CoreConsole

User's Guide

Actel®

Actel Corporation, Mountain View, CA 94043

Printed in the United States of America

Part Number: 5020056-2

Release: April 2006

Trademarks

Table of Contents

Introduction to CoreConsole

CoreConsole is a System-Level development tool and IP deployment platform that simplifies the task of assembling and connecting IP for implementation in Actel FPGAs. It enables you to select IP components from a database supplied by Actel and third party IP vendors, and graphically "stitch" them together to build a processor based System-Level integration (SLI) design. When the design is complete, the RTL and other files needed to implement it can be easily generated and written to disk, where they are available for import into the Actel Libero® Integrated Design Environment (IDE). CoreConsole also generates a testbench for the SLI design that you can build to assist in verification. The current release has a practical set of IP components for building CoreMP7 Advanced Microcontroller Bus Architecture (AMBA) based designs. Future releases will support a larger portfolio of IP components and a wide range of Generators for different software tools and operating systems, as well as unit testbenches.

1 CoreConsole in the Design Flow

You can use CoreConsole to generate a project for a top-level system design. Note that CoreConsole does not generate the top-level system nor map pins on the FPGA. Also, CoreConsole does not handle bi-directional buses; the tool outputs separate input and output buses for a bi-directional system bus along with an output enable signal. Your top-level is responsible for creating the bi-directional bus.

Once the design has been imported into Libero IDE, you must integrate it into your final system-level design (including mapping I/Os). The overview of the flow is shown in Figure 1-1.

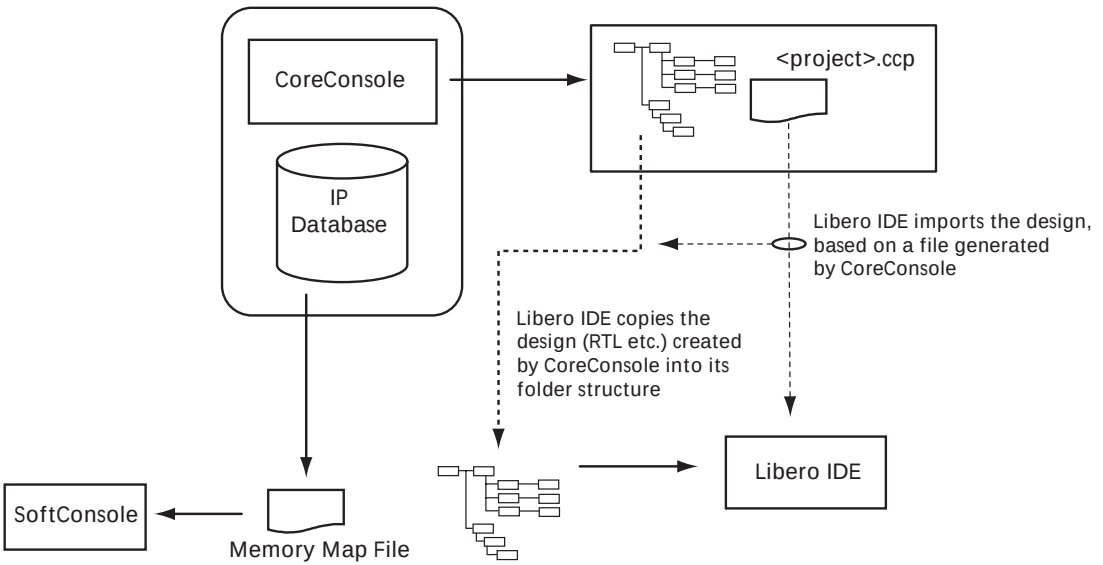

Figure 1-1. CoreConsole Flow

There are two CoreConsole design flow integration elements:

1. Integration with Libero IDE tool flow
2. Integration with Software Developers Console IDE (SoftConsole) tool flow

Libero IDE does not audit the CoreConsole output. If a new CoreConsole iteration is performed, the CoreConsole project must be reimported into Libero IDE.

Libero IDE Integration

Libero IDE integration is a seamless and automatic process. The files generated by CoreConsole and their locations on your PC's hard drive are described in a set of XML files (CCP and CXF) that contain the information needed by Libero IDE to locate them and import them into the correct simulation and design directories within the Libero IDE. Use the Libero IDE project import to build a CoreConsole generated design. You must take the project system, generated by CoreConsole, and instantiate it in your top-level design. In addition to the RTL for the design, Libero IDE generates a BFM (CoreMP7 Bus Functional Model) and script that enable you to test the CoreConsole project system. This is described in detail in the *CoreMP7 User's Guide*.

"Output Files" on page 30 describes the files generated by CoreConsole, and how they are organized to enable Libero IDE to import them.

SoftConsole Integration

There is automatic support for CoreConsole integration with the SoftConsole IDE. This includes export of a memory map file and any driver files that are available for peripheral cores used in a CoreConsole project design. Driver files are not available for all peripheral cores.

Integration with Other Tools

CoreConsole outputs RTL, and it has been verified that this RTL works with Model*Sim*®, Synplify®, Designer®, and the other tools incorporated into the Libero IDE. As a result, it is possible to use a point tool flow instead of the Libero IDE if desired. CoreConsole does not automatically generate any scripts to automate this process.

2 System-Level Design

CoreConsole enables you to build a system level design by automating the design process and connecting the sub-system and IP components that surround a processor core. CoreConsole is a bus-centric tool. The tool currently supports the AMBA bus interfaces to connect the IP within a design. Building the system around the bus fabric enables the extension of the system outside of CoreConsole.

You can export the AMBA Advanced High-Performance Bus (AHB) or Advanced Peripheral Bus (APB) interfaces to the CoreConsole Top Level and then connect them to AMBA compliant IP outside of CoreConsole (see Figure 2-1).

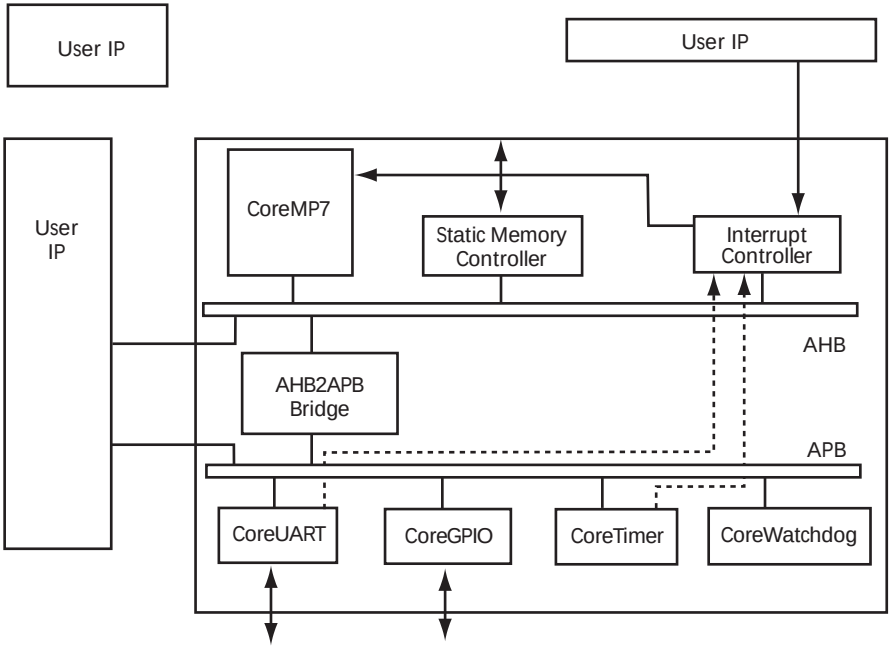

Figure 2-1. Example CoreMP7 System

A typical system consists of the project system generated by CoreConsole, user IP Components, and the top-level instantiation of the project system that connects it to any top-level IP and wires it to the I/O pads.

Bus-Centric Design

The current release of CoreConsole uses AMBA buses. Future versions of CoreConsole may support other bus protocols in addition to AMBA. The principal requirement of the bus-centric approach is that all of the IP components must have the correct bus interface to be integrated by CoreConsole into a design.

AMBA

The current release supports AMBA AHB and APB. Most of the components in the database have AHB or APB interfaces. The CoreMP7 processor has an AHB Master Interface, so it must be one of the three masters on the AHB bus.

3

CoreConsole User Interface

CoreConsole has a Components tab and Generate tab (Figure 3-1).

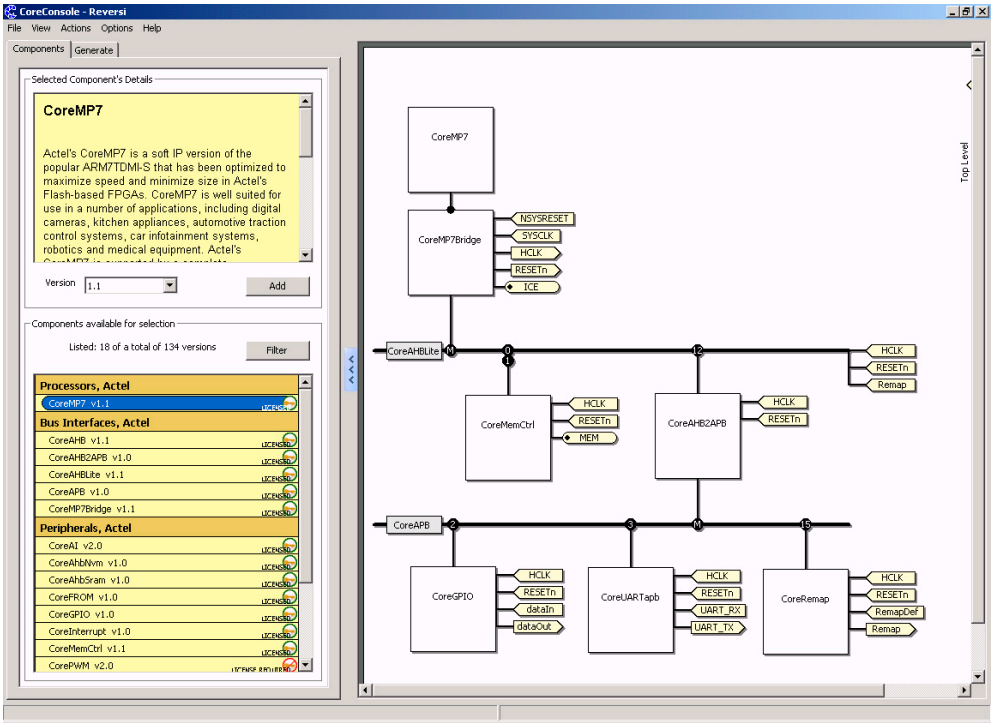

Figure 3-1. CoreConsole User Interface

The Components tab lists all of the IP components you can incorporate into your design. It provides you with a graphical representation of your design, and enables you to add or delete connections and configure cores.

The Generate tab is where you create the CoreConsole output that can you can import into the Libero IDE.

Components

The Components tab lists all the IP components available in the CoreConsole database (Figure 3-2). You can use this tab to configure the component display filter and select a component and add it to a design.

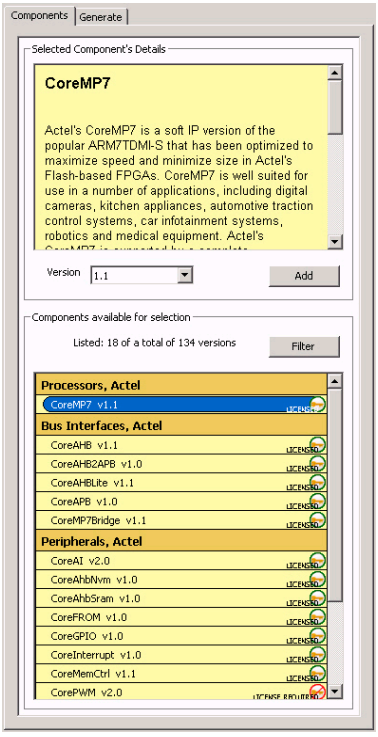

Figure 3-2. Components Tab

The Components tab lists Selected Component Details and core status. Selected Component Details displays a description of the selected component, the components functional category, the component provider, and the version number.

Components can be added in any order. Bus fabric components are included in the list of available components, and can be added to the design just like any other components, although they display as buses rather than as block components.

The status (whether it is licensed or not) for each component is adjacent to the component name. Use the System Options dialog box (from the **Options** menu, choose **System Options**) to display your core as an icon, text, or both.

If there are multiple versions of a component available in the vault, the latest (or highest) version will be the default selection and will be shown in the box. If a previous version of a component is available and you want to use it, you can select it with this menu.

The Components tab shows all of the connections in a design. You can activate **Show Linked Connections** to highlight where each connection goes. The only fixed element in the Schematic window is the **Top Level** bar (see Figure 3-3), which is all of the connections to the design outside of the CoreConsole project system.

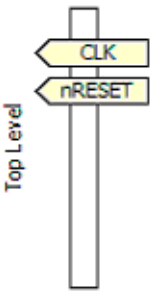

Figure 3-3. Top Level Bar

Component Filter

CoreConsole can be used with an unlimited number of IP components. To make it easier to find the specific component that you are looking for, CoreConsole has a component filter and search capability. Click the **Filter** button on the Component tab to open the Filter dialog. On the dialog, you can choose to list components with or without the version information and you can sort them by name or function. There is a search capability on the dialog if you are looking for a specific component; it enables you to search based on the component name, manufacturer, market segment, function, interface, and availability on your computer.

Generate

The Generate tab is used to set final system configuration options (HDL language and license options) and to generate the design for export to Libero IDE. This is described in detail in "Generating a Design" on page 29.

Connections

Connections are central to the operation of CoreConsole. The various types of connections define the nets that bind the components together. The main connections are the bus interfaces.

Bus Connections

The buses are represented in CoreConsole as shown in Figure 3-4. The Bus Interface connections are represented as solid vertical lines between a component and the bus.

Figure 3-4. Bus Representation

Each component connection to a bus displays either an M for a bus master, or a number (0-15) for a bus slave. This is shown in Figure 3-4 and in the standard AMBA bus dialogs box connections (Figure 3-5).

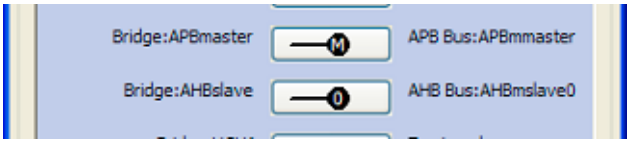

Figure 3-5. Bus Representation in Dialog Boxes

Non-Fabric Bus Connections

Bus connections can be made that do not attach directly to the AMBA fabric. This occurs when there are other groups of wires that are described as buses (e.g., the SRAM interface to the memory controller), or when the AMBA interfaces are exported to the external system outside the CoreConsole project. In these cases the buses are represented by the symbol shown in Figure 3-6.

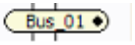

Figure 3-6. Bus Connection Icon

These bus interfaces can include connection of both input and output types. There are no bidirectional signals in CoreConsole, but buses can encapsulate both input and output signals.

Ad-Hoc Connections

Apart from the AMBA fabric connections, most of the connections shown on the diagram and in the connection dialog are **Ad Hoc** connections. These are equivalent to wires connecting pins on a component. They are unidirectional, point to point connections. They are represented by the symbol shown in Figure 3-7. The direction (i.e., whether it is Input or Output) is indicated by the orientation of the arrow on the signal.

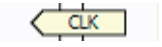

Figure 3-7. Ad Hoc Connection Symbol

4 Understanding Projects

CoreConsole organizes designs into projects. A project is a self-contained collection of all the files and information needed to generate an export set for Libero IDE and various other files including the Memory Map report, and any SoftConsole support files. When you start CoreConsole, you must choose to either create a New project or Open an existing one.

Existing Projects

To open an existing project, select **File** > **Open** and pick from the list of projects.

Saving Projects

If you open an existing project, you can save it by selecting **File** > **Save**, or you can create a new project from the existing project by saving it under a different name by selecting **File** > **Save As**. Designs should be saved when you make any changes.

Note: **There is no undo option**.

The Save and Generate button on the Generate tab also saves the project before generating the RTL.

5

Working with Diagrams

System Options

The schematic window is controlled by the options section of the System Options menu (see Figure 5-1). To open this menu, from the **Options** menu, choose **System Options**. Many of the options are self-explanatory, but they are also described here for clarity.

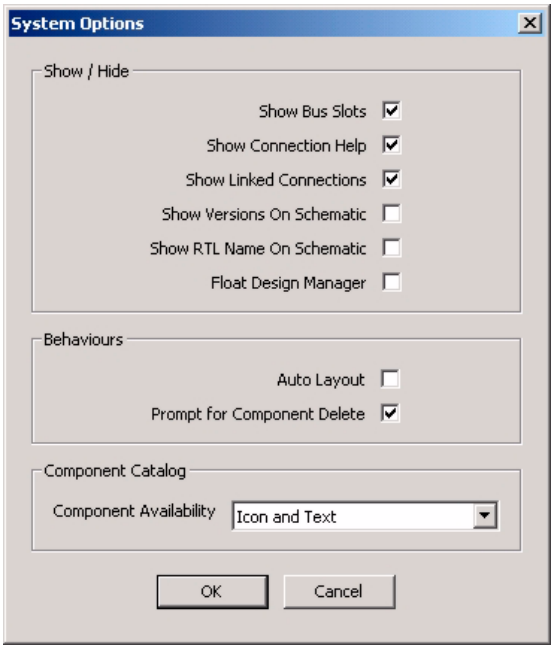

Figure 5-1. System Options

Show Bus Slots

Choose this option to display the slot number on the bus to which each component is attached. The slot number indicates the component's location in the project system memory map. If it is not selected, the slot number is replaced by a connection point only.

Show Connection Help

This option enables the help window for each component. See "Component Help" on page 22 for more information.

Show Linked Connections

Select this option to float over a connection in the schematic and highlight all the other points connected to the net.

Show Versions on Schematic

Select this option to display the version number of each block under the component name of the block.

Show RTL Name on the Schematic

Select this option to display the RTL of each block in your design under the component name.

Float Design Manager

Select this option to unlock the Design Manager tabs and move them around the screen; this enables you to create more screen area for the schematic window.

Auto Layout

Select this option to automatically place the components drawn in the schematic window.

Prompt for Component Delete

Select this option to enable or disable the confirm delete dialog box.

Component Availability

Set this option to display a components license status as text, a graphic, or both.

Schematic Window

A design project is represented graphically as a conventional bus-centric processor system in the Component window. The Top-Level box displays the connections to the rest of the design outside of the CoreConsole.

Components are added from the Components Tab, but once they have been added to the diagram, all other design activities can be carried out directly in the schematic window. Components are represented as boxes; float your mouse over a component and select an item from the tool bar, or right click and select an item from the shortcut menu to modify a component. Double-click a component to open the Configuration dialog box. You can move all components (including buses) around the screen.

6

Creating a CoreConsole Design

Adding Components

To add a component, select it in the Component tab and click **Add**, or double-click the component's name in the tab. The new component appears in the Schematic window.

You cannot drag-and-drop components into the Schematic window.

If you are using auto stitching, add the processor, MP7Bridge, and bus (in that order), so that auto stitching does most of the connection work.

Connecting Components

To connect components, open the Connection dialog box (right-click the block and select **Connection** from the shortcut menu) and select the Component and Pin for both the source and destination. If you use auto stitching most of the bus connections are made automatically.

Adding CoreMP7

When adding the CoreMP7 to the system, you must set your Debug/No Debug and Die options.

Debug or No Debug. Selecting the debug option (Figure 6-1 on page 19) includes the debug block in the core; this increases the overall size of the core.

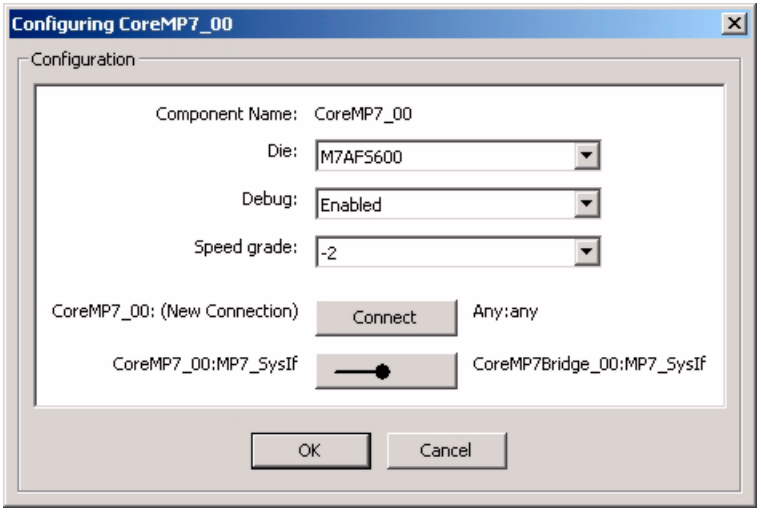

Figure 6-1. Debug Option

- Die – You must specify a die (target device) for your core. Cores are unique to each device; you must create a new core if you select a new device.

Connection Dialog Box

To open the Connection Dialog box:

- Click the **Connect** button in the tool bar (Figure 6-2).
- Click the Connect button in the Component Configuration dialog box.
- From the **View** menu, choose **Connections**.

Figure 6-2. Connect Button

Making Connections

When you have opened the connections dialog either directly or via the configuration window, you are ready to connect the component. Figure 6-3 shows a typical connection dialog.

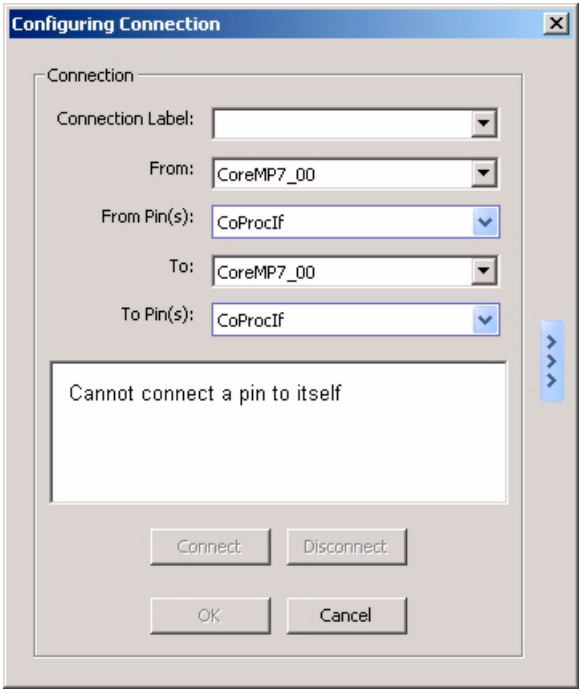

Figure 6-3. Configuring Connection

Enter the connection information in this dialog to make a connection.

To make a connection:

1. Select the source component in the **From** list.
2. Select the source pin or bus in the **From Pin(s)** list (this can be either an input or an output from this component).
3. Select the destination component in the **To** list.
4. Select the destination pin in the **To Pin(s)** list (this can be either an input or an output from this component). This list only shows compatible pins. If the destination you select is **Top Level,** then the **To Pin(s)** remains blank, and you add or select a label for the connection.

5. (Optional) If you are connecting to a connection that has not been labelled previously, you must enter a name in the **Connection Label.**

   If you are connecting to a point that has been labelled previously (e.g., SYSCLK or NSYSRESET at the Top Level), then select the name from the list in **Connection Label**.

   - If the **From** and **To** Component and Pins are compatible with the label selected, the **Connect** button will be activated.
   - If either the **From** or **To** Component and Pins are not compatible with the label selected (i.e., if the label is already in use for a net to which it cannot connect), a warning will be given and the **Connect** button remains greyed out.
   - If you are making a Bus connection, the label will be generated automatically.

6. After all selections are made and the connection is labelled, press **Connect** to make the connection. The **Connect** and **Ok** button remain greyed out until all of the selections and labels are correct.

   Note: Only the **Connect** button makes the connection. The **OK** button exits the window and does not make a connection.

Different connections in CoreConsole are explained in "Connecting to the Bus" and "Connecting to the Top Level".

Connecting to the Bus

When you connect a component to the AHB or ABP bus, CoreConsole automatically assigns a name to the connection. The point at which you connect on the bus also defines your memory map.

Connecting to the Top Level

Connecting to the Top Level is how you get signals into and out of the CoreConsole project system. These connections are made in the same way as other connections except that the **Top Level** does not have pins and you can either select an existing connection from the **Connection Label** drop down list, or if it does not exist, create a new one.

Component Help

Each component has online help associated with it, which can be useful when you need to find out what any or all of its connections are. This can be enabled or disabled in the **Options > System**

Options menu by selecting **Show Connection Help.** The help window itself can be expanded from the side of the component connection dialog window. See Figure 6-4.

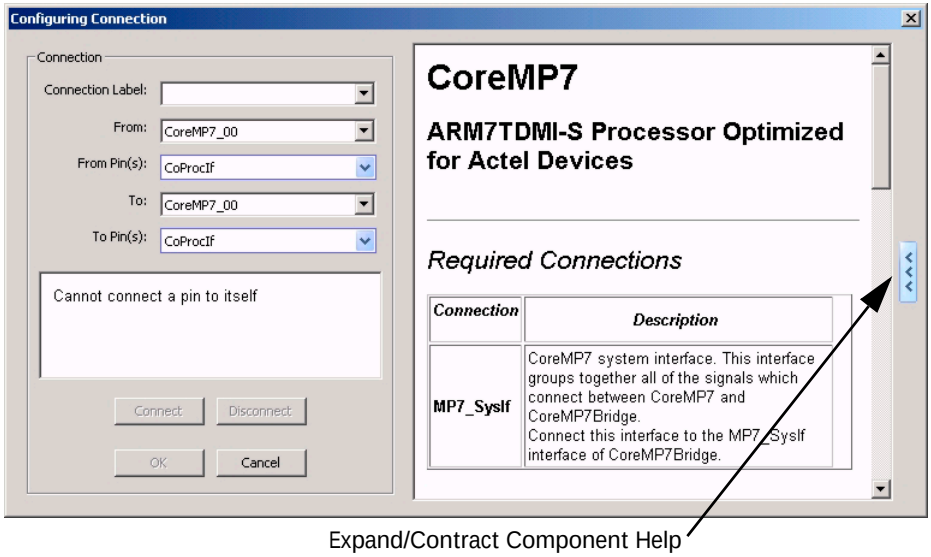

Figure 6-4. Component Help

Auto Stitching

CoreConsole has the ability to automatically connect together standard interfaces for components. This is referred to as auto stitching. Many of the interfaces to standard components are well defined and uniform and permit unambiguous automatic connection. These include Clock and Reset lines, AHB and APB connections, CoreMP7 debug interface and watchdog connections, and others.

CoreConsole makes its best "guess" at which connections need to be made and then presents the list of proposed connections to you for confirmation before making the connections.

Auto stitching can be applied repeatedly; for example, when you add new components.

To auto stitch:

1. Add components to your design.
2. From the **Actions** menu, choose **Auto Stitch.**
3. The Auto Stitching window appears with CoreConsole's proposed connections (Figure 6-5 on page 24).

4. If there are any connections you do not wish CoreConsole to make, **deselect** them in this window.
5. Click **Stitch** and watch as the connections are made.

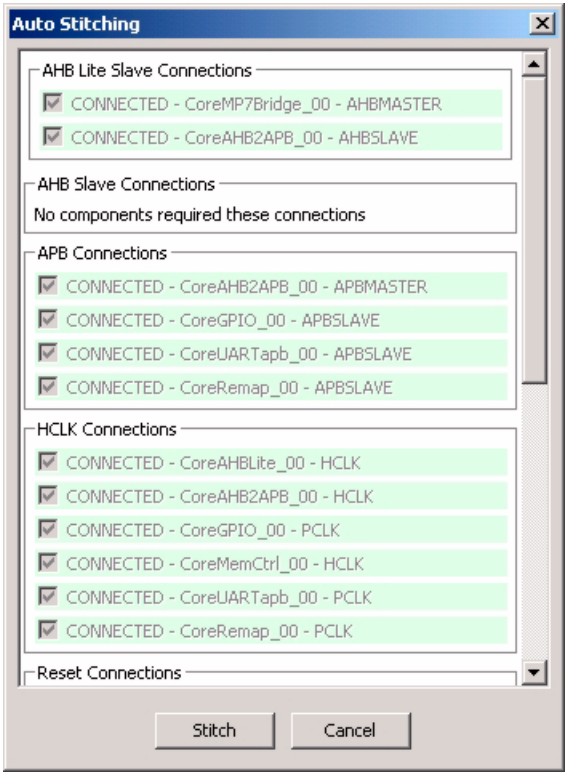

Figure 6-5. Auto Stitching

Auto stitching can be applied repeatedly and at any time. It is possible to make some of the connections you require manually and then select **Actions > Auto Stitch** if you wish. Actel recommends that you add all the components you need, then auto stitch, then add the additional connections manually (e.g., exporting Rx or Tx from the CoreUART).

Auto stitching is capable of connecting complex components correctly, in particular the Watchdog component, which requires different Reset connections.

Deleting Connections and Components

There are three ways to delete a component or connection.

- Delete using the toolbar – Float your mouse over a component and click the **Recycle Bin** icon in the Component Toolbar.
- Delete Key – Float your mouse over a component and press the **Delete** key.
- Shortcut Menu – Right-click a highlighted component to display the shortcut menu and choose **Delete**.

All of the ways are equivalent and are provided to support your personal preferences. When you delete a component, all of its connections are also deleted. When you delete a connection, only that specific connection will be removed even if the same source (e.g., CLK) is used elsewhere in the design.

It is important to note that actions done to the design, including **delete**, are not applied until the project has been saved or generated.

If you float your mouse over a component and press the delete key, the component will be deleted. If you have selected the “Prompt for Component Delete" option in the System Options menu, a Confirm Delete box will open that you have to click **Yes** to delete the component.

Configuring Components

AMBA Bus Configuration

The bus fabric generated by CoreConsole is either AMBA AHB or APB. The bus fabric consists of an AHB controller, an AHB2APB bridge, and an APB controller. The bridge can be connected to any slot on the AHB bus, setting the addresses for the peripherals connected to it.

Specifying Memory Map

The CoreConsole project system memory map is defined by the connections you make to the AHB and APB bus. The specific details of the bus fabric and components can be found in the specific datasheet for the bus or components.

The bus structure is hierarchical with the APB bus—always a slave of the AHB bus. The AHB address space is divided into 16 equal sized slots. Connecting a peripheral component to one of these slots defines its position in the project system memory map. The bridge can be connected to any slot on the AHB, setting the addresses for all the peripherals connected to it (see Figure 6-6 on page 26).

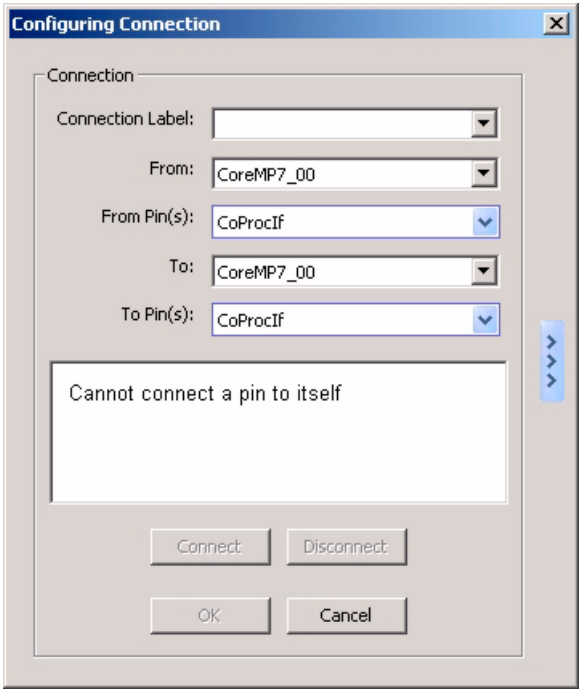

Figure 6-6. Systems Connection Dialog Box

Memory Map Requirements and Recommendations

- CoreMP7 must be added as an AHB Master on the AHB bus.
- The Memory Controller must be connected to Slot0 on the AHB bus, because on reset the processor vectors to this address.
- The Interrupt Controller if present is most often connected to Slot 15 on the AHB bus so that it sits at the top of memory.

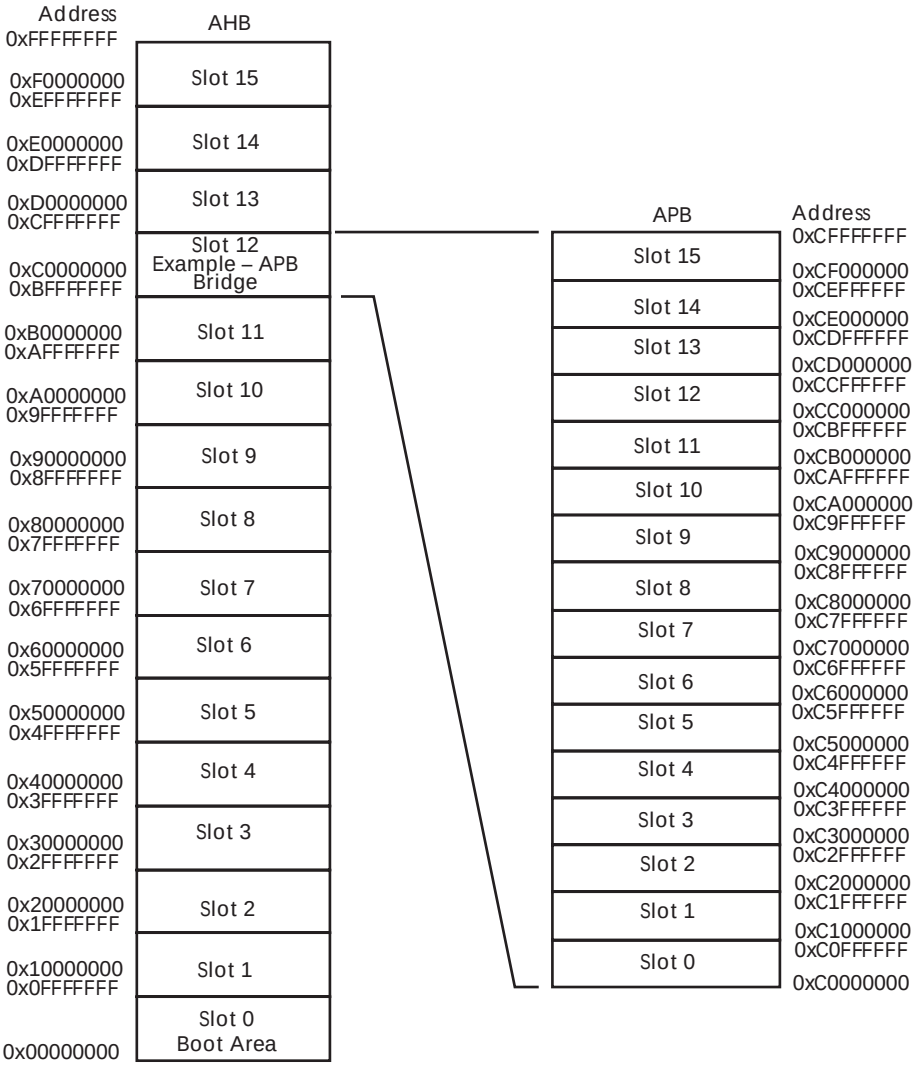

Figure 6-7. Project System Memory Map

7 Generating a Design

When you have finished connecting and configuring a design, select the **Generate** tab, select the elements you want to generate (all are selected by default), and click the **Generate** button. Figure 7-1 shows the layout of the **Generate** tab.

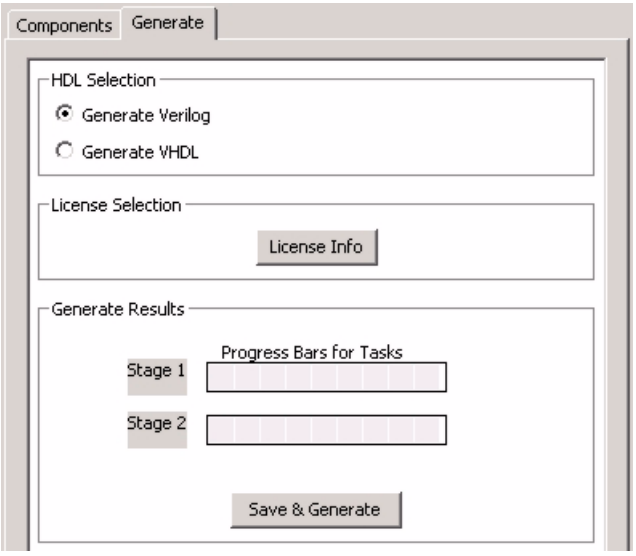

Figure 7-1. Generate Tab

HDL Selection

CoreConsole supports both Verilog and VHDL RTL generation. Selecting one or the other will cause your system project design to be generated and saved to disk on your computer in that format.

License Selection

Clicking on the **License Info** button opens a dialog box that lists all of the components and buses in your design and their license status. RTL output is generated only for components and buses for which you have an appropriate license.

Generate Results

Click the **Save & Generate** button to generate all items in your project design and save it to your specified directory.

Output Files

CoreConsole outputs the generated files into

CoreConsole\LiberoExport\<my_coreconsole_project>

Where *CoreConsole* is the root folder of a CoreConsole installation, and

< *my_coreconsole_project* >

is the name of the project being generated. These files can be examined by navigating to the folder on your computer.

Inside the *LiberoExport* folder, you will find the full list of projects that they have created. Within each project you will find the list of Components; they follow the template:

```
<my_coreconsole_project>/
        <my_coreconsole_project>.ccp
        <comp1>/
        <comp1>.cfx
        <comp2>/
<comp2>.cfx
```

The <*my_coreconsole_project*>*.ccp* is an XML file that fully describes the layout and content of the design, and is used by Libero IDE to import the CoreConsole project. Within the file set for each component there is a <*comp_n*>*.cfx* XML file, and describes the sub-component of the design.

You have to import *my_coreconsole_project/my_coreconsole_project.ccp* into Libero IDE, and the Libero IDE then copies the *my_coreconsole_project* folder.

Within the output file list there are a variety of different file types associated with the design including the CCP and CXF XML description files. Other file extensions used by CoreConsole and accepted by Libero IDE include the following:

- Verilog files (sources, stimulus, BFM): *.v
- VHDL files (sources, stimulus, packages): *.vhd
- Verilog headers: *.h
- PDC file: *.pdc
- DB file: *.cdb
- Simulation files: *.vec
- PALACE library: *.lib

8 SLI Testbench

CoreConsole generates a project system testbench that enables you to test the integrity and connectivity of a design. The testbench is supported by the CoreMP7 Bus Functional Model (BFM) that is output as part of the generated design and imported into Libero IDE.

The testbench creation is fully automatic, but it is possible to extend the level of testing by editing the *subsystem.bfm* test script from within Libero IDE.

The testbench generation logic in CoreConsole assumes that you have named the clock and reset signals coming into the system as **SYSCLK** and **NSYSRESET**, and these are the names it expects to see on the CoreConsole project system port list. The testbench simulation does not run properly if you don't use these names.

Limitations of the Testbench

The testbench only tests within the boundary of the CoreConsole project system design. It does not include memory or other external models.

The testbench tests the connectivity of the project system based on the SPIRIT component definitions. This is done by reading and writing to component registers that are exposed to the AMBA bus. It does not do any functional testing of the components; this has to be done before they are submitted to the CoreConsole database. It is possible for you to extend the testbench to achieve a higher degree of functional testing.

Extending the Generated Testbench and Scripts

When the testbench and script file (*subsystem.bfm*) have been imported into Libero IDE, it is possible to extend them and perform a higher degree of functional testing on a system. The most straightforward method is to extend the *subsystem.bfm* script. The **CoreMP7 Datasheet** gives details of the basic scripting syntax, and it is not difficult to add more or deeper tests to the ones automatically created.

Licensing

CoreConsole uses FLEXlm licensing to enable delivery of IP components. You must acquire the appropriate licenses from Actel. There are two classes of IP component delivery:

- Obfuscated: The component is delivered as obfuscated Verilog or VHDL, fully synthesizable RTL. This is not intended for user modification.
- RTL: The component is delivered as "clear" VHDL or Verilog.

The class of license you have determines the type of component delivered.

If you do not have an IP license but have a CoreConsole license, you can build your design, but you will not be able to Generate it. If you subsequently acquire a license, you can re-open your design and then generate it.

To display licensing information for the components in your design, from the **Generate** menu, choose **License Info**. (Figure 8-1).

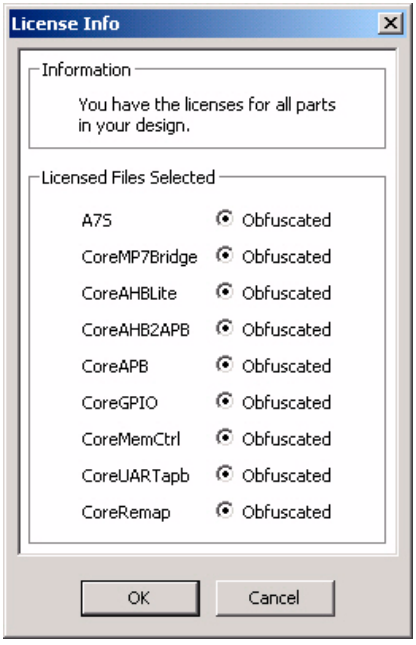

Figure 8-1. License Info

If you have licenses for both Obfuscated and RTL delivery, you can select which type of output you require on a per component basis by selecting the appropriate box beside each component.

9

Other Topics

Adding User IP

The current release of CoreConsole does not support the addition of user IP to CoreConsole. In future Releases, it will be possible to integrate user IP components into CoreConsole, and use them in exactly the same way as the supplied IP. In addition, there will be a tool provided to assist in the correct packaging to automate the process. The two key tasks necessary in adding user IP is ensuring that it has a correct bus interface and a full SPIRIT definition.

SPIRIT

SPIRIT (Structure for Packaging, Integrating, and Re-using IP within Tool-flows) is an industry standard with the goal of supporting automated IP re-use. CoreConsole uses a component's SPIRIT description to automatically integrate it into the design and to generate a project system testbench to verify the integrity of the design.

A

Product Support

Actel backs its products with various support services including Customer Service, a Customer Technical Support Center, a web site, an FTP site, electronic mail, and worldwide sales offices. This appendix contains information about contacting Actel and using these support services.

Customer Service

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

From Northeast and North Central U.S.A., call **650.318.4480**
From Southeast and Southwest U.S.A., call **650. 318.4480**
From South Central U.S.A., call **650.318.4434**
From Northwest U.S.A., call **650.318.4434**
From Canada, call **650.318.4480**
From Europe, call **650.318.4252** or **+44 (0) 1276 401 500**
From Japan, call **650.318.4743**
From the rest of the world, call **650.318.4743**
Fax, from anywhere in the world **650.318.8044**

Actel Customer Technical Support Center

Actel 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.

Actel Technical Support

Visit the Actel Customer Support website (www.actel.com/custsup/search.html) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the Actel web site.

Website

You can browse a variety of technical and non-technical information on Actel's 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. Several ways of contacting the Center follow:

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 tech@actel.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 (tech@actel.com) or contact a local sales office. Sales office listings can be found at www.actel.com/contact/offices/index.html.

Index

For more information about Actel's products, visit our website at http://www.actel.com

Actel Corporation *• 2061 Stierlin Court • Mountain View, CA 94043 USA*
Customer Service: 650.318.1010 • Customer Applications Center: 800.262.1060

Actel Europe Ltd*. • Dunlop House, Riverside Way • Camberley, Surrey GU15 3YL • United Kingdom*
Phone +44 (0) 1276 401 450 • Fax +44 (0) 1276 401 490

Actel Japan *• EXOS Ebisu Bldg. 4F • 1-24-14 Ebisu Shibuya-ku • Tokyo 150 • Japan*
Phone +81.03.3445.7671 • Fax +81.03.3445.7668 • www.jp.actel.com

Actel Hong Kong *• Suite 2114, Two Pacific Place • 88 Queensway, Admiralty Hong Kong*
Phone +852 2185 6460 • Fax +852 2185 6488 • www.actel.com.cn

5020056-2/4-06