DesignDirect-CPLD Tutorial Manual

Mail: Vantis Corporation
P.O. Box 3755
995 Stewart Drive
Sunnyvale, CA 94088
U.S.A.

Phone: (408) 616-8000
(888) 826-8472

Web: www.vantis.com

Table of Contents

DesignDirect-CPLD Tutorial Manual

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

Before starting these tutorials you should take the **DesignDirect Quick Start Tutorial** in the "Introduction" chapter of the *DesignDirect-CPLD User Guide*. After that, we suggest that you perform the following tutorials in the order in which they appear.

Tutorial 1

DesignDirect Basics

This tutorial provides an overview of the features and operation of DesignDirect, focusing on the Project Manager to complete a programmable device design.

Follow the steps in this tutorial to implement a design called **ddbasics** using a mixture of schematics and ABEL-HDL.

Prerequisites

Before attempting this tutorial, you should complete the **DesignDirect Quick Start Tutorial** in the *DesignDirect-CPLD User Guide*, "Introduction" chapter.

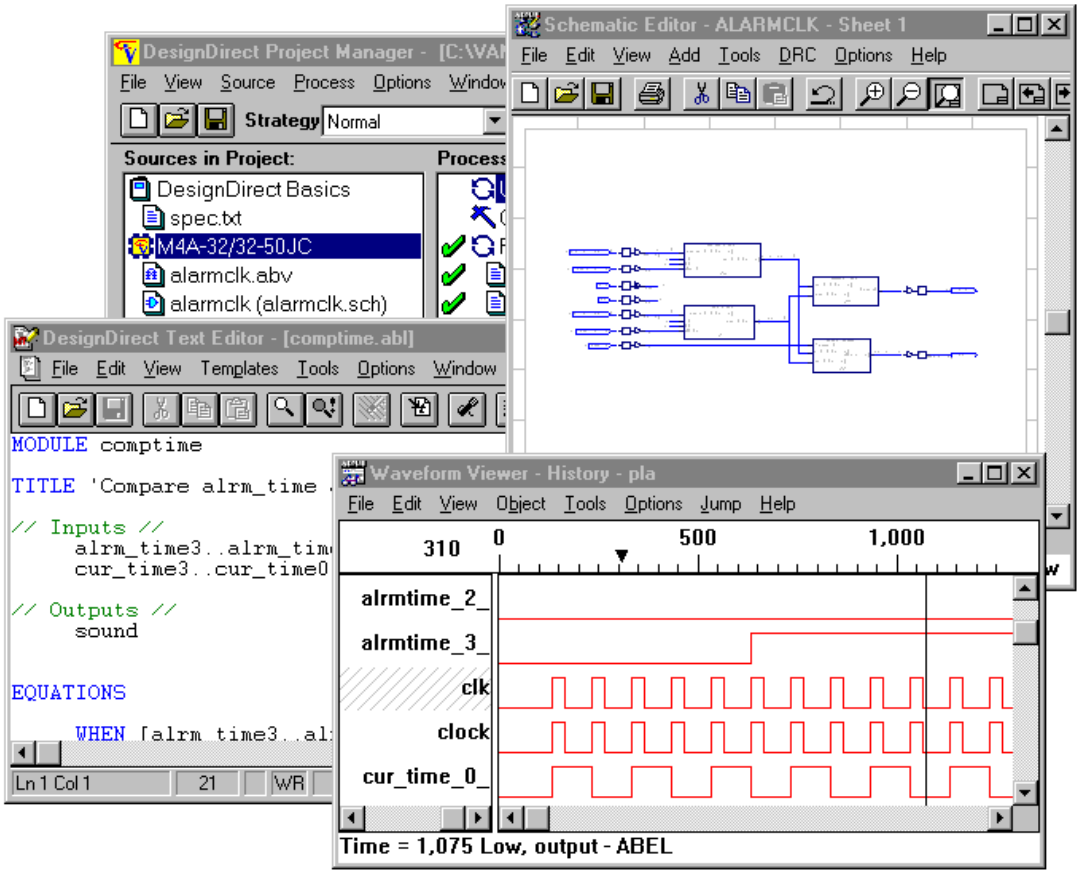
Learning Objectives

When you are finished with this tutorial, you should have a basic understanding of the steps and tools necessary to complete a Vantis programmable IC design using DesignDirect.

Time to Complete This Tutorial

On average, the time to complete this tutorial is the following:

About 30 minutes to complete *DesignDirect Basics.*



Task 1: *Open a Project*

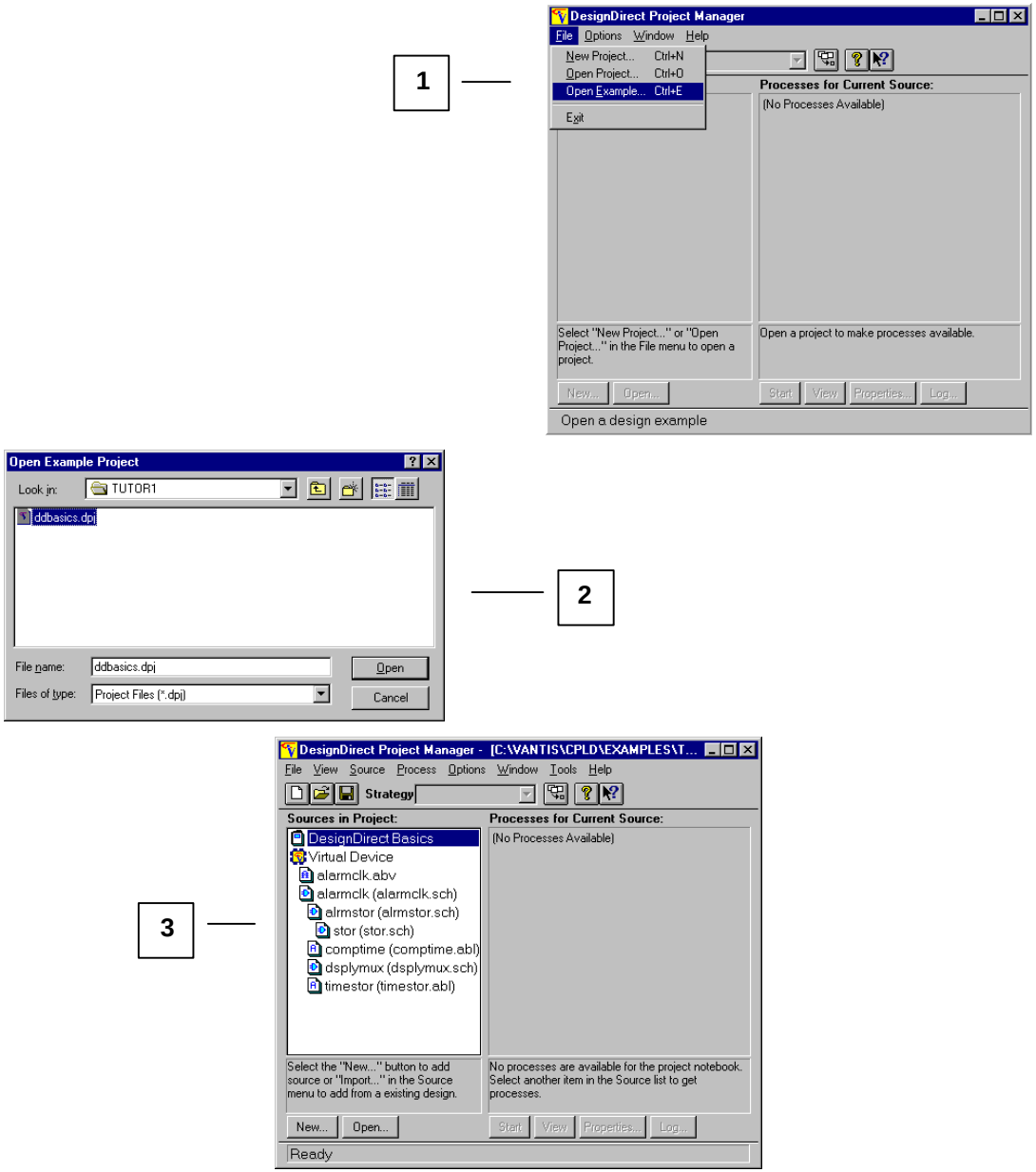
After starting the Project Manager, you are ready to open the tutorial project file.

To open a project:

1. Choose **File** > **Open Example** to open the Open Project dialog.
2. In the dialog, change directories to **TUTORIAL\TUTOR1.**
3. Select the **ddbasics.dpj** file and then click **Open**.

   The DesignDirect Basics project opens in the Project Manager. A project contains all the source files for the design, such as behavioral and schematic logic descriptions, test fixtures, test benches, test vectors, waveforms, device type, and design documentation.

Note: *To learn about creating a new project, see the Introduction, "Getting Started with DesignDirect " in the* DesignDirect-CPLD User Guide.



Task 2: *About the Project Manager Interface*

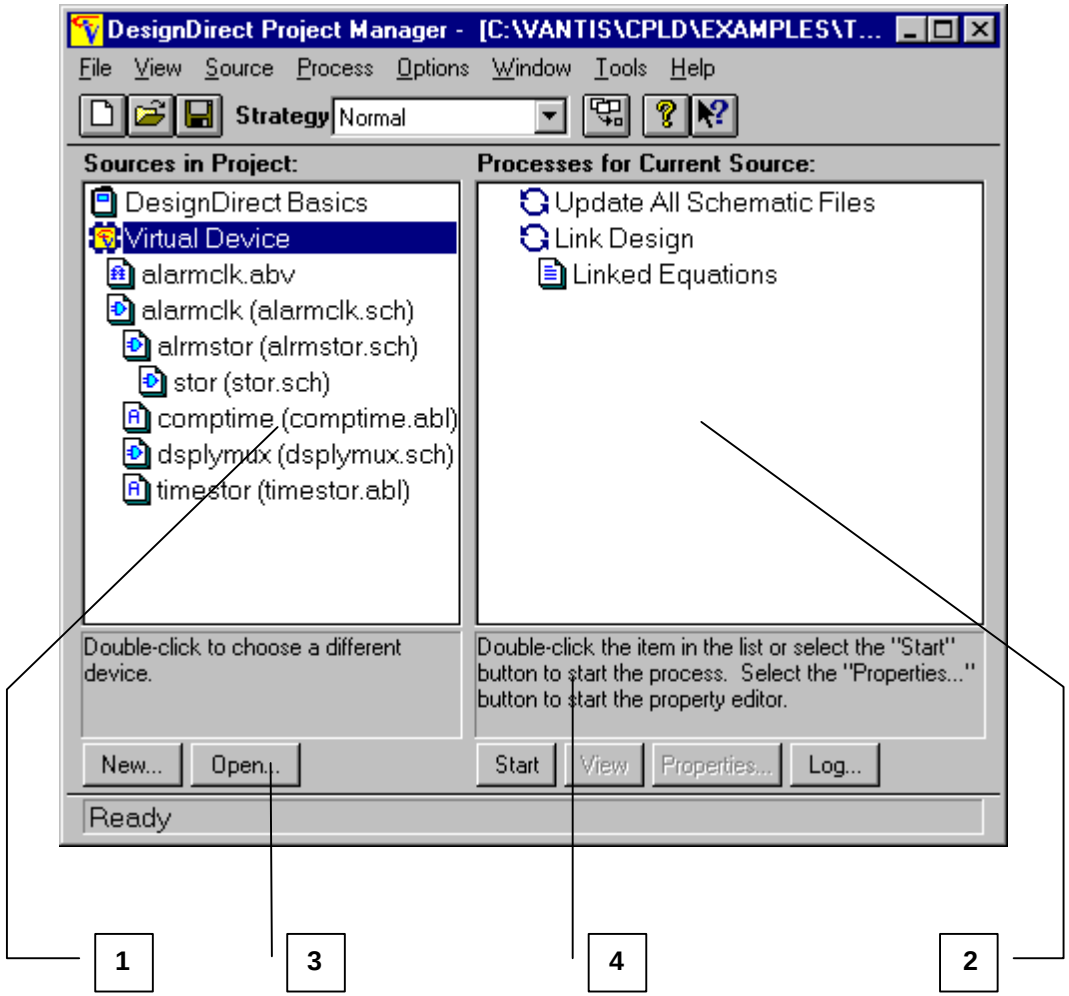
The Project Manager has two primary interface elements: the *Sources* window and the *Processes* window.

The Sources window (**1**) shows all the design files associated with a project, listed in their logical, hierarchical order. Each source in the list is identified with an icon.

The Processes window (**2**) shows all the processing tasks that apply to a selected source in the Sources window. Typical programmable device processing task includes netlisting, compiling, logic reduction, logic synthesis, fitting, and simulation model building.

Buttons at the bottom of the Project Manager (**3**) provide quick command access to key functions.

Also, there are two message areas (**4**) located above and below these buttons.



Task 3: *Examine the Process Flows*

The Project Manager is context sensitive—that is, it automatically adjusts the processes for you depending on what you want to do.

First, processing for a given file changes depending on the target device. This is called the *project-level* flow. For instance, a design targeted for a CPLD device is processed differently than a design targeted for a PAL device.

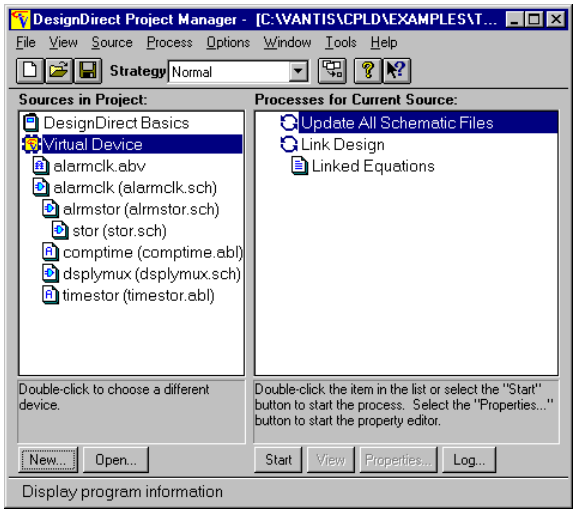
Second, the process flow changes depending on what kind of source file is highlighted in the Sources window. This is called the *source-level* flow. Say, for example, you select a schematic source (.sch), several processes appear in the Processes window. But if you select an ABEL test vector file (.abv), only the processes necessary for that source (functional simulation) are shown.

Examine the process flows for the current project:

1 To see the process-level flow, select the Device icon in the Sources window. The processes for the project appear in the Processes window.

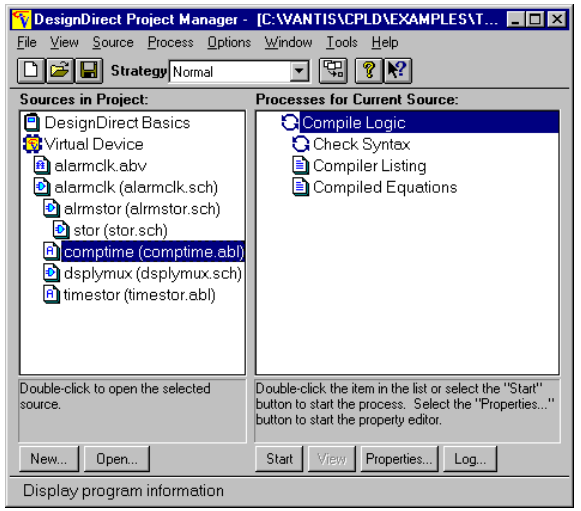
2 To see a source-level flow, select a source. The processes for the selected source appear in the Processes window.

Continue to select different sources and notice the processes change in the Processes window.



1

2



Task 4: *Create and Import a Design Documentation File*

The Project Notebook works much like an engineering notebook. You can give it any name to describe your project. In addition, you can include files created by Windows applications other than DesignDirect. For instance, you might want to include a project schedule, a specification document, or a timing diagram.

To create a text document and import it into the Project Notebook:

1 Choose **Source** > **New**.

2 In the New Source dialog, select **User Document**, and then click **OK**.

3 In the New File dialog, type **spec** in the File Name field. Then click **OK**.

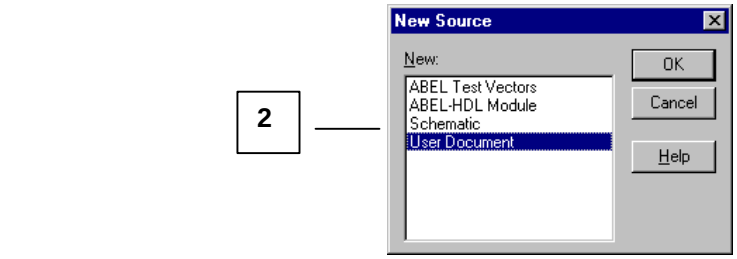
4 In the Text Editor, type **This is a sample text document**.

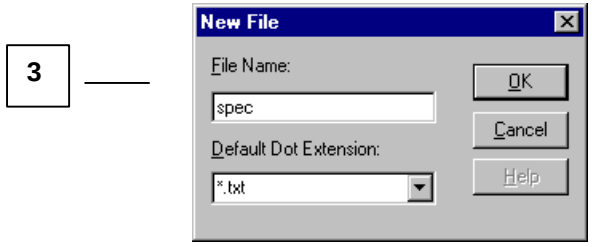
5 Choose **File** > **Save**.

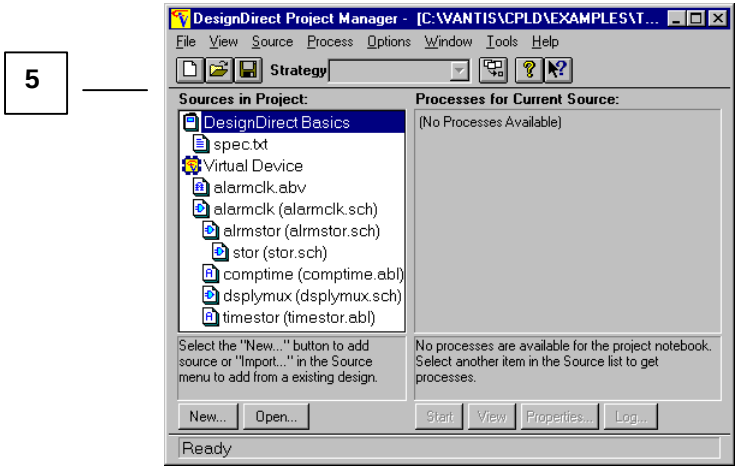
DesignDirect automatically adds the file to the Sources window, under the Project Notebook, as part of the **DesignDirect Basics** project.

> ***Note:*** *You can add logic source files (for example, schematics and ABEL-HDL files) in the same way, although they appear in a different place in the Sources window. The Project Notebook area is for non-design source files.*

6 Choose **File** > **Exit**.







Task 5: *Edit a Project Source*

You can edit project source files by double clicking the desired source in the Sources window. DesignDirect opens the appropriate editor with that source ready for editing.

In this step you will see how to open the Schematic Editor from a schematic source.

To edit a schematic source file:

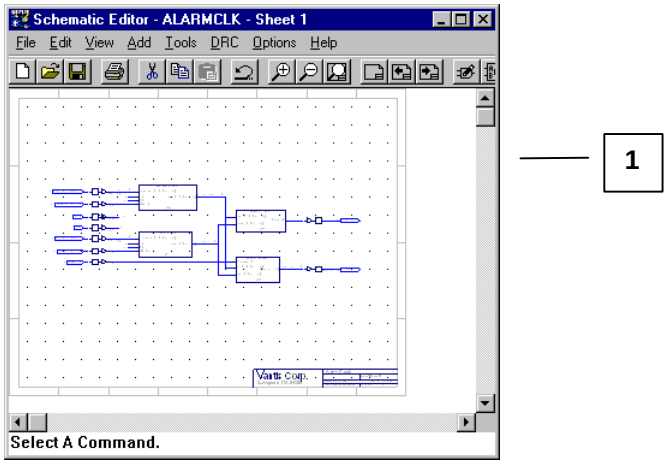
1 Double-click the top-level schematic **alarmclk** (alarmclk.sch) in the Sources window. The Schematic Editor opens on the schematic.

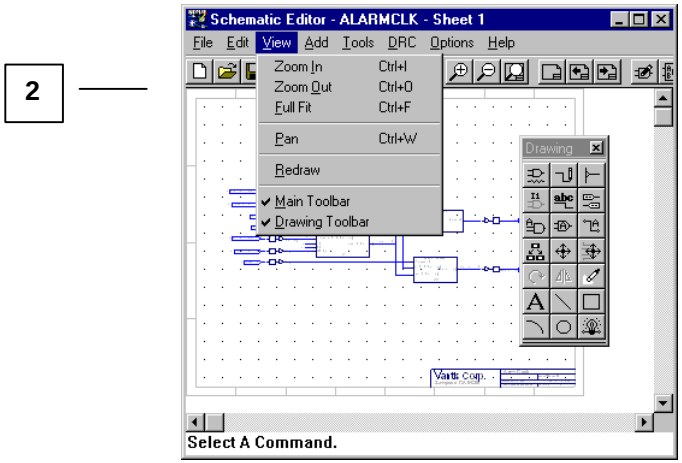
When you double-click any source, DesignDirect automatically opens the application associated with that source file.

2 On the **View** menu, make sure that there is a check mark beside the **Main Toolbar** and **Drawing Toolbar** commands. (Click the command to toggle the check mark.)

3 Choose **File** > **Exit** to exit the Schematic Editor.

Note: *You can also open a DesignDirect application from the Project Manager Window menu.*





Task 6: *View the Design Hierarchy*

You can view the hierarchy of a top-level schematic and lower-level sources using the Hierarchy Navigator. With the Hierarchy Navigator you can traverse a design and access a number of powerful database features, such as pushing into blocks to see the source they refer to, accessing connectivity information, and querying nets.

Note: *The Hierarchy Navigator is for viewing a design, not editing. For editing source files, use the Schematic Editor or Text Editor.*

To view the design hierarchy using the Hierarchy Navigator:

1 Select the top-level schematic **alarmclk** in the Sources window and double-click **Navigate Hierarchy** in the Processes window. The schematic opens in the Hierarchy Navigator.

2 Click **Push/Pop** ( ) on the toolbar. The cursor changes to cross hairs.

The Push/Pop command lets you "Push" down into a lower, more-detailed, level in the hierarchy or "Pop" up to a higher, less-detailed, level in the hierarchy.

3 Click *inside* the **ALRMSTOR** block symbol (upper-left) to push down into the block. The Hierarchy Navigator displays the contents for this block, which is another schematic. (You may need to use the Zoom command.)

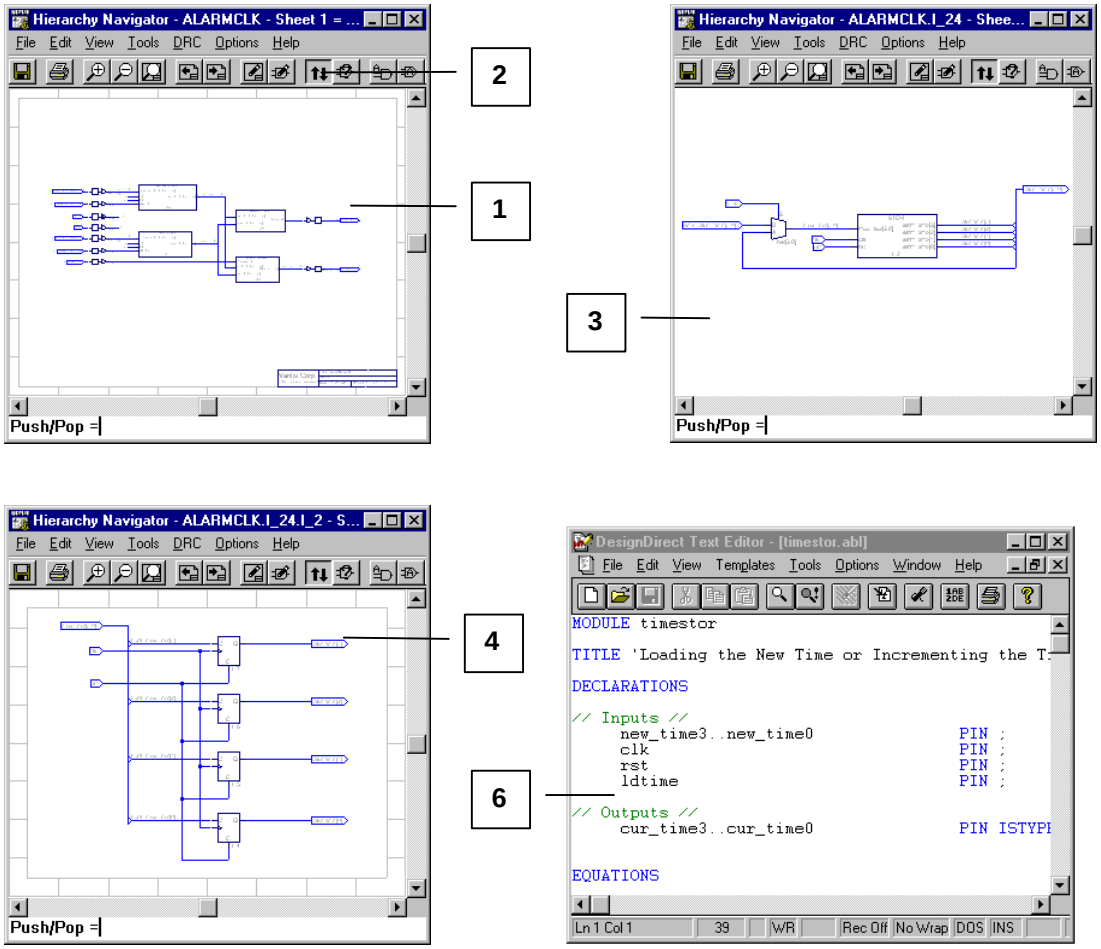
4 Now click *inside* the **STOR** block symbol to see another level of the hierarchy. Try to click on another symbol. Nothing happens; there is no more hierarchy in this schematic.

5 Next "pop" back up to the top level of the design by clicking *outside* any symbol, in the white space. The Hierarchy Navigator "pops" you up to the second level. Repeat this step again to return to the top level schematic.

6 Now, push into an ABEL-HDL block. Click *inside* the **TIMESTOR** block symbol (lower-left). DesignDirect opens the Text Editor and displays the ABEL-HDL source for this block.

7 Choose **File** > **Exit** to close the Text Editor.

8 Choose **File** > **Exit** to close the Hierarchy Navigator.



Task 7: *Process a Module*

DesignDirect allows you to think *less* about the tools and steps involved in getting to a result and *more* about the desired result. For example, if you want to view a report, you don't want to have to figure out which processes should be run to generate the report. DesignDirect determines what steps are involved and runs them automatically if required. Because you *intuitively* know what results you want, DesignDirect's results-oriented processing greatly simplifies learning new architectures because it minimizes the impact of having to learn new tools and procedures.

To process the counter module:

1 In the Project Manager Sources window, select the schematic **dsplymux**. Notice that there is one process for this schematic (Compile Schematic) that generates one report file (Compiled Equations).

While you can run each process separately, it's faster and easier to tell DesignDirect what end result you want, and let DesignDirect determine which steps should be run in what order.

2 In the Processes window, double-click **Compiled Equations**. DesignDirect compiles the schematic into ABEL-HDL equations and displays the results in the Report Viewer.

The Processes window displays the status of each process as it runs.

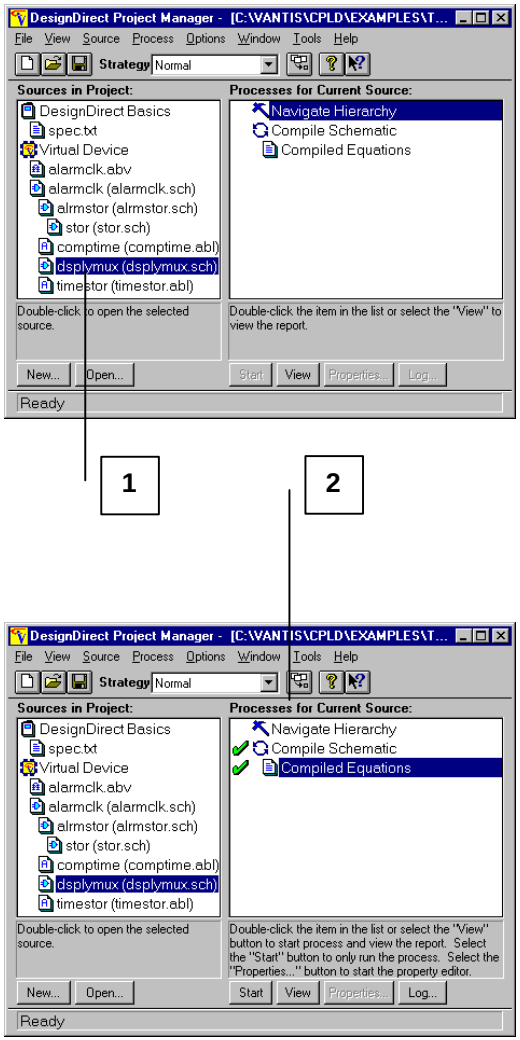
3 Close the Report Viewer.

4 Look at the Processes window. Note the processing-status icon that has appeared to the left of the processes, including the prerequisite processes that DesignDirect ran automatically.

DesignDirect tells you the current status of any process using icons next to the process name. Status icons and their meanings are listed below.

Icon	Meaning
✔	The process completed successfully and is up-to-date.
!	The process completed successfully, but warnings or messages were generated; these can be read in the Report Viewer.
✖	The process did not complete, because errors occurred. Errors must be corrected to go forward.

The Processes window is constantly updated. If you edit a source file, the green check marks disappear for processes that were previously run.



Task 8: *Run Equation Simulation*

Equation simulation uses design test vectors to simulate the design logic or equations, independent of any device. The more comprehensive and detailed your test vectors are, the more useful your simulation results will be.

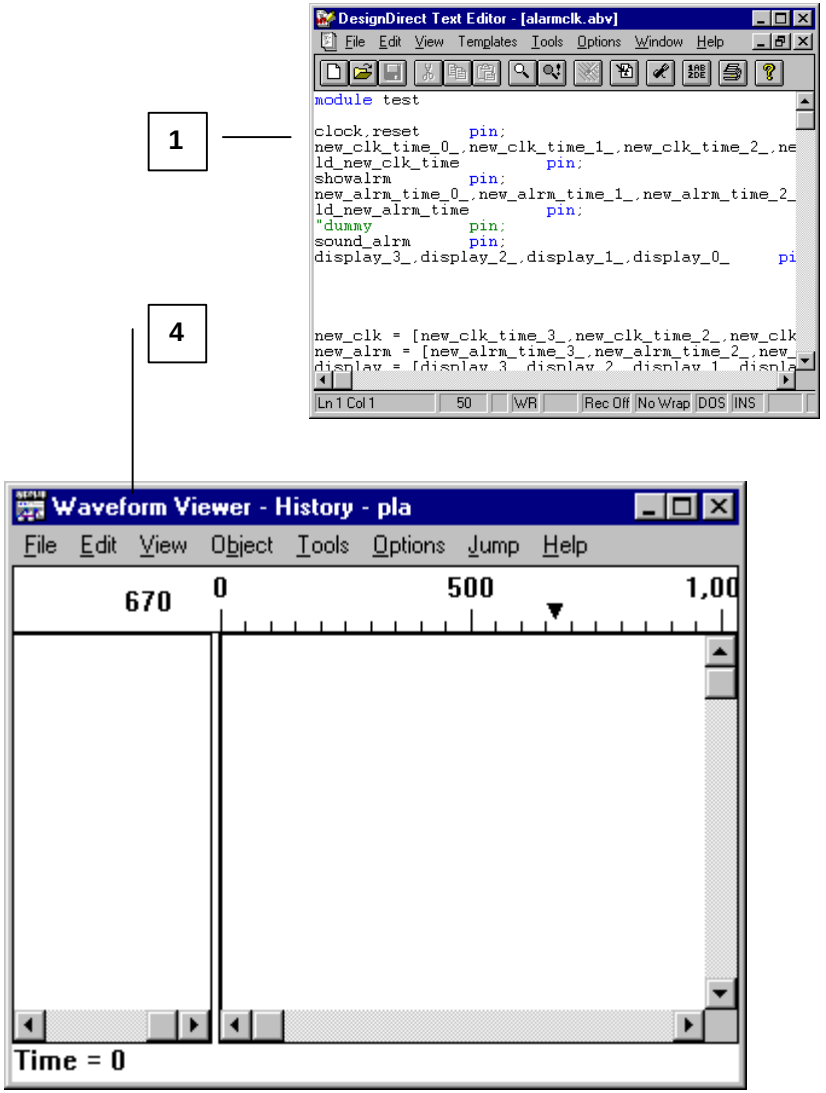
The simulator applies the inputs from the test vectors to the simulated circuit and compares the simulated output with the output specified in the test vectors. If there is any difference, an error is indicated.

Equation simulation tests your design without using device-specific information. Therefore, Equation simulation can be conducted before you select a device. Equation Simulation, however, only tests the equations in your design as specified by test stimulus (ABEL-HDL test vectors).

To behaviorally simulate your design:

1. In the Sources window, double-click **alarmclk.abv** to open the test vector file in the Text Editor.
2. Scroll the window and view the Test Vector file.
3. Close the file and exit the Text Editor.
4. In the Processes window, double-click **Equation Simulation Waveform** to open the Waveform Viewer. The viewer will be empty until you add waveforms to it.

   You can use the Waveform Viewer to view the output of a logic simulation. The state of any net on a schematic can be viewed as a time line trace, similar to what you would observe when using a logic analyzer. Traces may display individual signals or buses.
5. Choose **File** > **Exit** to exit the Waveform Viewer.



Task 9: *Select a Device*

Below the Project Notebook is the device icon that identifies the target device for the project. The Project Manager lets you target a design to a specific Vantis device at any time during the design process.

If you do not know the specific device, you can target a *Virtual Device*. A Virtual Device is a generic, architecture-independent device, allowing you to create a design that does not rely on specific device features to implement.

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

6 In the Sources window, double-click **Virtual Device** to open the Device Selector dialog. The dialog shows the available device families and the devices in the selected family. (The device list can vary depending upon which device kit you have installed. Check with your local representative for a complete list.)

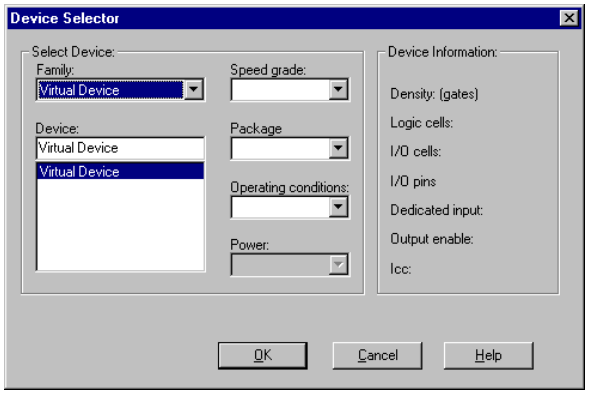
7 In the Device Selector dialog, select the **MACH 4A** device family.

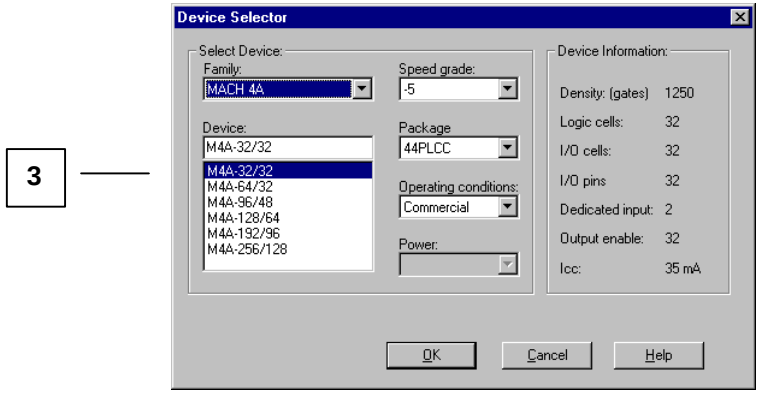
8 Select the **M4A-32/32** device.

9 Accept the default settings and then click **OK**.

10 Click **Yes** to confirm that you wish to change device kits.

When you change device kits, the DesignDirect design environment reconfigures to facilitate designing with the selected device kit.



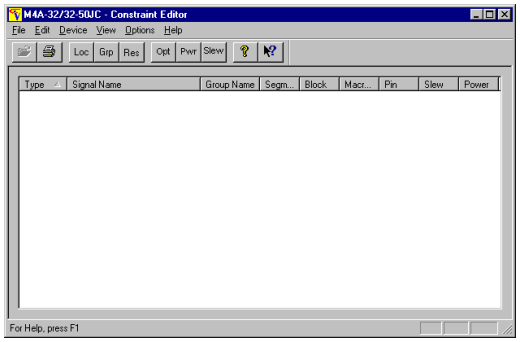


Task 10: *Fit the Design*

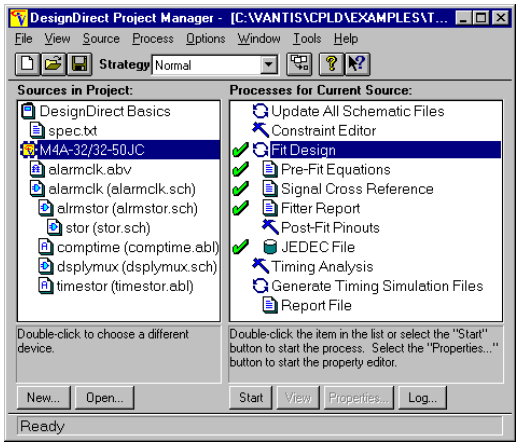
After you have targeted a specific device, you can "fit" your design. The Fitter is a program that maps designs into a targeted device. This process may include assigning pins and nodes, logic synthesis, and complex device routing.

To fit your design:

1. In the Sources window, select the target device (**M4A-32/32**) if it is not already selected.
2. Before running the Fitter, double-click **Constraint Editor** in the Processes window to open the dialog. Notice that there are no constraints assigned.
3. Choose **File** > **Exit** to exit the Constraint Editor.
4. In the Processes window, double-click **Fit Design**. The DesignDirect Process dialog opens to show the progress.
5. When the Fitter is finished running, DesignDirect updates the Processes window with check marks.
6. In the Processes window, double-click **Post-Fit Pinouts**. This dialog shows you the location assignments made by the Fitter.
7. Choose **File** > **Exit** to exit the Post-Fit Pinouts dialog.

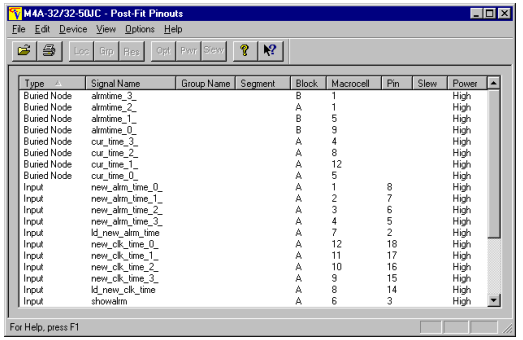


2



5

6



Task 11: *About Auto-Make*

The DesignDirect Auto-Make feature knows which processing steps are up-to-date and which must be re-run when you double-click on a process or resulting file.

Note: *Choose* ***Options > Environment*** *to open the Environment Options dialog and make sure that the Auto-Make feature is selected.*

To see a demonstration of Auto-Make:

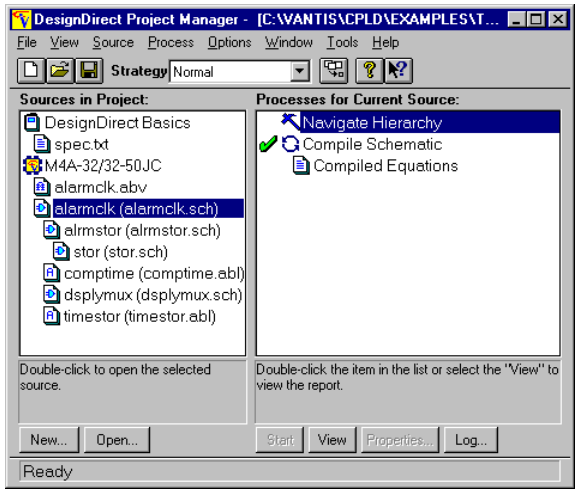
1 Select the schematic **alarmclk.sch** in the Sources window. Notice in the Processes window that there is a green check mark to the left of **Compile Schematic**. This indicates that the process has already run successfully.

2 Double-click **Compile Schematic**. The message bar at the bottom of the Project Manager displays:

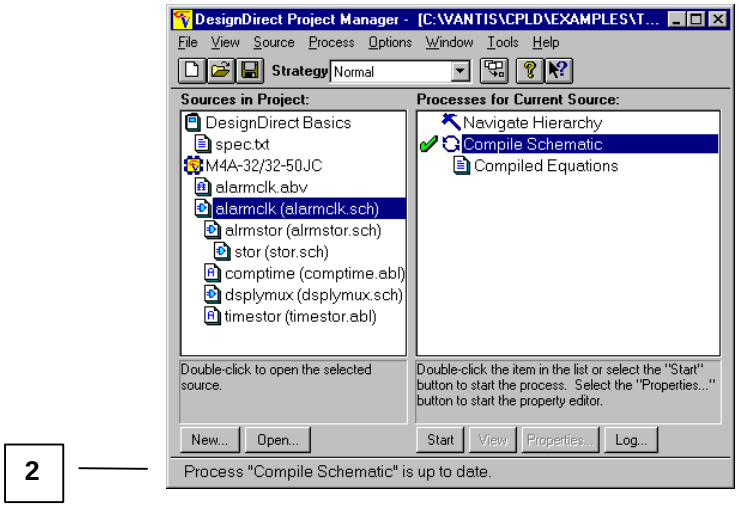
Process “Compile Schematic” is up to date.

The Auto-Make feature tells DesignDirect that it does not have to run this process again because the file is up to date.

1



2



Task 12: *Run Timing Analysis*

The Performance Analyst is a static timing analysis tool that enables you to quickly determine the performance of your design after it has been optimized and implemented by the Fitter.

Worst case signal delays are reported in a graphical spreadsheet format that you can filter to verify the speed of critical paths and identify performance bottlenecks.

To run the timing analysis:

1 In the Sources window, select the target device.

2 In the Processes window, double-click **Timing Analysis** to open the Timing Analyzer.

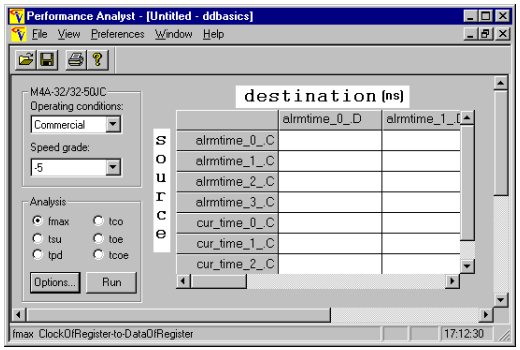
Under Analysis, there are six types of analysis you can perform using the Performance Analyst. Notice the Source and Destination report format on the right.

3 Under Analysis, select **tco** and notice the change in the Source and Destination format.

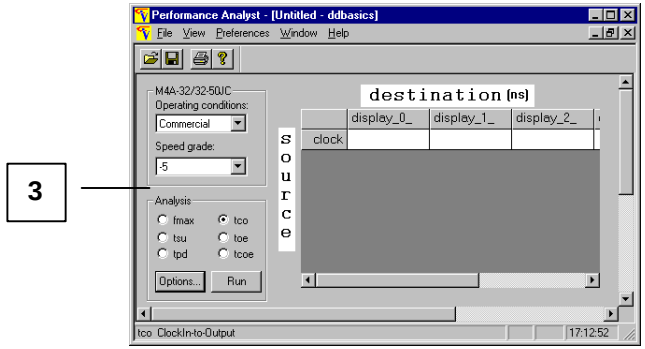
4 Under Timing Analyzer, click **Run** and see the timing report to the right.

You can double-click any number in the report to see how the Performance Analyst generated these numbers.

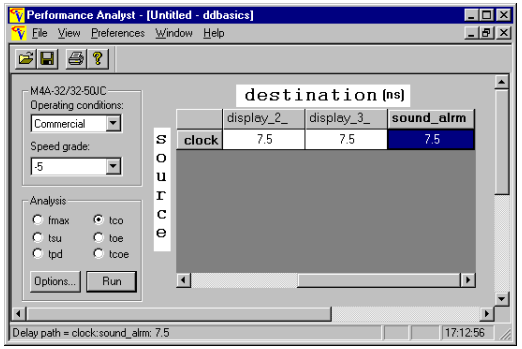
5 Choose **File** > **Exit** to exit the Performance Analyst.



2



3



4

Task 13: *Backannotate Pin Location Assignments*

You can backannotate assignments from the Fitter output using the Backannotation tab on the Constraints Options dialog. This feature lets you retain the assignments made by the Fitter so they can be used in the future.

You can only backannotate project assignments after the "Fit Design" process has been successfully completed.

To backannotate pin location assignments:

1 In the Project Manager, choose **Tools** > **Backannotate Project Assignments** to open the Constraints Options dialog.

2 On the Backannotate tab, select **Pin Assignments**, and then click **Apply**.

3 A warning message appears asking if you want to continue. Click **Yes**.

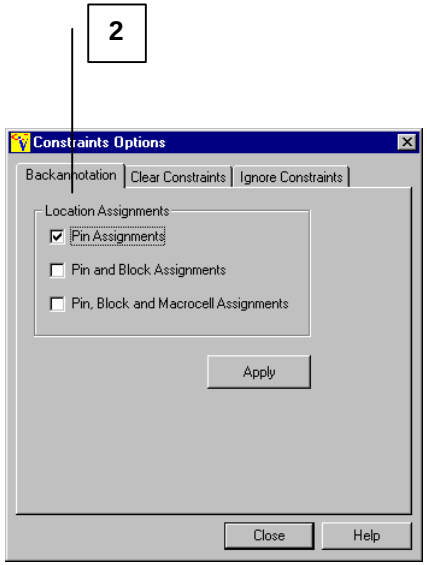
4 Click **Close** to close the dialog.

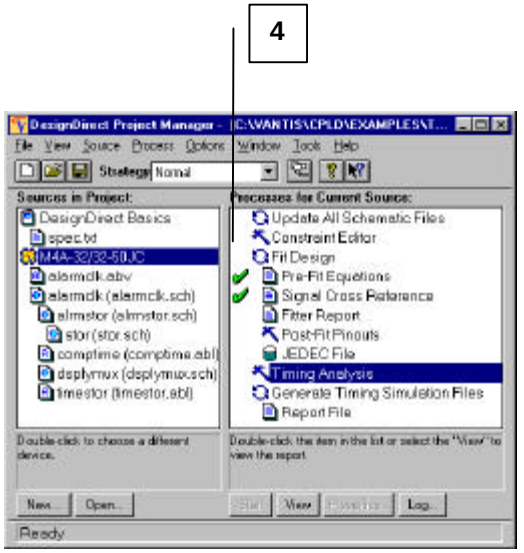
Notice that several process had their green check marks removed. This indicates that the previous fitter results are no longer valid because you have assigned pin location constraints.

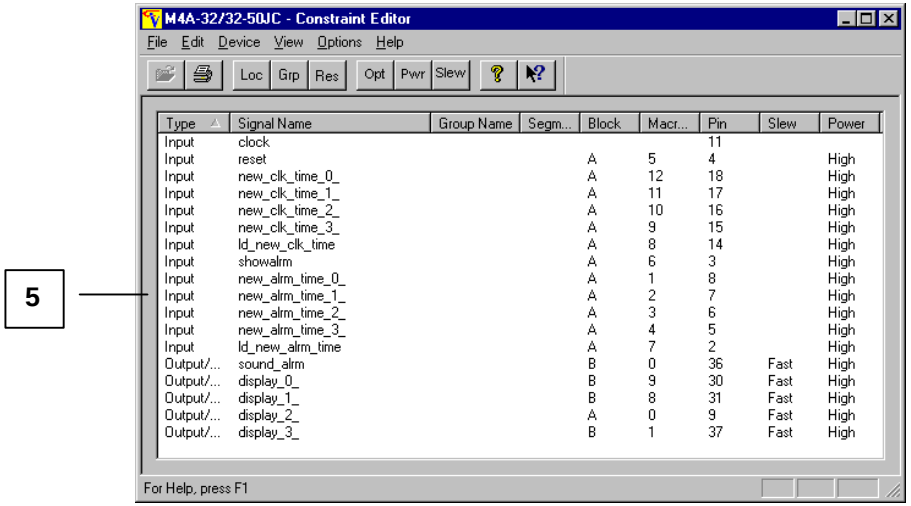
5 Double-click **Constraint Editor** to open the dialog. Notice that there are now pin location constraints assigned.

***Note**: If you wanted to edit certain constraints, you would use this editor (for example, **Edit** > **Location Assignment**).*

6 Double-click **Fit Design** to fit the design with these new constraints.







Task 14: *Congratulations!*

You have completed the DesignDirect Basics tutorial. In this tutorial you have learned:

- how to open a DesignDirect project.
- the different elements of the Project Manager user interface.
- that there are *project*- and *source-level* process flows, and how to switch between them using the Project Manager interface.
- how to create and import source file into a project.
- how to edit sources by double clicking the source file in the Project Manager.
- how to use the Hierarchy Navigator to view a design.
- how to process a design.
- how to run Equation Simulation and open the Waveform Viewer to analyze the results.
- how to select a target device and choose operating options.
- how to use the Location Assignments dialog to assign pins, and then save the information to the constraints file.
- how to place and route a design using the Fitter.
- how the Auto-Make feature eliminates unnecessary processing of already-completed steps.
- how to run static timing analysis on a design, and evaluate the results graphically.
- how to backannotate pin location assignments.

Tutorial 2

Schematic and ABEL-HDL Design Entry

This tutorial leads you through the design of 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.

Prerequisites

Before attempting this tutorial, you should complete Tutorial 1, *DesignDirect Basics*.

Learning Objectives

When you are finished with the tutorial, you should be able to:

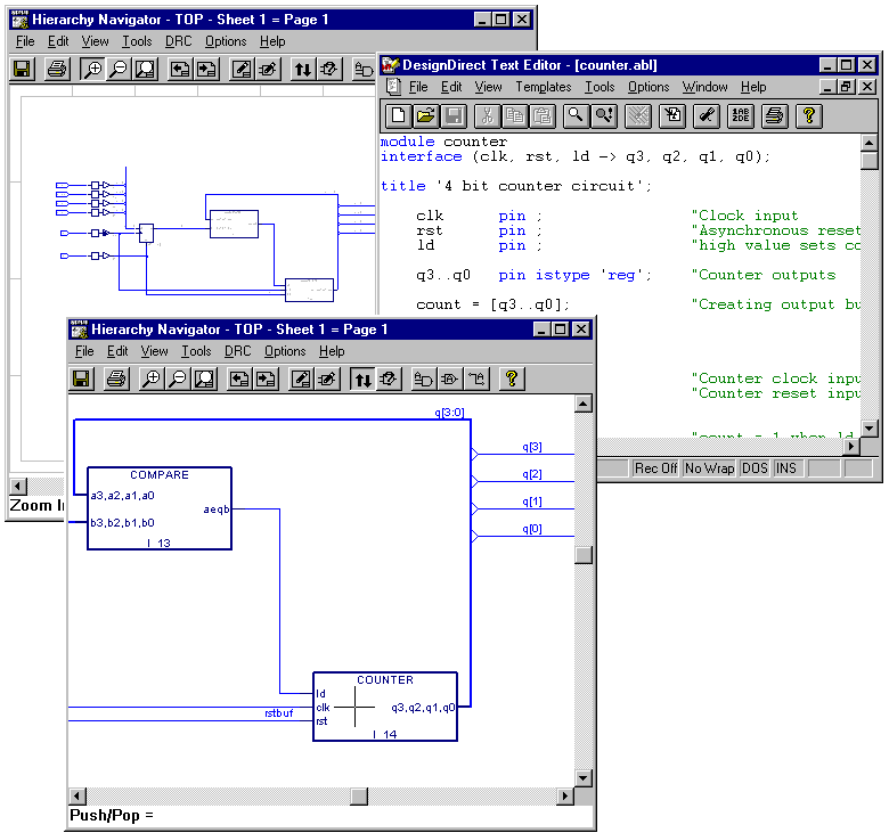
- Add various schematic elements to create a top-level schematic source
- Check the schematic for errors
- Use the Text Editor to create a new ABEL-HLD source
- Import an ABEL-HLD source into the project
- Use the Hierarchy Navigator to navigate through the design and try "debug" methods

Time to Complete This Tutorial

On average, it should take you approximately 90 minutes to complete this tutorial.

Assumptions

This tutorial assumes that DesignDirect-CPLD software is installed and operating properly.

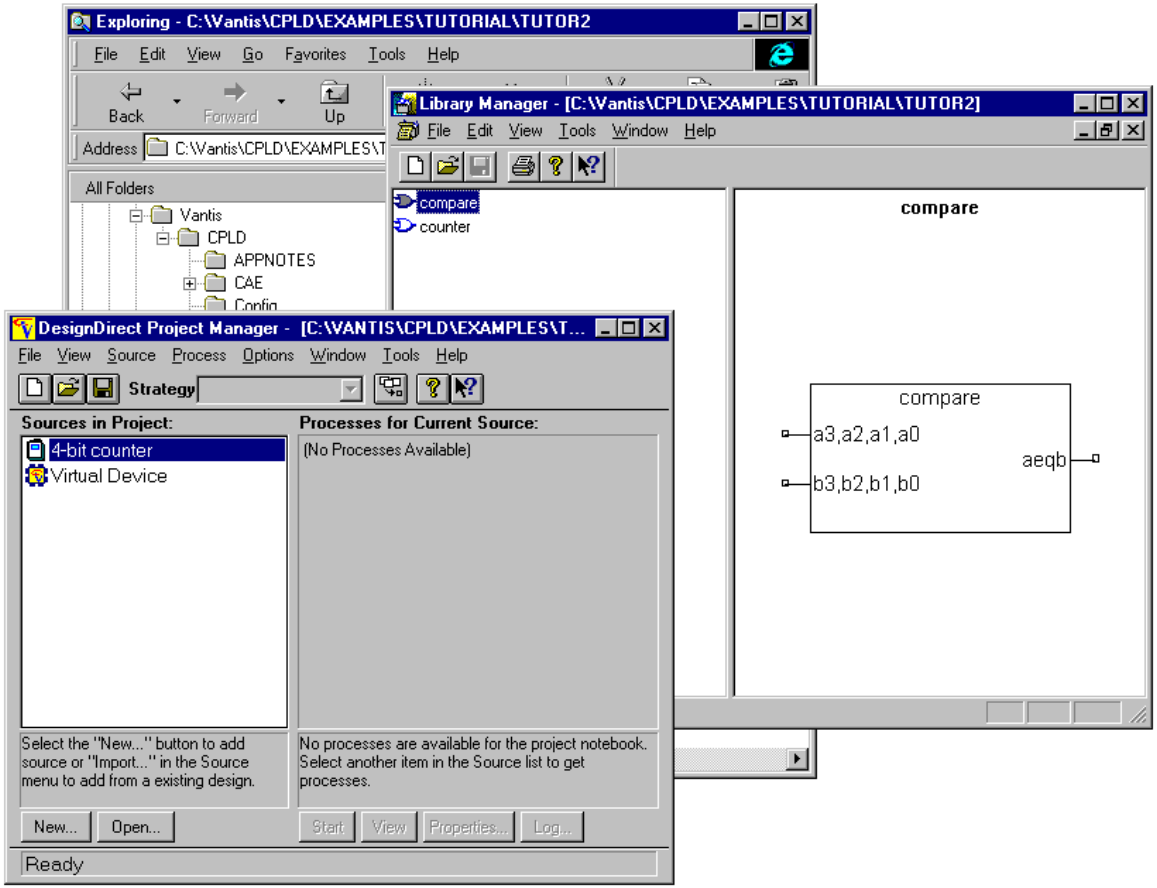


Lesson 1 – Setting Up the Project

In Lesson 1 you will get ready to start the tutorial project.

Tasks covered in this lesson are:

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

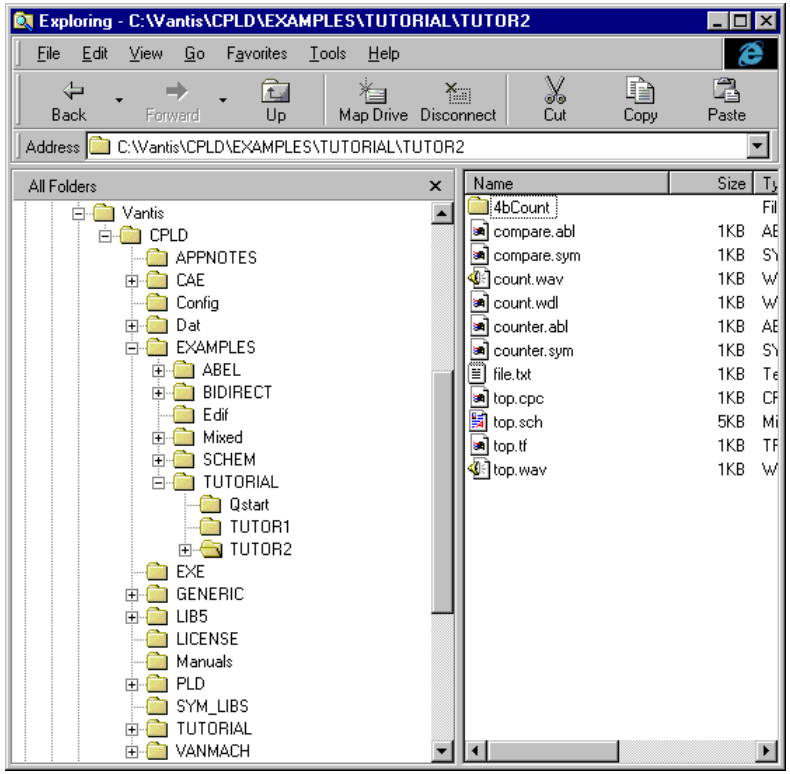


Task 1: *Create a Project Directory*

Before you start this tutorial, you need to create a project directory for all of your tutorial design files.

To create a project directory:

1. Start **Explorer**, or similar tool.
2. Go to the directory **\Vantis\CPLD\Examples\Tutorial\Tutor2**
3. In **Explorer**, create the directory **4bCount**.



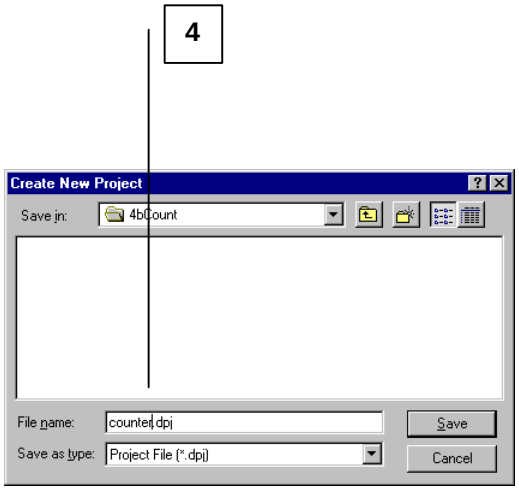
Task 2: *Create a New Project*

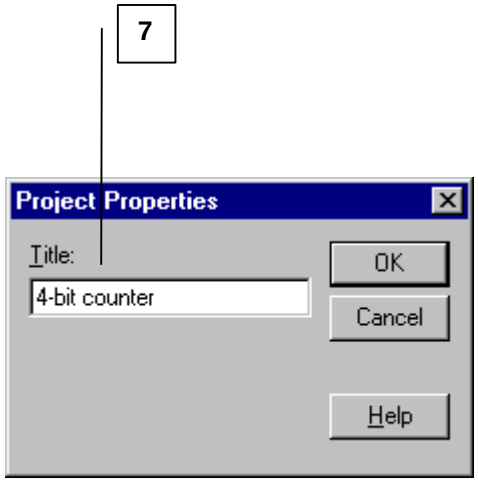
DesignDirect employs the concept of a project. A project is a design. Each project has its own directory in which all source files, intermediate data files, and resulting files are stored.

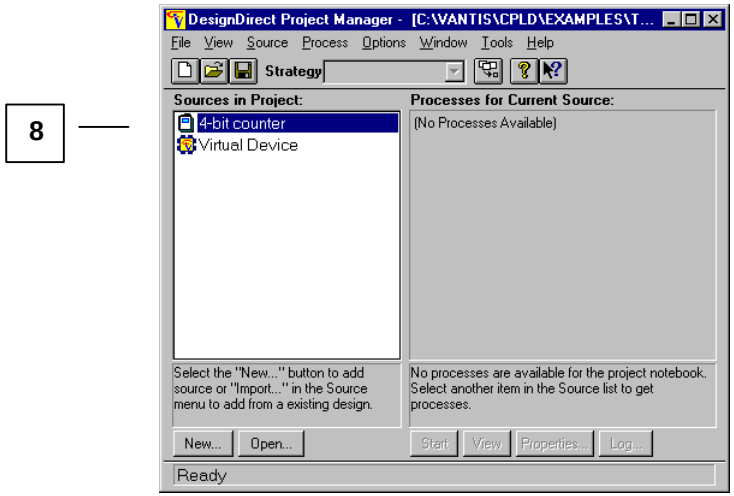
The first step to start a project is to create a project directory. Then, you need to name the project (.dpj) file, which the Project Manager uses later to reload the project.

To create a new project:

1. Start DesignDirect if it is not already running.
2. In the Project Manager, choose **File** > **New Project** to open the Create New Project dialog.
3. Go to the new **4bCount** directory.
4. The default project name is “Untitled.dpj.” The project name can be up to 8 characters long, spaces are not allowed. For this tutorial, name the project file **counter.dpj**.
5. Click **Save**. The untitled generic project appears in the Sources window of the Project Manager.
6. In the Sources window, double-click the title of the project (Untitled) to open the Project Properties dialog.
7. Type the name **4-bit counter** for the title of your project.
8. Click **OK**. The new project title appears in the Sources window.







Task 3: *Copy Schematic Symbols to the Project Directory*

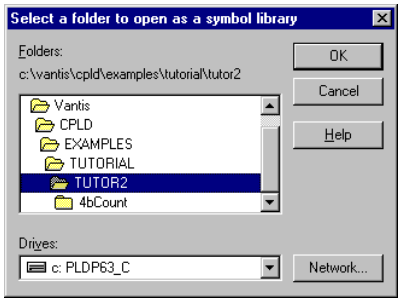
A schematic is composed of symbols, wires, I/O markers, graphics, and text. Symbols are graphic representations of components. The term "symbol" usually refers to an electrical symbol, such as a gate or a sub-circuit. You can draw graphic-only symbols (such as title blocks) with the Symbol Editor, but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

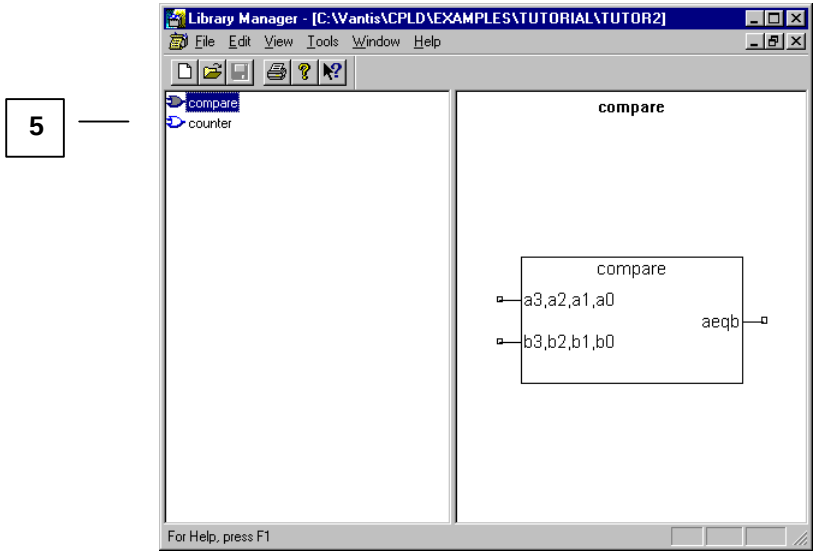
Symbols are the most basic elements of a schematic. Symbols represent primitive design elements, whether those elements are individual transistors, complete gates, or a complex IC. A symbol can also be the hierarchical representation of a sub-circuit (a "Block" symbol).

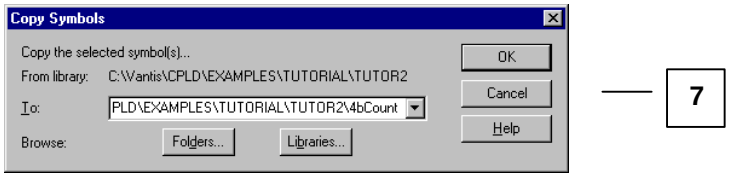
In this Step you will copy two symbol files to your project directory so you can use these pre-made symbols in your design.

To copy schematic symbols to the project directory:

1. In the Project Manager, choose **Window** > **Library Manager**.
2. In the Library Manager, choose **File** > **Open Folder** to open the dialog.
3. Go to the **\Vantis\CPLD\Examples\Tutorial\Tutor2** directory.
4. Click **OK** to open the library in the Library Manager.
5. Select the two symbols (**compare** and **counter**).
6. In the Library Manager, choose **Edit** > **Copy Symbol** to open the Copy Symbols dialog.
7. Click **Folders**, go to the **4bCount** folder, and then click **OK**.
8. Click **OK** to close the dialog.
9. Exit the Library Manager.







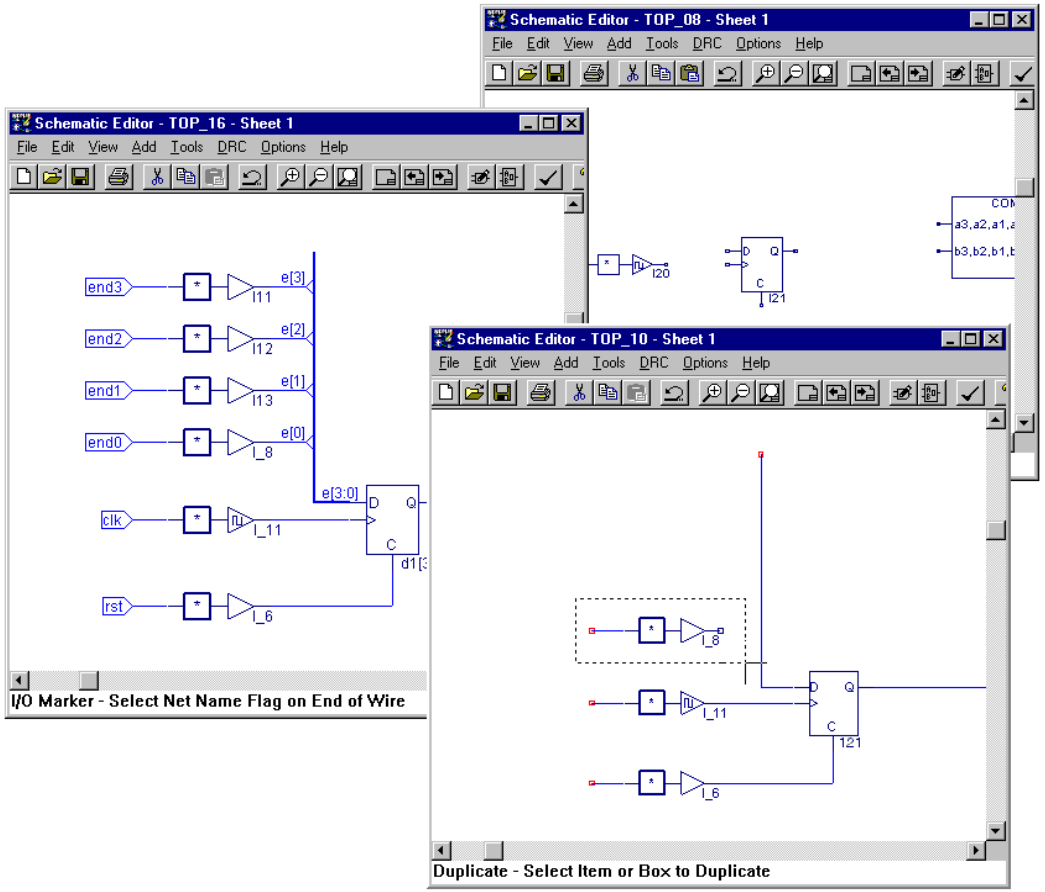
Lesson 2 – Creating a Top-Level Schematic Source

In Lesson 2, you will begin creating a top-level schematic source for the project.

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 to Connect the Symbols
- **Task 10**: Add Wires to Connect the Symbols (con't)
- **Task 11**: Duplicate the Input Pad and Wire Stub
- **Task 12**: Name the Buses
- **Task 13**: Add Bus Taps with Signal Names
- **Task 14**: Add Input Net Names
- **Task 15**: Add Data Input Net Names
- **Task 16**: Create Iterated Instances of the Flip-Flop
- **Task 17**: Add Input Markers

When you are finished with this lesson, the schematic should look like the one on the right.



Task 4: *Add a New Schematic to the Project*

Designing top-down, we'll first create the top-level source for the project. It can be any type of source. Because this design is ABEL-HDL-based, you can use either an ABEL-HDL or schematic source at the top level. For this tutorial we'll use a schematic.

To add a new schematic source to the project:

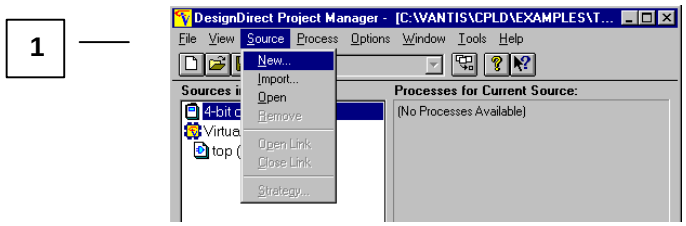
1 In the Project Manager, choose **Source** > **New** to open the New Source dialog.

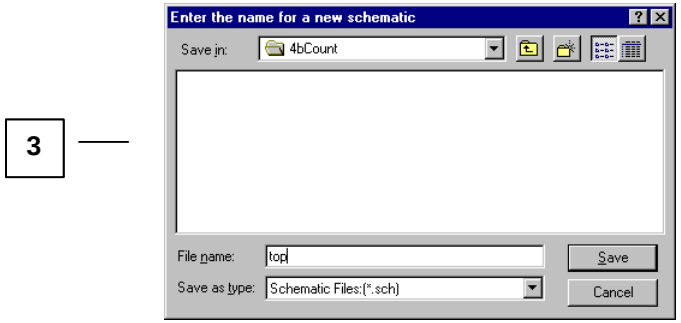
2 Select **Schematic**, and then click **OK**. The Schematic Editor opens and prompts you to enter a file name for the schematic.

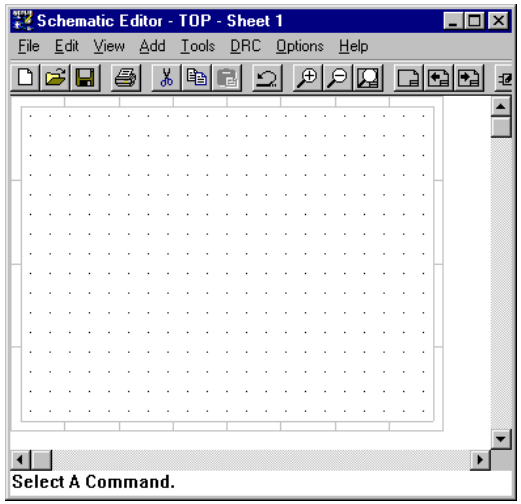
3 Type the name **top**, and then click **Save**.

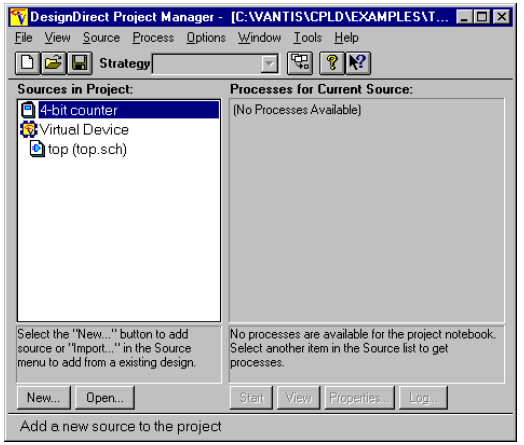
The Schematic Editor names the current schematic sheet as *top*, and the software imports the schematic into the Project Manager as a new source file.

***Note**: If the schematic is difficult to view, see the possible problems and solutions on the next page.*







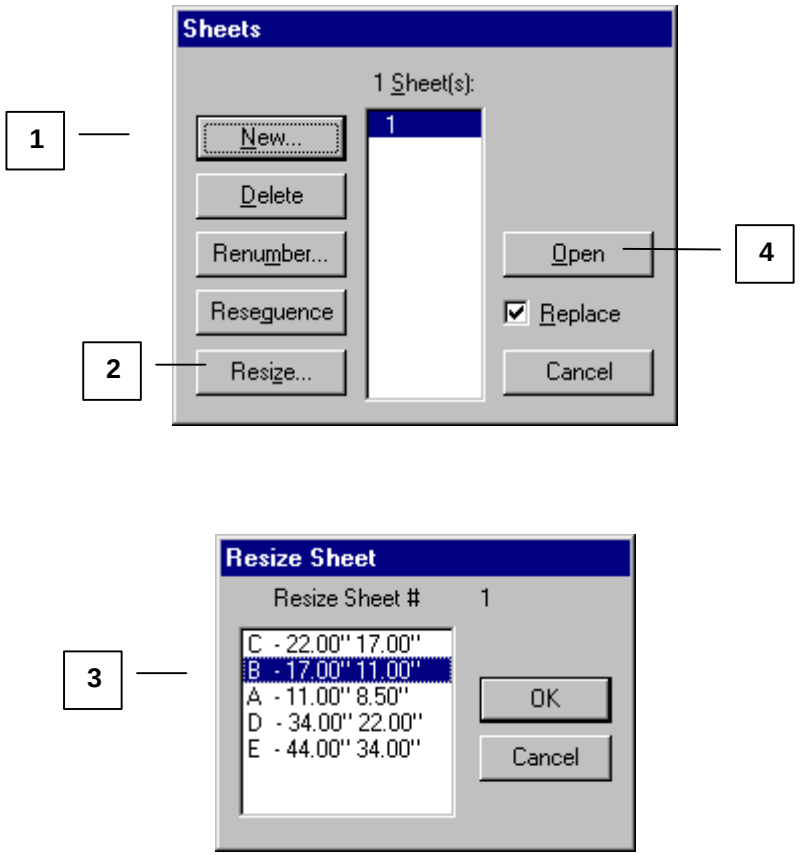


Task 5: *Resize the Schematic Sheet*

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, not the active sheet or the sheet currently being worked on.

To resize the schematic sheet:

1. Choose **File** > **Sheets** to open the Sheets dialog. Because there is only one sheet in the schematic, you cannot select another sheet.
2. Click **Resize** to open the Resize Sheets dialog. The current size of the selected sheet is highlighted. Other available sheet size choices are listed.
3. Select **B**, and then click **OK** to close the Resize Sheet dialog.
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 Choose **Add** > **Symbol** to open the Symbol Libraries dialog.

2 Under Library, select **Local**. This is the library in the project folder you created and copied two block symbols into at the beginning of this tutorial in Step 3.

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

4 Move the cursor to the upper middle of the schematic and click to place the symbol.

5 Right-click to remove the symbol from the cursor.

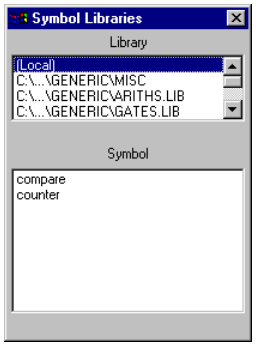
6 Select the **counter** symbol.

7 Move the cursor to the middle of the schematic and click to place the symbol.

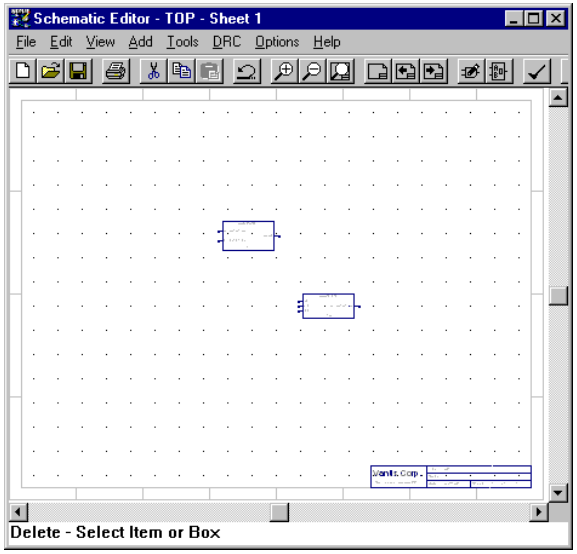
8 Again, right-click to remove the symbol from the cursor.

9 Leave the dialog 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 several unimportant factors that do not affect the proper use of this tutorial.*



1



8

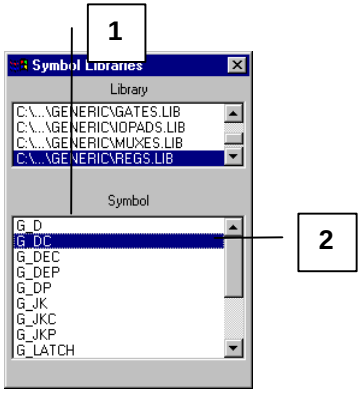
Task 7: *Place a Symbol from the REGS Generic Symbol Library*

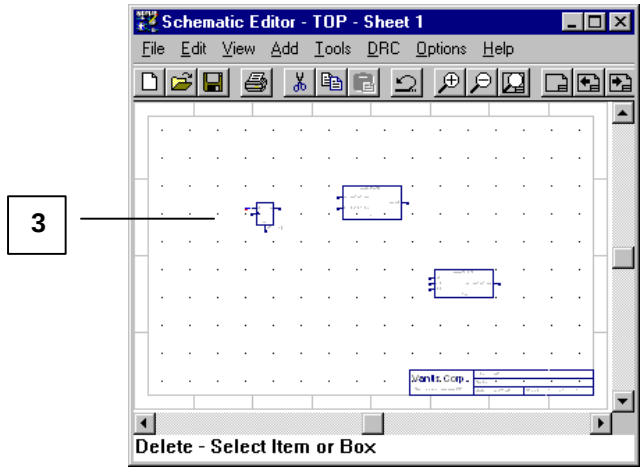
The target device selected for the project determines which symbol libraries are available. For a Virtual Device, only the Device-Independent (also called "Generic") library is available. When an actual device is selected you will have access to the vendor's library as well as the Generic symbol libraries. If you use symbols from the Generic Symbol Library, you can migrate designs to different devices without having to redraw the schematic.

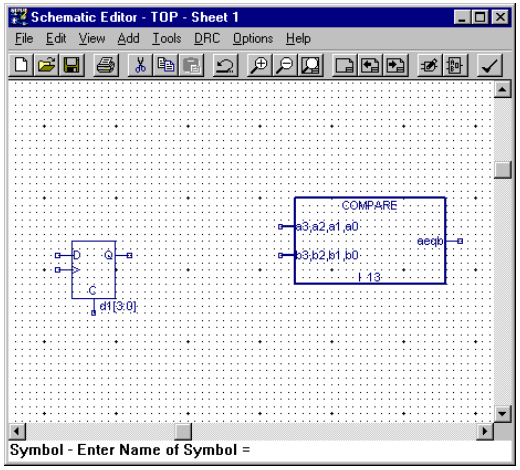
To place a symbol from the REGS generic symbol library:

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

1. In the Symbol Libraries dialog, under Library, select **GENERIC\REGS.LIB**.
2. Under Symbol, select **G_DC**.
3. Place the symbol so the **G_DC** symbol's output is aligned with the **b** input pin on the Compare block symbol. (Editor's Note: Ignore the title block in the schematic.)
4. Choose **View** > **Zoom In** and drag an area to zoom in to a view similar to the one on the right.







Task 8: *Place Three 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**.*

Add a clock buffer symbol:

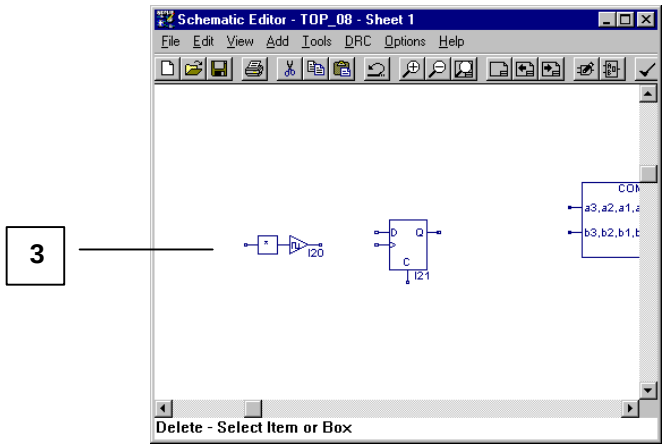
1. In the Symbol Libraries dialog, under Library, select **GENERIC\IOPADS.LIB**.
2. Under Symbol, select **g_clkbuf**.
3. Place the symbol so that the output is aligned with the **clk** input pin on the **g_dc** symbol.

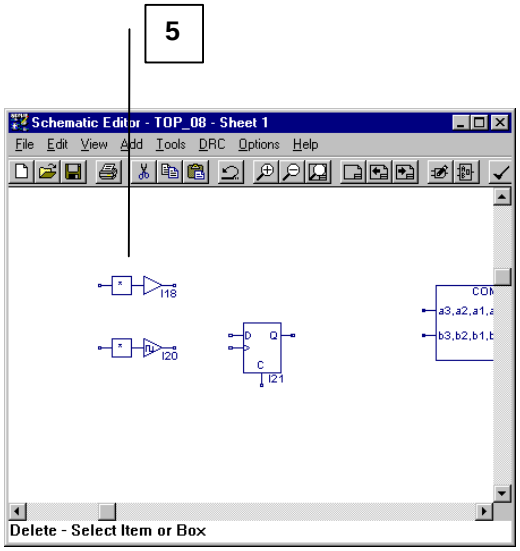
***Note**: The spacing of symbols in the schematic is very important. If symbols are placed too close together, you will not be able to add wires, bus taps, etc. between the symbols. In the following step, make sure the horizontal distance between the **g_clkbuf** symbol and **g_dc** symbol is equal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

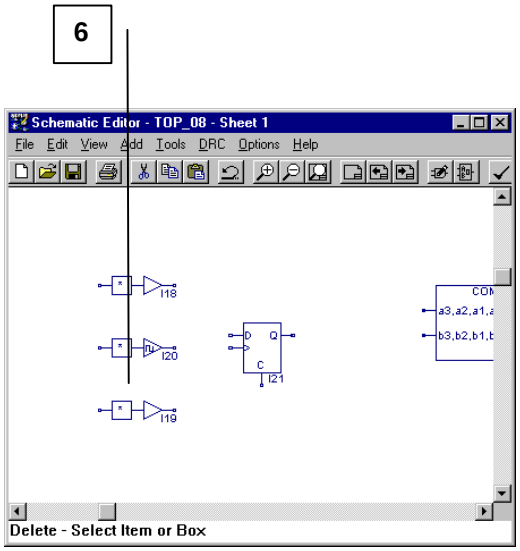
Add two input buffer symbols:

4. Under Symbol, select **g_input**.
5. Click once to place an instance. (Try to align symbols vertically as shown in the picture.)
6. Click once again to place an instance.

***Note**: The Add Symbol command will be canceled when you select the next command in the next section.*







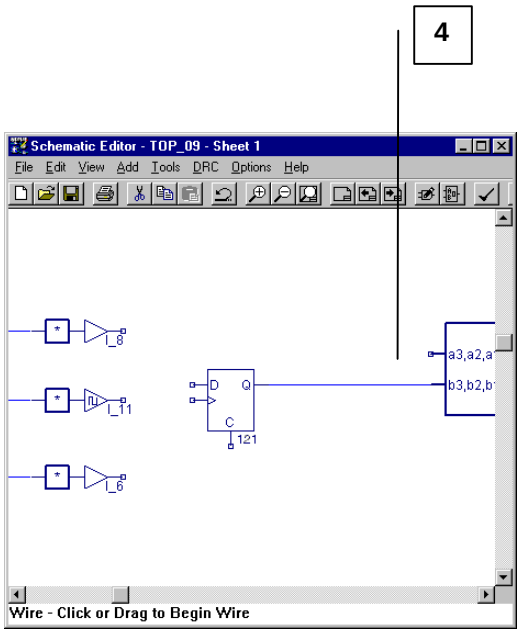
Task 9: *Add Wires to Connect the Symbols*

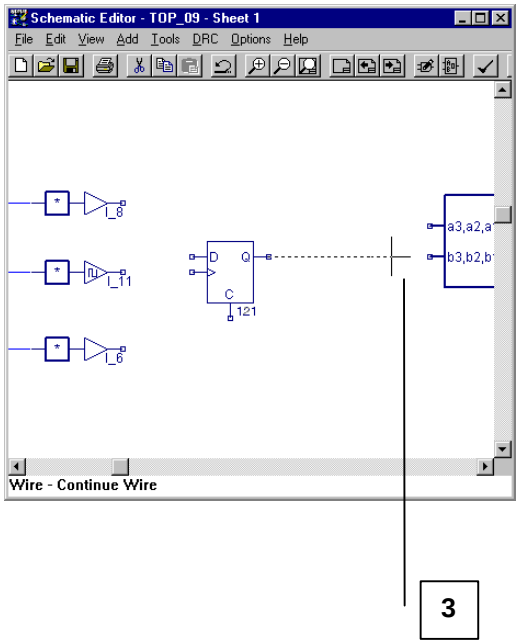
The next step is adding wires and buses to interconnect the symbols. You draw wires and buses the same way; the Schematic Editor knows if a wire is a bus or a single net by the wire's name. In this Step, you will draw all of the wires. Later, you will add names to the wires.

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 the **b** input of **Compare**.
4. Click on the **b** pin to end the wire segment.

***Note**: This Step is continued on the next page.*





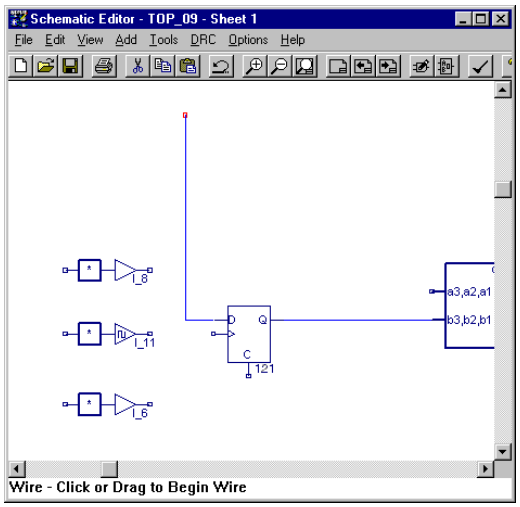
Task 10: *Add Wires to Connect the Symbols (con't)*

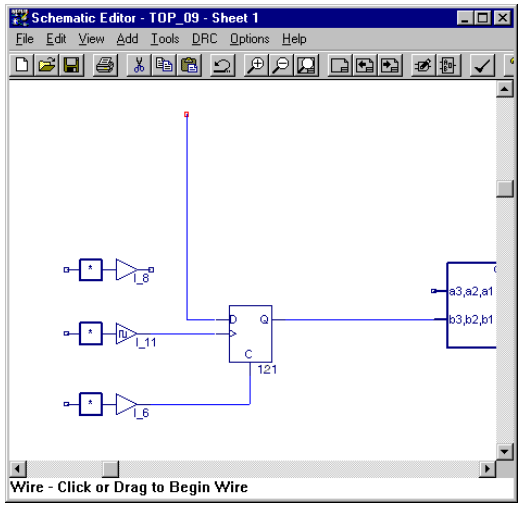
In the continuation of this step, you will add three wires to the flip-flop. Also, you will add wire stubs to the buffers

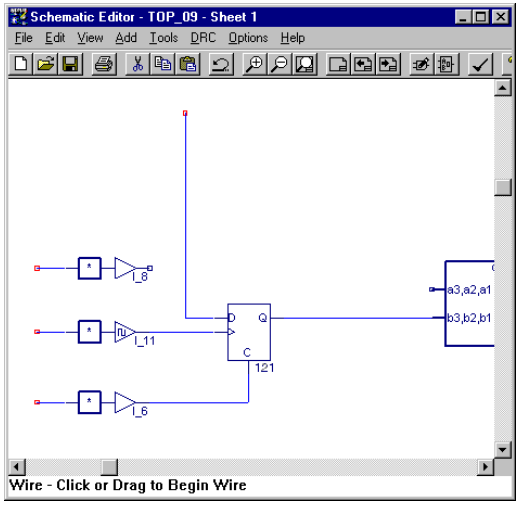
The length of the horizontal and vertical segments of the D input wire (Step 5) is important. The wire segments must be long enough so that you will have room to add a bus name to the horizontal wire segment and add bus taps between the vertical wire segment and input buffers.

Add a wire from D input:

1. 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.
2. Move the cursor up until the vertical height of the wire is about equal to 3 times the height of the flip-flop.
3. Double-click to end the wire.
4. Add a wire from the **C input** to the **input buffer**.
5. Add wire from the **clock input** to the **clock buffer**.
6. Add wire stubs (short wires) to the **input buffers** and the **clock buffer**. (Double-click to end a wire in space, or use the right mouse button to cancel the command.)







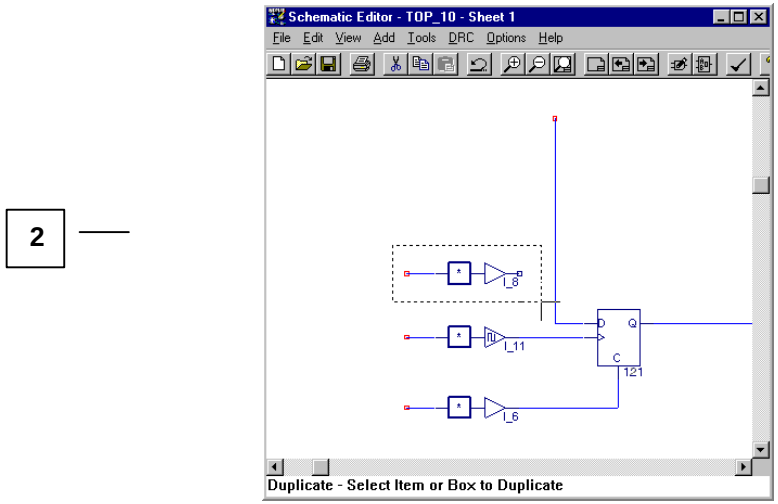
Task 11: *Duplicate the Input Pad and Wire Stub*

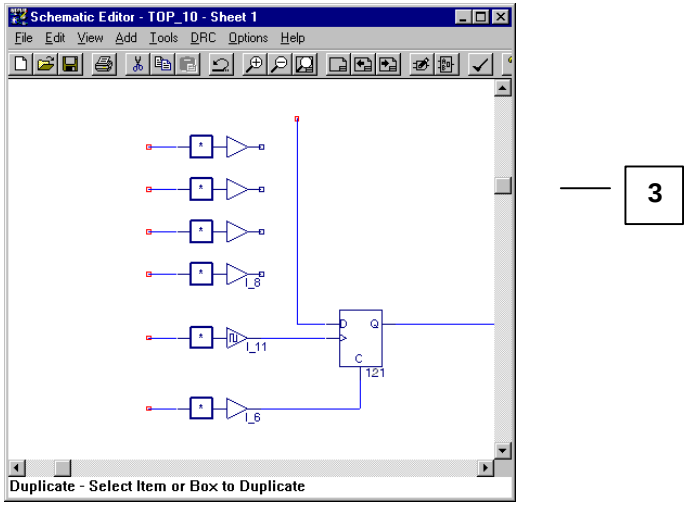
This step shows you how to use the duplicate command to quickly add the rest of the input pads.

The Duplicate command lets you copy one or more elements, 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 the duplicated buffer and wires until there are 4, as shown in the drawing on the right.
4. Right-click to cancel the command.





Task 12: *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 or not they are related.

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 **{ }.**

To name the buses:

1 Choose **Add** > **Net Name**.

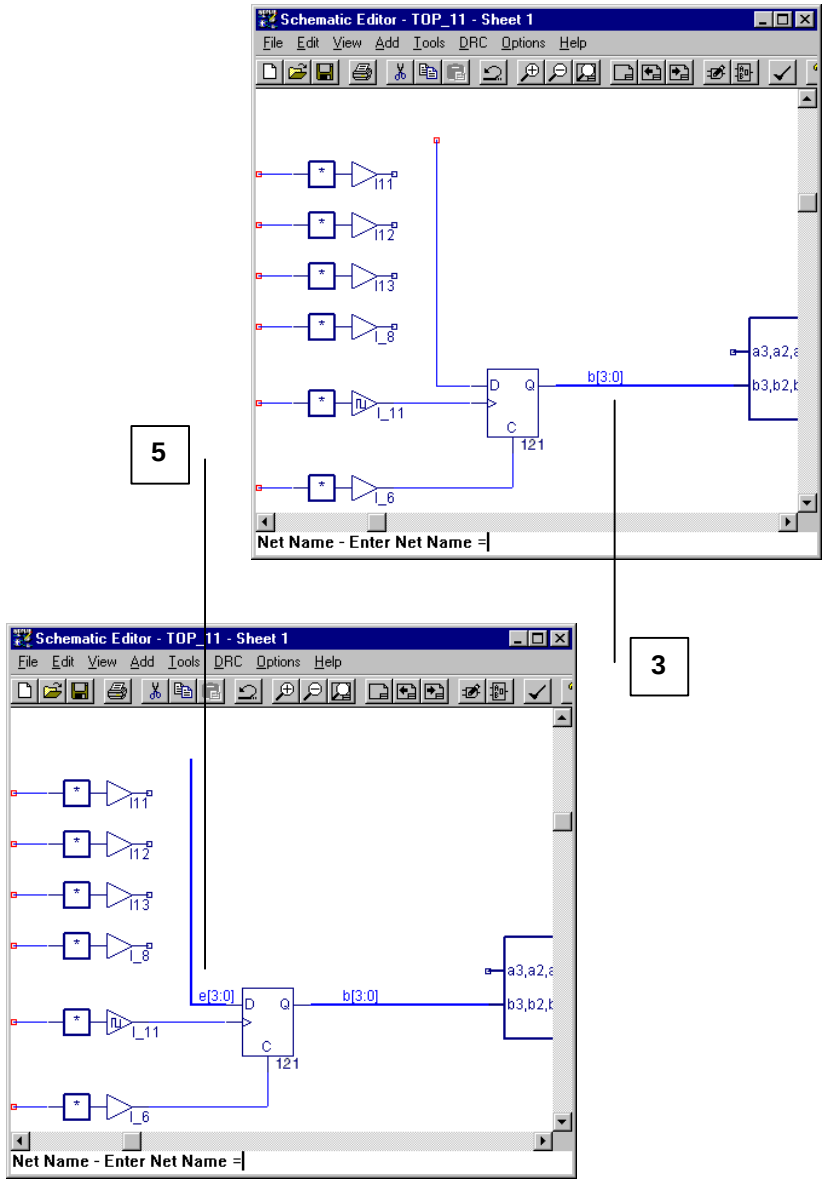
2 In the status area at the bottom of the Schematic Editor, type the name of the bus: **b[3:0]**, and then press **ENTER**. The name attaches to the cursor.

3 Click the wire between the flip-flop and the compare symbol to place the name on the wire. Also, when a wire is named as a bus, its shape thickens to indicate it is a bus.

4 Now name another bus. In the status area, type the name: **e[3:0]**, and then press **ENTER**.

5 Click the horizontal segment of the wire connected to the flip-flop input.

***Note**: Names should be placed on wires, not pins.*



Task 13: *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.

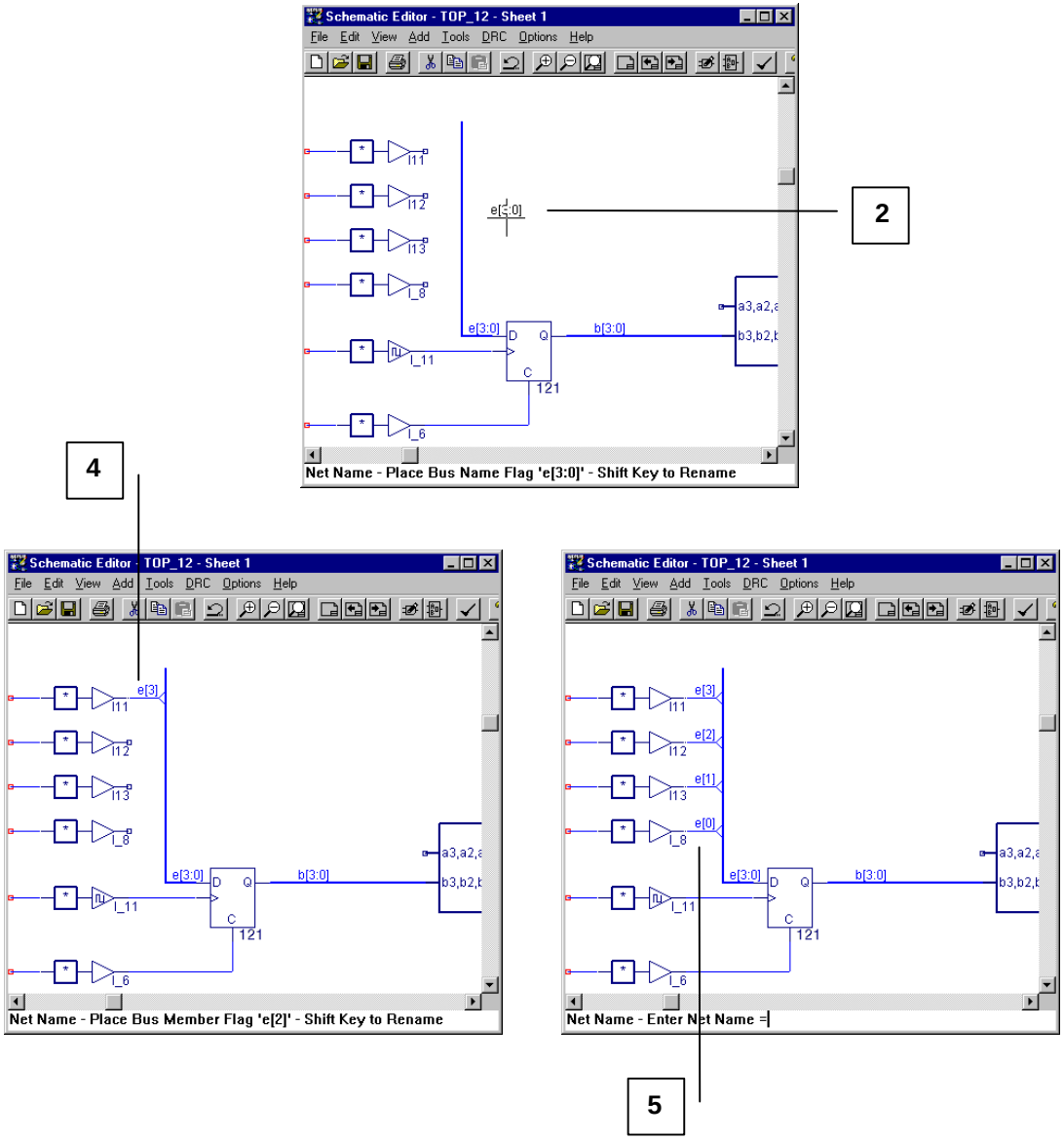
There are several ways to add bus taps. The following procedure shows you how to create a tap, the connecting wire, and the net name in one simple step using the Net Name command.

To add bus taps with signal names:

1. Choose **Add** > **Net Name**.
2. Click the bus labeled **e[3:0]**. The cursor pick 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.
5. Add the remaining taps (in decending order) just by clicking on the pins of the buffers.



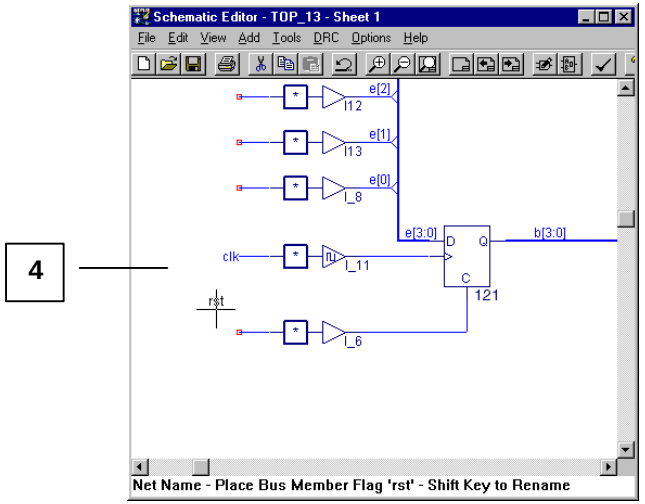
Task 14: *Add Input Net Names*

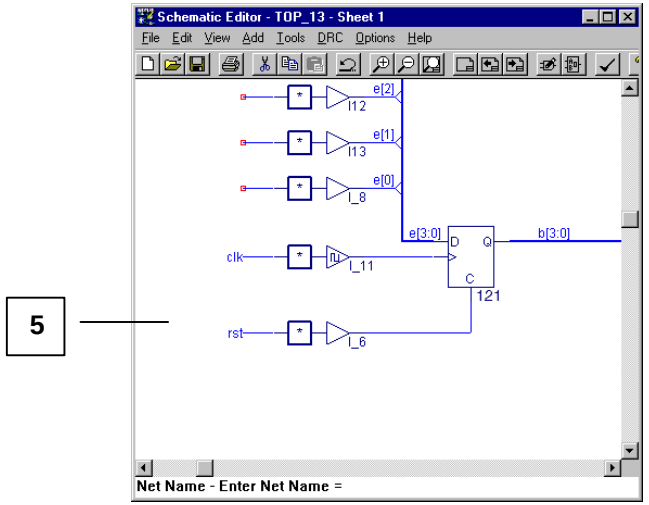
Every net has a name, either assigned by you or by the Schematic Editor. You can override any name assigned by the Schematic Editor by assigning one of your own using the Net Name command from the Add menu.

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. In the status area, 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.
5. Click the end of the rst input buffer to add the final name (**rst**) to the wire.





Task 15: *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**.

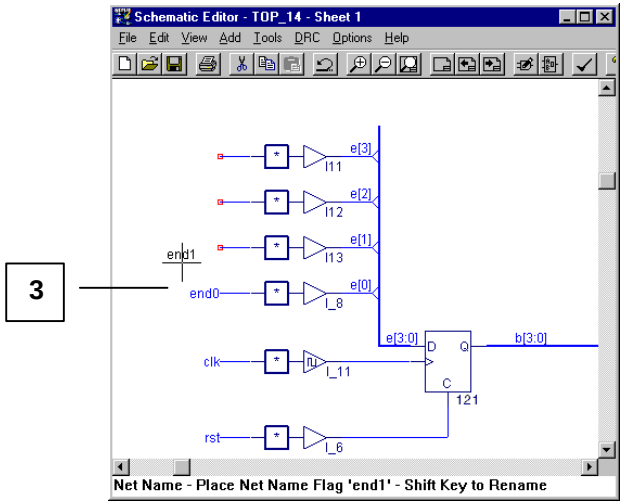
2 In the status area, type: **end0+**, and then press **ENTER**. The name is attached to the cursor.

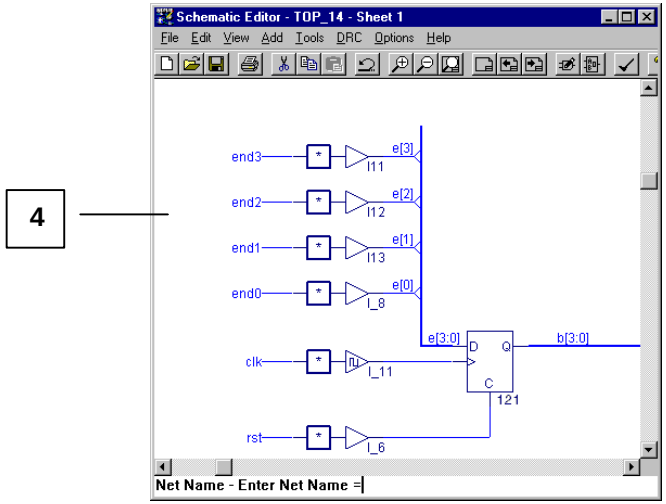
This tells the Schematic Editor to name the signal **end**, to start numbering at **0**, and to increment (+) the numbers as the names are placed.

3 Click on the bottom data input wire stub. The Schematic Editor places the name **end0** and automatically increments the net name.

4 Click on the next wire stub. Repeat until all data input wire stubs have net names.

5 Right-click when **end4** appears on the cursor to cancel the command.





Task 16: *Create Iterated Instances of the Flip-Flop*

A powerful feature in the schematic is its capability of using *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:

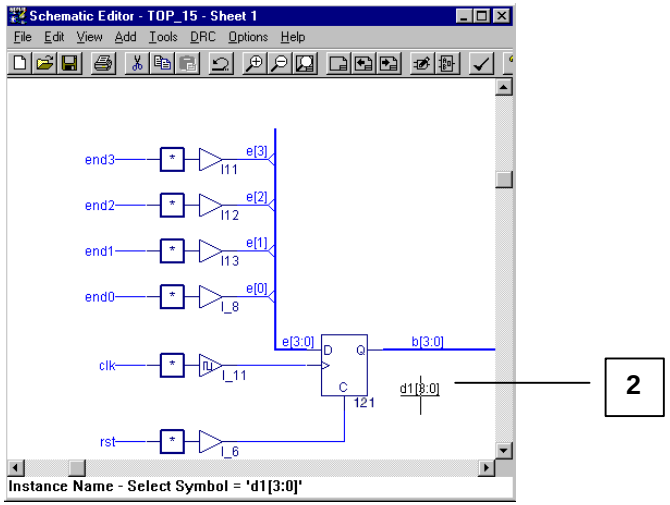
INV[3-10]

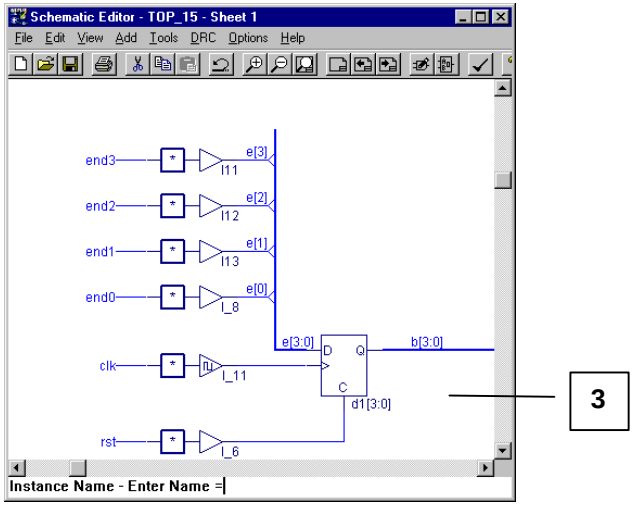
In this case, eight instances of the symbol you've named INV are created, but the symbol appears only once in the schematic.

To create iterated instances of the flip-flop:

1. Choose Add > Instance Name.
2. In the status area, type: **d1[3:0]**, and then press **ENTER**. The name is attached to the cursor.
3. Click once on the flip-flop. 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 17: *Add Input Markers*

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

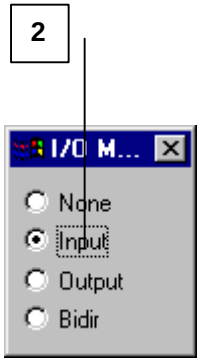
The Schematic Editor Consistency Check command uses I/O markers to flag any discrepancies in the polarity of marked signals and the symbol pins. Discrepancies in polarity are also flagged each time you run the Hierarchy Navigator.

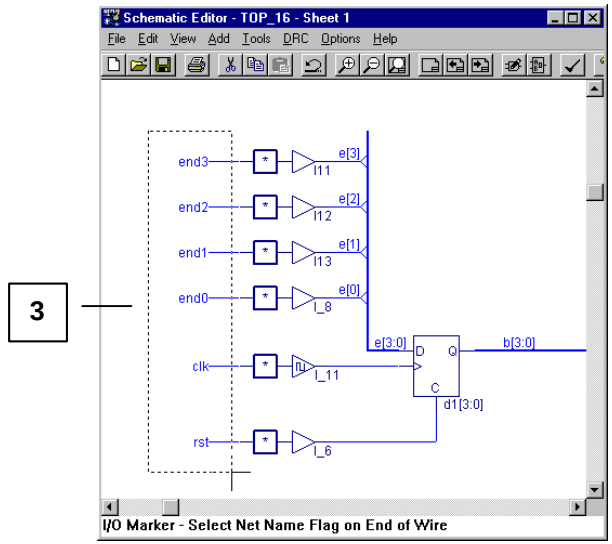
To add input markers:

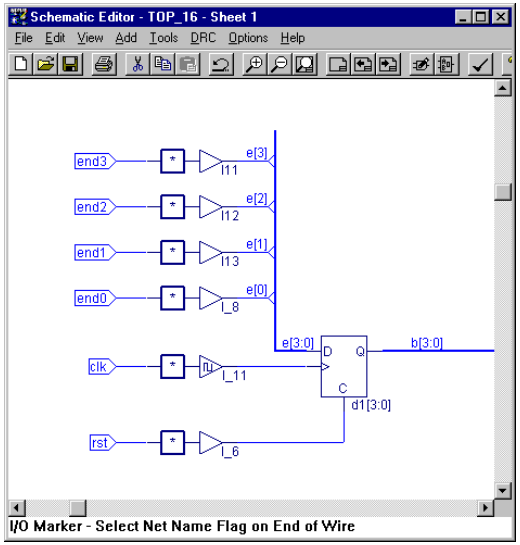
1. Choose **Add** > **I/O Marker** to open the dialog.
2. Select **Input**.

   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.
3. 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.
4. Close the I/O Marker dialog.

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







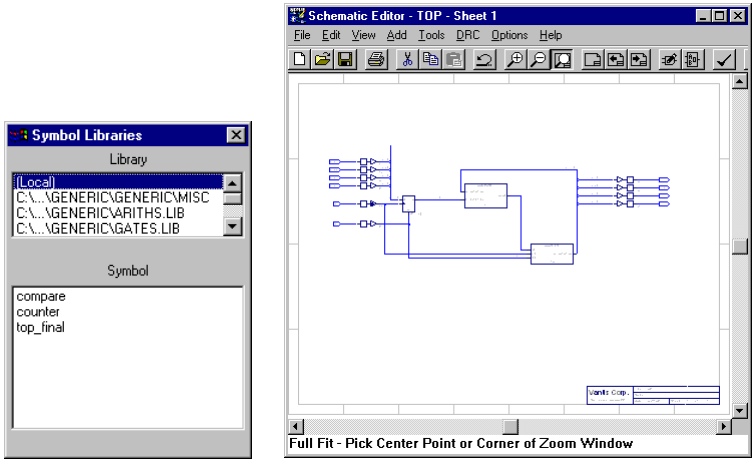
Lesson 3 – Finishing Up the Schematic

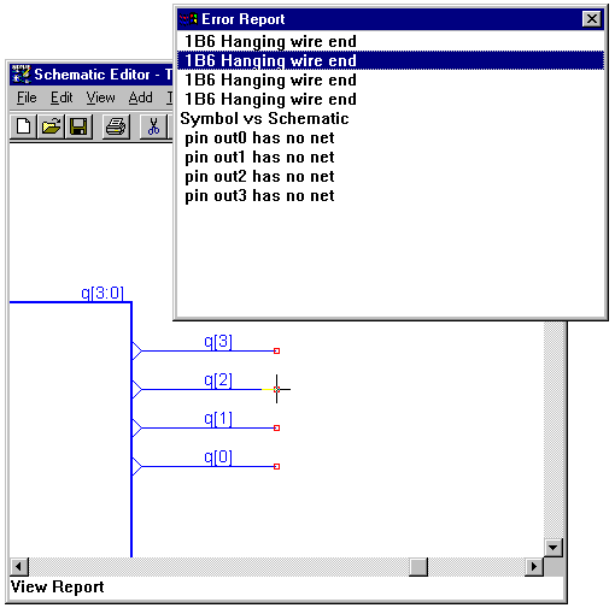
To save time, we have completed the schematic for you.

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. Also, you are shown how to check your schematic for design rule violations.

Steps covered in this lesson are:

- **Task 18**: Import the completed schematic
- **Task 19**: Create a Matching Symbol
- **Task 20**: Check the Schematic for Consistency Errors



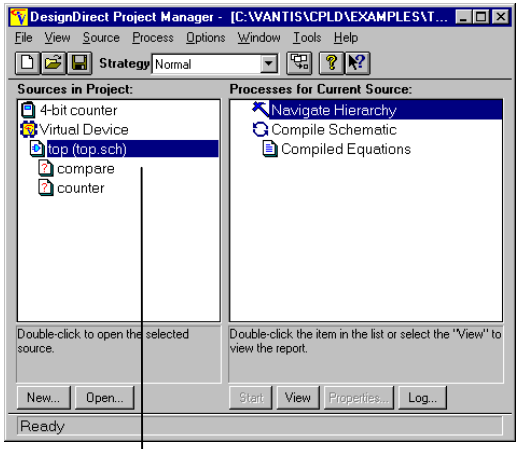


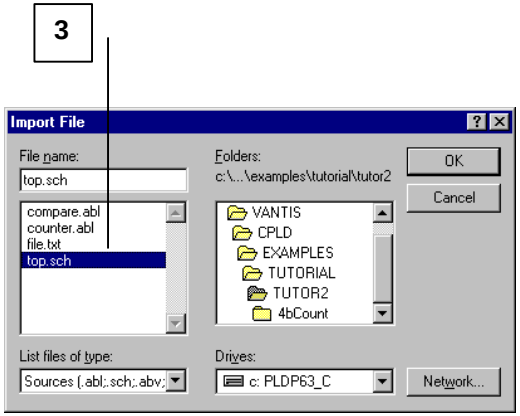
Task 18: *Import the Completed Schematic*

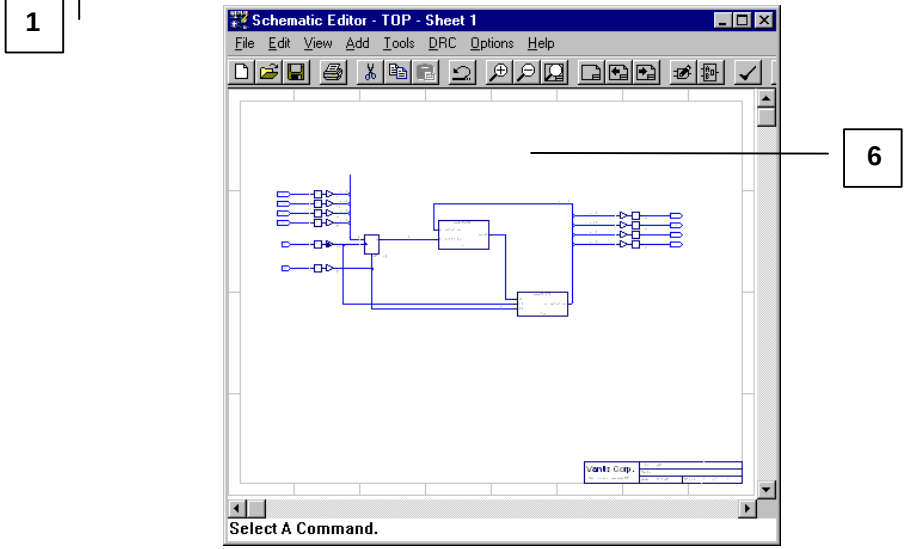
To save time, we have completed the schematic for you. In this step, you will remove the schematic source you have been working on and import the completed schematic of the same name.

To import the completed schematic:

1. In the Sources window, select the **top** schematic source.
2. Choose **Source** > **Remove** to delete the source from the project.
3. Choose **Source** > **Import** to open the Import Source dialog. Go to the **TUTOR2** direcctory and select **top.sch**. Then click **OK**.
4. Click **Yes** to confirm file replacement.
5. The completed schematic is now a top-level source in the project.
6. Double-click the **top** source to open the schematic in the editor.







Task 19: *Create a Matching Symbol*

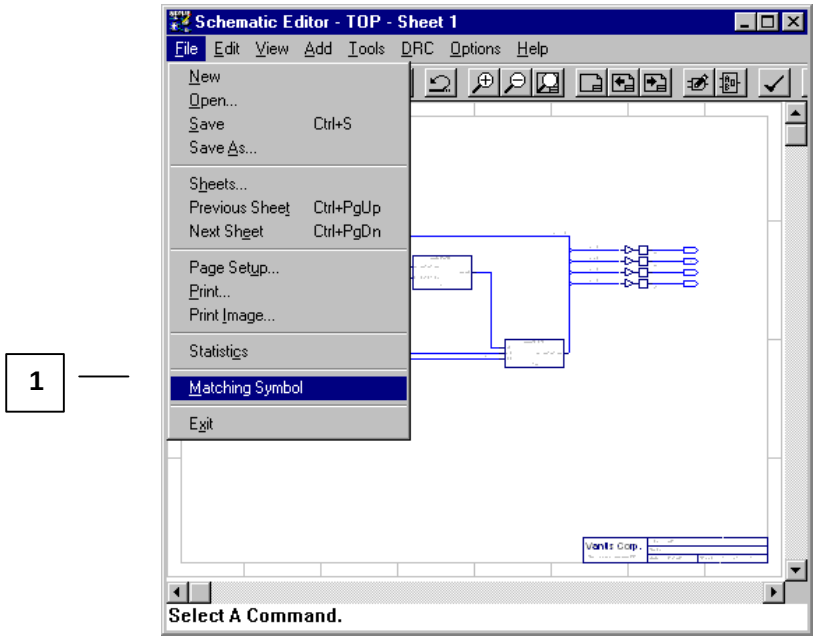
You can use the Matching Symbol command to create a symbol file (*.sym) for the schematic currently loaded, with the same base name. The input and output pins on the symbol have the same signal names and polarities as the I/O markers in the schematic.

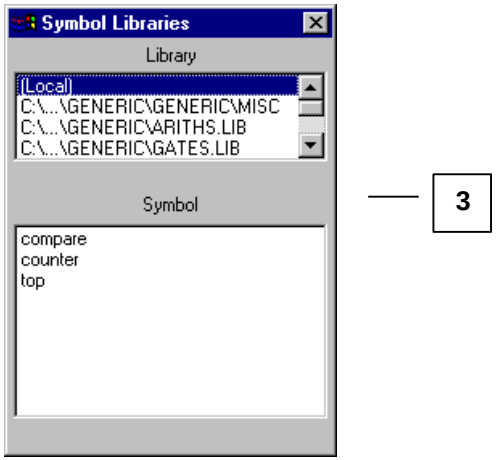
The Schematic Editor creates the symbol in the same directory as the schematic. You can use the Add Symbol command to insert the symbol into any other schematic.

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.
3. Under Library, scroll to the top and select **(Local)**. Notice the symbol named **top**.





Task 20: *Check the Schematic for Consistency Errors*

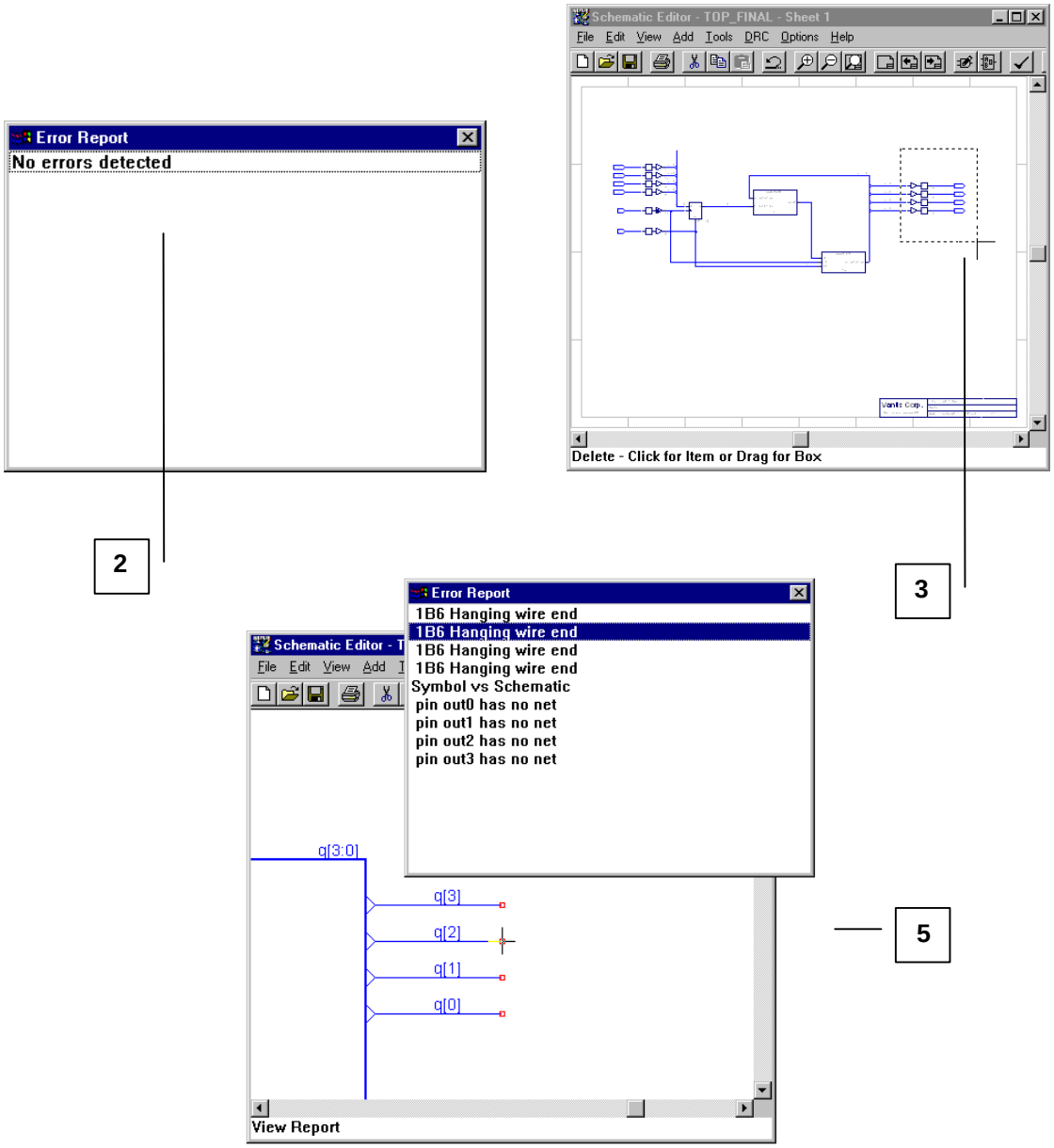
The Schematic Editor continually checks for errors, such as closed loops and shorted nets, while you're drawing your schematic. You can also check your schematic for other errors such as unconnected wires or pins or an unnamed signal tapped from a bus. Errors found are shown in a "hot" list box. Clicking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

To check for consistency errors:

1. Choose **File** > **Save** to save the schematic.
2. Choose **DRC** > **Consistency Check** to open the Error Report.

   There should be no errors in the report.
3. Choose **Edit** > **Delete** and delete the circuitry on the right side of the schematic.
4. Choose **DRC** > **Consistency Check** again.
5. In the Error Report, select an error. Notice that the schematic view shifts to the location of the selected error.
6. Choose **File** > **Exit** to exit the schematic. When asked if you wish 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

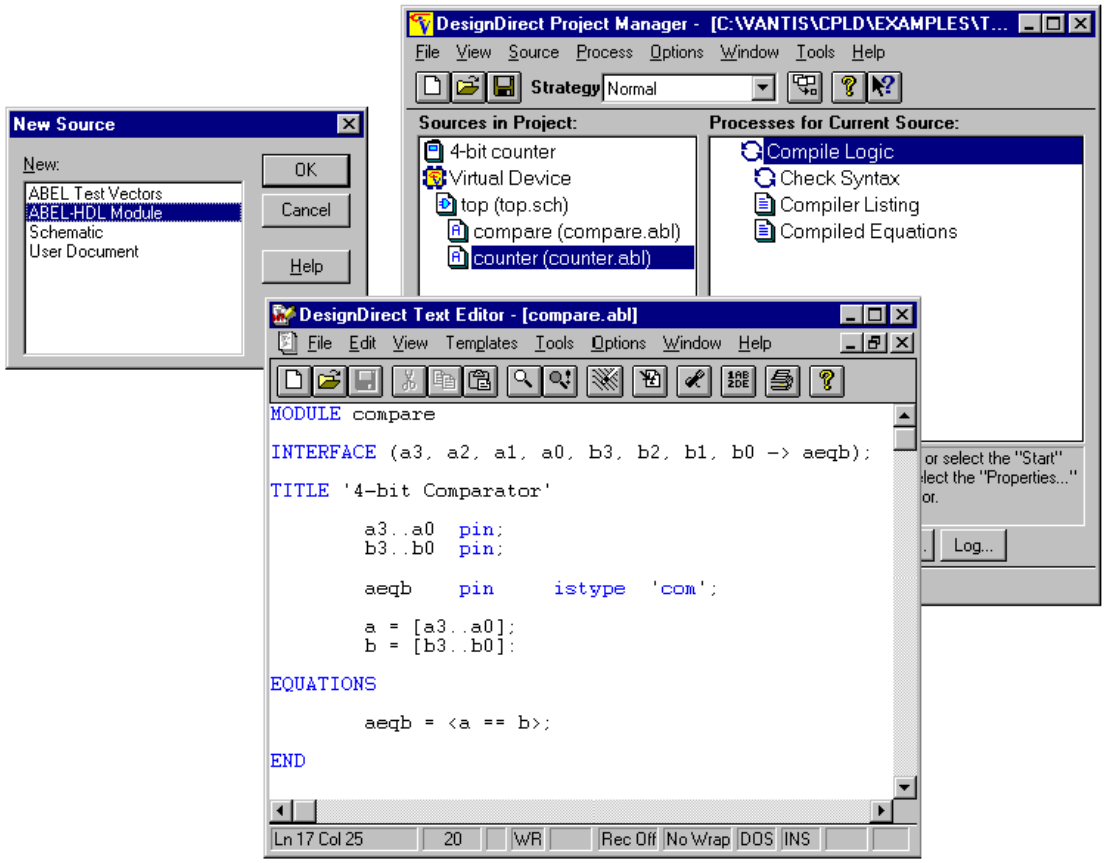
The Project Navigator now lists the schematic, **top**, and the two block symbols referenced in the schematic, **Counter** and **Compare**. These sources have the undefined icon next to them because they do not exist yet. Remember that this is a top-down design example.

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

.



Task 21: *Create a New ABEL-HDL Source File Template*

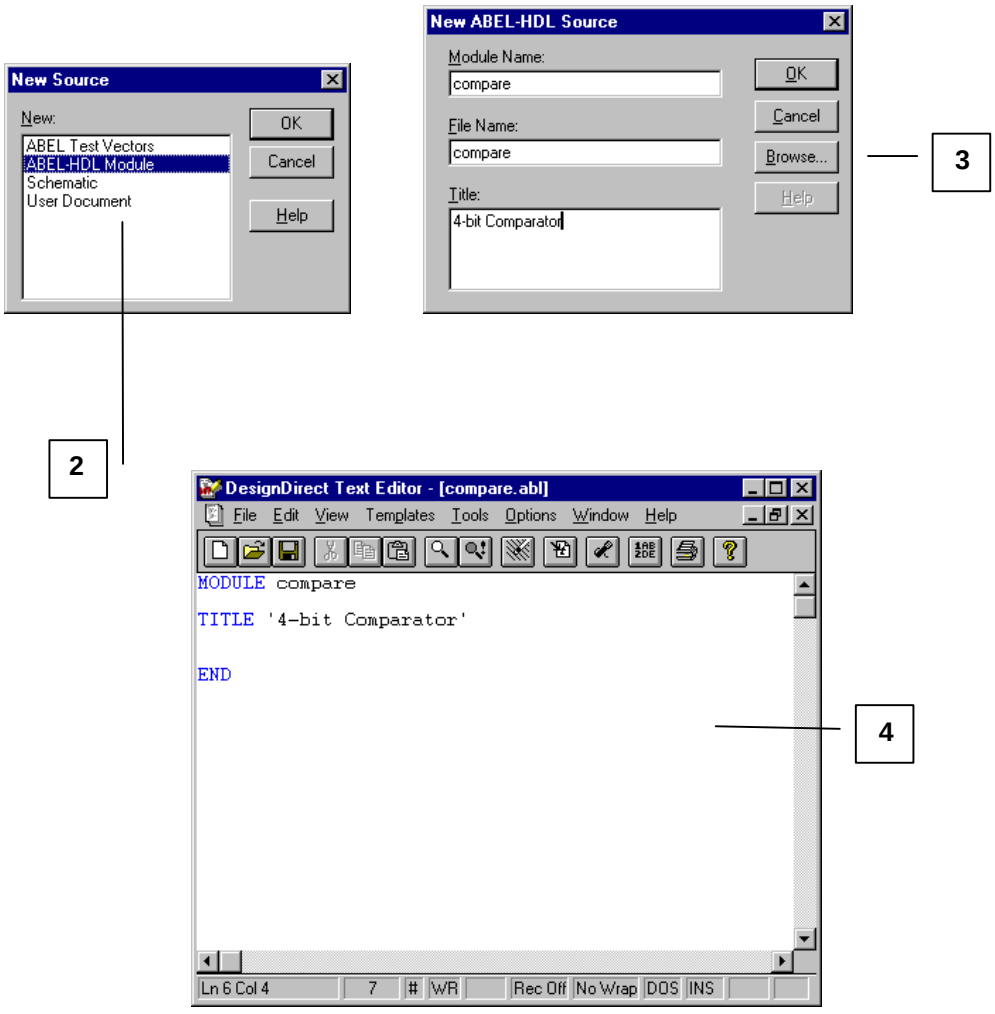
There are two steps to create a new ABEL-HDL source file. First you have to create the template. Then you enter the source description. In this step, you will create a template for the compare ABEL-HDL module.

To create an ABEL-HDL template:

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

   The Text Editor opens, as well as the **New ABEL-HDL Source** dialog, prompting you for additional information about the module.
3. In the dialog, type the information shown at the right. Make sure to enter the text in all lower-case (the text is case sensitive). Note that the Module Name matches the corresponding block symbol name. It is not necessary that the module name and file name be the same but it makes things simpler.
4. Click on **OK** to close the dialog.

   The Synario Text Editor appears with an ABEL-HDL template.



Task 22: *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:

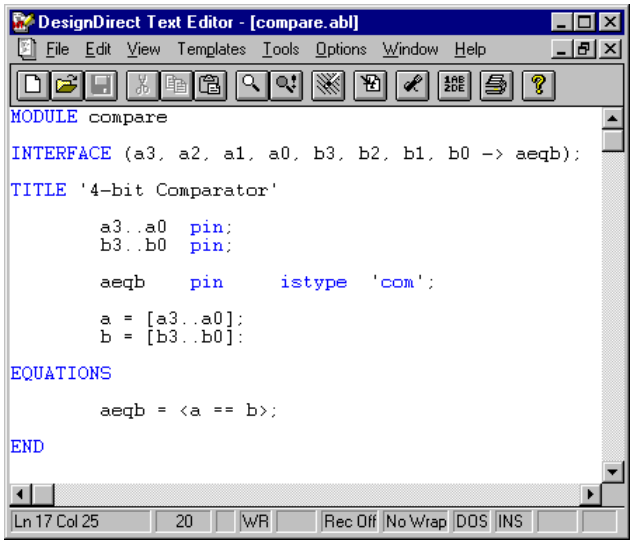
1. In the Text Editor, type the text description as shown in the picture on the right.

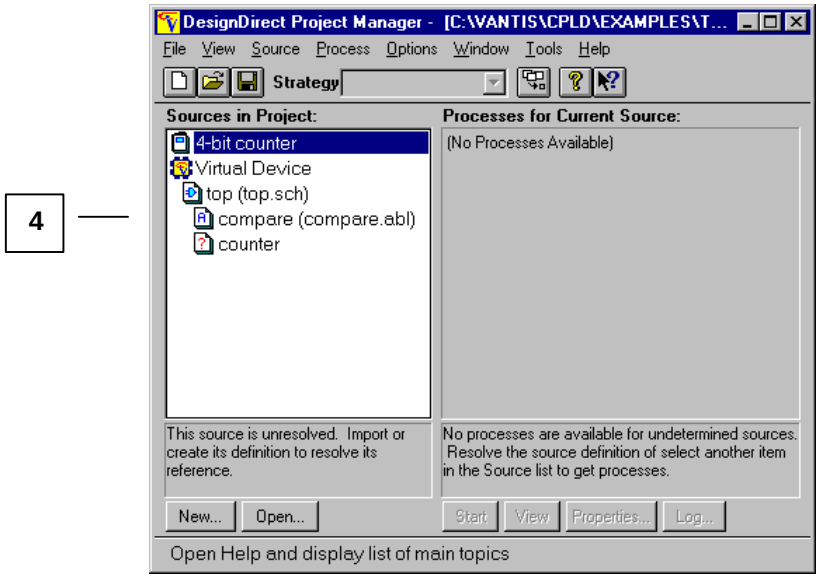
   Notice that the ABEL-HDL pin declarations must match the name and the case of the I/O pins on the corresponding block symbol.

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

3. Close the Text Editor.

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





Task 23: *Import an Existing ABEL-HDL Source File*

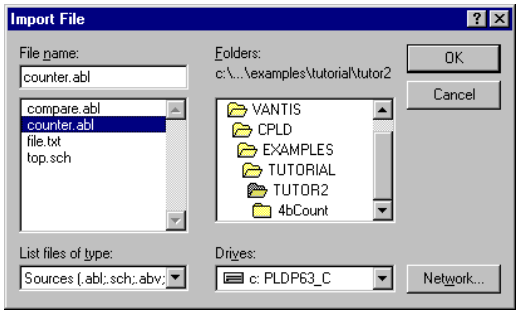
To save time in this tutorial, you will import the remaining ABEL-HDL source file. Although, in a practical sense, you will probably have existing ABEL-HDL files that you will want to import into a design in addition to creating new ones.

To import an existing ABEL-HDL source:

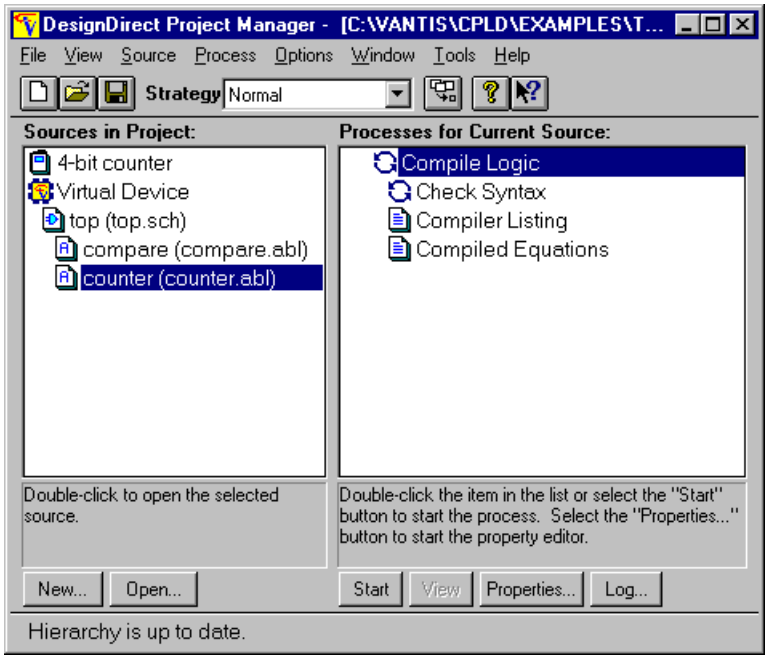
1. In the Project Manager, choose **Source** > **Import** to open the Import File dialog.
2. Go to the **\Vantis\CPLD\Examples\Tutorial\Tutor2** directory and select **counter.abl**.
3. Click **OK**. When prompted to replace the file, click **Yes**.
4. The Project Manager should now look like the image shown to the right.

   You'll take a look at the "insides" of this module in the following lesson

2



4



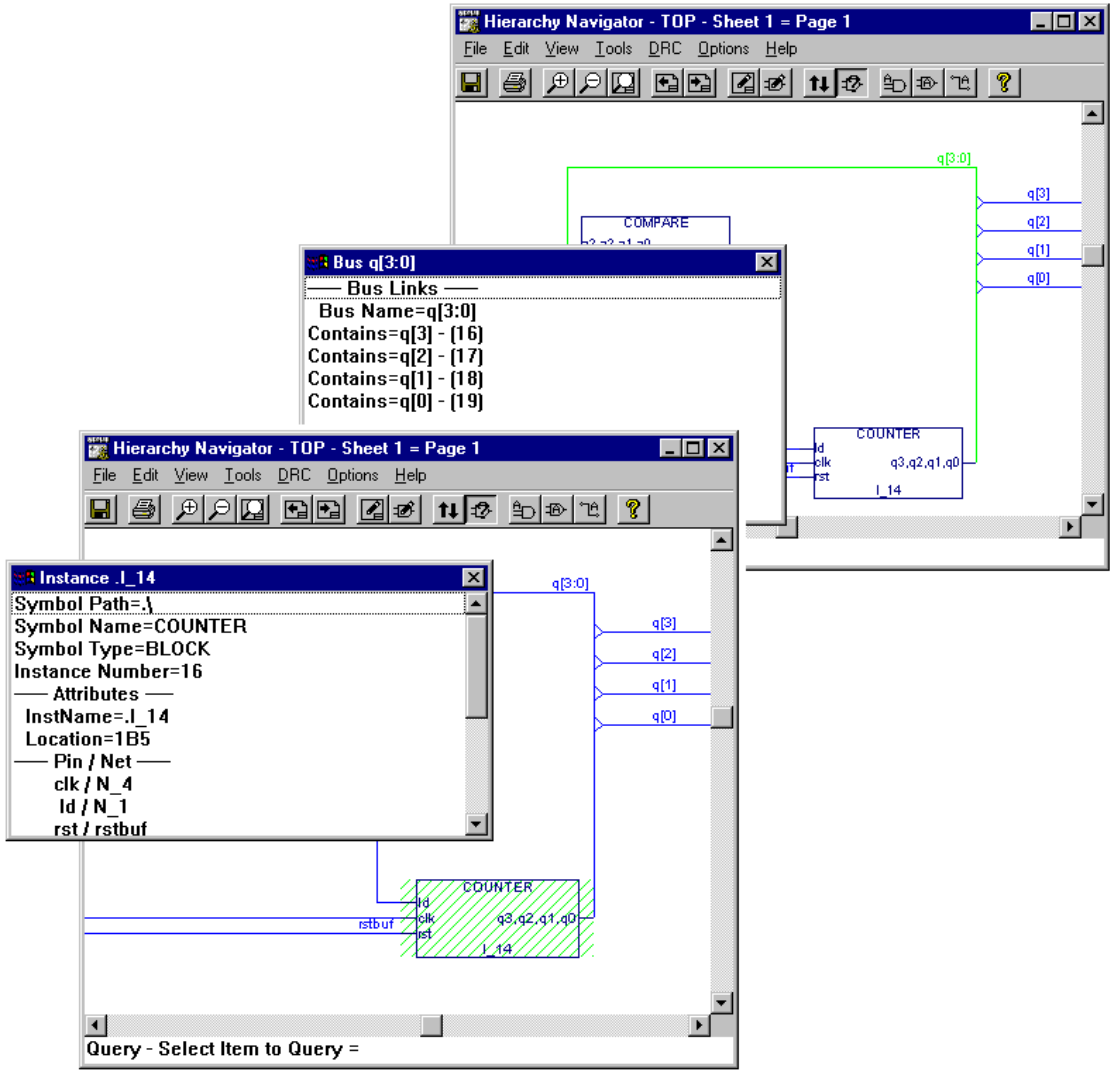
Lesson 5 – Navigating the Design

The Hierarchy Navigator program allows you to navigate through a schematic design consisting 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.

Lesson 5 shows you how to use the Hierarchy Navigator to perform several useful functions.

Steps covered in this lesson are:

- **Task 24**: Open the Hierarchy Navigator
- **Task 25**: Push into the Counter Block Symbol
- **Task 26**: Access Connectivity Information for the Counter Block
- **Task 27**: Query a Net



Task 24: *Open the Hierarchy 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.

You open the Hierarchy Navigator from the Project Manager.

To open the Hierarchy Navigator:

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

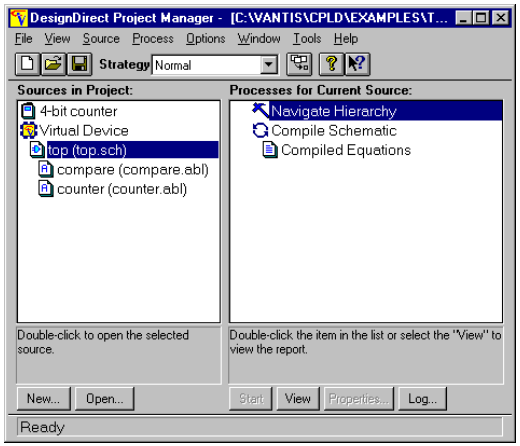
Notice the Navigate Hierarchy icon at the top of the Processes window is automatically selected. Also notice that the Hierarchy Navigator icon is visible *only* when a schematic source is selected.

2 Double-click the Navigate Hierarchy icon in the Processes window.

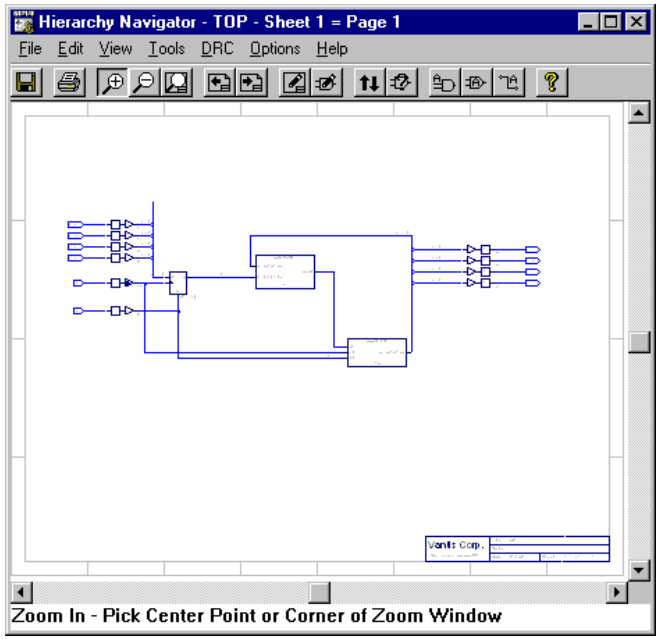
The Hierarchy Navigator opens with sheet 1 of the selected schematic source loaded.

***Note**: Remember, this is not the Schematic Editor. You can not 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.*

1



2



Task 25: *Push into the Counter Block Symbol*

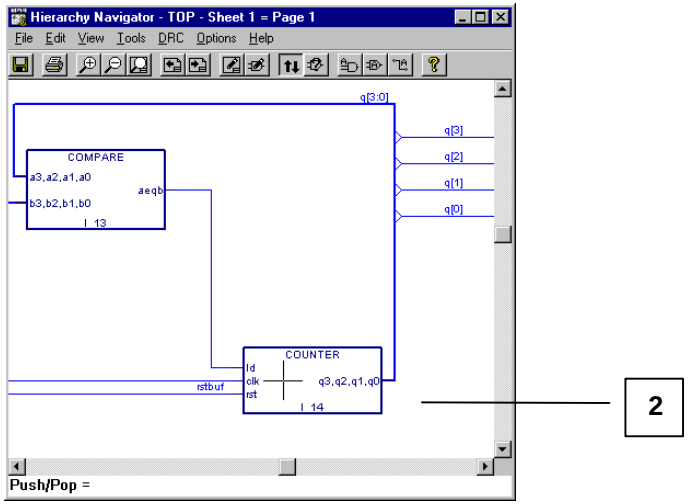
You can use the Push/Pop command on the View menu to move down and up (respectively) through the various hierarchical levels of a design. This command works on both schematics and ABEL-HDL modules.

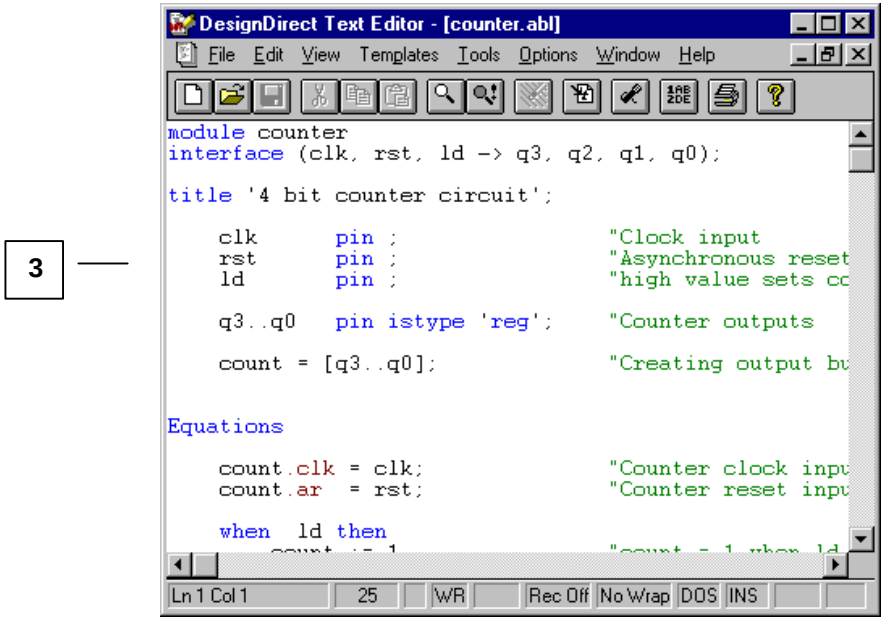
You may want to use the **Zoom In** command to view the **counter** module before starting this step.

To push into the Counter block symbol:

1. Choose **View** > **Push/Pop**. The cursor changes to a cross hair.
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.
3. View the contents of the **counter** module. When you are finished, close the **Text Editor**.





Task 26: *Access Connectivity Information for the Counter Block*

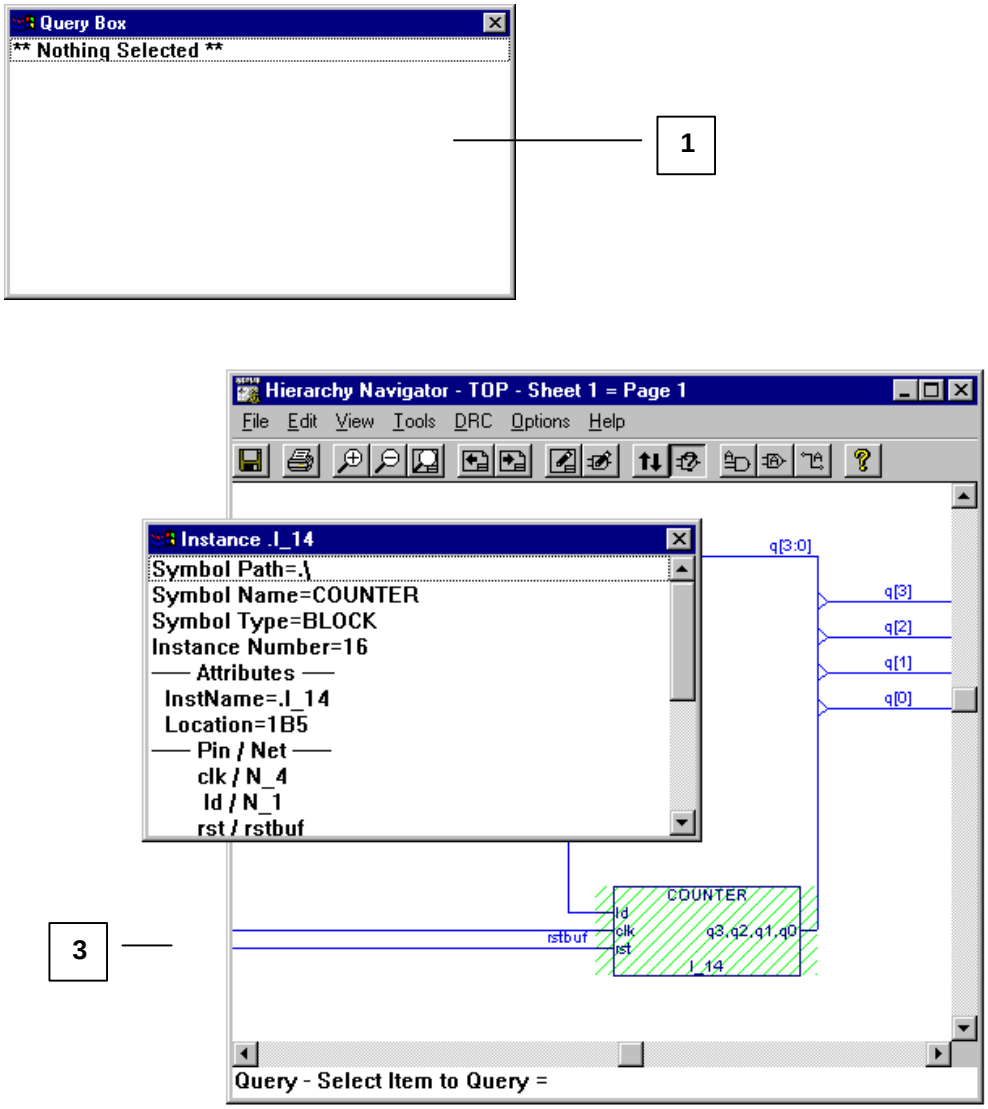
You can use the Query command to display additional information about circuit elements. The information appears in a text box that pops up when the first element is selected, and is updated when another element is selected.

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. Choose **DRC** > **Query**. The Query text box opens with the message: Nothing Selected.
2. Move the text box so that you can see the counter block.
3. Click the **counter** block. The Query text box updates to show various information about the counter block.
4. 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

.



Task 27: *Query a Net*

Although design problems are usually observed at the top level, the source of those 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, the Query command lets you quickly determine what components are attached to the net and what will be affected by changes to the net.

To query a net:

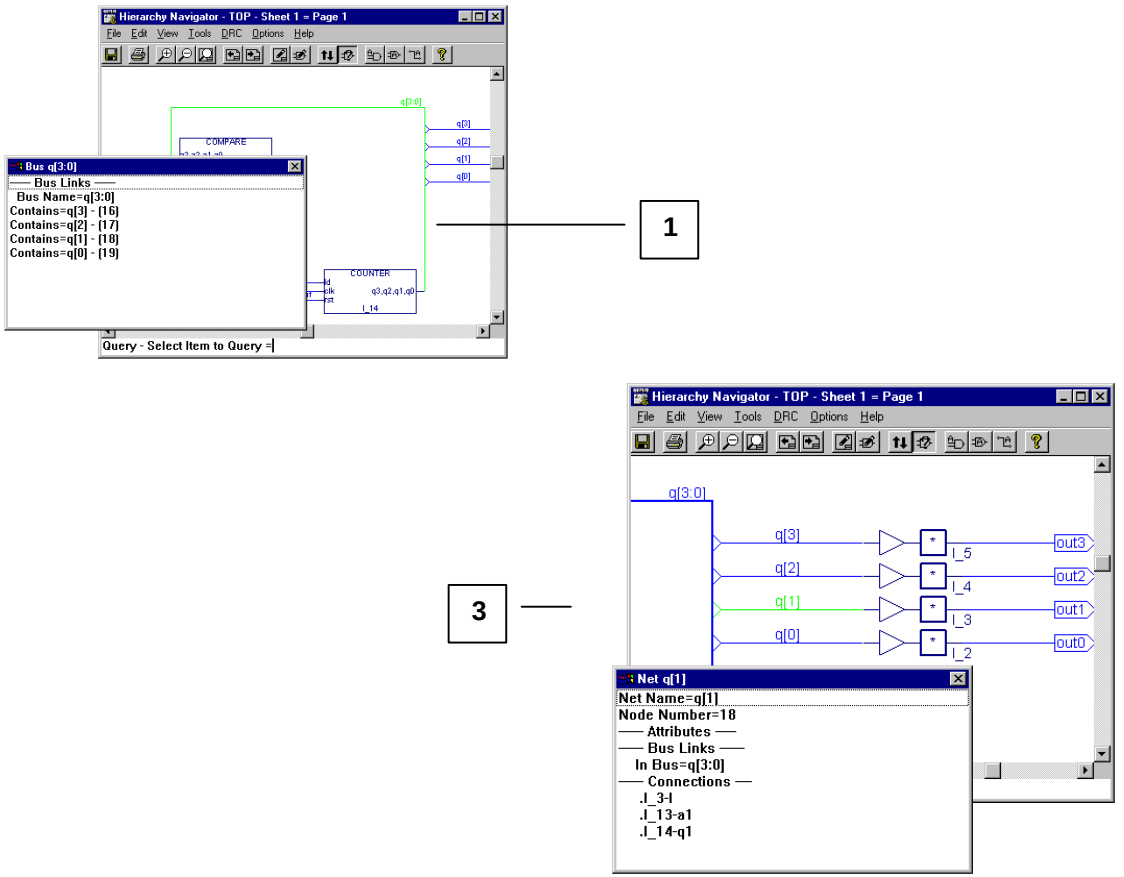
1 With the Query command still active, click the bus labeled **q[3:0]** in the schematic. A list of its content signals appears in the Query text box.

2 Observe that the signals displayed in the hot report window are links (pointers) to the individual nets.

3 In the Query Text box, click **q[1]**. The report updates showing 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.)

4 In the Query Text 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.

5 Close the Query Text box and the Hierarchy Navigator. If prompted, **Data has been modified, yes/no?** click **no**.



Congratulations

You have finished…

- 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-HLD source.
- import an ABEL-HLD source into the project.
- use the Hierarchy Navigator to navigate through the design and try “debug” methods.

Now you are ready to implement the design. You can continue working with this design in Tutorial 4, *Implementing a Design*

Tutorial 3

HDL Design Entry with LeonardoSpectrum

This tutorial shows you how to use LeonardoSpectrum to synthesize a VHDL design and generate an EDIF file for a MACH device. Then you will import the EDIF file into DesignDirect and implement it in a MACH4 device.

Prerequisites

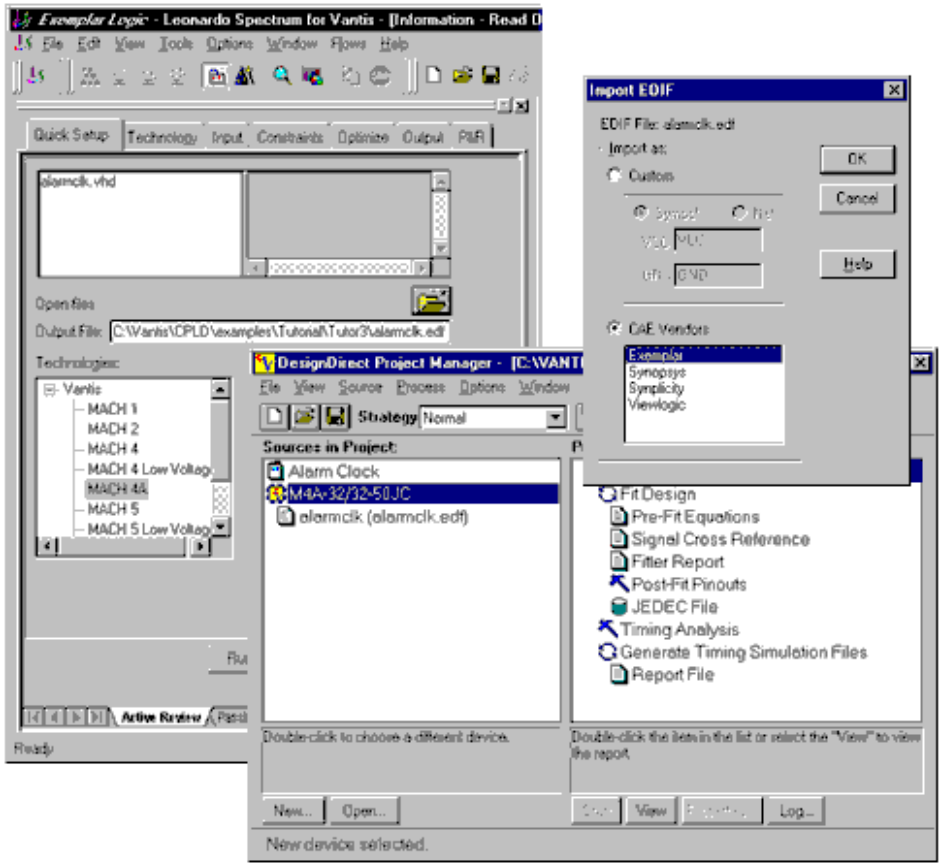
Before attempting this tutorial, you should complete the **DesignDirect Quick Start Tutorial** in the *DesignDirect-CPLD User Guide*, "Introduction" chapter.

Learning Objectives

When you are finished with this tutorial, you should understand the steps and tools necessary to generate an EDIF file from a third-party synthesis tool and import it into DesignDirect. You will also learn how to target a MACH device, fit the design, and generate a JEDEC file.

Time to Complete This Tutorial

The time to complete this tutorial is about 20 minutes.

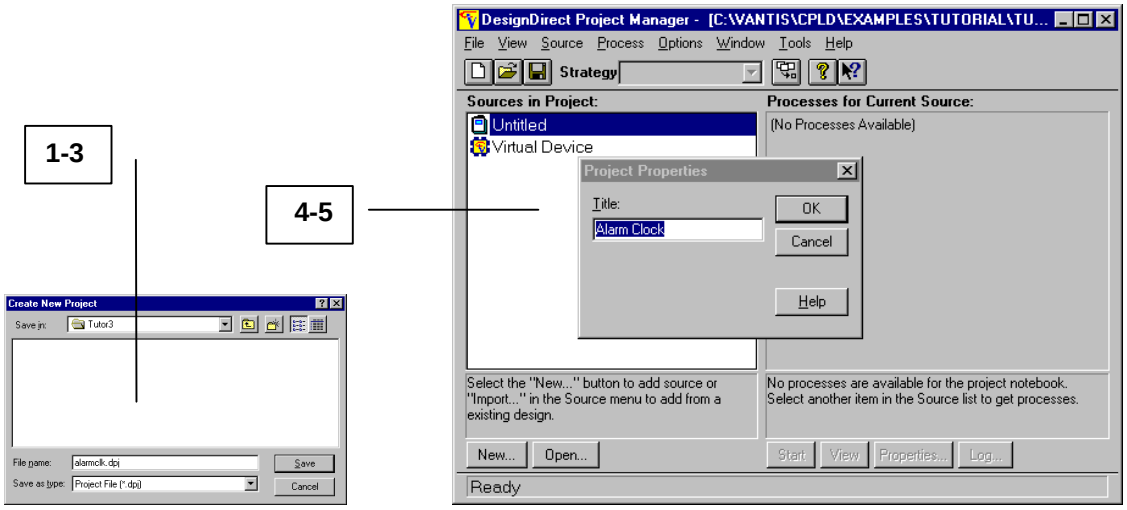


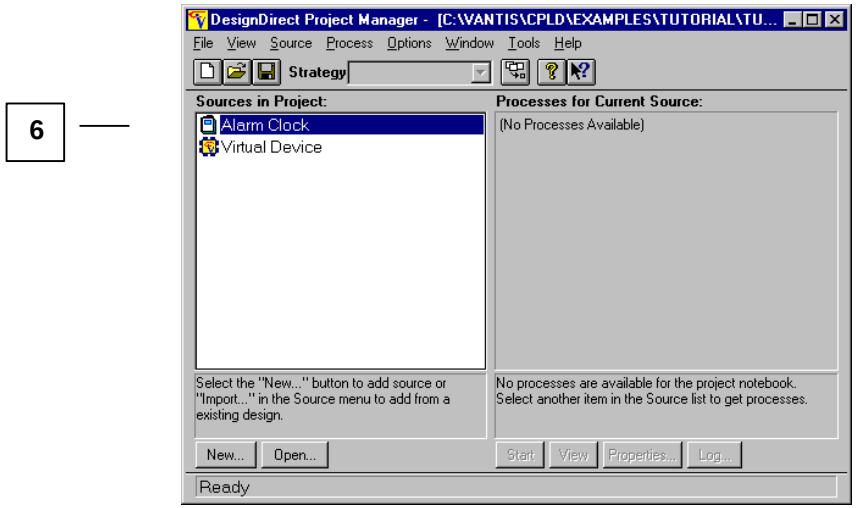
Task 1: *Create a new project in DesignDirect*

After starting the Project Manager, you are ready to create a new project.

To create a project:

1. Choose **File** > **New Project** to open the Create New Project dialog.
2. In the dialog, change directories to **Vantis\CPLD\Examples\Tutorial\Tutor3**.
3. Name the file **alarmclk.dpj** file and then click **Save**. The untitled generic project appears in the Sources window of the Project Manager.
4. In the Sources window, double-click the title of the project (Untitled) to open the Project Properties dialog.
5. Type the name **Alarm Clock** for the title of your project.
6. Click **OK**. The new project title appears in the Sources window.



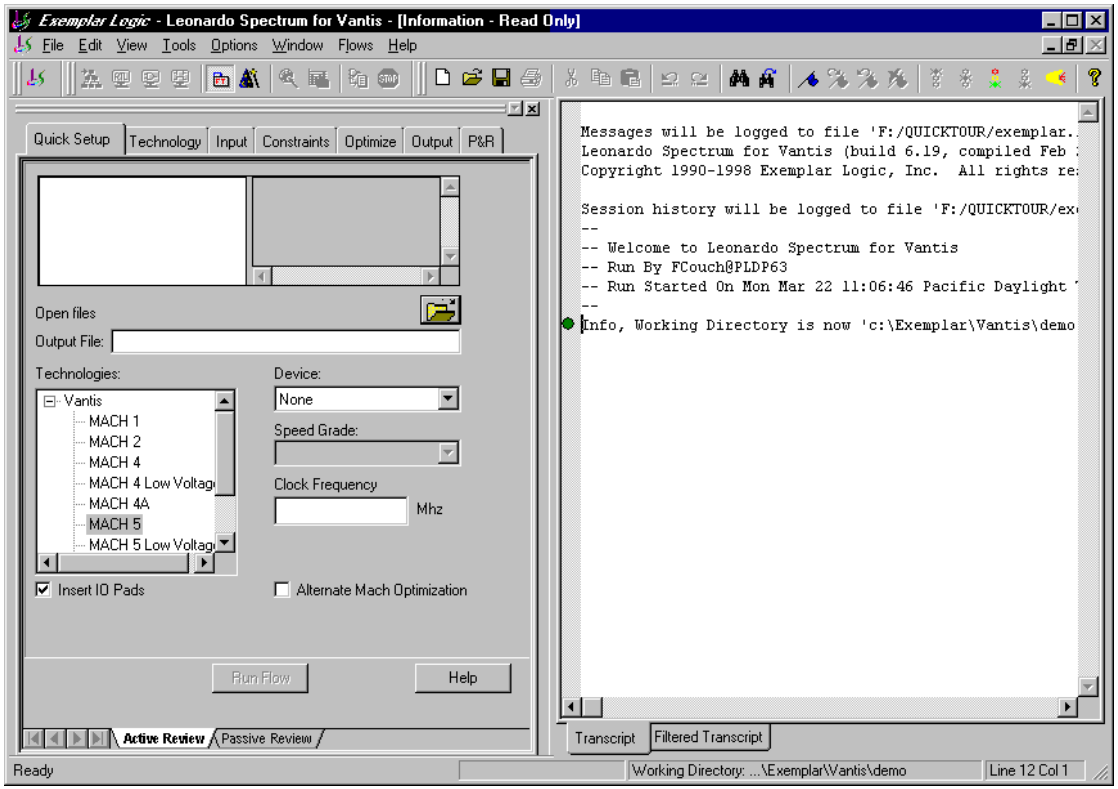


Task 2: *Start LeonardoSpectrum from DesignDirect*

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

To start LeonardoSpectrum:

1. On the DesignDirect menu, choose **Tools** > **LeonardoSpectrum Synthesis** to open LeonardoSpectrum synthesis tool.
2. There are three ways to synthesize your design: Synthesis Wizard, Quick Setup, and FlowTabs. For this tutorial you will use the Quick Setup method.
   - Click **OK** to close the Tip of the Day.
   - Click **Cancel** to close the Device Settings dialog.
3. Make sure the **Quick Setup** tab is selected. Your screen should look similar to the one on the right.



Task 3: *Use Quick Setup to Synthesis the Design*

Everything that can be specified in the synthesis wizard can be specified on the Quick Setup tab. Once specified, you can click Run Flow to run the entire synthesis flow, including synthesis, global constraints, optimization, and writing netlist. Additionally, Quick Setup automatically sets up all options, defaults, and settings in the FlowTabs to assist you when walking through the more advanced tabs.

To set up the synthesis run:

1 Click the **Open File** icon and select the VHDL input file.

Go to the ..\examples\Tutorial\Tutor3 directory and select **alarmclk.vhd**. Notice that the software automatically points the output file to the project directory.

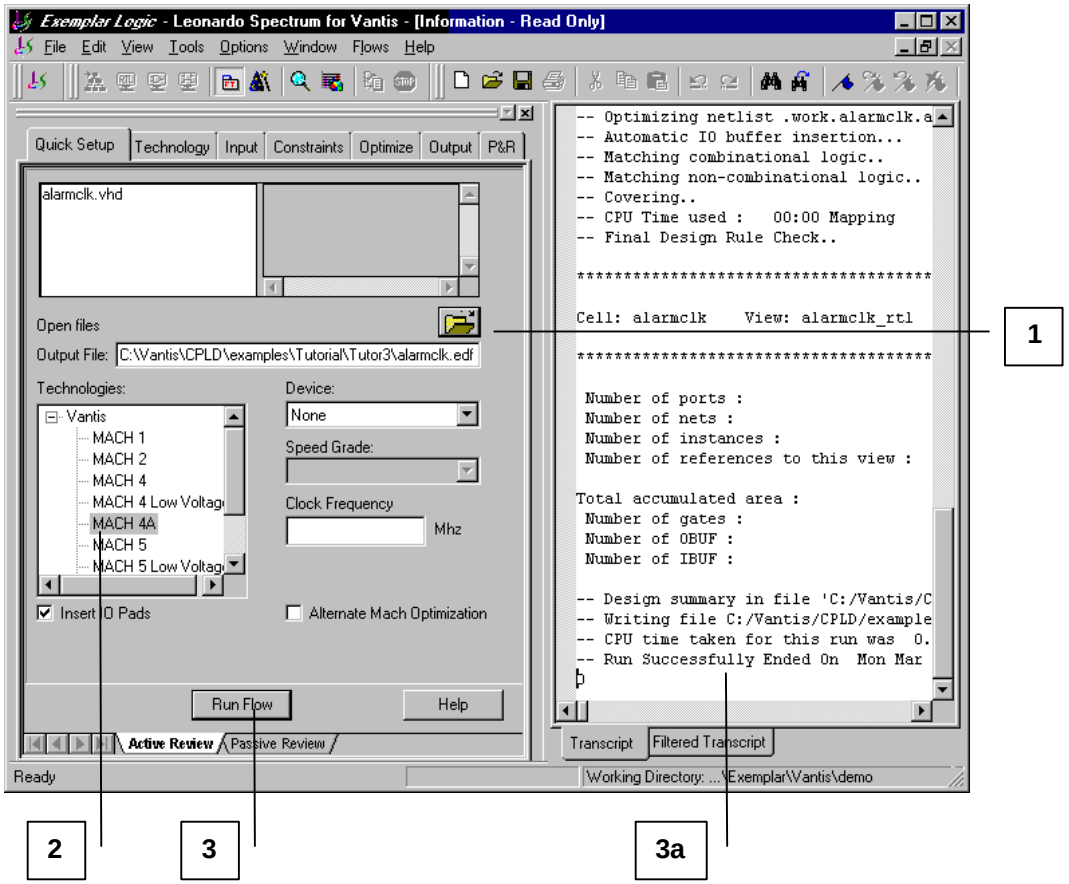
2 Under Technologies, select the **MACH 4A** device family. You do not need to select a specific device; you will do this in DesignDirect.

3 Click **Run Flow**. LeonardoSpectrum runs the entire flow using all options set on all dialogs and tabs, including options which are not shown on this tab, and creates an EDIF file.

Note: The Run Flow button is not active until you have selected your Input File(s) and target technology.

Notice in the Information window on the right (**3a**) that it says the run successfully ended when the synthesis process is complete.

Note: Click the **Stop** icon on the synthesis toolbar to cancel the run before it completes.



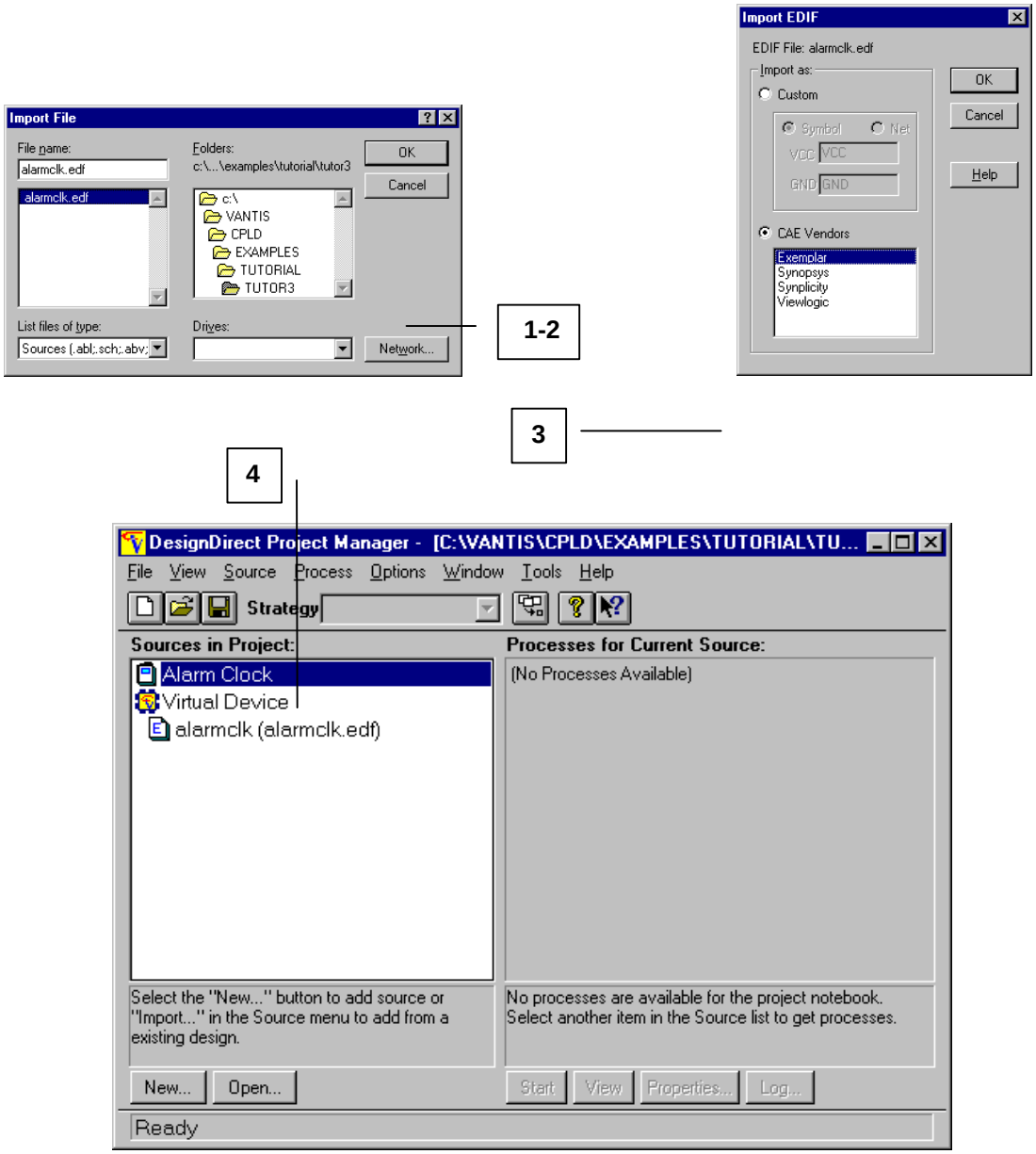
Task 4: *Import the EDIF file into DesignDirect*

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

To import an EDIF netlist into your project:

1. In the Project Manager, choose **Source** > **Import** to open the Import File dialog.
2. Select **alarmclk.edf**.
3. Click **OK** to open the Import EDIF dialog.
4. Accept the default settings (CAE Vendors, Exemplar) and click **OK**. The software adds the selected EDIF file (alarmclk.edf) to the project sources.

***Note**: After you import an EDIF file into the DesignDirect project, it is always linked in the Project Manager. Therefore, if you make changes and recompile your VHDL file to create a new EDIF file, you just have to choose **View** > **Update** in the Project Manager to update the process.*



Task 5: *Target a Device and Fit the Design*

The Project Manager lets you target a design to a specific Vantis device at any time during the design process.

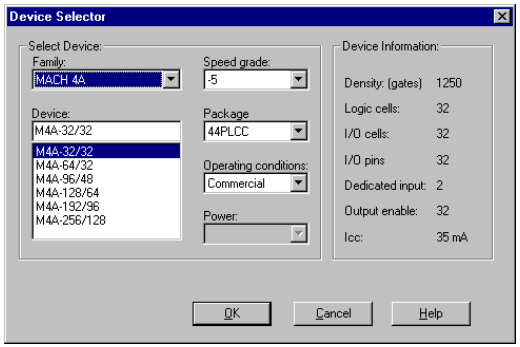
To target a device and run the Fitter:

1. In the Sources window, double-click **Virtual Device** to open the Device Selector dialog. The dialog shows the available device families and the devices in the selected family.
2. In the Device Selector dialog under Family, select the **MACH 4A** device family.
3. Select the **M4A-32/32** device.
4. Accept the default settings and then click **OK**.
5. Click **Yes** to confirm that you wish to change device kits.

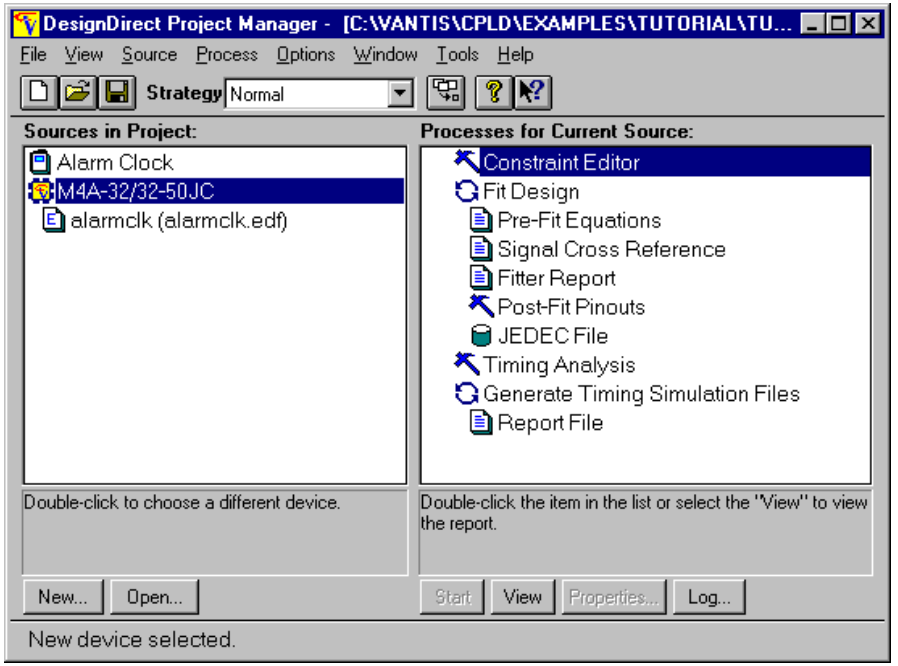
   Note: When you change device kits, the DesignDirect design environment reconfigures to facilitate designing with the selected device kit.
6. The DesignDirect softeware shows the selected device in the Sources window.
7. With the target device selected in the Sources window, double-click **Fit Design** in the Processes window to run the Fitter. DesignDirect fits the design in the specified device and generates a JEDEC file.

If you want to see how to simulate this design using Model*Sim*, see Tutorial 5.

1-4



6



Task 6: *Congratulations!*

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

- create a new project in DesignDirect.
- start LeonardoSpectrum from DesignDirect.
- use the Quick Setup tab in LeorardoSprectum to set up and run synthesis, creating an EDIF files.
- import the EDIF file into DesignDirect.
- target a device for the design.
- run the Fitter and automatically generate a JEDEC file.

Tutorial 4

Design Implementation and Verification

This tutorial provides an overview of the features and operation of DesignDirect, focusing on the tasks and tools needed to implement and verify a programmable device design.

Prerequisites

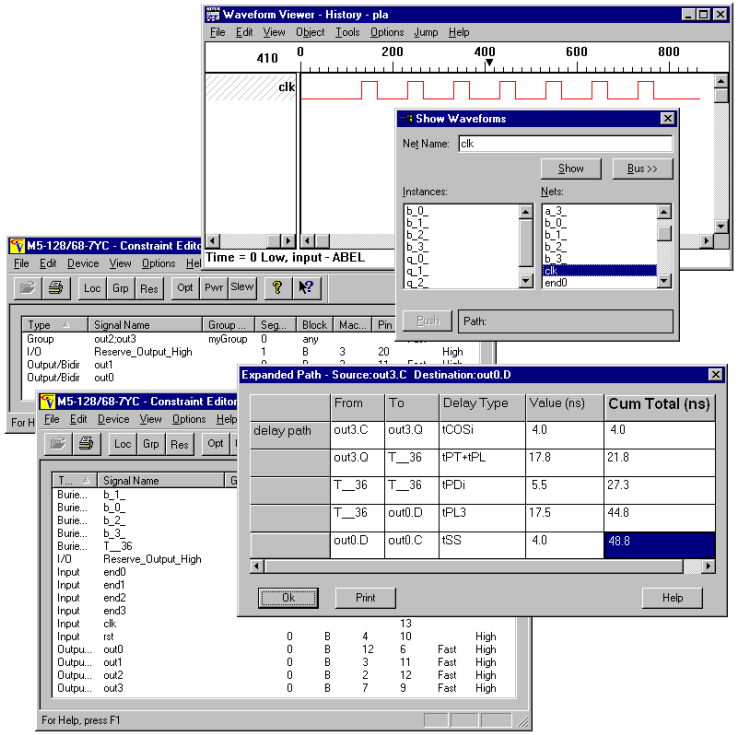
Before attempting this tutorial, you should complete either the **Schematic and ABEL-HDL Design Entry** tutorial.

Learning Objectives

When you are finished with this tutorial, you should have a basic understanding of the steps and tools necessary to implement and verify a Vantis programmable IC design using DesignDirect.

Time to Complete This Tutorial

On average, the time to complete this tutorial is about 30 minutes.



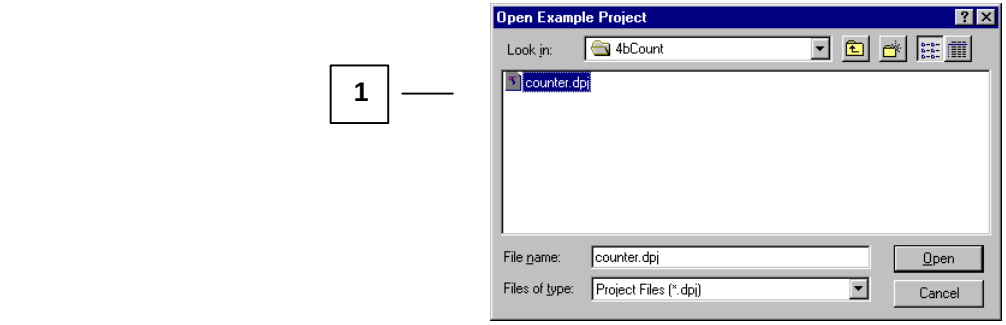
Task 1: *Open the Project*

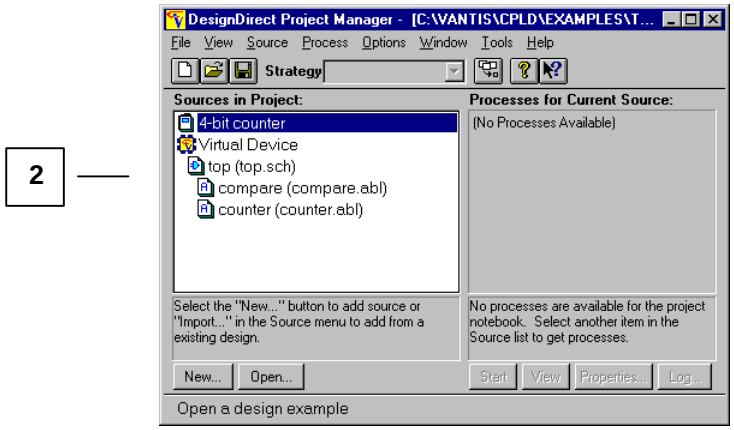
To continue with the design example, open the project for this tutorial.

To open the tutorial project:

1. In the Project Manager, choose **File** > **Open Example** to open the dialog.
2. Open the file **\Tutorial\Tutor4\4bCount\counter.dpj**.

   The 4-bit counter project opens in the Project Manager.





Task 2: *Select a Device*

The Project Manager lets you target a design to a specific Vantis device at any time during the design process.

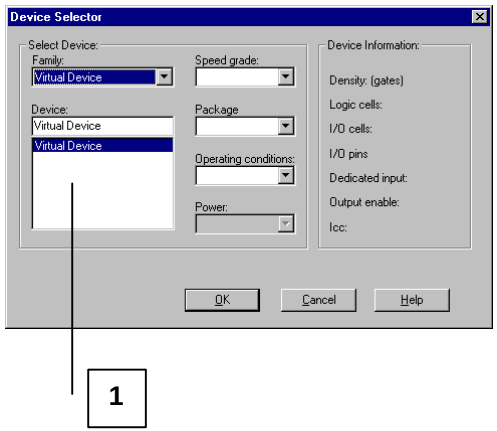
If you do not know the specific device, you can target a *Virtual Device*. A Virtual Device is a generic, architecture-independent device, allowing you to create a design that does not rely on specific device features to implement.

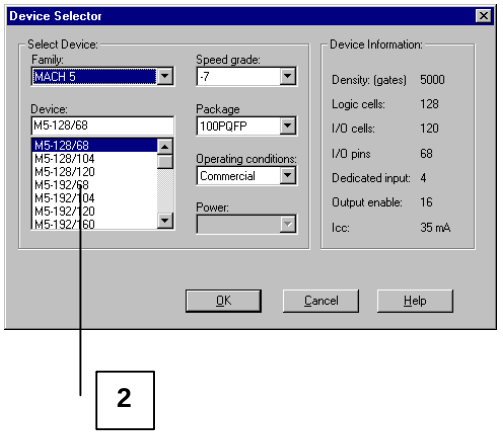
In this task you will select a device for your design.

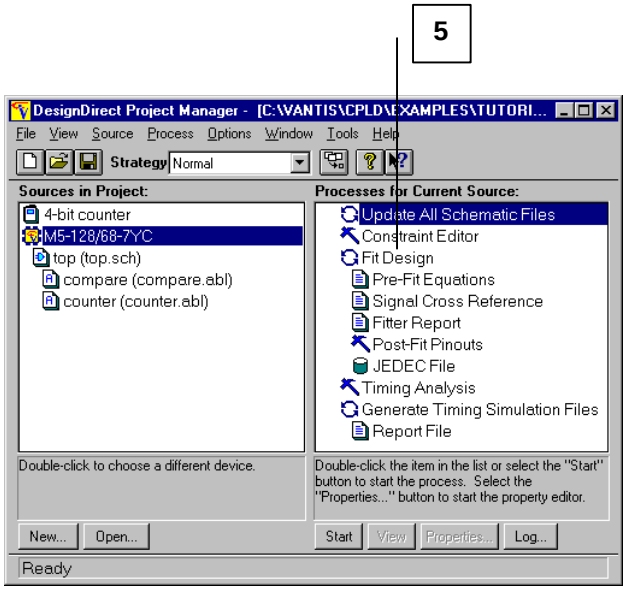
To select a device:

1. In the Sources window, double-click **Virtual Device** to open the Device Selector dialog. The dialog show the available device families and the devices in the selected family. (The device list can vary depending upon which device kit you have installed. Check with your local representative for a complete list.)
2. In the Device Selector dialog, select the **MACH 5** device family.
3. Select the **M5-128/68** device.
4. Accept the default settings and then click **OK**.
5. Click **Yes** to confirm that you wish to change device kits.

   When you change device families, the DesignDirect design environment reconfigures to facilitate designing with the selected device kit.





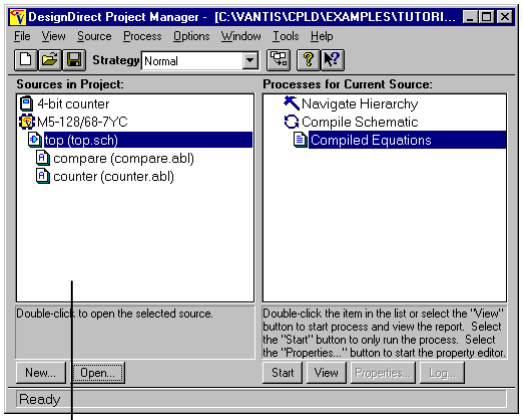


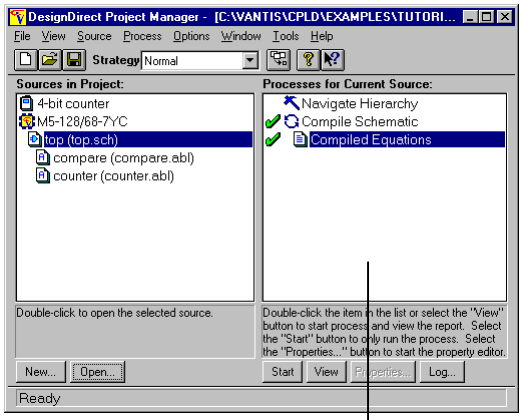
Task 3: *Compile a Source File*

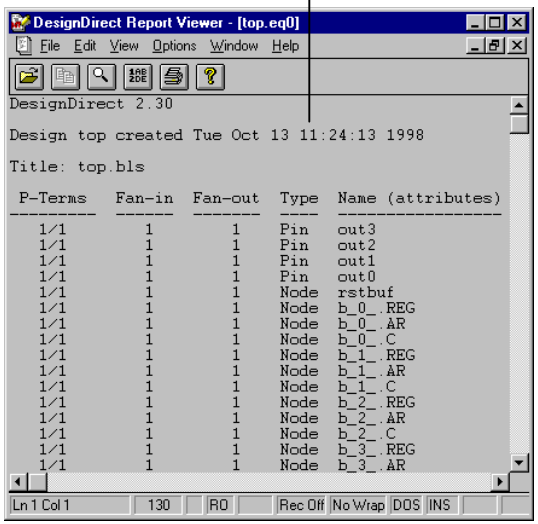
When you compile a design, you are changing your design entry format into Boolean equations, which serve as input to simulation and device implementation programs. The Project Manager processes each ABEL-HDL module or schematic file to obtain an intermediate file that can later be linked together before fitting the design into a Vantis device.

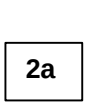
To compile a source and automatically view the report:

1. In the Sources window, select the **top.sch** source.
2. In the Processes window, double-click **Compiled Equations**. The DesignDirect software compiles the source file and automatically displays the resulting file in the Report Viewer (**2a**). If an ABEL-HDL file contains syntax errors, the software displays the errors in a view window and an error indication appears in the Processes window.
3. Close the Report Viewer.









Task 4: *Simulate the Equations and View the Results*

Equation simulation uses design test vectors, that you supply, to simulate the design logic or equations independent of any device.

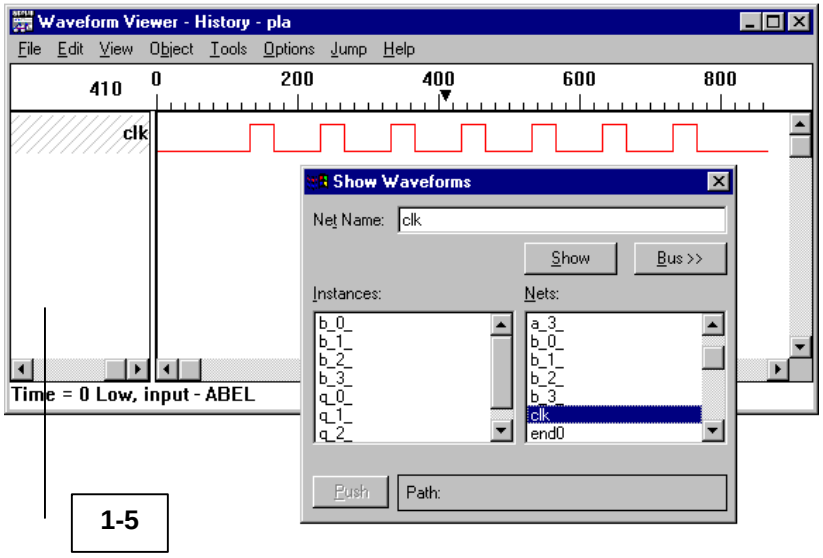
There are two ways to specify test vectors. The most common method is to put test vectors into the ABEL-HDL source file. The other way to specify test vectors is to create a "real" test vector file (*filename.abv*).

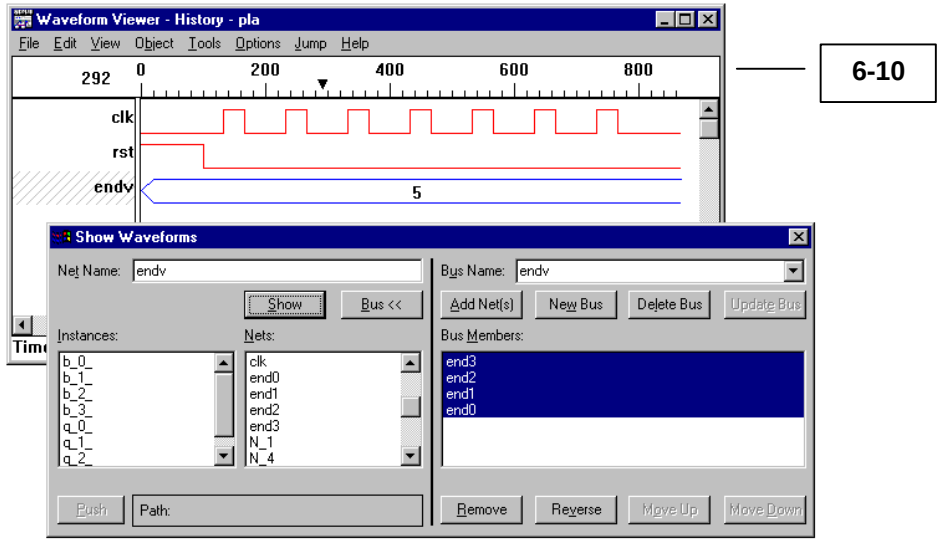
The advantages to placing test vectors in the ABV file instead of in the ABEL-HDL source is an improvement in processing time. By placing test vectors in the ABV file you will be able to change the test vectors and re-simulate without having to re-compile the logic. This can make a significant difference in large hierarchical designs.

In this next step, you will import an ABV source file into the project. Then you will run equation simulation and view the results using the Waveform Viewer.

To simulate equations and view the results:

1. Choose **Source** > **Import**, select **top.abv**, and then click **OK** to load the file into the project.
2. In the Sources window, select **top.abv** and then double-click **Equation Simulation Waveform** in the Processes window. The software runs equation simulation and opens the Waveform Viewer.
3. Choose **Edit** > **Show** to open the Show Waveforms dialog.
4. Under Nets, scroll down and select **clk**. Then click **Show**. The software adds the waveform to the viewer.
5. Scroll down more and select **rst**. Then click **Show**.
6. Now you'll create a bus. Click **Bus** to expand the dialog.
7. In the Bus Name field, type **endv**.
8. Under Nets, select **end0** to **end3**. Then click **Add Nets**.
9. Click **Reverse** to reverse the order.
10. Click **Save Bus**. Then click **Show** to add the bus to the viewer.
11. Exit the Waveform Viewer. When prompted to save the file, click **No**.





Task 5: *Assign Some Constraints*

The Constraint Editor lets you specify pin and node assignments, group assignments, pin reservations, power level settings, output slew-rates and JEDEC file options. The 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

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 In this step, you will specify two pin assignments. On the toobar, click the **Loc** button to open the Location Assignment dialog.

3 Under Filter, select **Output/Bid**. In the Signals List, select **out0**. Under Asignment select pin **6**. Then click **Add**. The software adds the pin assignment to the assignment list.

4 Now, using the same procedure, assign **out1** to pin **11**, and then add it to the assignment list. Click **OK** to close the dialog.

5 Next, you'll create a group and assign it to any block . On the toolbar, click the **Grp** button to open the Group Assignment dialog.

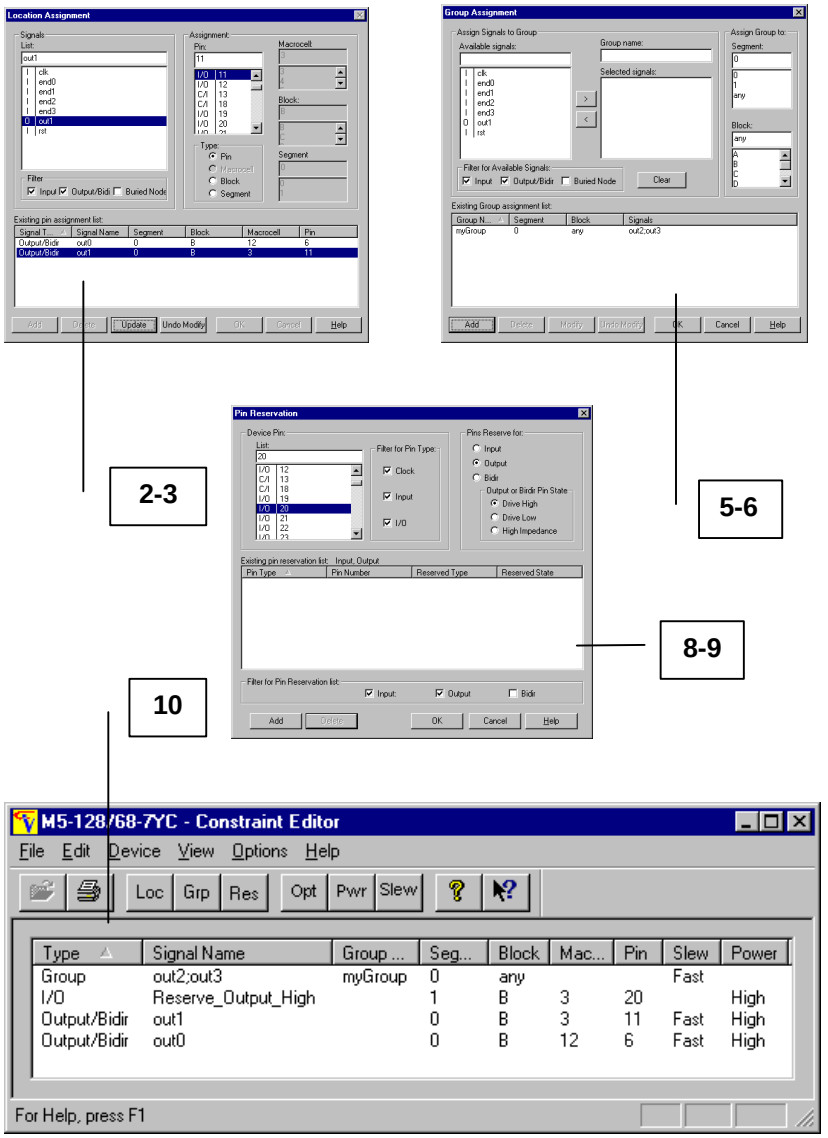
6 Under Filter, select **Output/Bid**. In the Available Signals list, select **out2** and **out3**. Then click the **right-arrow** button to add them to the Selected Signals list.

7 In the Group Name field, type **myGroup**. Under Assign Group To, select block **Any**, if it is not already selected. Then click **Add** to add the group to the assignment list. Click **OK** to close the dialog.

8 Finally, you will specify Pin Reservation constraints, which are "soft constraints." That is, if the design fails to fit, the pin reservation constraints are ignored by the Fitter. On the toolbar, click the **Res** button to open the Pin Reservation dialog.

9 Select pin **20**, select **Output**, select **Drive High**, and then click **Add**. Click **OK** to close the dialog.

10 Look at the assignments in the Constraint Editor, and then exit the editor.



Task 6: *Set Fitter Optimization Options*

You set the optimization options for the Fitter using the Global Optimization dialog. The default Fitter options are set up to achieve the highest possible performance in the smallest possible device, for most designs. You can choose to maximize design flexibility by spreading out logic or exercise tighter control over the fitting process by packing the logic to achieve your design goals.

To set the Fitter optimization options:

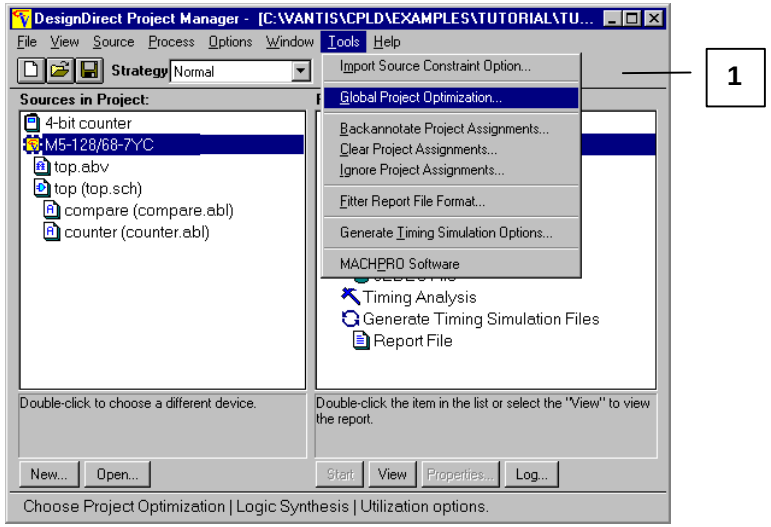
1. In the Project Manager, choose **Tools** > **Global Project Optimization** to open the dialog.

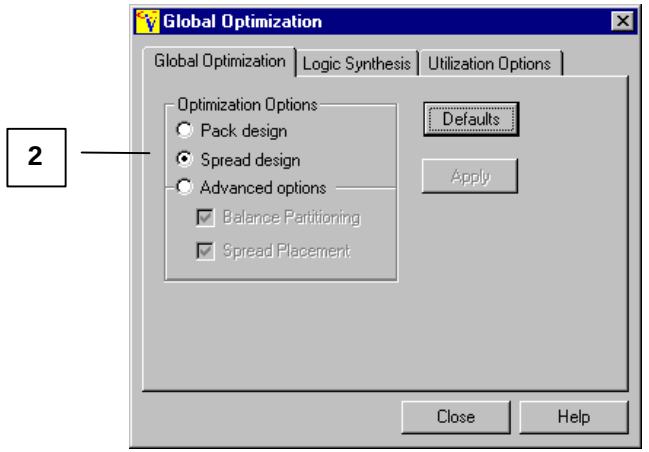
2. On the Global Optimization tab, select **Spread Design**.

   The Spread Design option spreads the design throughout the device, allowing the design to be partitioned and placed in a spread out method. This placement method enhances the upgrade capability of the design by spreading the resources throughout the device. However, the disadvantage is that more resources may be used because the design is partitioned evenly through all the blocks.

   Later, you will choose Pack design and see the difference in performance results.

3. Click **Apply**, and then click **Close** to close the dialog.





Task 7: *Fit the Design*

DesignDirect 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, DesignDirect generates a JEDEC file, as well as a fitter report, so that you can see how DesignDirect has utilized and routed the part.

There are four phases to the fitting process. These are:

Initialization - When you start a new project, DesignDirect automatically copies a default constraint file from the DesignDirect directory into your project directory.

Optimization - Each clock signal is evaluated and classified as a global clock or a non-global clock. The Fitter attempts to place all global clock signals at global clock pins (check the log file for the status of all clock signals after optimization). The Fitter assigns all other clock signals to I/O pins and implements them as Product Term clocks, if the architecture supports Product Term clocks. Input pins and nodes that are defined but not referenced (not used by another equation) are discarded from the design during optimization (warning messages are generated).

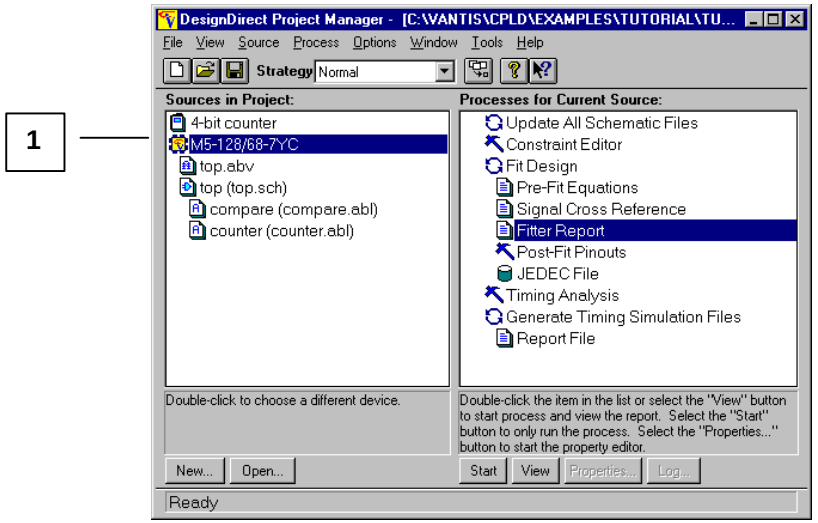
Partitioning - After optimization, DesignDirect partitions the design into individual blocks of the specified MACH device. Partitioning is achieved by assigning logic to specific PAL blocks, based on several considerations.

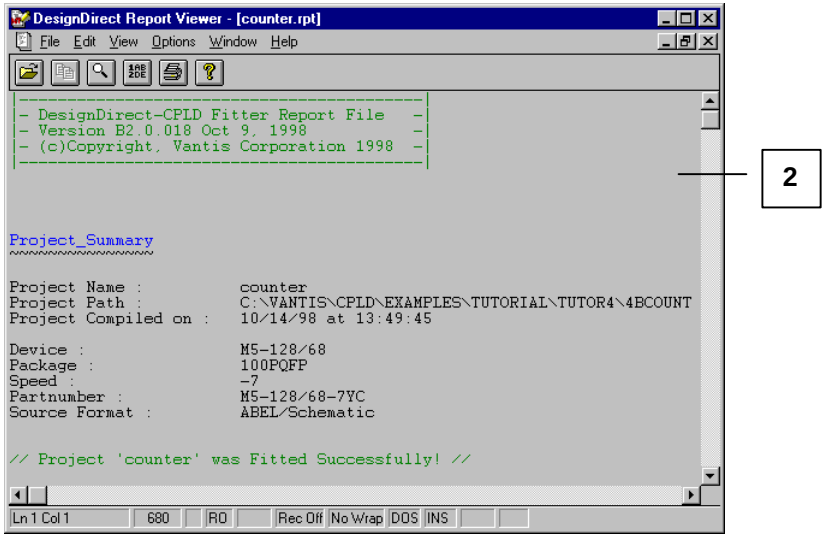
The Partitioner considers commonality of signals, macrocell requirements, Set/Reset requirements, product-term requirements, and other factors to determine which partition is most likely to succeed in fitting the design. Only partitions that are likely to succeed (according to the Partitioner's rules) are attempted.

Placement and Routing - In the placement phase of the fitting process, individual equations are assigned to physical resources. In the routing phase, the Fitter attempts to route input, output, and feedback signals to and from the physical resources assigned in the placement phase.

To run the Fitter and view the Fitter report:

1. In the Sources window, select the target device.
2. In the Processes window, double-click **Fitter Report**. This starts the Fitter and automatically opens the Fitter report in the Report Viewer.
3. Browse the Fitter report and then close the Report Viewer.





Task 8: *Analyze the Timing Results*

The Performance Analyst analyzes the performance of your design after it had been optimized and implemented by the Fitter.

The Timing Analyzer traces each logical path in the design and calculates the path delays using the timing model and worst case values supplied in the device data sheet.

The Performance Analyst performs six distinct analysis types: fMAX, tSU, tPD, tCO, tOE and tCOE. Timing filters, source and destination filters and path filters can be used to independently fine-tune each analysis.

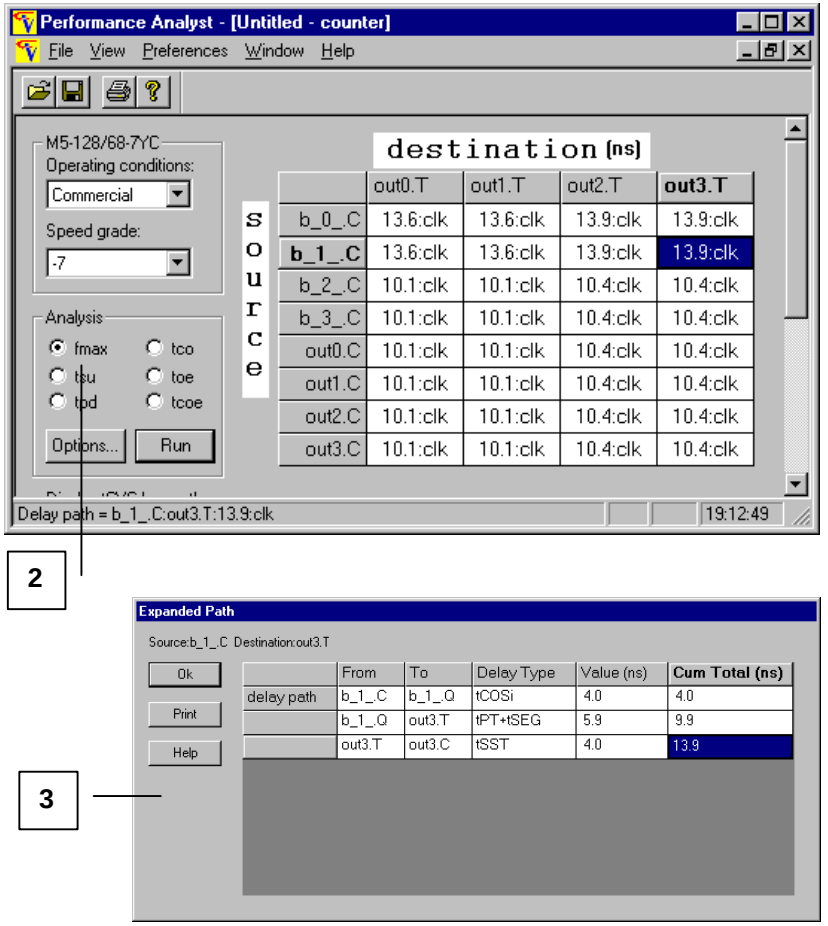
The timing analysis results are displayed in a graphical spreadsheet with sources displayed on one axis and destinations displayed on the other. The worst case delay value is displayed in the spreadsheet cell if there is a path between the source and destination. Path details are obtained by placing the cursor on the cell and double-clicking the left-hand mouse button, allowing the user to easily identify performance bottlenecks.

To run the Performance Analyst and view the results:

1. In the Sources window, select the target device. Then in the Processes window, double-click **Timing Analysis**.
2. Select **fmax** and then click **Run**. The overall timing results appear in the spreadsheet.

   ***Note**: Your numbers may differ from those shown in this example.*

3. Double-click **inside the highlighted cell** to open the Extended Path dialog and analyze the timing results of an individual path.
4. Notice the timing results. The longest delay was 13.9 ns.
5. (*Optional*) **Print** both output reports. You will use this in the next task to comapre timing results.
6. Close both spreadsheet dialogs without saving.



Task 9: *Change the Fitter Optimization Options and Re-analyze the Timing*

In a previous task you learned how to set Fitter optimization options using the Global Optimization dialog. If you recall, the default Fitter option is Spread Design. This option allows you to achieve the highest possible performance, while leaving room for any additional functionality that you may want to add in the future.

Now you will choose another option, Pack Design. This option lets you pack as much logic into the device as possible. Pack Design allows you to achieve the highest possible performance in the smallest possible device, for most designs. Each block may be completely filled, leaving less room for any design changes or logic additions.

After you change the Fitter optimization option, you will re-run timing analysis and see the performance differences with the new setting.

To change the Fitter optimization options and re-run timing analysis:

1 In the Project Manager, choose **Tools** > **Global Project Optimization** to open the dialog.

2 On the Global Optimization tab, select **Pack Design** and then click **Apply**. Click **Close** to close the dialog.

***Note**: You must click **Apply** to register the changes in the dialog.*

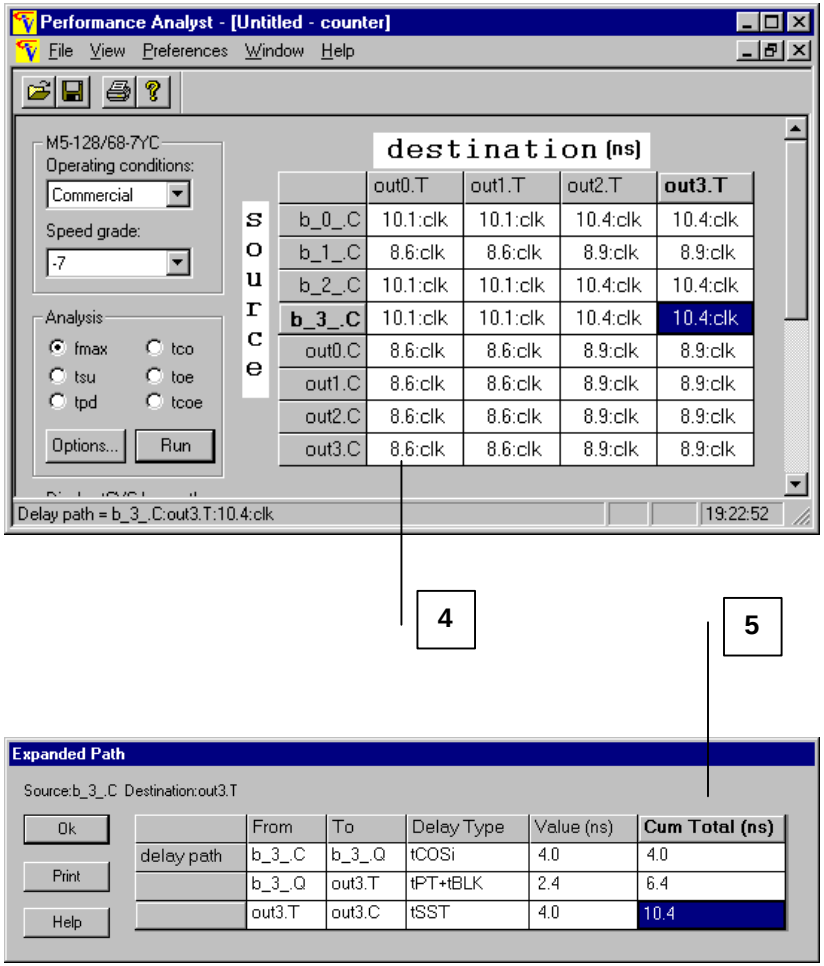
3 In the Sources window, select the target device. Then in the Processes window, double-click **Timing Analysis** to open the Peformance Analyst. Notice this runs the Fitter again.

4 Select **fmax** and click Run. The overall timing results appear in the spreadsheet. Notice the the performance has improved. Now the longest dealy is 10.4 ns.

***Note**: Your numbers may differ from those in this example. What is important to notice is the "relative" improvement in performance compared to the previous run.*

5 Double-click **inside the highlighted cell** to open the Extended Path dialog and analyze the timing results of an individual path. Again, notice that the cumulative total has improved.

6 Close both spreadsheet dialogs without saving.



Task 10: *Backannotate Project Assignments*

You can backannotate assignments from the Fitter output to the project constraint file using the Backannotation tab on the Constraints Options dialog. This feature lets you retain the assignments made by the Fitter so they can be used in the future.

You can only backannotate project assignments after the "Fit Design" process has been successfully completed. An error message appears if DesignDirect detects that this process did not complete successfully.

To backannotate assignments:

1 In the Project Manager, choose **Tools** > **Backannotate Project Assignments** to open the Constraints Options dialog.

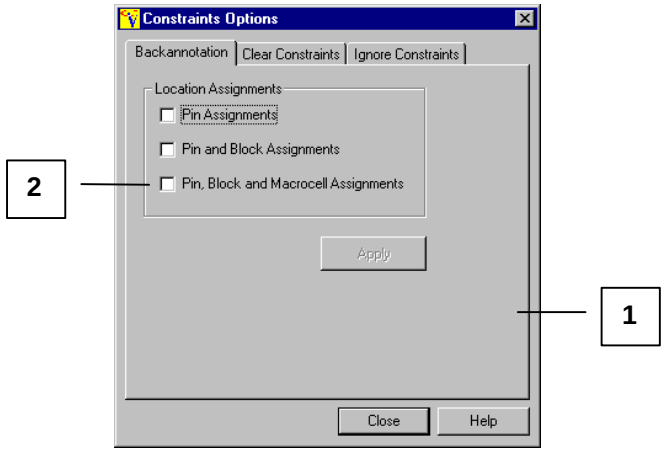
2 Select **Pin, Block, and Macrocell Assignments**, and then click **Apply**.

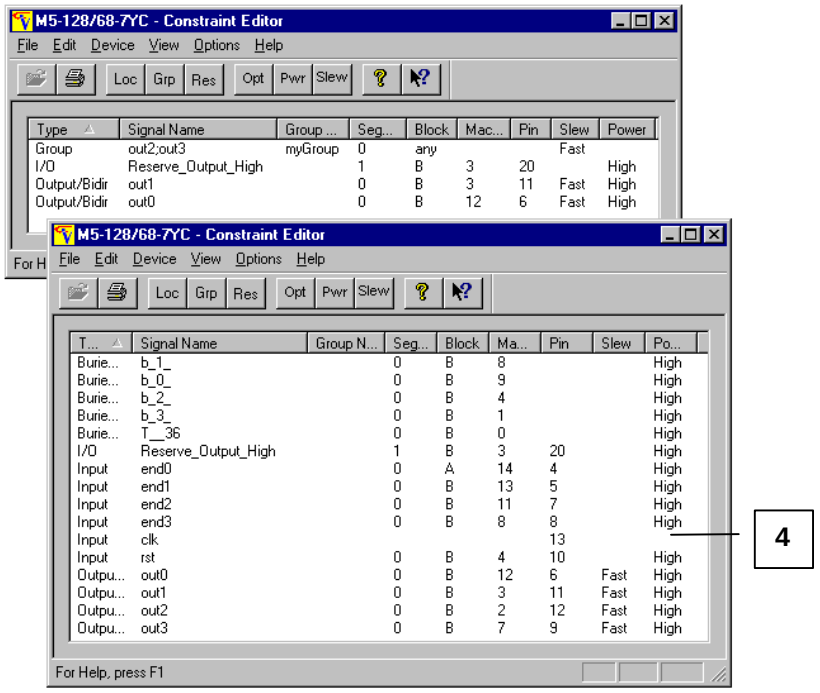
3 Click **Yes** to continue, and then click **Close**. Notice that the green check mark next to Timing Analysis is removed, indicating that timing constraints have changed and that the design should be processed again.

4 Now open the Constraint Editor and view the file contents. In the Sources window, **select the device**. In the Processes window, double-click **Constraint Editor** to open it.

The picture on the right [**4**] shows the smaller file containing the constraints set earlier in Task 5. The larger file on top contains the constraints that have been backannotated.

Notice that there are many more constraints now than there previously were. Also notice that the constraints you specified in Task 5 are still retained. For example, signal **out0** is assigned to pin **6**.





Task 11: *Congratulations!*

You have completed the Design Implementation and Verification tutorial. In this tutorial you have learned how to:

- choose a device family and select a specific device.
- compile a source file, changing your design entry format into Boolean equations.
- import a test vector file into the project, and then run equation simulation and view the results using the Waveform Viewer.
- use the Constraint Editor to specify pin and node assignments, group assignments, pin reservations, power level settings, output slew-rates and JEDEC file options for the Fitter.
- set optimization options for the Fitter using the Global Optimization dialog.
- run the Fitter and open the Fitter report.
- use the Performance Analyst to analyze the timing results.
- change the Fitter optimization options and re-analyze the timing results.
- backannotate the project assignments and view them using the Constraint Editor.

Tutorial 5

Simulating a Design with ModelSim

This tutorial shows you how to simulate a gate-level netlist for a MACH device after it has been implemented with DesignDirect. This tutorial begins with a completed VHDL design and ends when the gate-level netlist file is simulated using Model*Sim*.

Prerequisites

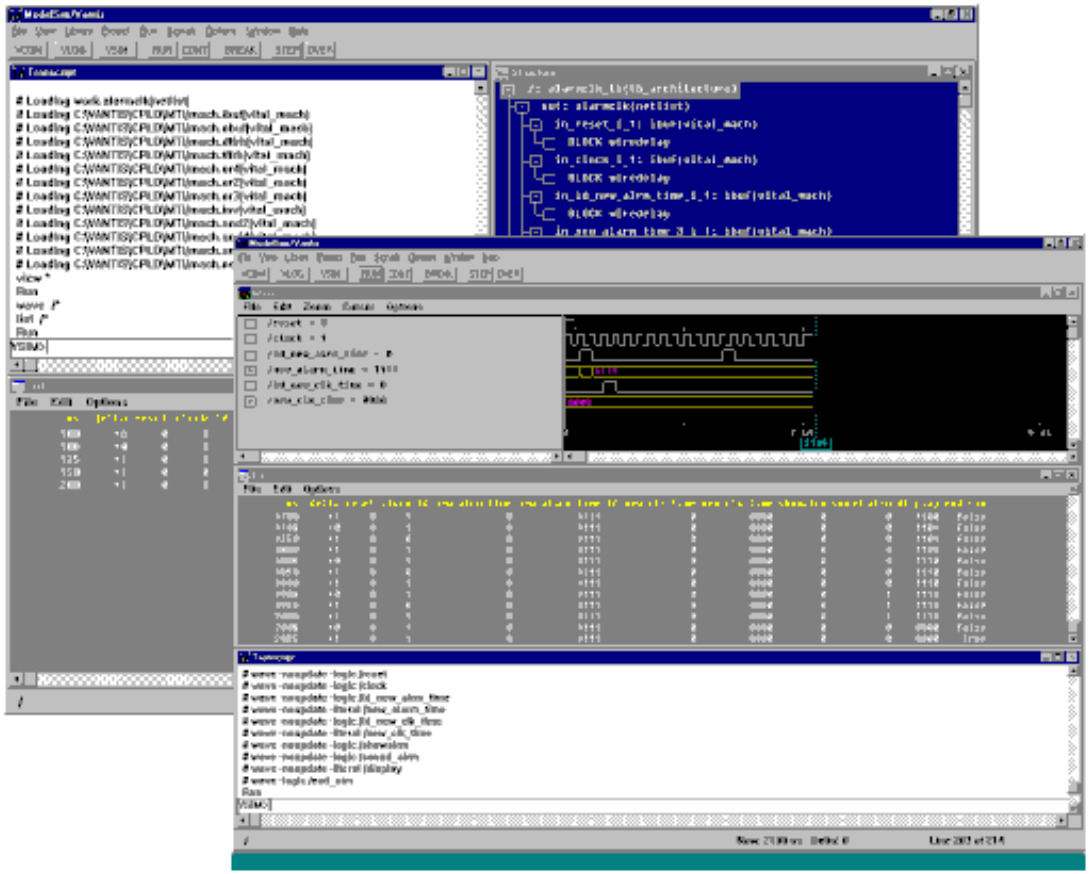
Before attempting this tutorial, you should complete the **DesignDirect Quick Start Tutorial** in the *DesignDirect-CPLD User Guide*, "Introduction" chapter.

Learning Objectives

When you are finished with this tutorial, you should know how to generate a VHDL netlist and SDF file in DesignDirect. You will also understand how to import these files to ModelSim, run a simulation, and view the results.

Time to Complete This Tutorial

The time to complete this tutorial is about 30 minutes.



Task 1: *Create a Structural Netlist in DesignDirect*

Before beginning your final verification after a design has been implemented, you must create a simulation netlist.

From the DesignDirect-CPLD program you can output VHDL or Verilog gate-level netlist and Standard Delay Format (SDF) files. These netlists can then be simulated in ModelSim using MACH primitive libraries provided with DesignDirect-CPLD and a user created test bench.

To create simulation input files:

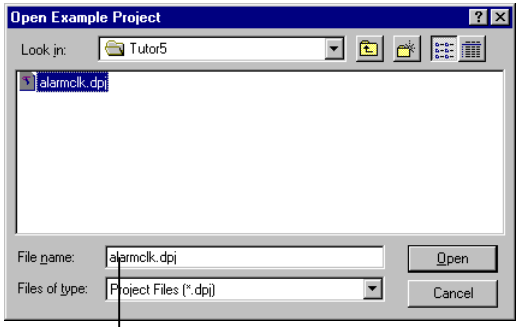
1 Start DesignDirect, if it is not already running.

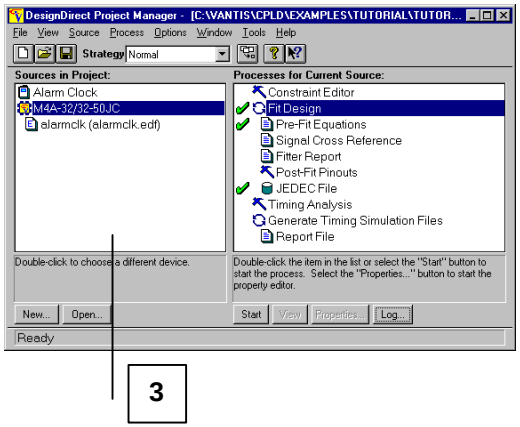
2 Choose **File** > **Open Example** and open the ..\Tutorial\Tutor5\alarmclk.dpj file.

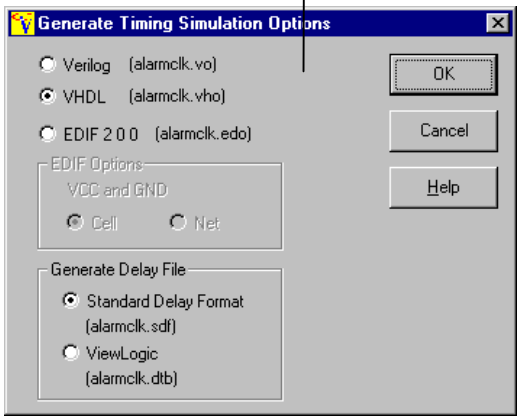
3 In the Sources window, select the target device. Notice that the design has been fit.

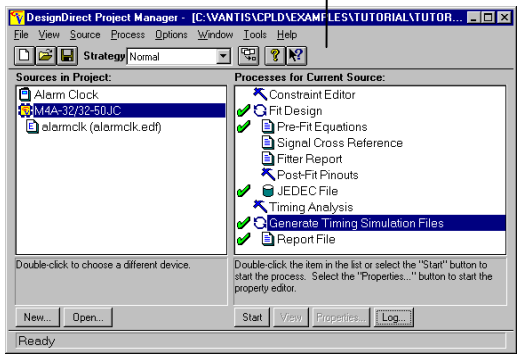
4 Now you want to create the post-fit timing simulation files. Choose **Tools** > **Generate Timing Simulation Options**. This dialog lets you specify the format of the simulation file. Leave the default options (VHDL and SDF) and click **OK**.

5 In the Processes window, double-click **Generate Timing Simulation Files**. DesignDirect creates two files that you will use later to simulate the design: *alarmclk.vho* (VHDL net list file) and *alarmclk.sdf* (timing delay file).









Task 2: *Start ModelSim and Create a Project*

To start ModelSim:

1 In DesignDirect, choose **Tools > ModelSim Simulator**. ModelSim starts with the Transcript window open. Maximize the Transcript window.

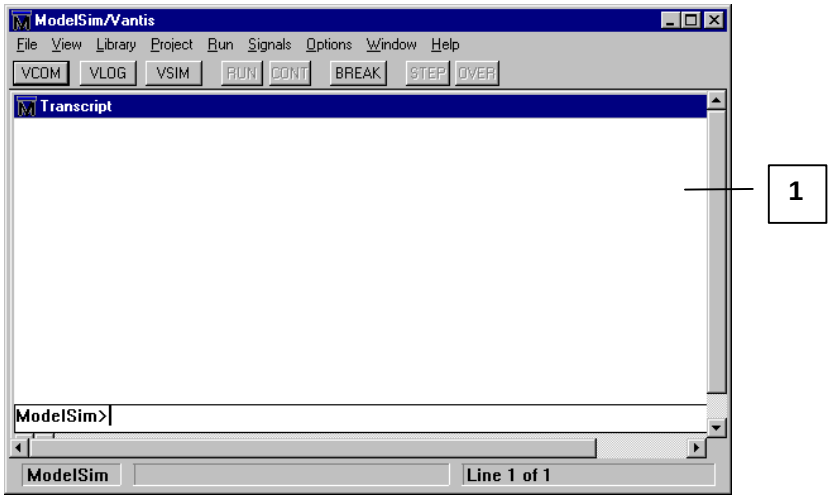
2 Change to your design directory. Choose **File** > **Directory** to open the dialog. You should be in the ..\Tutorial\Tutor5 directory. Click **OK**.

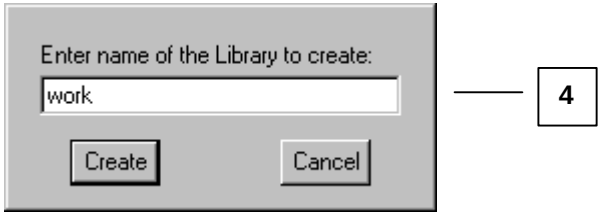
3 Now you need to create a new project file in your working directory. This project file will remember things about your environment (such as window sizes and positions, simulation options, etc.), so that next time you work in this directory, Model*Sim* will automatically reload your environment for you. Choose **Project** > **New**. In the dialog, type **modelsim.ini**, and then click **Create**. ModelSim sets the project file to the specified path.

> ***Note**: Model*Sim *automatically loads a new project file when you change directories. If there is a* modelsim.ini *file in the directory you change to, it will be loaded. If there is not one there, the original* modelsim.ini *file will be loaded. Therefore, to avoid overwriting the original* modelsim.ini file, it is important that you create a new file.

4 Before you can compile a VHDL source file, you need to create a design library to hold the compilation results. Choose **Library** > **New**. In the dialog, type **work**, and then click **Create**. This creates a VHDL library named work under the current directory.

> ***Note**: To create a library folder, you must use the previous step. You cannot use the DOS "mkdir" command or Microsoft Windows to create this directory.*





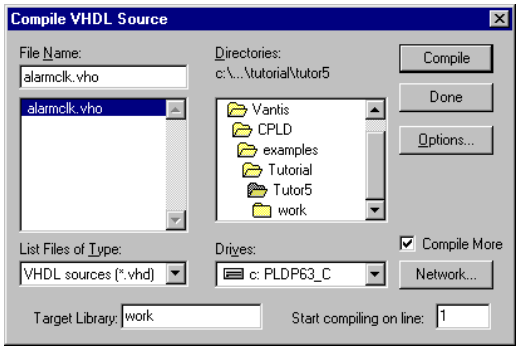
Task 3: *Compile the Design*

Now you are ready to compile the design. Model*Sim* compiles one or more VHDL design units with a single invocation of VCOM, the VHDL compiler. The design units are compiled in the order that they appear on the command line. For VHDL, the order of compilation is important – you must compile any entities or configurations before an architecture that references them.

In this tutorial, there are two files that you will compile: *alarmclk.vho*, the VHDL net list file, and *alarmclk_TB.vhd*, the test bench file.

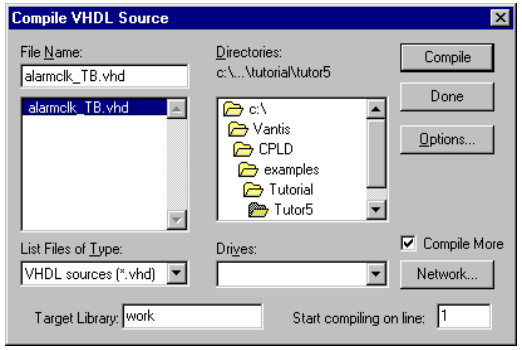
To compile the design:

1. Choose **File** >**Compile VHDL** to open the dialog.
2. In the File Name field, type ***.vho** and press **Enter** to view the files.
3. Select the **alarmclk.vho** file and click **Compile** to invoke VCOM, the Model Technology VHDL compiler.
4. When the process is complete, click **Done** to close the dialog.
5. Now you have to compile the test bench file. Choose **File** >**Compile VHDL** to open the dialog again.
6. Select the **alarmclk_tb.vhd** file and click **Compile**.
7. When the process is complete, click **Done** to close the dialog.



1 - 4

5 - 7



Task 4: *Prepare to Run the Simulation*

After compiling the design units, you can proceed to simulate your designs with VSIM.

To start the simulation process:

1 Choose **File** > **Simulate** to open the dialog.

The Simulate a Design dialog has four tabs - Design, VHDL, Verilog, and SDF - that let you select various simulation options.

2 On the Design tab, select the test bench design unit, **alarmclk_tb**.

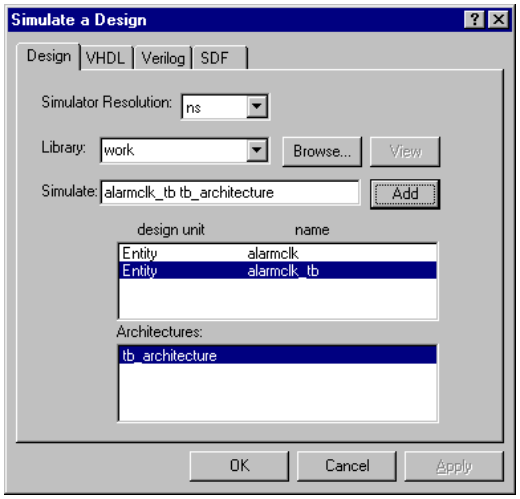
3 Click **Add** to add the test bench file and the architecture to the timing simulation.

4 On the SDF tab, click **Browse** and select the timing delay file, ..\Tutor5**alarmclk.sdf**.

5 Click **Open** to add the timing delay file to the timing simulation.

6 The Apply to Region field lets you specify the design region to use with the selected SDF options. Type a design module instantiation label: **/UUT**.

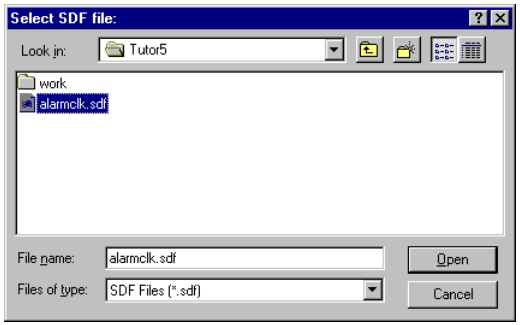
7 Click **OK**.

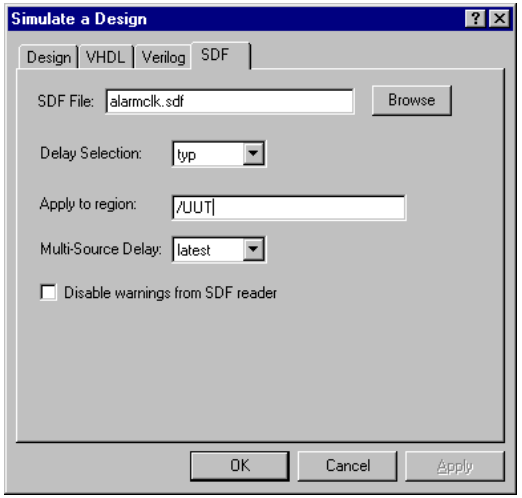


1 - 3

6 - 7

4 - 5





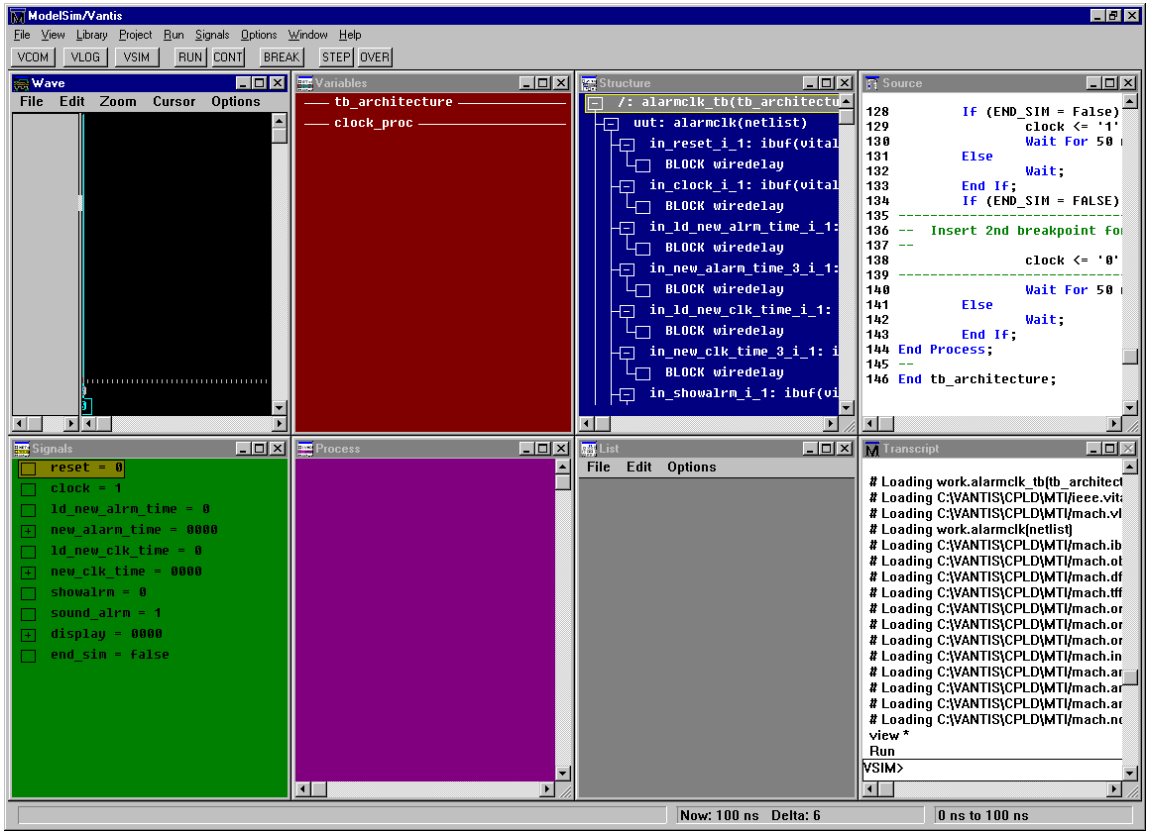
Task 5: *Set up the Windows to View the Results of the Simulation*

In addition to the Transcript window, the Model*Sim* application window also accommodates seven other windows for use during your design compilation, simulation, and debugging. They are the List, Signals, Source, Structure, Process, Variables, and Wave windows.

To open all windows:

1. Maximize the ModelSim application.
2. Choose **View** > **All** to open all of the ModelSim windows.
3. Choose **Window** > **Tile Vertically** to organize the windows.

***Note**: The windows in your application may not be organized exactly like the picture on the right. That's okay.*

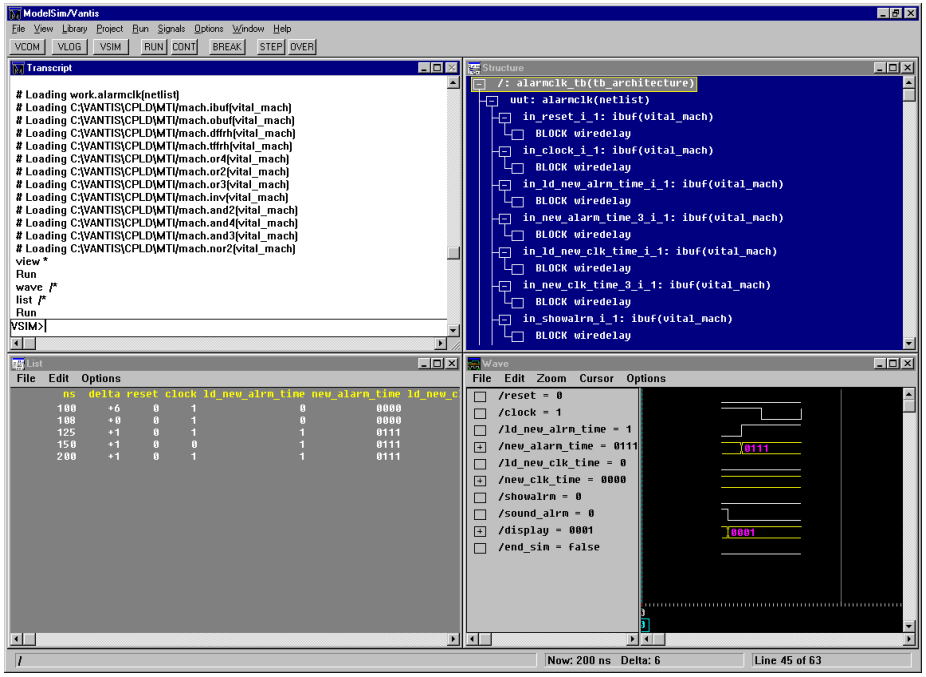


Task 6: *Add Signals in Region to Wave and List Windows*

The Wave and List windows let you view the results of your simulation in either waveform or table formats.

To add signals in region:

1. Close the **Source**, **Varialbles**, **Signals**, and **Process** windows.
2. Choose **Window** > **Tile Vertiaclly** to resize the windows again. Your screen should look something like the one on the right.
3. The Structure window provides a hierarchical view of the structure of your design. A level of hierarchy is created by each HDL item within the design. In the Structure window, select the top-level structure- the test bench (**alarmclk_tb**).
4. Choose **Signals** > **Add to Waveform** > **Signals in Region**. The software displays all signals and nets in the HDL item selected as the current region (alarmclk_tb).
5. Now choose **Signals** >**Add to List** > **Signals in Region**. The software lists all signals and nets in the HDL item selected as the current region (alarmclk_tb).



Task 7: *Specify Simulation Options and Run the Simulation*

To specify simulation options:

1 Choose **Options** > **Simulation Options** to open the dialog.

2 On the VSIM tab, the Default Run Length is 100 ns. Change this to **2100** ns and click **OK**.

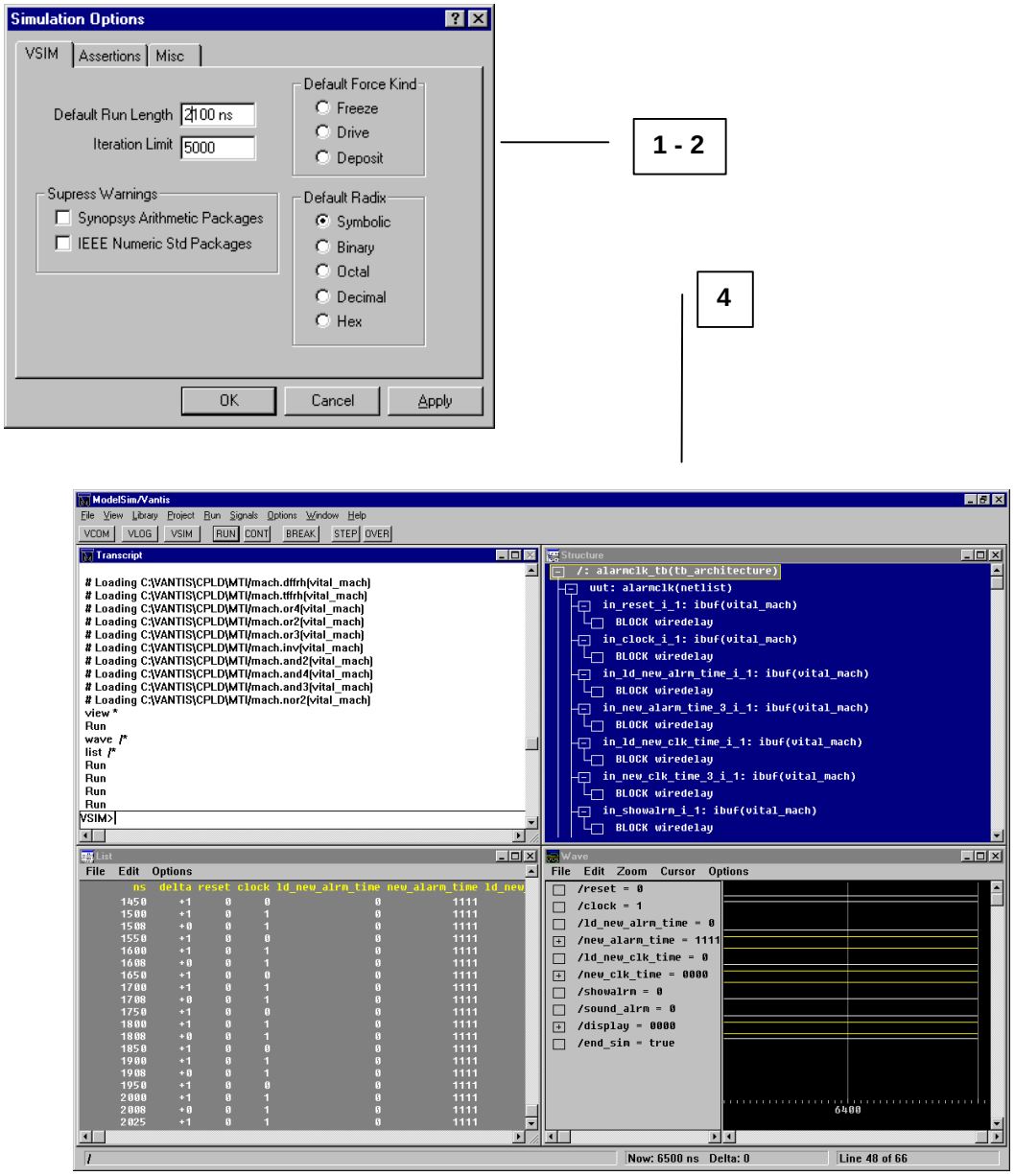
3 A dialog appears asking you "Do you wish to save the new settings in your project file?" Click **Yes**. This saves the setting to the *modelsim.ini* file you created at the beginning of the tutorial.

4 Click **Run**. The Run command advances the simulation by the specified number of timesteps (2100 ns).

Notice the content of the **Wave** and **List** windows has changed.

The Wave window displays the waveforms of the included signals. The data displayed in this window should match the results you had in your functional simulation, except timing delays have been added.

Notice that The List window displays the information in table format.

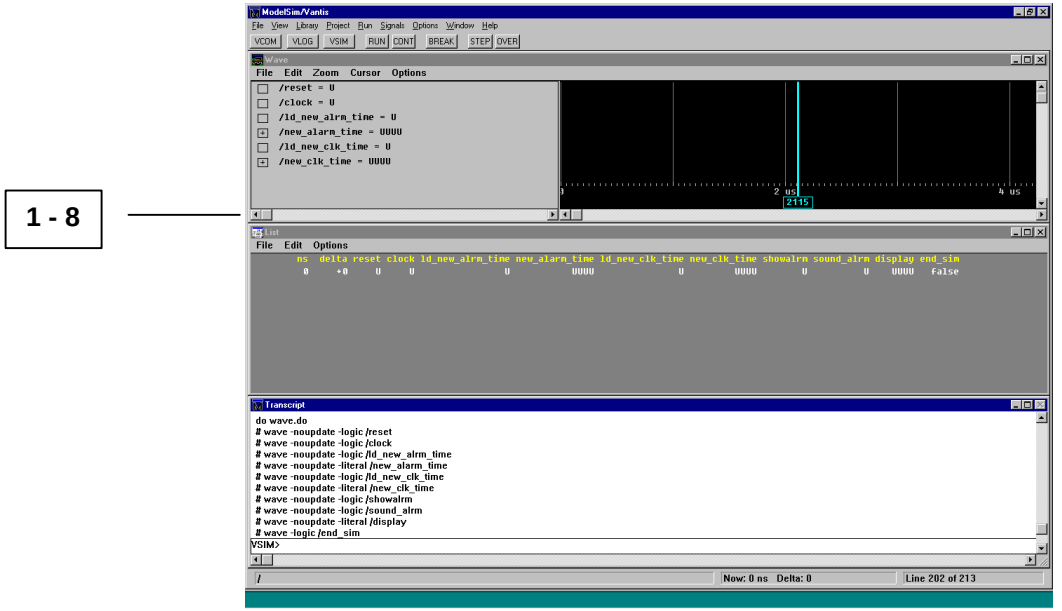


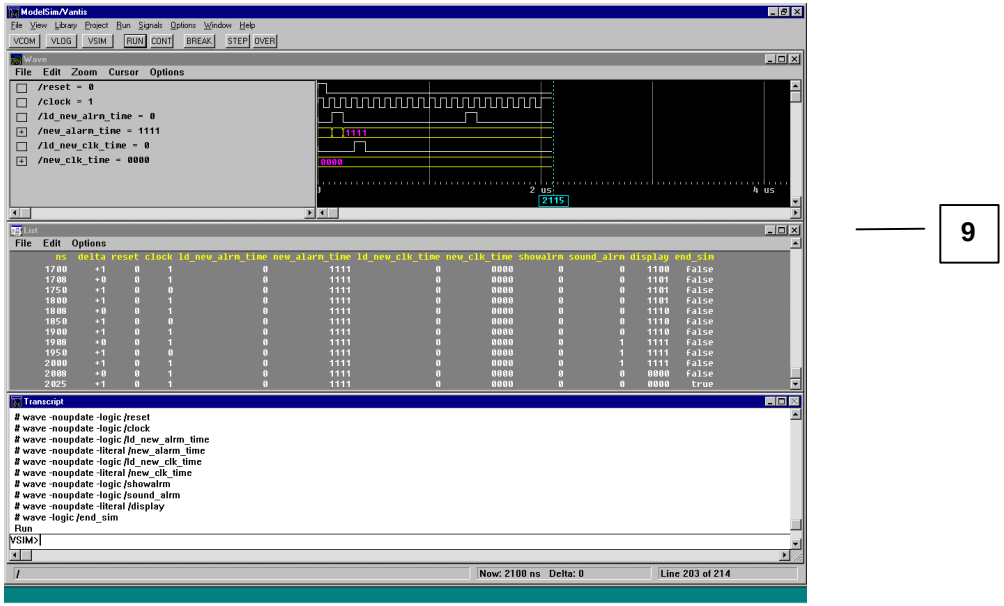
Task 8: *Save the Configuration and Re-run the Simulation*

You can save the List and Wave window configurations. This lets you re-run a simulation more quickly.

To save the configuration and re-run the simulation:

1. In the List window, choose **File** > **Save Configuration** to open the dialog.
2. Make sure you are still in your project directory (Tutor5), and save the file as **list.do**.
3. Follow the same procedure for the Wave window, except name the file **wave.do**.
4. In the main window, choose **File** > **End Simulation**. Click **Yes** to the question.
5. Now, start a new simulation run using the saved configurations. On the toolbar, click **VSIM** to open the dialog.
6. Add the test bench file (alarmclk_tb) to the simulation and click **OK**.
7. In the Transcript window command line, type **do list.do**, and then press **Enter**. The software reloads the saved List configuration.
8. At the command line, type **do wave.do**, and then press **Enter**. The software reloads the saved Wave configuration.
9. Click **Run** to run the simulation again. Notice that the software updates the content of the List and Wave windows.





Task 9: *Congratulations!*

You have completed the Simulating a Design with ModelSim tutorial. In this tutorial you have learned how to:

- create a structural netlist in DesignDirect.
- start ModelSim and create a project.
- compile the design.
- prepare to run the design.
- set up the windows to view the results of the simulation.
- add signals in region to the Wave and List windows.
- specify simulation options and run the simulation.
- save the configuration and re-run the simulation