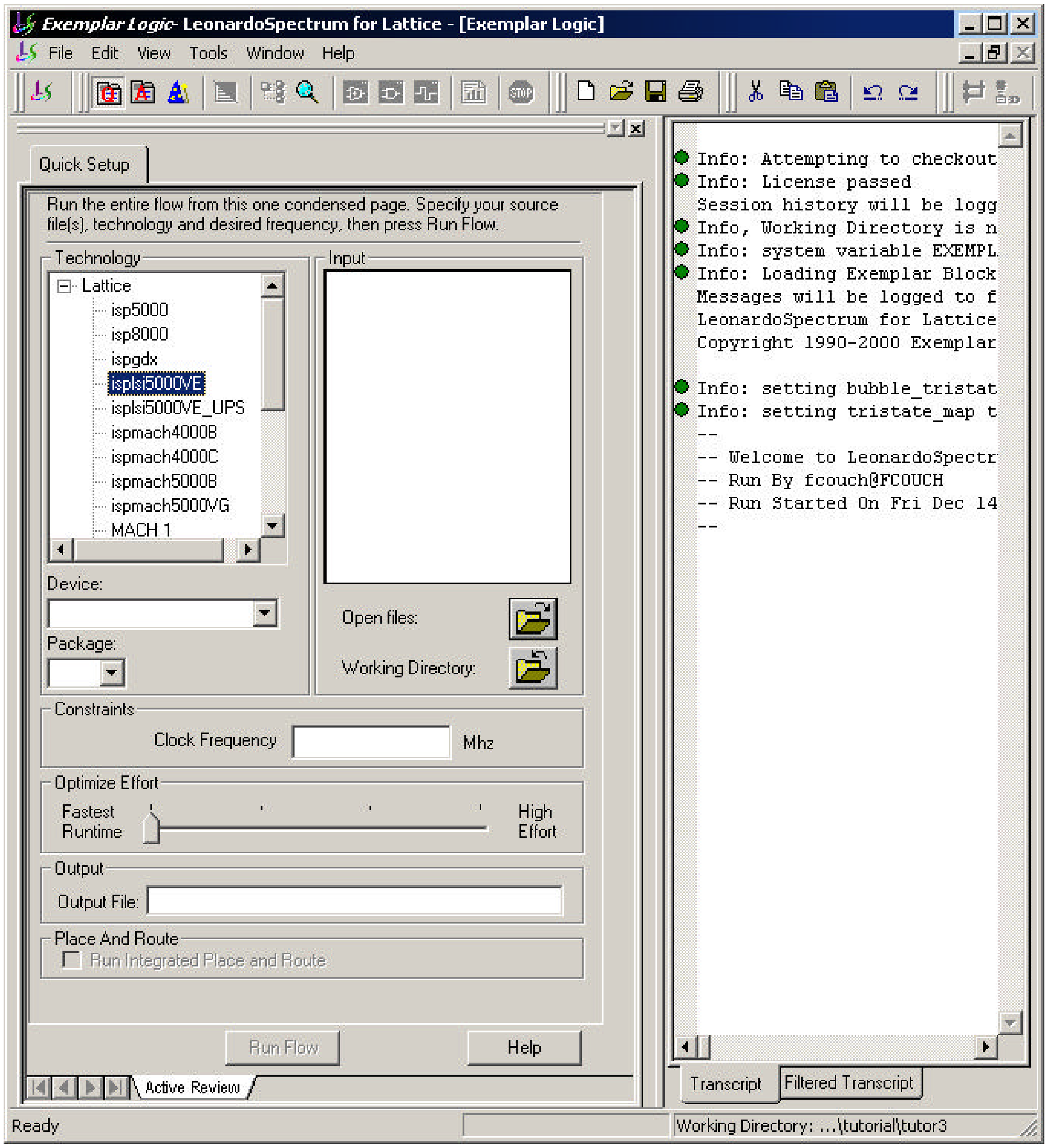
```
ispLEVER Auto-Make Log File
---------------------------

Starting: 'C:\ispTOOLS_1.0\ispcpld\bin\checkini.exe -err=auto

Done: completed successfully.
```

Task 3: Start LeonardoSpectrum from ispLEVER

When you start LeonardoSpectrum for the first time, the main window is maximized and displays the Tip of the Day and an information screen.

To start LeonardoSpectrum:

1. In the Project Navigator, choose **Tools > LeonardoSpectrum Synthesis** to open the LeonardoSpectrum synthesis tool.
2. Click **OK** to close the Tip of the Day.



Run the entire flow from this one condensed page. Specify your source file(s), technology and desired frequency, then press Run Flow.

Lattice
- isp5000
- isp8000
- ispgdx
- isplsi5000VE
- isplsi5000VE_UPS
- ispmach4000B
- ispmach4000C
- ispmach5000B
- ispmach5000VG
- MACH 1

Clock Frequency Mhz

Output File:

Run Integrated Place and Route

```
Info: Attempting to checkout
Info: License passed
Session history will be logg
Info, Working Directory is n
Info: system variable EXEMPL
Info: Loading Exemplar Block
Messages will be logged to f
LeonardoSpectrum for Lattice
Copyright 1990-2000 Exemplar

Info: setting bubble_tristat
Info: setting tristate_map t
--
-- Welcome to LeonardoSpectr
-- Run By fcouch@FCOUCH
-- Run Started On Fri Dec 14
--
```

Working Directory: ...\tutorial\tutor3

There are three ways to synthesize your design: *Quick Setup*, *Advanced Flow Tabs*, and *Synthesis Wizard*. For this tutorial you will use the Quick Setup method.

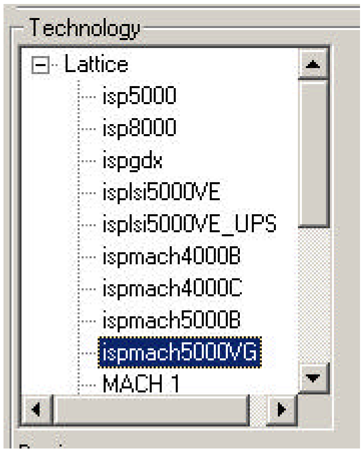
3. Make sure the **Quick Setup** tab is selected on the toolbar. Your screen should look similar to the one shown above. If not, choose **Tools > Quick Setup**.

Task 4: Use Quick Setup to Synthesize the Design

Quick Setup is a "push button" flow that you can use to achieve good first-pass synthesis results. You specify the target technology, open your input design files, optionally set the target clock frequency, and verify the name of the output netlist. When you click Run Flow, the entire synthesis flow is executed from start to finish. This includes synthesis, global constraints, optimization, and writing netlist. The output is an EDIF netlist that can be read by ispLEVER. Attributes that are placed on design objects by the HDL source code and LeonardoSpectrum are converted to properties in the EDIF netlist.

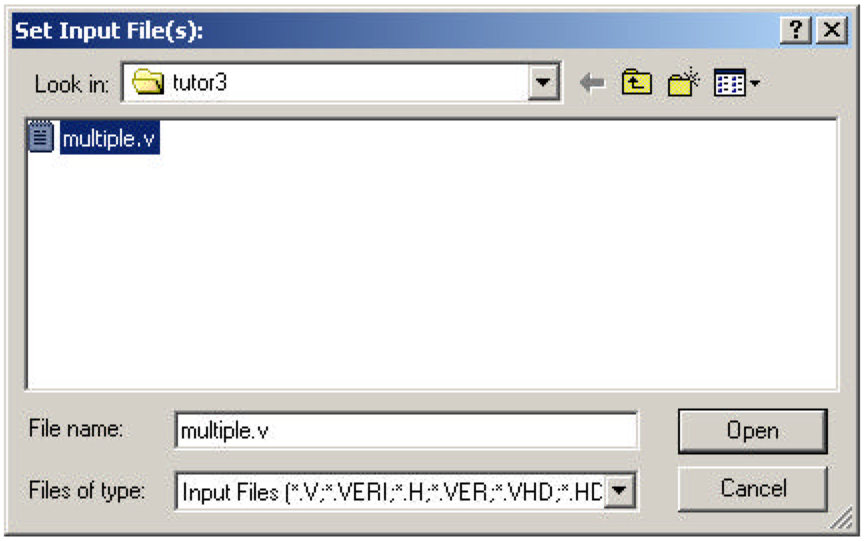
To synthesize the design:

1. On the Quick Setup tab under Technology, select the **ispmach5000VG** device family.



2. Under Input, click the **Open files** icon to open the Set Input File(s) dialog box.

   LeonardoSpectrum does not read pre-compiled HDL designs from disk. Instead, the source files are read directly into memory where LeonardoSpectrum builds an EDIF-like in-memory database.



Look in: tutor3

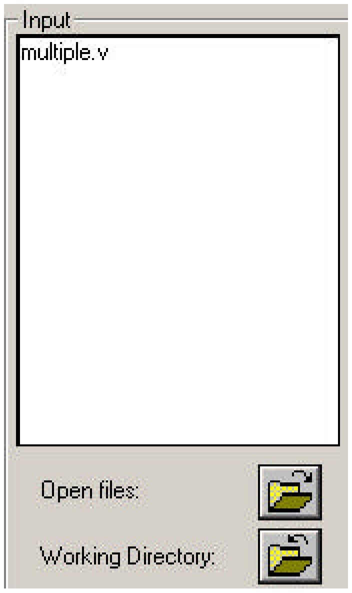
multiple.v

File name: multiple.v

Files of type: Input Files (*.V;*.VERI;*.H;*.VER;*.VHD;*.HD

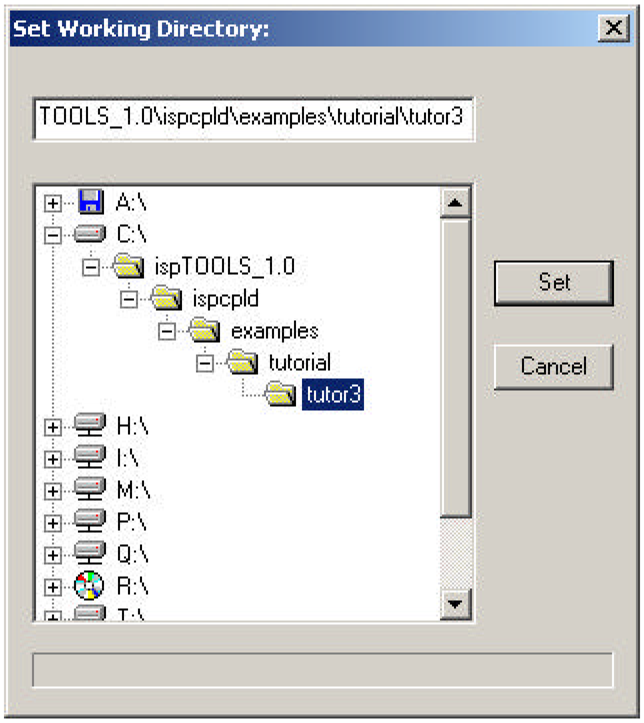
3. Make sure you're in the directory `tutorial\tutor3`. Select **`multiple.v`** and click **Open**.

   Notice that LeonardoSpectrum automatically points the output file to the project directory and places the file name in the Input box.

   multiple.v

4. Directly below the Open Files icon, click the **Working Directory** icon to open the Set Working Directory dialog box. The working directory is where LeonardoSpectrum places all generated output files. These files include the output files from the synthesis process. For ispLEVER projects, you should make the working directory the same as your project directory.
5. Make sure the path is pointing to `<install_path>\ispCPLD\examples\tutorial\tutor3`, and then click **Set** to close the dialog box.



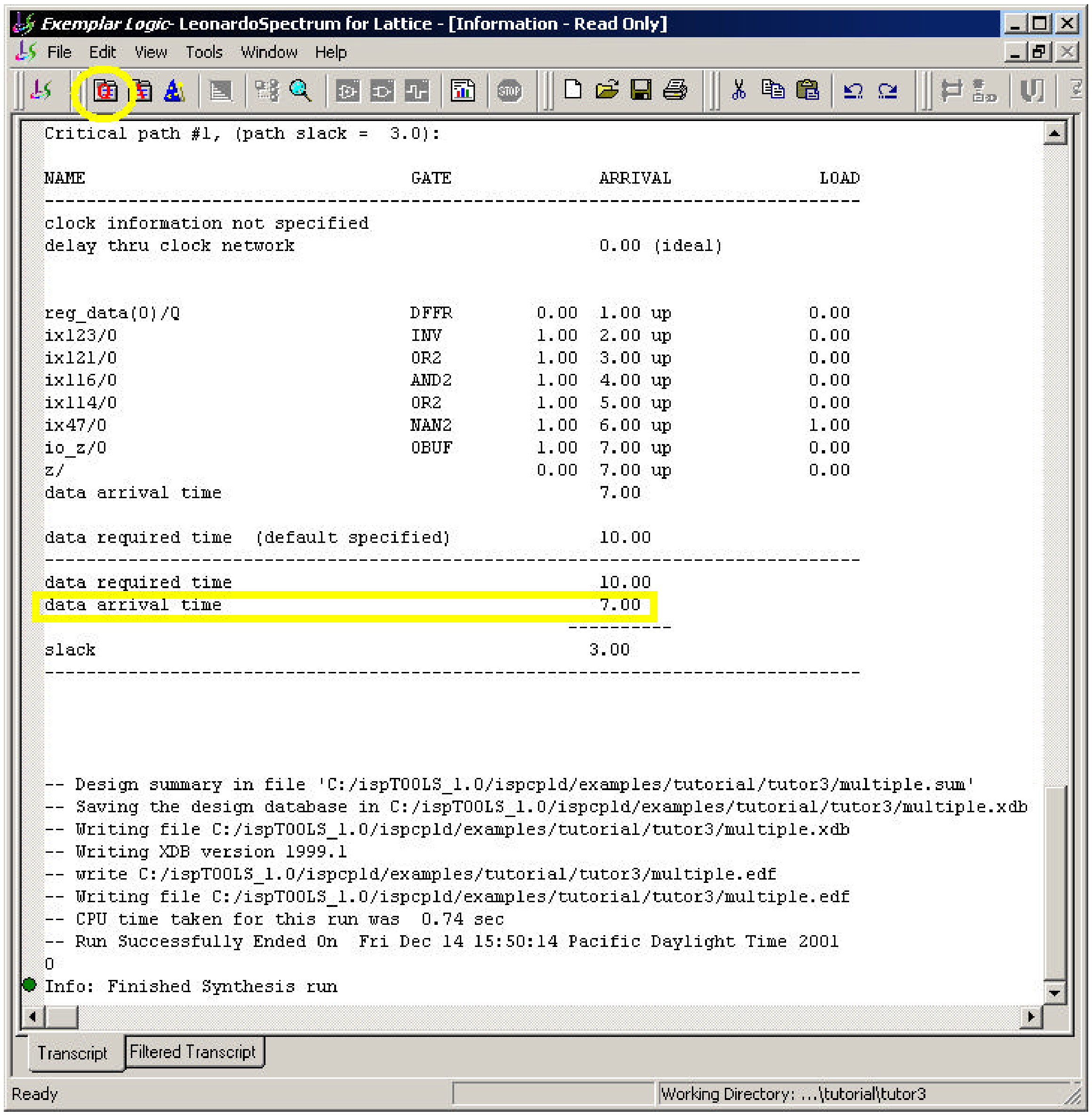
***Note**: Only a few generated files will be generated in this tutorial. Because the number of generated files in a real project can be many, it is a good practice to separate your design source files and batch scripts into a separate sub-directory. For example, the input source files could be kept in a sub-directory named* src. *Then, if your first synthesis run generates the "fastest" possible circuit, you may want to do one or more optional runs to evaluate the tradeoffs between speed and area. You can simply copy the src sub-directory into a new working directory named "smallest", for example, and the new generated files for the next run will be placed there.*

6. At the bottom of the Quick Setup tab, click **Run Flow**.

   LeonardoSpectrum reads the opened input files and creates an in-memory EDIF style database. This is called the RTL database and the design is composed of generic gates and non-mapped (black box) modules such as operators, counters, and inferred RAMs. Next, the in memory design is mapped to the specified technology, globally optimized, and the results for each module are saved. If a timing constraint is not met at this point, additional critical path optimizations are run to try to meet the constraints. The results are kept in a second in-memory technology-mapped design database. The output EDIF netlist and support files are then automatically generated and written to the working directory.

***Note**: The **Run Flow** button is not active until you have selected your Input File(s) and target technology. When the synthesis process is complete, the Information window on the right says that the run successfully ended.*

7. On the toolbar, click the **Toggle Quick Setup** button to close the tab. In the Information window, notice the *estimated* **data arrival time = 7.00**. You will reference this information in just a moment.


Exemplar Logic LeonardoSpectrum for Lattice - [Information - Read Only]
File Edit View Tools Window Help
Transcript
Filtered Transcript
Ready
Working Directory: ...\tutorial\tutor3

```
Critical path #1, (path slack =  3.0):

NAME                               GATE              ARRIVAL              LOAD
------------------------------------------------------------------------------
clock information not specified
delay thru clock network                             0.00 (ideal)


reg_data(0)/Q                      DFFR        0.00  1.00 up             0.00
ix123/O                            INV         1.00  2.00 up             0.00
ix121/O                            OR2         1.00  3.00 up             0.00
ix116/O                            AND2        1.00  4.00 up             0.00
ix114/O                            OR2         1.00  5.00 up             0.00
ix47/O                             NAN2        1.00  6.00 up             1.00
io_z/O                             OBUF        1.00  7.00 up             0.00
z/                                             0.00  7.00 up             0.00
data arrival time                                    7.00

data required time  (default specified)              10.00
------------------------------------------------------------------------------
data required time                                   10.00
data arrival time                                    7.00
                                                  ----------
slack                                               3.00
------------------------------------------------------------------------------




-- Design summary in file 'C:/ispTOOLS_1.0/ispcpld/examples/tutorial/tutor3/multiple.sum'
-- Saving the design database in C:/ispTOOLS_1.0/ispcpld/examples/tutorial/tutor3/multiple.xdb
-- Writing file C:/ispTOOLS_1.0/ispcpld/examples/tutorial/tutor3/multiple.xdb
-- Writing XDB version 1999.1
-- write C:/ispTOOLS_1.0/ispcpld/examples/tutorial/tutor3/multiple.edf
-- Writing file C:/ispTOOLS_1.0/ispcpld/examples/tutorial/tutor3/multiple.edf
-- CPU time taken for this run was  0.74 sec
-- Run Successfully Ended On  Fri Dec 14 15:50:14 Pacific Daylight Time 2001
0
Info: Finished Synthesis run
```

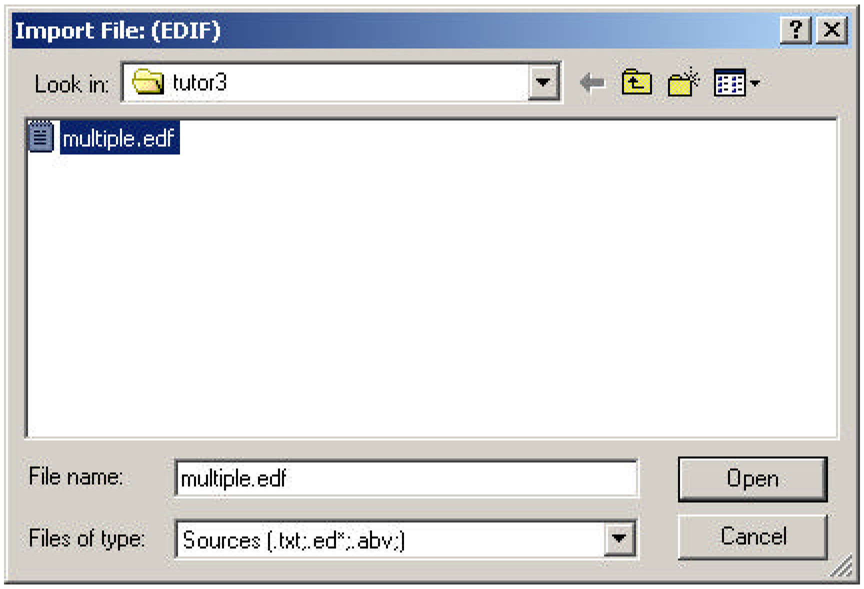
8. Minimize Leonardo Spectrum. You may want to refer to it later.

Task 5: Import the EDIF File into Your Project

You can import EDIF 2.0.0 netlists from third-party synthesis tools, such as Exemplar LeonardoSpectrum, into ispLEVER.

To import an EDIF netlist into your project:

1. In the ispLEVER Project Navigator, choose **Source** > **Import** to open the Import File dialog box.



Import File: (EDIF)

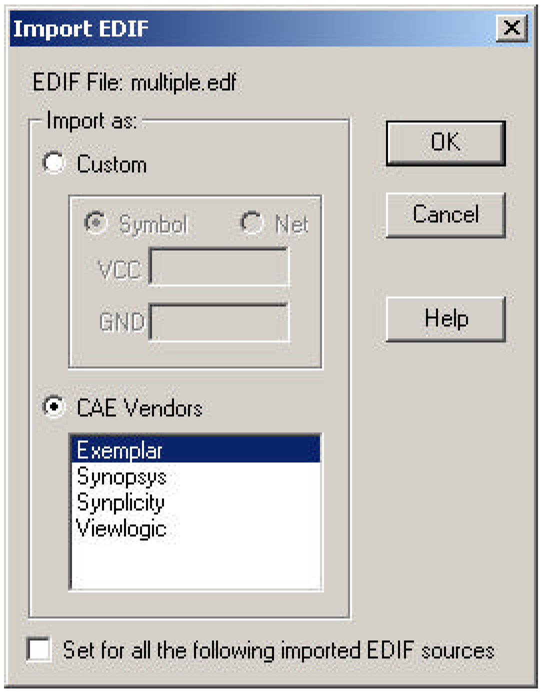
Look in: tutor3

multiple.edf

File name: multiple.edf

Files of type: Sources (.txt;.ed*;.abv;)

2. Select **`multiple.edf`**, and then click **Open** to open the Import EDIF dialog box.
3. Under CAE Vendors, select **Exemplar** and click **OK**.



Import EDIF

EDIF File: multiple.edf

Import as: ( ) Custom [✓ CAE Vendors]

( ) Symbol ( ) Net

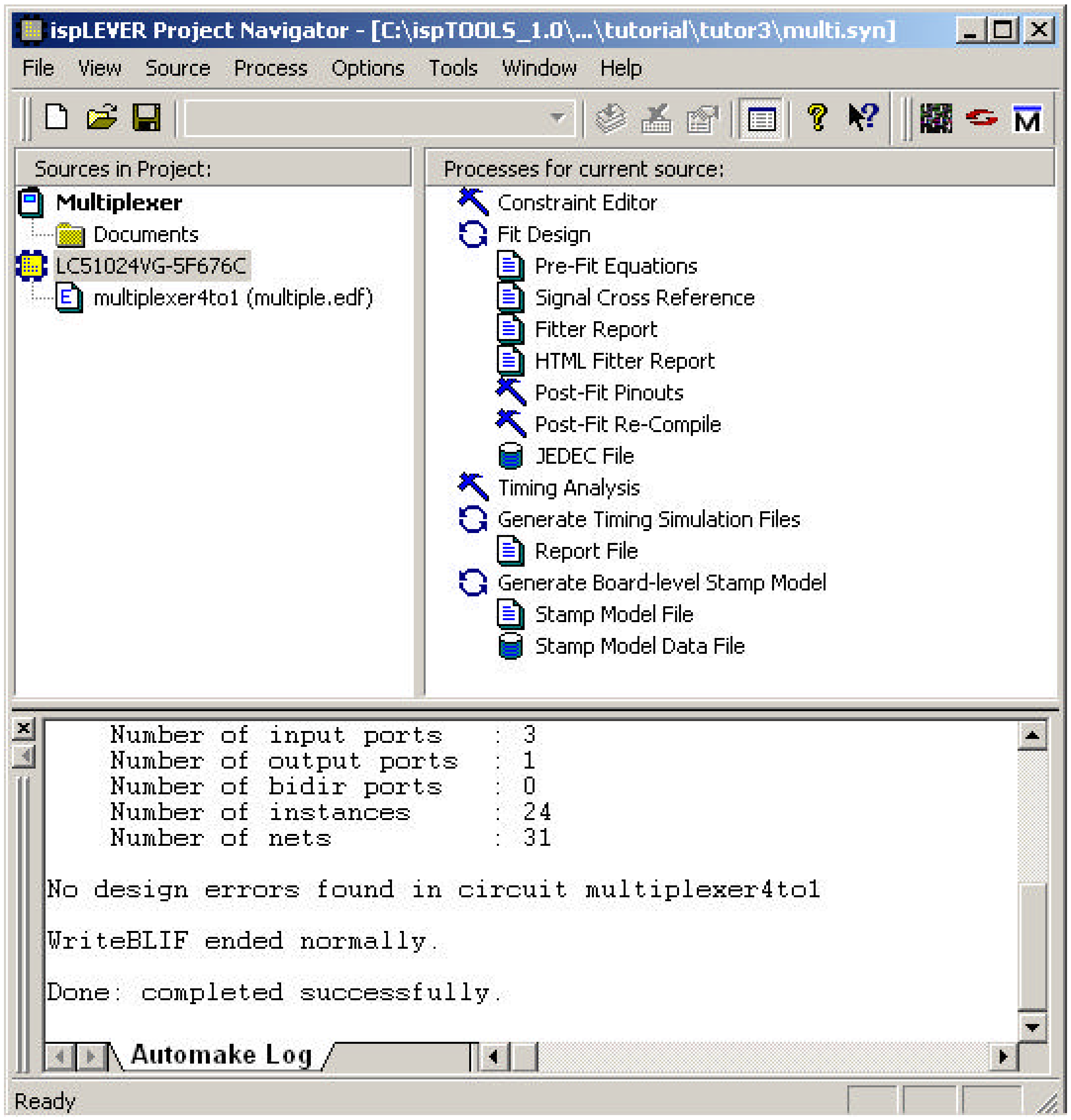
VCC

GND

CAE Vendors: [✓ Exemplar] Synopsys Synplicity Viewlogic

[ ] Set for all the following imported EDIF sources

The software adds the selected EDIF file (`multiple.edf`) to the project sources.



```
    Number of input ports    : 3
    Number of output ports   : 1
    Number of bidir ports    : 0
    Number of instances      : 24
    Number of nets           : 31

No design errors found in circuit multiplexer4to1

WriteBLIF ended normally.

Done: completed successfully.
```

Note: *After you import an EDIF file into the ispLEVER project, it is always linked to the Project Navigator. Therefore, if you make changes and recompile your HDL file to create a new EDIF file, your project is automatically updated as well.*

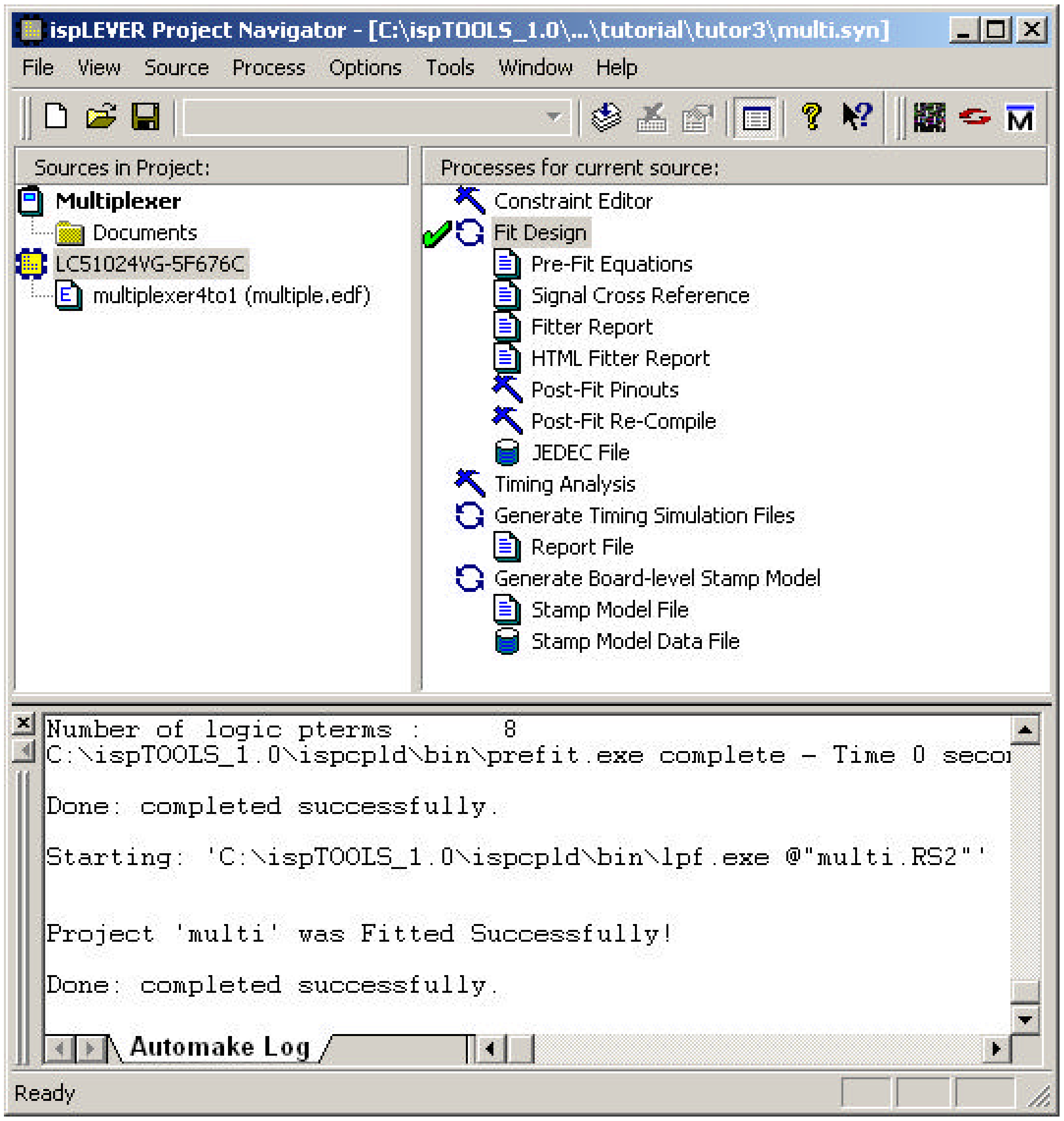
Task 6: Fit the Design and View the Report

The ispLEVER software has a single user interface with all options preset to deliver the highest possible push-button performance for most devices. At the end of a successful fitter run, the ispLEVER software generates a JEDEC file, as well as a fitter report so that you can see how the ispLEVER software has utilized and routed the part.

To run the Fitter and view the report:

1. With the target device selected in the Sources window, double-click **Fit Design** in the Processes window to run the Fitter. The ispLEVER software successfully fits the design in the specified device and generates a JEDEC file.

***Optional**: If you like, you can right-click on the **JEDEC File** process and select **View** to create and look at the contents of the JEDEC file. Close the file when you are through.*



```
Number of logic pterms :      8
C:\ispTOOLS_1.0\ispcpld\bin\prefit.exe complete - Time 0 secon

Done: completed successfully.

Starting: 'C:\ispTOOLS_1.0\ispcpld\bin\lpf.exe @"multi.RS2"'


Project 'multi' was Fitted Successfully!

Done: completed successfully.
```

2. Double-click on the **HTML Fitter Report** process to open the report in your browser. View the contents and then close the report.

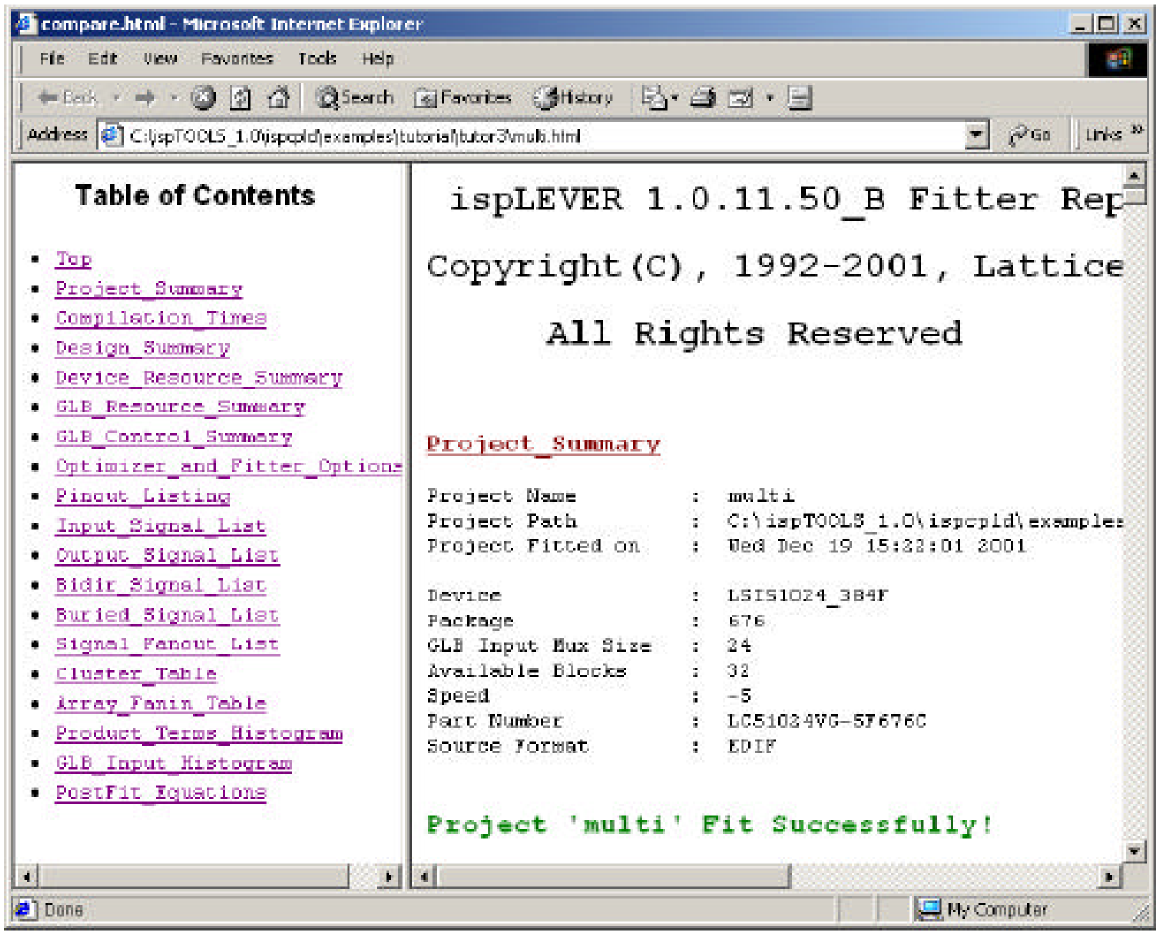

compare.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\ispTOOLS_1.0\ispcpld\examples\tutorial\tutor3\multi.html
Go Links
Done
My Computer

Table of Contents

- Top
- Project_Summary
- Compilation_Times
- Design_Summary
- Device_Resource_Summary
- GLB_Resource_Summary
- GLB_Control_Summary
- Optimizer_and_Fitter_Options
- Pinout_Listing
- Input_Signal_List
- Output_Signal_List
- Bidir_Signal_List
- Buried_Signal_List
- Signal_Fanout_List
- Cluster_Table
- Array_Fanin_Table
- Product_Terms_Histogram
- GLB_Input_Histogram
- PostFit_Equations

```
ispLEVER 1.0.11.50_B Fitter Rep
Copyright(C), 1992-2001, Lattice
        All Rights Reserved

Project_Summary

Project Name           :  multi
Project Path           :  C:\ispTOOLS_1.0\ispcpld\examples
Project Fitted on      :  Wed Dec 19 15:33:01 2001

Device                 :  LSIS1024_384F
Package                :  676
GLB Input Mux Size     :  24
Available Blocks       :  32
Speed                  :  -5
Part Number            :  LC51024VG-5F676C
Source Format          :  EDIF

Project 'multi' Fit Successfully!
```

Task 7: Run Static Timing Analysis

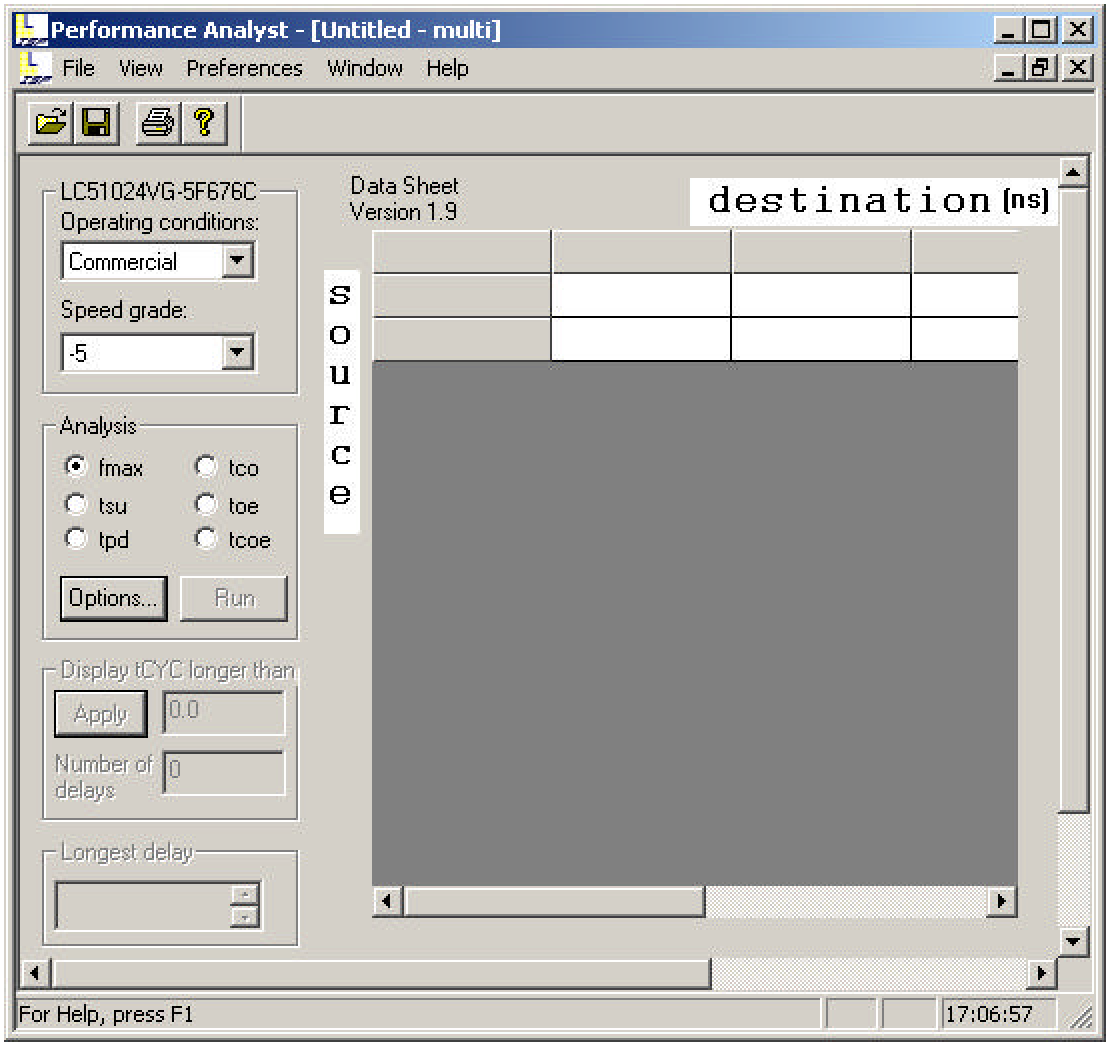
Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data such as the critical path, setup/hold time requirements, and the maximum frequency.

The Performance Analyst traces each logical path in the design and calculates the path delays using the device's timing model and worst-case AC specs supplied in the device data sheet.

The timing analysis results are displayed in a graphical spreadsheet with source signals displayed on the vertical axis and destination signals displayed on the horizontal axis. The worst-case delay value is displayed in a spreadsheet cell if there is at least one delay path between the source and destination. To more easily identify performance bottlenecks, you can double-click a cell to view the path delay details.

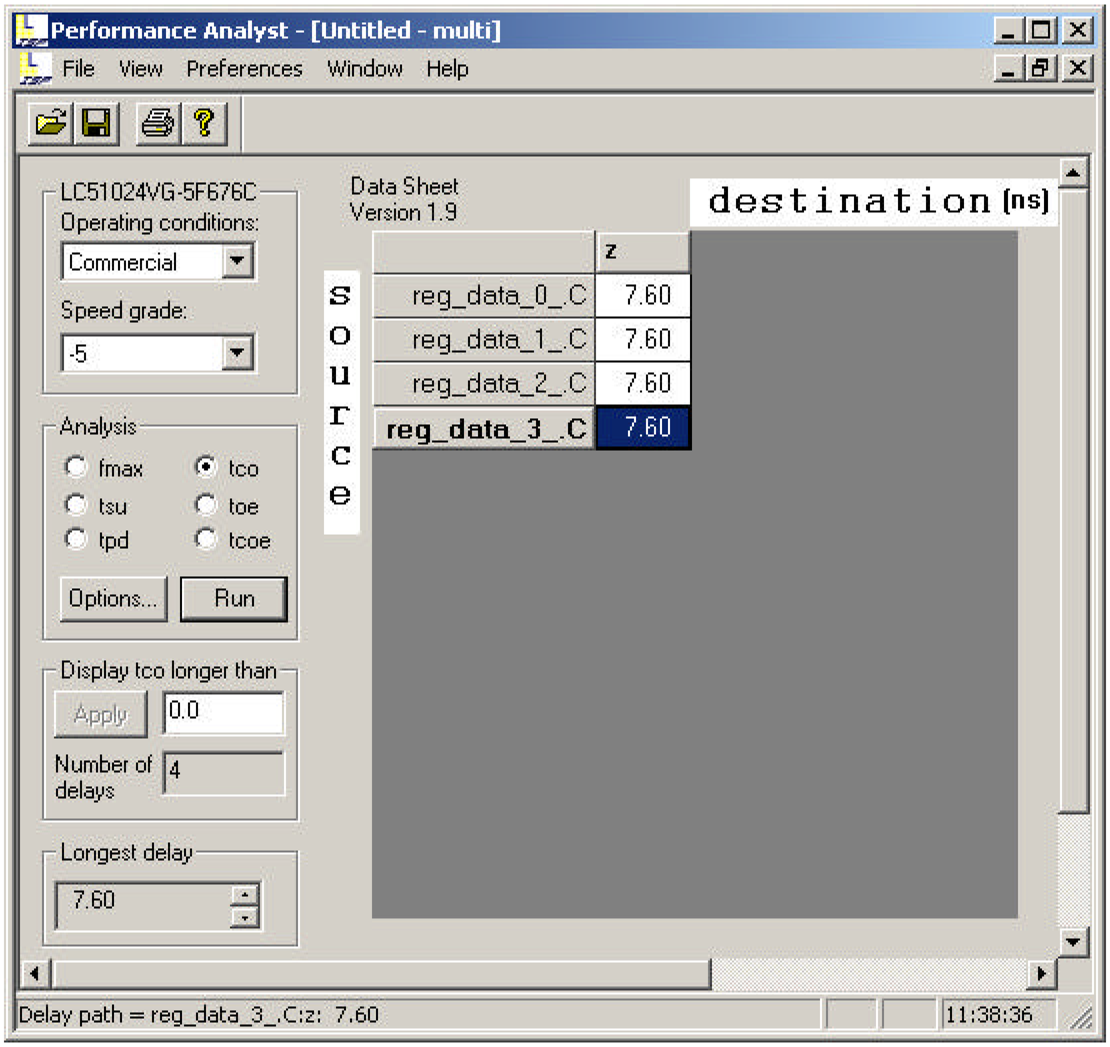
To run timing analysis:

1. In the Project Navigator Sources window, select the target device. In the Processes window, double-click the **Timing Analysis** process to open the Performance Analyst.



2. The Performance Analyst performs six distinct analysis types: fMAX, tSU, tPD, tCO, tOE, and tCOE. The first type, fMAX, is an internal register-to-register delay analysis. fMAX measures the maximum clock operating frequency, limited by worst-case register-to-register delay. The remaining five types are external pin-to-pin delay analysis. Timing threshold filters, source and destination filters, and path filters can be used to independently fine-tune each analysis.

   Under Analysis, select **tco** and then click **Run**. The tCO path trace analysis reports clock-to-out delay starting from the primary input, going through the clock of flip-flops or gate of latches, and ending at the primary output. In this case it is 7.60ns.


Performance Analyst - [Untitled - multi]
File View Preferences Window Help
Options... Run
Apply

LC51024VG-5F676C
Operating conditions: Commercial
Speed grade: -5

Analysis: fmax, [✓ tco], tsu, toe, tpd, tcoe

Display tco longer than: 0.0
Number of delays: 4
Longest delay: 7.60

Data Sheet
Version 1.9

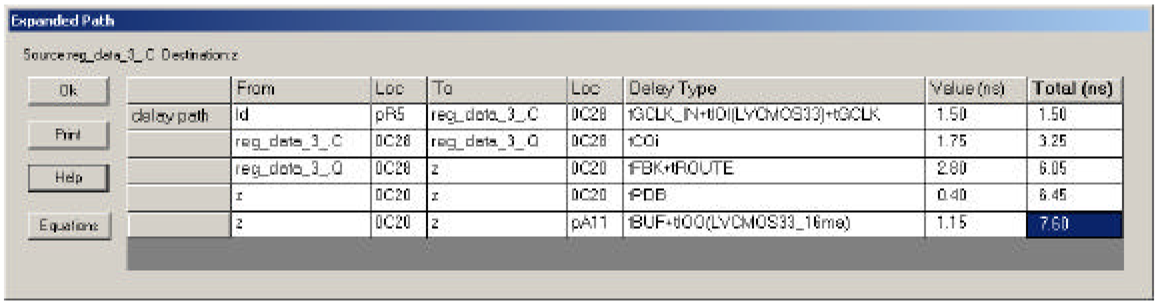
destination (ns)

source	z
reg_data_0_.C	7.60
reg_data_1_.C	7.60
reg_data_2_.C	7.60
reg_data_3_.C	7.60

Delay path = reg_data_3_.C:z: 7.60   11:38:36

Remember from Task 4 that the Data Arrival Time was 7.0. This number was the simulated estimate. Now, using the Performance Analyst, you can see the actual delay of 7.60ns.

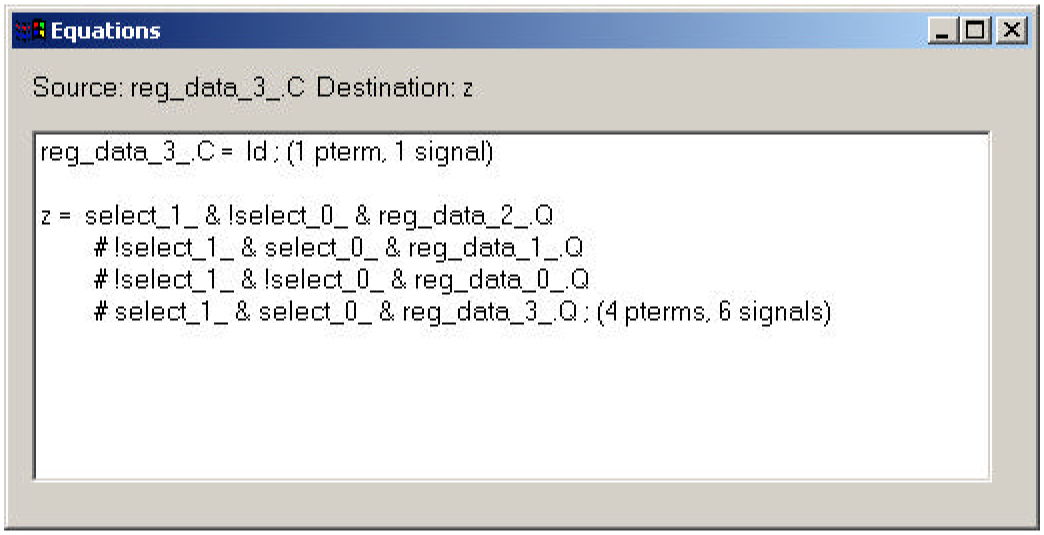
3. Double-click the highlighted cell (**7.60**) in the spreadsheet window to open the Expanded Path dialog box. This dialog lets you analyze individual timing components used to calculate the timing path. There is a source pin (From) and a destination pin (To). Also shown is the delay type, the delay of that path (Value ns), and the cumulative delay of all the signals.



Source:reg_data_3_C Destination:z

	From	Loc	To	Loc	Delay Type	Value (ns)	Total (ns)
delay path	Id	pR5	reg_data_3_C	0C28	tGCLK_IN+tIOI(LVCMOS33)+tGCLK	1.50	1.50
	reg_data_3_C	0C28	reg_data_3_Q	0C28	tCOi	1.75	3.25
	reg_data_3_Q	0C28	z	0C20	tFBK+tROUTE	2.80	6.05
	z	0C20	z	0C20	tPDB	0.40	6.45
	z	0C20	z	pA11	tBUF+tIOO(LVCMOS33_16ma)	1.15	7.60

4. Click **Equations** to open the Equations dialog box, which shows the functional relationship between the selected source/destination.



Source: reg_data_3_C Destination: z

```
reg_data_3_.C = Id ; (1 pterm, 1 signal)

z = select_1_ & !select_0_ & reg_data_2_.Q
      # !select_1_ & select_0_ & reg_data_1_.Q
      # !select_1_ & !select_0_ & reg_data_0_.Q
      # select_1_ & select_0_ & reg_data_3_.Q ; (4 pterms, 6 signals)
```

5. Close the Performance Analyst without saving.
6. Close LeonardoSpectrum, and then close ispLEVER without saving.

Congratulations

You have completed the HDL Synthesis Design with LeonardoSpectrum tutorial. In this tutorial you have learned how to:

- Create a new EDIF project in the ispLEVER system and target a device.
- Launch LeonardoSpectrum from within ispLEVER and generate an EDIF netlist file using the Quick Setup tab flow.
- Import the EDIF file into the ispLEVER system, fit the design, generate a JEDEC file, and view the Fitter report.
- Run static timing analysis using the Performance Analyst and view the results.