Libero® IDE Quick Start Guide

for Software v9.0

Microsemi®

Table of Contents

Introduction

This tutorial introduces you to the Microsemi SoC group FPGA development flow using the Libero Integrated Design Environment. It is a starting point for any FPGA design engineer who is new to Microsemi SoC, or just wants to learn more about the Libero IDE. After completing this tutorial, you will know the basics of how to use the Libero IDE and its tools to create a simple design incorporating Libero IDE Catalog IP core macros, library primitives, and Verilog or VHDL code.

This tutorial uses the IGLOO nano Starter Kit and AGLN250 device, but the same design could be ported to a different development board with slight changes in pin assignments.

This tutorial includes an introduction to the following tools and features:

- Libero IDE Project Manager
  - Design Explorer
  - Project Flow Window
  - SmartDesign
  - Core Catalog
- Mentor Graphics ModelSim
- Synplicity Synplify Pro
- Microsemi Designer
  - Compile
  - Place-and-Route
  - I/O Attribute Editor in MultiView Navigator
- FlashPro

Design Overview

The example design that you build in this tutorial is a counter design with a PLL (Phase Locked Loop) that blinks LEDs on the target board. Figure 1 shows the block diagram of the design.

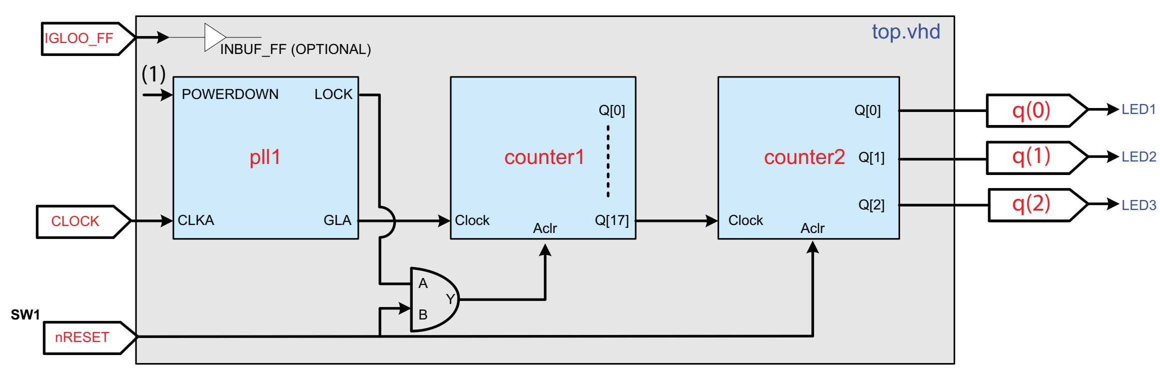

Figure 1 **• Sample Design Block Diagram**

The example design contains the following blocks:

- 1 PLL
- 2 Counters
- 1 AND gate

The example design contains the following inputs and outputs:

- Inputs: Clock and Reset

- Outputs: 3 LED Drivers

The function of the PLL in the design is to slow down the oscillator clock frequency from 50 MHz to 0.75 MHz. Counter1 is an 18-bit counter with clock input frequency of 0.75 MHz, which slows down the clock to 2.86 Hz. The Counter2 clock is driven by the 17th bit of the Counter1 and its 3-bit count is displayed on external LEDs.

The reset provided on the board is used for Asynchronous clear (Aclr) signal. The Lock signal and the nRESET are connected to a logical AND gate to generate the Aclr signal to the Counter1. This way, the Counter1 is reset on power-up by the PLL Lock signal, and you have the option to reset the device using an external reset if needed.

Note: Counter2 is reset by external nRESET only, so that we can see the Flash*Freeze (FF) demo. The LEDs driven by Counter2 would restart from the same count, as we get in and out of Flash Freeze using the external switch SW3.

When the FPGA is in Flash*Freeze mode, the lock signal is LOW. If we use a logically ANDed reset for Counter2, this would have reset Counter2, clearing the data in registers.

1 – Tutorial

This tutorial provides step-by-step instructions for developing and programing a design on the IGLOO nano Starter Kit board.

Step 1 - Create a New Project

1. From the **Start** menu, choose **Programs > Libero IDE v9.0 > Libero IDE v9.0**. The Libero IDE Project Manager opens (Figure 1-1).

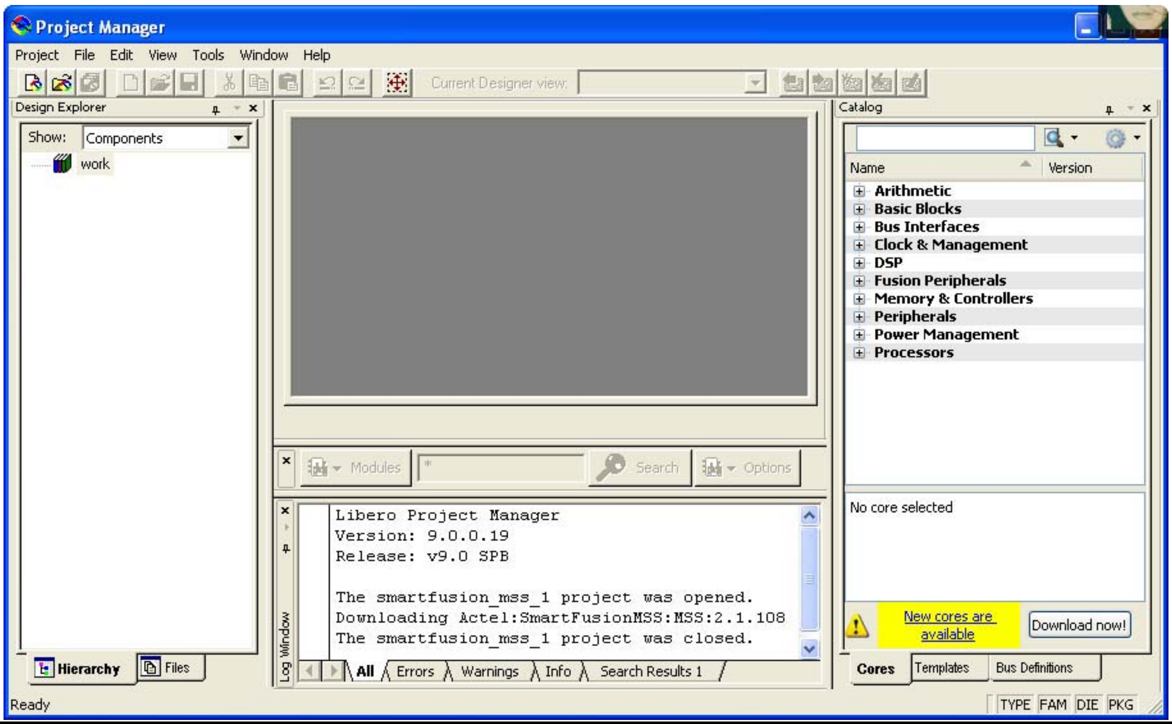

Figure 1-1 **• Libero IDE Project Manager**

2. From the **Project** menu, choose **New Project**.
3. Enter the following values (as shown in Figure 1-2), and click **Next**.

   Project Name: **Libero_tutorial**

   Project Location: **c:\Actelprj\Libero_tutorial**

   Preferred HDL Type: **Verilog** or VHDL (this tutorial uses Verilog but either option will return the same results)

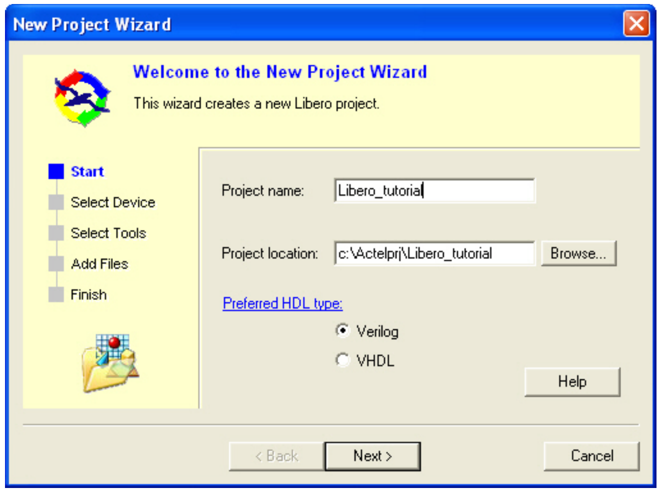

Figure 1-2 • **Libero IDE New Project Wizard**

4. Enter the following values for Family, Die, and Package (as shown in Figure 1-3) and click **Next**:
   Family: **IGLOO**
   Die: **AGLN250V2Z**
   Package: **100VQFP**

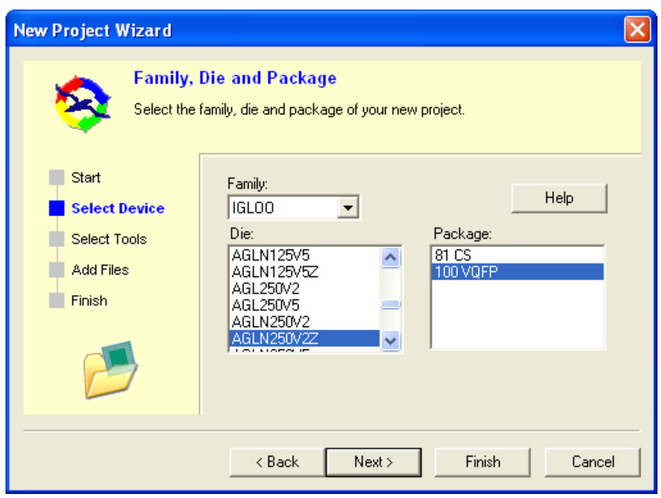

Figure 1-3 • **Select the Family, Die and Package**

5. Select the tools you wish to use in the Libero IDE (Figure 1-4). If the Libero IDE cannot locate a tool it displays a '**?**'. If necessary, click the tool and click the **Edit** button to update the tool profile. Click **Next** to proceed.

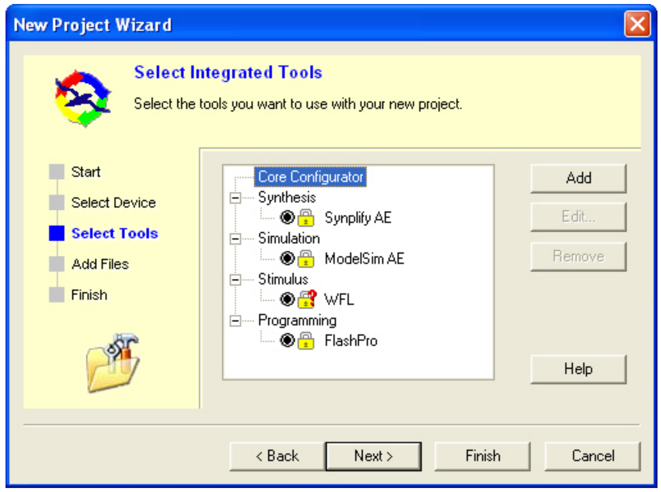

***Figure 1-4* • Select Integrated Tools in the New Project Wizard**

Note: With Libero IDE v9.0 and onwards you will see a '?' for WFL, unless you had an earlier version of Libero IDE (such as v8.6) installed on the same PC.

6. Click **Next** in the **Add Files to your Project** dialog box.

Note: You can add files to your new project by clicking **Add** and selecting the types of files you wish to add. We will not add any files for this tutorial.

7. Review your project information (Figure 1-5). Click **Back** to return to any step of the Wizard and correct information in your project. Click **Finish** to close the Wizard and create your new project.
8. Click **Save All** to save your new project.

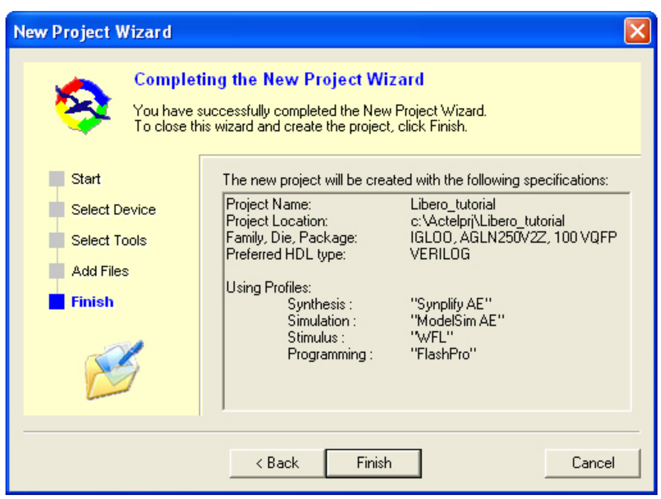

***Figure 1-5* • Review Your Project Settings**

When you create a new project the Libero IDE Project Manager displays the Design Entry, synthesis, simulation, and programming tools (Figure 1-6).

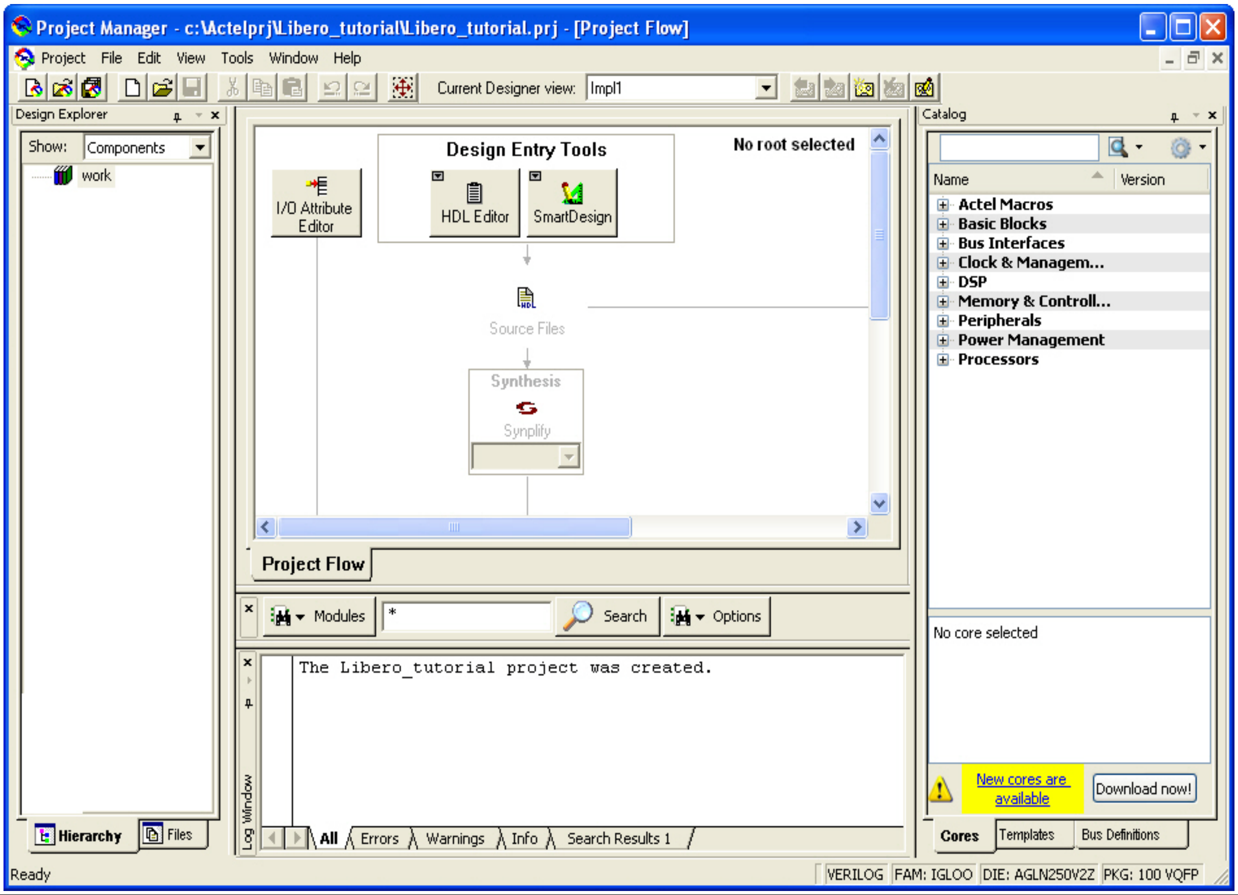

***Figure 1-6* • Libero IDE Project Manager**

The main components of the Project Manager are:

Design Explorer

This area provides a hierarchical listing of all the blocks (components) in your design

To view specific files in your design that have not been designated as blocks, such as HDL files, from the **Show** pull-down menu choose **Modules**.

Project Flow

This window offers design flow navigation, enabling you to quickly invoke the tools you need, when you need them. Your design flow status is tracked and displayed graphically in this window as you develop your design.

Catalog

The Catalog provides a browsable listing of all available macros and IP cores available for the selected device family.

Log Window

This window actively displays important information about your design as you progress through the various design stages. Errors and warnings can be viewed by selecting individual tabs.

Information Window

This window offers links to new features and lists important properties of your design, pertinent to the active design stage.

Step 2 - Creating Counter and PLL Designs

The blocks required in the design can be created using the Catalog. First you will create a PLL block and the counter blocks, and then glue them together by creating a top level HDL module.

To create a PLL and counter block:

1. From the **Catalog**, in the **Cores Tab**, under **Clock & Management**, double-click **PLL- Static** (see Figure 1-7).

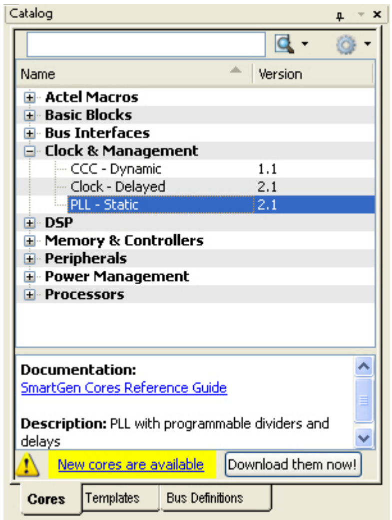

Figure 1-7 **• Project Manager Catalog - Cores Tab**

2. Enter the following in the Static PLL: Create Core dialog box (as shown in Figure 1-8):

- **Input Clock (CLKA) Frequency**: 50 MHz
- **Input Clock (CLKA) Source**: External I/O
- **Primary Output Frequency**: 0.75 MHz
- **VCO Phase Shift for Primary**: 0 deg
- **Additional Output Delay**: 0 ns

Hover your mouse pointer over a field to view a tooltip description of the field.

Accept the default settings for any values not listed above.

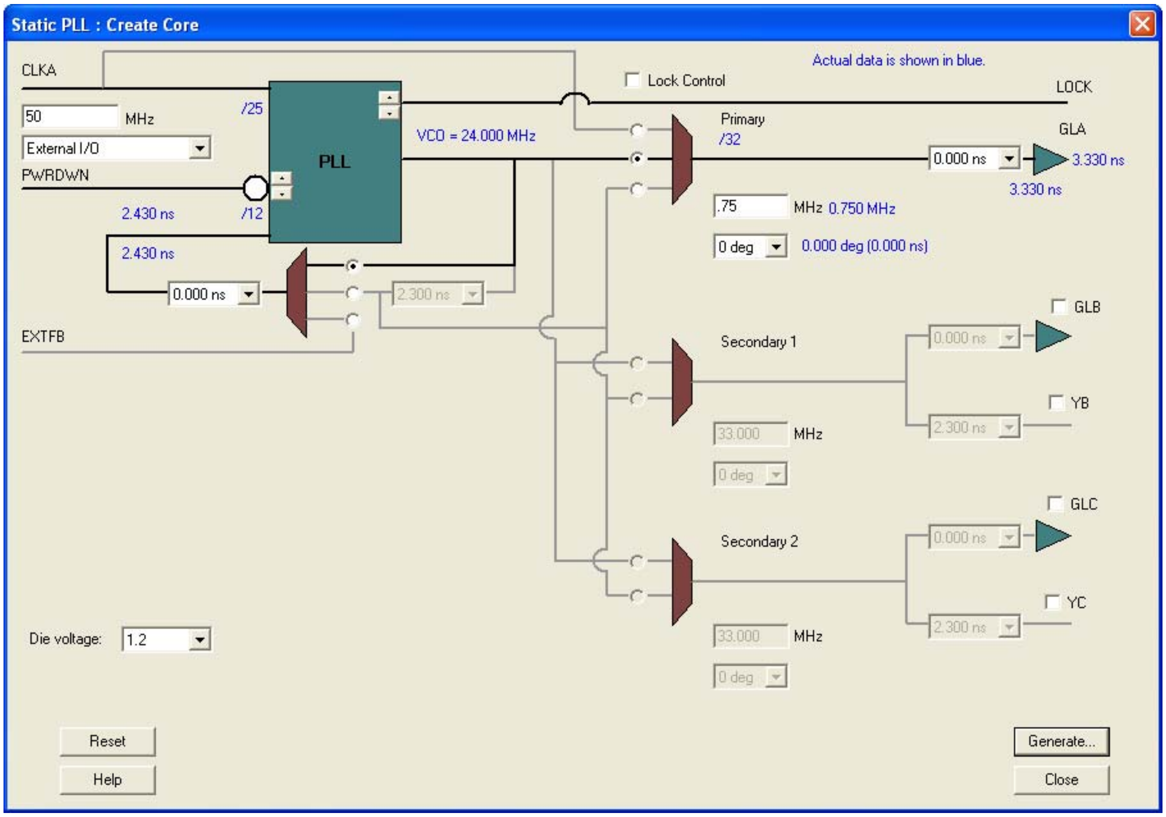

Figure 1-8 • **Static PLL Configuration**

3. Click **Generate**. The Generate Core dialog box opens.
4. Name the core **CLKGEN** (as shown in Figure 1-9) and click **OK**.

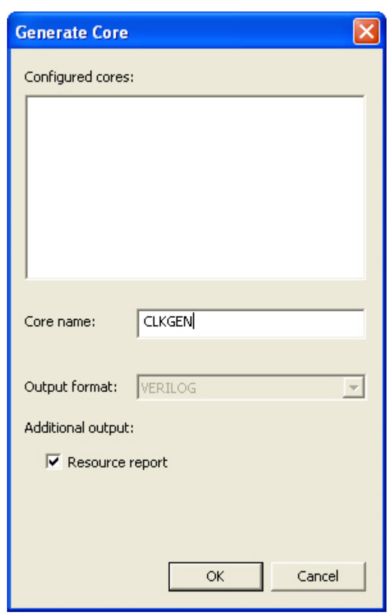

Figure 1-9 • **Generate Core Dialog Box**

A component named CLKGEN is now visible in the Design Explorer Hierarchy tab and the Files tab.

5. In the **Catalog**, double-click **Counter** (in **Basic Blocks**) to generate the counter blocks for this tutorial (Figure 1-10).

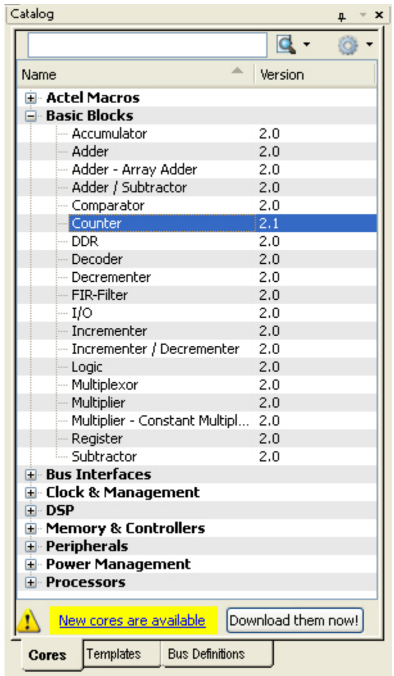

Figure 1-10 **• Counter in the Catalog**

6. The Counters: Create Core dialog box opens. Enter the following values; accept defaults for any values not specified (Figure 1-11).

   Width: 18 bits (this will reduce the frequency to 2.88 MHz)

   Async Clear: Active Low

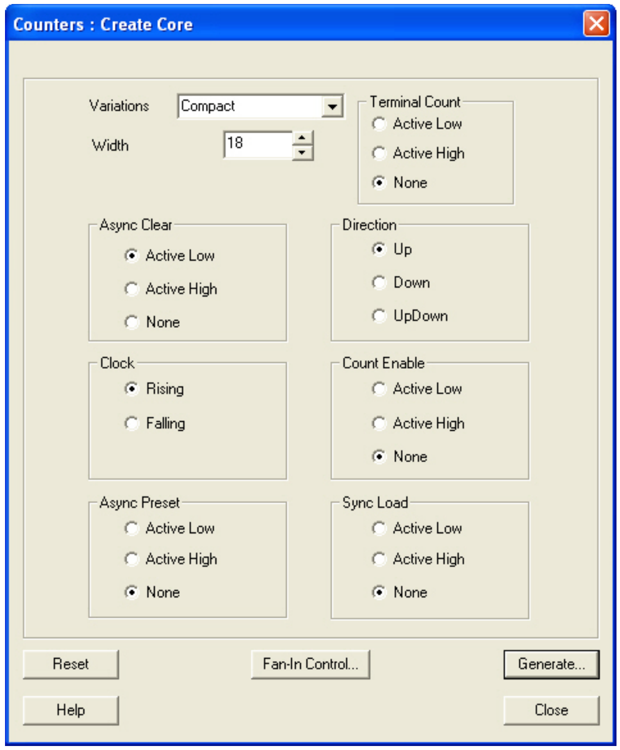

Figure 1-11 • **Counters: Create Core Dialog Box**

7. Click **Generate**. Enter the Core name **Counter1** and click **OK** to continue.
8. Double-click **Counter** in the Catalog to create another counter to drive the LEDs on the development board. Enter the following values; accept defaults for any values not specified:

   Width: 3 bits

   Async Clear: Active Low
9. Click **Generate**. Enter the Core name **Counter2** and click **OK** to continue.

   The PLL and counters are visible on the Project Manager Hierarchy tab and Files tab (Figure 1-12).

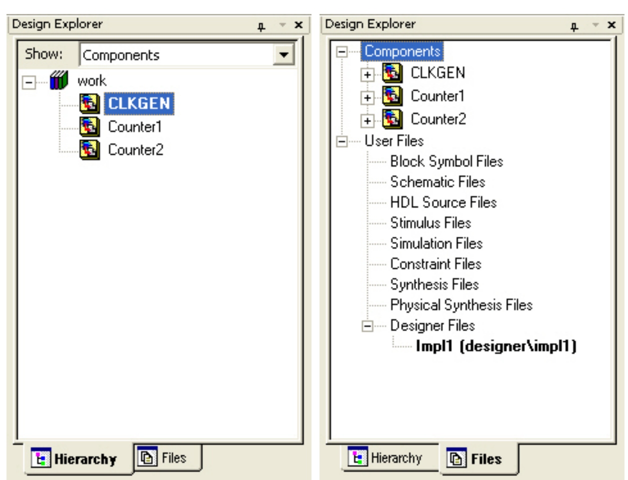

Figure 1-12 • **Project Manager Hierarchy and Files Tabs**

10. Click the **Save Project** button to save your project.

Step 3 - Creating a Top Level with SmartDesign

The next step is to connect the components and make the top-level signal connections. This step can be done multiple ways, including writing a VHDL or Verilog description. In this tutorial you will use Microsemi's SmartDesign tool to create the top level design.

To create a top level with SmartDesign:

1. In the **Project Manager**, click the **SmartDesign** button (under Design Entry Tools, Figure 1-13). The New file dialog box opens with a SmartDesign component selected.

Figure 1-13 **• Design Entry Tools in the Project Manager**

2. Enter **Top** in the name field and click **OK** to continue (Figure 1-14).

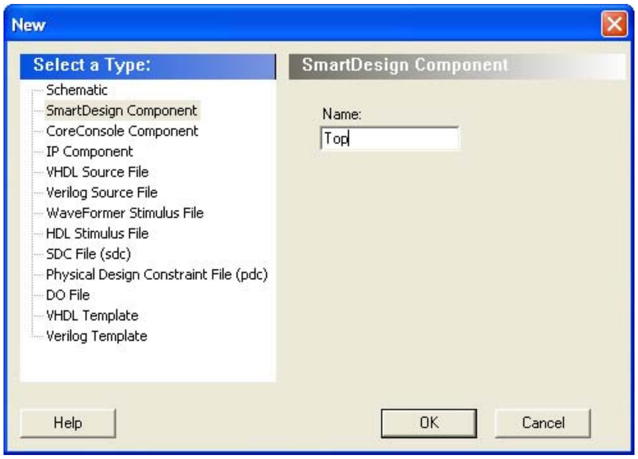

Figure 1-14 **• New SmartDesign Component Dialog Box**

3. Right-click the PLL component **CLKGEN** in the Hierarchy tab and choose **Instantiate in Top** to add it to the SmartDesign Canvas (Figure 1-15).
4. Right-click **Counter1** in the Hierarchy tab and choose **Instantiate in Top**.
5. Right-click **Counter2** in the Hierarchy tab and choose **Instantiate in Top**.

You can Shift+click to select all three components, right-click, and choose Instantiate in Top to instantiate them all at once.

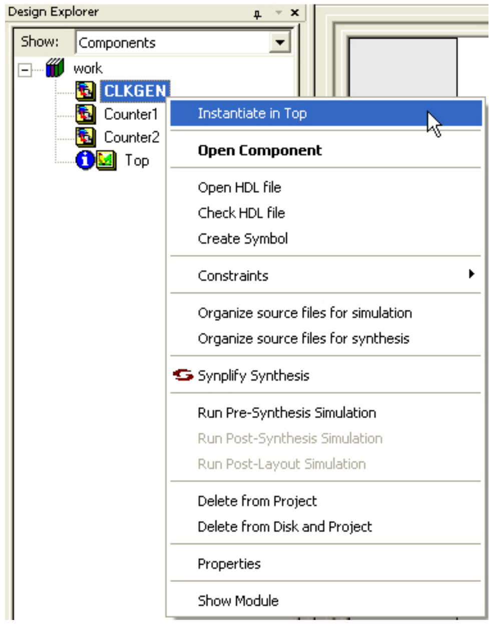

***Figure 1-15* • Instantiate Components to SmartDesign Canvas - CLKGEN**

6. In the Catalog, **right-click** the **AND2** macro and choose **Instantiate to Top** to add the AND2 cell to the SmartDesign Canvas. The AND2 macro is in the **Catalog > Cores** tab under **Actel Macros**.
7. In the SmartDesign **View** menu, choose **Maximize Work Area** to expand the SmartDesign Canvas. This will make it easier to work in SmartDesign.

8. In the **Canvas** menu, choose **Auto-arrange Instances**. After adding and arranging the components the SmartDesign Canvas resembles Figure 1-16. Click and drag your components to arrange them if you are not satisfied with the auto-arrange.

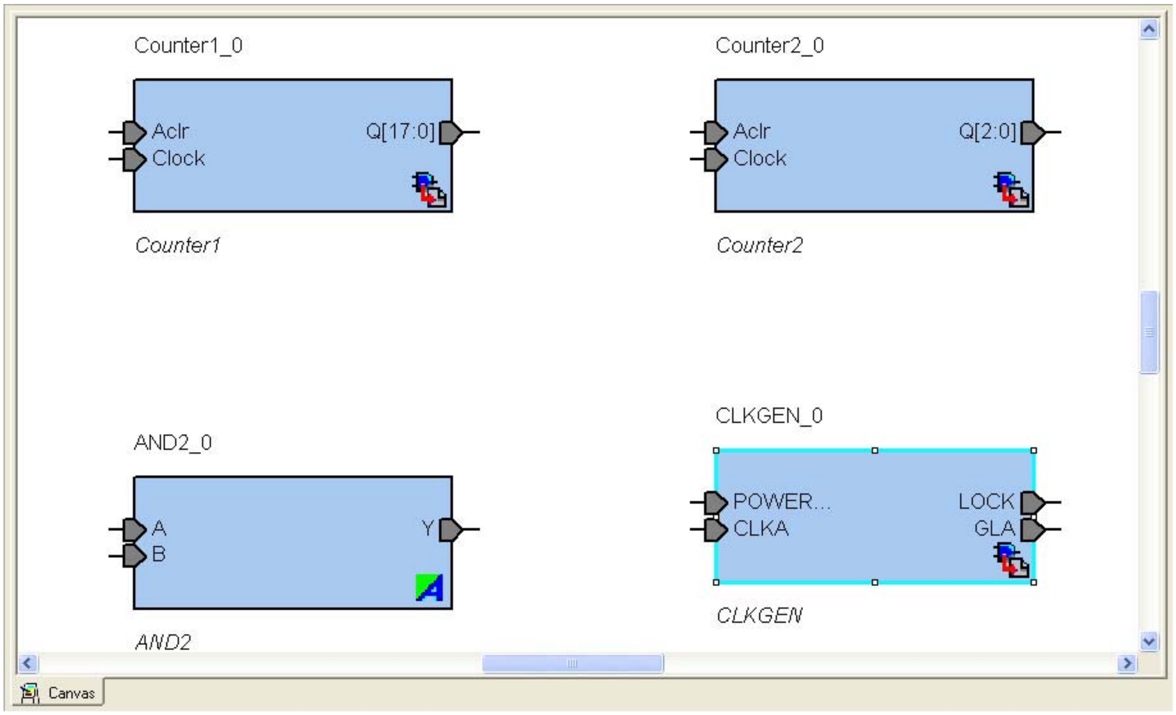

Figure 1-16 **• SmartDesign Canvas with Components**

9. Connect the output LOCK port of the CLKGEN_0 component to the input A port of the AND2_0 component as follows:

   Click the **LOCK** port on the **CLKGEN_0** component.

   Hold the **CTRL** key and select the **A port** of the **AND2_0** component.

   While holding the **CTRL** key, right-click and choose **Connect**.
10. Repeat the procedure above to make the following connections:

   CLKGEN_0:GLA -> Counter1_0:Clock

   AND2_0:Y -> Counter1_0:Aclr
11. In the **Canvas** menu, choose **Auto-arrange Instances**.

12. Right-click the **CLKGEN_0:CLKA** port and choose **Promote to Top Level** (Figure 1-17) to connect the port to the top level. This port will be connected to a dedicated input pin on the AGLN250 FPGA.

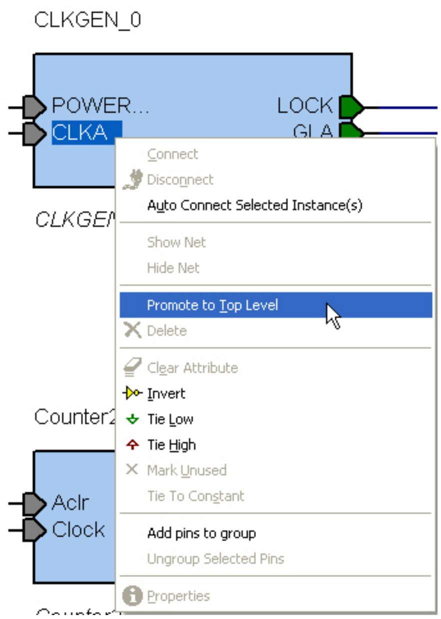

***Figure 1-17* • Connect CLKGEN_0:CLKA to Top Level**

13. Right-click the **AND2_0:B** port and choose **Promote to Top Level** to connect the port to the top level.
14. Right-click the **Counter2_0:Q[2:0]** port and choose **Promote to Top Level** to connect the port to the top level.
15. Right-click **top-level port B** and choose **Modify Port** to change the name. Enter the new name as shown in Figure 1-18:

    Name: NSYSRESET

    Direction: Input (default)

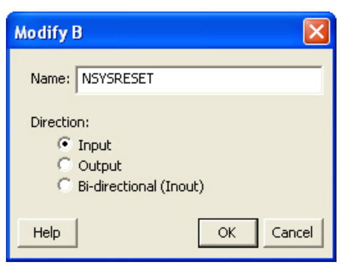

***Figure 1-18* • Modify Port Name**

16. Right-click **Counter1_0 port Q[17:0]** and choose **Add Slice**. The Add Slice dialog box appears.

17. Enter **17:17** in the Add Slice dialog box and click **OK** to continue (Figure 1-19).

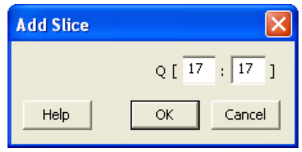

Figure 1-19 **• Add Slice Dialog Box**

18. Click the **+** sign on **port Q** of **Counter1_0** to expose the slice created in the previous step. Connect port **Q[17]** of **Counter1_0** to the clock input (**Clock**) of **Counter2_0** (Figure 1-20).

    To connect Counter1_0:Q[17] to Counter2_0:Clock:

    Click **Counter1:0Q[17]** to select it.

    Control-click **Counter2_0:Clock**.

    Right-click **Counter2_0:Clock** and choose **Connect**.

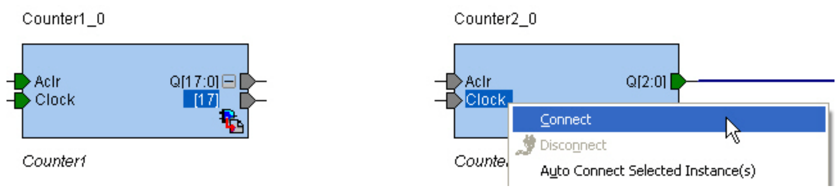

Figure 1-20 **• Connecting a Sliced Port to a Clock Input**

19. Right-click the **CLKGEN_0:POWERDOWN** port and choose **Tie High** to tie it to logic 1.
20. Connect the top level **NSYSRESET** port to **Counter2_0:Aclr** component as follows:

    Click the **Aclr** port on the **Counter2_0** component.

    Hold the **CTRL** key and select the **NSYSRESET** of the **Top level** component.

    While holding the **CTRL** key, right-click and choose **Connect**.
21. Click the **Restore Work Area** button to view the Catalog.
22. In the **Catalog**, under **Actel Macros**, double-click **INBUFF_FF** to instantiate it.

    The input of the INBUFF_FF controls when the device goes into Flash*Freeze mode (active low input) and must be promoted to the top level (this is the default setting). Leave the output of the macro INBUFF_FF floating as it is hardcoded into the silicon to control FlashFreeze mode.

After connecting the components on your SmartDesign your Canvas should resemble Figure 1-21. Click and drag the components if you are not satisfied with the auto-arrange.

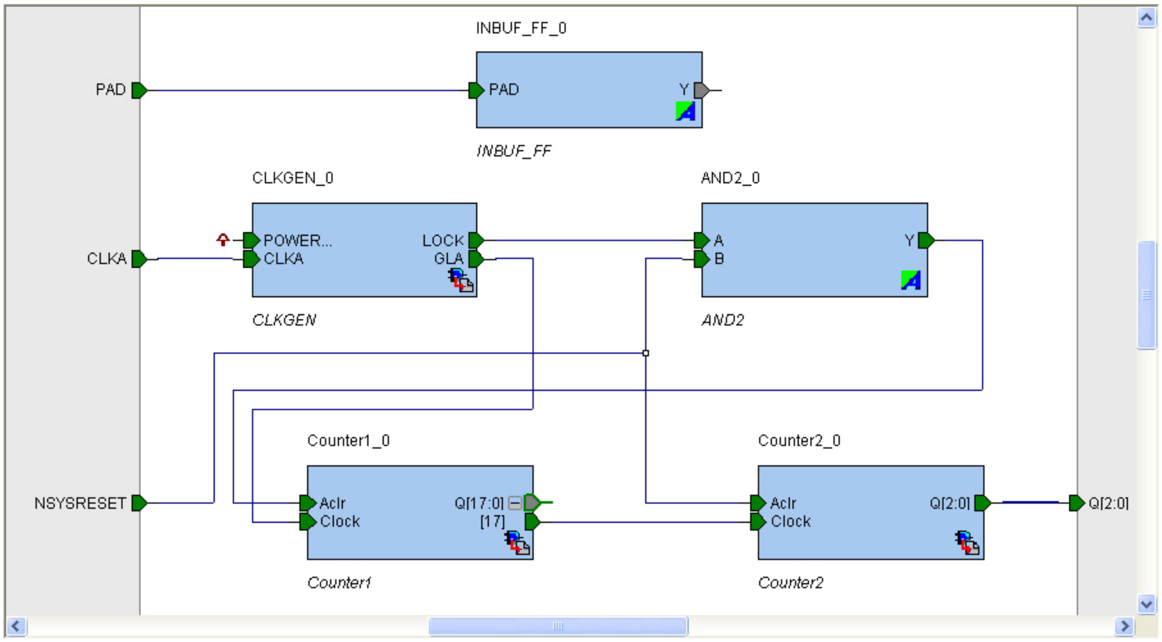

Figure 1-21 • **SmartDesign Canvas After Connecting Components**

23. Click the **Save Project** button to save your design.
24. From the **SmartDesign** menu, choose **Check Design Rules** to run the Design Rule Checker. The checker ensures that there are no errors in the design.

    Ignore the warning about the floating driver on Counter1_0. Correct any other reported errors.
25. From the **SmartDesign** menu choose **Generate Design**. Confirm that the design was generated successfully (Figure 1-22). Click OK to continue.

Figure 1-22 • **Design Generated Successfully**

26. From the **View** menu, choose **Restore Work Area** to restore the work area to its original size.

The component Top is now visible in the Project Manager Hierarchy and Files tabs. The Hierarchy tab displays the full design hierarchy (Figure 1-23).

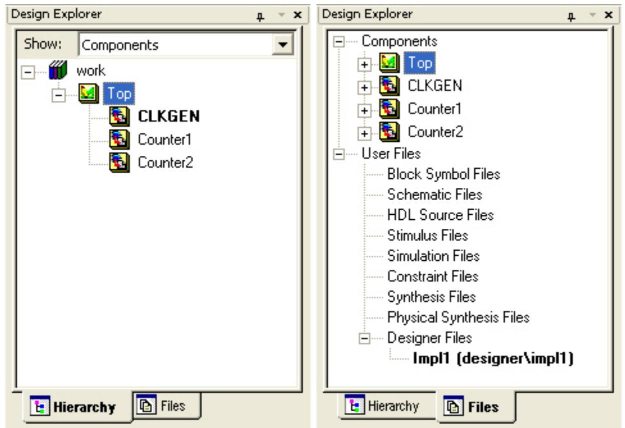

***Figure 1-23* • Top Component in Hierarchy and Files Tabs**

27. From the **File** menu, choose **Close** to close SmartDesign.

Step 4 - Modifying the SmartDesign Testbench

SmartDesign creates a testbench that you can use to simulate the design. The testbench has a default clock frequency of 10 MHz. In this step you will modify the testbench to generate a 50 MHz clock.

To modify the SmartDesign testbench:

1. Click the + sign next to Top on the Project Manager Files tab to expand the hierarchy.
2. Click the + next to Stimulus Files to expose the testbench (testbench.v(hd)) (Figure 1-24).

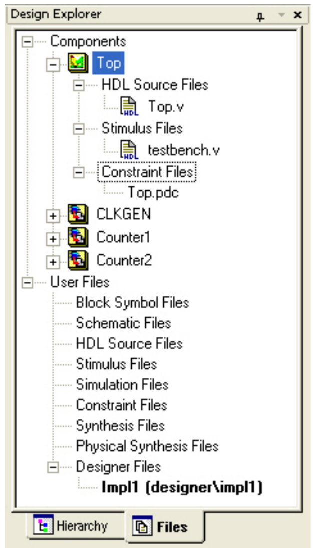

Figure 1-24 **• SmartDesign Testbench on Project Manager Files Tab**

3. Double-click **testbench.v(hd)** to open the testbench in the Libero IDE text editor. From the **File** menu, choose **Save testbench.v(hd) As**. Save the file with the following values:

   Save in: C:\Actelprj \Libero_tutorial\stimulus

   Filename: usr_testbench.v(hd)

   The file usr_testbench.v(hd) is visible in the Design Hierarchy Files tab under **User Files > Stimulus Files** (Figure 1-25).

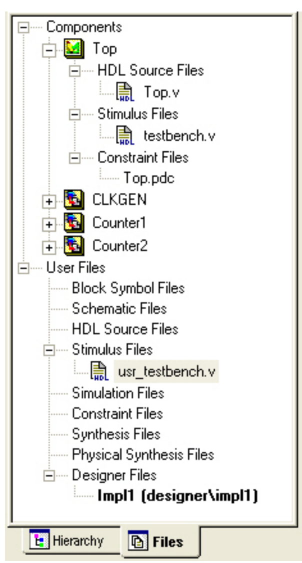

Figure 1-25 **• usr_testbench.v(hd) in Design Hierarchy Files Tab**

4. Double-click **usr_testbench.v(hd)** to open it in the Libero IDE text editor.
5. Change the **SYSCLK_PERIOD** constant (or parameter) from **100** to **20**.
6. Click **Save All** to save your changes.
7. From the **File** menu, choose **Close** to close the usr_testbench.v(hd) file in the editor.
8. Confirm that Top is set as root in the Project Manager Project Flow window. If it is not, right-click **Top** in the **Hierarchy tab** and choose **Set as Root** (Figure 1-26).

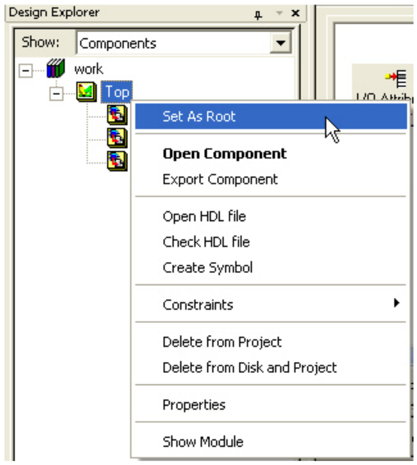

***Figure 1-26* • Set As Root in Hierarchy Tab**

9. Right-click **Top** in the Hierarchy tab and choose **Organize Stimulus**. The Organize Stimulus dialog box appears (Figure 1-27).

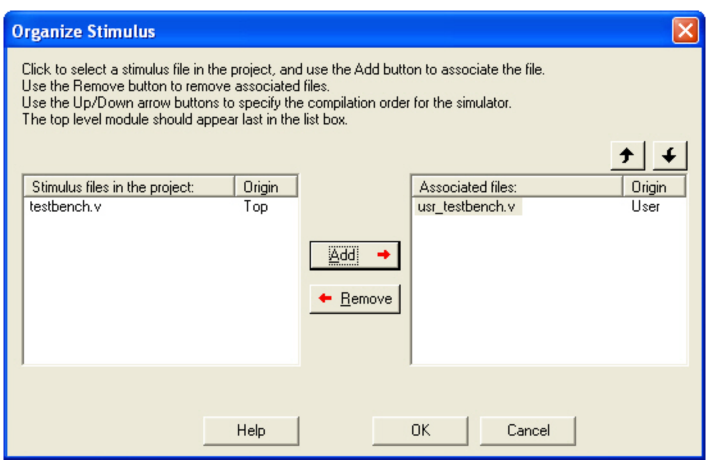

***Figure 1-27* • Organize Stimulus Dialog Box**

10. Click **testbench.v(hd)**, then click **Remove**.
11. Click **usr_testbench.v(hd)** and click **Add**, then **OK**. This forces the Libero IDE to use the modified testbench for simulation.
12. Click the **Save Project** button to save your project.

Step 5 - Performing Pre-Synthesis Simulation

In this step you will perform pre-synthesis simulation of the design.

To perform pre-synthesis simulation:

1. From the **Project** menu, choose **Settings** and click the **Simulation** tab (Figure 1-28).
2. Click **Do File** in **ModelSim options** and set
   Simulation runtime: 200 ms
3. Click **Waveforms** in **ModelSim options** and set:
   Include Do file: Unchecked (default)
   Display waveforms for: DUT
   Log all signals in design: Unchecked (default)
4. Click **OK** to close the Project Settings dialog box.

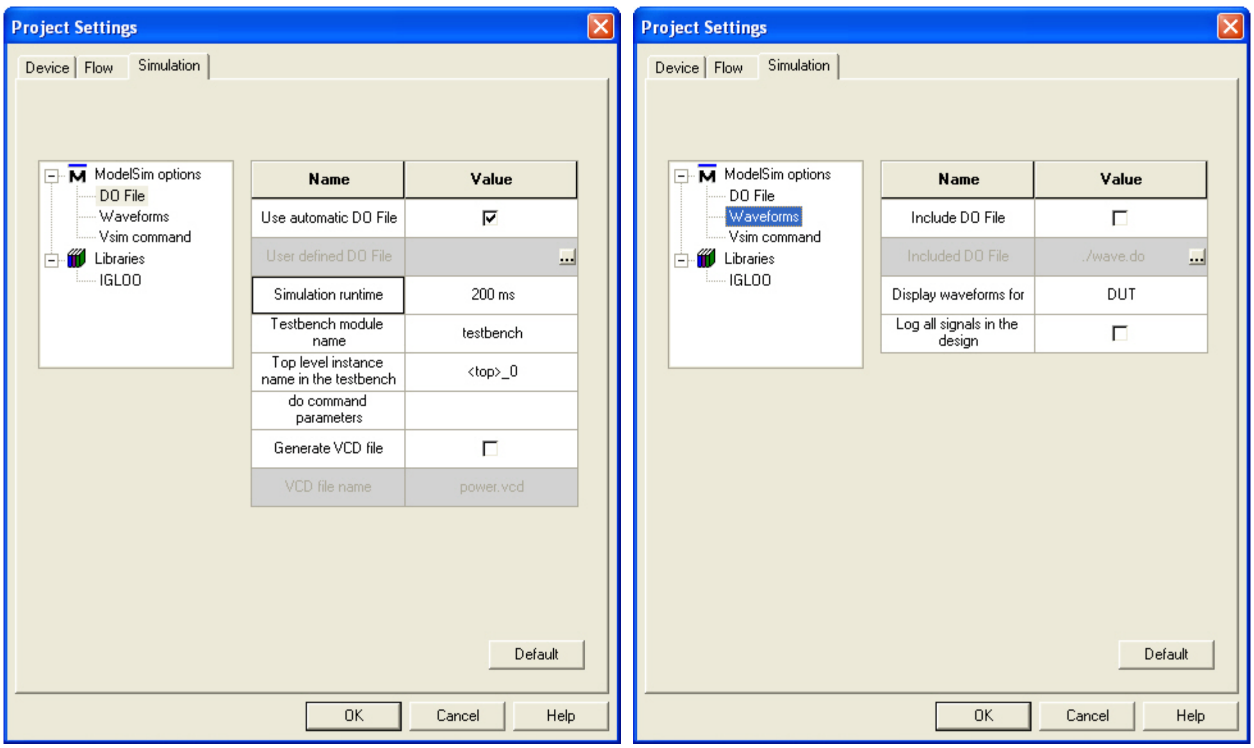

Figure 1-28 **• Libero IDE Project Settings - Simulation ModelSim Options (DO File and Waveforms)**

5. Click the **Save All** button to save your project.
6. Click the **ModelSim** button in the **Project Flow window** to launch ModelSim. ModelSim opens and automatically imports a run.do macro file that contains the links to the design files and gives simulation commands. The simulator compiles the source files and runs for 200 ms. It will take a few moments for ModelSim to finish the simulation.

   ModelSim displays the signals in the component Top along with the simulation results. (Figure 1-29)

7. Scroll in the ModelSim Wave window to verify that the two counters are working properly. You can undock the simulator to make it easier to view the signals.

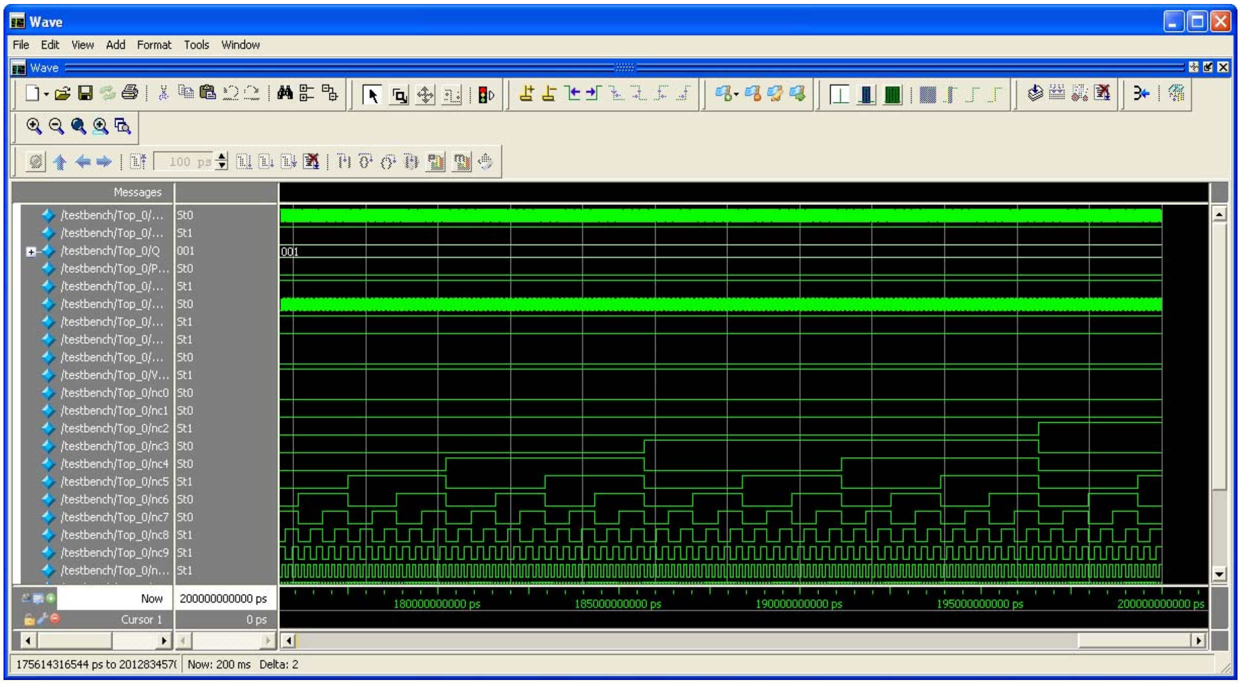

***Figure 1-29* • ModelSim Wave Window**

8. In the ModelSim Wave window, from the **File** menu choose **Quit** to close ModelSim. Click **Yes** when prompted about quitting.

Step 6 - Synthesizing the Design Using Synplify Pro

Synplify Pro compiles and synthesizes the design into an EDIF (*.edn) file. Your EDIF Netlist is then automatically translated by Libero IDE into an HDL Netlist. The resulting *.edn and *.vhd files are visible in the Project Manager Files tab under Synthesis Files.

To synthesize the design using Synplify Pro:

1. Click the **Synthesis Synplify** button in the Project Manager Project Flow window to launch Synplify Pro.
2. Change the **Frequency** in the Synplify Pro GUI to 50 MHz (Figure 1-30).
3. Click the **Run** button to map the design. When the **Ready** on the user interface in Synplify Pro changes to **Done** the design has been mapped successfully. For this tutorial you can IGNORE warnings or notes. The warnings generated by the compiler are due to the removal of an instance which is instantiated in the HDL code, but its output does not drive the outputs of the top-level design.

4. From the **File** menu, choose **Exit** to close Synplify Pro. Select **Yes** if prompted about saving changes to the project.

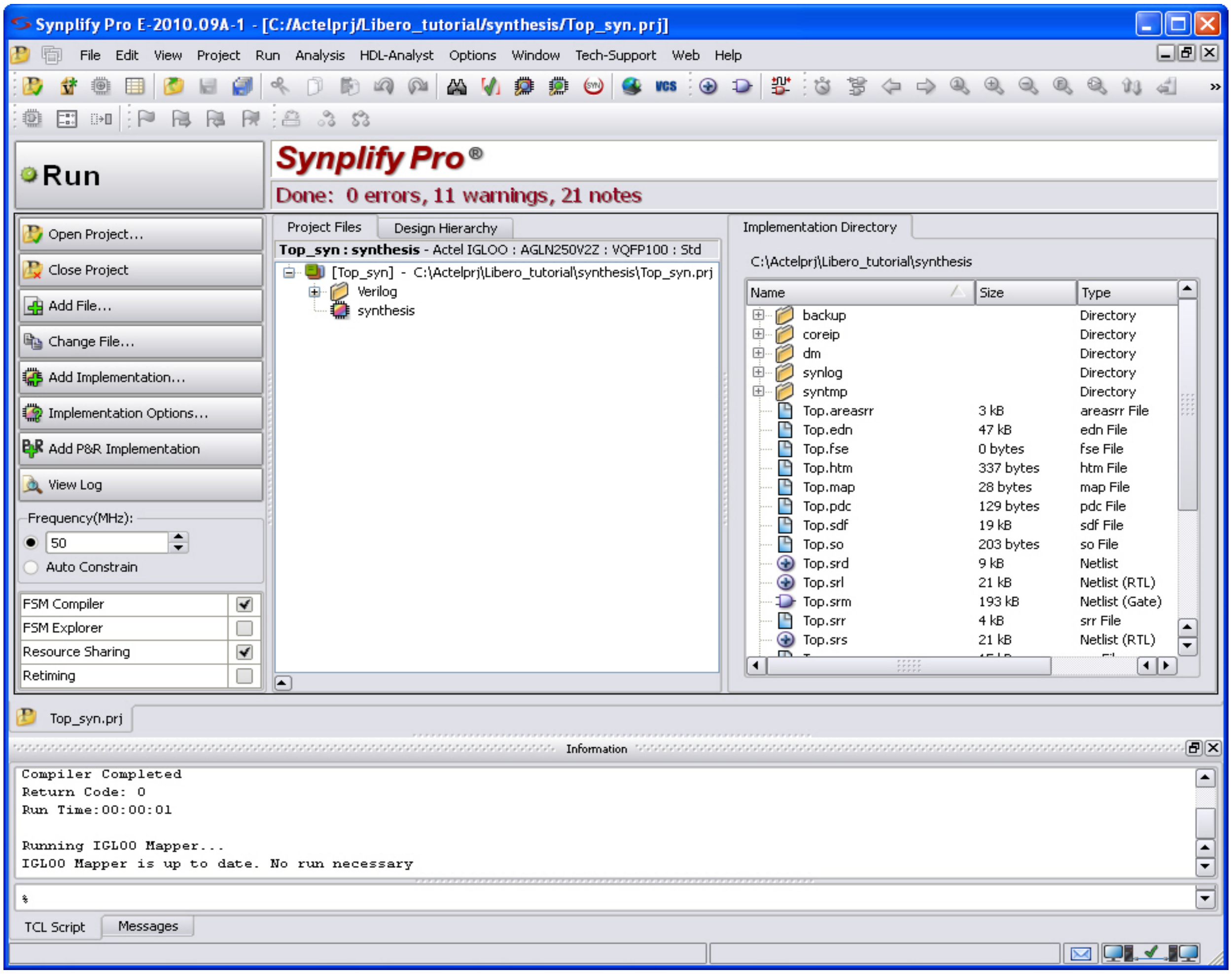

Figure 1-30 **• Synplify Pro GUI**

5. Click the **Save Project** button to save your project.

Step 7 - Implementing the Design with Designer (Place-and-Route)

The next step is to implement the design using Designer. You can use Designer to place-and-route the design into the FPGA fabric and perform various other implementation tasks such as setting I/O constraints, performing static timing analysis, and analyzing/estimating power consumption.

To place-and-route with Designer:

1. Click the **Place&Route Designer** button in the Project Manager. The Organize Constraints dialog box appears (Figure 1-31).

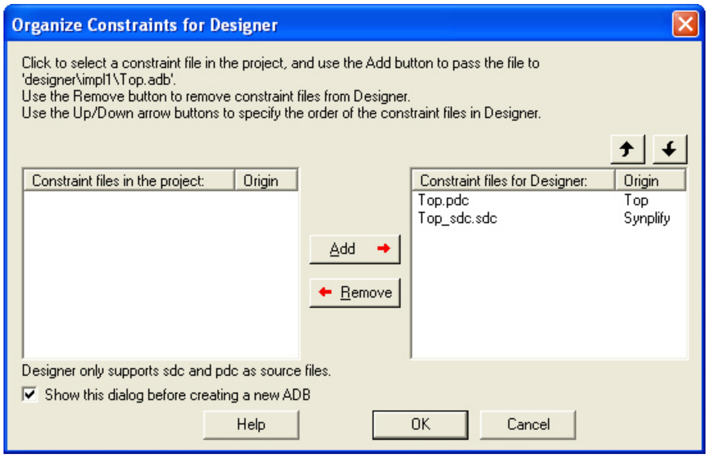

***Figure 1-31* • Organize Constraints Dialog Box**

2. Confirm **Top_sdc.sdc** appears under Constraints for Designer and click **OK**. This is a timing constraint file that was generated by Synplify
3. Accept the default die and package settings in the Device Selection Wizard and click **Next**.
4. Accept the default settings in the Device Selection Wizard - Variations dialog box and click **Next**.
5. Accept the default settings in the Device Selection Wizard - Operating Conditions dialog box and click **Finish**.

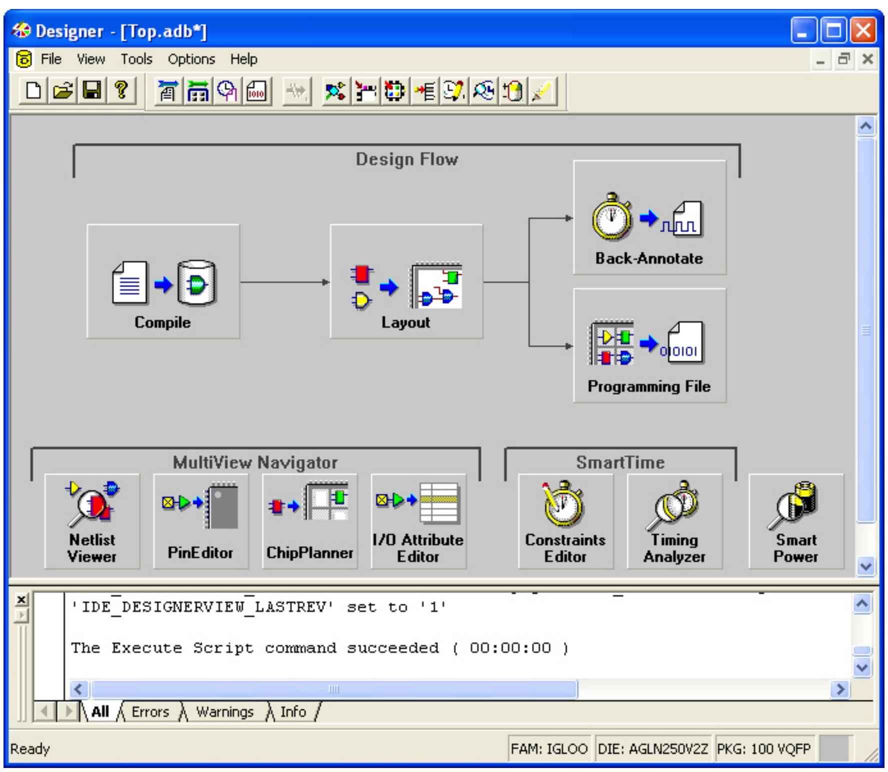

***Figure 1-32* • Designer GUI**

6. Click the **Compile** button in the Designer GUI to compile the design (Figure 1-32). Compile contains a variety of functions that perform legality checking and basic netlist optimization. It also calculates and displays the utilization of the design for the selected device.
7. In the **Compile Options** dialog box, choose **Globals Management** and enter the following (Figure 1-33):

   Promote regular nets whose fanout is greater than: Checked

   Fanout limit: 15

   Do not promote more than: 3

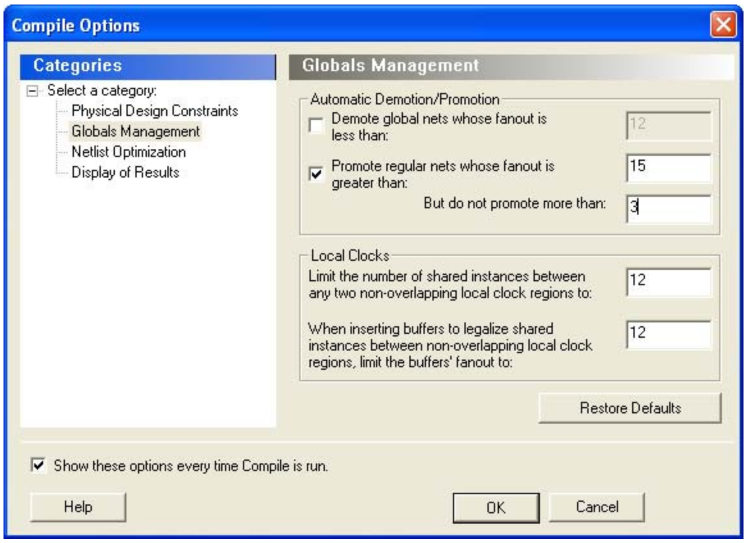

Figure 1-33 **• Compile Options - Globals Management**

8. Click **OK** to close the Compile Options dialog box and run compile. The Compile button turns green to indicate the design compiled without any errors.

   Designer includes an I/O Attribute Editor that enables you to make pin assignments and set I/O configurations for your design.
9. Click the **I/O Attribute Editor** button in the Designer GUI to open the I/O Editor. The I/O Attribute Editor opens in the MultiView Navigator (Figure 1-34).

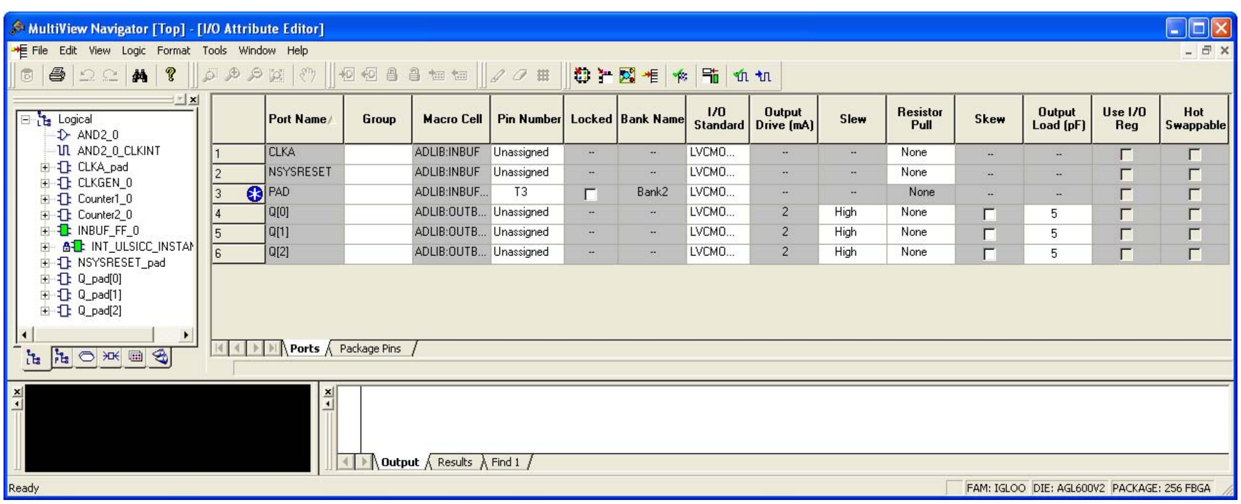

Figure 1-34 **• I/O Editor in MultiView Navigator**

10. To make a pin assignment, select the pin number for each port from the pull-down menu (Figure 1-35).

Note: The pin assignments for the IGLOO nano Starter Kit are shown in figure Figure 1-35; if you are using other boards or kits, see the schematics included with the kit for the correct port and pin assignments.

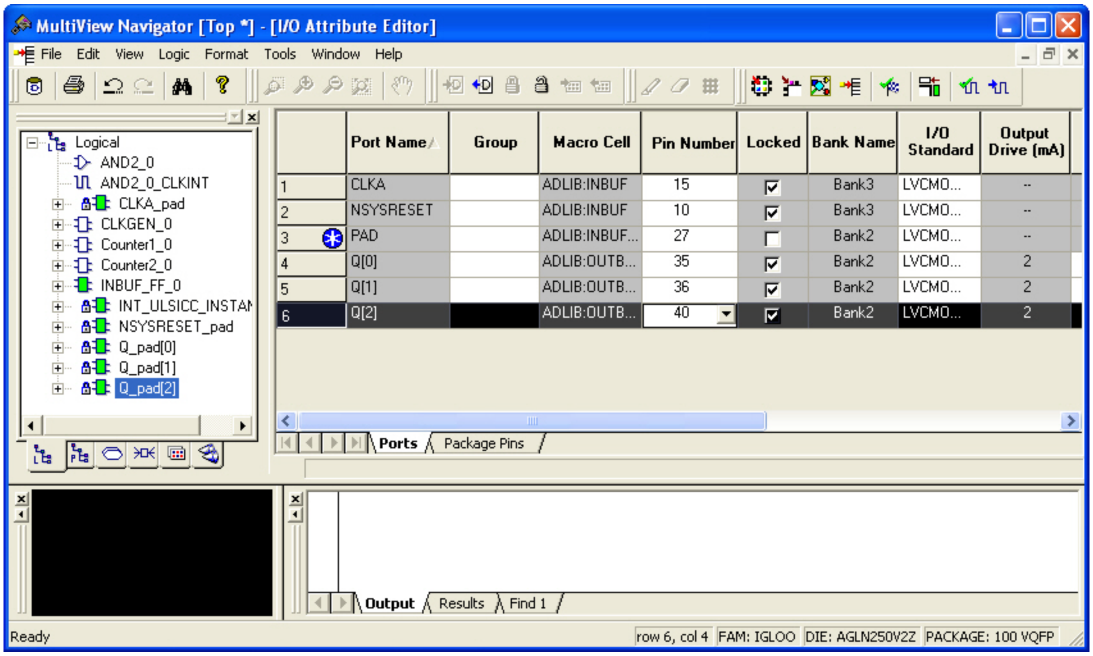

***Figure 1-35* • I/O Attribute Editor with Pin Assignments for the IGLOO Starter Kit**

11. From the MultiView Navigator **File** menu, choose **Commit and Check** to commit your changes.
12. From the MultiView Navigator **File** menu, choose **Exit** to close and continue.
13. In Designer, click the **Save** button to save your design.
14. Click **Layout** in Designer to run layout on the design.
15. Click **OK** to accept the default layout options (Figure 1-36).

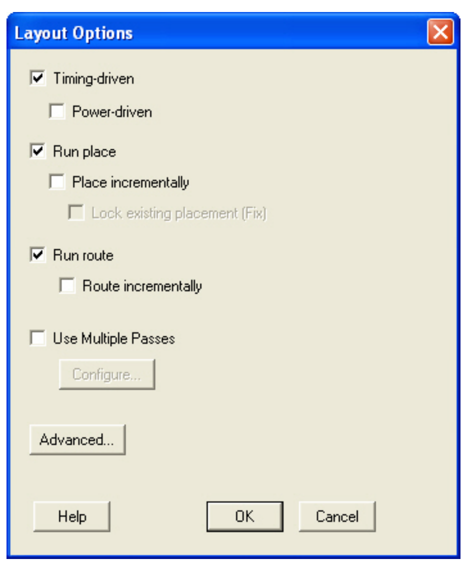

***Figure 1-36* • Layout Options in Designer**

The Layout button turns green to indicate the design completed layout without any errors (Figure 1-37).

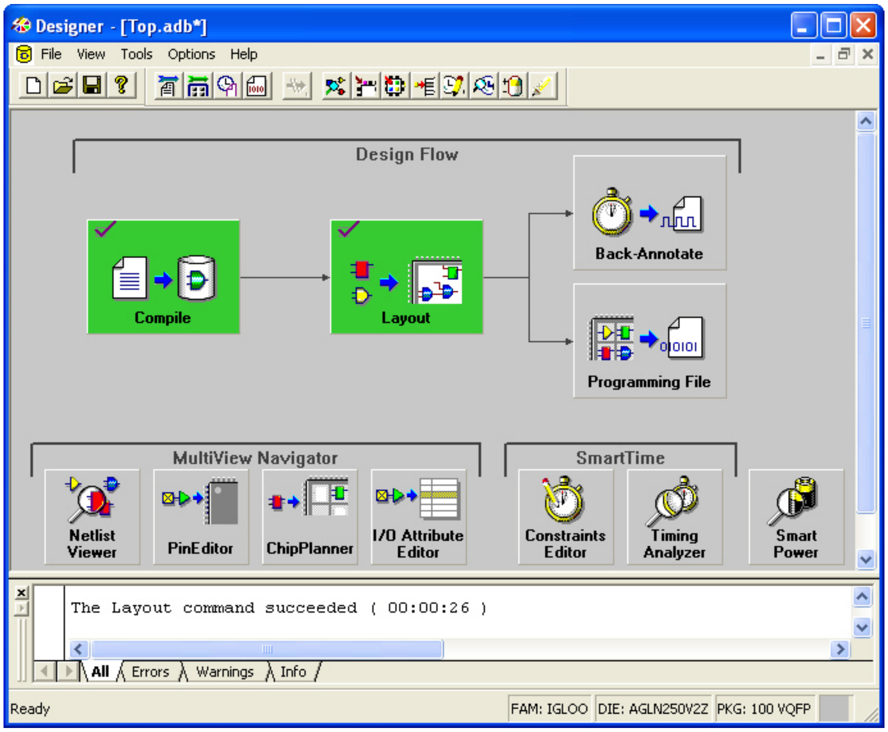

Figure 1-37 **• Designer GUI - Successful Layout**

Using SmartTime

Next we will perform static timing analysis using SmartTime. SmartTime reads your design and displays postlayout timing information (pre-layout, if invoked before place-and-route). SmartTime includes a Constraints Editor and a Timing Analyzer.

Timing constraints from Synthesis are imported into Designer by default. If you want to add additional clock constraints or over-constrain the present clock for place-and-route, you can do that in SmartTime.

To perform static timing analysis with SmartTime:

1. Click the **Constraints Editor** button to open the SmartTime Constraints Editor (Figure 1-38). You will create a clock constraint for the clock in your design.

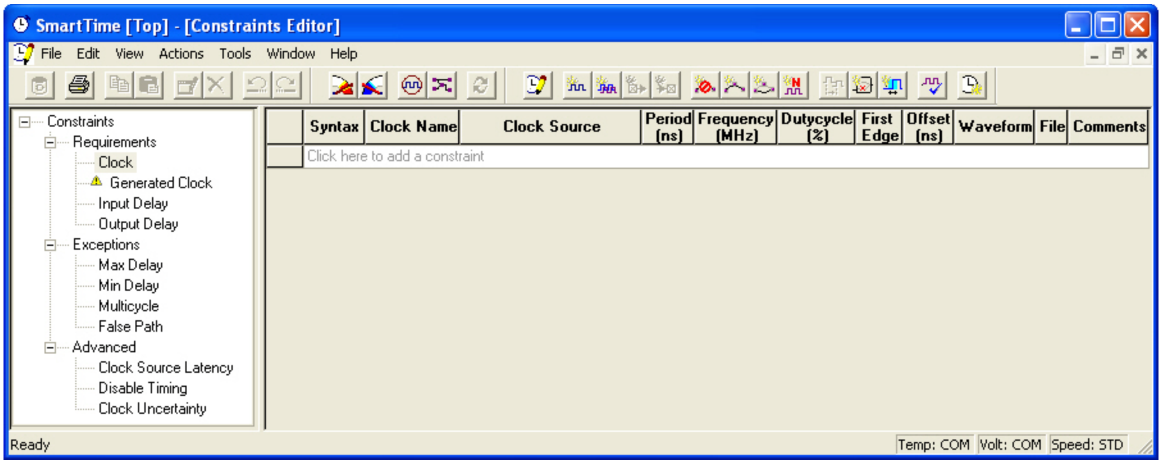

***Figure 1-38* • SmartTime Constraints Editor**

2. Right-click **Clock** in the Constraints Editor and choose **Add Clock Constraint**. The Create Clock Constraint dialog box appears (Figure 1-39).

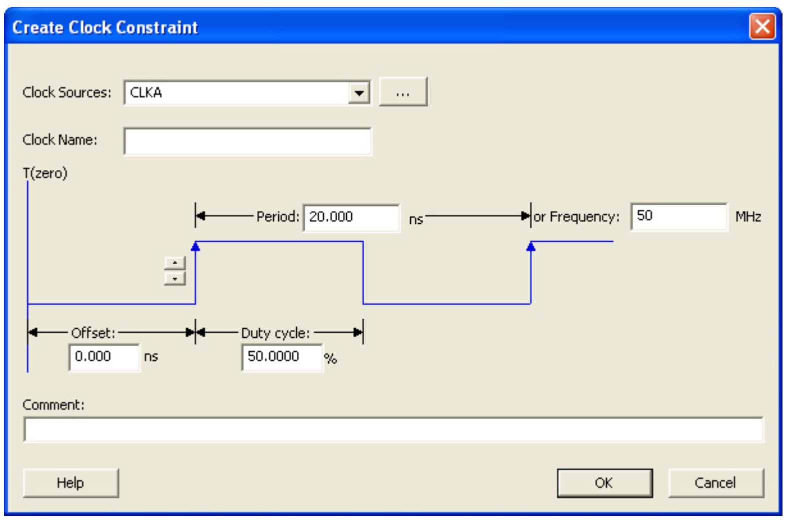

***Figure 1-39* • Create Clock Constraint Dialog Box**

3. Set the following values for your clock constraint:

   Clock: CLKA

   Frequency: 50 MHz

Click **OK** to continue. A green flag next to the name of the clock indicates that the constraint was entered correctly (Figure 1-40).

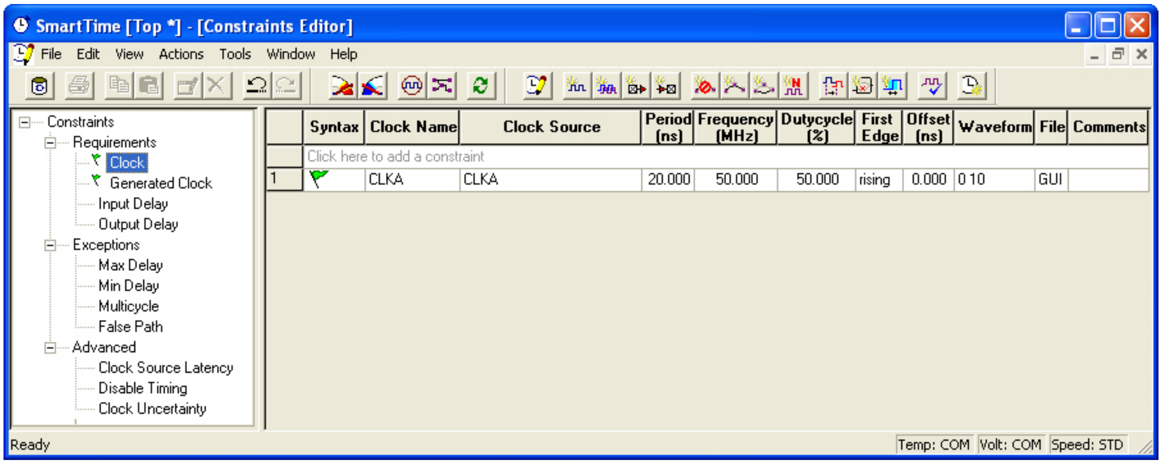

Figure 1-40 **• SmartTime Constraints Editor - Clock Constraint with Frequency of 50 MHz**

The generated clock is set automatically based on the PLL input (which is CLKA), and the PLL setting. Click **Generated Clock** in the SmartTime constraint editor to see the generated clock (Figure 1-41)

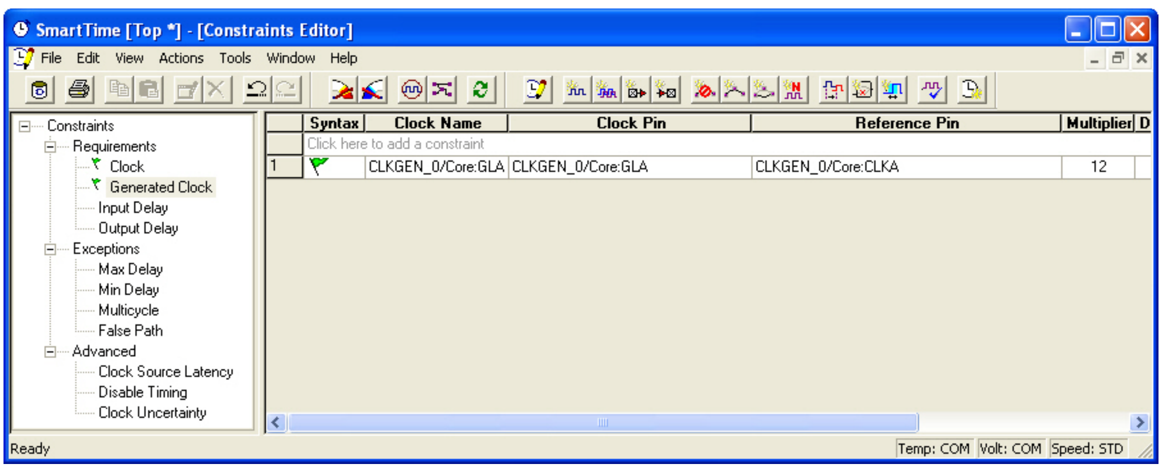

Figure 1-41 **• Automatically Generated PLL Clock in Constraints Editor**

4. From the **Tools** menu, choose **Timing Analyzer > Maximum Delay Analysis**. Select the **Register to Register** path set for the **CLKGEN_0/Core:GLA** clock domain. Observe that the Slack column is positive, indicating that there are no timing violations (Figure 1-42).

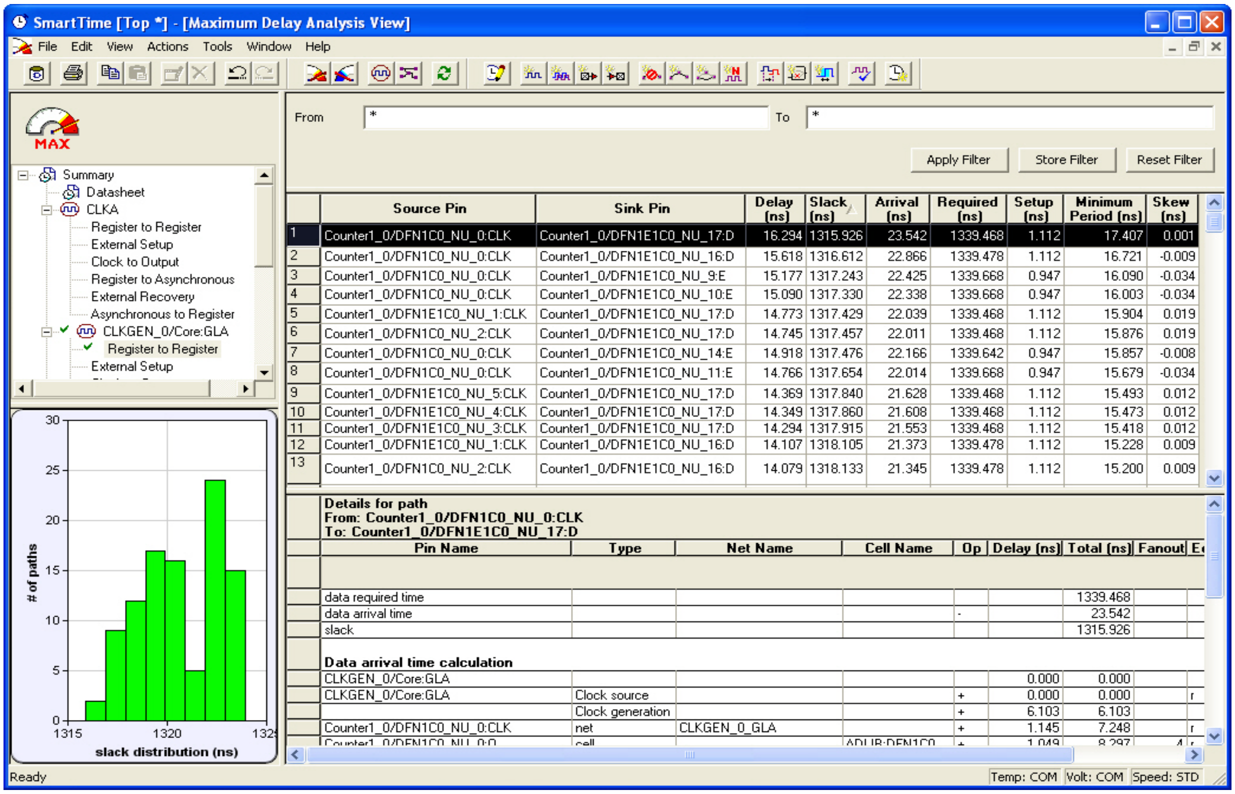

***Figure 1-42* • SmartTime Max Delay Timing Analysis**

Click one of the source pins to view detailed timing analysis for the selected path in the Path Details.

5. From the **Tools** menu, choose **Timing Analyzer > Minimum Delay Analysis**. Select the **Register to Register** path set for the **CLKGEN_0/Core:GLA** clock domain. Note that there are

no hold violations, as the slack column is positive. Double-click a path to view path details (Figure 1-43).

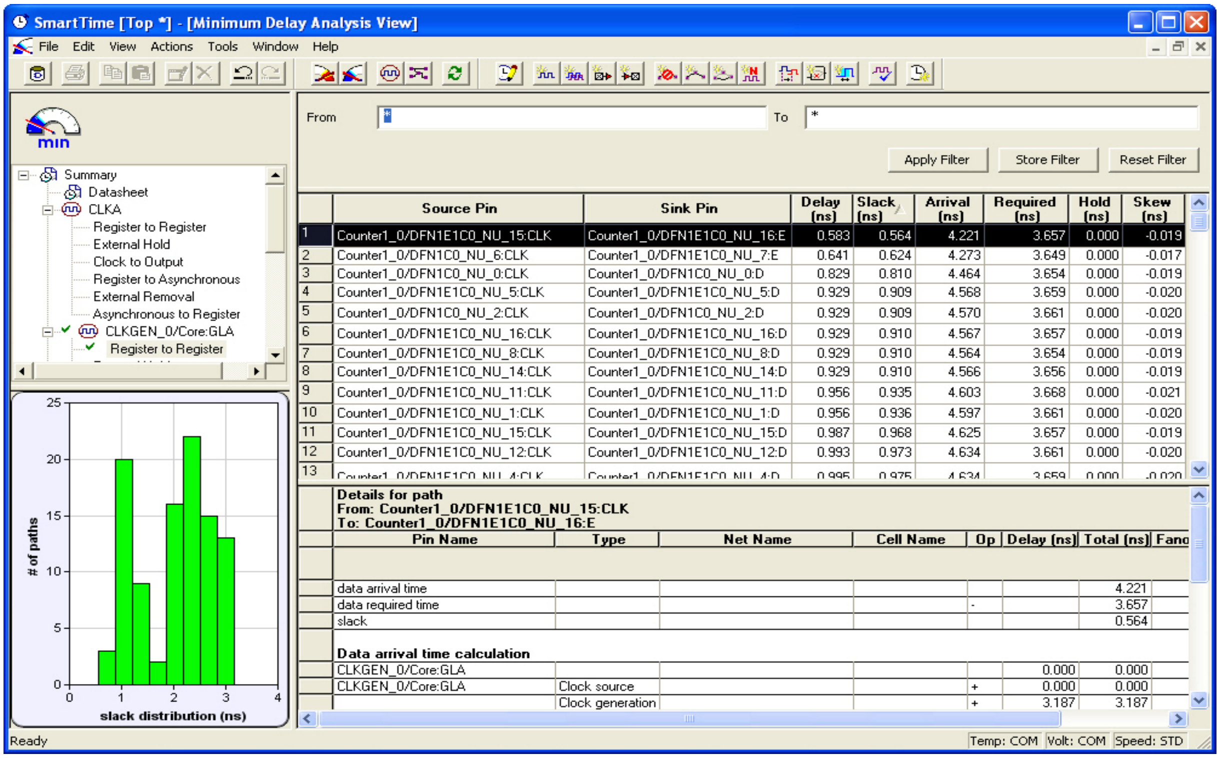

Figure 1-43 **• SmartTime Min Delay Timing Analysis**

6. From the SmartTime **File** menu, choose **Commit** to commit your changes.
7. From the SmartTime **File** menu, choose **Exit** to close SmartTime.

Using SmartPower (Optional)

SmartPower (Figure 1-44) enables you to estimate the power consumption in your design. This enables you to make adjustments (where possible) to reduce it.

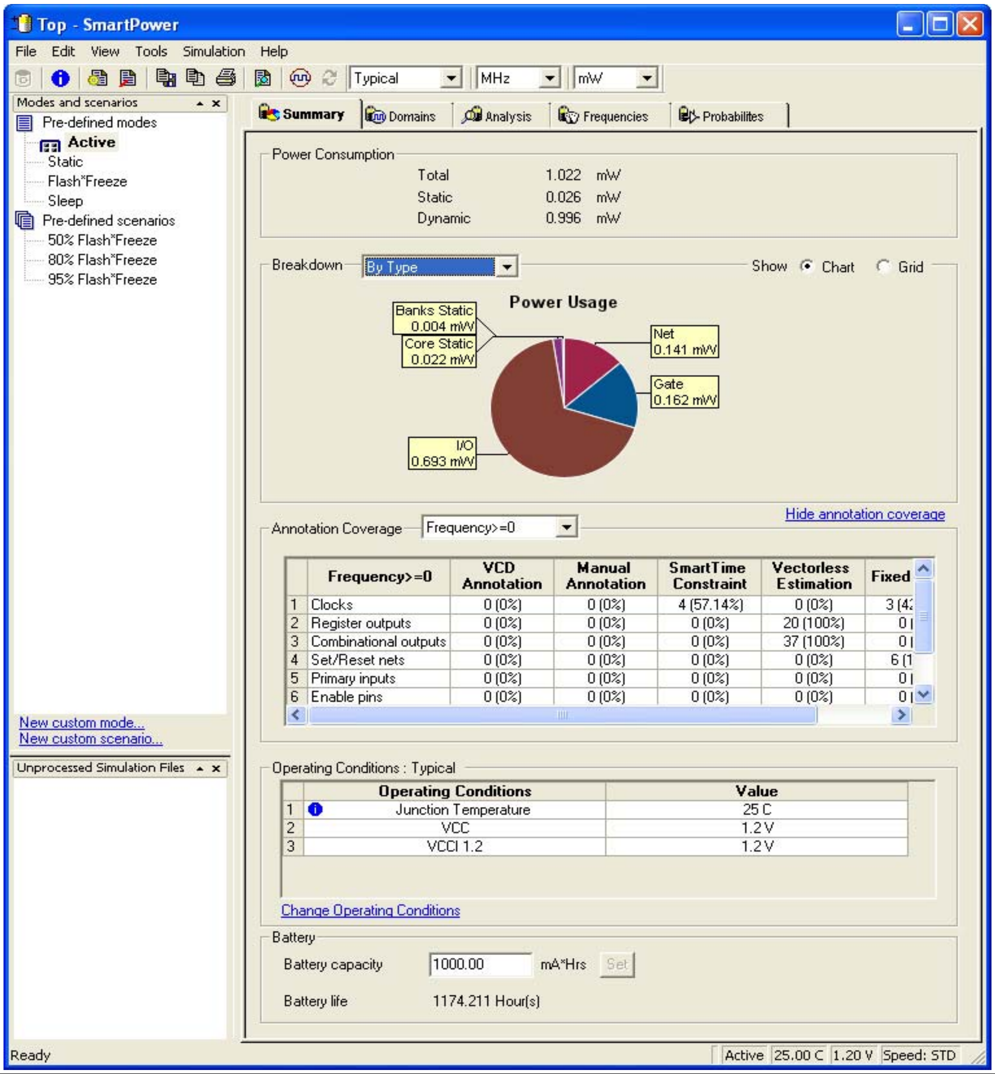

Figure 1-44 **• SmartPower Analysis GUI**

The constraints that we set in Smart Time are imported into Smart Power. SmartPower includes the following tabs:

- **Summary -** Displays the total power consumption, the temperature and voltage operating conditions, battery capacity in mA/hr and the projected battery life.
- **Domains -** Displays a list of existing domains with their corresponding clock and data frequencies. Use the Domains tab to set different clock frequencies and observe the effect on power consumption.
- **Analysis -** Displays detailed hierarchical reports of the power consumption.
- **Activity -** Used to attach switching activity attributes to the interconnects of the design.

Back-Annotation

Back-Annotation generates a SDF file that contains timing information for your design. It is used for post-layout, timing-accurate simulation.

To run Back-Annotation:

1. Click **Back-Annotate** to extract post-layout timing delay information from your design for simulation. The Back-Annotate dialog box appears (Figure 1-45).

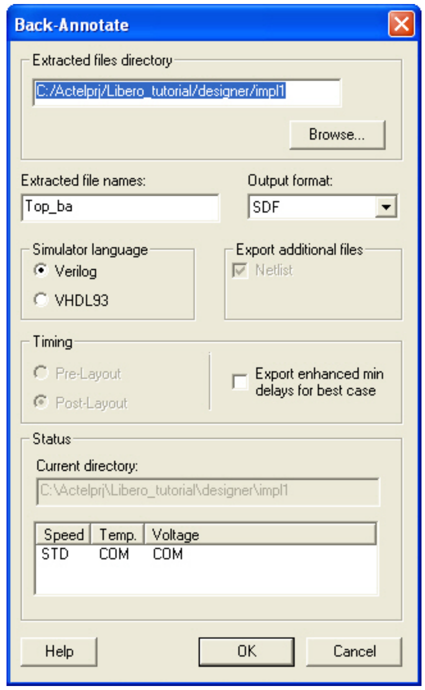

Figure 1-45 **• Back-Annotate Dialog Box**

2. Leave the default settings and click **OK** to continue. The Designer Log window shows the command succeeded. The Back-Annotate icon in Designer turns green.
3. In Designer, from the **File** menu, choose **Save** to save your changes to Top.adb. Close Designer to continue the tutorial (**File > Exit**).
4. In the Project Manager Hierarchy tab, right-click **Top** and choose **Run Post-Layout Simulation**. ModelSim opens.
5. In ModelSim, let the simulation run for a short period of time (wait 10-20 seconds). You can stop the simulation from the ModelSim menu (**Simulate > Break**). Use the zoom buttons to change the scale and view details in the waveforms.
6. Close ModelSim.

Generating a Programming File

1. Click **Designer** in the Project Manager to open it. Make sure that the Top.adb file is open.

2. Click **Programming File** to open the FlashPoint Programming File Generator (Figure 1-46).

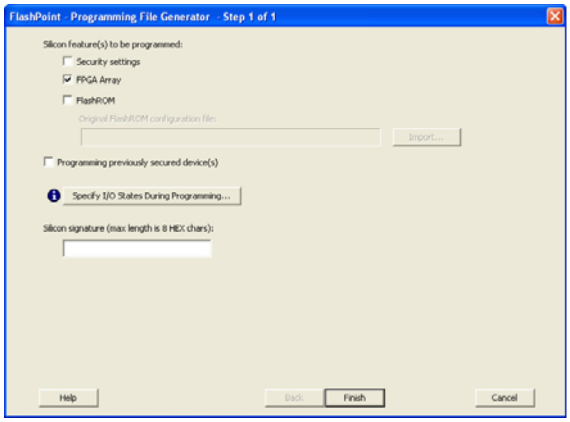

Figure 1-46 **• FlashPoint Programming File Generator Dialog Box**

3. Leave the default options in FlashPoint and click **Finish**.

   The Generate Programming Files dialog box appears (Figure 1-47).

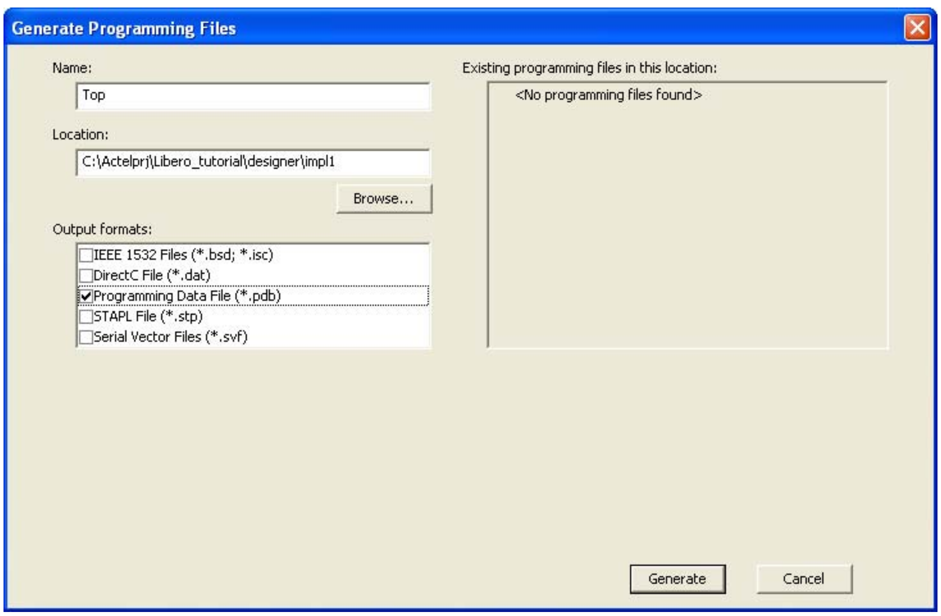

Figure 1-47 **• Generate Programming Files Dialog Box**

4. Specify the **PDB output format** and click **Generate**.

   That the Programming File button in Designer turns green and the log window shows that the programming command succeeded. Save the file and close ModelSim.

   IMPORTANT: Before programming the device, ensure that the I/O bank voltages are configured properly on the target board. Table 1-1 lists the default software settings for I/O voltages on the target devices. If you are using the default I/O voltage settings use this table to define the appropriate I/O voltage settings on your target board. If you modified the I/O voltage settings, set the I/O voltages on the target board to match the I/O voltage settings you have selected in Designer.

Table 1-1 **• Default Software Settings for I/O Supply Voltage**

Target Development Board	VCCI
IGLOO nano STARTER KIT	1.2V
IGLOO Icicle Board	1.2V
IGLOO nano Starter Kit	3.3V
IGLOO PLUS Starter Kit	1.2V
ProASIC3 Starter Kit	3.3V
Cortex-M1 enabled IGLOO or ProASIC3 Starter Kit (with 3 I/O connectors)	3.3V
Cortex-M1 enabled ProASIC3 Starter Kit (with 2 I/O connectors)	3.3 V

Step 9 - Programming the Device

These instructions are written specifically for the IGLOO nano Starter Kit. Jumper and LED numbers are different for other boards.

Initial Setup

Flash-based FPGAs are programmed by configuring the flash switches in the device, which determine the interconnect routing and configuration of logic to be used. The programming file created earlier in this tutorial contains all the information that the programmer needs to program the device. Since no external PROM is needed for flash FPGAs, you program and reprogram the device directly. Microsemi’s flash FPGAs are programmable by either a generic, standalone device programmer or by In-System Programming (ISP). Microsemi supports ISP using JTAG, which is supported by the FlashPro3, an on-board microprocessor, or other generic JTAG programmers.

Please visit Microsemi's website for more information about programming flash FPGAs (http://www.actel.com/products/hardware/default.aspx).

Please visit your specific board’s documentation for instructions on how to set up the hardware for programming. Although the programming hardware may differ, the programming software is the same for the FlashPro3 or FlashPro4.

You must set up the programmer before you can program your device. To do so:

1. Plug the USB cable into the board to power it up.

   The board is powered by the USB connection and no external supply is required; a 5V wall-jack connector is provided as an alternative if USB power is not available.
2. LED9 illuminates, indicating that power is supplied to the board.
3. Install the driver for the programmer once the system prompts and shows found new hardware.

Programming the Device

To program the device:

1. Make sure that you have made all the connections on the board. In the Project Flow window in the Project Manager click **Programming-FlashPro**. When you open FlashPro, the software automatically connects to the Low Cost Programming Stick (LCPS) (Figure 1-48).

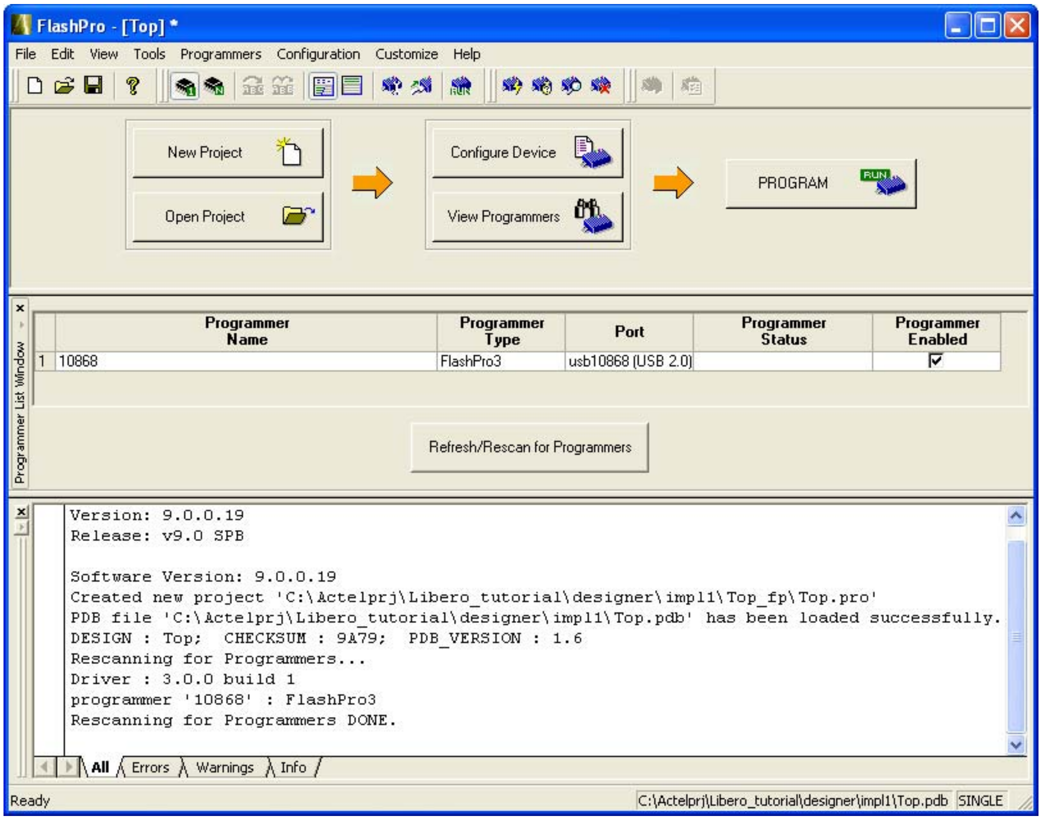

***Figure 1-48* • FlashPro Programmer**

The Programmer List window shows the programmer name, type, and port used. A successful connection, or any error, appears in the FlashPro log window. The Top.pdb file is automatically loaded when FlashPro3 is opened using the Libero IDE Project Manager.

2. Optional: If you want to load the file manually, click **Configure Device**, browse your project directory and load the Top.pdb file.
3. When the configuration is complete, click **Program** in the FlashPro screen to program the device. The FlashPro log and the Programmer List windows display current programming status (Figure 1-49)

Do not disturb the power cord when programming; it could damage the device.

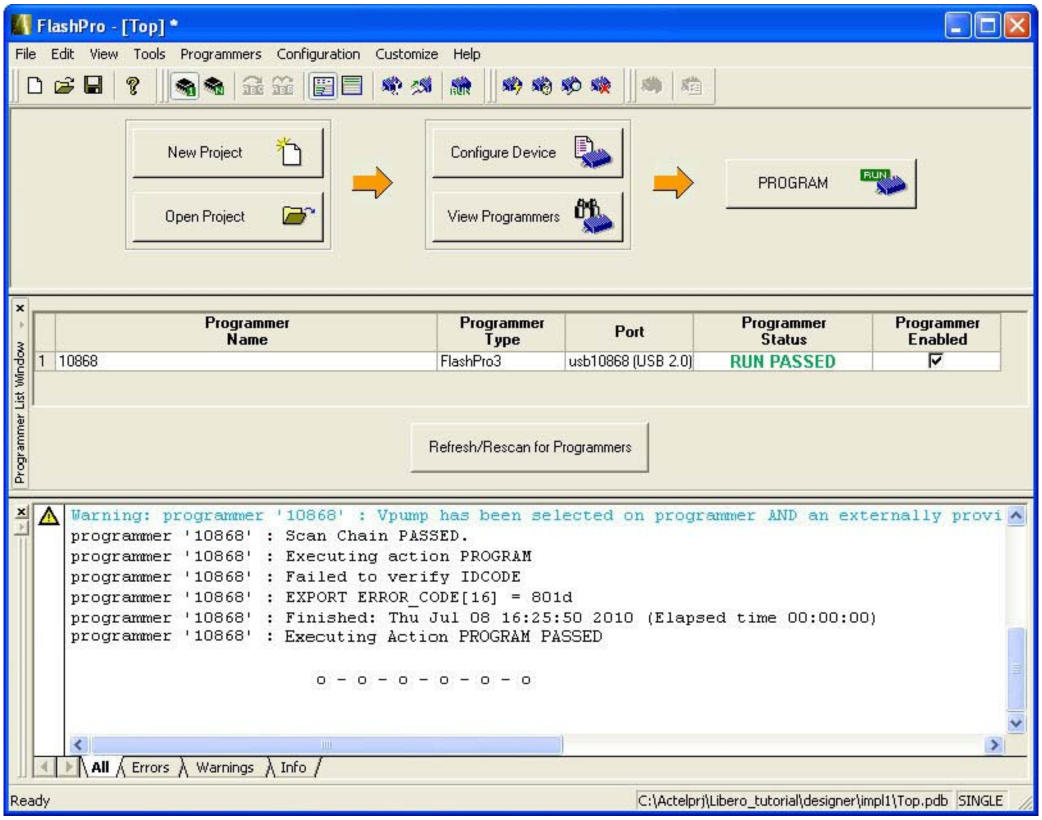

Figure 1-49 **• FlashPro Programming Status**

Running the Design

After the programming completes, the device will be active, and you should observe the LEDs counting. Perform a reset by pushing the reset button on the board to reset the counters at any time (SW1).

You can also observe the FlashFreeze mode for the device. When counter is running, use SW6 to assert FF externally. The counter would go off, as the FPGA would be in low power FlashFreeze mode, while the content of all the registers in the design are preserved. Turning on the switch starts the LED count from the state where it was when FlashFreeze was asserted.

Congratulations! You have completed the Libero IDE tutorial.

A – Product Support

The Microsemi SoC Products Group backs its products with various support services including a Customer Technical Support Center and Non-Technical Customer Service. This appendix contains information about contacting the SoC Products Group and using these support services.

Customer Technical Support Center

The SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions. The Customer Technical Support Center spends a great deal of time creating application notes and answers to FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

SoC Products Group Technical Support

Visit the SoC Products Group Customer Support website (www.actel.com/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC Products Group home page, at www.actel.com.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center from 7:00 a.m. to 6:00 p.m., Pacific Time, Monday through Friday.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

Phone

Our Technical Support Center answers all calls. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:00 a.m. to 6:00 p.m., Pacific Time, Monday through Friday. The Technical Support numbers are:

650.318.4460
800.262.1060

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found on the website at www.actel.com/company/contact/default.aspx.

Non-Technical Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

Actel's customer service representatives are available Monday through Friday, from 8 AM to 5 PM Pacific Time, to answer non-technical questions.

Phone: +1 650.318.2470

Microsemi Corporation (NASDAQ: MSCC) offers the industry's most comprehensive portfolio of semiconductor technology. Committed to solving the most critical system challenges, Microsemi's products include high-performance, high-reliability analog and RF devices, mixed signal integrated circuits, FPGAs and customizable SoCs, and complete subsystems. Microsemi serves leading system manufacturers around the world in the defense, security, aerospace, enterprise, commercial, and industrial markets. Learn more at **www.microsemi.com**.

Corporate Headquarters
Microsemi Corporation
2381 Morse Avenue
Irvine, CA
92614-6233
USA
Phone 949-221-7100
Fax 949-756-0308

SoC Products Group
2061 Stierlin Court
Mountain View, CA
94043-4655
USA
Phone 650.318.4200
Fax 650.318.4600
www.actel.com

SoC Products Group (Europe)
River Court, Meadows Business Park
Station Approach, Blackwatery
Camberley Surrey GU17 9AB
United Kingdom
Phone +44 (0) 1276 609 300
Fax +44 (0) 1276 607 540

SoC Products Group (Japan)
EXOS Ebisu Building 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668

SoC Products Group (Hong Kong)
Room 2107, China Resources Building
26 Harbour Road
Wanchai, Hong Kong
Phone +852 2185 6460
Fax +852 2185 6488

5-02-9123-16/01.11