united nations
educational, scientific
and cultural
organization

international atomic
energy agency

the
abdus salam
international centre for theoretical physics

301/1152-11

Microprocessor Laboratory
Sixth Course on Basic VLSI Design Techniques
8 November - 3 December 1999

ADDITIONAL MATERIAL TO LECTURES
BY

Nizar ABDALLAH
ACTEL Corporation
955, East Arques Avenue
94086-4533 Sunnyvale, CA
U.S.A.

Actel Desktop 2.0

Lab Guide

November 17, 1999

Lab Introduction

Requirements

These labs will take you step-by-step through the process of capturing, simulating and netlisting a design which you will then program into an Actel FPGA.

These labs reflect the Actel DeskTop version R3-1998.

Windows 95 is the assumed operating system.

General Format

All the labs in this manual are organized the same way. On the left, you will see a TASK column. This is what you do.

On the right, there is a RESULT column. This is the result of the task on the left and, in some cases, offers extra explanation or questions to remind you of why you are performing this task.

It is hoped that the TASK column will suffice for quick refreshers later when you get back to work.

All typed commands will appear in **lower case bold**, for example: Type **comp**

A carriage return must be entered after every typed command

All menu selections will be shown in **Bold** with only the first letter capitalized (the way they appear in the menu). Walking selections will be separated by a ">" character, for example: Select **File > Output Format**

All directory and file names will appear in *lower case italics*, for example: *\actel_designs\dmac*

Lab Setup

Creating a VeriBest Project with Actel Libraries

TASK	RESULT
1. Insert your training floppy into the A drive.	
2. From the Windows Start button, select **Programs > Actel DeskTOP > Actel DeskTOP**	The VeriBest DesignView window will open.
3. From the top line menu, select **Project > New**	A dialog will appear prompting you to specify the project name and location.
4. Make the following entries: Project Name = **DMAC** Location of Project File = **C:\actproj\dmac**	
5. Click **Next**.	Another dialog will appear prompting you to specify a simulation language and an Actel technology (family).
6. From the Simulation Language pull-down: • Verify that **VHDL87** is selected. • From the Technology listing, select **Actel 3200DX Technology** • Click **Next**	A window will appear prompting you to add stimulus files to the project.
7. Click **Next**	A window will appear prompting you to add design files to the project.
8. Click **Next**	A window will appear displaying a summary of your project definition.
9. Verify that your project matches the one shown in Figure 1 below.	

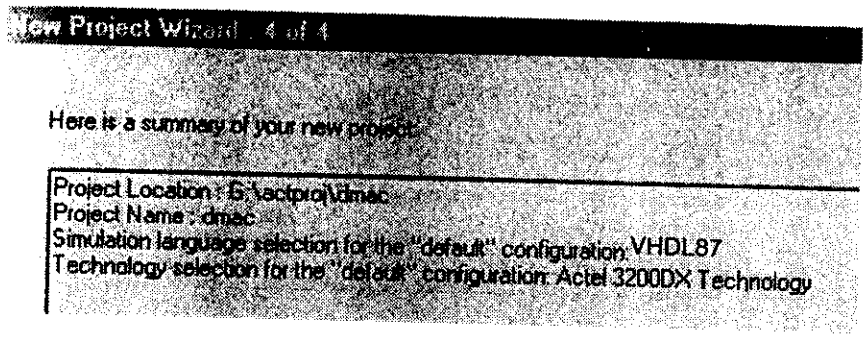


Figure 1. New Project Information Summary for DMAC Project

10. Click **Finish**.

Adding Your Own Libraries to the Project

We have created an Actel border symbol and placed it in our own library. In order to access this border, you must add our border library to the Actel libraries already present in your project.

TASK	RESULT
1. Insert your training floppy into the A drive.	
2. From the DesignView top-line menu, select **Project > Settings**	A Project Settings window will open with three tabs: Design File Locations Libraries
3. Click the **Libraries** tab. In the Library Search Order toolbar, click the New Library icon	A new library line will be added to the end of the search order.
4. At the end of the new, blank library entry, click the Browser icon. In the resulting browser, locate and open the following file: **A:\DeskTOP_Training\act_bord.slb** We need our custom border symbol to be found first, so it's .slb file must be first in the library seach order.	The *act_bord.slb* file will be added to the end of the search order. A *.slb file is a library file that contains various symbols.
5. Highlight the *act_bord.slb* entry and click the Move Up icon until it is listed first.	
6. Click **OK** to exit the Project Settings window.	
7. Select **File > Exit** to quit from VeriBest DesignView.	

Lab 1

Generating Macros with ACTgen

Tasks

In this lab, the student will:

1. Invoke ACTgen.
2. Generate a 6x8 RAM macro in structural VHDL format.
3. Generate an up counter macro in structural VHDL format.
4. Generate a down counter macro in structural VHDL format.

Generating Macros with ACTgen

TASK	RESULT
1. Select **Programs > Actel DeskTOP > Actel DeskTOP**	The VeriBest DesignView window will open.
2. Select **Project > Open**	A browser will appear.
3. Find and open your *C:\actproj\dmac\dmac.prj* file.	The project you created in the last lab will open showing an empty design hierarchy
4. In the right side of the top tool bar, click the ACTgen icon	The ACTgen window will open.
5. From the Family pull-down menu, select **3200DX** family then click the **RAM** macro type.	The RAM fill-in-the-blank form will open.
6. In the appropriate fields, describe the RAM as follows: **Depth = 6** **Width = 8** **RAM Fanin = Auto** **Write Enable = Active High** **Read Enable = Active High** **Write Clock = Rising** **Read Clock = Rising**	

7. Verify that your settings match those shown in Figure 2.

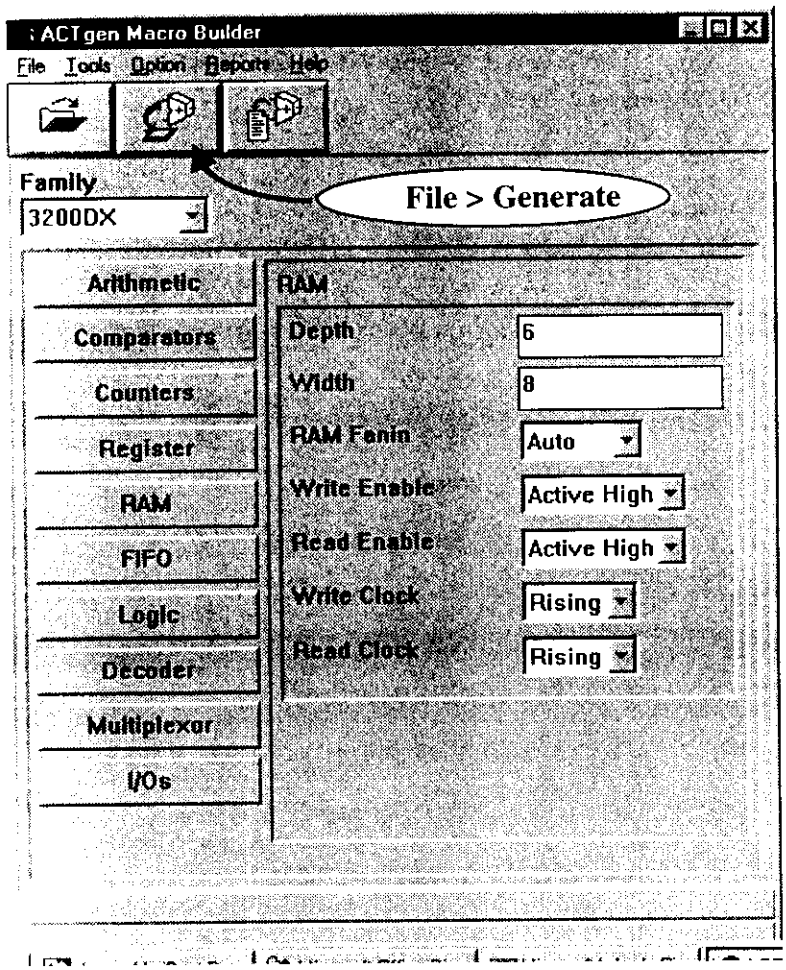


Figure 2. The ACTgen RAM Window

8. Click the **File > Generate** button (as shown in Figure 2.)

   The Generate Macro window will open

9. From the Netlist/CAE Formats list, select **VHDL**

   For the **File Name** enter **MY_RAM** and click **OK**

   An ACTgen Report window will open, listing pertinent implementation information.

   This will take a minute to complete.

10. Verify that MY_RAM has been added to the design hierarchy shown in the DesignView window.

11. In the ACTgen Macro Builder window, click the **Counters** button.

    The counters form will appear.

12. Specify the parameters listed below:

    Counter = Linear
    Variations = Balanced
    Width = 8
    Sequential Type = Default
    Direction = Up
    Asyn Clear = Active Low
    Sync Load = Active High
    Count Enable = Active High
    Clock = Rising
    Terminal Count = None

13. Generate the macro with the file name UPCNT.

    Verify that it is added to your design hierarchy.

14. Return to ACTgen. This time generate a linear, balanced, 8-bit **DOWN** counter with an **active high Terminal Count**.

    Accept all the other default values.

    Name it DWNCNT.

15. Try to exit ACTgen.

    You will be prompted to save the log file.

16. Elect to save the log file to a file called *actgen.log*. Notice where the file is saved.

    Your Save As window should match Figure 3.

17. Click **OK** to save the file and then exit ACTgen.

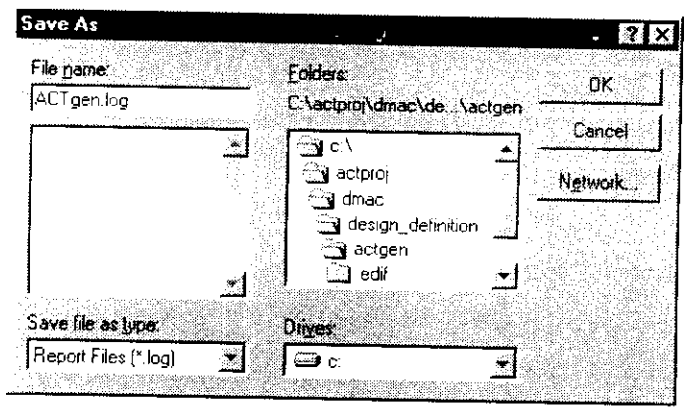


Figure 3. The Save As Window for the ACTgen Report File

Lab 2

Generating VHDL Blocks with VeriBest

Tasks

In this lab, the student will:

1. Create new VHDL files in a VeriBest project.
2. Add existing VHDL design files to the current VeriBest project.

Creating New VHDL Files in a Project

This lab assumes that the Actel DeskTOP is running and that the *dmac.prj* project is open.

TASK	RESULT
1. In DesignView's top line menu, select **File > New**	A window will appear prompting you to specify the type of the new block.
2. In the resulting dialog box, **select VHDL File** and click **OK**.	The VHDL editor will open for a new file named VHDL File1.
3. In the VHDL editor, type in the description below. Notice that all VHDL key words are recognized by the editor and displayed in a different color.	

```
library ieee;
use ieee.std_logic_1164.all;
entity decode is
 port(data: in std_logic_vector(4 downto 0);
      aeb:  out std_logic);
end decode;
architecture behave of decode is
  begin
  with data select
    aeb <= '1' when "11111",
           '0' when others;
end behave;
```

4. When finished, select **File > Save As** and enter *decode* as the new name.

5. Select **File > Close**

The VHDL will close and you will be returned to the Design Hiearchy view. Notice that the new DECODE block has been added to the project.

Adding Existing VHDL Files to a Project

In the last section, you created a VHDL block "from scratch". In this section, you will copy existing VHDL files into your project.

TASK	RESULT
1. In the top line menu, select **Project > Add to Project > Add Design File**	A browser will open.
2. Focus the resulting browser on your A: drive.	Three VHDL files will appear: *strobe_gen.vhd, controll.vhd* and *dmac_stim.vhd.*
3. Select ***controll.vhd*** and click **Open**.	A window will appear asking whether the file should be linked to the project or copied.
4. Click **Copy**	The VHDL block will be added to the project.
5. Use the same procedure to copy in the ***strobe_gen.vhd*** file.	
6. From the Design Hierarchy, highlight the ***strobe_gen.vhd*** file, right-click your mouse and select **Open** from the resulting pop-up.	Notice that DesignView recognizes the block as a VHDL file and opens the VHDL editor accordingly.

Lab 3

Simulating with Veribest VHDL Simulator

Tasks

In this lab, the student will:

1. Generate stimulus for VHDL blocks.
2. Associate stimulus for VHDL blocks.
3. Use the VeriBest VHDL Simulator to simulate VHDL blocks.

Generating and Associating Stimulus for a VHDL Block

TASK	RESULT
1. From the Design Hierarchy view, highlight the ***decode.vhd*** file, right-click your mouse and try to select **Generate Stimulus**. What happens? _____	OOPS - the menu is greyed-out! This is because decode has not been set as the "root" design.
2. Right-click your mouse again, but this time select **Set as Root**.	The *decode.vhd* file will change to bold face.
3. Now select Generate Stimulus	A Wavebench tool will open with a new file named *decode.wsw*
4. Click the Insert Edge icon  . Move your cursor to the waveform area of the Data[4:0] bus and click the mouse to insert transitions at approximately 20, 40, 60 and 80 ns. Don't worry about the values in each segment of the bus yet.	Bus transitions should appear at each insert point.
5. Click the Escape icon  to exit from the "insert edge" mode.	

6. In the first bus segment of Data[4:0] (from 0ns to 20ns), double click the value (for example, hZZ)

   The value will highlight and turn into an editable field.

7. Enter a value of **00** and hit Enter.

   Notice that the value shown is **h 00**. The hex prefix is added automatically.

8. Repeat this procedure to assign five values as shown in Figure 4 below:

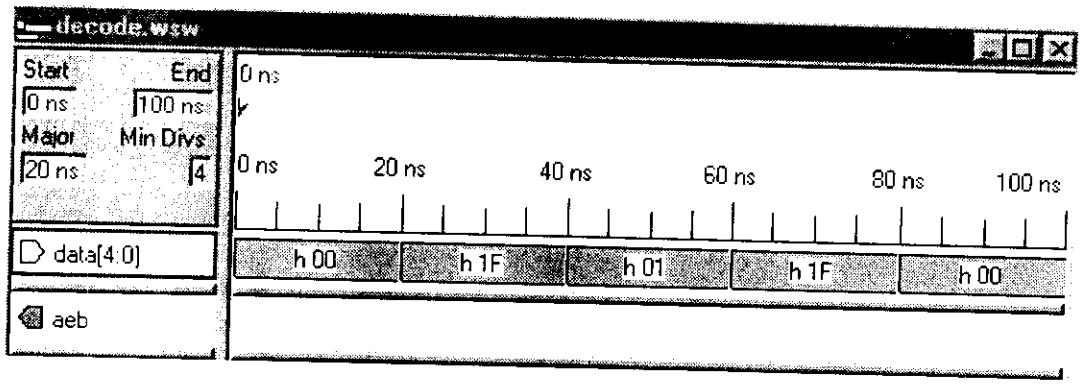


Figure 4. The decode.wsw Stimulus File

9. When you are done, select **File > Close** and elect to save the file in all the resulting windows.

   You will be returned to the Design Hierarchy view.

Congratulations! You just created a stiumulus file! This file has automatically been named and saved to the current project. To see the file . . .

10. Click on the **Simulation Manager** tab.

    The VB Project Simulation Environment display will appear.

11. Double click on ***decode.wsw***

    All the stimulus files for the decode block will be displayed. At this point, there should only be one: *decode_stim.vhd*.

    To create different stimulus files for this same block in the future, make sure you select **File > Save As** in WaveBench tool. Otherwise, this same *_stim.vhd* file will be overwritten.

Associating Stimulus to a Block

Before simulation can begin, you must associate a particular stimulus file to the block that you want to simulate.

TASK	RESULT
1. In the DesignView window, return to the **Design Hierarchy** tab.	
2. Highlight ***decode***, right-click your mouse and select **Associate Stimulus**	A window will open displaying all the known stimulus files for this block.
3. Under the *decode.wsw* entry, select the ***decode_stim.vhd*** file and click **OK**.	The decode block is now ready for simulation. Notice that a small blue symbol has been added to the decode block icon. This symbol indicates that the block has been associated with a stimulus file.
4. In the future, you may forget which stimulus file has been associated. To determine this, highlight the block (in this case, **decode**) and right-click your mouse to select **Properties**.	A Properties window will appear. Notice the **Stimulus File** entry at the very bottom.

Performing Behavioral Simulation of a VHDL Block

TASK	RESULT
1. From the Design Hierarchy tab, highlight the ***decode*** block and right-click your mouse to select **Open in Simulator**.	You may see a prompt stating that: There is existing library environment . . . etc If you do, select **Yes** to reinitialize. Eventually, your simulator will open displaying the decode source hierarchy.
2. Verify that you have two files shown in the Decode source hierarchy in this order: *Decode.vhd* *Decode_stim.vhd.*	

3. Click the Compile All icon 

   If any errors are listed, double click them. A VHDL editor will open and display the offending code. If you have questions about how to correct the error, call the instructor.

   All the .vhd files will be compiled in the appropriate order.

4. Click the Execute Simulator icon 

   All the .vhd files will be linked and the simulator will be invoked.

5. Click the Wavebench icon 

   The Wavebench tool will open.

6. From the Wavebench toolbar, select the Add Signals icon 

   A window will open displaying all the I/O signals of the decode block.

7. Highlight the **Data[4 downto 0]** vector click the **Add** button. Repeat this for the **AEB** output signal.

8. Click the Run Simulation icon  (it will run for 100ns by default)

   Partial waveform data will appear in the Wavebench tool.

9. From the Wavebench toolbar, click the Zoom Fit icon 

   Full waveform data will appear.

10. Verify that your simulation results match those shown in Figure 5.

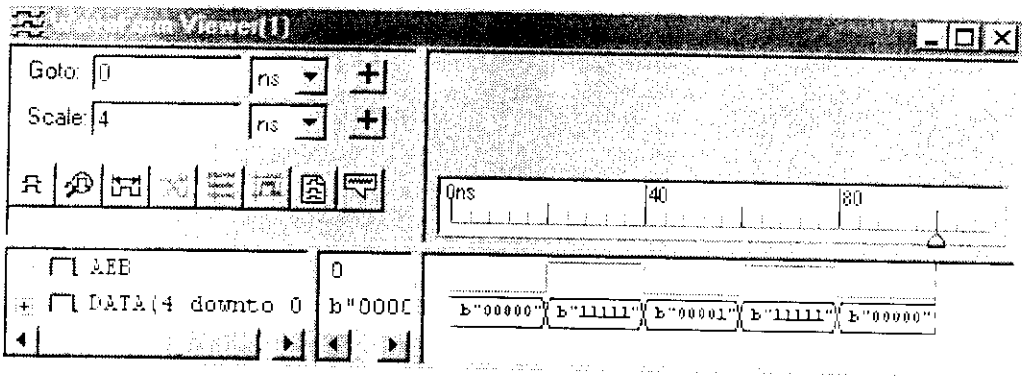


Figure 5. Results from the Decode Simulation

11. Exit from the Wavebench tool and the simulator.

Lab 4

Synthesis with Synplify

Tasks

In this lab, the student will:

1. Synthesize a behavioral VHDL block with Synplicity's Synplify.
2. Perform post-synthesis VHDL simulation withVeriBest's VHDL Simulator.

Synthesizing a VHDL Block

TASK	RESULT
1. From the Design Hierarchy view, highlight the ***decode.vhd*** file and right-click your mouse to select **Open in Synthesis**.	The Synplify tool will open displaying various pieces of information about the block you have selected to synthesize.
2. Maximize the Synplify window.	
3. Take a few minutes to verify that the files listed in the Source section are correct. What is the name of the Result file and where will it be placed? _____	
4. Click the **Change** button adjacent to the Target options.	A Set Device Options window will open.
5. In the Part pull-down, select the **A32100DX** device and check the **Disable I/O Insertion** box.	This defines the target FPGA device and prohibits the synthesis tool from inserting I/O macros on the ports of the VHDL block. If this VHDL file represented the entrire FPGA, the insertion of I/O macros would be desirable, but this is only one block of many in the design so I/Os are NOT desireable.
6. Verify that your settings match Figure 6 and click **OK**.	The Set Device Options window will close and your new options will appear on the Target line of the Synplify window.

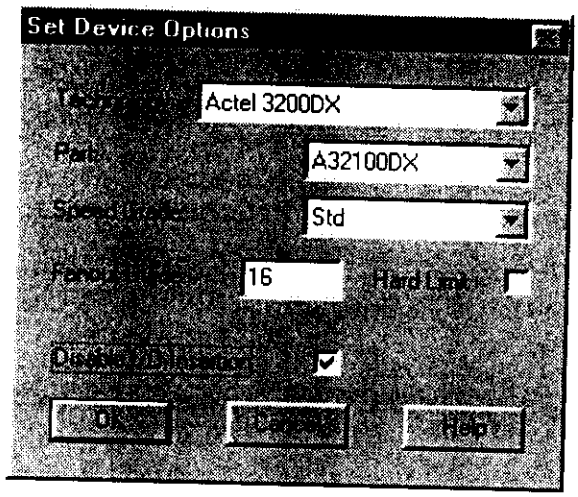


Figure 6. Synplify's Set Device Options Window

7. Click the **Run** button.

   The synthesis will start by compiling, then mapping and will then be done.

8. Click the **View Log** button and scroll down in the log file until you see a line that is highlighted yellow.

   All errors, warnings or notes are highlighted yellow. Each highlighted line should be investigated.

9. Double click on the highlighted line.

   A VHDL source editor will open and highlight the part of the description that caused the error, warning, or note.

10. Verify that you only have notes. If you have errors or warnings, call the instructor.

11. Select **File > Exit** to quit the Synplify tool.

    You will be prompted to save the current information to *decode.prj*.

12. Click No to the decode.sdc is already part of p...

    DesignView will automatically recog...

    to a structural VHDL file by running a program called *edn2_nal*.

13. To see the effect of the synthesis, go to DesignView and click the **Vendor Manager** ...

    A Default Vendor Configuration hierarchy will appear.

14. Verify that *decode.edn* appears under the **Place-Route Files** directory and that *decode.vhd* appears under the **Synthesis Files** directory.

15. Highlight ***decode.vhd***, right-mouse click and select **Open**

    A VHDL source editor will open.

    Notice that it is a structural description comprised of CM8 modules (Actel's basic building block).

16. Close the VHDL editor and return to the **Design Hierarchy** tab in DesignView.

Review: Simulating Structural VHDL

To simulate a structural VHDL block, you can use the same stimulus that you created for behavioral simulation.

TASK	RESULT
1. In DesignView, select the **Vendor Manager** tab.	The Synplify tool will open displaying various pieces of information about the block that you have selected to synthesize.

Is the *decode.vhd* file (under Synthesis Files) associated with a stimulus file? _____

How can you tell? _____

2. Highlight ***decode.vhd*** and associate it to the same stimulus file that you used for behavioral simulation.

   (Hint: *decode.wsw* > *decode_stim.vhd*)

3. Again highlight *decode.vhd* and right-click your mouse to select **Open Post-Synthesis Simulator**.

   The VeriBest VHDL Simulator will open.

4. Compile all the VHDL files () and execute the simulator ).

5. Open the WaveBench tool () and add all the I/O signals () to trace

6. Run the simulator  and check your results to see if they match the behavorial results we saw in the last lab.

7. When you are finished, exit the simulator.

8. Under the Simulation Manager tab, expand the *decode.wsw* and *decode_stim.vhd* directories.

Notice that two .vpd files exist: *decode.vpd* and*PostSyn.vpd*

These two files are a record of the workspace settings of the behavioral simulation (*decode.vpd*) and the structural, or post-synthesis simulation (*PostSyn.vpd*)

The next time you simulate this block, its .vpd file will be opened and you won't have to configure the WaveBench tool.

Lab 5

Completing the DMAC Schematic

Tasks

In this lab, the student will:

1. Add an existing schematic file to the current project.
2. Add ACTgen blocks and VHDL blocks into a top-level schematic.

Adding an Existing Schematic to the Current Project

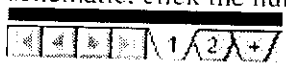
This lab assumes that the Actel DeskTOP is running and that the *dmac.prj* is open.

TASK	RESULT
1. Select the **Design Hierarchy** tab in DesignView. From the top line menu in DesignView, select **Project > Add to Project > Add Design File**	A browser will open.
2. Focus the resulting browser on your A: drive.	One schematic file will appear: *dmac.sbk*
3. Select *dmac.sbk* and click **Open**.	A window will appear asking whether the files should be linked to the project or copied.
4. Click **Copy**	The schematic file will be added to the project.
5. From the Design Hierarchy, highlight the *dmac* file, right-click your mouse and select **Open** from the resulting pop-up.	Notice that DesignView recognizes the file as schematic and opens the schematic editor accordingly. The schematic editor is called the VeriBest Design Capture tool, or VBDC.

Completing the Schematic - Adding VHDL Blocks

TASK	RESULT
1. Compare page 1 of your hard copy of the DMAC schematic to page 1 of what you see in DesignView. Notice that the schematic that you just added to the project is only partially completed.	
2. To add the missing VHDL blocks, click the Block icon .	A blank Place Block window will open.
3. In the Block name field, click the Browser button	A list of available blocks will appear. Notice that all your VHDL blocks are included in the list.
4. In the resulting list of available blocks, double click the CONTROLL block.	The window will close and the CONTROLL block will be attached to your cursor.
5. Move the CONTROLL block into position (use your hard copy schematic as a guide) and left-click your mouse to place it. Verify that it is truly connected by moving it and making sure that the wires "rubber band" with it.	
6. Repeat this process to add the other two VHDL blocks: STROBE_GEN and DECODE.	

Completing the Schematic - Adding ACTgen Blocks

TASK	RESULT
1. In the lower left-hand corner of the schematic, click the number 2 tab  to view page 2 of the schematic.	Page 2 of the schematic will open.
6. Notice that three ACTgen blocks are missing: MY_RAM, UPCNT and DWNCNT. Select the Symbol icon	A Symbol Menu window will appear.
7. Double click the VBACTgen entry.	A list of the ACTgen macros in this project will appear.

8. Double click the MY_RAM entry and place it in the schematic (use your hard copy as a guide).

   Verify that it is truly connected by moving it and making sure that the wires "rubber band" with it.

9. Double click the MY_RAM block.

   A Properties window will open.

10. In the Properties window, click the **Text** tab and change the Instance Name to **MY_RAM** as shown in Figure 7.

    Initially, ACTgen blocks have generic instance names (something like XCMP15). To make timing analysis easier later on, it is a good idea to change the instance name to something meaningful - like the title of the block.

11. Click **OK** to close the Properties window.

12. Repeat this procedure to place UPCNT and DWNCNT.

    Don't forget to change their instance names.

    Save the schematic.

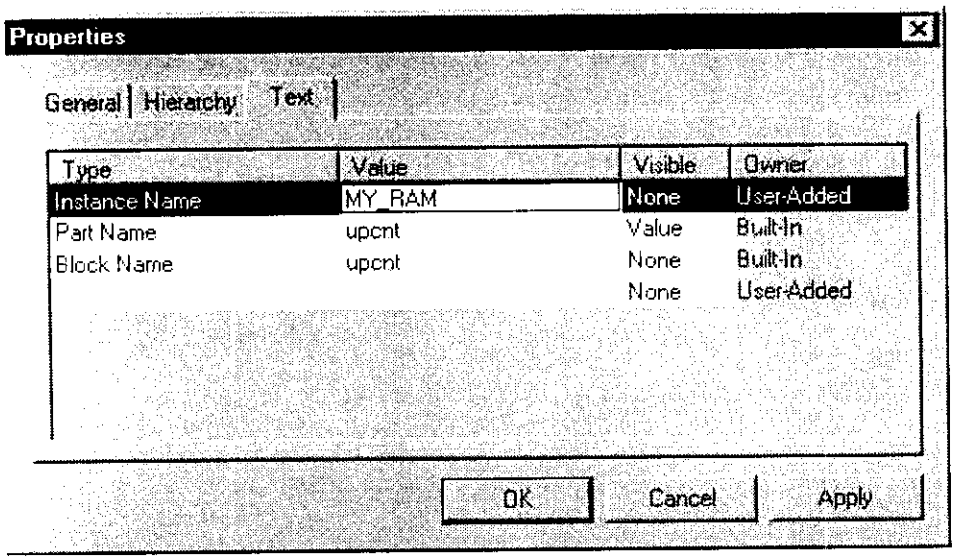


Figure 7. Adding a Meaningful Instance Name

Verifying the Schematic

Once you have completed and saved the schematic, you must verify it.

TASK	RESULT
1. Click the Verify icon  .	A File Verify window will open displaying the various checks that will be run on the design.
2. Accept the default checks and click **OK**	The verification will run and any errors or warnings will be displayed below the schematic.
3. If you have any errors or warnings, double click them and the erroneous component of the schematic will be highlighted. Call the instructor if you have errors you cannot correct. Debug and verify until you have 0 errors and 0 warnings.	

Lab 6

Synthesizing and Simulating the DMAC Schematic

Tasks

In this lab, the student will:

1. Force Synplcity's state machine encoding method to sequential.
2. Synthesize the top level DMAC schematic.
3. Add an existing stimulus file to the current project and associate it to DMAC
4. Perform the post-synthesis simulation of the DMAC design.

Synthesizing the Top Level Design

This lab assumes that the Actel DeskTOP is running and that *dmac.prj* is the open project.

TASK	RESULT
1. From the Design Hierarchy view in DesignView, highlight **dmac** and right-mouse click to select **Open in Synthesis.**	Synplicity's Synplify tool will open.
2. From the top-line menu, select **Options > Configure VHDL Compiler**	The VHDL Compiler Configuration window will open.
3. From the **Default Enumeration Encoding** pull-down, select **sequential**. Click **OK**	This will override Synplicity's Symbolic FSM Compiler and force the encoding method to be sequential.
4. Click the **Change** button adjacent to the Target options display.	The Setup Device Options window will appear.
5. Select the 32100DX device but DO NOT disable I/O insertion as shown in Figure 8.	Since this is the top level design, we want to insert I/O macros. Otherwise, the final EDIF netlist will not import into Actel's Designer.

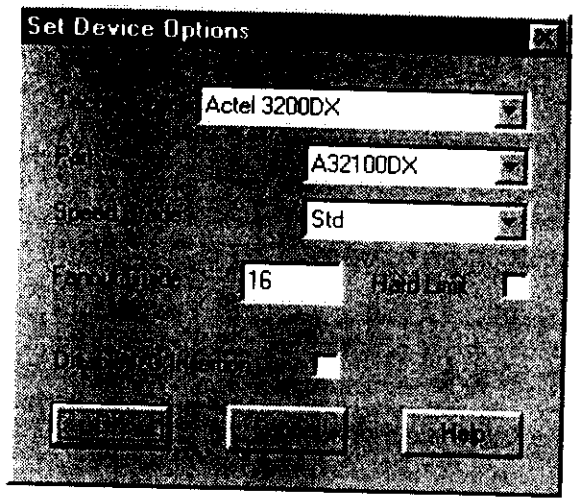


Figure 8. Synplify's Target Set Device Options Window for Top-Level Synthesis

6. Click **OK** and click the **Run** button to begin the synthesis.

   The Set Device Options window will close and the synthesis will complete

7. When the synthesis is done, click the **View Log** button.

   A log of the synthesis will appear.

   There should only be notes and warnings - no errors. Double click each highlighted entry in the log file to view the code that caused the note or warning.

   Call the instructor if you have errors or warnings that you don't understand.

8. Exit from Synplify.

   It will ask if you want to save the changes to your *dmac.prj* file. It is referring to the Target changes we made (that is, device and I/O insertion)

9. Elect to save the changes.

   Synplify will close and two new files will appear under the Vendor Manager tab:

   Dmac.vhd - a structural VHDL representation of the design

   Dmac.edn - an EDIF representation of the design.

Adding and Associating an Existing Stimulus File to the Current Project

TASK	RESULT
1. From the top line menu in DesignView, select **Project > Add to Project > Add User Stimulus**	A browser will open.
2. Focus the resulting browser on your A: drive.	One stimulus file will appear: *dmac_stim.vhd*
3. Select ***dmac_stim.vhd*** and click **Open**.	A window will appear asking if you want to copy or link the file.
4. Click **Copy**	A window will appear stating that you need to select the block to which the stimulus file applies.
5. Click **OK**	Another window will appear listing all the blocks in the current project.
6. Select the **dmac** block and click **OK**	
7. In the **Vendor Manager** tab, highlight **dmac** and right-mouse click to select **Associate Stimulus**.	An Associate Stimulus window will open listing all the stimulus files available for the dmac design.
8. Double click the ***dmac_stim.vhd*** file.	A small blue symbol will be added to the dmac icon in the Design Hierarchy view - this indicates that the block has an associated stimulus file.

Simulating the Structural VHDL DMAC Design

TASK	RESULT
1. From the **Vendor Manager** tab, highlight **dmac.vhd** and right-mouse click to select **Open Post-Synthesis Simulator** Elect to reinitialize the library environment if asked.	VeriBest's VHDL Simulator will open.
2. Compile all  the VHDL files and execute the simulator  .	
3. Invoke the WaveBench tool  and set it up to display all the I/Os (Hint: Click the  icon).	

4. Set the simulation time to 3500ns and click the Run icon  .

   The simulation will complete and the partial waveforms will be displayed in WaveBench.

5. Click the Zoom Fit icon  to see the full waveforms.

   Verify that they match those shown on your hard copy.

Congratulations! You are now ready to take your fully verified EDIF netlist into Actel's Designer for place and route!

Lab 7

Importing and Compiling an EDIF Netlist

Tasks

In this lab, the student will import and compile an EDIF netlist in Actel's Designer software.

Creating a Designer Database

This lab assumes that the Actel DeskTOP is running and that *dmac.prj* is the open project.

TASK	RESULT
1. From the **Vendor Manager** tab in DesignView, go to the Place-Route Files directory and highlight the ***dmac.edn*** entry. Right-mouse click to select **Open in Place and Route.** The EDIF netlist should import without error. Call the instructor if you get errors.	Actel's Designer software will be invoked and the *dmac.edn* netlist will automatically be imported. In addition, a database file called *dmac.adb* will be created and added to the project.
2. From the Designer flow chart, click **Compile**.	The Compile routine will begin and you will be prompted to enter a device, package, and speed grade.
3. Make the following selections: Device = **A32100DX** Package = **84 PLCC** Speed Grade = **STD** Click **Next**.	A Device Variations window will open, prompting for voltage and special pin reservations.
4. In the Pin Restrictions section, check both **Reserve JTAG Pins** and **Reserve Probe Pins** (we will discuss these later).	

5. Click **Next**

An Operating Conditions window will open.

Notice that if the Range is specified as either commercial or industrial, the Temperature and Voltage values are preset.

Custom temperatures and voltages can be specified by setting Range to Custom.

6. In the *Operating Conditions* dialog box, click **Finish** to accept the defaults.

The Compiler will begin to run.

When the compile is complete, the Compile icon will be highlighted. The status area will display any error or warning messages.

7. Verify that you have no errors.

8. Save the database and exit from the Designer.

Lab 8

Making Pin Assignments with Pin Edit

Tasks

In this lab, the student will:

1. invoke the Pin Edit tool.
2. make single pin assignments and group pin assignments.
3. save pin assignments.
4. generate a pin report.

Making Pin Assignments

TASK	RESULT
1. From the **Vendor Manager** tab in DesignView, highlight the ***dmac.adb*** file and right-click to select **Open**.	The Designer will be invoked and the *dmac.adb* database will open.
2. Click **Pin Edit** from the Designer's flow chart.	The Pin Edit tool will open displaying a package pinout diagram for the 32100DX and a list of unplaced I/O nets.
3. Use the **View > Zoom Area** command to zoom in on the lower left corner of the package pinout diagram.	
4. Click on **addr(0)** from the **Unplaced** list and drag it over to pin 30.	**addr(0)** will disappear from the Unplaced list and appear on the Placed list. Also, pin 30 will fill in, showing addr(0) adjacent to it.
5. Use the same procedure to make the following assignments: **addr(1) > pin 31** **addr(2) > pin 32**	
6. Scroll up to the upper left corner of the package pinout.	

7. From the Unplaced list, scroll down and click **wen_cntl(0)**. Hold down the Shift key and click **wen_cntl(1), wen_cntl(2), wen_cntl(3),** and **wen_cntl(4)**

   All five net names should be highlighted.

8. From the top menu, select **Edit > Place**.

   This selection is necessary whenever you are assigning a group of nets.

9. In the pinout diagram, click on pin 13.

   Since wen_cntl(0) was the first net selected, it will be assigned to pin 13.

10. In the pinout diagram, click on pins 14, 15, 16 and 17 (in that order).

    The following assignments should appear:

    wen_cntl(1) > pin 14

    wen_cntl(2) > pin 15

    wen_cntl(3) > pin 16

    wen_cntl(4) > pin 17

11. Zoom in on the top, left corner and scroll around until you see the PRB pin (pin 4). Try to place the **en(1)** net on this pin.

    It will accept your assignment, but you will receive a warning:

    ```
    pin is also used by probes
    ```

    The PRB pin is used by probe circuitry of the Silicon Explorer.

12. Drag **en(1)** from the Placed list to the Unplaced list.

    en(1) will be unplaced.

13. Try to assign **rd_clk** to pin 3.

    Why doesn't this work?

    You will receive an error:

    ```
    location incompatible with type
    of macro
    ```

14. From the main menu, select **Options > Configure List Boxes**

    A *Configure Listboxes* dialog box will appear.

15. In the **Name** field, type **re***.

    Click **OK**.

    The Unplaced list will be filtered to only display I/O nets that start with **re**.

16. Make the following assignments:

    req(1) > pin 24
    req(2) > pin 25
    req(3) > pin 26

17. Select **File > Commit** to commit your pin assignments.

18. Select **File > Close** to close Pin Edit.

19. Save the database.

20. From the Designer's main menu, select **Reports > Pin**.

    A dialog will open allowing you to specify how the pin assignment report is sorted.

21. From the List **By** pull-down, select **Number**.

    Click **OK**.

    The pin assignment report will open and be displayed.

22. From the Pin Report menu, select **File > Save.**

    A Save As window will open.

    In what directory is the pin report going to be saved? _____

23. In the File name field, enter **dmac.prp** and click **OK**.

24. Close the Pin Report window.

Lab 9

Analyzing Pre-Layout Timing

Tasks

In this lab, the student will:

1. generate a pre-layout timing report and determine whether or not timing constraints are necessary.
2. use the DirectTime Analyzer to perform a detailed timing investigation.

Generating Timing Reports

TASK	RESULT
1. Verify that the *dmac.adb* database is still open.	
2. From the Designer's main menu, select **Reports > Timing**.	A Warning box will appear prompting you to select pre or post layout.
3. Be sure to select **Pre-layout**. If you select Post-layout, the layout will automatically run first.	A Timing Report window will open.
4. Sort by **Actual**. In the **Maximum Paths** field, enter **10**. Click **OK**.	A Timing Report will open, displaying the set-to- set measurements discussed in class.
5. Review the information in the report. If you have any questions about what you see, ask the instructor to explain.	
6. Go down to the section titled: **Register (rd_clk_1_net) to Register (rd_clk_1_net)** Based on this delay, what is the maximum operating frequency for this design? _____	
7. In the Timing Report window, select **File > Save**. Name the file ***std.trp***.	This will save the report to an ASCII file which may be useful for future comparisons.
8. Select **File > Close Window** to exit the Timing Report.	

TASK	RESULT
1. From the Designer's flow chart, click the **DT Analyze** icon and again select **Pre-layout**.	The Filters window will open as shown in Figure 9. You will use this window to filter the data reported by the DT Analyze tool.

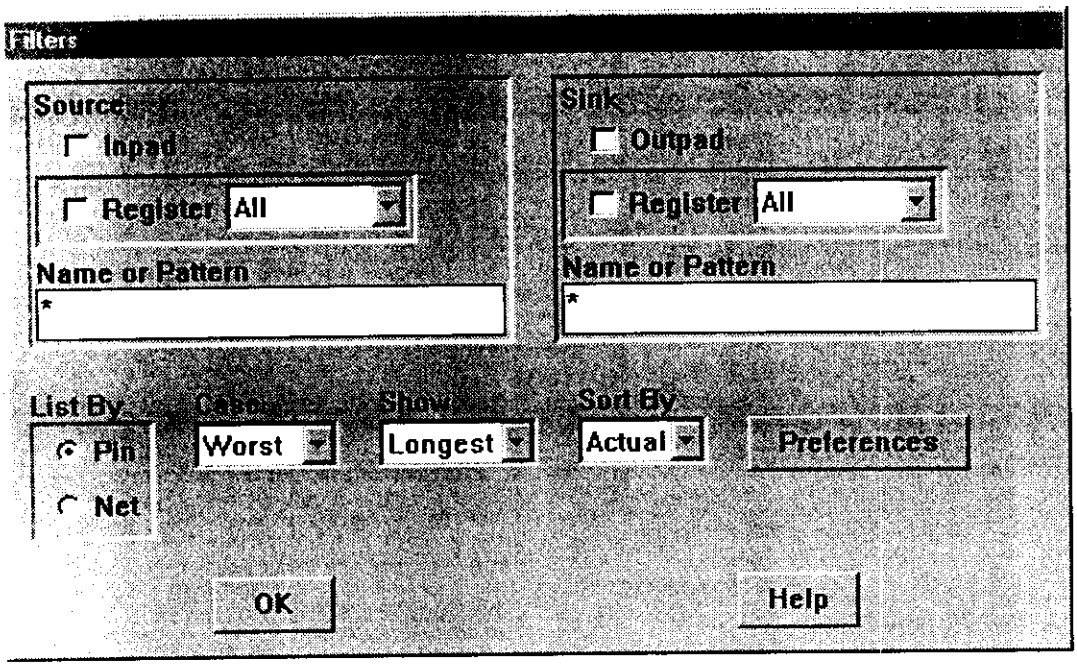


Figure 9. The DT Analyze Filters Window

2. On the **Source** side, click the **Register** box. Select **wrt_clk_1_net** from the adjacent pull- down. On the **Sink** side, click the **Register** button and select **wrt_clk_1_net** from the adjacent pull-down. Leave all the other buttons at their default setting.	This will set-up the DT Analyze tool to display only the register-to-register delays for registers driven by wrt_clk
3. Click the **Preferences** button.	The Preferences window will open similar to the one shown in Figure 10.

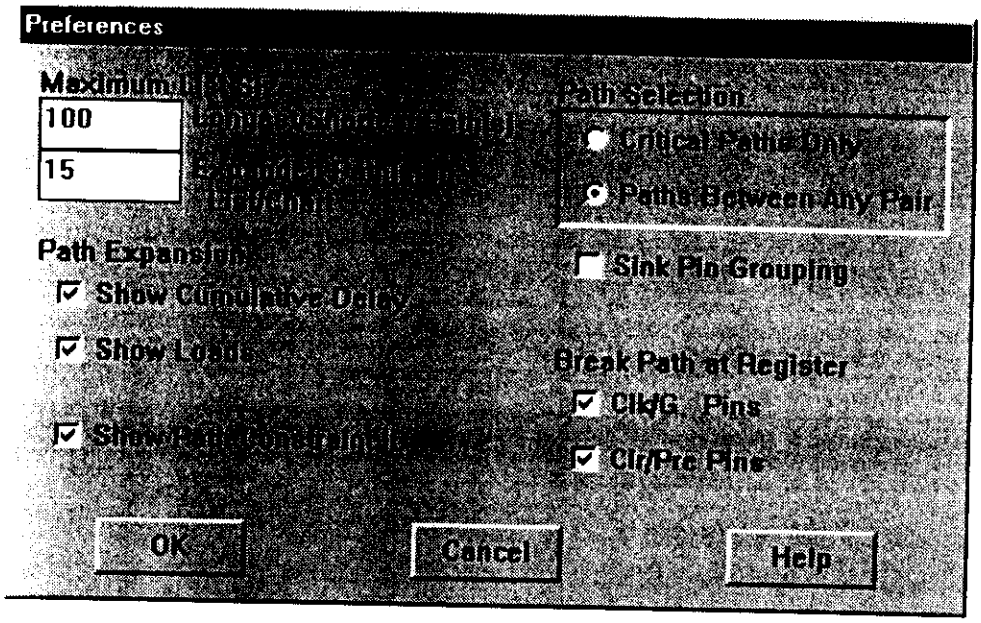


Figure 10. The DT Analyze Preferences Window

4. In the Preferences window enter:

   10 for Longest/Shortest Paths
   5 for Expanded Paths

   Leave all the other settings at their default value and click **OK.**

5. In the Filters window, click **OK.**

   The DT Analyze tool will list the specified register to register delays.

6. From DesignView, open the **dmac** schematic. Use the Start and End points shown in the DT Analyzer results to trace the paths on the schematic.

   If you have any questions about your results, call the instructor over.

7. In the DT Analyzer window, select **Options > Filters.**

   The Filters window will open again.

8. This time set the Source side to **Register - rd_clk_1_net** and the Sink side to **Register- rd_clk_1_net.**

   Click **OK.**

   The DT Analyze tool will list the specified register to register delays similar to those shown in Figure 11.

   NOTE: Your delays may vary from those shown.

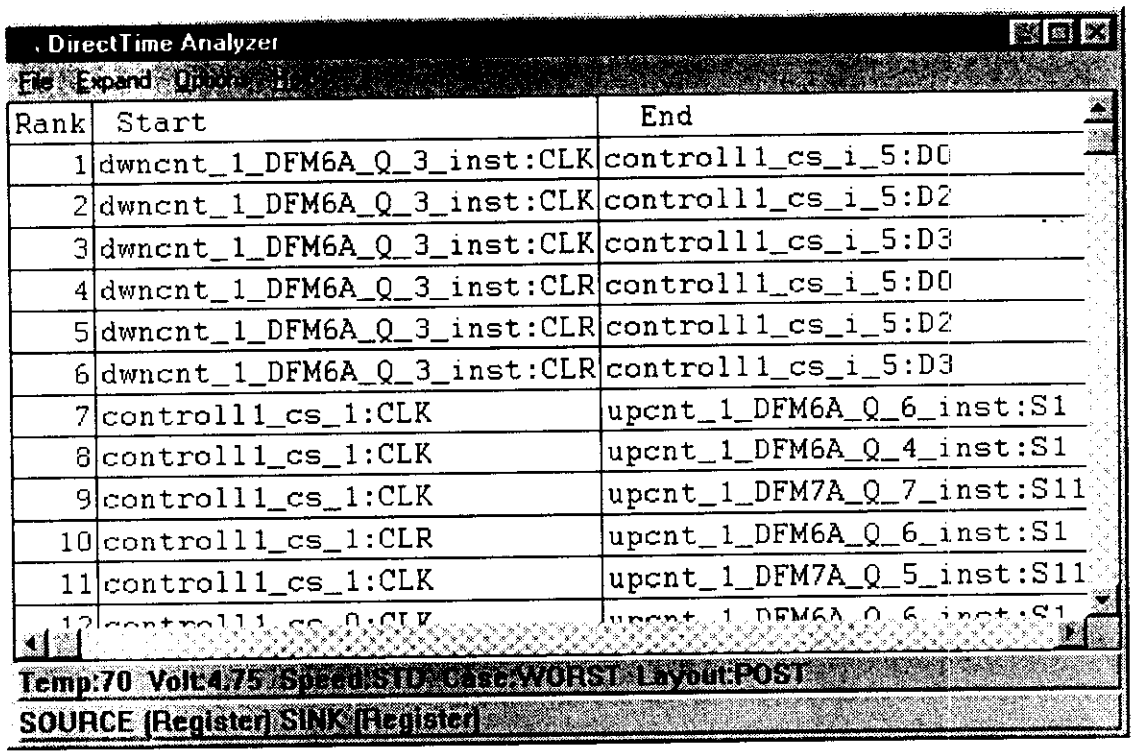


Figure 11. The DT Analyzer Results for rd_clk Register to Register Delays

9. From the DT Analyze results, click on the first **dwncnt_1** entry so it highlights.

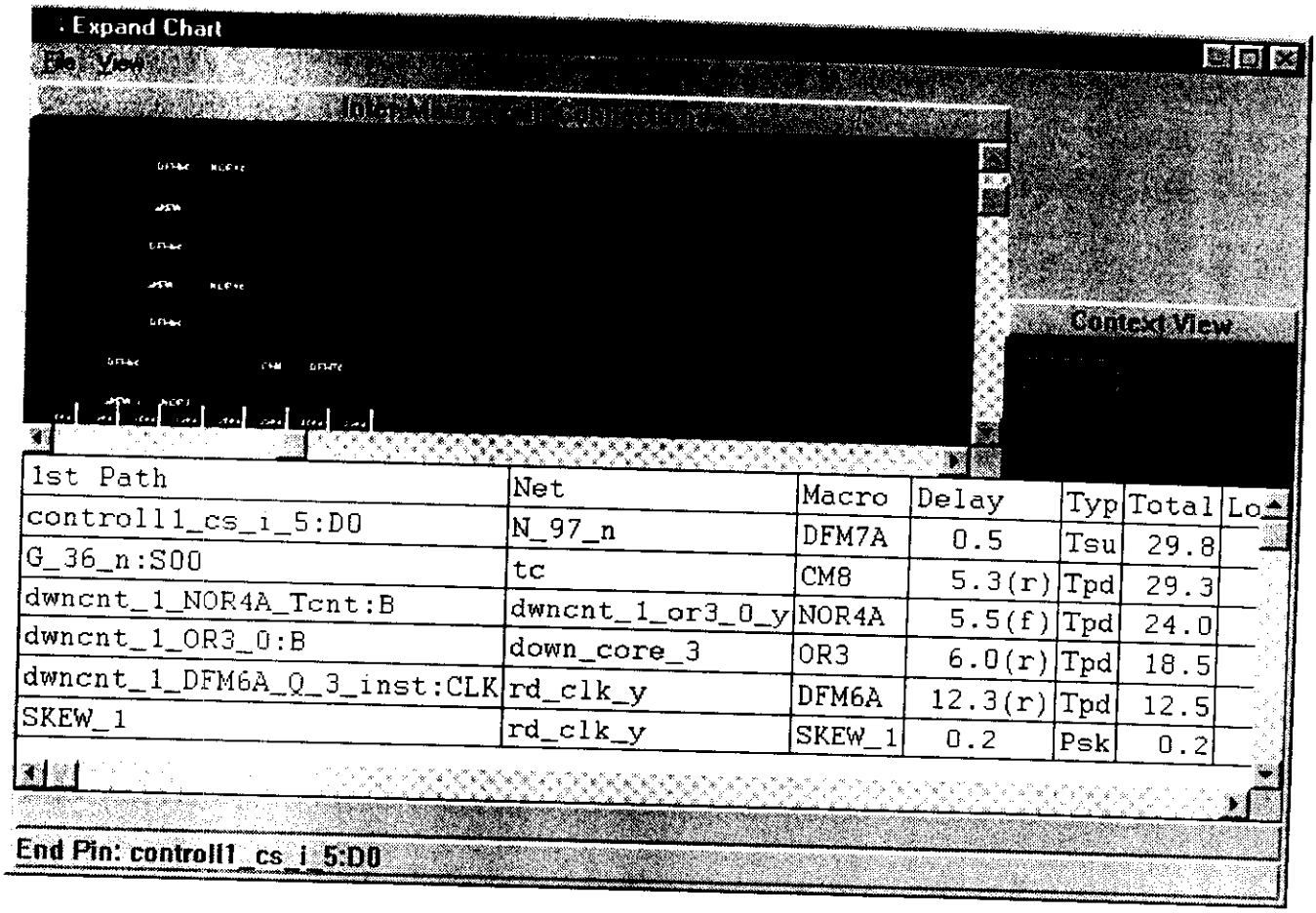
   From the main menu, select **Expand > Chart**.

   An Expand Chart window will open as shown in Figure 12. This window has two areas:

   A graphical display showing ALL paths from the endpoint in question to a various startpoints.

   A tabular display with an expanded listing of each path.

   NOTE: Your data may vary slightly from that shown.



1st Path	Net	Macro	Delay	Typ	Total	Lo
control1_cs_i_5:D0	N_97_n	DFM7A	0.5	Tsu	29.8	
G_36_n:S00	tc	CM8	5.3(r)	Tpd	29.3	
dwncnt_1_NOR4A_Tcnt:B	dwncnt_1_or3_0_y	NOR4A	5.5(f)	Tpd	24.0	
dwncnt_1_OR3_0:B	down_core_3	OR3	6.0(r)	Tpd	18.5	
dwncnt_1_DFM6A_Q_3_inst:CLK	rd_clk_y	DFM6A	12.3(r)	Tpd	12.5	
SKEW_1	rd_clk_y	SKEW_1	0.2	Psk	0.2	

Figure 12. The Expand Chart Display for the 1st Path

10. Read through the DT Analyze expansion from bottom to top and follow the signal through DWNCNT and CONTROLL.

    If you have any questions about reading this expansion ask the instructor.

11. In the tabular display headings, click the text: **1st Path.**

    What happens?

12. In the tabular display headings, click the text: **2nd Path** (you may need to scroll to the right to find it).

    What happens?

13. Back in the DT Analyzer window, select **File > Close**

Lab 10

Applying Constraints with DT Edit

Tasks

In this lab, the student will:

1. invoke DT Edit.
2. use the Clock Constraints Editor to apply clock constraints and designate clock exceptions.
3. use the Path Constraints Editor to apply timing constraints on individual paths and the clock exception paths.
4. practice exporting and importing timing constraint files.
5. generate "slack" reports.
6. change the speed grade.

Setting Clock Constraints

TASK	RESULT
1. If it is not still open, invoke the Designer and open the **dmac.adb** database.	
2. Click the **DT Edit** icon.	After a few seconds, the DirectTime Edit window will open, displaying the Clock Constraints editor.
3. Display the pull-down menu under **Global Signal**. What signals do you see and why are they listed? _____ _____ _____ _____	
4. From the list, select **rd_clk_1_net** and enter a period of **33ns** and a **50%** duty cycle.	This sets a constraint of 33ns on the register-to- register delay for all sequential elements driven by rd_clk.

Setting Constraints on Other Paths

TASK	RESULT
1. In the **Path Constraints** area of the DT Edit window, click the **New** button.	A blank Path Constraints Editor will open.
2. In the lower left hand corner, designate **Inpad** as the source. In the lower right hand side, designate **Register** as the sink.	
3. In the filter box on the Source side, set the filter to **R*** and hit return	Only the source pins that start with "R" will be displayed as shown in Figure 13.

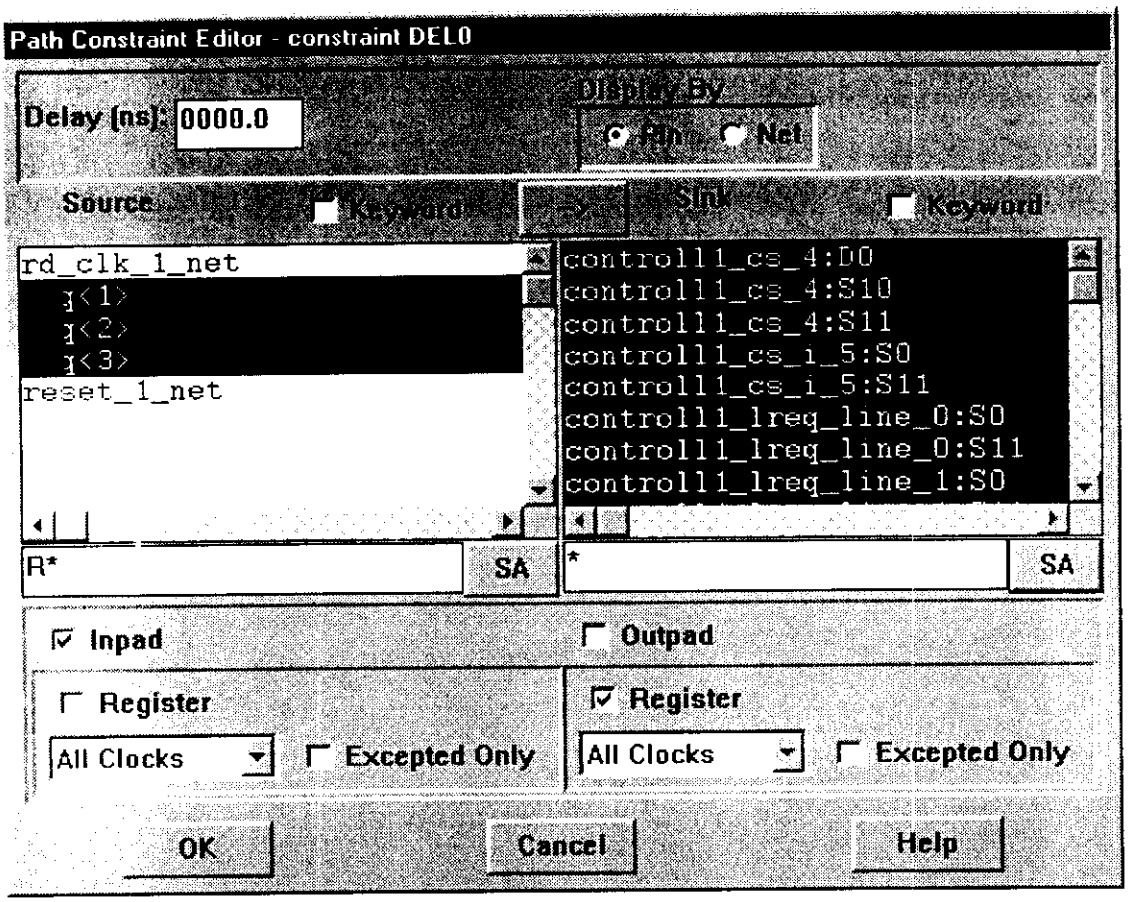


Figure 13. The Path Constraint Editor

4. Use the Shift key to select the **req<1>, req<2>** and **req<3>** pins.	All three pins should be highlighted as shown in Figure 13. When a source net is selected, all its register endpoints will appear in the Sink list.

5. From the **Sink** list, click the **SA** button (**Select All**).

   This will highlight all the end points. At this point, you have selected the on-chip delays for the **req** inputs.

6. In the **Delay** field at the top, type **25**.

   Click **OK**.

   You have now specified that the on-chip delay for the **req** signals be 25ns or less.

   The Path Constraint Editor will close and you will see your path constraint listed in the Path Constraints area of the DT Editor window.

   Notice also that next to the title Path Constraints there is now a 1 of 1 count displayed.

   When this constraint is displayed, you can use the Edit or Delete buttons to change or remove it.

7. Use this same procedure to set a 20ns constraint on the on-chip delay from all the **addr** inputs to their subsequent register inputs.

8. From the main menu of the DT Edit window, select **File > Commit** and then **File > Close.**

9. Save the database.

Exporting and Importing a Delay Constraints File

The Designer allows you to export your delay constraints to a file so you can investigate more than one set of constraints without overwriting your old ones. In this section, you will practice exporting and importing your delay constraints file.

TASK	RESULT
1. From the Designer's main menu, select **File > Export**.	An Export dialog box will open.
2. In the **File Type** pull-down, select **Auxiliary File.**	
3. In the **File Name** field, type **dmac_1.dcf** In the Parameters area, make sure that the **Type** is **Delay Constraints**.	

4. Click **OK**.	This will save your current delay constraints to a file so that it can be imported later. Using this feature, you can set up multiple constraint scenarios and easily switch from one to another.
5. To practice importing, go to the Designer's main menu and select **File > Import > Auxiliary File**	An Import dialog box will open.
6. Verify that the **File Type** is **Delay Constraint** and use the **Browse** button to find and select the file name **dmac_1.dcf** Click **OK**	Assuming this was a different file, it would overwrite any constraints that were currently in the database.
7. Save the database.	

Generating Slack Reports and Changing the Speed Grade

TASK	RESULT
1. In the Designer's main menu, select **Reports > Timing** and specify **Pre-layout**	The Timing Report dialog box will open.
2. In the Timing Report dialog box, set **Sort By** to **Slack.** Notice that the Slack Threshold is 0.0. If we are meeting all our timing constraints, what will this slack report look like? _____	
3. Click **OK**	A timing report will appear displaying all the negative slack values (if any exist).
4. Close the report and go back to the Designer. From its main menu, select **Options > Device Setup Wizard**. In the top of the Operating Conditions dialog box, set the **Speed Grade** to **-1**.	
5. Regenerate your slack timing report. Do the prelayout timing estimates meet the constraints? _____	
6. Save the database.	

Lab 11

Laying Out DMAC

Tasks

In this lab, the student will:

1. perform a Standard Mode layout.
2. use Timing Reports to view any post-layout timing failures.
3. perform a DirectTime Mode layout.
4. change the DMAC schematic slightly and perform an Incremental layout.

Performing a Standard Mode Layout

TASK	RESULT
1. From the Designer's flow chart, click the **Layout** icon.	A dialog box will open, prompting you to select the mode of layout.
2. Select **Standard** and verify that Incremental is **OFF**. Click **OK**.	The layout will begin. It will report an estimate of the time required to complete the layout.
3. What length of time is estimated for Standard Mode layout? _____ Generate a timing report. Were the timing requirements met?	

Performing a DirectTime Mode Layout

TASK	RESULT
1. From the Designer's flow chart, click the **Layout** icon.	A dialog box will open, prompting you to select the mode of layout.
2. Select **DirectTime** and verify that **Incremental** is **OFF**. Click **OK**.	The layout will begin. It will report an estimate of the time required to complete the layout.

3. Which layout took longer?_____

   Were the timing requirements met? _____

   If not, change the speed grade.

4. Save the database and exit the Designer.

Changing the DMAC Schematic and Running Incremental Layout

TASK	RESULT
1. In the **Design Hierarchy** tab of DesignView, open the **dmac** schematic and go to page 2.	
2. Use the Bus icon to add a bus and hierarchical output connector to the Q[7:0] output of the DWNCNT block as shown in Figure 14. Be sure to name the hierarchical connector **pin** as down[7:0] as shown.	

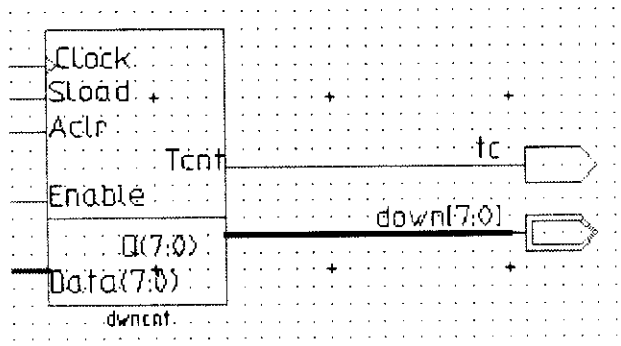


Figure 14. Additional OUTBUF8 block for DMAC.2

3. Save the schematic and verify it ( )
4. Close the schematic.
5. From the Design Hierarchy tab, re-synthesize dmac.

   Hint (Highlight and right-click to select **Open in Synthesis**)
6. Exit from the Synplify.

7. From the **Vendor Manager** tab, highlight ***dmac.adb*** and right-mouse click to select **Open**.

   The Designer will be invoked and the *dmac.adb* database will open.

8. What do you notice about the flow chart?

   The Designer recognizes that the EDIF netlist has been updated. Because of this, your compile and layout steps are invalidated and will no longer be highlighted green.

   _____

   _____

   _____

9. Click **Layout** and make the following settings:

   Mode - Standard
   Incremental - FIX

   Before the layout is begun, notice that the database automatically recompiles.

Lab 12

VeriBest Backannotated Simulation

Tasks

In this lab, the student will:

1. Extract the actual timing delays for a given set of operating conditions.
2. Use the extracted delay information to perform a post-layout simulation

Extracting and Exporting Timing Delays

TASK	RESULT
1. From the Designer's flow chart, click **Extract**.	The Extract window will open as shown in Figure 15. The **Status** area of this window displays the operating conditions for which the delays will be extracted. To extract a different set of conditions, change the conditions first with the **Options > Device Setup Wizard** menu.

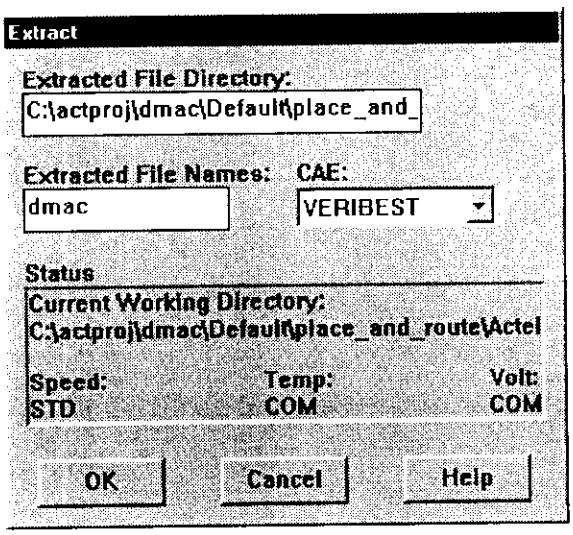


Figure 15. The Extract Window

2. Verify that the CAE pull-down is set to **VERIBEST**.

   Click **OK**.

   The delays will be extracted and exported to a file.

   What is the name of the extracted delay file and where is it placed?

   _____

3. Save the database.

   Exit from the Designer.

4. In the Vendor Manager tab of DesignView, look under the **Place-Route Files** directory.

   You should see a new *dmac.sdf* file.

Using the dmac.sdf File to Perform Post-Layout Simulation

To perform post-layout simulation, you must work with the *dmac.vhd* file that resides in the **Place-Route Files** directory of the **Vendor Manager** tab listing.

TASK	RESULT
1. Before we can simulate a .vhd file, we must associate stimulus. From the **Vendor Manager** tab in DesignView, go to the **Place-Route Files** directory, highlight ***dmac.vhd*** and associate it to the same *dmac_stim.vhd* stimulus file that we used for previous simulations.	
2. Once a stimulus file has been associated, highlight ***dmac.vhd*** again and right click your mouse to select **Open Post-Layout Simulator**.	This simulator will open as we have seen before.
3. Perform a 2000ns simulation and display the waveforms for all I/Os.	Hint: Compile all modules Execute the simulator Open WaveBench and add the I/Os Set the simulation time to 2000ns. Click the Run button.

You will receive several assertion warnings of the form:

```
Assertion [WARNING] in  . . .

VitalGlitch: GLITCH Detected on port Y ; Preempted Future Value :=
0 @ 864.5 ns; Newly Scheduled Value := 1 @ 866.2 ns;
```

These glitch warnings are normal. You can ignore them.

4. Verify that your results match the pre-layout simulation results shown on your hard copy.

5. Exit from the simulator.

   Notice in the Vendor Manager tab, that a new file has been added to the Place-Route Files directory: *PostRoute.vpd*

   What is the purpose of this file?

   _____

   _____

   _____

Lab 13

Using Activators to Program an Actel FPGA

Tasks

In this lab, the student will:

1. Specify a silicon signature for an Actel FPGA design.
2. Invoke the Windows Programming software and open an existing database.
3. Perform a blank check on a 32100DX chip provided by the instructor.
4. Program the 32100DX without security.

Creating the Fuse File and Silicon Signature

TASK	RESULT
1. Open the dmac.adb database.	
2. In the Designer flow chart, click **Fuse**.	The Export dialog box will open as shown in Figure 16.

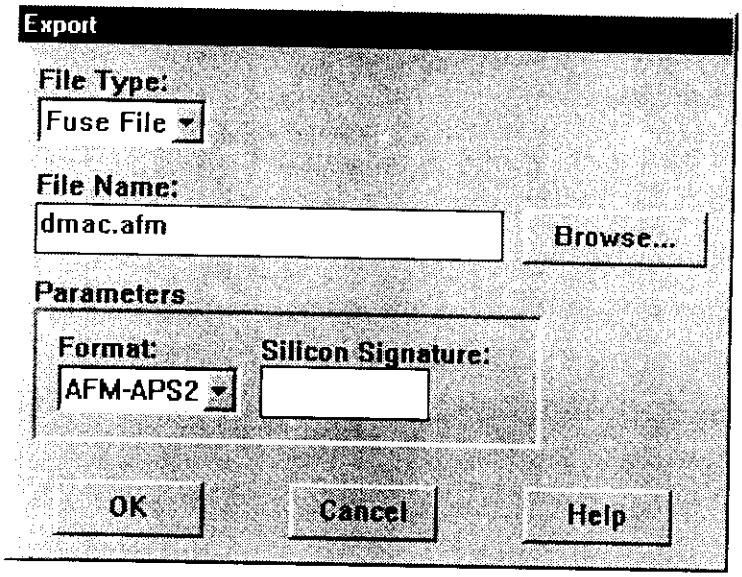


Figure 16. The Export Dialog Box

3. All the defaults should be correct, except that we want to enter a silicon signature.

   In the **Silicon Signature** field, enter some 5 digit hexidecimal.

   Click **OK**.

4. Save your design database, then exit from the Designer.

5. In the **Vendor Manger** tab of DesignView, notice that a new *dmac.afm* file has been added to the Place-Route Files directory.

   This is the file that will be opened to start the programmer.

Programming an FPGA

TASK	RESULT
1. Get a 32100DX FPGA from the instructor. Line it up properly and latch it into your Activator 2 or 2S.	
2. From the **Vendor Manager** tab's Place-Route Files directory, highlight ***dmac.afm*** and right mouse click to select **Open**.	The Windows APS window will appear.
3. After the database has opened, click the **Blankcheck** button. If your chip is not blank, see the instructor.	The inserted chip will be scanned to determine whether or not it has been programmed.
4. Click the **Activate** button.	
5. Verify that only Array Fuses are set to be programmed. Click **OK**.	Device programming will begin.
6. When programming has completed, again perform a Blankcheck. Does the silicon signature match the value you set when you ran FUSE? Congratulations! You just programmed an Actel FPGA!	

Lab 14

Running Actel's Debugger - INCOMPLETE!!!

Tasks

In this lab, the student will:

1. invoke Actel's Debugger.
2. run Debugger commands interactively.
3. load and run a provided Debugger command file.
4. create and run their own Debugger command file.

Invoking the Debugger

NOTE: The Debugger can only be run on a programmed chip which is clamped in the Activator.

TASK	RESULT
1. If it is not already running, invoke the Windows Programming tool by selecting and opening the dmac.afm file.	
2. From the Windows APS main menu, click the **Debugger** button.	The Debugger will be initialized.

Running a Debug Session

TASK	RESULT
1. In the **Command** field at the top of the Windows APS window, type **loadfile "initial.deb"** Look at your hard copy of the *initial.deb* file and make sure you understand all the commands. If you have any questions, call the instructor.	The initial.deb file will load and run. You will see the following message: [illegible] PRINTING BY PRINTING [illegible] ADATA=ZZZZ

2. Now that the required vectors and macros (CYCLEMAIN and CYCLE) have been loaded, you will use interactive commands to test the serial to parallel shift function.

   In the **Command** field, enter each of the debugger commands shown below. Remember that each time you CYCLE, the signals in the TABLIST will be printed.

   emit "TESTING S2P REGISTER\n"

   tablist reset srst

   emit "RESET CIRCUIT\n"

   L reset

   assign 0b1111 ctl

   cycle

   H reset

   cycle

   tablist serial p ydata

   emit "CYCLING 1s ON SERIAL INPUT\n"

   H serial

   repeat 6 (cycle)

   At the end of this series of commands, you should see that YDATA = 1111.

3. In the **Command** field, load the *adder.deb* file.

   Look at your hard copy of the *adder.deb* file and make sure you understand all the commands. If you have any questions about the results, call the instructor.

4. Load and run the upcntr.*deb* file.

   Look at your hard copy of the *upcntr.deb* file and make sure you understand all the commands. If you have any questions about the results, call the instructor.

5. Use the Windows Explorer to copy the *upcntr.deb* to a file named *downcntr.deb*.

   Modify *downcntr.deb* so that it tests the

down counter (you may find the ACE truth table in your schematic handout to be useful).

6. Load and run your new file in the Debugger until it works successfully.

Debugger Command Files

INITIAL.DEB

```
(vector ctl ctl3 ctl2 ctl1 ctl0)
(vector adata adata3 adata2 adata1 adata0)
(vector bdata bdata3 bdata2 bdata1 bdata0)
(vector p p3 p2 p1 p0)
(vector up up3 up2 up1 up0)
(vector down down3 down2 down1 down0)
(vector ydata ydata3 ydata2 ydata1 ydata0)
(define (cyclemain) (L sysclk) (step) (H sysclk) (step) (print) (emit "\n"))
(define (cycle) (L ctlclk) (step) (H ctlclk) (step) (cyclemain))
(emit "TEST THE PRINTING BY PRINTING ADATA\n")
(tablist adata)
(print)
```

S2P.DEB

```
(emit "TESTING THE SERIAL TO PARALLEL SHIFT\n")
(tablist reset)
(emit "RESET THE CIRCUIT\n")
(L reset)
(assign 0b1111 ctl)
(cycle)
(H reset)
(cycle)
(tablist serial p ydata)
(emit "CYCLING IN 1s ON THE SERIAL LINE\n")
(H serial)
(repeat 6 (cycle))
(emit"DONE (YDATA should equal 1111)\n")
```

ADDER.DEB

```
(emit "TESTING THE ADDER\n")
(tablist reset)
(emit "RESET THE CIRCUIT\n")
(L reset serial)
(assign 0b0000 adata)
(assign 0b0000 bdata)
(assign 0b0011 ctl)
(cycle)
(H reset)
(cycle)
(tablist adata bdata ydata co)
(emit "ADD 1 + 1\n")
(assign 0b0001 adata)
(assign 0b0001 bdata)
(repeat 3 (cycle))
(emit "ADD 1 + 15\n")
(assign 0b1111 adata)
(repeat 3 (cycle))
(emit"DONE\n")
```

UPCNTR.DEB

```
(emit "TESTING THE UP COUNTER\n")
(tablist reset)
(emit "RESET THE CIRCUIT\n")
(L reset serial)
(assign 0b0000 adata)
(assign 0b0000 bdata)
(assign 0b0111 ctl)
(cycle)
(H reset)
(cycle)
(tablist bdata up ydata)
(emit "SET BDATA TO 8\n")
(assign 0b1000 bdata)
(repeat 2 (cycle))
(emit "DISABLE LOAD AND COUNT UP TO 11\n")
(L ctl1)
(repeat 4 (cycle))
(emit "DONE\n")
```

Chapter 1

Software Overview and Design Flow

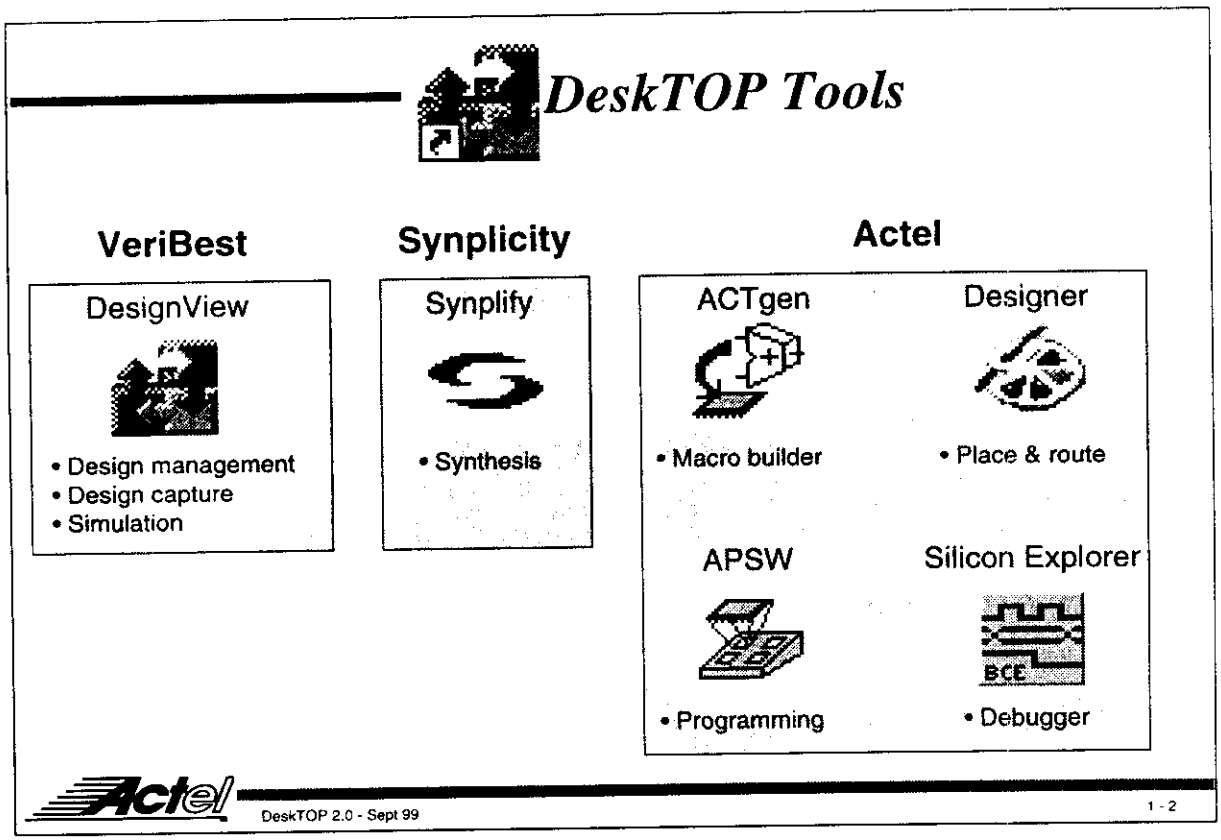
DeskTOP 2.0 - Sept 99

1 - 1

Objectives

After completing this chapter, you will be able to:

- List the three vendors that are combined in the Actel DeskTOP
- Explain how to obtain, install, license and get technical support for the DeskTOP software.
- Briefly describe the example design that will be used in class.
- Describe the purpose of VeriBest's DesignView software and explain the four different DesignView tabs.



Notes

The Actel DeskTOP is an FPGA design environment that tightly integrates the tools from three vendors: VeriBest, Synplicity and Actel.

The DeskTOP interface and design management is handled by VeriBest's DesignView tool. Through this tool, you can capture the design in many forms including schematic, VHDL, Verilog, state diagrams, flowcharts, truth tables and ACTgen blocks. Besides design capture and management, VeriBest also provides their VHDL simulator which can perform behavioral, structural and post-place & route simulations.

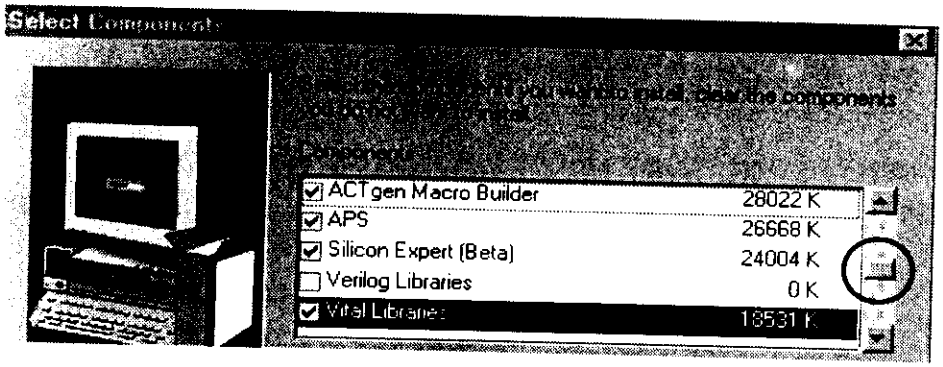
HDL synthesis is handled by Synplicity's Synplify tool. After synthesis, which creates an EDIF file, VeriBest automatically performs an EDIF-to-VHDL translation to facilitate post-synthesis simulation.

All the FPGA definition and layout is handled by Actel's Designer. Once the design has been successfully placed and routed, the Designer can export a delay file that can be used to perform post-place&route simulation.

Programming and debugging is also provided by Actel tools.

Obtaining & Installing

- **DeskTOP CD is free - order from the web**
  - **www.acteldesktop.com**
- **Don't forget to install the right Actel library.**



- **VeriBest and Synplicity tools can be installed and upgraded independently**

Notes

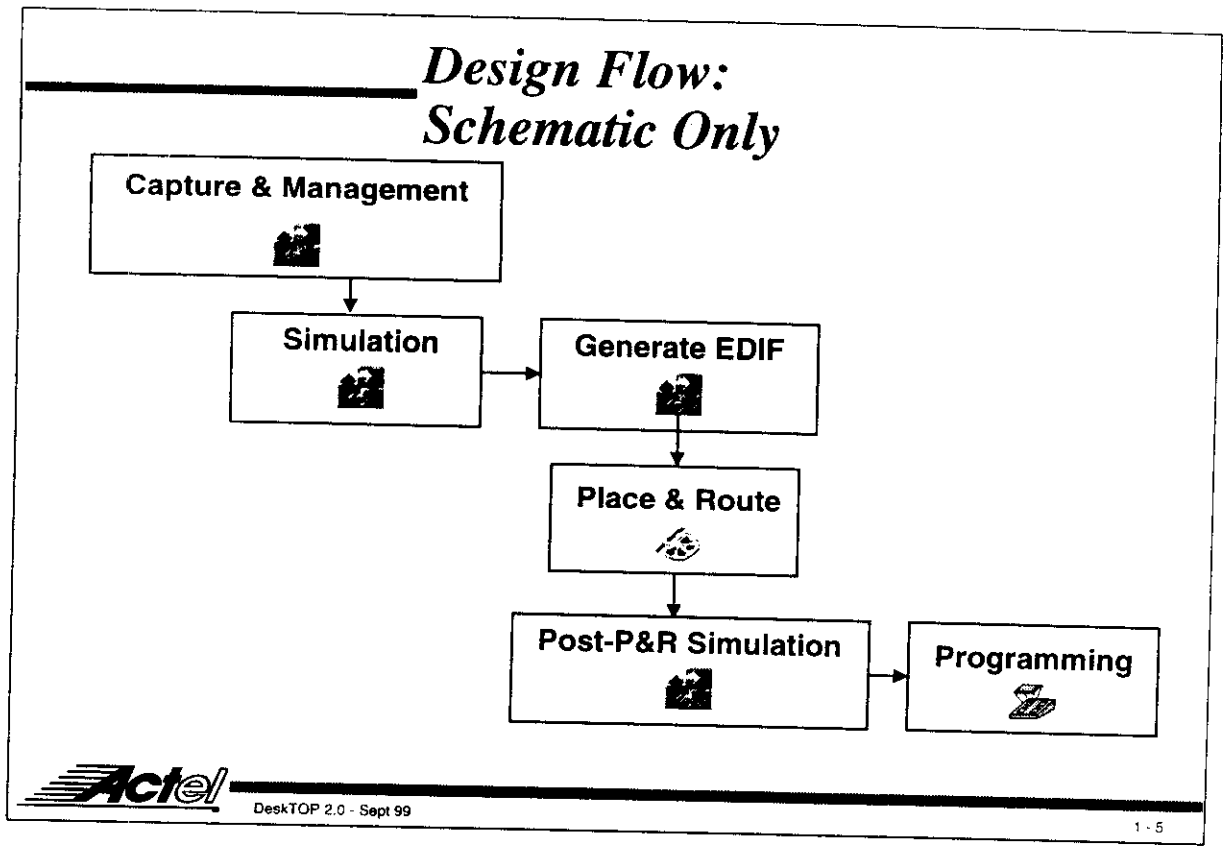
Licensing and Support

- **License is required**
  - order via email
  - reference must be added to ***autoexec.bat*** file or Systems window:

    SET LM_LICENSE_FILE=C:\flexlm\license.dat

- **Support is only available if you purchase maintenance**
  - $995 for one year.

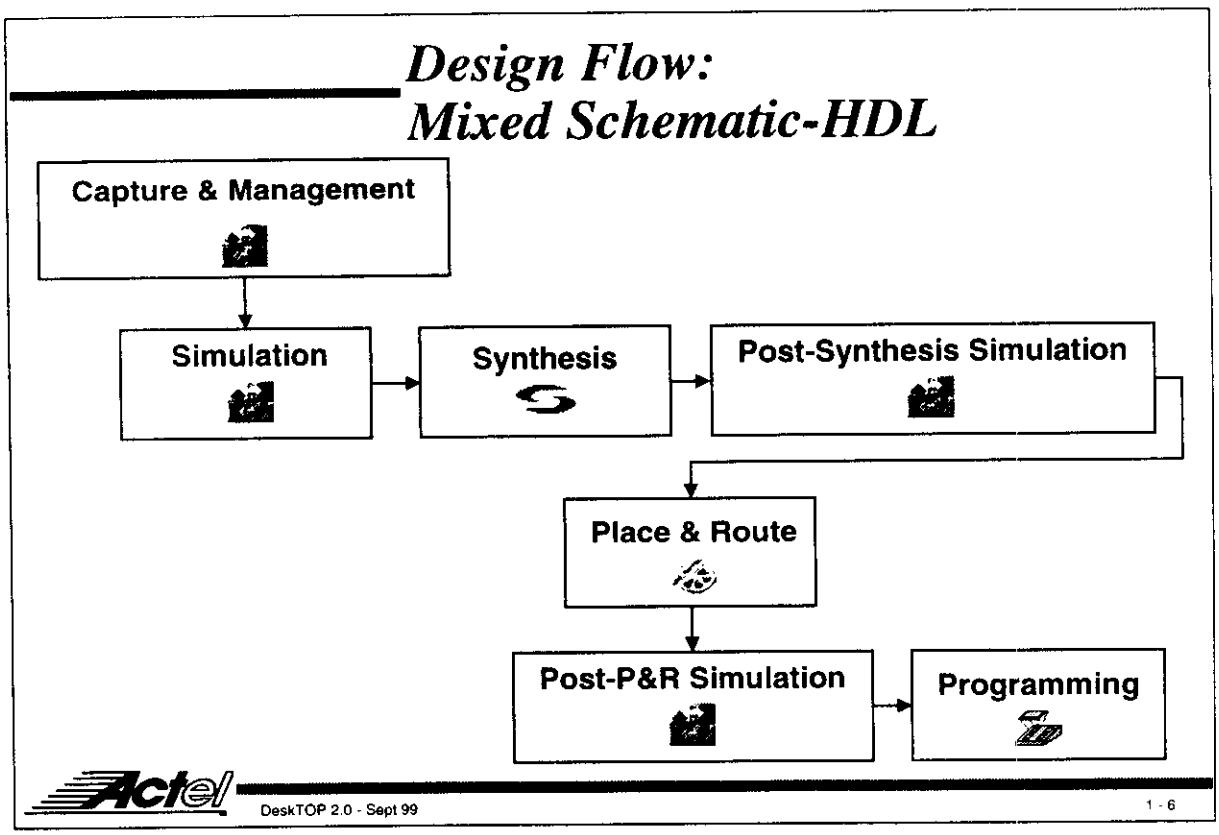
Notes



Notes

Schematic-only designs can still be simulated with the VeriBest VHDL Simulator. Upon invoking the simulator, VeriBest automatically translates the schematic netlist into a structural VHDL format.

When you are ready to move on to place & route, you will have to manually generate the EDIF netlist in DesignView.

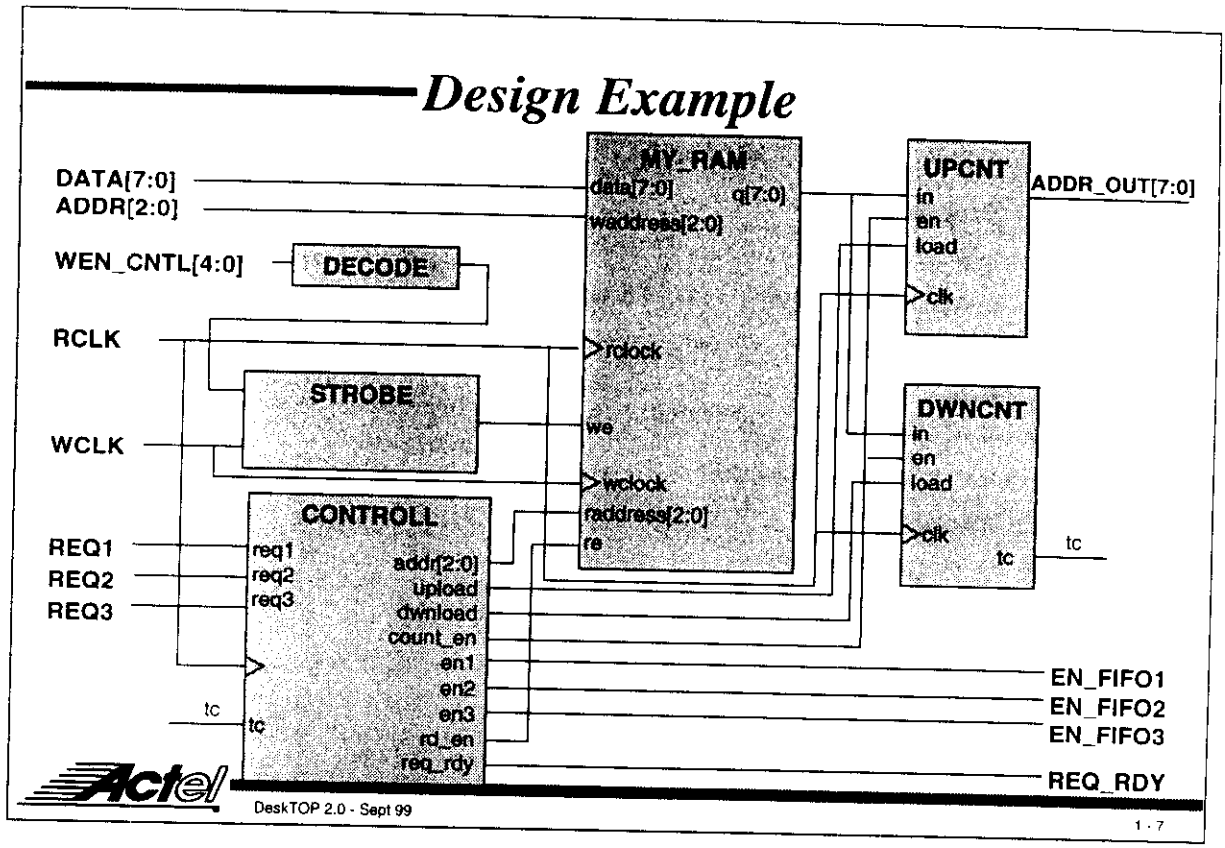


Notes

Notice that the presence of and VHDL blocks (including those generated by ACTgen) adds two new steps to the recommended design flow:

- Synthesis
- Post-Synthesis Simulation

The result of synthesis with Synplify is an EDIF netlist so the Generate EDIF step is inherent and does not need to be run manually.



Notes

The functional block diagram of the DMAC design example is shown above.

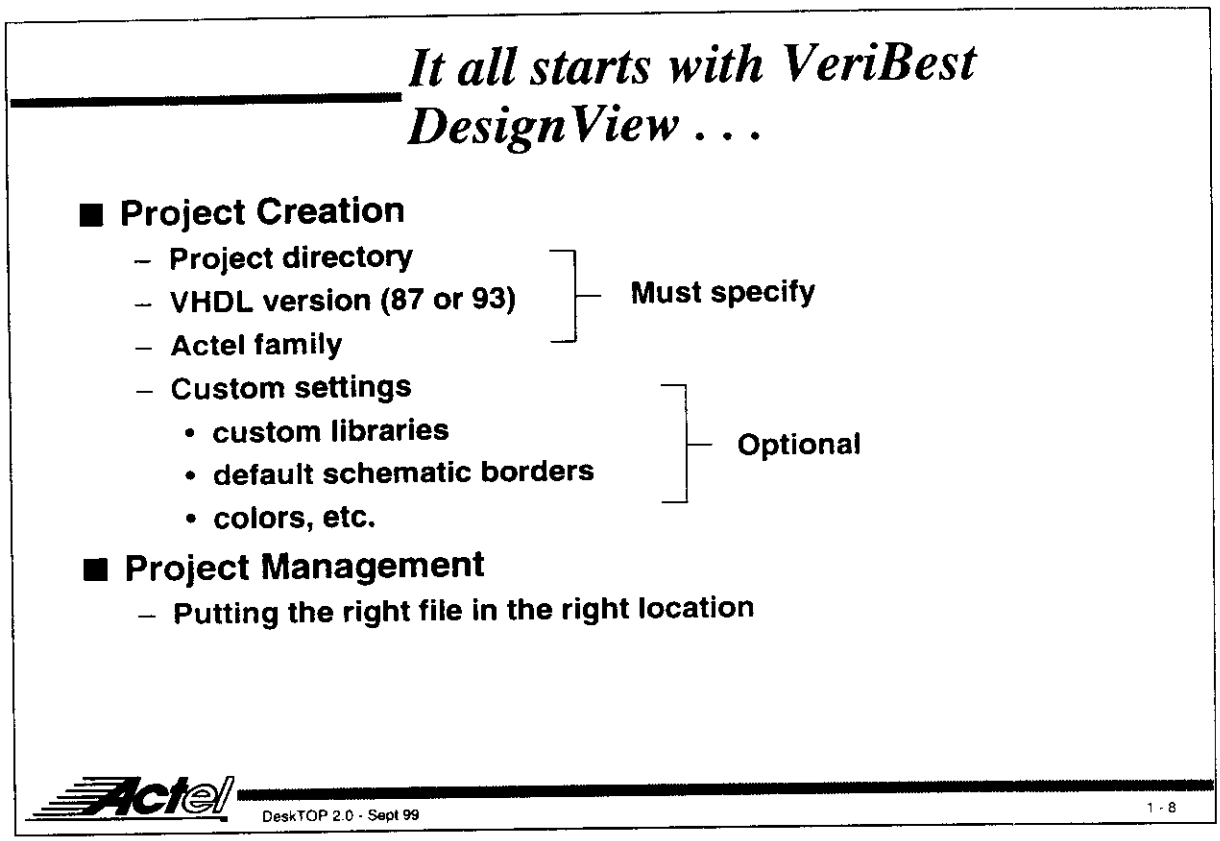
The purpose of DMAC is to control the filling of three external FIFOs. Which FIFO is enabled depends on which one is requested (see the REQ1, REQ2, and REQ 3 signals).

The starting address for the appropriate FIFO is read from MY_RAM and is then incremented by the ADDR_OUT[7:0] output of UPCNT. The data (which also comes from some external source) is shifted in until the TC signal from DWNCNT is asserted.

Therefore, the data contained in MY_RAM are pairs of starting addresses and shift lengths.

The CONTROLL block is a state machine that selects the appropriate pair from MY_RAM based on the request signals (REQ1, 2, and 3). The CONTROLL block also loads the counters and enables both the counters and external FIFOs.

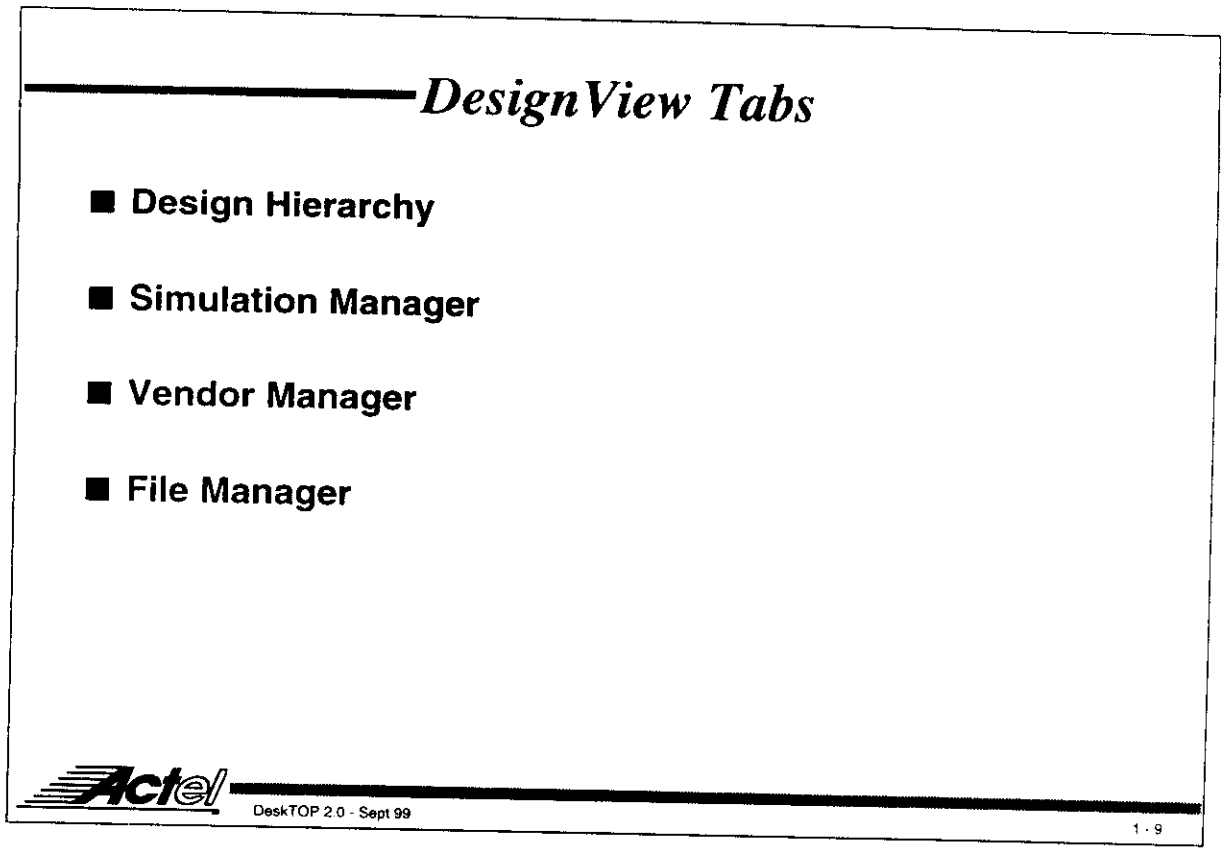
See your schematic handout for a detailed diagram of DMAC and all its hierarchical blocks.



Notes

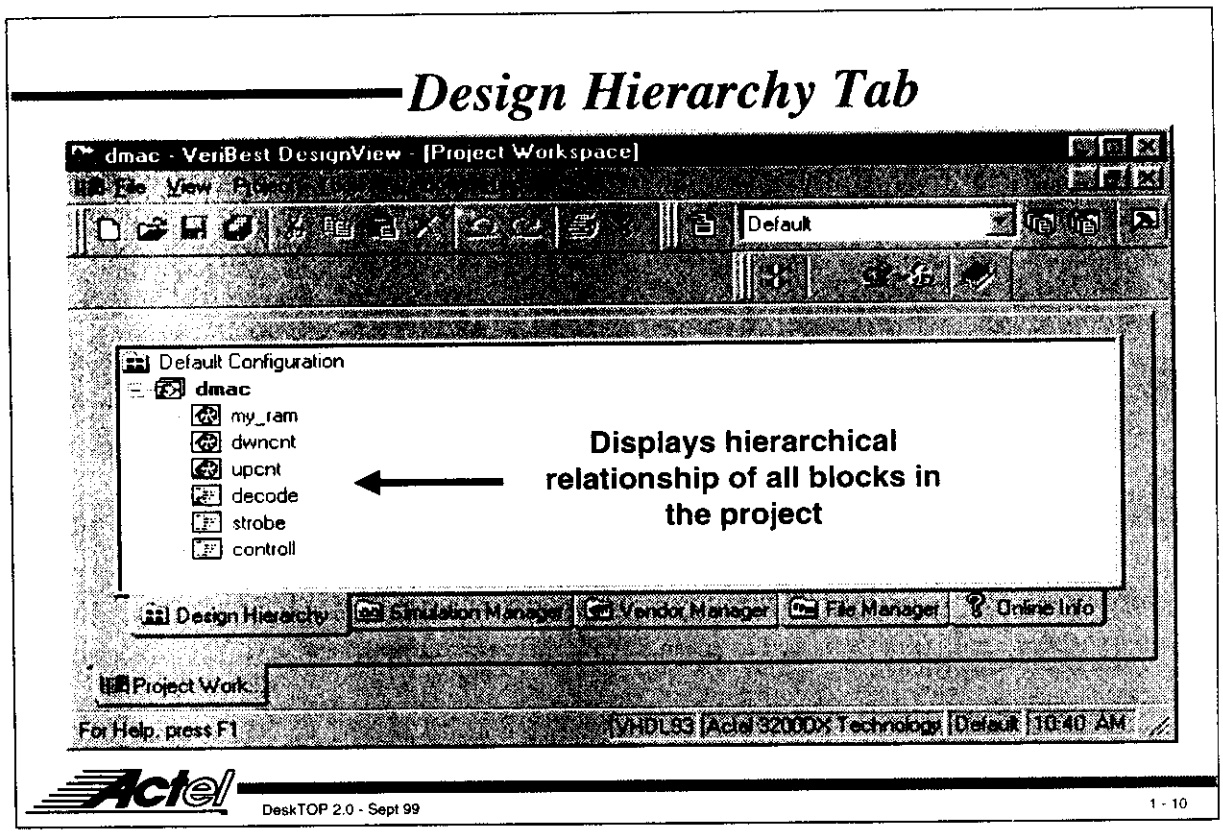
The first step (after installation and licensing), is to create a VeriBest project. VeriBest makes this easy with a step-by-step user interface that prompts you for the information above. The new project is saved to a file with a *.prj* extension.

Once the project is created, you can open it and select **Project > Settings** to make further customizations.



Notes

There are four tabs in the DesignView Window. The next 4 slides will explain how the project data is arranged in each tab.



Notes

The **Design Hierarchy** tab lists all the blocks in the project, arranged by hierarchy. From the listing above, you can see that the **dmac** schematic includes 6 lower-level blocks: **my_ram, dwncnt, upcnt, decode, strobe and controll.**

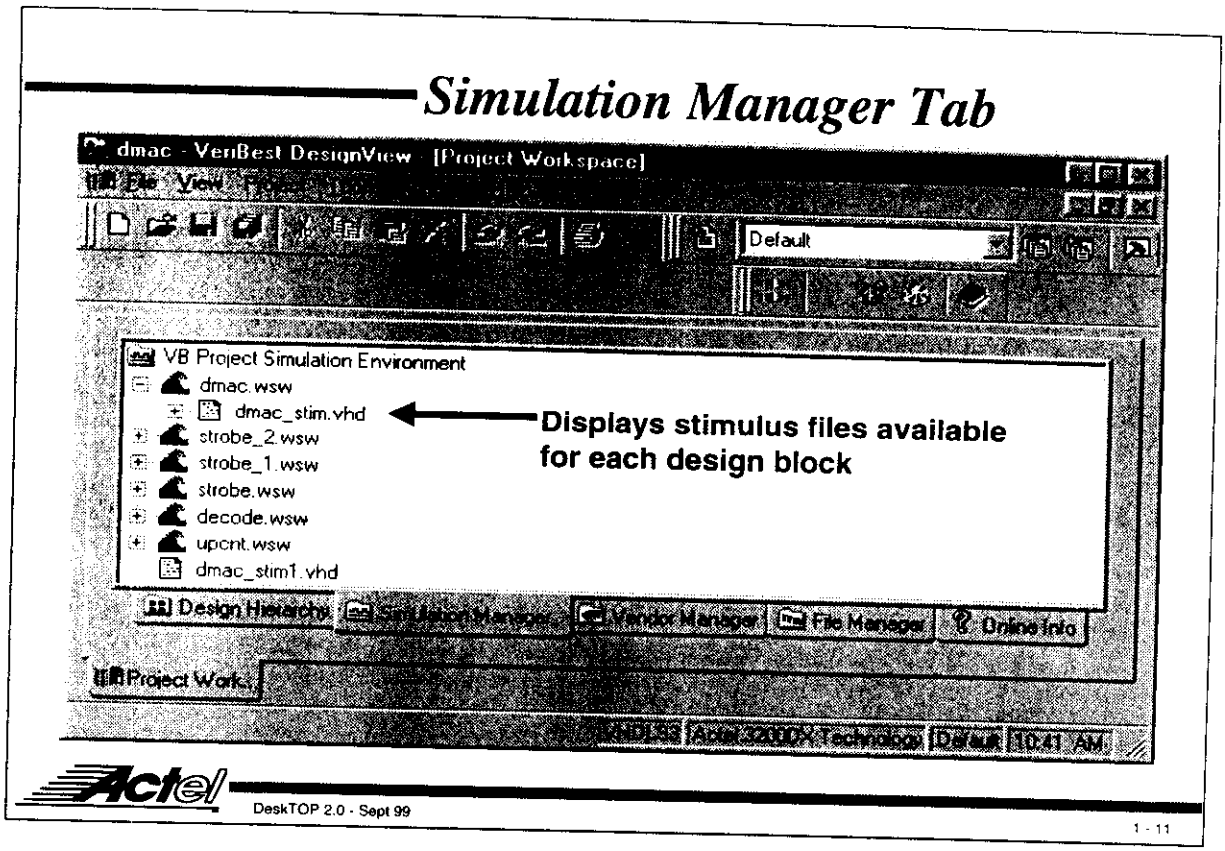
Notice that the icon for each block indicates its type:


is a schematic block


is an ACTgen block (that is, created by Actel's ACTgen)


is a VHDL block

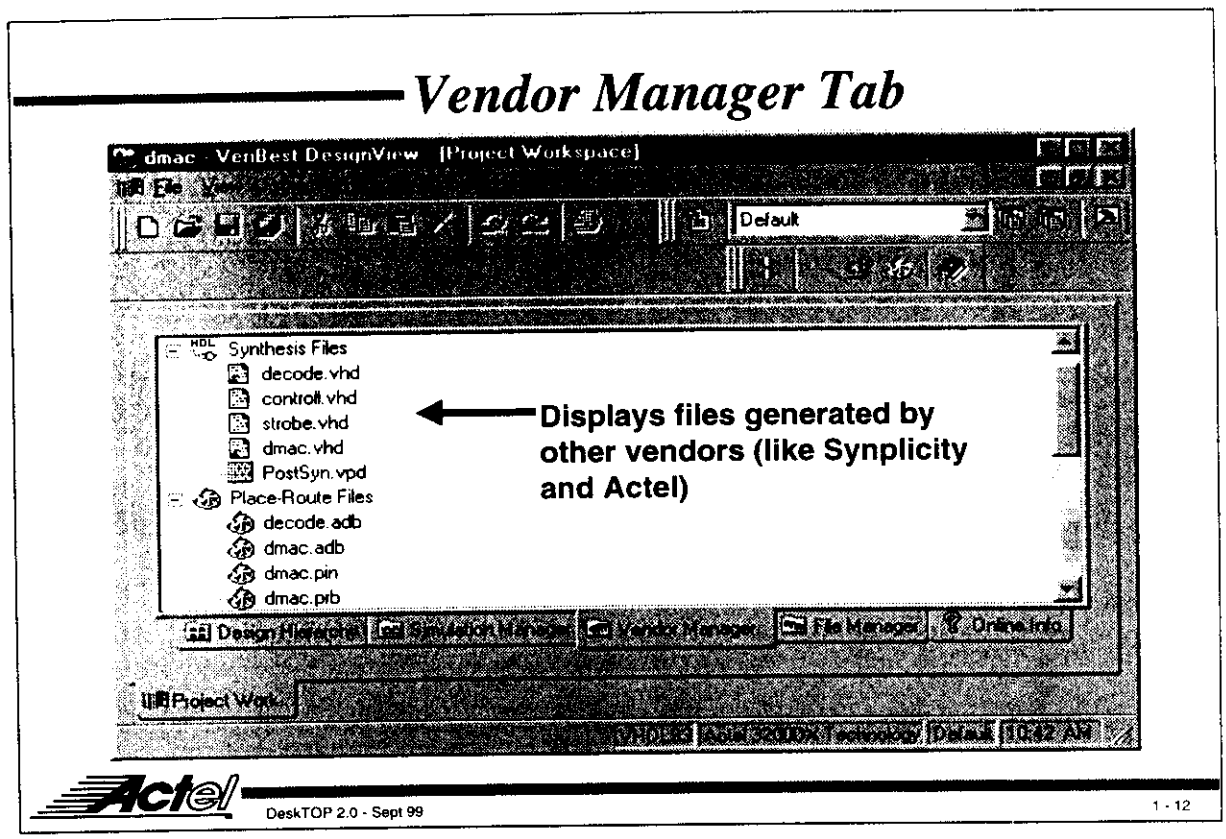
There are also icons for state diagram blocks, flowcharts and truth tables.



Notes

The **Simulation Manager** tab displays all the stimulus files that belong to the project. Each stimulus file is displayed "under" the *.wsw* directory for its corresponding block.

For example, the DMAC block has a stimulus directory called *dmac.wsw*. Within that directory is a stimulus file called *dmac_stim.vhd*

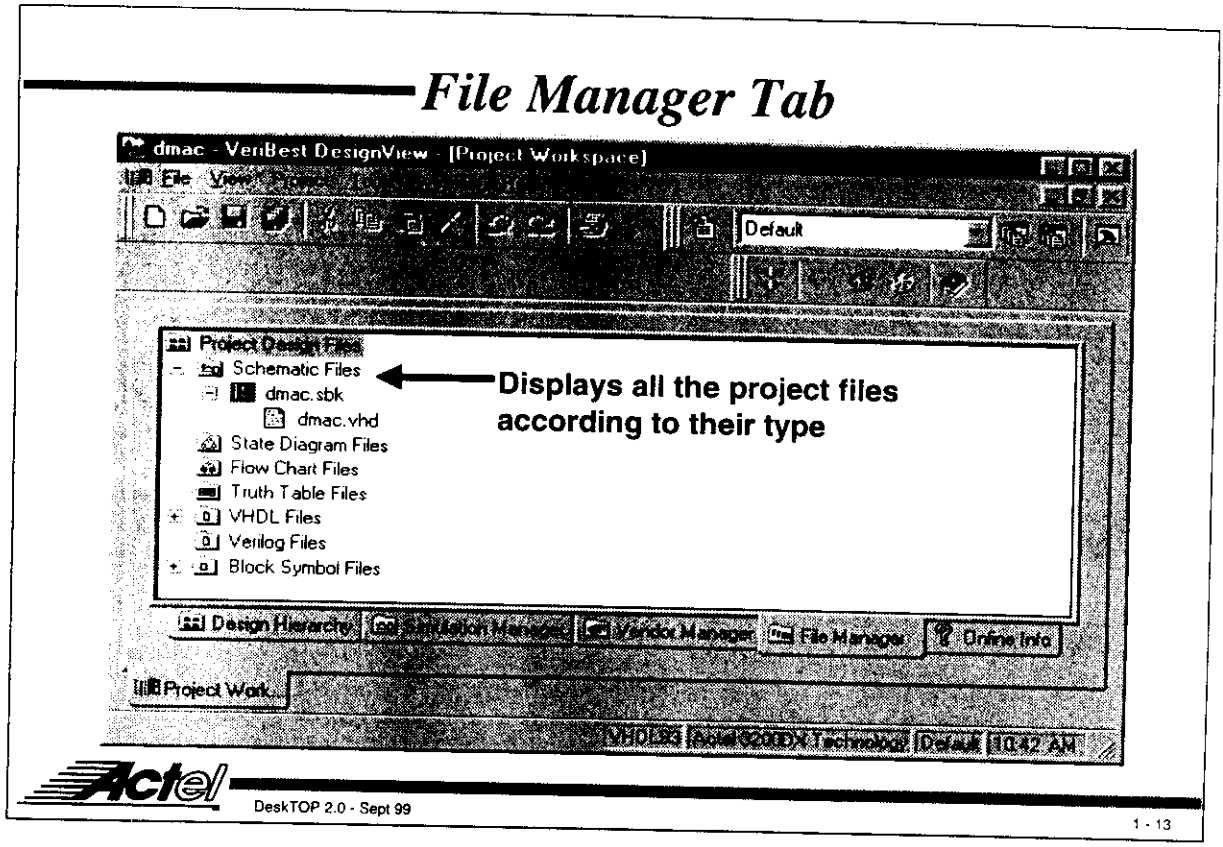


Notes

The **Vendor Manager** tab displays all the files generated by or for other vendors. In the case of the Actel DeskTOP, the other vendors are Synplicity and Actel.

When synthesizing with Synplify, two files are generated: an EDIF file and a structural VHDL file. The VHDL file is place in the **Synthesis Files** directory and can be used to perform post-synthesis simulation. Since the EDIF file is used by Actel, it is placed the **Place-Route Files** directory.

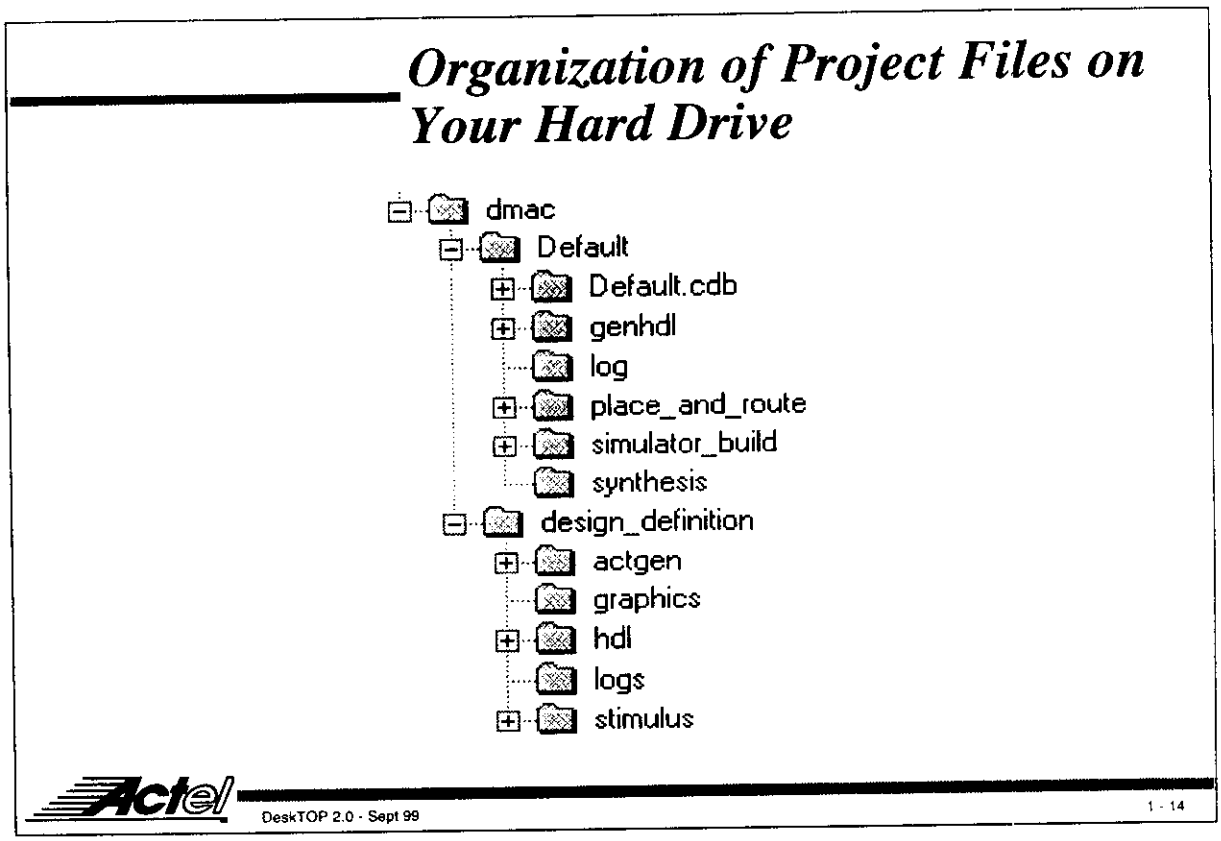
In addition, all files generated by Actel are placed in the **Place-Route Files** directory.



Notes

As shown above, the **File Manager** tab organizes the project files based on their type. It is important to keep in mind that this is NOT the way the files are arranged on your hard disk. This is not a Windows File Manager.

The project files arrangement on your hard disk is shown on the next page.



Notes

The slide above shows how a project's files are organized on your hard drive. Most of the directory names are self-explanatory with the exception of **graphics** which contains all the schematics. Schematics have an *.sbk* extension for "schematic block".

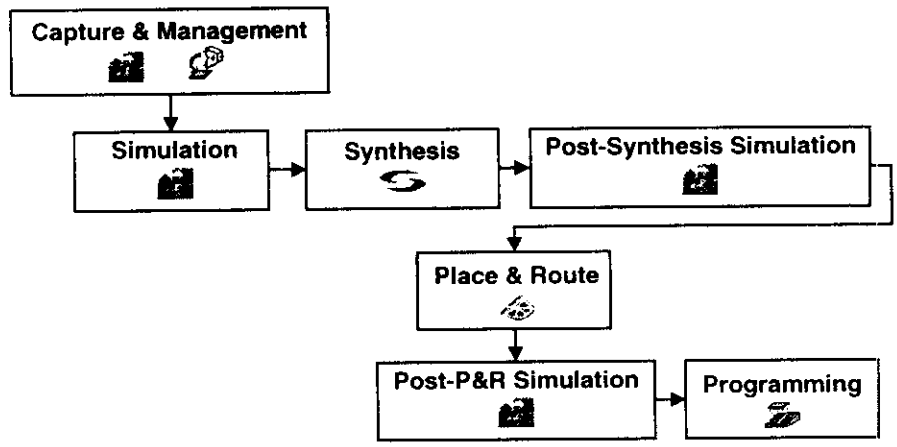
It is SERIOUSLY SUGGESTED that you never manipulate, edit, move, copy, etc. any of your project files using any tool other than DesignView.

Introduction to Lab

- **Complete the Lab Setup**
  - Create a new project

Chapter 2

Block Generation with ACTgen and VHDL



Objectives

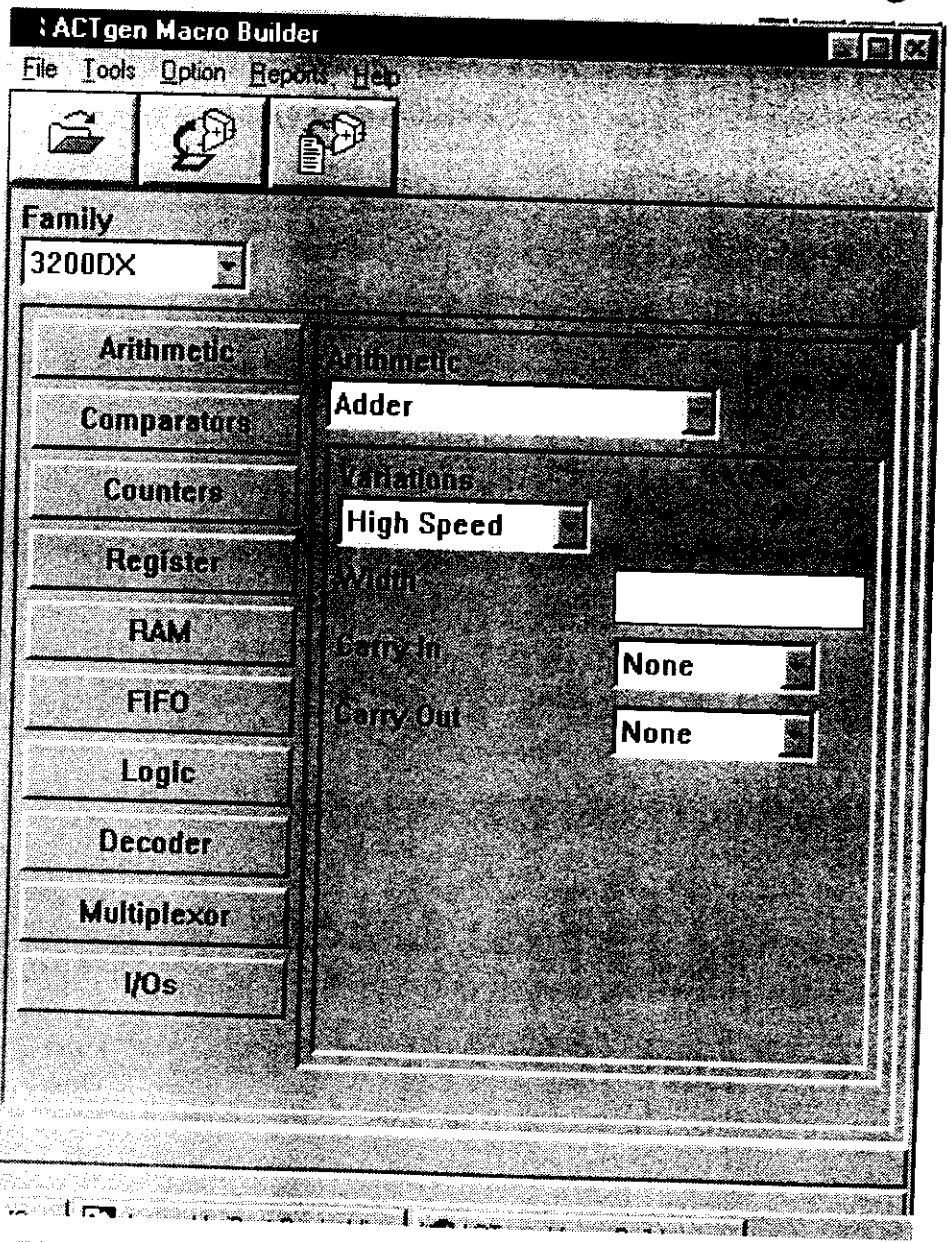
At the end of this chapter, you will be able to:

- Describe the purpose of ACTgen.
- Invoke ACTgen.
- List the types of macros that ACTgen can generate.
- List the different output formats of ACTgen and describe circumstances for using each one.
- Create new VHDL blocks in a project.
- Add existing VHDL blocks to a project.

The ACTgen User Interface

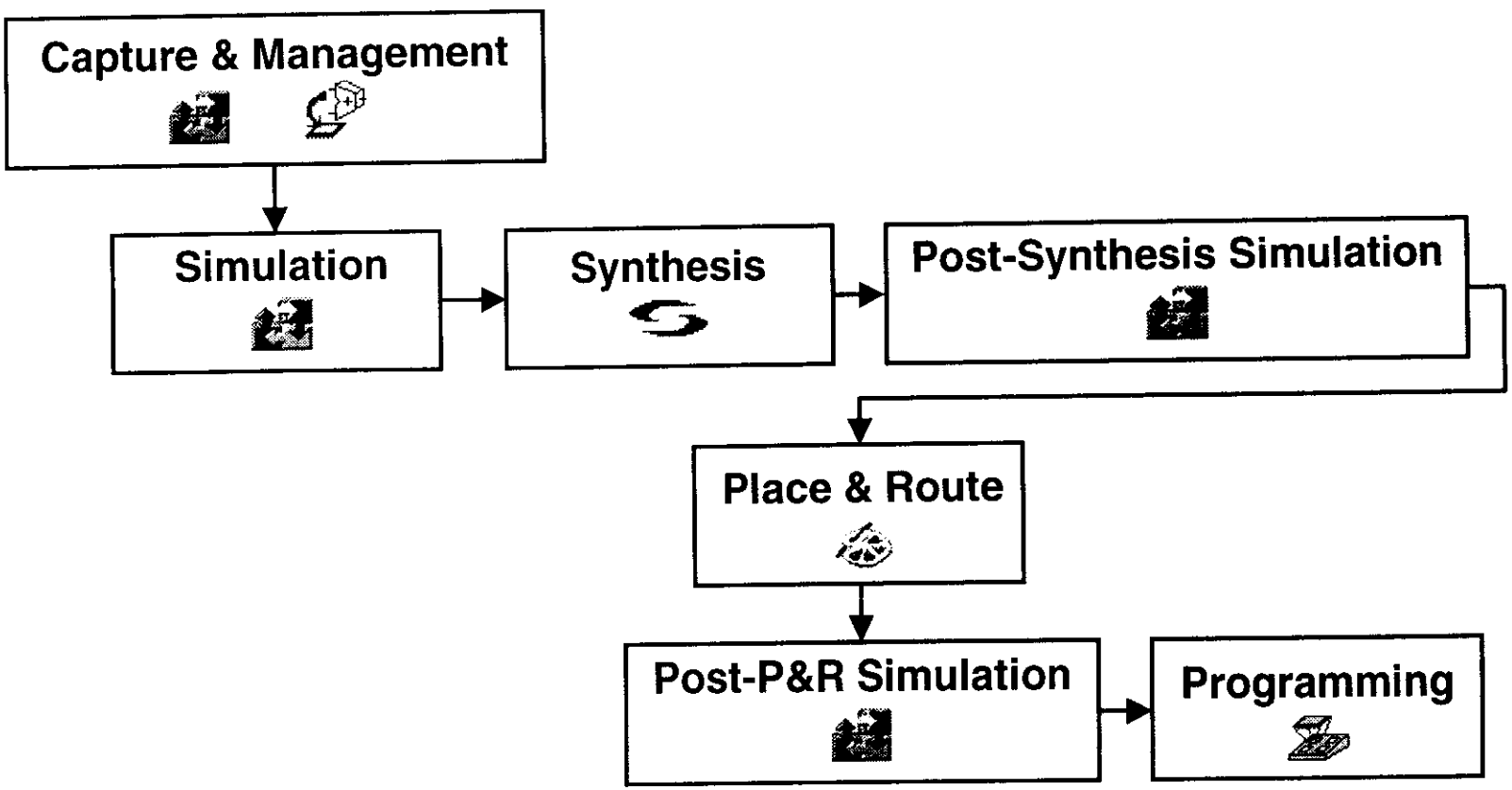
Select the family

Select the macro type



Chapter 2

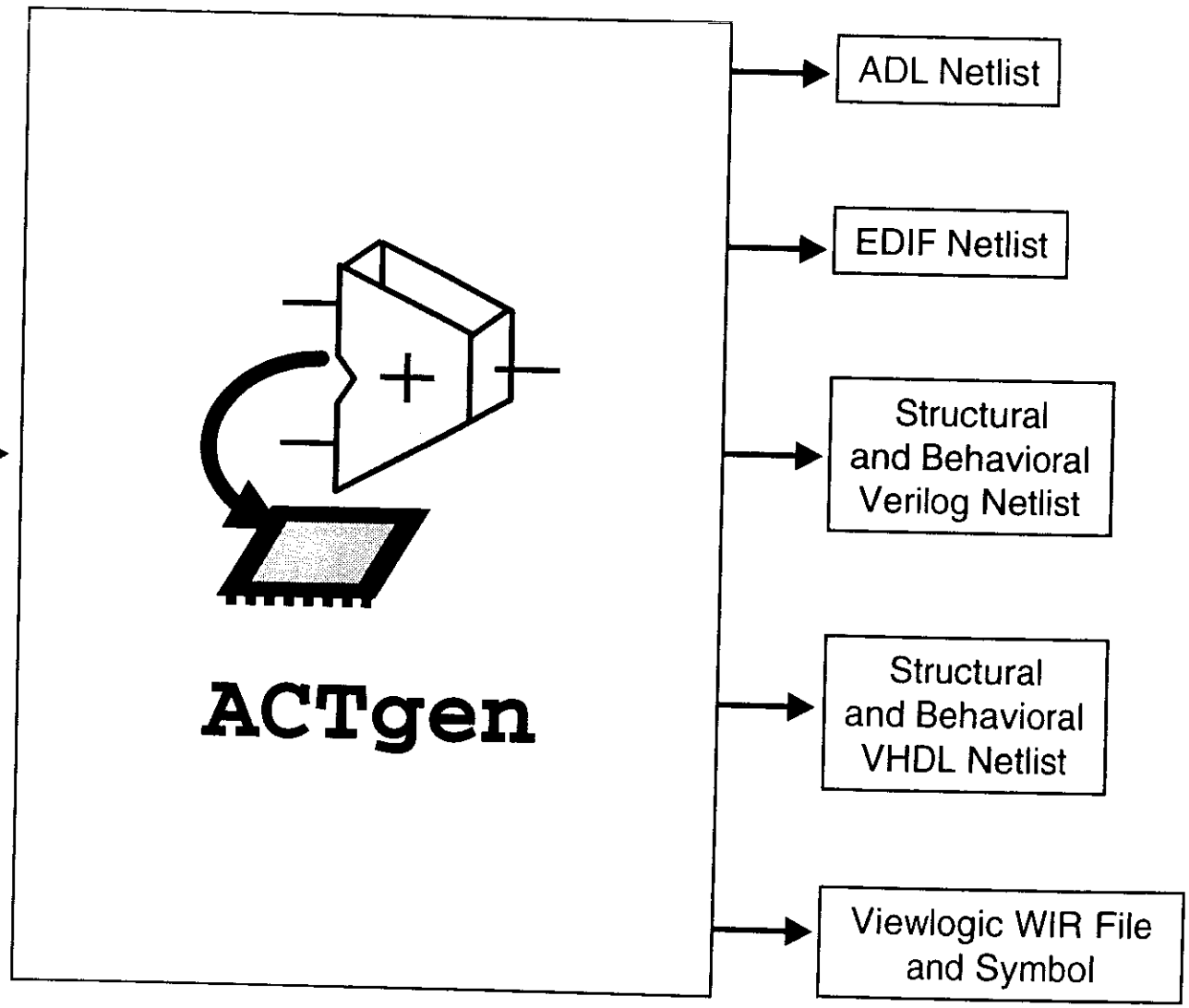
Block Generation with ACTgen and VHDL



ACTgen Overview

Fill-in-the-blank description for:

- Counter
- Adder
- Subtractor
- Accumulator
- Register
- Comparator
- I/O Blocks
- Logic Blocks
- Decoder
- Multiplexor
- RAMS
- FIFOS



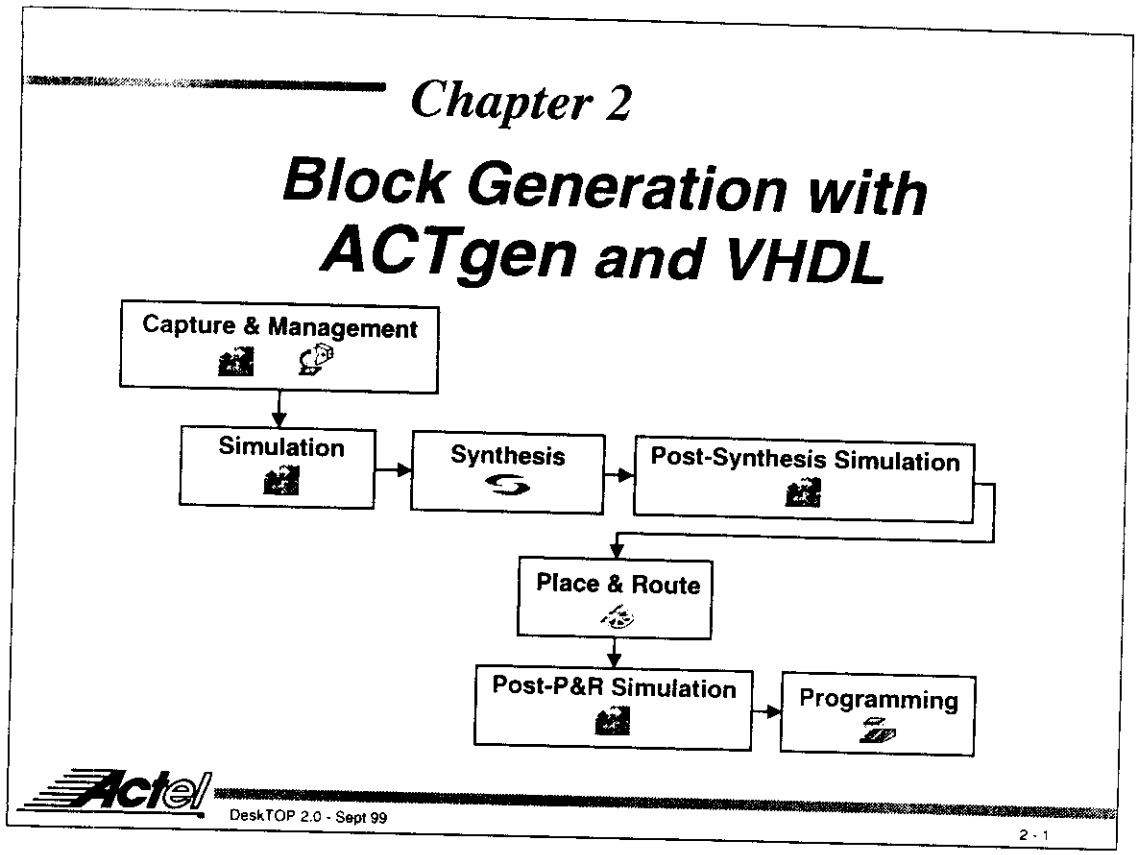
Chapter 1

Software Overview and Design Flow

Objectives

After completing this chapter, you will be able to:

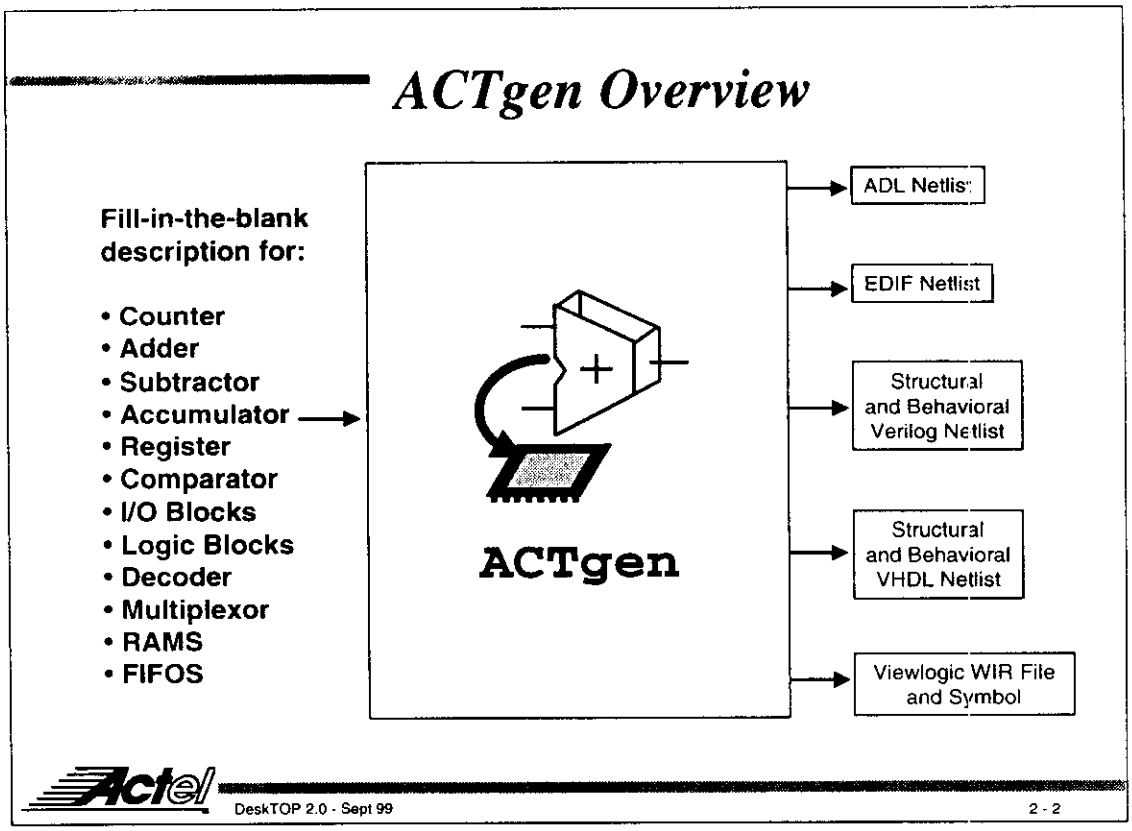
- List the three vendors that are combined in the Actel DeskTOP
- Explain how to obtain, install, license and get technical support for the DeskTOP software.
- Briefly describe the example design that will be used in class.
- Describe the purpose of VeriBest's DesignView software and explain the four different DesignView tabs.



Objectives

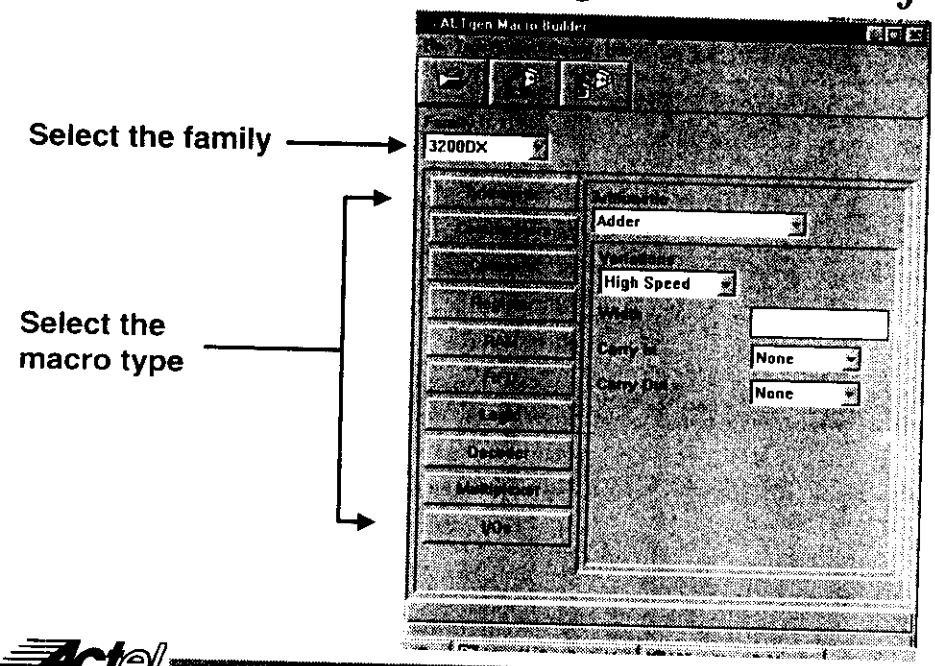
At the end of this chapter, you will be able to:

- Describe the purpose of ACTgen.
- Invoke ACTgen.
- List the types of macros that ACTgen can generate.
- List the different output formats of ACTgen and describe circumstances for using each one.
- Create new VHDL blocks in a project.
- Add existing VHDL blocks to a project.


ACTgen Overview
Fill-in-the-blank description for:
• Counter
• Adder
• Subtractor
• Accumulator
• Register
• Comparator
• I/O Blocks
• Logic Blocks
• Decoder
• Multiplexor
• RAMS
• FIFOS
ACTgen
ADL Netlist
EDIF Netlist
Structural and Behavioral Verilog Netlist
Structural and Behavioral VHDL Netlist
Viewlogic WIR File and Symbol
Actel
DeskTOP 2.0 - Sept 99
2 - 2

Notes

The ACTgen User Interface

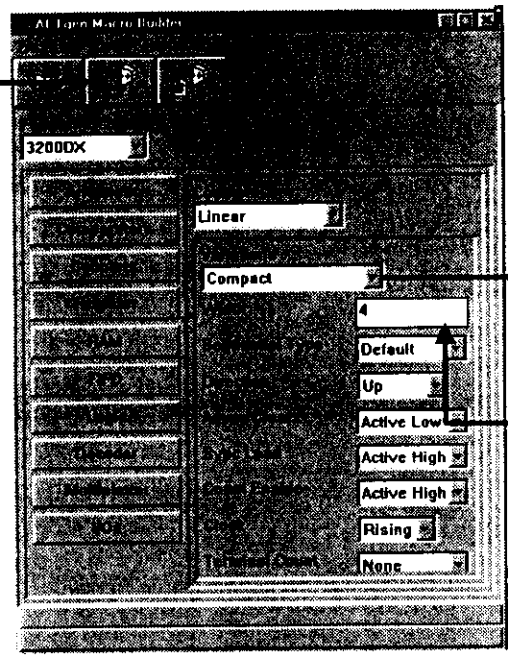


DeskTOP 2.0 - Sept 99

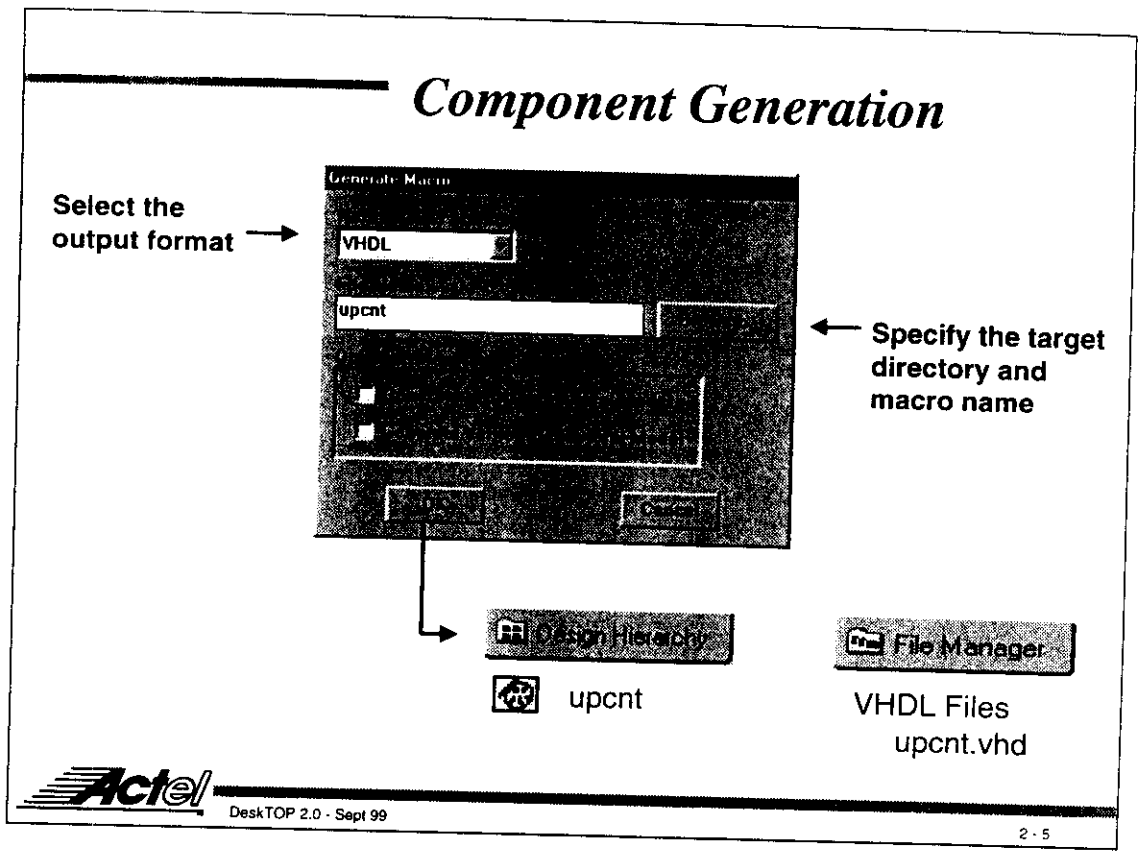
2 - 3

Notes

For Example: A Counter

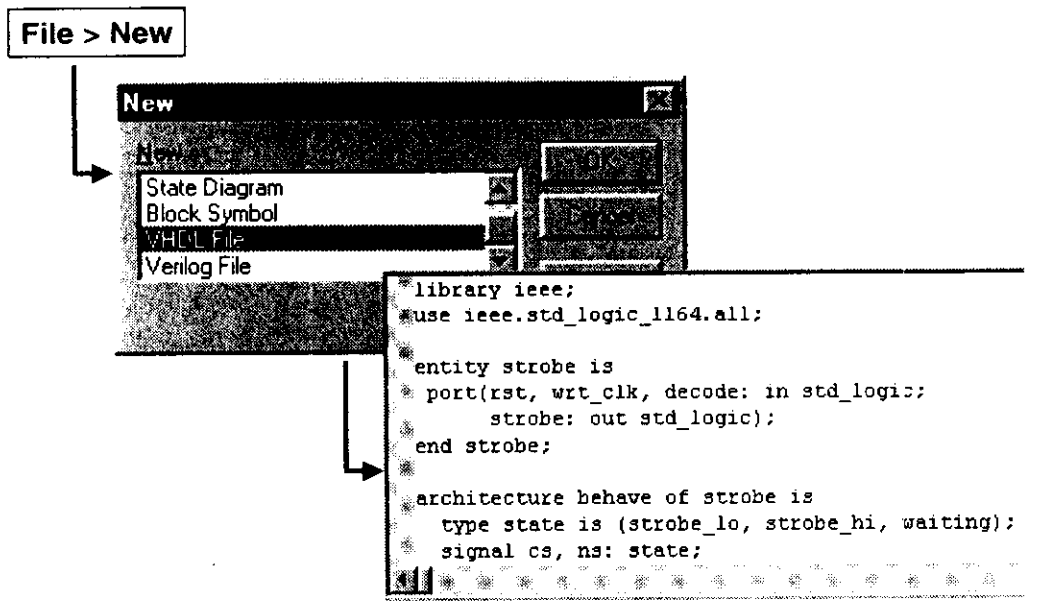


Notes


Component Generation
Select the output format
VHDL
upcnt
Specify the target directory and macro name
Design Hierarchy
File Manager
upcnt
VHDL Files
upcnt.vhd
Actel
DeskTOP 2.0 - Sept 99
2-5

Notes

Creating New VHDL Macros



Notes

Lab Introduction

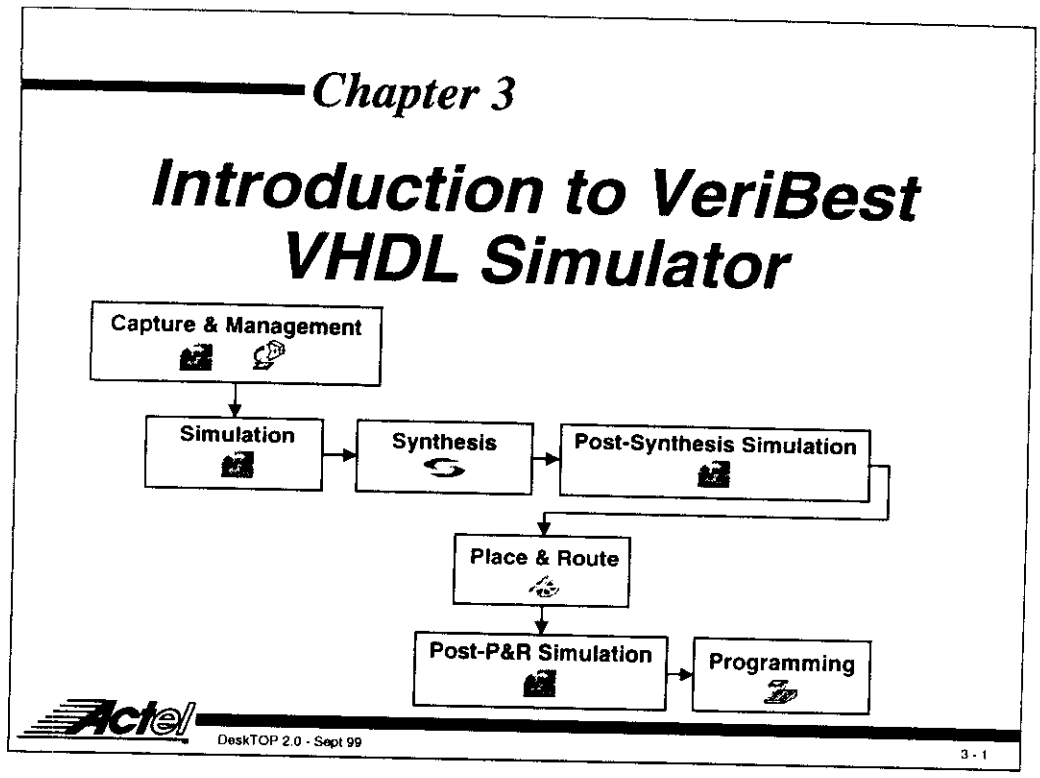
- **Complete Lab 1**
  - Perform all labs in your *dmac.prj* project.
  - Generate three ACTgen macros:
    - MY_RAM
    - UPCNT
    - DWNCNT
- **Complete Lab 2**
  - Create a new VHDL block:
    - DECODE
  - Add two existing VHDL blocks to the project:
    - CONTROLL
    - STROBE

Notes

Review

- What is ACTgen for?
- What output format is automatically selected by the DeskTOP? Why?
- What useful information is stored in the *.log* file?
- What is the .gen file for?

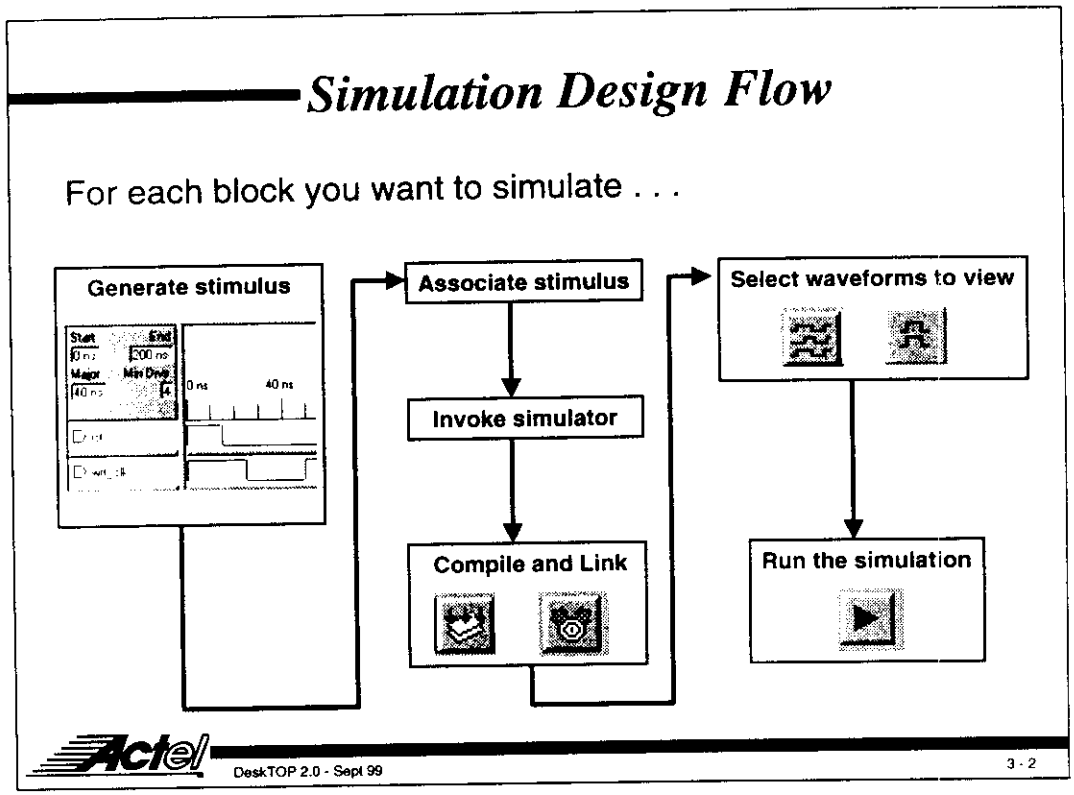
Notes



Objectives

After completing this chapter, you will be able to:

1. Diagram the design flow for stimulus generation and simulation
2. Explain how to generate a stimulus file.
3. Associate a stimulus file.
4. Invoke the VeriBest VHDL Simulator.
5. Explain how to view simulation results for block I/Os as well as internal nets (including cross-probing)
6. Show how to restart a simulation.



Notes

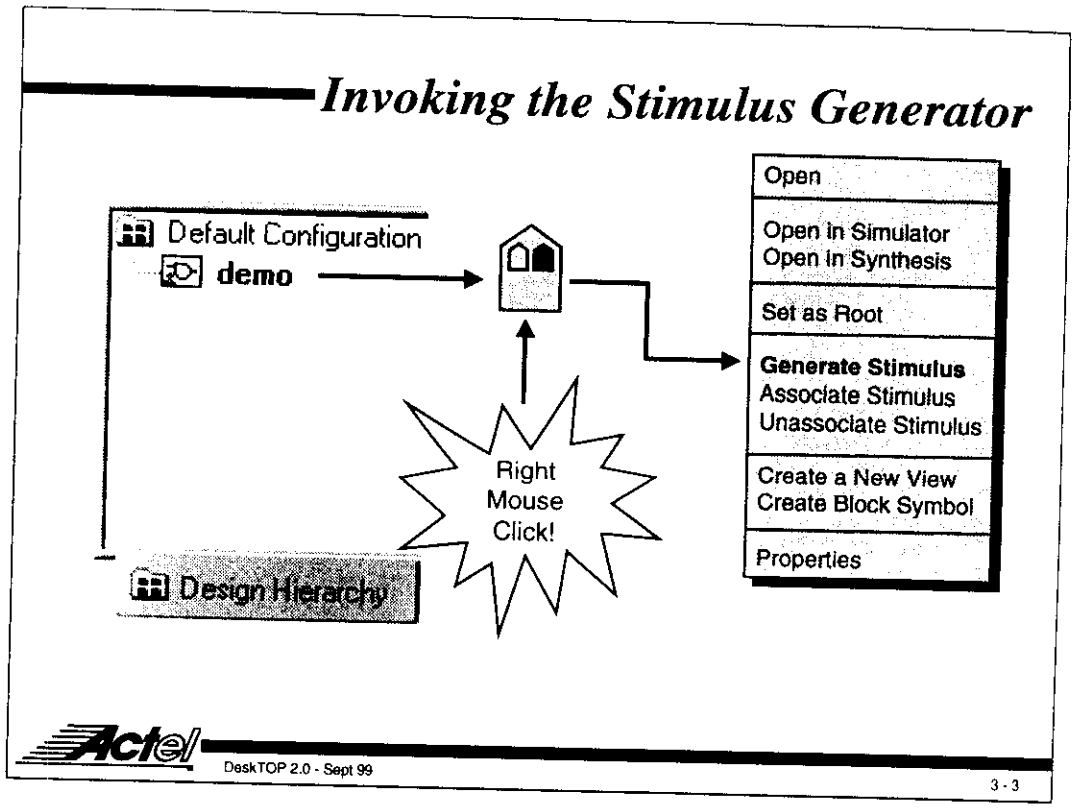
Before you can simulate, you must have a stimulus file. This file is easily created using VeriBest's graphical stimulus generation tool.

Once created, the stimulus file must be associated to the block that you want to simulate. This allows for easy switching from one stimulus file to another.

When the simulator is invoked, the VHDL block and its associated stimulus file must be compiled and linked.

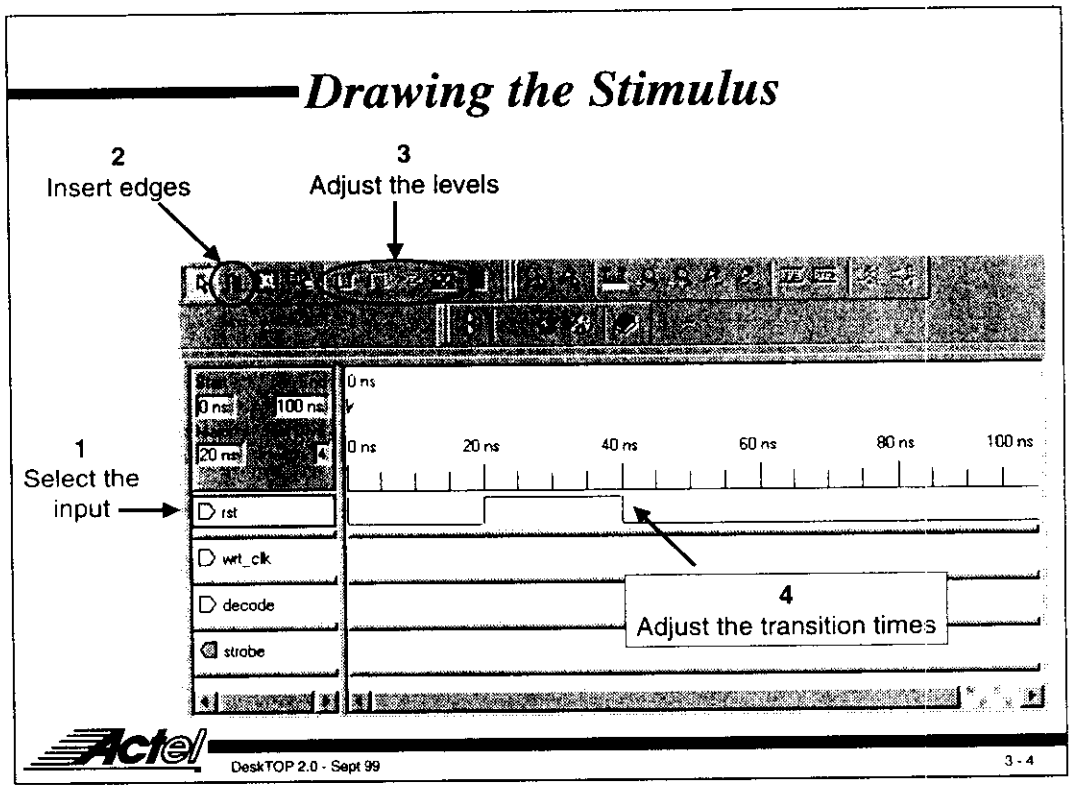
Next you must use the Wave Viewer to select nets that will be tracked and displayed during simulation. You may select block I/Os or use cross-probing to select internal nets as well.

Finally you run the simulation and review the results.



Notes

To invoke the stimulus generator tool, highlight the desired block and right-mouse click to select **Generate Stimulus**.



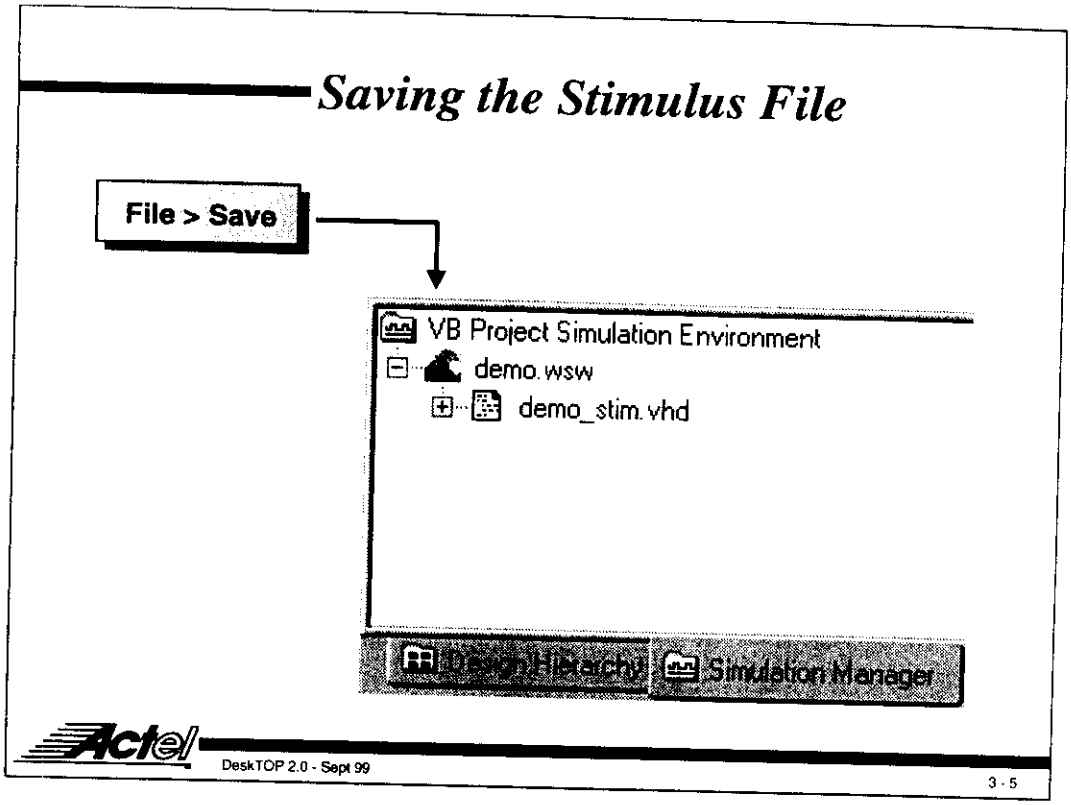
Notes

The stimulus generator has icon buttons that make it very easy to quickly draw the input waveforms.

First you select the waveform and insert edges at approximately the desired times.

Next, you select each segment and adjust its level to high, low, tristate or unknown.

Finally, you select each transition edge and position it exactly in time using the mouse or a type-in field.

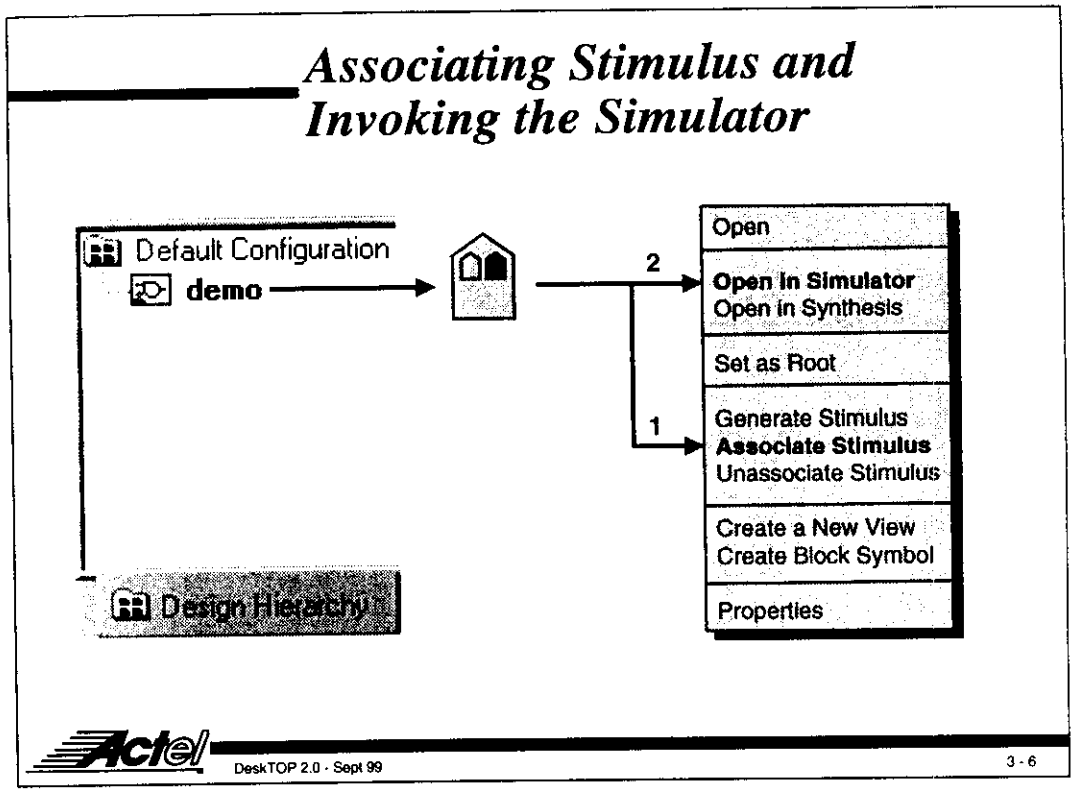


Notes

By default, the stimulus file is name *<block>_stim.vhd* and placed in a *<block>.wsw* directory under the **Simulation Manager** tab. The next time the stimulus generator is invoked, this same file will automatically be opened.

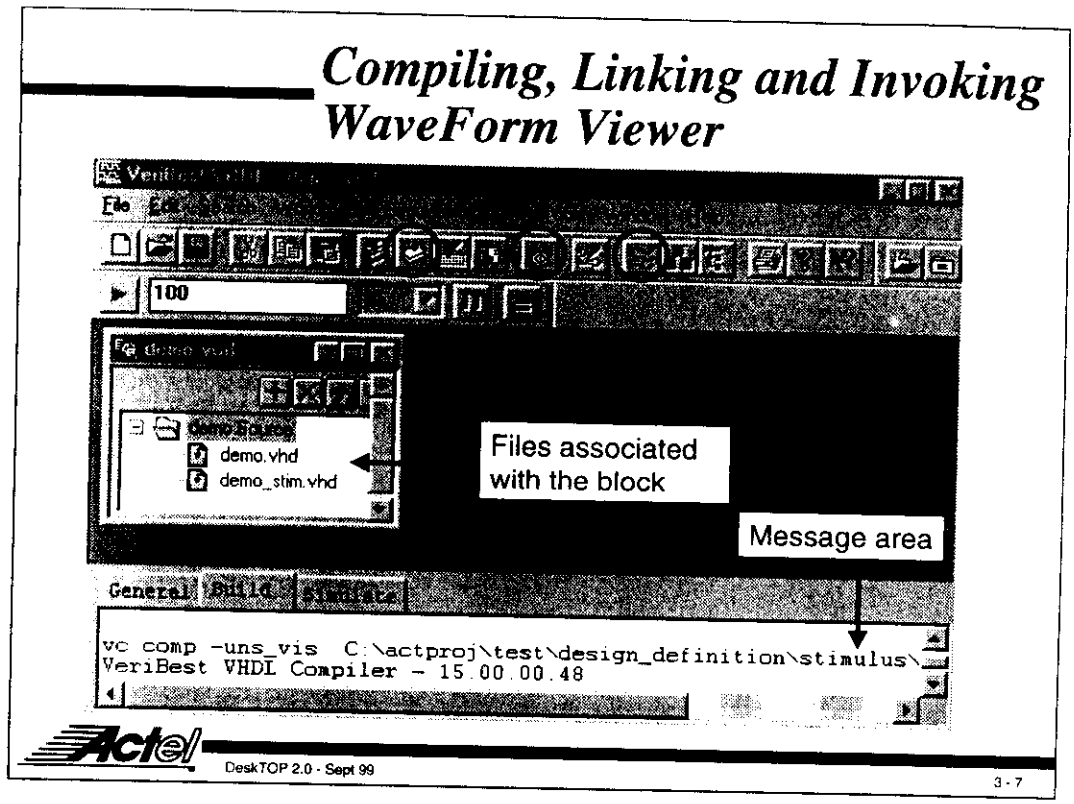
To generate another, separate stimulus file, you must make modifications to an existing one and be sure to select **File > Save As**. Otherwise, the old file will automatically be overwritten.

C4



Note

To use the stimulus file, you must associate it to the block as shown above. Once associated, the block can be opened in the simulator as shown above.



Notes

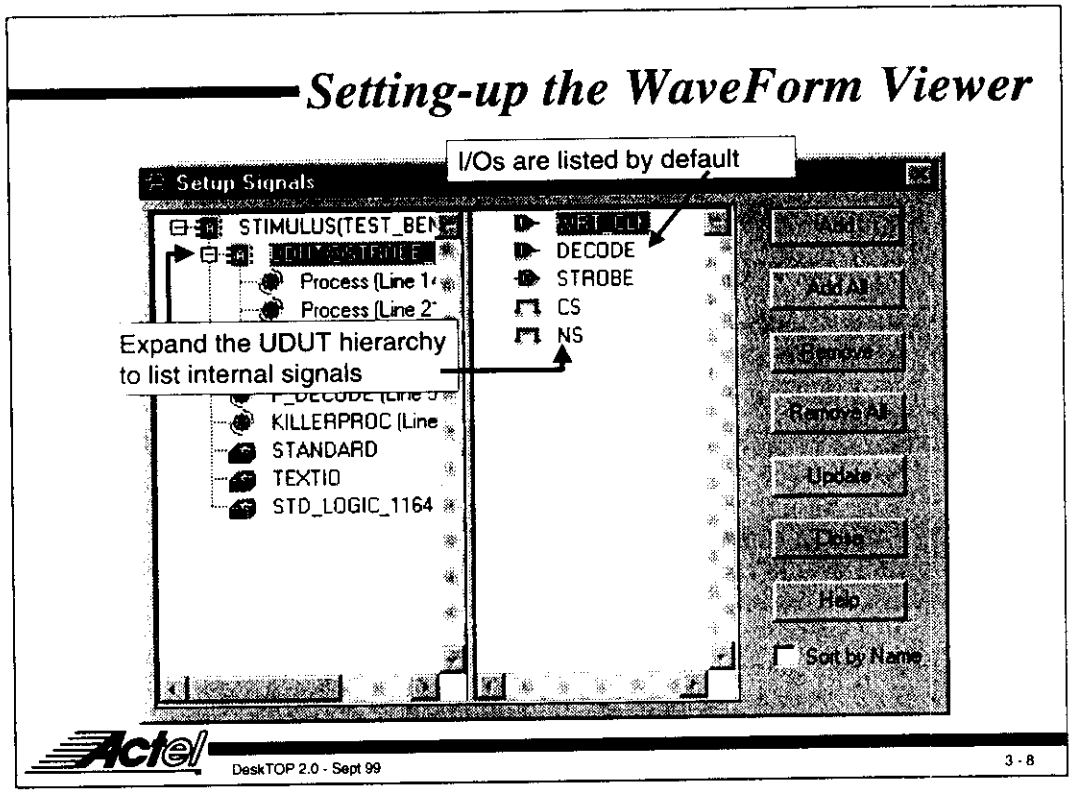
After invoking the simulator, you should first verify that all the files associated with the block are listed and in the correct order. The top -to-bottom order should match the hierarchy order from lowest to highest level, ending with the stimulus file. This prevents one block from referencing an uncompiled one.

If all the files are listed, you should:

- compile all  and then click 

When compilation and elaboration are successful, you should:

- set up the WaveForm Viewer by clicking 



Notes

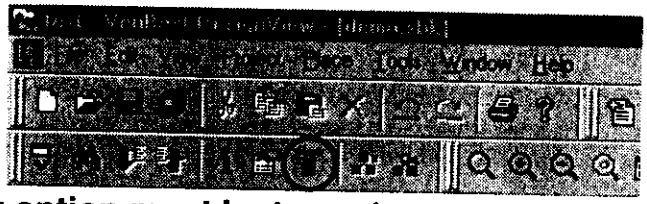
Before running the simulation, you must indicate the waveforms that you are interested in viewing. To do this, click the **Add Signals** icon in the WaveForm Viewer.

The result is shown above. By default, the block I/Os will be listed and ready to add (with the **Add** button).

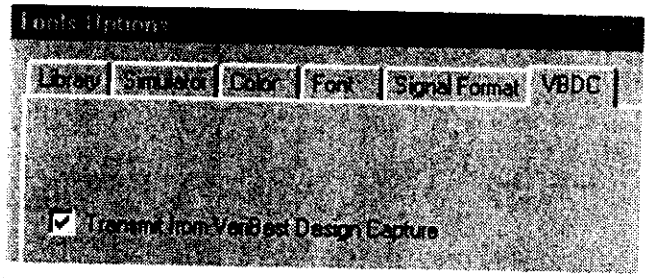
To see a list of internal signals, expand the UDUT entry as shown above.

Cross-Probing to View Internal Nets

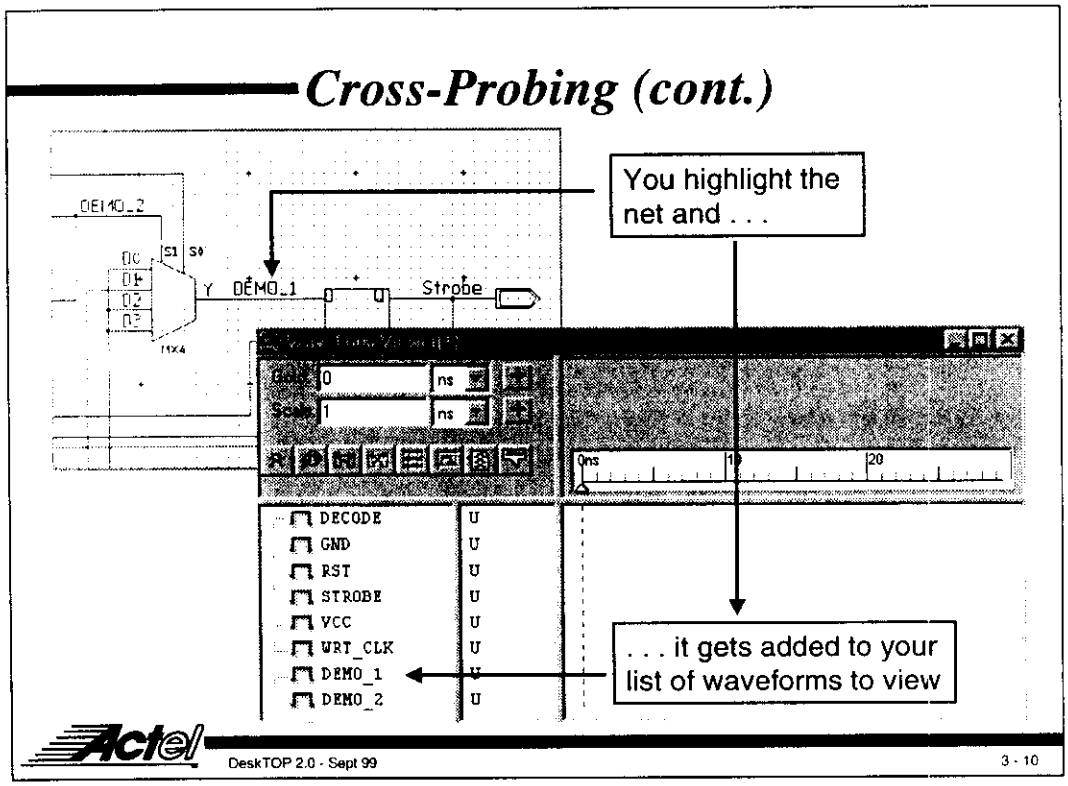
- **Schematic must be open and Transmit icon must be selected.**



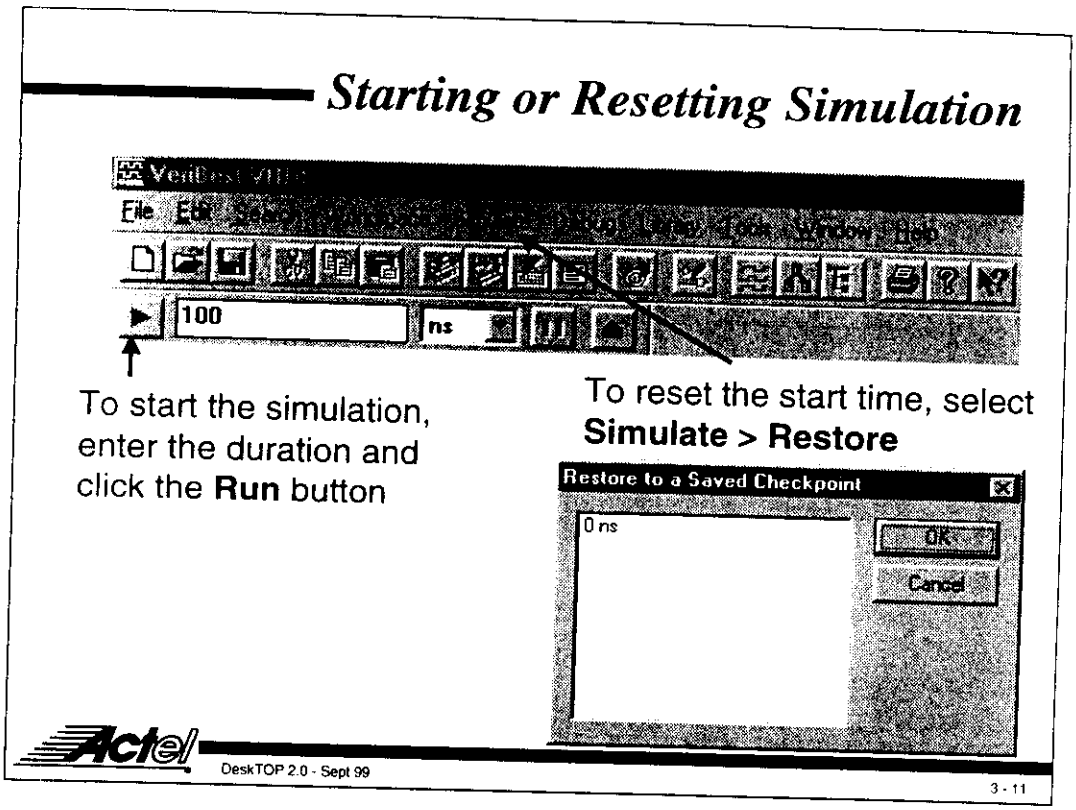
- **Simulator option must be turned on**



Notes


Cross-Probing (cont.)
You highlight the net and . . .
. . . it gets added to your list of waveforms to view
DEMO_1
Strobe
DECODE
GND
RST
STROBE
VCC
WRT_CLK
DEMO_1
DEMO_2
DeskTOP 2.0 - Sept 99
3 - 10

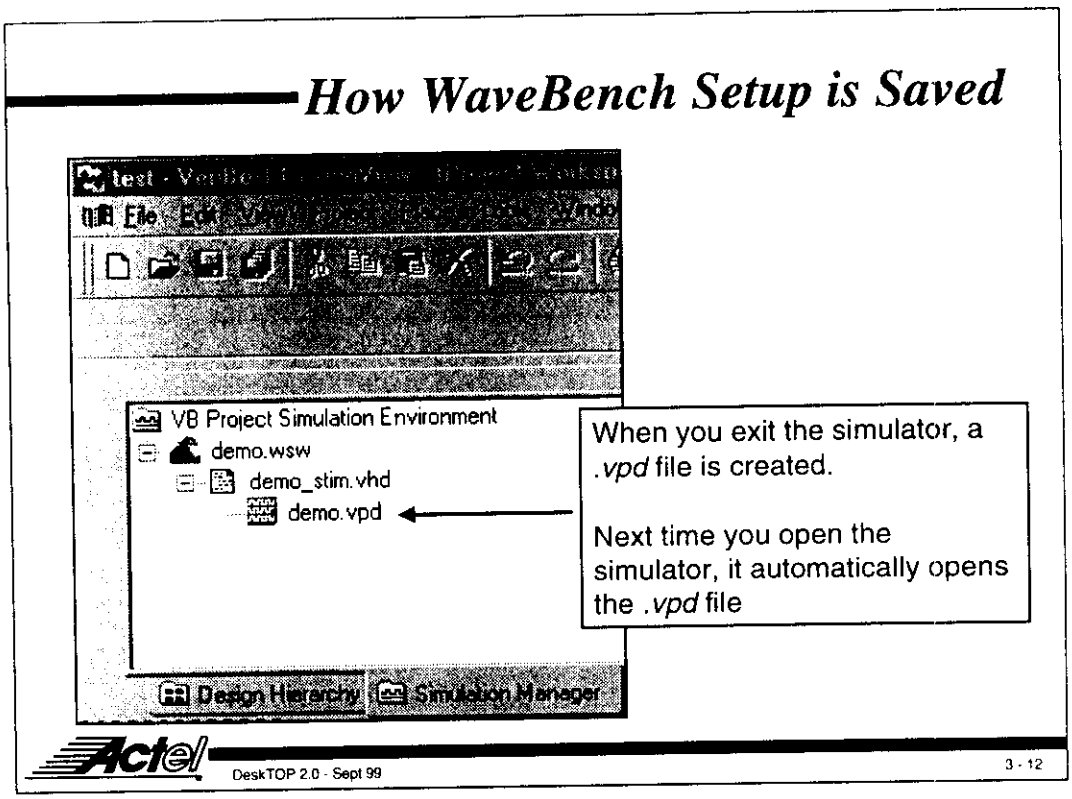
Notes



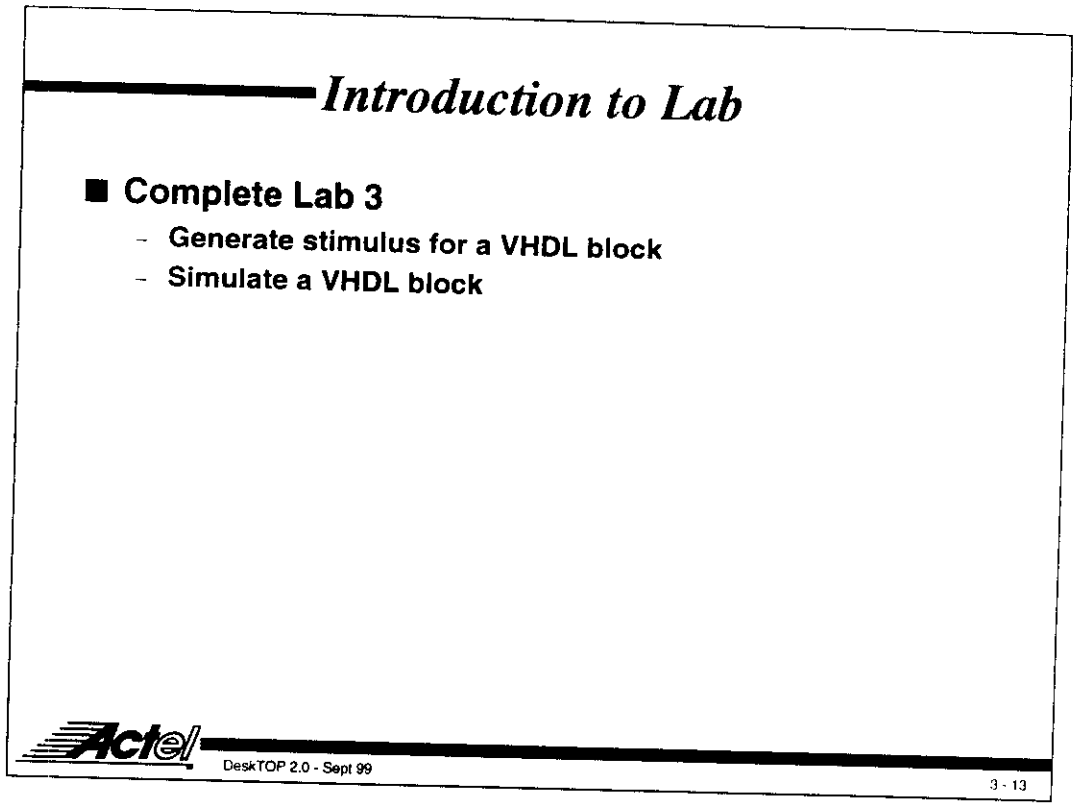
Notes

In the example above, the initial simulation is set to run from 0 to 100 ns. If another simulation is performed, it will run from 100 to 200ns. By default, the start time of a subsequent simulation is always the end time of the previous simulation.

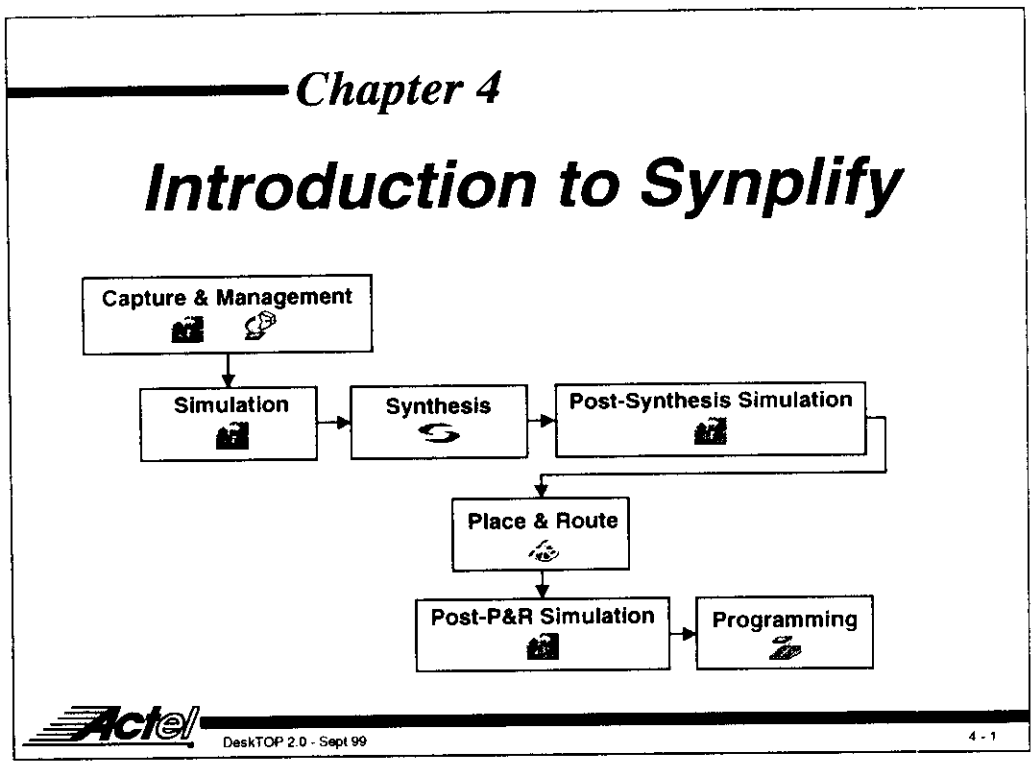
To reset the start time, select **Simulate > Restore** as shown above, then select the desired start time and click **OK**.


How WaveBench Setup is Saved
File
VB Project Simulation Environment
demo.wsw
demo_stim.vhd
demo.vpd
When you exit the simulator, a .vpd file is created.
Next time you open the simulator, it automatically opens the .vpd file
Design Hierarchy
Simulation Manager
Actel
DeskTOP 2.0 - Sept 99
3 - 12

Notes


Introduction to Lab
Complete Lab 3
Generate stimulus for a VHDL block
Simulate a VHDL block
Actel
DeskTOP 2.0 - Sept 99
3 - 13

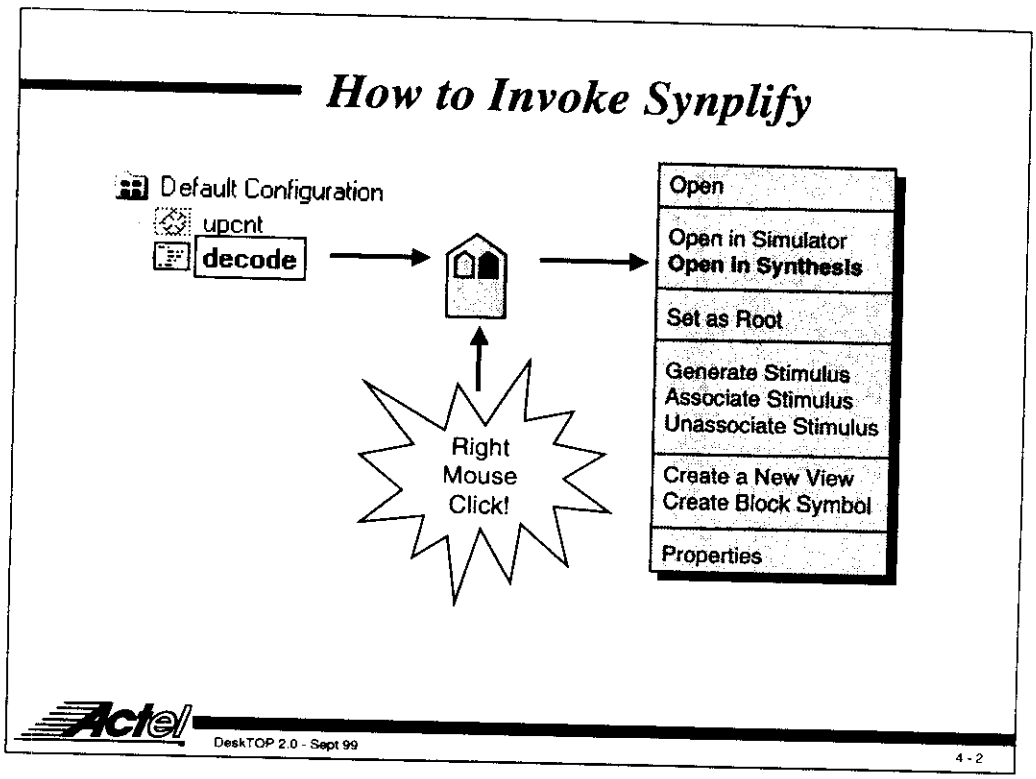
Notes



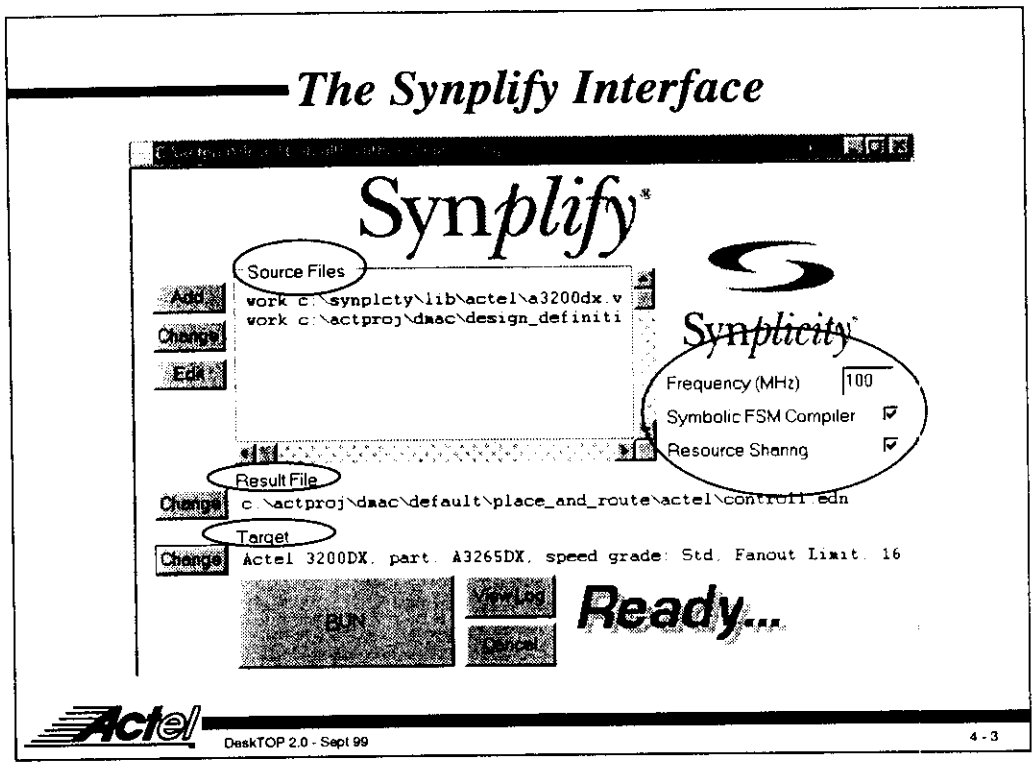
Objectives

After completing this chapter, you will be able to:

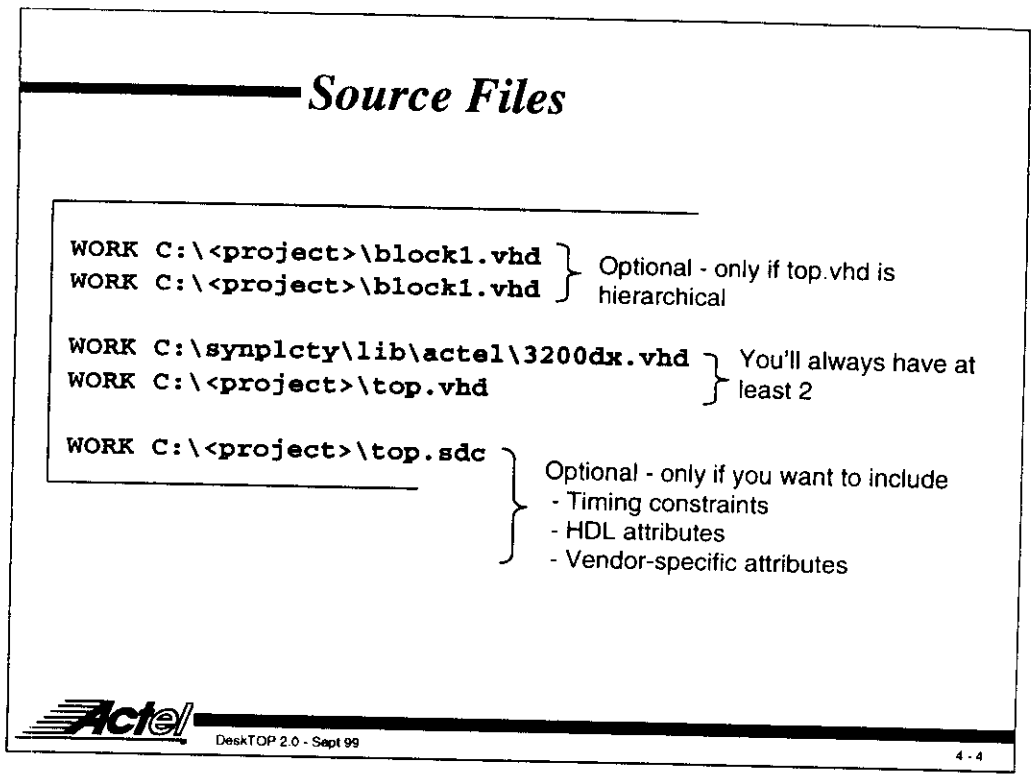
- Invoke Synplify
- Explain the various files that may appear in the Source window.
- Explain the global synthesis options.
- Explain the various Target options.
- Perform synthesis
- Use the log file to view and locate the cause of errors, warnings and notes.
- State the name and location of the file that should be opened to perform post-synthesis simulation.


How to Invoke Synplify
Default Configuration
upcnt
decode
Right
Mouse
Click!
Open
Open in Simulator
Open in Synthesis
Set as Root
Generate Stimulus
Associate Stimulus
Unassociate Stimulus
Create a New View
Create Block Symbol
Properties
Actel
DeskTOP 2.0 - Sept 99
4 - 2

Notes


The Synplify Interface
Synplify
Source Files
Add...
Change...
Edit...
work c:\synplcty\lib\actel\a3200dx.v
work c:\actproj\dmac\design_definiti
Synplicity
Frequency (MHz)
100
Symbolic FSM Compiler
Resource Sharing
Result File
Change...
c:\actproj\dmac\default\place_and_route\actel\control.edn
Target
Change...
Actel 3200DX, part: A3265DX, speed grade: Std, Fanout Limit: 16
RUN
View Log
Cancel
Ready...
Actel
DeskTOP 2.0 - Sept 99
4 - 3

Notes

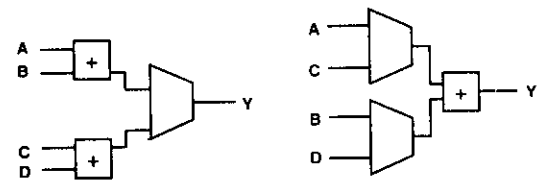

Source Files
WORK C:\<project>\block1.vhd
WORK C:\<project>\block1.vhd
Optional - only if top.vhd is hierarchical
WORK C:\synplcty\lib\actel\3200dx.vhd
WORK C:\<project>\top.vhd
You'll always have at least 2
WORK C:\<project>\top.sdc
Optional - only if you want to include
- Timing constraints
- HDL attributes
- Vendor-specific attributes
Actel
DeskTOP 2.0 - Sept 99
4 - 4

Notes

Synthesis Options

- **Frequency -**
- **Symbolic FSM Compiler - Turns on proprietary state machine optimization techniques.**
- **Resource Sharing - Turns on automatic sharing of operators (adders, subtractors, etc.) for example:**

```
If s='1', then
   Y<= A + B;
Else
   Y <= C + D;
```

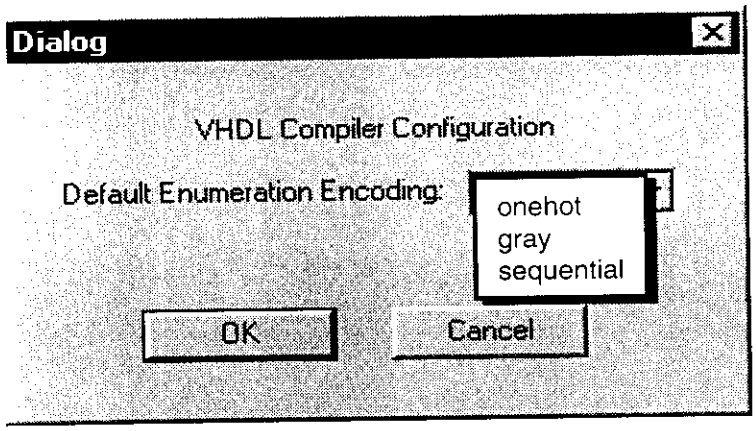


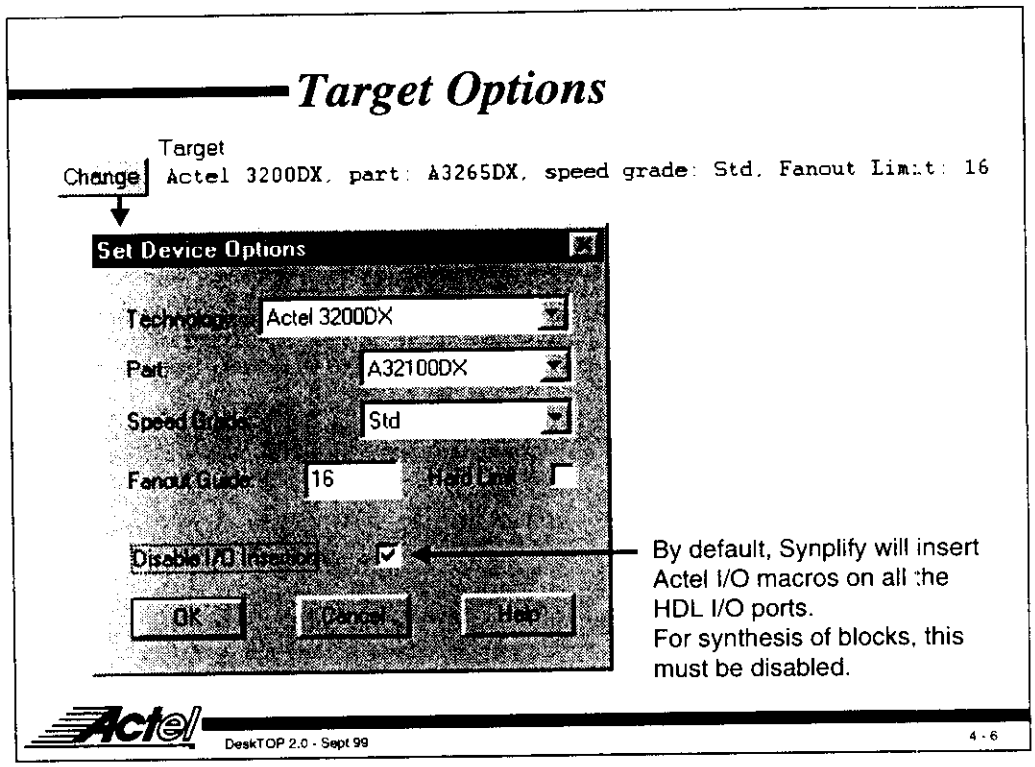
Without resource sharing With resource sharing

Notes

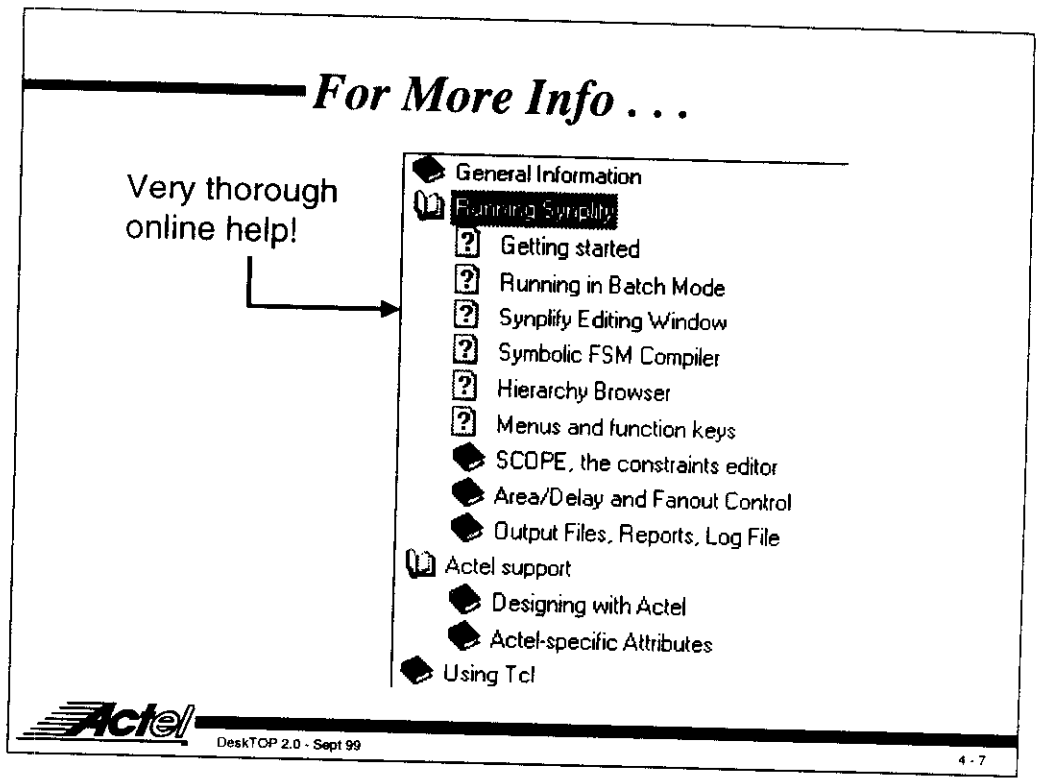
The options above are global and each can be overridden by constraints/attributes embedded in the code itself.

If you want to specify a global state machine encoding method (instead of letting the Symbolic FSM Compiler decide for you), select Tools > Options and you'll see the window shown below

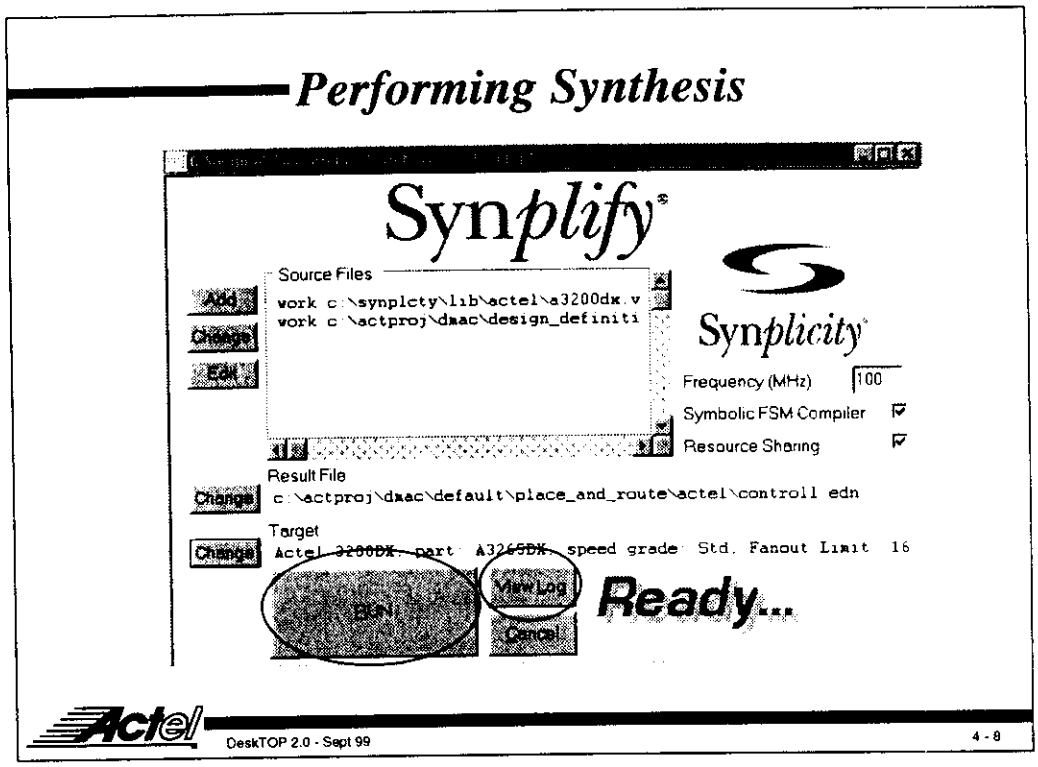



Target Options
Target
Change
Actel 3200DX, part: A3265DX, speed grade: Std, Fanout Limit: 16
Set Device Options
Technology:
Actel 3200DX
Part:
A32100DX
Speed Grade:
Std
Fanout Guide:
16
Hard Limit
Disable I/O Insertion
OK
Cancel
Help
By default, Synplify will insert Actel I/O macros on all the HDL I/O ports.
For synthesis of blocks, this must be disabled.
Actel
DeskTOP 2.0 - Sept 99
4 - 6

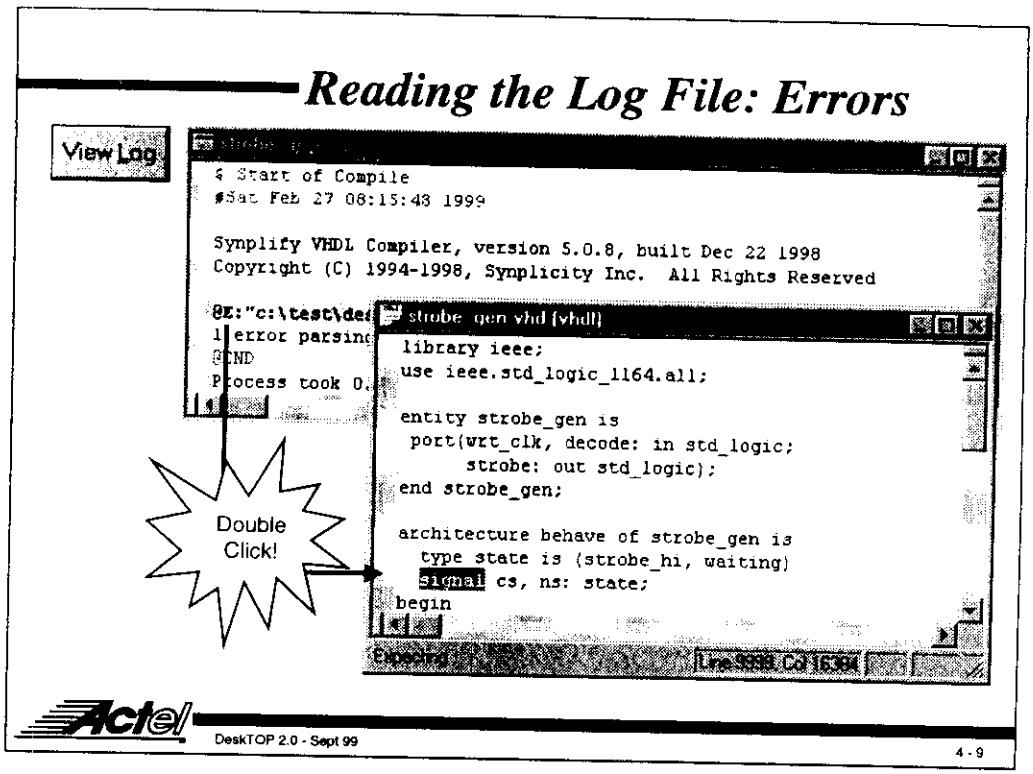
Notes


For More Info . . .
Very thorough online help!
General Information
Running Synplify
Getting started
Running in Batch Mode
Synplify Editing Window
Symbolic FSM Compiler
Hierarchy Browser
Menus and function keys
SCOPE, the constraints editor
Area/Delay and Fanout Control
Output Files, Reports, Log File
Actel support
Designing with Actel
Actel-specific Attributes
Using Tcl
Actel
DeskTOP 2.0 - Sept 99
4 - 7

Notes


Performing Synthesis
Synplify
Source Files
work c:\synplcty\lib\actel\a3200dx.v
work c:\actproj\dmac\design_definiti
Add
Change
Edit
Synplicity
Frequency (MHz)
100
Symbolic FSM Compiler
Resource Sharing
Result File
Change
c:\actproj\dmac\default\place_and_route\actel\controll.edn
Target
Change
Actel 3200DX, part A3265DX, speed grade Std, Fanout Limit 16
RUN
View Log
Cancel
Ready...
Actel
DeskTOP 2.0 - Sept 99
4 - 8

Notes



Notes

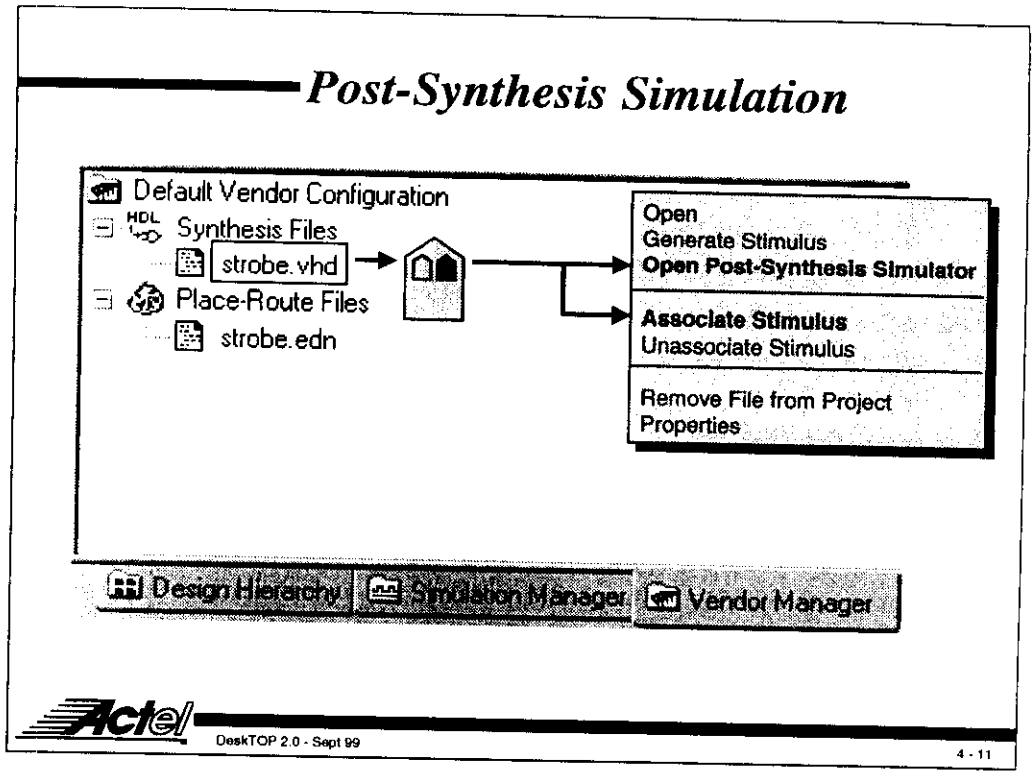
All errors, warnings and notes will be highlighted in the log file. To investigate, double click on the highlighted line and a VHDL editor will open, showing the offending code and the error message.

In this editor, you can correct the problems, save the file and immediately retry synthesis.

Reading the Log File Report

```
VHDL syntax check successful!
File c:\actproj\dmac\design_definition\hdl\vhdl\str
Synthesizing work.strobe.behave
@N:"c:\test\design_definition\hdl\vhdl\strobe.vhd":
Post processing for work.strobe.behave
@END

Synthesized design as a chip
------------------------------------
Resource Usage Report

Target Part: a3265dx
Combinational Cells: 0
Sequential Cells:    2 of 510 (0%)
Total Cells:         2 of 985 (1%)
Clock Buffers:       1
IO Cells:            4

Details:
   dfm7a:          2    seq:1

   clkbuf:         1    clock buffer
   inbuf:          2
   outbuf:         1
```


Notes

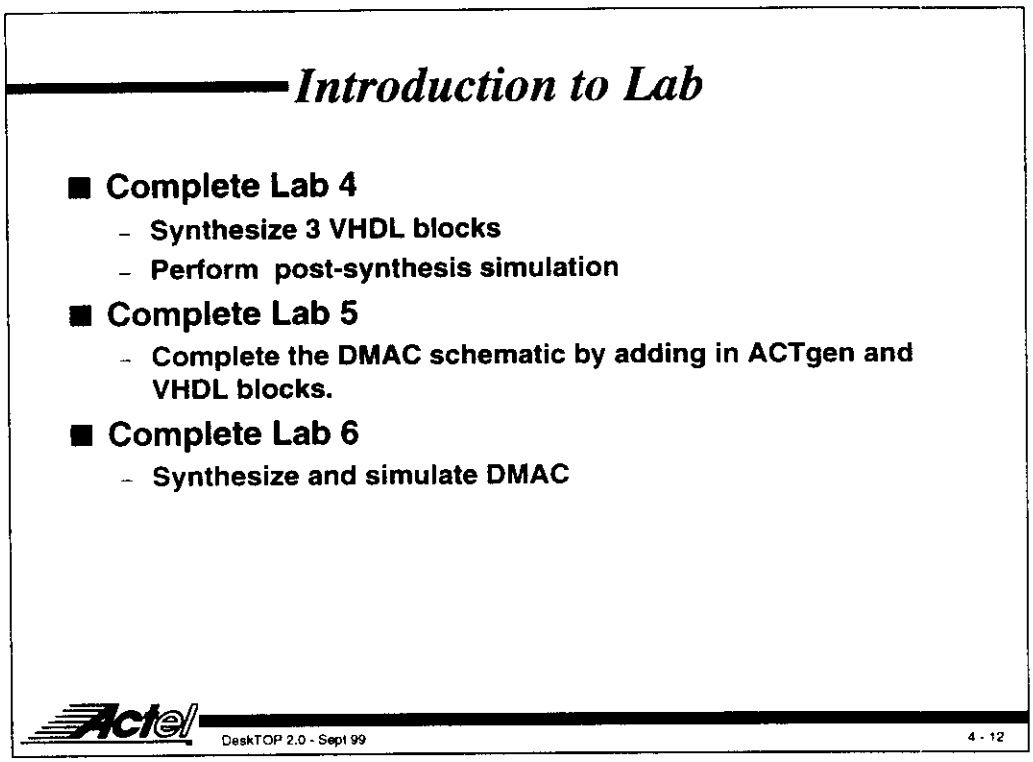


Notes

When synthesis is successfully completed, two new files are added to the project: the EDIF netlist and a structural VHDL file. Both files are displayed in the Vendor Manager tab as shown above.

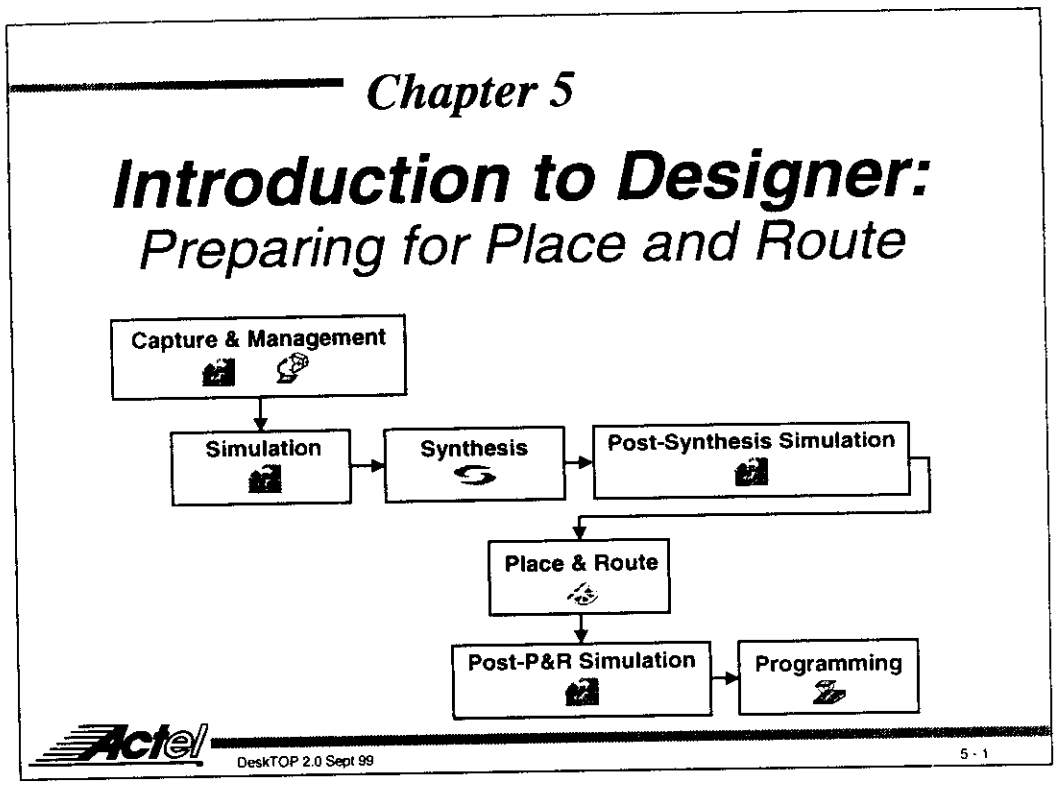
The EDIF netlist can be imported into Actel's Designer (if it represents the whole FPGA).

The structural VHDL file is used to perform post-synthesis simulation


Introduction to Lab
Complete Lab 4
Synthesize 3 VHDL blocks
Perform post-synthesis simulation
Complete Lab 5
Complete the DMAC schematic by adding in ACTgen and VHDL blocks.
Complete Lab 6
Synthesize and simulate DMAC
Actel
DeskTOP 2.0 - Sept 99
4 - 12

Notes

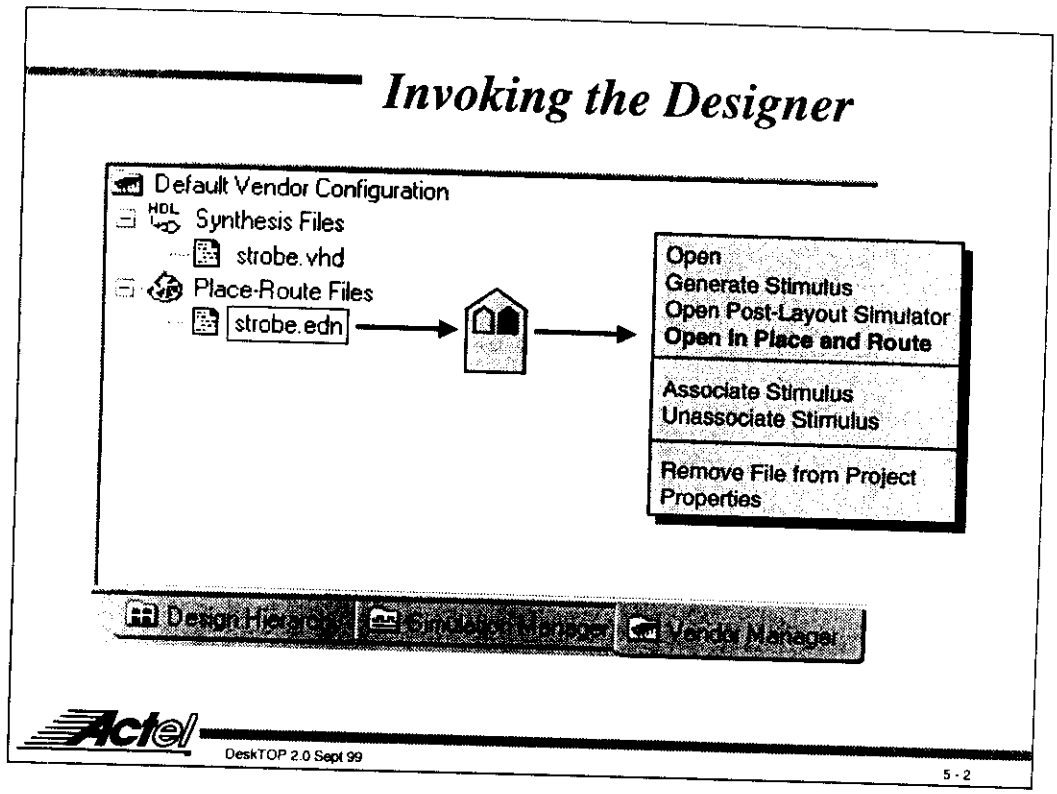
92



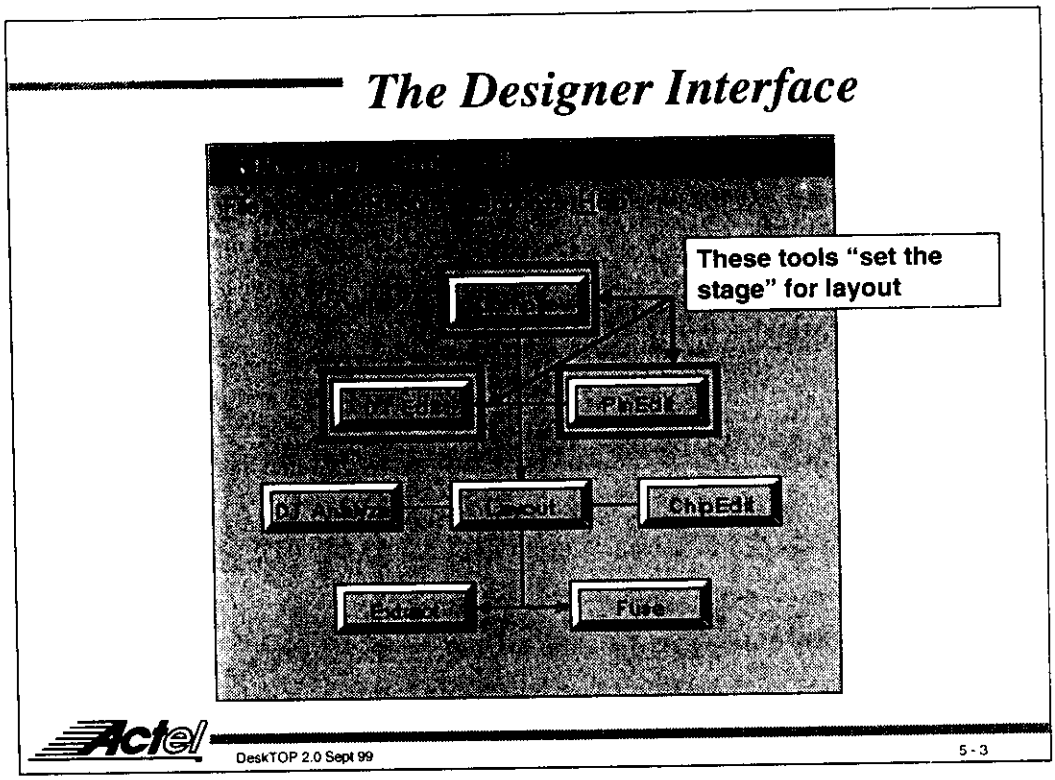
Objectives

At the end of this chapter, you will be able to:

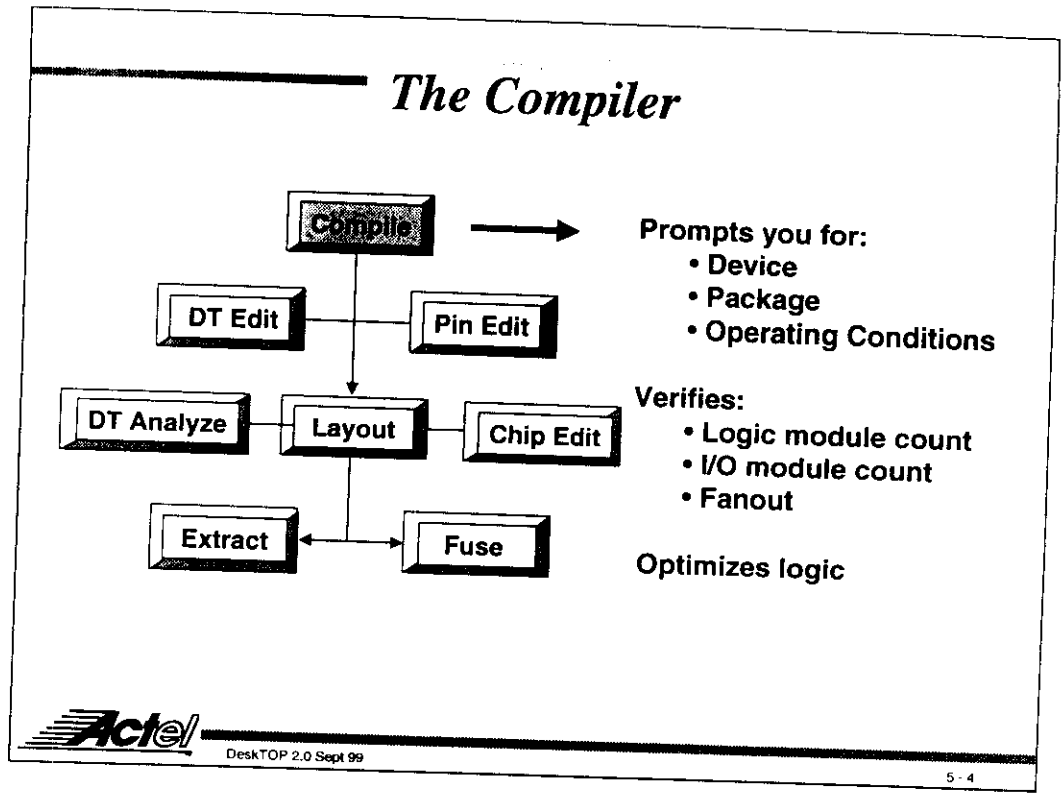
- List the Designer tools that impact how an FPGA is placed and routed or "layed out".
- Differentiate between opening a new design and opening an existing one.
- Describe the purpose of the Compiler.
- Discuss the pros and cons of making pin assignments.
- Explain the Pin Edit user interface and how it is used to make pin assignments.
- Differentiate between an assigned pin and a fixed pin and explain how they "end up that way".


Invoking the Designer
Default Vendor Configuration
Synthesis Files
strobe.vhd
Place-Route Files
strobe.edn
Open
Generate Stimulus
Open Post-Layout Simulator
Open in Place and Route
Associate Stimulus
Unassociate Stimulus
Remove File from Project
Properties
Design Hierarchy
Simulation Manager
Vendor Manager
Actel
DeskTOP 2.0 Sept 99
5 - 2

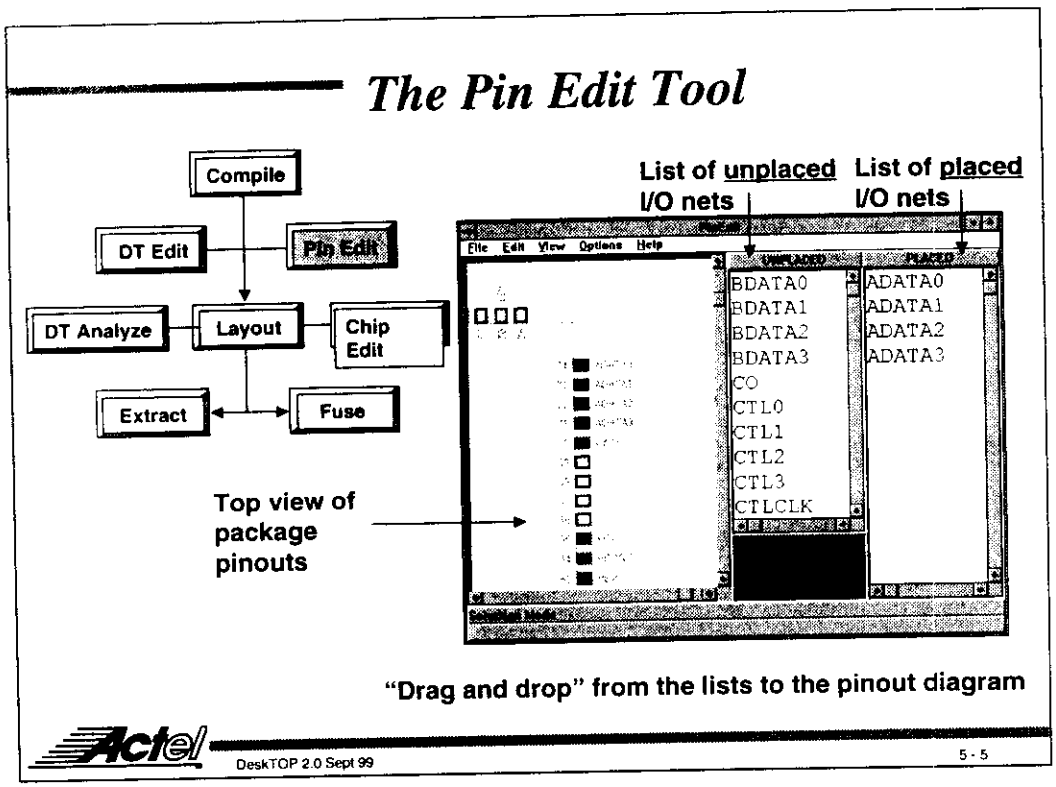
Notes


The Designer Interface
These tools "set the stage" for layout
PinEdit
ChipEdit
Fuse
DeskTOP 2.0 Sept 99
5 - 3

Notes

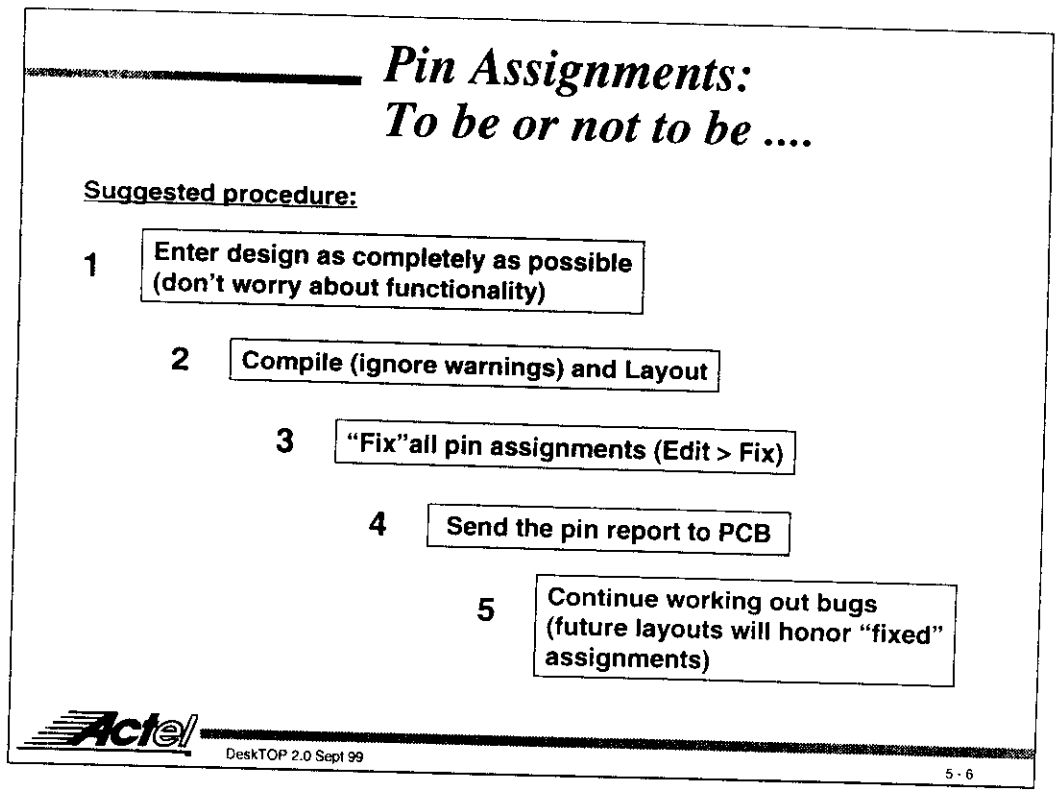

The Compiler
Compile
DT Edit
Pin Edit
DT Analyze
Layout
Chip Edit
Extract
Fuse
Prompts you for:
• Device
• Package
• Operating Conditions
Verifies:
• Logic module count
• I/O module count
• Fanout
Optimizes logic
Actel
DeskTOP 2.0 Sept 99
5 - 4

Notes


The Pin Edit Tool
Compile
DT Edit
Pin Edit
DT Analyze
Layout
Chip Edit
Extract
Fuse
Top view of package pinouts
List of unplaced I/O nets
List of placed I/O nets
File Edit View Options Help
BDATA0
BDATA1
BDATA2
BDATA3
CO
CTL0
CTL1
CTL2
CTL3
CTLCLK
ADATA0
ADATA1
ADATA2
ADATA3
"Drag and drop" from the lists to the pinout diagram
Actel
DeskTOP 2.0 Sept 99
5 - 5

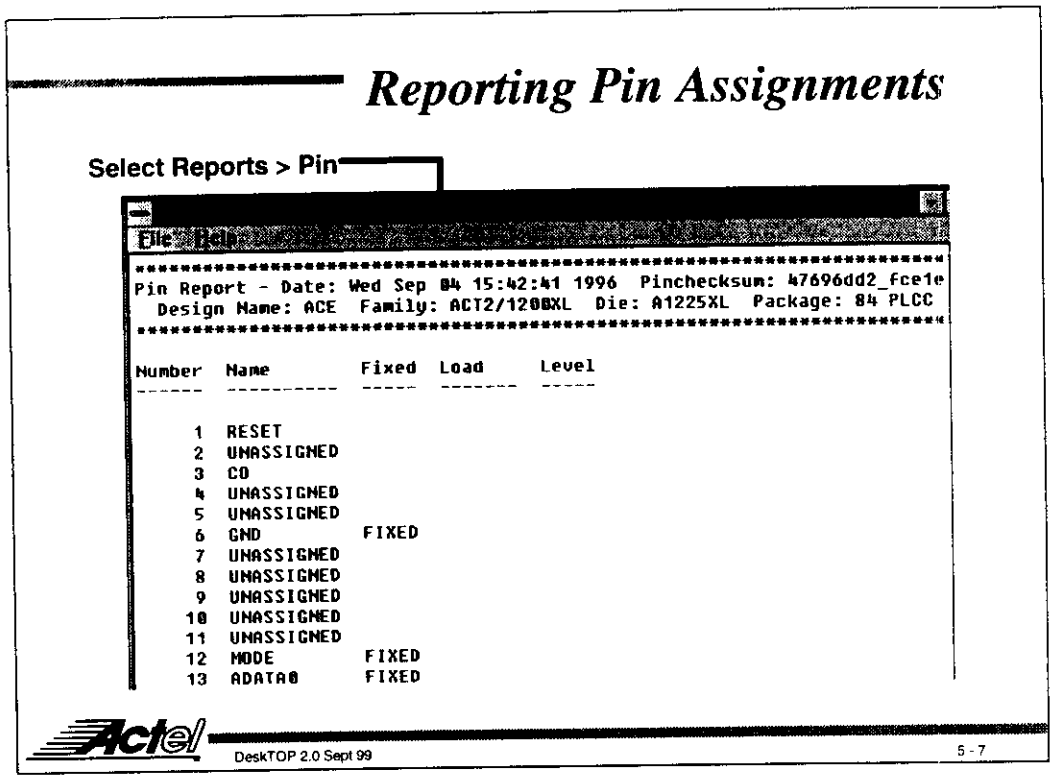
Notes

Q5

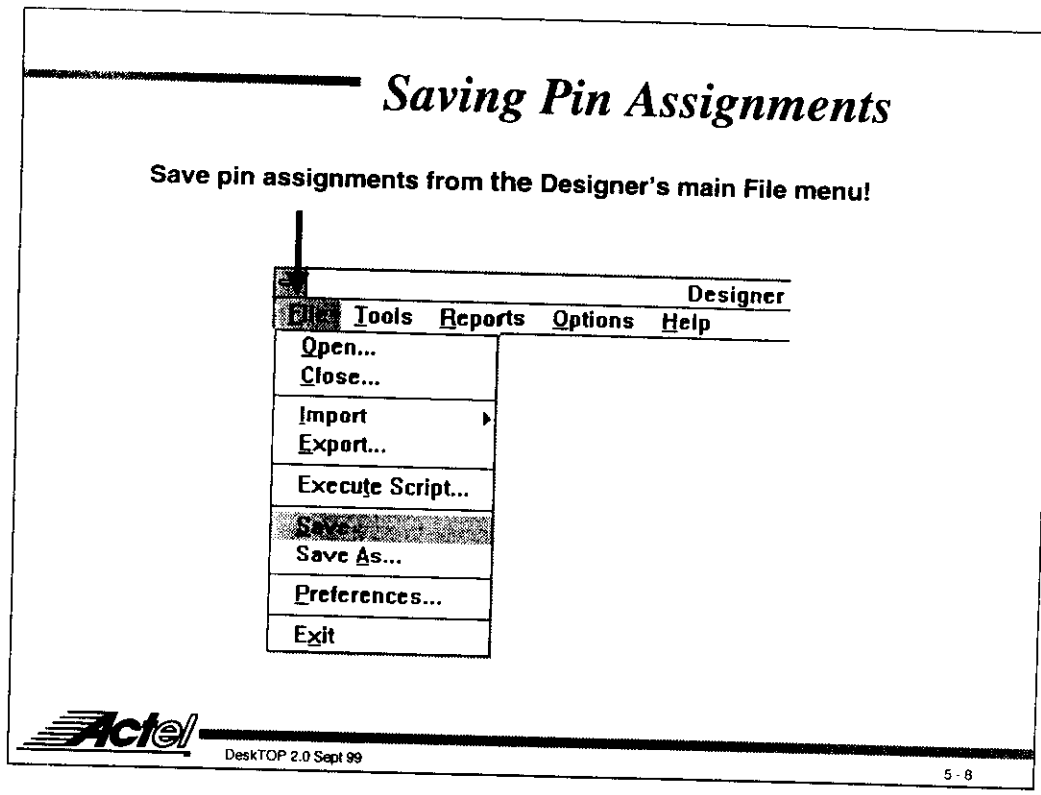

Pin Assignments:
To be or not to be ....
Suggested procedure:
1
Enter design as completely as possible (don't worry about functionality)
2
Compile (ignore warnings) and Layout
3
"Fix"all pin assignments (Edit > Fix)
4
Send the pin report to PCB
5
Continue working out bugs (future layouts will honor "fixed" assignments)
Actel
DeskTOP 2.0 Sept 99
5 - 6

Notes

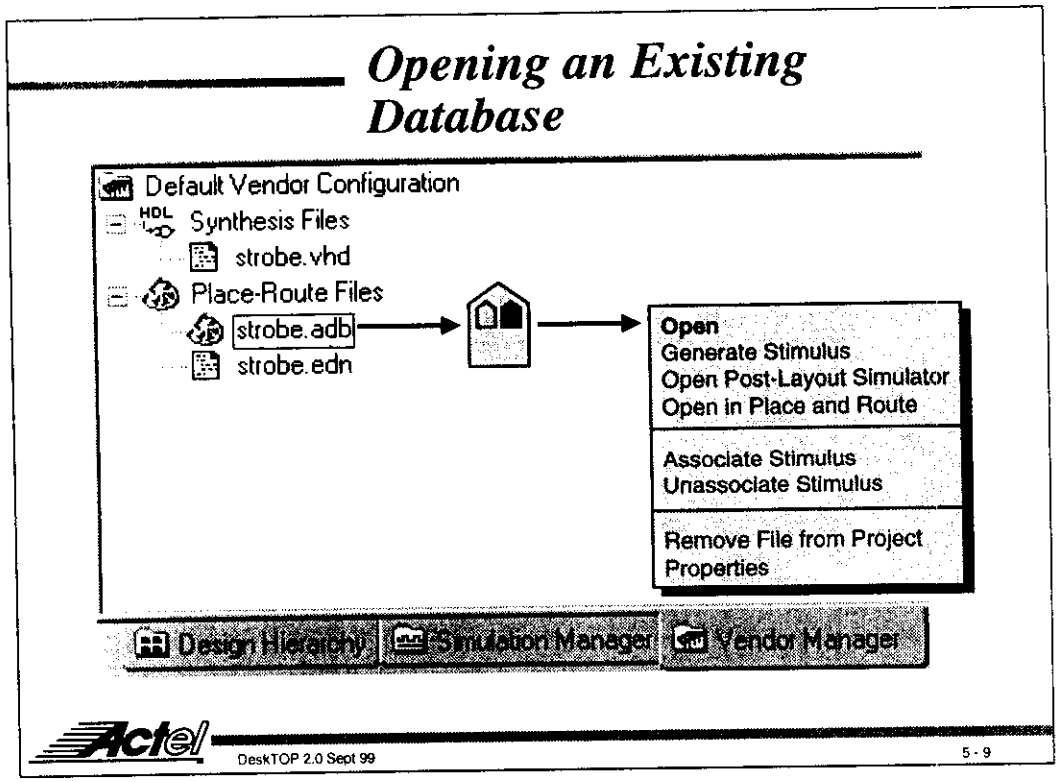
CIC1


Reporting Pin Assignments
Select Reports > Pin
File Help
Pin Report - Date: Wed Sep 04 15:42:41 1996 Pincheckcsum: 47696dd2_fce1e
Design Name: ACE Family: ACT2/1200XL Die: A1225XL Package: 84 PLCC
Number Name Fixed Load Level
1 RESET
2 UNASSIGNED
3 CD
4 UNASSIGNED
5 UNASSIGNED
6 GND FIXED
7 UNASSIGNED
8 UNASSIGNED
9 UNASSIGNED
10 UNASSIGNED
11 UNASSIGNED
12 MODE FIXED
13 ADATA0 FIXED
Actel
DeskTOP 2.0 Sept 99
5 - 7

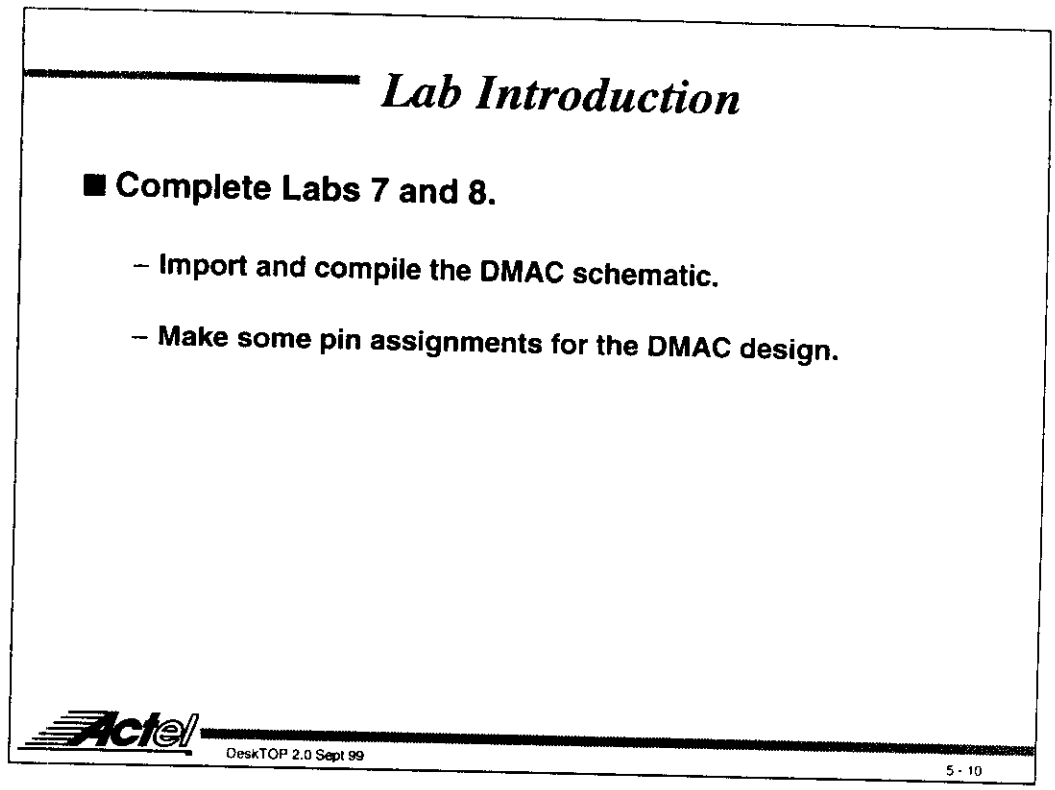
Notes


Saving Pin Assignments
Save pin assignments from the Designer's main File menu!
Designer
File Tools Reports Options Help
Open...
Close...
Import
Export...
Execute Script...
Save
Save As...
Preferences...
Exit
Actel
DeskTOP 2.0 Sept 99
5 - 8

Notes


Opening an Existing Database
Default Vendor Configuration
HDL
Synthesis Files
strobe.vhd
Place-Route Files
strobe.adb
strobe.edn
Open
Generate Stimulus
Open Post-Layout Simulator
Open in Place and Route
Associate Stimulus
Unassociate Stimulus
Remove File from Project
Properties
Design Hierarchy
Simulation Manager
Vendor Manager
Actel
DeskTOP 2.0 Sept 99
5 - 9

Notes

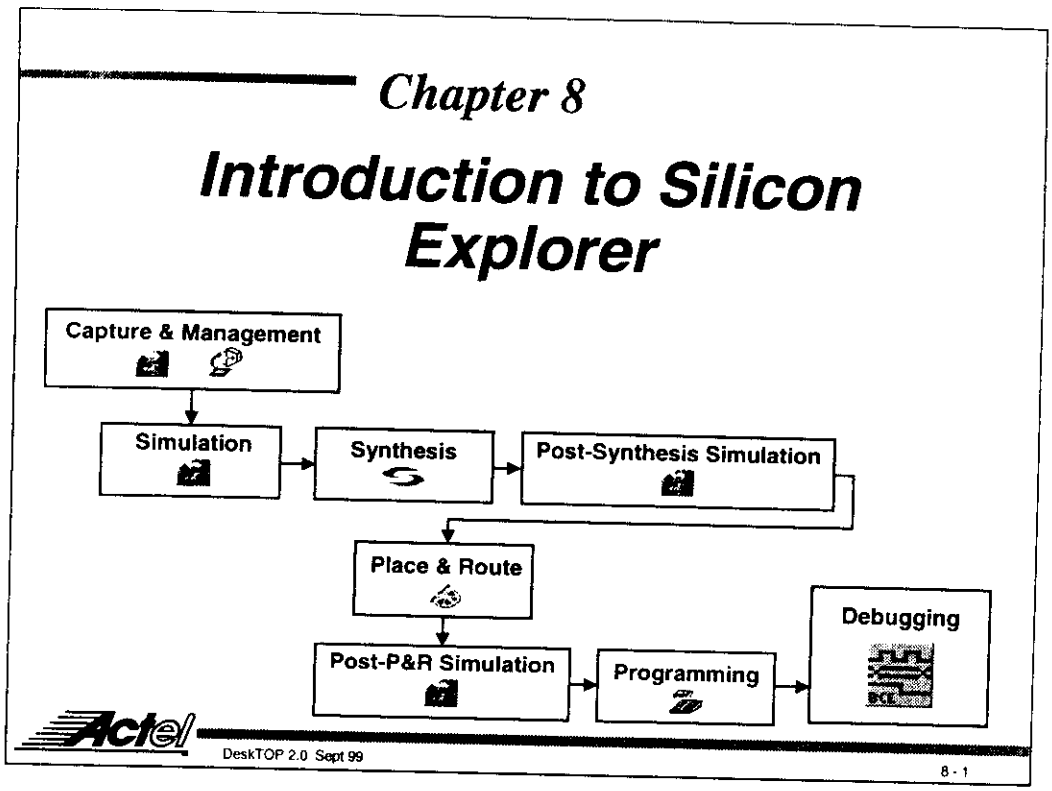

Lab Introduction
Complete Labs 7 and 8.
– Import and compile the DMAC schematic.
– Make some pin assignments for the DMAC design.
Actel
DeskTOP 2.0 Sept 99
5 - 10

Notes

Review

- How do you begin a Designer session for a new design?
- What does the Compiler do?
- What is Actel's recommendation with respect to manual pin assignments?
- Besides the Pin Edit tool, what other ways can you make manual pin assignments?
- Where is the Designer data stored? Where is it displayed in DesignView?

Notes

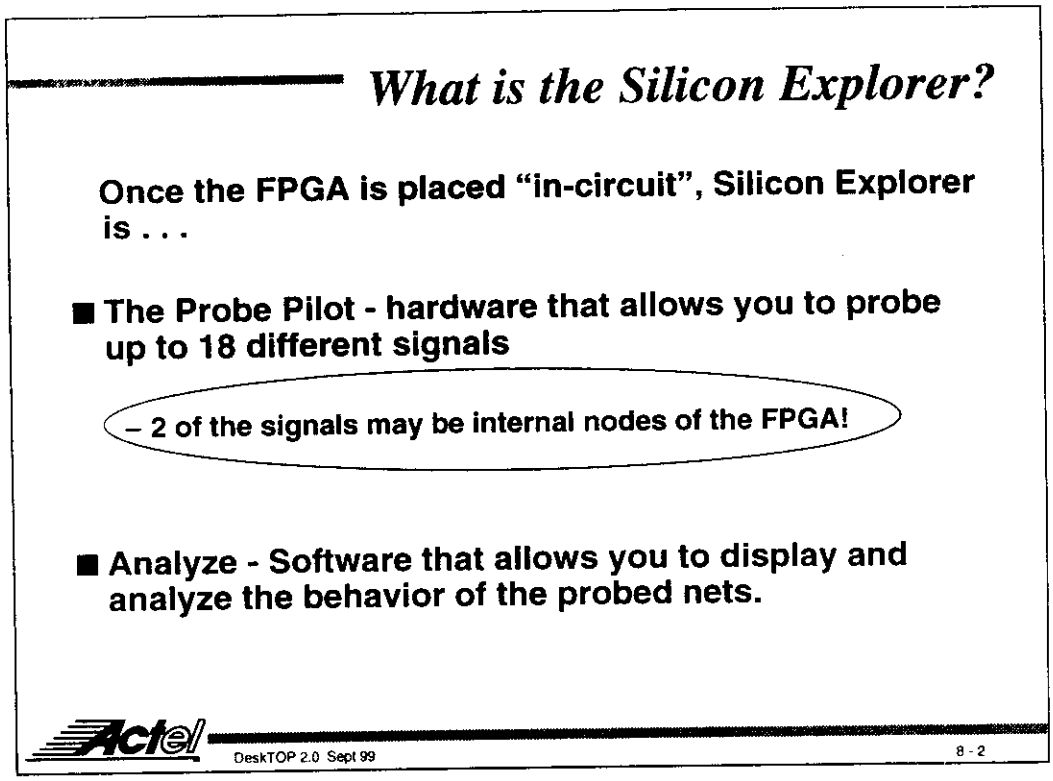


Objectives

At the end of this chapter, you will be able to:

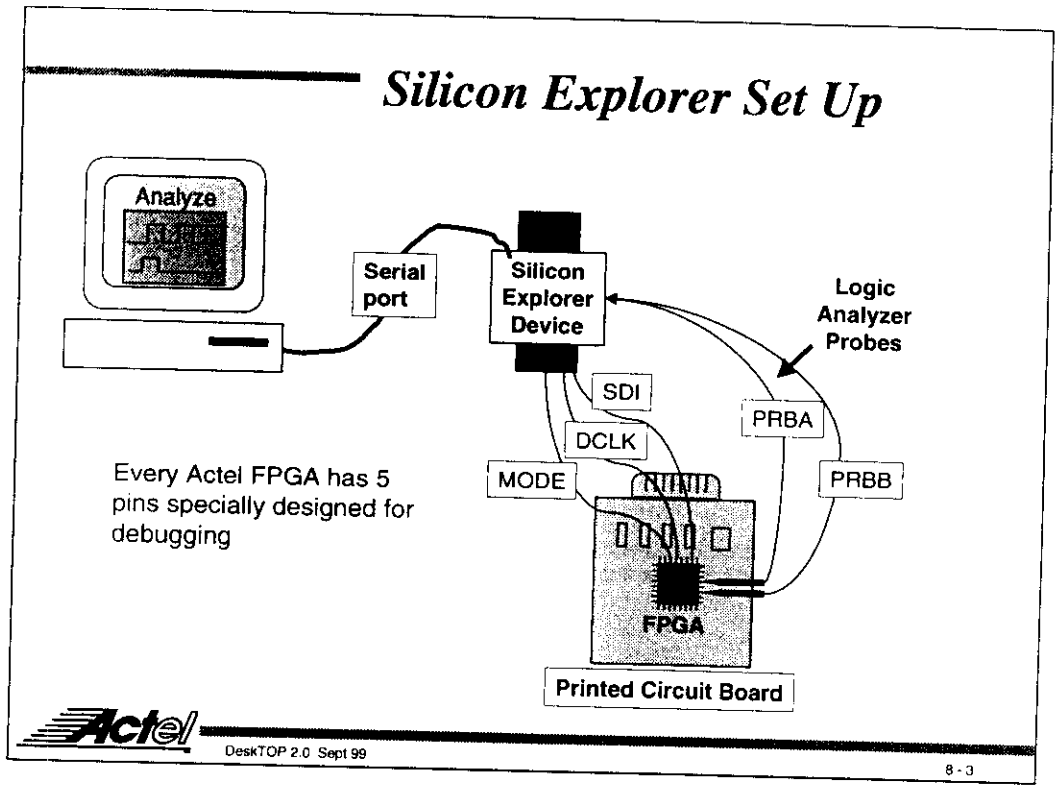
- Describe the overall functionality of the Silicon Explorer.
- Describe the purpose of the MODE, SDI, PRA, PRB and DCLK pins on Actel devices.
- Correctly connect a Probe Pilot to an "in-circuit" FPGA.
- Explain the user interface of the Analyze software.
- Explain the three steps that prepare a design for debugging.

ACS


What is the Silicon Explorer?
Once the FPGA is placed "in-circuit", Silicon Explorer is . . .
The Probe Pilot - hardware that allows you to probe up to 18 different signals
– 2 of the signals may be internal nodes of the FPGA!
Analyze - Software that allows you to display and analyze the behavior of the probed nets.
Actel
DeskTOP 2.0 Sept 99
8 - 2

Notes

100



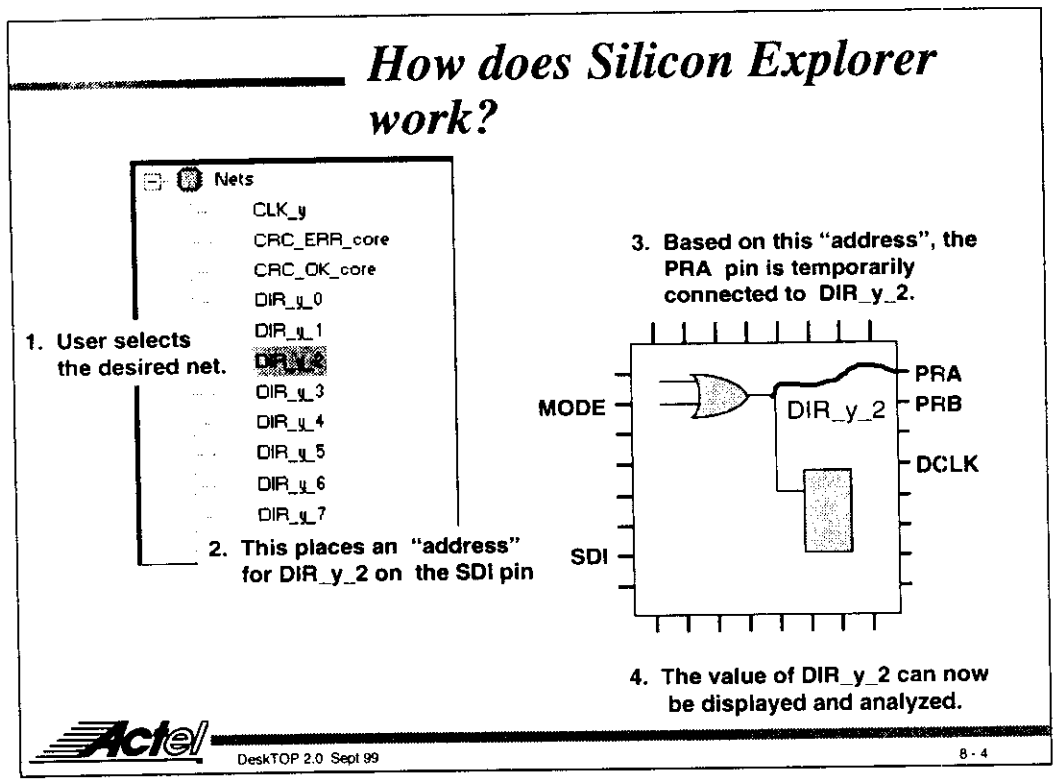
Notes

Every Actel device has 5 pins that are specially designed for debugging with the Silicon Explorer.

Three of the pins, MODE, SDI and DCLK are inputs to the FPGA.

The other 2 pins, PRA and PRB send the behavior of internal nets back for viewing with the Analyze software.

The Silicon Explorer itself is connected to the computer via the serial port.

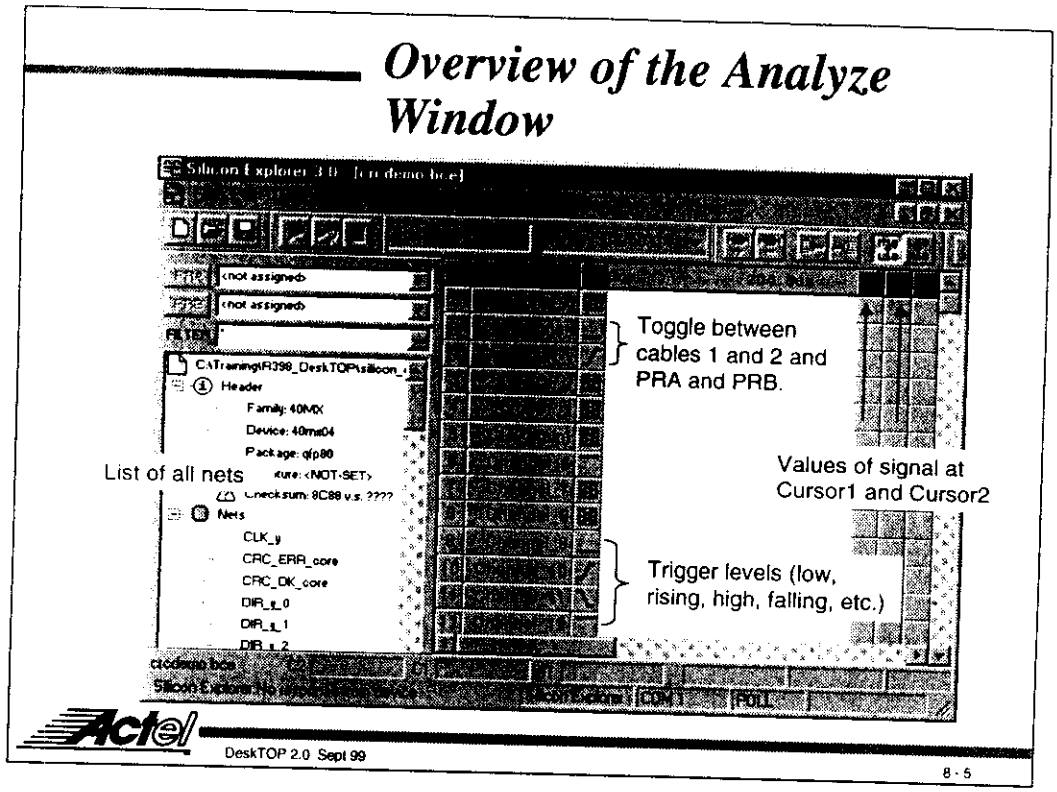


Notes

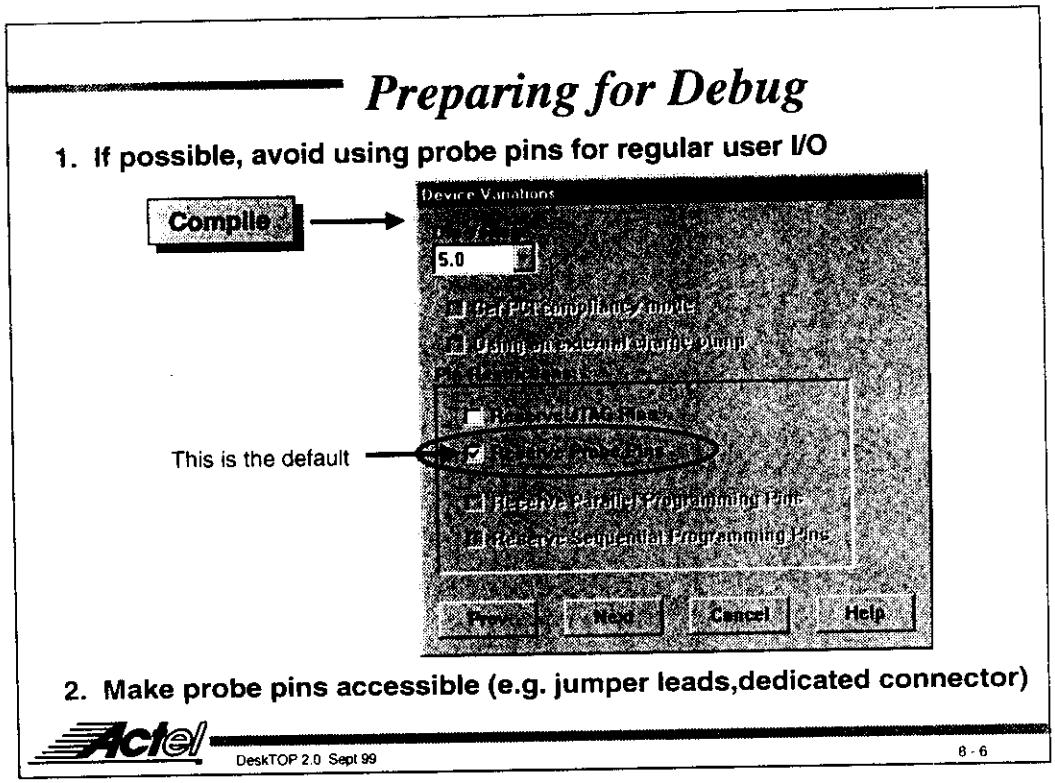
When a user invokes the Silicon Explorer software, the MODE pin is set high and the PRA, PRB, SDI and DCLK pins operate as follows:

Based on a user's request for the value of an internal signal (e.g. the DIR_y_2 net above), an "address" corresponding to that signal's location in the FPGA is generated and placed on the SDI pin and clocked in with DCLK.

Based on the SDI signal, the node of the signal in question is multiplexed out and made externally available on either the PRA or PRB pin. From here, its behavior can be viewed with the Analyze software.

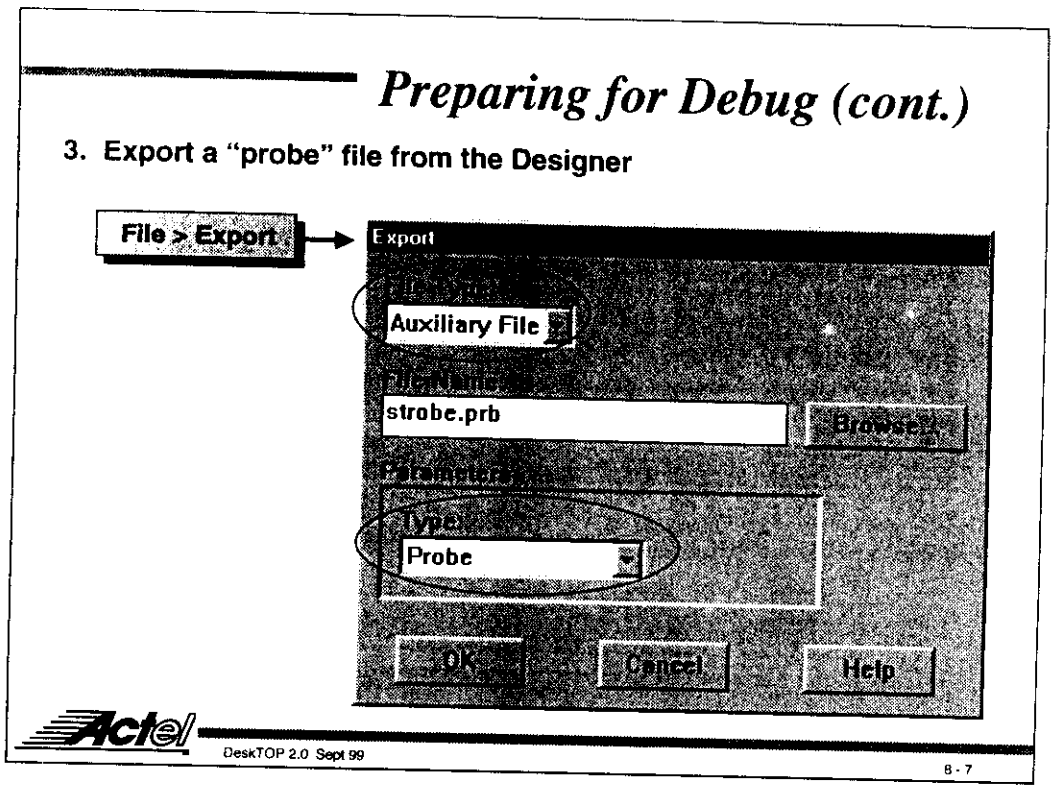

Overview of the Analyze Window
Toggle between cables 1 and 2 and PRA and PRB.
List of all nets
Values of signal at Cursor1 and Cursor2
Trigger levels (low, rising, high, falling, etc.)
DeskTOP 2.0 Sept 99
8 - 5

Notes



Notes

When you compile your netlist in the Designer, you are prompted to set the device variations as shown above. If possible, avoid using the probe pins by leaving the **Reserve Probe Pins** box checked as shown above. This is the default.



Notes

The Silicon Explorer understands a design's netlist by reading its "probe" file.

This file has a .prb extension and is created in the Designer by selecting **File > Export** and making the selection shown above.

Lab Introduction

- **Complete Lab 14.**
  - **Connect the Silicon Explorer to the CRCDEMO board.**
  - **Open the provided .prb file.**
  - **Investigate some internal nets.**

Notes