UG0779
User Guide
SmartDesign

Libero SoC v12.0

Microsemi

a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer’s responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided “as is, where is” and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

Contents

Tables

Figures

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 3.0

Revision 3.0 includes the following changes:

- Template updates.
- Added links to Tcl Commands Reference Guides (section 2.5).

1.2 Revision 2.0

Revision 2.0 includes the following changes:

- Reorganized Chapter 4 for improved usability.
- Added section 5.2.3 Importance of .cxf files in SmartDesign.
- Minor updates throughout document.

1.3 Revision 1.1

Minor updates for clarity.

1.4 Revision 1.0

Revision 1.0 is the initial publication of this document.

2 Introduction

SmartDesign is a visual block-based design creation/entry tool for the instantiation, configuration, and connection of Microsemi IPs, user-generated IPs, and custom/glue-logic HDL modules. This tool provides a canvas, which is analogous to a breadboard, for stitching together the different components in the design. The final result from SmartDesign is a design-rule-checked and automatically abstracted synthesis-ready HDL file. A generated SmartDesign can be the entire FPGA design or a component subsystem to be reused in a larger design.

The following design objects can be instantiated in the SmartDesign Canvas:

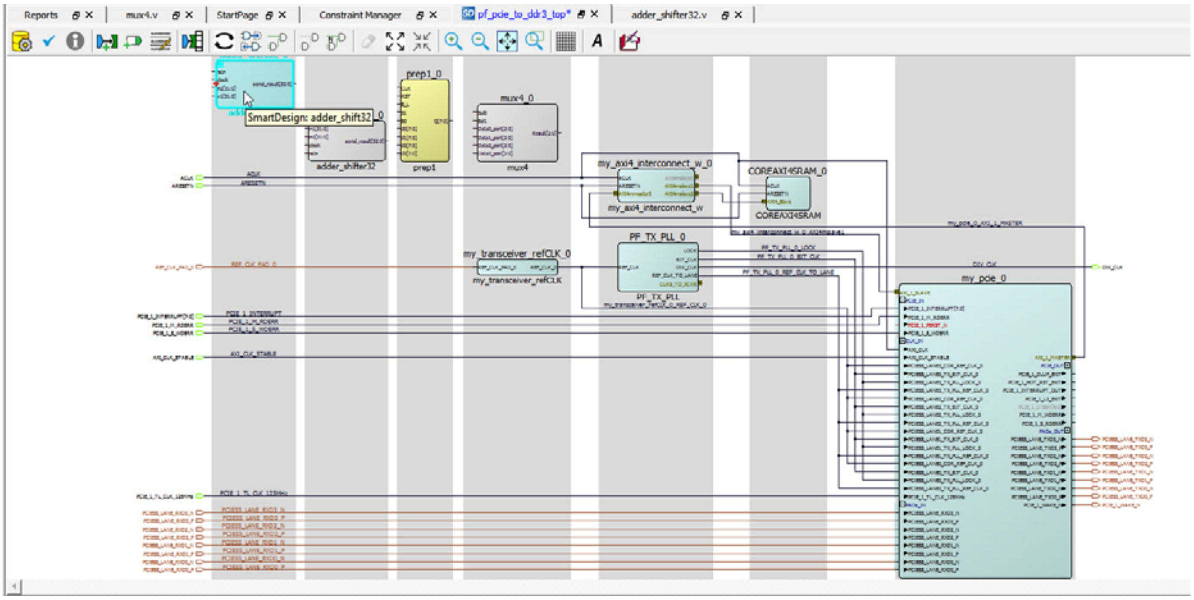
- Microsemi IP Cores
- User-generated or third-party IP Cores
- HDL design files
- HDL + design files
- Basic macros
- Other SmartDesign components (*.cxf files) generated from SmartDesign in the current Libero SoC project or imported from other Libero SoC projects
- Reusable design blocks (*.cxz files) published from Libero SoC

2.1 Canvas Layout

The SmartDesign canvas places all components in columns, with the nets vertically routed in the space between columns. Top-level input ports are placed in the leftmost column. Top-level output ports and inout ports are placed in the rightmost column.

Components can be moved up and down the columns. When components are instantiated in SmartDesign, they are placed in an existing column or a new column created for them. When components are moved up and down, the column boundaries are shown. When components are moved horizontally, the instance can be moved to a different column, the column it is in can be moved, or a new column can be created at the new location for the instance. Figure 1 shows the SmartDesign canvas.

Figure 1 • **SmartDesign Canvas**



2.2 SmartDesign in the Libero SoC Design Flow

SmartDesign enables you to stitch together design blocks of different types (HDL, IPs, re-usable design blocks, lower-level SmartDesign blocks, etc.) and generate a top-level design. The Files tab lists your SmartDesign files in alphabetical order.

You can build your design using SmartDesign with the following steps:

- **Step One – Instantiating components:** This step is analogous to inserting design components onto the breadboard. In this step you add one or more building blocks, HDL modules, components, and schematic modules from the Project Manager to your design. The components can be design blocks, IP cores from the Catalog, basic macros, design blocks (*.cxz) and other SmartDesign components (.cxf) file imported into the Libero SoC project.
- **Step Two – Connecting bus interfaces:** In this step, you can add connectivity via standard bus interfaces to your design. This step is optional and can be skipped if you prefer manual connections. Components generated from the Catalog may include pre-defined interfaces that allow for automatic connectivity and design rule checking when used in a design.
- **Step Three – Connecting instances:** The Canvas enables you to create manual connections between ports of the instances in your design. Unused ports can be tied off to GND or VCC (disabled); input buses can be tied to a constant, and you can leave an output open by marking it as unused.
- **Step Four – Generating the SmartDesign component:** In this step, you generate a top-level (Top) component and its corresponding HDL file. This component can be used by downstream processes, such as synthesis and simulation, or you can add your SmartDesign HDL into another SmartDesign. When you generate your SmartDesign, the tool invokes the Design Rules Checker to verify the connectivity of your design. Undriven/floating ports and other DRC violations are reported in the Log/Message window. Error Messages, if any, must be corrected before a component can be successfully generated. The design flow cannot proceed if component generation fails.

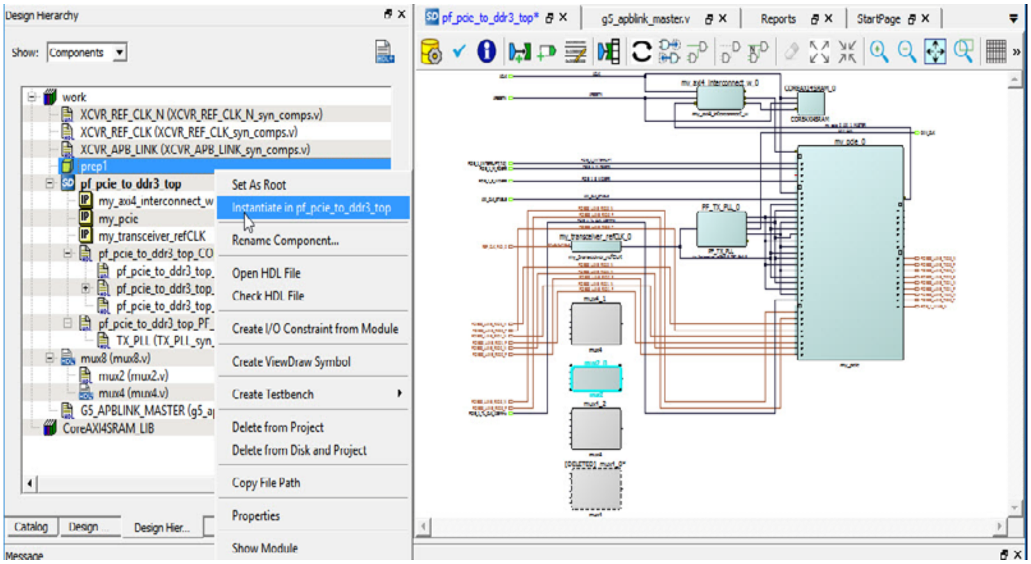
2.3 Instantiating Components into SmartDesign Canvas

For all the following design objects, drag-and-drop or **right-click object > Instantiate** to instantiate the design objects:

- IP cores—To instantiate IP cores, drag the IP core from the Catalog into the SmartDesign Canvas. You may have to configure the IP cores in the IP Core Configurator first before it can be dropped in the SmartDesign Canvas. IP cores displayed in the Catalog as italics are cores that are available in the Microsemi IP Core Repositories but are not yet downloaded to the vault (the disk location where downloaded cores are stored). Download the IP core first before configuration and instantiation.

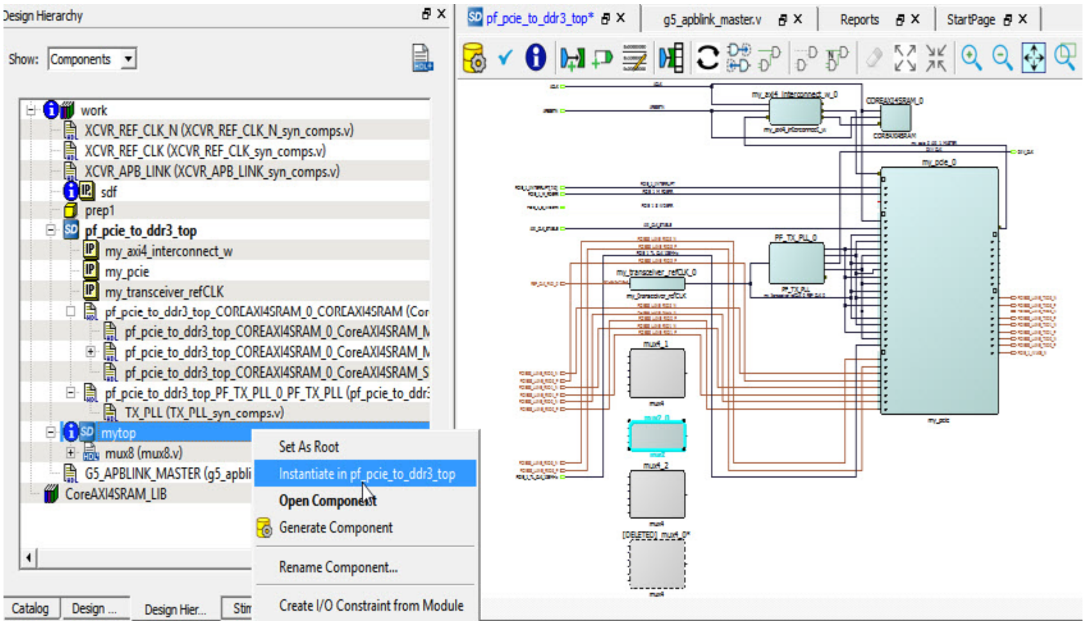
- HDL files—To instantiate HDL design blocks, drag the HDL design file from the Design Hierarchy into the SmartDesign Canvas.
- HDL+ cores—HDL+ cores are HDL files where the parameters and generics are used and a core has been generated out of it (right-click HDL file > Create Core from HDL). Drag and drop the HDL+ module into the SmartDesign canvas. The HDL core becomes configurable inside the SmartDesign canvas. Open the configurator to set the values for the parameters and generics. An interface bus may also be added to the HDL+ core, and it will show up in the SmartDesign with a bus interface pin that can be used to easily connect to the appropriate bus IP Core inside SmartDesign Canvas.
- Re-usable Design Blocks (*.cxz) file—Drag and drop the design blocks (*.cxz) from the Design Hierarchy into the SmartDesign canvas. Design blocks are components that may have completed layout in a different Libero SoC project and exported/published, to be ready for re-use in a higher level design as a component. See the *PolarFire FPGA Block Flow User Guide* for details. The design blocks must be imported into the current Libero SoC project to be instantiated in SmartDesign. Figure 2 shows how to instantiate a design block.

Figure 2 • **Design Block (*.cxz) Instantiation**



- SmartDesign Components—Another SmartDesign component (*.cxf) can be instantiated in the SmartDesign. Drag and drop the Smart Design component (*.cxf) from the Design Hierarchy into the SmartDesign canvas. Instantiation can also be done by right-clicking the (top) module name in the Design Hierarachy and choosing **Instantiate**. The component (*.cxf) and its corresponding HDL files and IP cores must be imported into the current project and configured before instantiating in the new SmartDesign project.

Figure 3 • **SmartDesign Component (*.cxf) Instantiation**


Design Hierarchy
Show: Components
work
XCVR_REF_CLK_N (XCVR_REF_CLK_N_syn_comps.v)
XCVR_REF_CLK (XCVR_REF_CLK_syn_comps.v)
XCVR_APB_LINK (XCVR_APB_LINK_syn_comps.v)
sdf
prep1
pf_pcie_to_ddr3_top
my_axi4_interconnect_w
my_pcie
my_transceiver_refCLK
pf_pcie_to_ddr3_top_COREAXI4SRAM_0_COREAXI4SRAM (Cor
pf_pcie_to_ddr3_top_COREAXI4SRAM_0_CoreAXI4SRAM_M
pf_pcie_to_ddr3_top_COREAXI4SRAM_0_CoreAXI4SRAM_M
pf_pcie_to_ddr3_top_COREAXI4SRAM_0_CoreAXI4SRAM_S
pf_pcie_to_ddr3_top_PF_TX_PLL_0_PF_TX_PLL (pf_pcie_to_ddr3
TX_PLL (TX_PLL_syn_comps.v)
mytop
mux8 (mux8.v)
G5_APBLINK_MASTER (g5_apbli
CoreAXI4SRAM_LIB
Set As Root
Instantiate in pf_pcie_to_ddr3_top
Open Component
Generate Component
Rename Component...
Create I/O Constraint from Module
Catalog
Design ...
Design Hier...
Stim
pf_pcie_to_ddr3_top*
g5_apblink_master.v
Reports
StartPage
my_axi4_interconnect_w_0
COREAXI4SRAM_0
my_pcie_0
PF_TX_PLL_0
my_transceiver_refCLK_0
mux4_1
mux4_2
[DELETED] mux4_0*

The functional or process block diagram(s) need to depict the organizing schema of the content.

2.4 SmartDesign Canvas and Component Display

The SmartDesign Canvas window can be docked/undocked by clicking the 🗗 icon. The canvas displays different component types with different colors. When the mouse is hovered over the component, a tooltip displays the type and name of the component. For IP Cores, the core version is also displayed.Table 1 lists the component types and their appearance in the SmartDesign canvas.

Table 1 • **Component Type and Name in Canvas**

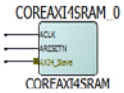
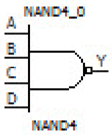
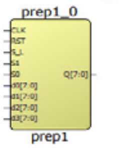
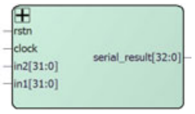
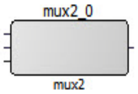
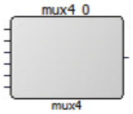
Graphic Display	Component Type	Tooltip Information
 COREAXI4SRAM_0 ACLK ARESETN AXI4_Slave COREAXI4SRAM	IP Cores from Catalog	Core: COREAXI4SRAM 2.1.105
	Basic Macros/Macro Library from Catalog	Macro: NAND4
	Re-usable Design Blocks (*.cxz file) imported into Libero SoC project	Block: prep1

Table 1 • **Component Type and Name in Canvas**

Graphic Display	Component Type	Tooltip Information
 rstn clock in2[31:0] in1[31:0] serial_result[32:0]	SmartDesign Components (*.cxf file) imported into Libero SoC project	SmartDesign: adder_shift32
	Design HDL file imported into Libero SoC project	HDL: mux2
	Core generated from HDL	HDL+: mux4

2.5 SmartDesign Tcl Commands

For details about the Tcl commands supported by SmartDesign, refer to the SmartFusion2, IGLOO2, RTG4 Tcl Commands Reference Guide or the PolarFire FPGA Tcl Commands Reference Guide.

3 SmartDesign Icons, Hotkeys and Menu Items

3.1 SmartDesign Icons

Across the top of the SmartDesign canvas is a list of icons. Use the icons to:

- Make connections
- Control the canvas display
- Invoke Design Rule Check
- Generate the HDL for the component
- Adding Text
- Save the SmartDesign to PDF

Table 2 lists the SmartDesign icons. Tool-tips are provided for each icon. Hover the mouse over the icon to display the tool-tips.

Table 2 • **SmartDesign Icons**

Icon	Action/Tool-tips	Descriptions
⚙	Generate Component	Generates the design. Converts the visual design created by the user into an HDL file for use in the project if there are no errors in the design. This action implicitly and automatically invokes the Design Rule check. DRC Error messages are generated in the Log/Message window. A DRC report is also generated (Design > Reports > <top_level_smart_design_name>_DRC.xml).Clicking this button affects the design state.
✓	Design Rule Check	Invokes the Design Rule Checker (DRC). If the design is not valid (DRC violations), then errors appear in the Log/Message window. A DRC report is also generated (Design > Reports > <top_level_smart_design_name>_DRC.xml). No HDL file is generated even if the DRC check passes. Clicking this button does not affect the design state. To generate the HDL for the component, click the Generate Component button.
ℹ	Toggle Highlighting Instances with Status Messages	Click this icon to highlight any instances on the canvas that have status information for the instance (e.g. “A new version is available”). Hovering the mouse over an instance opens a tooltip to display the info/message (e.g. “A new version is available”). Click this icon if you have made any changes to a component, reconfigured a core or have downloaded newer versions of the core used in the design.
⇥	Connection Mode	This mode allows the user to make connections between two pins more quickly. Click to enter the connection mode. The mouse cursor turns into a cross. Left-click on a pin, drag the mouse to another pin and then release the mouse. Click over the pin and a net is added between the two pins. It can also be used to connect a pin to a net or vice versa. In this mode, the right-click menu items are available when design objects are selected. In this mode, the drag down right is not available to select an area.

Table 2 • **SmartDesign Icons *(continued)***

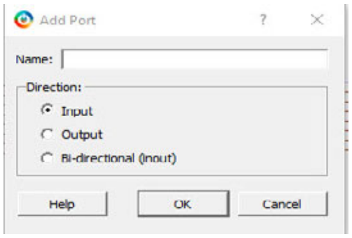
Icon	Action/Tool-tips	Descriptions
	Add Port	Add a new top-level port to the design. A pop-up window appears for you to enter the name of the port and the directions (Input/Output/Inout). Input Ports are added in the leftmost column of the canvas. Output and Inout ports are added in the rightmost column. After the port is added, hover the mouse over the input port to display the port name, direction and fanout. Hover the mouse over the output port to display the port name, direction and drivers of output port. Bus ports can also be added, e.g myInputBus[7:0]  Add Port ? Name: Direction: Help OK Cancel ● Input ○ Output ○ Bi-directional (Inout) Alternatively, right-click on empty space inside the canvas and choose Add Port. A port is added in the canvas and is anchored at the Y co-ordinates of cursor.
	Modify Memory Map	Click this icon to open a dialog to assign or change the connections based on the memory address. This option is highlighted only if the design includes peripheral connections to buses.
	QuickConnect	Opens a dialog that helps the user to make connections more quickly. This dialog lists the pins in the design and the user can select multiple pins and make a connection directly.
	Reset Layout	Click to reset the layout view. Clicking this button removes all presentation information (position, size, highlights, modified pin orders).
	Auto-arrange Layout	Click to redo the layout of where components are placed on the screen. Only the location (x-y coordinates) of the instances and ports are changed. All presentation information remain intact.
	Compress Layout	Click to push the instances and ports towards each other in order to remove extra white space between them on the screen. The relative positions of the instances on the screen are preserved. The result is a more compact display of the design.
	Hide Nets	Toggle button. Click to hide nets and make them invisible on the canvas. All nets on the canvas are hidden. This button has precedence over net filtering and over-rides all net filters. To hide some but not all nets, do not use this button. Use the Filter Net widget at the far right of the toolbar. Hiding nets also hides the net names (if present). When a net is hidden, the net stubs that the hidden net is connected to are still visible. Selecting the net stubs shows the RATS net connection of the net.
	Show Nets	Toggle button. Click to show/Hide nets. All nets on the canvas are shown/hidden. If nets are shown, then all nets matching the net filter (at the far right of toolbar) are shown.

Table 2 • **SmartDesign Icons *(continued)***

Icon	Action/Tool-tips	Descriptions
N	Show/Hide Net Names	Toggle button. Click to show net name displayed alongside the net. Hiding net names makes the canvas less cluttered for big designs. Net names are always displayed in a tooltip when the mouse is hovered above the net.
	Unhighlighted All	Remove all highlighting of all design objects (nets, pins, ports, instances) on the canvas. This option is highlighted only if design objects are already highlighted.
⤢	Expand All Instances	Expand (display the hierarchy to the lowest level) in place of all instances in the canvas.
	Fold All Instances	Collapse the hierarchy of all expanded instances into the top-level hierarchy.
🔍	Zoom In	Zoom in on the canvas
🔍	Zoom Out	Zoom Out on the canvas
✥	Zoom to Fit	Adjust the zoom so that everything on the canvas just fits snuggly inside of the visible viewport with no extra empty space around the design.
🔍	Zoom to Selection	Click this icon and drag the mouse to draw a rectangle which when released causes a zoom in so that the visible viewport area is approximately the size of the drawn rectangle.
▦	Show Grid	Click to show a background grid behind the items on the canvas. If the grid does not appear when the button is clicked, zoom in until the grid shows. The grid pattern may not show if the canvas is zoomed too far out.
A	Add Note	Click to enter the Add Note mode. The next mouse click on the canvas opens a dialog box for entering the text and font size for the text (anchored at the mouse click location).
	Save to PDF	Click this icon to bring up a dialog that allows the user to save a picture of all/part of their design to a PDF document.

3.2 SmartDesign Hotkeys

Table 3 lists the Hotkeys available in the SmartDesign canvas.

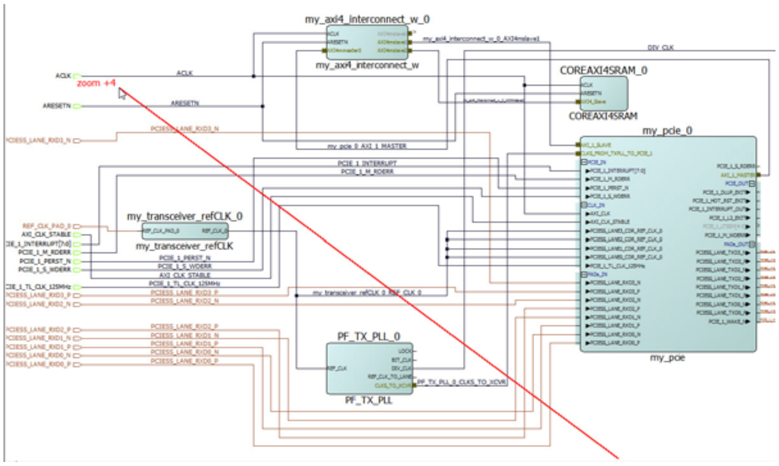
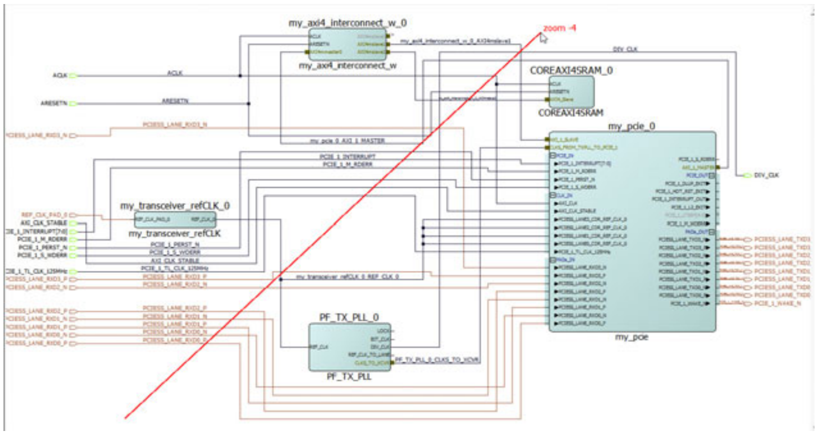
Table 3 • **SmartDesign Hotkeys**

Hotkey	Descriptions
CTRLCTRL + w	Maximize the canvas work area
CTRL + f	Opens the Find dialog box
CTRL + +/CTRL + =	Zoom in, same as clicking the Zoom In button in toolbar. or CTRL+ mouse scroll wheel up.
CTRL + -	Zoom out, same as clicking the Zoom Out button in toolbar, or CTRL + mouse scroll wheel down.
CTRL + c	Clear the clipboard. If a single instance is selected before the Hotkey then it is added to the clipboard. The display name of the design object is also copied.
CTRL + v	If the copy information of a single instance is in the clipboard then call clone instance on it. For details, see “Help” on page 17 for details.
CTRL + z	To undo the last operation
CTRL + y	To redo the last operation
SHIFT + click	Click one item and shift-click another item. The first clicked item, the second clicked item and all items of the same type between the first and second clicks are selected. This command is helpful when multiple items needed to be selected for the same command (promotion to top level/Add to group).
CTRL + click	Control-click is a toggle switch that selects or de-selects an item under the mouse cursor.
CTRL+SHIFT+click and then drag	Selects a pin of an instance and drag the mouse to a new location to move the pin of the instance. This is not available for macros and expanded in place instances.

3.3 Click and Drag Operations

Table 4 lists the mouse click-and-drag operations available in the SmartDesign canvas.

Table 4 • **Mouse Click-and-Drag Actions**

Mouse Actions	Descriptions
Left-click and drag towards upper left corner and then release.	Zoom in. The distance of the mouse travel determines the magnitude of the zoom-in and is indicated by a positive integer in red.  zoom +4 my_axi4_interconnect_w_0 my_axi4_interconnect_w COREAXI4SRAM_0 COREAXI4SRAM my_pcie_0 my_pcie my_transceiver_refCLK_0 my_transceiver_refCLK PF_TX_PLL_0 PF_TX_PLL
Left-click and drag towards upper right corner and then release.	Zoom out. The distance of the mouse travel determines the magnitude of the zoom-out and is indicated by a negative integer in red  zoom -4 my_axi4_interconnect_w_0 my_axi4_interconnect_w COREAXI4SRAM_0 COREAXI4SRAM my_pcie_0 my_pcie my_transceiver_refCLK_0 my_transceiver_refCLK PF_TX_PLL_0 PF_TX_PLL
Left-click and drag towards lower left and then release	Zoom to fit. Change the display to fit the canvas view area snugly.
Left-click and drag toward lower right and then release	Draw a rectangle on the canvas. Instances, pins and ports must be fully contained inside the rectangle to be selected. Nets partially inside the rectangle are selected.

4 SmartDesign User Actions

The SmartDesign tool has a rich set of menu items, accessible from the right-click menu of a selected design object, for the user to:

- Make connections between design objects
- Rename design objects
- Highlight design objects
- Traverse up/down the hierarchy of instances
- Assign attributes of pins/nets/buses.

The list of available user actions varies with the design object. Some actions are common to all while others are unique to the design objects (net/instances/pins/ports).

4.1 Net Actions

When a net or multiple nets are selected, the following actions are available in the right-click menu item of the net/nets.

4.1.1 Connect

This action combines with all selected pins/ports to form a connection.Selecting a net is the same as if all pins/ports this net is connected to are selected instead. It can also connect a net to another net if one of the net is not driven.

4.1.2 Go to Driver

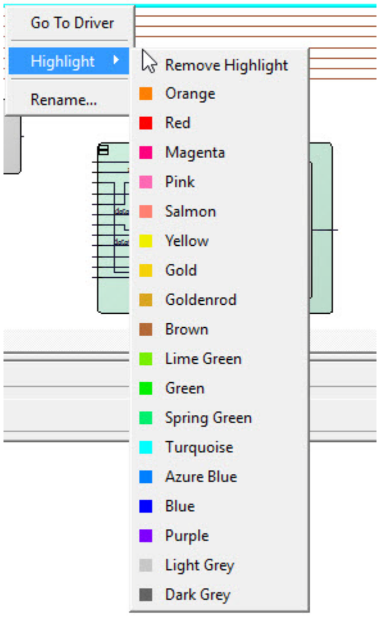
When a net is selected, the Go to Driver action centers the view on the net’s driver pin/port, zooms away, and then selects the net driver. The driver must be an input port/output pin. It cannot be an inout pin/port. This action is not available when multiple nets are selected or the selected net has no driver. Go to Driver traces the net to the driver at the local level of hierarchy. It does not traverse hierarchy.

4.1.3 Highlight

Right-clicking a design object (nets/pins/ports/bus interface/instance) menu opens a menu of colors for highlighting. Clicking a color selects that color to highlight the selected design objects. If any design objects are already highlighted then highlighting a different color will overwrite the previous highlight color. The highlight action is available when a single or multiple design objects are selected.

Note: Highlighting a net highlights the net and all the pins/ports (through the hierarchy) connected to the net.

Figure 4 • **Highlight Colors**



To remove the highlight on a single design object, right click the design object and select **Remove Highlight.**

To remove all highlighted design objects, click the **Unhighlight All** icon in the toolbar.

4.1.4 Rename

The rename action opens a dialog for a new net name to be entered to replace the old name. Clicking OK changes the net name to the new net name if it is valid (e.g. not already used). Clicking OK prints an error in the Log window and the dialog does not close if the name already exists or is invalid.

4.1.5 Delete

Deletes the net. Net names, if shown, are also deleted.

4.2 Instance Actions

When an instance or multiple instances are selected, the following actions are available in the right-click menu item of the instance.

4.2.1 Configure

If the instance is a SmartDesign component, the Configure action opens the SmartDesign canvas for edits. If the selected instance is an IP Core, the Configure button opens the Configurator dialog for the core to be configured. If the selected instance is an HDL, the configure action opens the HDL file in the text editor. This action is equivalent to double-clicking the instance. This action is available only when one single instances is selected.

This action is also available in the right-click menu of low level instances in the Expanded Inplace view.

4.2.2 Modify HDL

This action opens the instance’s HDL source file in the text editor. Available only when an instance of an HDL+ component is selected.

Note: If the component is an HDL file, this option is not available. To open an HDL file, double-click the HDL component on the SmartDesign canvas or right-click the HDL component and choose **Configure**.

4.2.3 Highlight

This action opens a menu with multiple highlight color selections. Clicking that color selects that color to highlight the selected items. If any items are already highlighted then highlighting a different color will overwrite the previous highlight color. The action is available when a single or multiple nets are selected.

Highlighting an instance automatically highlights the non-highlighted pins of the instance as well.

Clicking the Unhighlight all icon in the toolbar removes the highlight color of all highlighted design objects, including highlighted nets.

This action is also available in the right-click menu of low level instances in the Expanded Inplace view.

4.2.4 Rename

This action opens the rename dialog for a new instance name to be entered. Clicking OK will change the instance to this name if it is valid and then close the dialog.

Clicking OK prints an error in the Log window and causes a popup with the warning message and won't close if the name already exists or is invalid.

4.2.5 Copy/Paste

Copy and Paste are used to copy instance to the clipboard and paste the copied instance on the canvas.

The copy action is also available in the right-click menu of low level instances in the Expanded Inplace view.

Copying an IP Core on the canvas copies into the clipboard the following:

- The configuration of the core
- Pin attributes (Tie high/low/Mark Unused/Inverted)
- Name of the instance
- The input pin connections -- Available only if pasted inside the same SmartDesign component
- Presentation Information, e.g. Highlighting, size, pin order - Available only if pasted inside the same SmartDesign component

Note: If a change is made to the instance after the copy but before the paste command, the paste command uses the most current version of the instance, not the version at the time of copy.

Note: It does not support copying from another Libero SoC project.

4.2.6 Delete

This action deletes the selected item. When multiple items are selected, all are deleted.

Note: Not all design objects can be deleted. Pad ports cannot be deleted.

4.2.7 Expand Inplace/Fold Instance

The Expand Inplace action is equivalent to clicking the + (Expand) sign in the upper left corner of the instance. It causes the selected instance to expand (to expose all levels of hierarchy) in place if not expanded. It is available to all instances at all levels of hierarchy except the lowest level. If only one instance is selected then after the expand/fold is completed, the viewport will zoom to the instance. This action is for SmartDesign components and is read-only. No changes can be made to the expanded hierarchy. Changes to the components at the lower level of hierarchy must be made by opening the low level component directly.

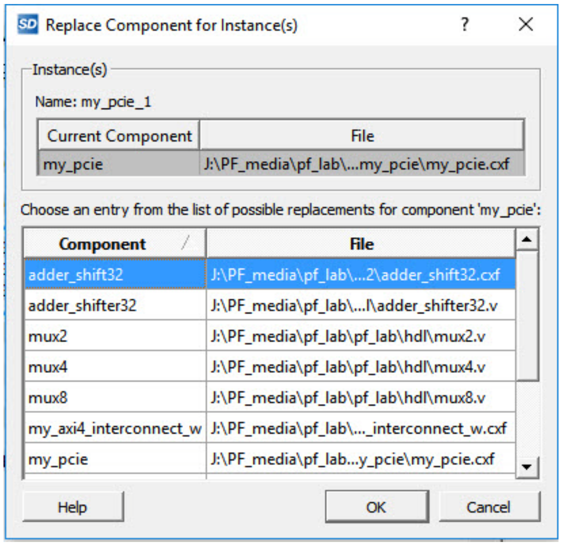
See “Expand/Fold Instance” on page 35for details.

4.2.8 Replace Component

This action brings up the Replace Component for Instance(s) dialog. Selecting an instance and then clicking OK replaces every instance of that component in the current design with an instance of the new component if that component is valid. All pins that have the same name on the new component will keep their connections, whereas pins that no longer exist will lose their connection and a warning will be printed in the Log window. The dialog will then close.

Selecting an instance and then clicking OK for a non-valid component will close the dialog and print an error in the Log window

Figure 5 • **Replace Component Dialog.**



Instance(s)

Name: my_pcie_1

Current Component	File
my_pcie	J:\PF_media\pf_lab\...my_pcie\my_pcie.cxf

Choose an entry from the list of possible replacements for component 'my_pcie':

Component	File
adder_shift32	J:\PF_media\pf_lab\...2\adder_shift32.cxf
adder_shifter32	J:\PF_media\pf_lab\...l\adder_shifter32.v
mux2	J:\PF_media\pf_lab\pf_lab\hdl\mux2.v
mux4	J:\PF_media\pf_lab\pf_lab\hdl\mux4.v
mux8	J:\PF_media\pf_lab\pf_lab\hdl\mux8.v
my_axi4_interconnect_w	J:\PF_media\pf_lab\..._interconnect_w.cxf
my_pcie	J:\PF_media\pf_lab...y_pcie\my_pcie.cxf

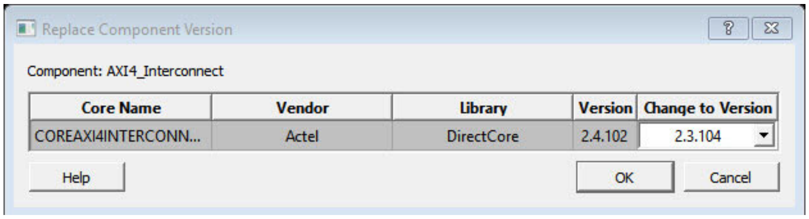
4.2.9 Update Component

This action updates all instances of that component in the current SmartDesign with the newest version of that component. This is available only when the port list has changed. Changing the port list of an instance causes it to return to a new default size.

4.2.10 Replace Component Version

This action exists for IP cores. Under Design Hierarchy, select an IP core, right-click, and choose **Replace Component Version**. This opens the Replace Component Version dialog.

Figure 6 • **Replace Component Version Dialog**



Component: AXI4_Interconnect

Core Name	Vendor	Library	Version	Change to Version
COREAXI4INTERCONN...	Actel	DirectCore	2.4.102	2.3.104

Changing the version in the dialog and then clicking **OK** changes all instances of that component in the current SmartDesign canvas to the specified version of that component.

4.2.11 Modify Memory Map

This action opens the Modify Memory Map dialog for you to connect peripherals to buses. To open the dialog box, right-click the bus instance in the canvas and choose Modify Memory Map. Alternatively, click the Modify Memory Map icon in the toolbar .

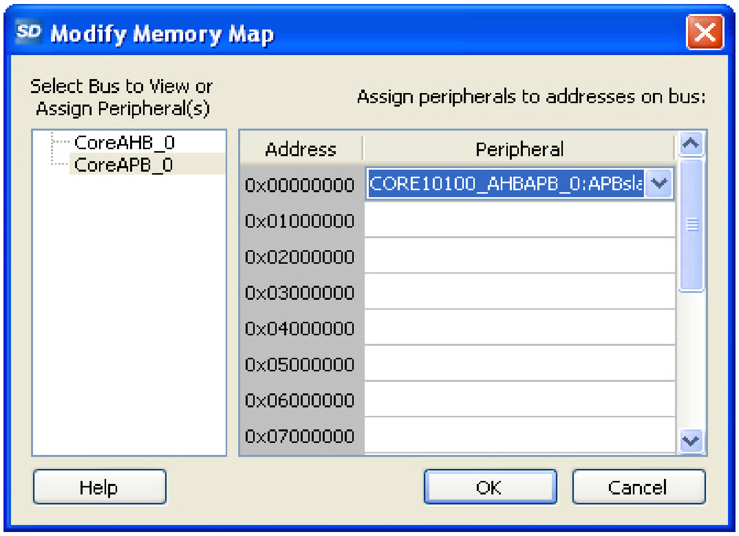
If this action is accessed from the right-click menu of an instance, the instance is selected in the Modify Memory Map dialog box by default.

This dialog box simplifies connecting peripherals to specific base addresses on the bus. The dialog box shows all the buses in the design; select a bus in the left pane to assign or view the peripherals on a bus. Busses that are bridged to other busses are shown beneath the bus in the hierarchy.

Click the Peripheral drop-down menu to select the peripheral you wish to assign to each address. To remove (unassign) a peripheral from an address, click the drop-down and select the empty element.

Click **OK** to create the connections between the busses and peripherals in the design.

Figure 7 • **Modify Memory Dialog**



Modify Memory Map

Select Bus to View or Assign Peripheral(s)

- CoreAHB_0
- CoreAPB_0

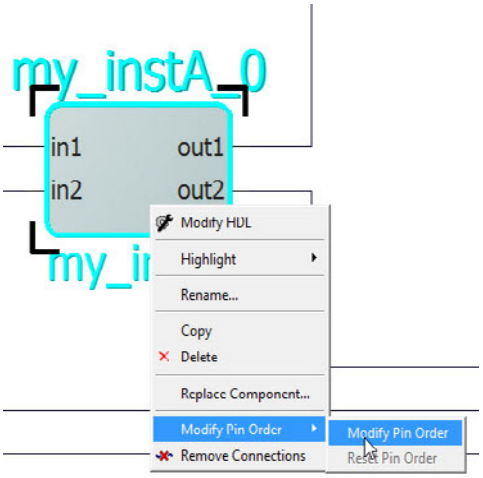
Assign peripherals to addresses on bus:

Address	Peripheral
0x00000000	CORE10100_AHBAPB_0:APBsla
0x01000000	
0x02000000	
0x03000000	
0x04000000	
0x05000000	
0x06000000	
0x07000000	

4.2.12 Modify Pin Order

This a toggle switch. Right-click an instance to open the drop-down menu. Select Modify Pin Order. A pop-up window appears with instructions on how to reset pin order. Press CTRL+SHIFT and left click to select the pin you want to move.

Figure 8 • **Modify Pin Order**



Drag the mouse to a new location and release the mouse. The pin is moved to a new location. By default, input pins are on the left of the instance while output pins/inout pins are on the right. Modify Pin Order allows you to place the pins on any one of the four sides of the instance. A pin that has been moved away from default locations is identified by a bold arrow head. An inward-pointing arrow head indicates an input pin and an outward-pointing arrow head indicates an output pin. Inout pins do not have an arrow head when they are moved away from the default locations (right side of instance).

Note: If there are two instances on the canvas that communicate with each other, such that the outputs of one component instance A communicates with the inputs of another component instance B and vice versa, rmodifying the pin order gives a less cluttered view of the SmartDesign component. See “Modify Pin Order Before Connections” on page 31 for details.

Note: Modify Pin Order is disabled when the instance is expanded in place. The modified pin order may not be preserved when an instance is expanded but will retain the set order when folded.

4.2.13 Reset Pin Order

This is a toggle switch. Reset pin order reverses the modification and resets the pins to the default location. If the user has moved an instances pin order, this action unresizes the instance and reload the order from the SmartDesign model. Note that if the user moves a pin and then moves it back to the original position this will still be enabled (and then do nothing but resize).

Figure 9 • **Modify/Reset Pin Order**

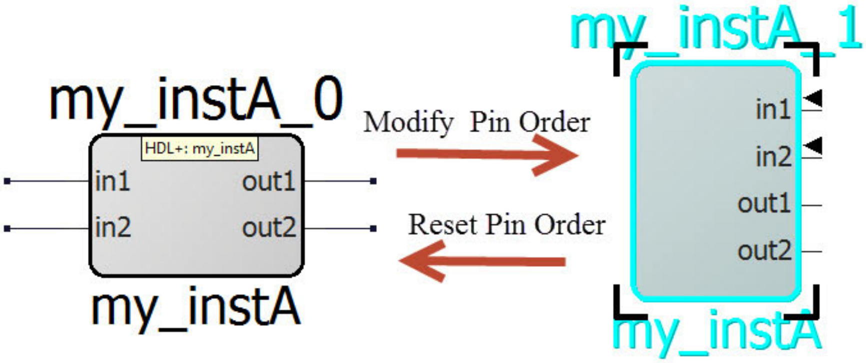
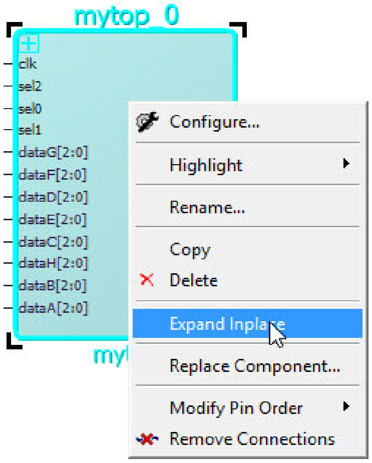
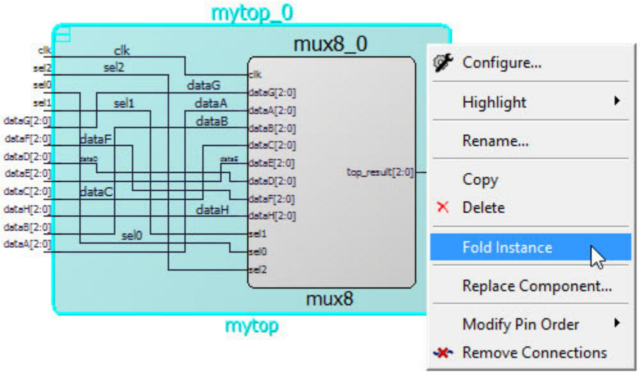


Figure 10 • **Expand in Place**



The Fold Instance is a toggle switch. It is equivalent to clicking the - (Fold) sign of an expanded instance. It causes the expanded hierarchy of the instance to collapse to the top level.

Figure 11 • **Fold Instance**



4.2.14 Remove Connections

This action disconnects all pins that can be disconnected. Pins that cannot be disconnected (for example pins connected to pads) are logged in the Log window.

4.2.15 Help

Opens the Core Handbook, Release Notes or Configuration User Guides for the core.

4.3 Pin/Port Actions

Right-click a pin or a port and the drop-down menu appears. Not all actions appear for every port/pin.

4.3.1 Connect

The Connect action adds a net to the design that connects the selected pins/ports. This is the only pin/port action that takes selected nets into account. Selecting a net works as if the user had instead selected all of the nets connected pins/ports.

4.3.2 Disconnect

This action disconnects all selected non-pad pins/ports from their attached net if it is allowed (not violating the DRC rules).

4.3.3 Promote to Top Level

This option is available to input/output/inout pins (scalar and bus), slices, and Bus Interfaces (BIF). It creates a top-level port and a net connecting the top-level port to the pin or slices. If a port with that name already exists, a new unique port name is created.

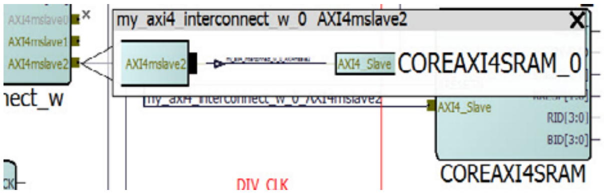
4.3.4 Go to Driver

This action centers on, zoom around, and selects the driver of the pin/port. This action is not available for output pins and top-level input ports. The driver cannot be an inout.

4.3.5 Magnify Pin

This action opens the Magnify window to display more information about the pin. It is equivalent to double-clicking the pin. It displays connection information (driver/load) of the pin.

Figure 12 • **Magnify Pin Window**



See “Magnify Pin” on page 38 for details.

4.3.6 Highlight

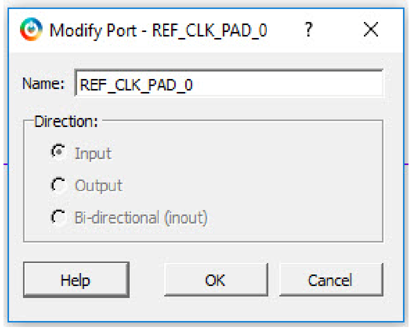
This action opens a menu with multiple highlight color selections. Clicking that color selects that color to highlight the selected items. If any items are already highlighted then highlighting a different color will overwrite the previous highlight color. The action is available when a single or multiple pins/ports are selected.

Clicking the Unhighlight all icon in the toolbar removes the highlight color of all highlighted design objects, including highlighted pin/ports.

4.3.7 Modify/Rename

This action opens a Modify Port dialog.

Figure 13 • **Modify/Rename Dialog**



Modify Port - REF_CLK_PAD_0

Name: REF_CLK_PAD_0

Direction:
- ● Input
- ○ Output
- ○ Bi-directional (inout)

It allows the top-level port name and the range to be changed.

Note: All slices of the bus are deleted if the range is changed.

4.3.8 Delete

This action deletes all selected items that can be deleted: slices, groups, group members, top-level ports.

The delete action deletes all items selected, even if the selected items are of different types. When a group member is deleted, the member is deleted from the group only. The actual pin is not deleted.

4.3.9 Expanding and Collapsing Bus

Expanding a bus displays the slices of the bus and collapsing the bus hides the slices. Slices with net connections cannot be hidden and cannot be collapsed into the bus.

4.3.10 Flip Bit Order

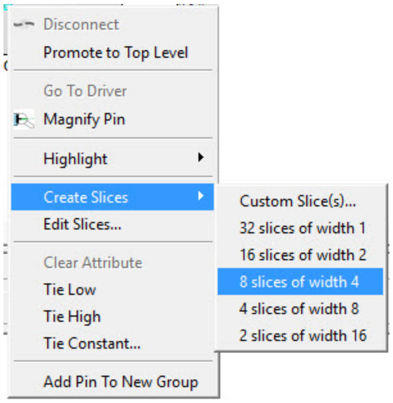
This option is available only to slices. It allows you to flip the upper range (MSB) and lower range (LSB) of the slice. All connection/tieoff information and presentation information are retained.

4.3.11 Create Slices

Open a menu of slice options that can be created from the bus pin/ports. A custom slice option to create any slice/bit combination of your choice and the more common possible combinations of slices are available. Take a 32-bit bus as an example. The Custom Slices option allows the creation of any slice/bit combinations, e.g. a slice of 10 bits and another slice of 22 bits. To make it convenient to create slices, the more common, off-the-shelf slices, of the bus, 32-bit in this case, are listed in the drop-down menu:

- 32 slices of width 1
- 16 slices of width 2
- 8 slices of width 4
- 4 slices of width 8
- 2 slices of width 16

Figure 14 • **Slice Creation for a 32-bit Bus**



On a components bus pin the slices will be expanded by default.

The directions of the slices (input/out) are indicated by an arrow head.

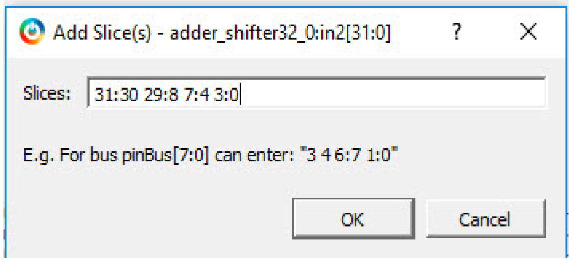
On a top-level bus port the slices will be placed in a column below the bus port.

Note: If slices exist before new off-the-shelf slices are created, the existing slices are deleted first before new ones are created.

4.3.12 Custom Slices

This option opens a dialog for entering a list of slices. If these slices are all valid then they are added to the bus. If the slices are not valid (e.g. slice bits already exist and used in an existing slice), the error is reported in the Log window. The dialog supports any separator character except colon because the colon is used to specify a range. No characters other than the colon is allowed adjacent to the two range indices).

Figure 15 • **Custom Slices Dialog**



Slices: 31:30 29:8 7:4 3:0

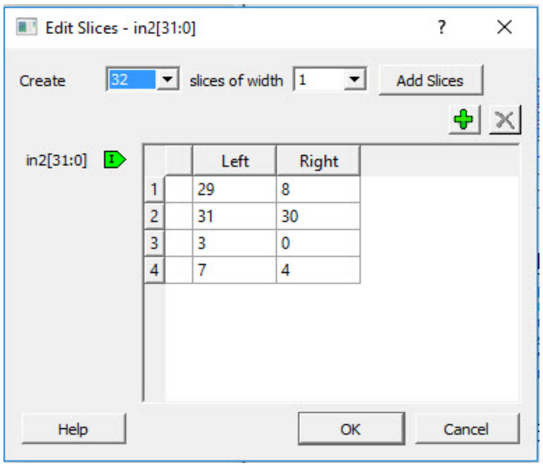
E.g. For bus pinBus[7:0] can enter: "3 4 6:7 1:0"

Note: Creating a custom slice does not delete pre-existing slices.

4.3.13 Edit Slices

The Edit Slice option opens the Edit Slices dialog. It allows existing slices to be modified.

Figure 16 • **Edit Slices Dialog**



Edit Slices - in2[31:0]

Create 32 slices of width 1 | Add Slices

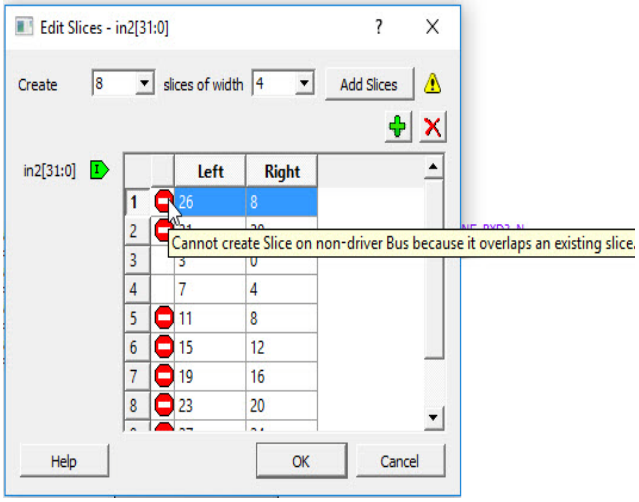
in2[31:0]

	Left	Right
1	29	8
2	31	30
3	3	0
4	7	4

Use the dialog to change the range of the bits, add a slice or delete a slice. Modifying the slices then clicking **OK** initiates the changes if they are valid and close the dialog.

Modifying the slices then clicking **OK** prints an error in the Log and close the dialog if the changes are not valid. Some error messages may be reported in the Edit Slice dialog. Hover the mouse over the error icon to display details of the error.

Figure 17 • **Tooltip and Error Message**



Edit Slices - in2[31:0]

Create 8 slices of width 4 | Add Slices

in2[31:0]

		Left	Right
1	⛔	26	8
2	⛔	31	30
3		3	0
4		7	4
5	⛔	11	8
6	⛔	15	12
7	⛔	19	16
8	⛔	23	20

Cannot create Slice on non-driver Bus because it overlaps an existing slice.

4.3.14 Clear Attributes

This action clears the pin attributes (Tie to High/Low/Constant, Inversion/Marked Unused).

4.3.15 Mark Unused

This is available to the output pins (scalar and bus) of a component instance. This action specifies the pin/port as unused so it can pass DRC check without being connected.

4.3.16 Invert

This action inverts the input/output scalar pin and port. A bubble is added to indicate inversion.

4.3.17 Tie High/Low

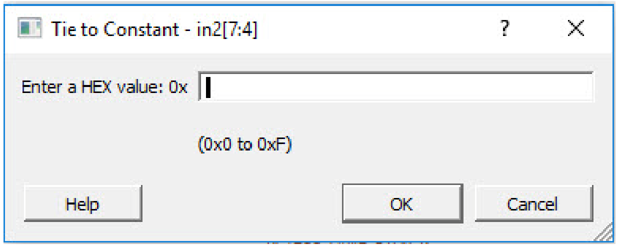
The Tie High action connects the pin (scalar and bus) to VCC. For a bus pin this action deletes all slices. For a group this action is applied to all non-output member pins in the group.

The Tie Low action connects the pin (scalar and bus) to Ground. For a bus pin this action deletes all slices. For a group this action is applied to all non-output member pins in the group.

4.3.18 Tie Constant

This action is available only to bus pins and slices (except single-bit slice). It opens the Tie to Constant dialog box for a constant value in HEX to be entered for the bus pins and slices. Only valid values in HEX (given in parenthesis) are allowed.

Figure 18 • **Tie to Constant Dialog**



Tie to Constant - in2[7:4]

Enter a HEX value: 0x

(0x0 to 0xF)

4.3.19 Add Pin to New Group/Add Pin to Group

This menu item is available to instance pins. If a group is selected first, right-click a pin and choose **Add Pin to Group** to add the pin to the selected group. If no group is selected first, the menu item is called “Add Pin to New Group”, a new group with the default group name Group/Group_1/Group_2/Group_3/ and so on is created and then all selected pins are added to the newly created group.

A pin group can be expanded to display the member pins or collapsed to hide the member pins. Member pins connected to nets cannot be hidden and cannot be collapsed into the group. Only member pins (in a pin group) with Pin Attributes (Tie high/low/marked unused) can be hidden and collapsed into the pin group.

4.3.20 Rename (Group)

This is available only to group pins. It opens the Rename Group dialog for the group name to be changed. If a valid name is entered then clicking **OK** renames the group. The new name field defaults to the current group name if no new name is entered in the dialog.

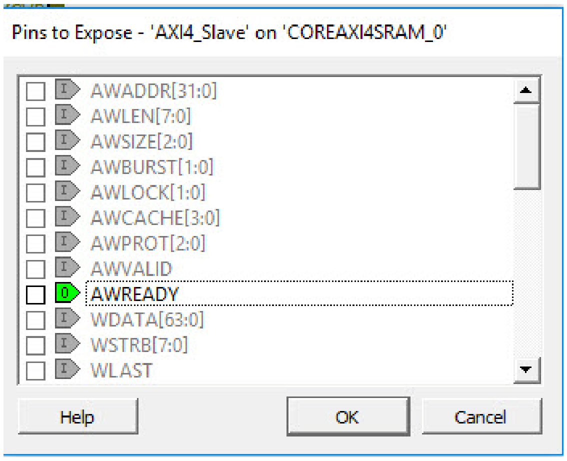
If an invalid name is entered or a pin with that name already exists then clicking **OK** prints an error in the Log and closes the dialog.

4.3.21 Show/Hide BIF Pins

This opens the Show Pins to Expose dialog to display all the Bus Interface Pins available to be exposed or hidden from the instance. Check the pins in the BIF you want to expose and uncheck the pins you want to hide. Hidden pins will not be exposed on the interface.

Note: The pins that can be exposed is dependent on the BIF. Not all pins can be exposed. If the BIF is connected already, then none of the input pins can be exposed. If the BIF is not connected, then every item in the menu can be exposed.

Figure 19 • **Show to Expose Dialog**



Pins to Expose - 'AXI4_Slave' on 'COREAXI4SRAM_0'

- □ I AWADDR[31:0]
- □ I AWLEN[7:0]
- □ I AWSIZE[2:0]
- □ I AWBURST[1:0]
- □ I AWLOCK[1:0]
- □ I AWCACHE[3:0]
- □ I AWPROT[2:0]
- □ I AWVALID
- □ O AWREADY
- □ I WDATA[63:0]
- □ I WSTRB[7:0]
- □ I WLAST

5 Designing with SmartDesign

SmartDesign is a GUI-driven block-based design entry tool for the instantiation, configuration and connection of various types design blocks. The SmartDesign canvas is analogous to a breadboard where the components of different types are assembled (instantiated), stitched together (connections made via nets) to create a design-rule-checked synthesis-ready HDL file for the complete FPGA design process.

The steps of designing with SmartDesign are:

1. Create a top-level SmartDesign component (analogous to the breadboard)
2. Configure/Instantiate components on the top-level SmartDesign
3. Making the connections (analogous to making wire connections to the different components)
4. Add top-level ports
5. Invoke DRC (Design-rule-check) on the design
6. Generate the top-level component

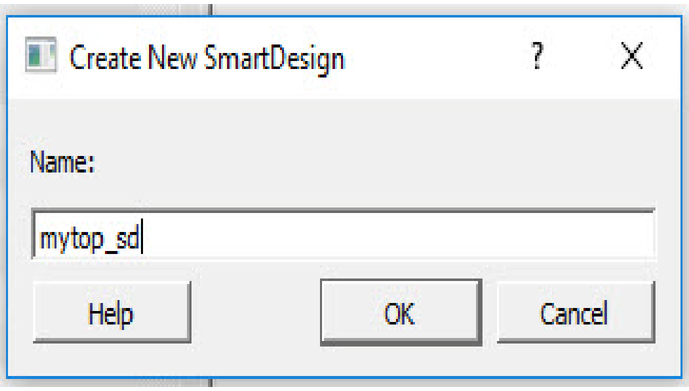
5.1 Create a Top-Level SmartDesign Component

A SmartDesign Component must be first created. This SmartDesign component may be the top level of the design or it may be used as a lower level SmartDesign component (after successful generation) in another design.

To create a SmartDesign Component:

1. From the File menu, choose **New > SmartDesign** or in the Design Flow window or double-click **Create SmartDesign**. The Create New SmartDesign dialog box opens.

Figure 20 • **Create New SmartDesign**



Name:

mytop_sd

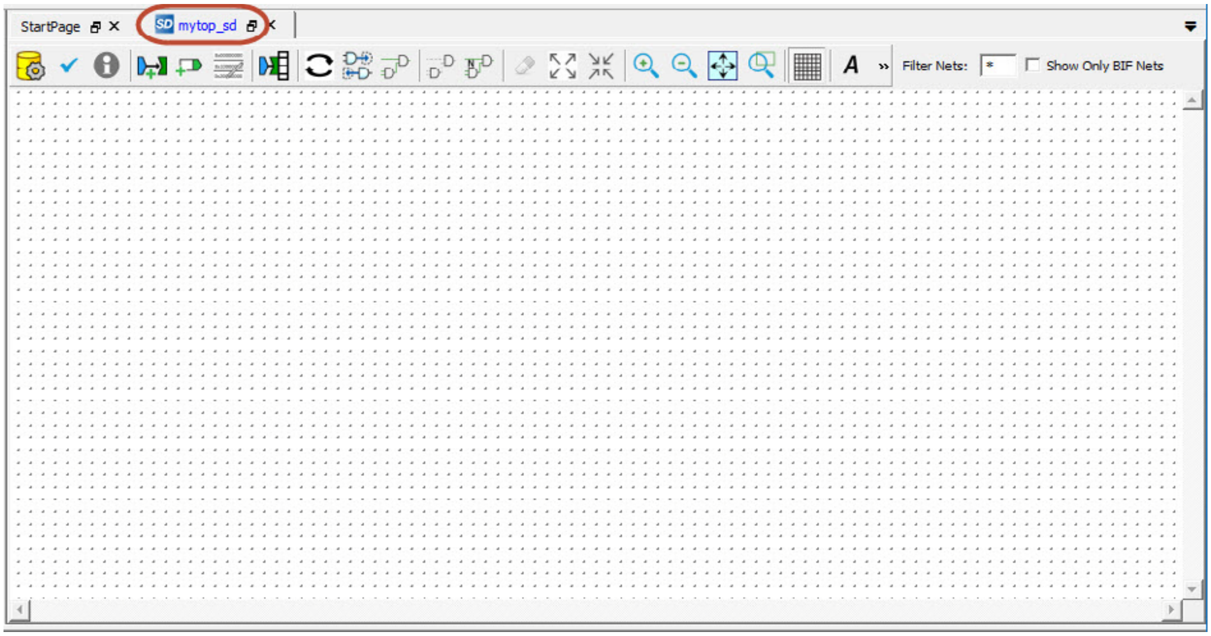
2. Enter a name and click **OK**. The component appears in the Design Hierarchy tab of the Design Explorer.

Note: The component name must be unique in your project.

The main window displays the SmartDesign canvas (the breadboard) with the component name you have entered displayed in a tab across the window.

Note: Figure 21 shows the SmartDesign Canvas with the Grid turned on. By default, the grid is turned off.

Figure 21 • **SmartDesign Canvas with Grid Turned On**



5.2 Configure/Instantiate Components

5.2.1 Configure

This step is required for IP cores such as Clock Conditioning Circuitry (CCC), DDR3/4 Memory Controllers, SRAMs, AMBA Bus Interface cores, and Transceiver Interface cores. These cores are available in the Catalog of the Design Explorer. If the core name appears in italics, double-click the core to download the core from the Microsemi IP Core Repository to your hard disk (the vault) first. Double-click the core in the catalog to open the Core's Configurator to configure the core. Refer to the respective core's Configuration User Guide for details.

5.2.2 Instantiate

The SmartDesign canvas accepts the following component types for instantiation:

- Configured Microsemi IP Cores
  - Drag and drop from the Catalog
  - IP cores must first be configured before they can be instantiated. If the IP core is not configured before instantiation, the drag-and-drop operation invokes the Core Configurator.for the core to be configured.
- User-generated or third-party IP Cores
  - Drag and drop from the Catalog
- HDL design files
  - Drag and drop from the Design Hierarchy
- HDL + design files (from the Design Hierarchy)
  - HDL+ design files are parameterized HDL design files or HDL files with Buses attached
  - Drag and drop from the Design Hierarchy
- Basic macros
  - Drag and drop from Macro Library section in the Catalog
- Other SmartDesign components
  - These are *.cxf files generated from SmartDesign in the current Libero SoC project or imported from other Libero SoC projects.
  - Drag and drop from the Design Hierarchy
- Re-usable design blocks
  - These are *.cxz files published from Libero SoC and used as design blocks

- Drag and drop from the Design Hierarchy

Each of these design components, after instantiation, is identified by different colors in the Design Canvas A tooltip pops up to provide details of the component when the mouse is hovered over the component.

Table 5 • Component Types in the SmartDesign Canvas and Tool-tip

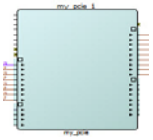
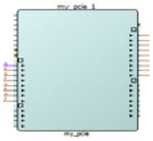
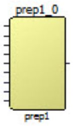
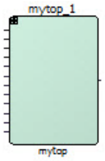
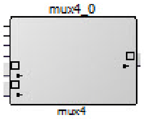
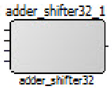
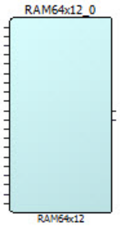
Icon	Type of Design Components	Tooltip Information
	Configured IP Core Component	Component: my_pcie Core: PF_PCIE 1.0.217
	IP Core directly instantiated from the Catalog	Core: PF_PCIE 1.0.217
	Block (*.cxz) file	Block: prep1
	SmartDesign component (*.cxf) file	SmartDesign: mytop
	HDL+	HDL+: mux4
	HDL	HDL:adder_shifter32

Table 5 • Component Types in the SmartDesign Canvas and Tool-tip *(continued)*

Icon	Type of Design Components	Tooltip Information
	Macro	Macro: RAM64x12

The tooltip displays information about the component:

- Type of component
- Name of component
- Version of Core (For IP cores only)

5.2.3 Importance of .cxf files in SmartDesign

SmartDesign components can be instantiated in another SmartDesign component by dragging and dropping them into the SmartDesign canvas. The low level SmartDesign components in the hierarchy must be imported before instantiating in a new project. This is done by selecting **Import > Components** in the File menu. The HDL and corresponding cores associated with these components must also be imported and configured before instantiating in the new SmartDesign component. If you just import the .cxf file and do not import files corresponding to the low level instances, you will see those instances highlighted in red and you will be informed of missing components. See the following example figure.

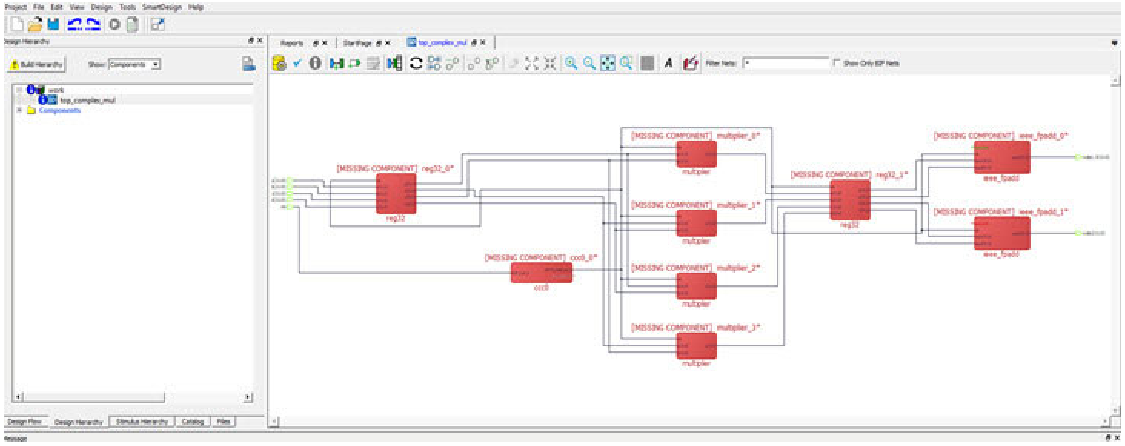

Project File Edit View Design Tools SmartDesign Help
Design Hierarchy
Build Hierarchy
Show: Components
work
top_complex_mul
Components
Reports
StartPage
top_complex_mul
Filter Nets:
Show Only EP Nets
[MISSING COMPONENT] reg32_0*
reg32
[MISSING COMPONENT] ccc0_0*
ccc0
[MISSING COMPONENT] multiplier_0*
multiplier
[MISSING COMPONENT] multiplier_1*
multiplier
[MISSING COMPONENT] multiplier_2*
multiplier
[MISSING COMPONENT] multiplier_3*
multiplier
[MISSING COMPONENT] reg32_1*
reg32
[MISSING COMPONENT] ieee_fpadd_0*
ieee_fpadd
[MISSING COMPONENT] ieee_fpadd_1*
ieee_fpadd
Design Flow
Design Hierarchy
Stimulus Hierarchy
Catalog
Files
Message

Figure 22 • **Errors While Importing SmartDesign Components with HDL Files and IP Cores**

Using .cxf files is important in the following cases:

- When doing multiple designs with very few changes in the original design blocks.
- When dividing a large design into smaller designs with very few additional blocks.

Using .cxf files provides the following advantages:

- Significant time savings (decreased generation time)
- Reduces errors in manual connections

See the following design example.

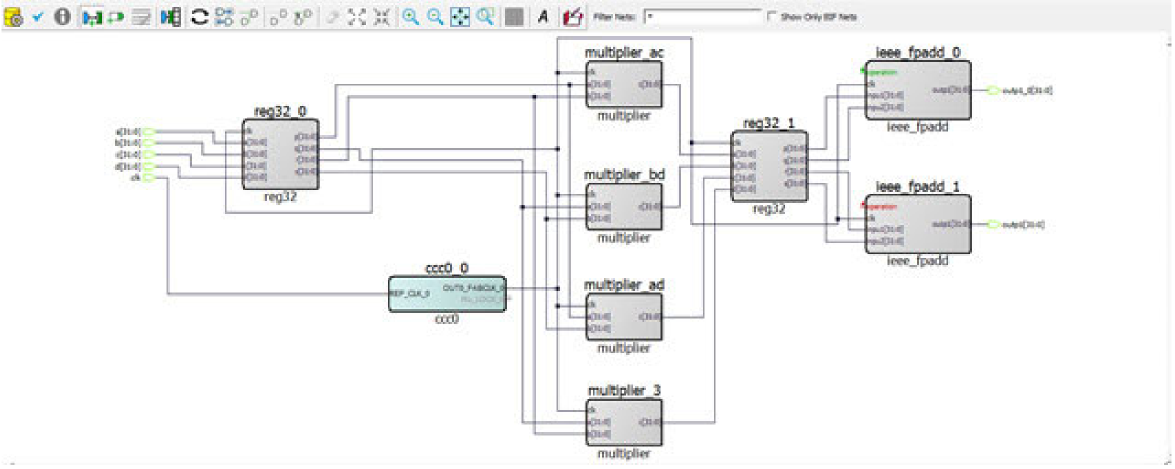


Figure 23 • **SmartDesign Example for Complex Multiplication**

The example design above has clocking blocks, multiplier blocks, and adder blocks. You could separate this design into two separate designs—complex_real and complex imaginary—in either of the following ways:

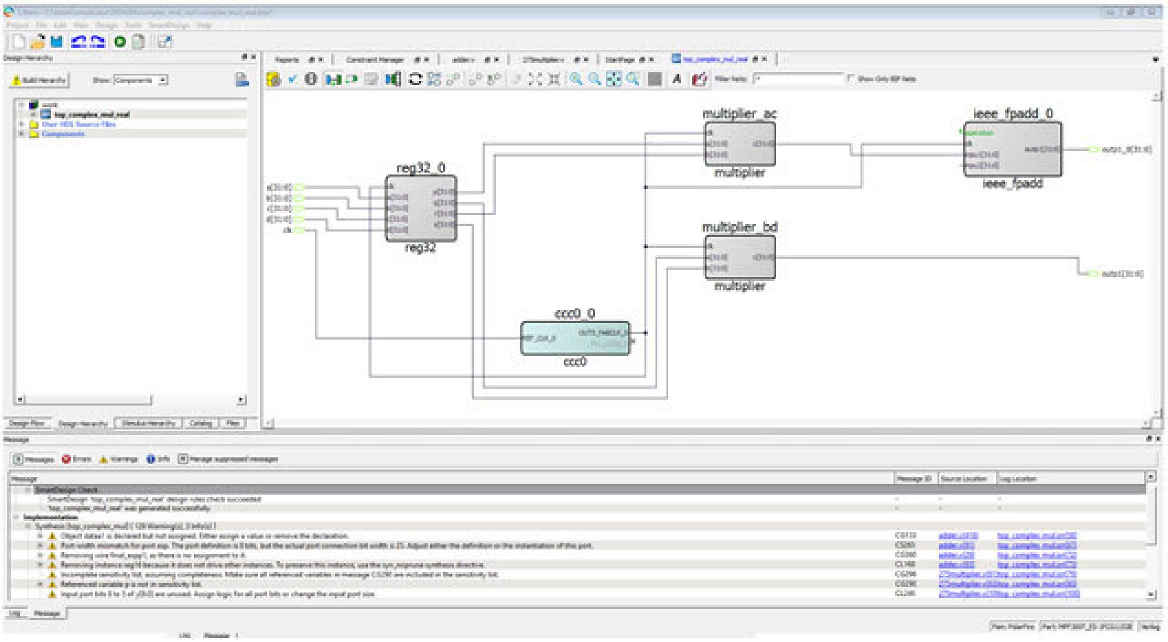
1. Without using .cxf files—create two new projects, import and configure all necessary HDL files and IP cores, and make the connections manually.
2. Using .cxf files—From the Project menu, save the original project with a new name using the **Save As** option. Make the necessary changes in the new project by deleting the blocks and ports you no longer need and adding additional blocks (if necessary), restoring the connetions between existing blocks from original design. This saves time while dividing large designs into smaller designs, without the need to make all the connections manually.

To save the original project with a new name, from the Project menu,select the **Save As** option. Click **OK** after entering the new project name.

To change the name of the SmartDesign component, right-click the componet in the Design Hierarchy and choose **Rename Component**.

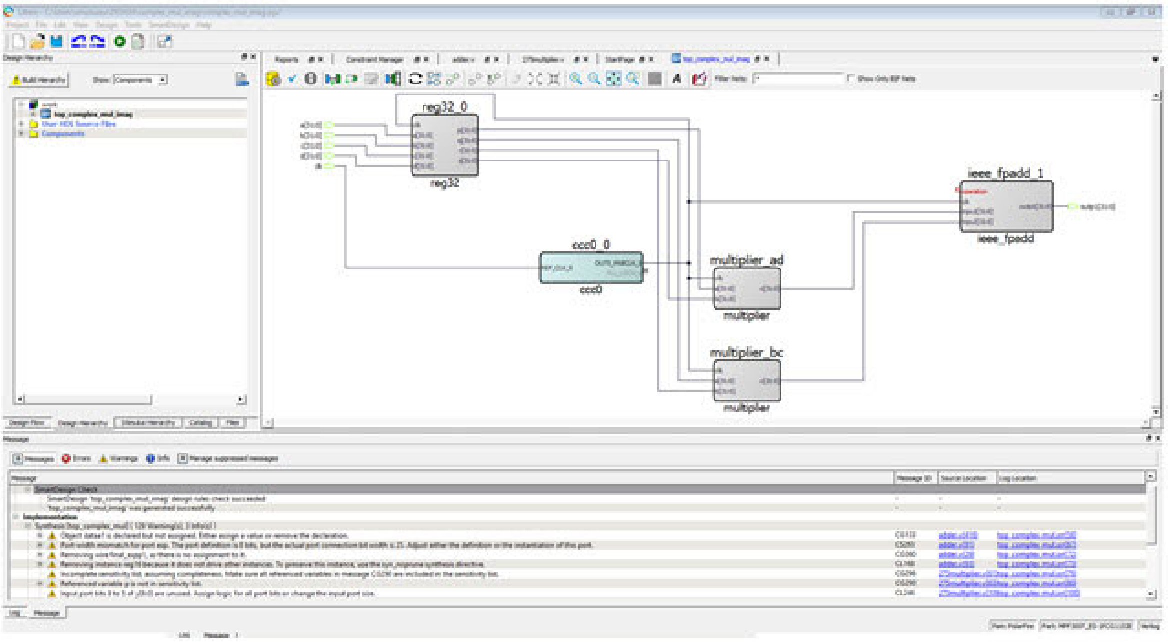
To create a SmartDesign component for calculating the real part of a complex multiplication, remove the blocks not necessary for calculating real part, and also remove the unused ports. This eliminates many of the manual connections and decreases generation time. Real designs may not be as simple as this, but this type of dividing saves time while preserving the connections between existing blocks.

See the following example figures.



Message | Message ID | Source Location | Log Location
SmartDesign Check
SmartDesign 'top_complex_mul_real' design rules check succeeded
'top_complex_mul_real' was generated successfully
Implementation
Synthesis (top_complex_mul) (128 Warning(s), 3 Info(s))
Object datas? is declared but not assigned. Either assign a value or remove the declaration.
Port width mismatch for port xop. The port definition is 8 bits, but the actual port connection bit width is 23. Adjust either the definition or the instantiation of this port.
Removing wire final_expp?, as there is no assignment to it.
Removing instance reg?4 because it does not drive other instances. To preserve this instance, use the syn_noprune synthesis directive.
Incomplete sensitivity list, assuming completeness. Make sure all referenced variables in message CG290 are included in the sensitivity list.
Referenced variable p is not in sensitivity list.
Input port bits 0 to 5 of y[8:0] are unused. Assign logic for all port bits or change the input port size.

Figure 24 • **SmartDesign Component for Real Part Calculation of Complex Multiplication**



Message | Message ID | Source Location | Log Location
SmartDesign Check
SmartDesign 'top_complex_mul_imag' design rules check succeeded
'top_complex_mul_imag' was generated successfully
Implementation
Synthesis (top_complex_mul) (128 Warning(s), 3 Info(s))
Object datas? is declared but not assigned. Either assign a value or remove the declaration.
Port width mismatch for port xop. The port definition is 8 bits, but the actual port connection bit width is 23. Adjust either the definition or the instantiation of this port.
Removing wire final_expp?, as there is no assignment to it.
Removing instance reg?4 because it does not drive other instances. To preserve this instance, use the syn_noprune synthesis directive.
Incomplete sensitivity list, assuming completeness. Make sure all referenced variables in message CG290 are included in the sensitivity list.
Referenced variable p is not in sensitivity list.
Input port bits 0 to 5 of y[8:0] are unused. Assign logic for all port bits or change the input port size.

Figure 25 • **SmartDesign Component for Imaginary Part Calculation of Complex Multiplication**

5.3 Making the Connections

There are several ways to make net connections in the SmartDesign Canvas.

5.3.1 QuickConnect

Click the QuickConnect icon to open the QuickConnect dialog box. QuickConnect is useful if you have a large design with many pins to connect and know the names of the pins you wish to connect without looking at the SmartDesign canvas. Connections are reflected in the Canvas as you make them in the dialog box. Error messages appear in the Log window immediately. It may be useful to resize the QuickConnect dialog box so that you can view the Log window or SmartDesign Canvas while you make the connections.

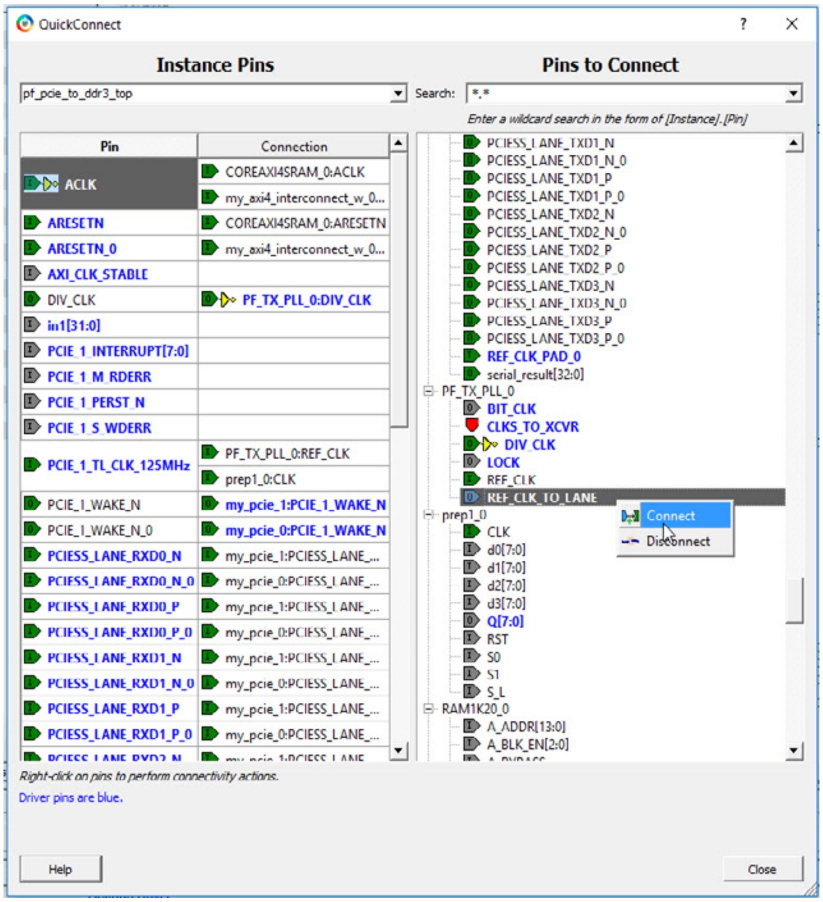
To connect pins using QuickConnect:.

1. Find the Instance Pin you want to connect and click to select it.
2. In Pins to Connect, find the pin you wish to connect, right-click and choose Connect. If necessary, use the Search field to narrow down the list of pins displayed in Pins to Connect.

Note: This dialog box enforces DRC checks. If the connection is invalid then the Connect menu item is grayed out.

Connections can also be made by selecting pins in the same column.

Figure 26 • **QuickConnect Dialog**



Instance Pins

pf_pcie_to_ddr3_top

Pin	Connection
ACLK	COREAXI4SRAM_0:ACLK my_axi4_interconnect_w_0...
ARESETN	COREAXI4SRAM_0:ARESETN
ARESETN_0	my_axi4_interconnect_w_0...
AXI_CLK_STABLE	
DIV_CLK	PF_TX_PLL_0:DIV_CLK
in1[31:0]	
PCIE_1_INTERRUPT[7:0]	
PCIE_1_M_RDERR	
PCIE_1_PERST_N	
PCIE_1_S_WDERR	
PCIE_1_TL_CLK_125MHz	PF_TX_PLL_0:REF_CLK prep1_0:CLK
PCIE_1_WAKE_N	my_pcie_1:PCIE_1_WAKE_N
PCIE_1_WAKE_N_0	my_pcie_0:PCIE_1_WAKE_N
PCIESS_LANE_RXD0_N	my_pcie_1:PCIESS_LANE_...
PCIESS_LANE_RXD0_N_0	my_pcie_0:PCIESS_LANE_...
PCIESS_LANE_RXD0_P	my_pcie_1:PCIESS_LANE_...
PCIESS_LANE_RXD0_P_0	my_pcie_0:PCIESS_LANE_...
PCIESS_LANE_RXD1_N	my_pcie_1:PCIESS_LANE_...
PCIESS_LANE_RXD1_N_0	my_pcie_0:PCIESS_LANE_...
PCIESS_LANE_RXD1_P	my_pcie_1:PCIESS_LANE_...
PCIESS_LANE_RXD1_P_0	my_pcie_0:PCIESS_LANE_...

Right-click on pins to perform connectivity actions.

Driver pins are blue.

Pins to Connect

Search: *.*

Enter a wildcard search in the form of [Instance].[Pin]

```
    PCIESS_LANE_TXD1_N
    PCIESS_LANE_TXD1_N_0
    PCIESS_LANE_TXD1_P
    PCIESS_LANE_TXD1_P_0
    PCIESS_LANE_TXD2_N
    PCIESS_LANE_TXD2_N_0
    PCIESS_LANE_TXD2_P
    PCIESS_LANE_TXD2_P_0
    PCIESS_LANE_TXD3_N
    PCIESS_LANE_TXD3_N_0
    PCIESS_LANE_TXD3_P
    PCIESS_LANE_TXD3_P_0
    REF_CLK_PAD_0
    serial_result[32:0]
⊟ PF_TX_PLL_0
    BIT_CLK
    CLKS_TO_XCVR
    DIV_CLK
    LOCK
    REF_CLK
    REF_CLK_TO_LANE
⊟ prep1_0
    CLK
    d0[7:0]
    d1[7:0]
    d2[7:0]
    d3[7:0]
    Q[7:0]
    RST
    S0
    S1
    S_L
⊟ RAM1K20_0
    A_ADDR[13:0]
    A_BLK_EN[2:0]
```

5.3.2 Connection Mode

This mode is useful if you want to quickly connect two pins with mouse clicks. Click the Connection Mode in the toolbar. Click one of the instance pins in the SmartDesign canvas. A + sign appears. Drag and drop the + sign to another pin and a green + sign appears to indicate valid connection. Release the mouse and the connection is made between the two pins.

Figure 27 • Connection Mode - Valid Pin Connection

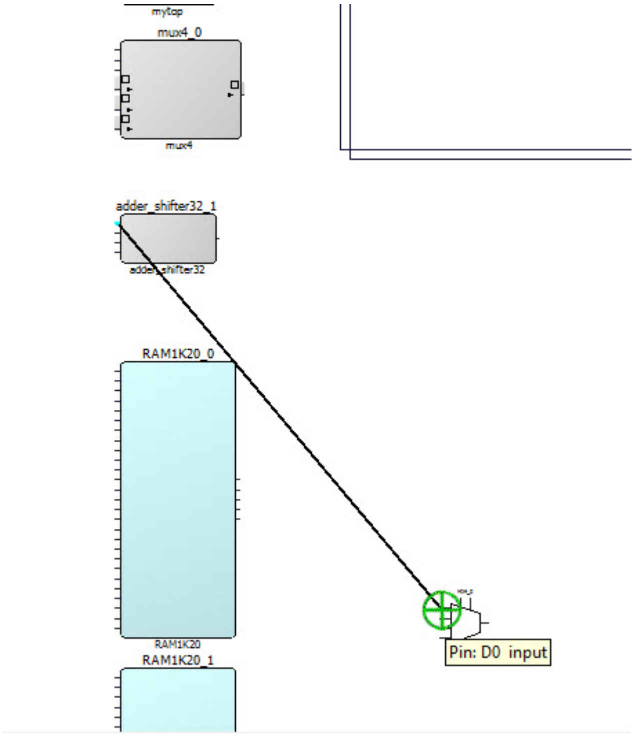
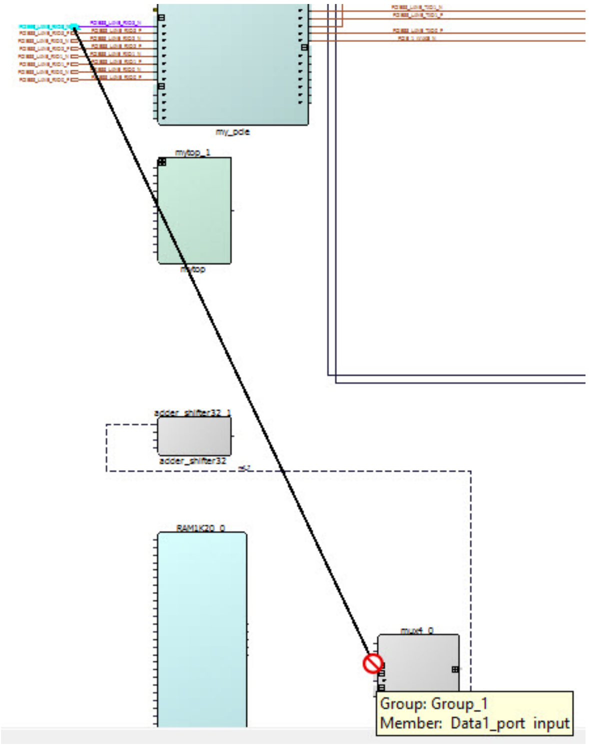


Figure 28 • Connection Mode - Invalid Pin Connection



Invalid connections are not allowed and are flagged with a red crossbar.

5.3.3 Control + Click Connection

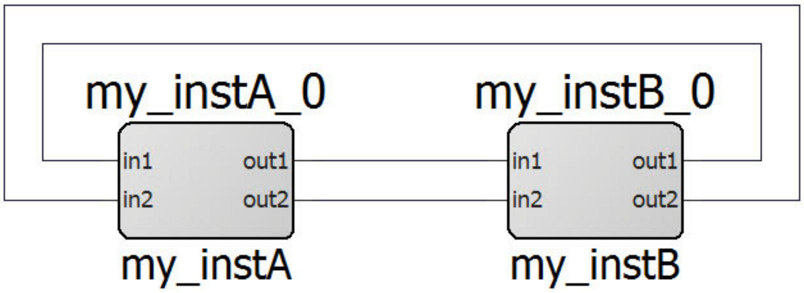
Use Control + Click to connect two pins on the canvas:

1. Left-click one pin.
2. Control and left-click another pin.
3. Right-click the second pin and select Connect.

5.3.4 Modify Pin Order Before Connections

When two instances on the Canvas need to communicate with each other such that the output pins of one instance drive the input pins of another instance and vice versa, some of the nets between the two instances may have to go around the instance to make the connections.

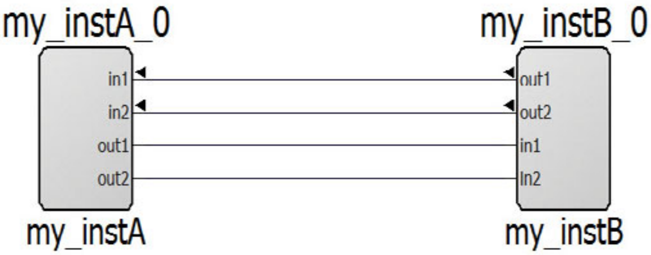
Figure 29 • **Connections between Two Instances with Regular Pin Order**



However if the pin order are modified such that the output pins of my_instance B are put on the left side and the input pins of Instance A are put on the right side of the instance, the net connection between the two instances are direct straight lines. The nets do not need to go around the instance.

Refer to “Modify Pin Order” on page 15 for details on how to modify the pin order.

Figure 30 • **Connections between Two Instances with Modified Pin Order**



After the pin order is modified, input pins are identified by inward-pointing arrow heads and output pins are identified by outward pointing arrow heads. The arrow head helps you identify at a glance the direction of the pins (input/output) when they are not in the default locations. You can always hover the mouse over the pin to open the tooltip message to see the direction.

Note: Inout pins, by default, are put on the right side of the instance. They do not have arrow heads when their location is modified. Hover the mouse over the pin to open the tooltip message to see the direction of the pin (input/output/inout).

5.3.5 Splitting the Bus Before Connections

A bus may need to be split, before connections are made to the bus, for the following reasons:

- Different bits of the bus go to different parts of the design.
- Some bits of the bus need not to be connected and have attributes on them (output pins mark unused, tie to high/low or tie to constant).

The bus pin can be split into:

- Standard Slices

These are the more common, off-the-shelf slices available in the right-click menu of a selected bus pin or port.

- Custom slices

These are various slice/width combinations that are possible for the given bus width but not available from the off-the-shelf listings in the right-click drop-down menu. A special dialog is available to customize the slices.

Refer to the “Custom Slices” on page 19.for details.

5.3.6 Search Design Objects to Connect

For a large design with many instances and pins, it may be difficult to find the pin/port/instance/net to make connections.

Two different ways are available:

- Find Window
- Net Filter in SmartDesign Canvas

5.3.6.1 Find Window

The Find Window is outside the SmartDesign canvas but it can be used to easily find design objects to make net connections.

1. Open the SmartDesign component in the canvas.
2. Click **Edit > Find** in the Libero SoC menu to open the Find window at the bottom of the Libero main window. You can also press **CTRL+F** to open the Find window.
3. Select Current Open SmartDesign in the **Search in** drop-down menu.
4. Enter the type of design objects (Instance/Net/Pin) in the **Type** drop-down menu.
5. Enter the string of characters in the **Find**
6. Click Find All.
7. Click the item in the Search Results (if there are matches).

The SmartDesign canvas will zoom in and highlight the design object. Zoom out far enough (with the design object still selected) to see the rest of the design and make the net connections.

5.3.6.2 Net Filter in SmartDesign Canvas

Use net filters in the SmartDesign Canvas to filter nets. Net filtering will filter out any nets that do not match the filter and make the SmartDesign Canvas less cluttered.

1. Open the SmartDesign component.
2. Enter a string in the **Filter Nets** field. By default, the filter field has the “* “character, which means all nets are shown (no filtering). Enter the net name or parts of the net name to filter the net you are interested in. To find all clock nets, you may want to put in *clk* in the field. Valid characters to enter in the field are:
   - “*” - a wild card that matches any number of characters
   - “?”- matches only one single character
   - [a-z, A-Z, 0-9] - matches any character inside the bracket
   - The comma “,” character or the space character is used to delimit multiple matches. A filter field with the entry “clk,reset” or “clk reset” matches any net names with the “clk” or “reset” string.

Note: For any term entered in the filter field that contains no “*” ,“?”, “[“, ”]”, the tool adds the wild-card match “*” before and after each term. “clk reset” and “*clk* *reset*” give the same matches.

Note: Net filtering is case-insensitive

To narrow down the filter, click the **Show Only BIF Nets** checkbox. Only Bus Interface (BIF) nets that match the filter are shown. BIF nets that do not match the filter and non-BIF nets are hidden.

BIF pins are identified by the icon on the instance.BIF nets are nets connected to BIF pins.

5.4 Add or Modify Top Level Ports

You can add ports to, and/or rename ports in your SmartDesign.

5.4.1 Add Prefixes to Bus Interface / Group Names on Top-level Ports:

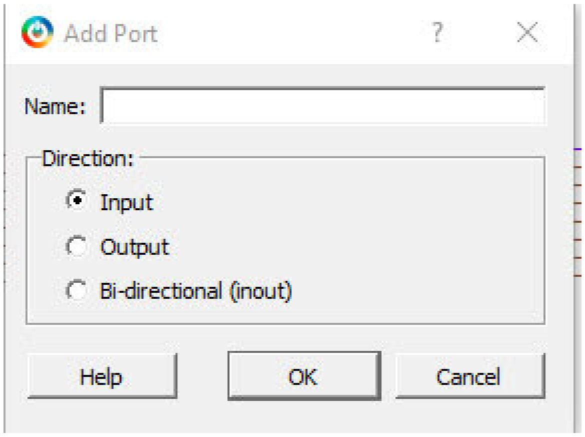
Bus Interfaces and Groups are composed of other ports. On the top level, you can add prefixes to the group or bus interface port name to the sub-port names. To do so, right-click the group or bus interface port and choose **Prefix <name> to Port Names**.

5.4.2 Adding/Removing Ports

To add ports:

1. From the SmartDesign menu, choose **Add Port.** The Add Port dialog box appears (as shown below).

Figure 31 • **Add New Port Dialog Box**



2. Specify the name of the port you wish to add. You can specify a bus port by indicating the bus width directly into the name using brackets [ ], such as mybus[3:0].
3. Select the direction of the port.

To remove a port from the top level, right-click the port and choose **Delete**.

5.4.3 Modify Port

To rename a top-level port, right-click the top-level port and choose Modify Top Level Port. You can rename the port, change the bus width (if the port is a bus), and change the port direction.

Right-click a top-level port and choose Modify Port to change the name and/or direction (if available).

5.5 Invoke DRC (Design-rule-check) on the Design

The Design Rules Check runs automatically when you generate your SmartDesign; the results appear in the Reports tab. You can also initiate a Design Rules Check by clicking on the Design Rule Check Icon of the SmartDesign Canvas Icons.

For more information on running and understanding the results of the DRC, see “Appendix C - DRC Check” on page 50.

5.6 Generate the Top-Level Component

The SmartDesign that has been created must now be generated. Generating a SmartDesign component may fail if there are any DRC errors. DRC errors must be corrected before you generate your SmartDesign top-level component.

If the ports of a sub-design have changed, then the parent SmartDesign component will be annotated with the icon ⊗ in the Design Hierarchy tab of the Design Explorer. This issue must also be corrected before you generate your SmartDesign top-level component

Once there are no further DRC errors, the top-level component can be generated either recursively or non-recursively. Non-Recursiive generation is enabled by default.

5.6.1 Recursive Generation

In the "recursive generation" mode, the Generate button will attempt to generate all sub-design SmartDesigns, depth first. The parent SmartDesign will only be generated if all the sub-designs are generated successfully. To enable Recursive Generation, from the Project Menu, choose **< Project -> Preferences -> Design Flow -> Generate Recursively**.

5.6.2 Non-Recursive Generation

In the "non-recursive generation" mode, the Generate button will only attempt to generate the specified SmartDesign. This generation can be marked as successful even if a sub-design is un-generated (either never attempted or unsuccessfully attempted). An ungenerated component will be annotated with an 1 icon in the Design Hierarchy tab of the Design Explorer.

6 Design Navigation Features

The SmartDesign Canvas provides the following navigational features:

- Expand/Fold Instance
- Magnified Window

6.1 Expand/Fold Instance

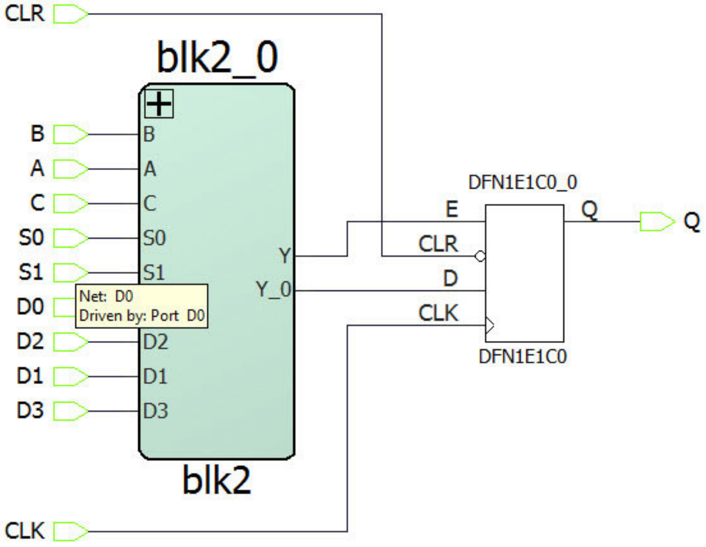
The Expand/Fold instance feature allow you to traverse up (Fold Instance) and down (Expand Instance) the design hierarchy of a SmartDesign component.

An instance of a SmartDesign component which contains lower level hierarchies can be expanded or folded in place. Expanded means traversing one level of hierarchy for viewing. Fold meas the design is collapsed to the next higher level. The expand/fold instance is a read-only view of the component.

The Expand and Fold actions are executed in place. The result of the Expand/Fold Instance is displayed in the same location, relative to the rest of the design in the same window. No new window is opened.

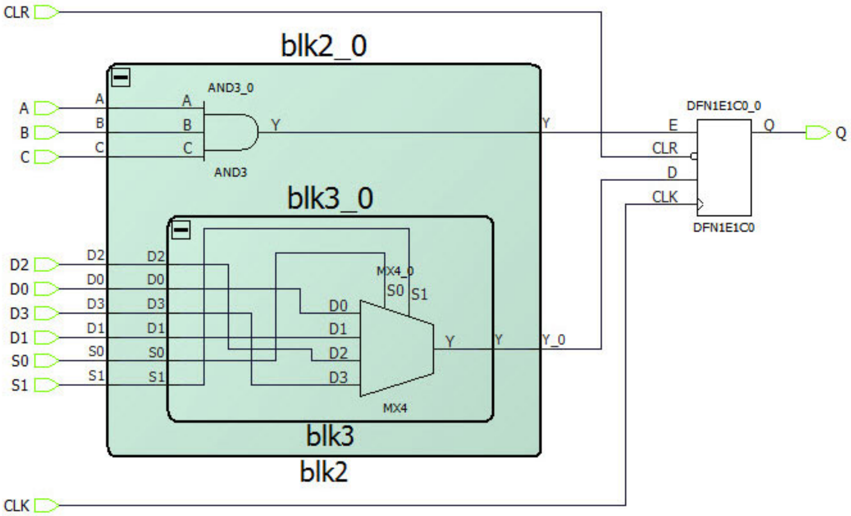
A folded design is indicated by a + sign at the upper left corner of the design. Clicking the + sign expands the folded design to the next lower levels of hierarchy. Alternatively, right-click the instance and select **Expand Inplace**.

Figure 32 • **Top-Level Design - Folded**



An expanded design is indicated by a - sign at the upper left corner of the design. Clicking the - sign collapses the design hierarchy to the next higher level of hierarchy. Alternatively, right-click the instance and select **Fold Instance**.

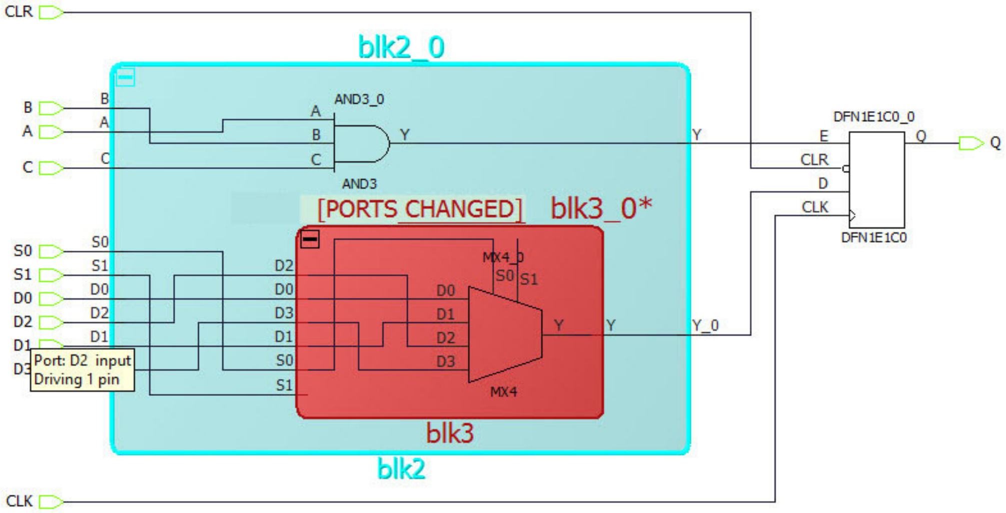
Figure 33 • **Top-Level Design - Expanded**



When a lower level of the design hierarchy is modified, the parent block (which instantiates the modified lower level) is indicated by the asterisk "*" next to the name in the SmartDesign Component Name tab to indicate a change. If the modified block is two levels of hierarchy down, the design blocks of the next two higher levels have the asterisk (*) next to their names (in the SmartDesign tab) to indicate that the design component has changed. The modified block is highlighted in red and clearly identified as [PORTS CHANGED].

Note: The red highlight and the [PORTS CHANGED] identification is visible only after the higher level block(s) is folded and expanded again to expose the changed lower level block.

Figure 34 • **Red Highlight and Changed_Module Identification**



Note: Instead of [PORTS CHANGED], the error becomes [MISSING MODULE] if any one of the following conditions occurs:

- A lower level SmartDesign component instantiated in a higher level module/block has been deleted from the Project (**Design Flow window > Right-click SmartDesign Component > Delete from Project**)
- An IP core instantiated in a higher level module/blocks is not found or has been removed from the vault.

If the changed or missing module/block is at the lowest level of hierarchy, the **Update Component** and **Generate Components** steps must be repeated at every level of hierarchy above the changed module/block until the top-level hierarchy is reached, updated and regenerated.

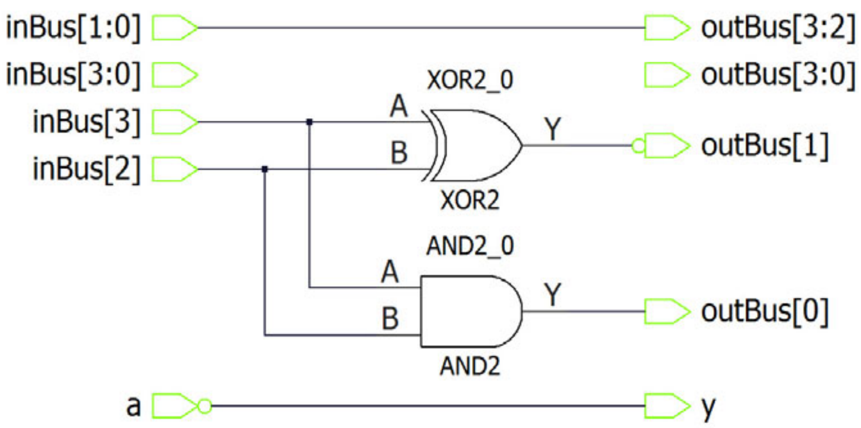
6.1.1 Expand Inplace and Low Level Blocks

When a design component is expanded inplace, SmartDesign reads in the current state of the low level blocks after it expands. If changes are made to the low level blocks (e.g. name changes, connectivity changes, addition or deletion of instances, pin order changes), the Expand Inplace view must be folded and expanded again before the Expanded Inplace view shows the update.

The top-level ports in the low level blocks are shown differently in the Expanded Inplace view if these top-level ports are:

- Inverted
- Sliced

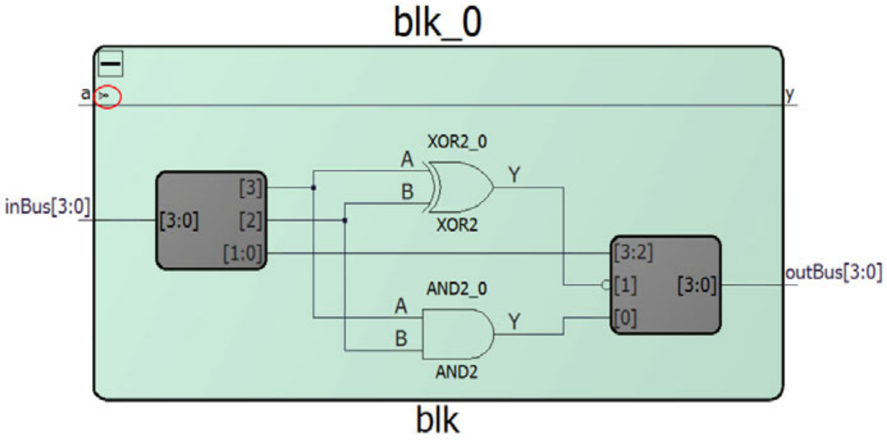
Figure 35 • **Low Level Block View with Inverted Port and Sliced Port**



At the Expanded Inplace view, for inverted ports, an inverter symbol is displayed next to the port (circled in Figure 36).

At the Expanded Inplace view, for the slice ports, a slicer instance is inserted for each sliced bus (Figure 36).

Figure 36 • **Expanded Inplace View with Inverter Symbol and Slicer Ports**



6.1.2 Component Regeneration

When the port list of the lower level block has changed, you need to:

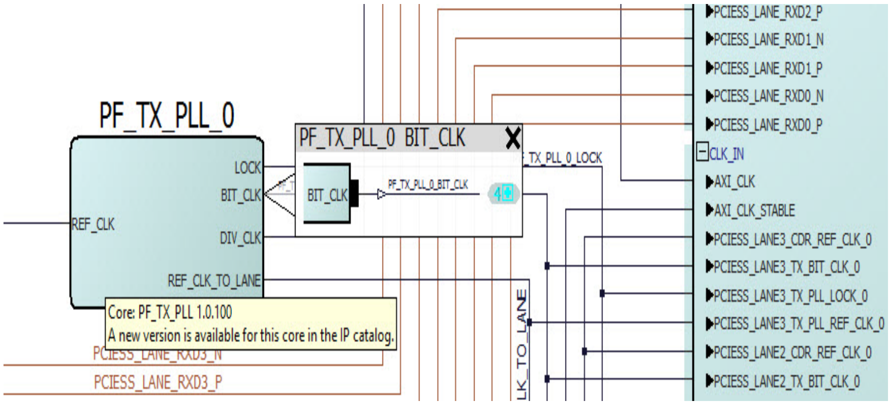
1. Go to next higher level of the design hierarchy,
2. Right-click the lower level component and select **Update Component**
3. Generate the higher level component.

The red highlight and the [PORTS CHANGED] text disappear when the component update is successful.

6.2 Magnify Pin

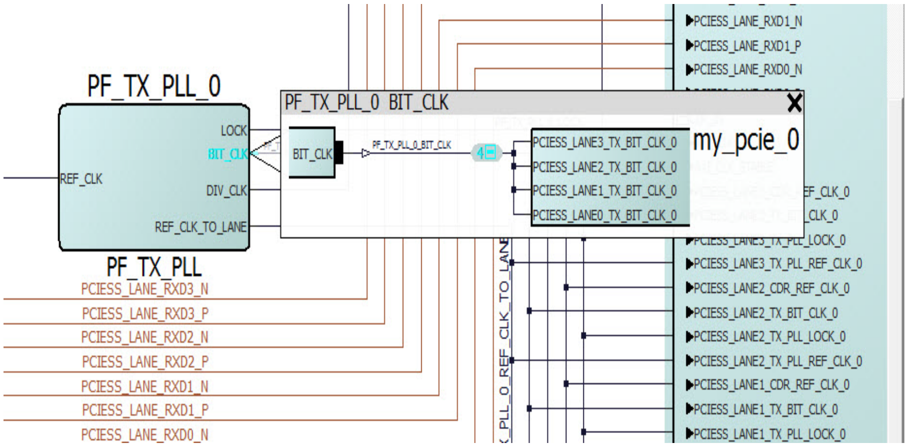
This option is available to pins and ports only. Double-click a pin or a port or right-click and select Magnify Pin to access this option. A Magnify Pin Window opens and shows the specified pin/port's connections.

Figure 37 • **Magnify Pin Window**



If the pin has a fanout of more than 1, the number "4" in Figure 37 beside the + sign shows the total fanout. Click the + sign to see all the fanouts of the pin (Figure 38).

Figure 38 • **Magnify Pin Window with Fanout Expanded**


PF_TX_PLL_0
LOCK
BIT_CLK
REF_CLK
DIV_CLK
REF_CLK_TO_LANE
PF_TX_PLL
PF_TX_PLL_0 BIT_CLK
BIT_CLK
PF_TX_PLL_0_BIT_CLK
4
PCIESS_LANE3_TX_BIT_CLK_0
PCIESS_LANE2_TX_BIT_CLK_0
PCIESS_LANE1_TX_BIT_CLK_0
PCIESS_LANE0_TX_BIT_CLK_0
my_pcie_0
PCIESS_LANE_RXD3_N
PCIESS_LANE_RXD3_P
PCIESS_LANE_RXD2_N
PCIESS_LANE_RXD2_P
PCIESS_LANE_RXD1_N
PCIESS_LANE_RXD1_P
PCIESS_LANE_RXD0_N
PLL_0_REF_CLK_TO_LANE
PCIESS_LANE_RXD1_N
PCIESS_LANE_RXD1_P
PCIESS_LANE_RXD0_N
PCIESS_LANE3_TX_PLL_LOCK_0
PCIESS_LANE3_TX_PLL_REF_CLK_0
PCIESS_LANE2_CDR_REF_CLK_0
PCIESS_LANE2_TX_BIT_CLK_0
PCIESS_LANE2_TX_PLL_LOCK_0
PCIESS_LANE2_TX_PLL_REF_CLK_0
PCIESS_LANE1_CDR_REF_CLK_0
PCIESS_LANE1_TX_BIT_CLK_0
PCIESS_LANE1_TX_PLL_LOCK_0

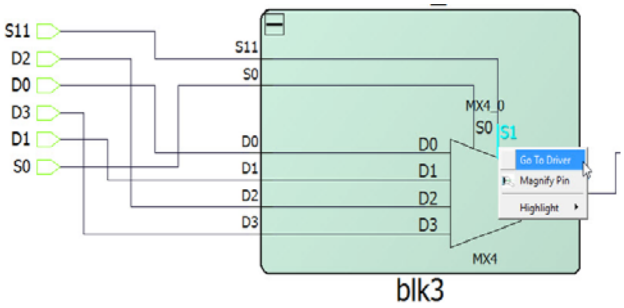
Double-clicking the net/pin/port/instance inside the Magnify Window zooms into and selects the item. This makes it easy to navigate inside large designs.

To close the Magnify Window, click the Delete (X) button on the Magnify Window or press ESC on the keyboard.

6.3 Go To Driver

Right-click a net/pin/port and select **Go To Driver**. SmartDesign zooms in and selects the driver of the net/pin/port at the current level of hierarchy.

Figure 39 • **Go To Driver**



7 Appendix A - FAQ

The collection of SmartDesign Frequently Asked Questions are useful for anyone new to SmartDesign.

The questions are grouped as follows:

7.1 General Questions

1. What is SmartDesign?
2. How do I create my first SmartDesign?
3. What are the differences between the Old and New SmartDesign Canvas?
4. What do I need to do with my existing SmartDesign component when I switch to the new SmartDesign?

7.2 Instantiating into your SmartDesign

1. Instantiating Into Your SmartDesign
2. How do I create my first SmartDesign?
3. I have a block that I wrote in VHDL (or Verilog), can I use that in my SmartDesign?
4. My HDL module has Verilog parameters or VHDL generics declared, how can I configure those in SmartDesign?

7.3 Working in SmartDesign

1. How do I make manual connections?
2. How do I make manual connections?
3. How do I connect a pin to the top level?
4. Oops, I just made a connection mistake. How do I disconnect two pins?
5. I need to apply some simple 'glue' logic between my cores. How do I do that?
6. My logic is a bit more complex than inversion and tie offs - what else can I do? How do I create a new top-level port for my design?
7. How do I rename one of my instances?
8. How do I rename my group pins? I need to reconfigure one of my Cores, can I just double-click the instance?
9. I want more Canvas space to work with!

7.4 Working With Processor-Based Designs in SmartDesign

1. But I need my peripheral at a specific address or slot.
2. But I need my peripheral at a specific address or slot.
3. How do I view the Memory Map of my design?
4. I have my own HDL block that I want to connect as a peripheral on the AMBA bus. How can I do that?
5. How do I start writing my application code for my design?
6. How do I start writing my application code for my design?
7. How do I simulate my processor design?

7.5 VHDL Construct Support in SmartDesign

1. What VHDL constructs do you support?
2. How can I import files with VHDL Special Types into SmartDesign?
3. What is the purpose of the mapping file?
4. Where will the mapping file meta.out be generated?
5. What are the VHDL special types that are not generated automatically?
6. What do I do if I am using VHDL types that are not generated automatically?
7. What is the meta.out file format?

7.6 Making The Design Look Nice

1. Can the tool automatically place my instances on the Canvas to make it look nice?
2. My design has a lot of connections, and the nets are making my design hard to read. What do I do?

7.7 Generating The Design

General Questions

What is SmartDesign?

SmartDesign is a design entry tool. It's the first tool in the industry that can be used for designing System on a Chip designs, custom FPGA designs or a mixture of both types in the same design. A SmartDesign can be the entire FPGA design, part of a larger SmartDesign, or a user created IP that can be stored and reused multiple times. It's a simple, intuitive tool with powerful features that enables you to work at the abstraction level at which you are most comfortable.

It can connect blocks together from a variety of sources, verify your design for errors, manage your memory map, and generate all the necessary files to allow you to simulate, synthesize, and compile your design.

How do I create my first SmartDesign?

In the Libero SoC Project Manager Design Flow window, under Create Design, double-click **Create SmartDesign**.

What are the differences between the Old and New SmartDesign Canvas?

There are no major differences between the old and new SmartDesign canvas. The layout and features are similar.

What do I need to do with my existing SmartDesign component when I switch to the new SmartDesign?

If you want to import SmartDesign components generated in the old canvas into a new SmartDesign canvas, just import the .cxf file and the associated HDL files and IP cores and configure them. However, if you want to open a project in the new version, you must update project data.

Instantiating Into Your SmartDesign

Where is the list of Cores that I can instantiate into my SmartDesign

The list of available cores is displayed in the Catalog. This catalog contains all DirectCore IP, Design Block cores, and macros.

Drag and drop the core from the Catalog onto your SmartDesign Canvas. The Core Configurator opens. Configure the core and an instance of your Core appears on the Canvas.

I have a block that I wrote in VHDL (or Verilog), can I use that in my SmartDesign?

Yes! Import your HDL file into the Project Manager (**File > Import Files**). After you do this, your HDL module will appear in the Project Manager Hierarchy. Then, drag-and-drop it from the Design Hierarchy onto your SmartDesign Canvas.

My HDL module has Verilog parameters or VHDL generics declared, how can I configure those in SmartDesign?

If your HDL module contains configurable parameters, you must create a 'core' from your HDL before using it in SmartDesign. Once your HDL module is in the Project Manager Design Hierarchy, right-click it and choose **Create Core from HDL**. You will then be allowed to add bus interfaces to your module if necessary. Once this is complete, you can drag your new HDL+ into the SmartDesign Canvas and configure your parameters by double-clicking it.

Working in SmartDesign

How do I make manual connections?

Enter Connection Mode and click and drag from one pin to another. Click the **Connection Mode** button in the Canvas to enter Connection Mode.

Alternatively:

1.Select the pins you want connected by using the mouse and the CTRL key.

2.Right-click one of the selected pins and choose Connect.

How do I connect a pin to the top level?

Right-click the pin and choose Promote to Top Level. You can even do this for multiple pins at a time, just select all the pins you want to promote, right-click one of the pins and choose Promote to Top Level. All your selected pins will be promoted to the top level.

Oops, I just made a connection mistake. How do I disconnect two pins?

Use CTRL+Z to undo your last action. If you want to undo your 'undo', hit redo (CTRL+Y).

To disconnect pins you can:

1. Right-click the pin you want to disconnect and choose Disconnect
2. Select the net and hit the delete key

I need to apply some simple 'glue' logic between my cores. How do I do that?

For basic inversion of pins, you can right-click a pin and choose Invert. An inverter will be placed at this pin when the design is generated. You can also right-click a pin and choose **Tie Low** or **Tie High** if you want to connect the pin to either GND or VCC.

To tie an input bus to a constant, right-click the bus and choose **Tie to Constant**. To mark an output pin as unused, right-click the pin and choose **Mark as Unused**.

To clear these, just right-click on the pin again and choose **Clear Attribute**.

My logic is a bit more complex than inversion and tie offs - what else can I do?

You have full access to the library macros, including AND, OR, and XOR logic functions. These are located in the Catalog, listed under Macro Library. Drag the logic function you want onto your SmartDesign Canvas.

How do I create a new top-level port for my design?

Click the **Add Port button** in the Canvas toolbar

How do I rename one of my instances?

Right-click the instance on the Canvas and a dialog opens for you to rename the instance. The instance name is located directly above the instance on the Canvas.

How do I rename my top-level port?

Right-click the port you want to rename and choose Modify/Rename.

How do I rename my group pins?

Right-click the group pin you want to rename and choose Rename.

I need to reconfigure one of my Cores, can I just double-click the instance?

Yes.

I want more Canvas space to work with!

Maximize your workspace (**CTRL-W**), and your Canvas will maximize within the Project Manager. Press **CTRL-W** again if you need to see your Hierarchy or Catalog.

Working with Processor-Based/AMBA-Bus Designs

But I need my peripheral at a specific address or slot.

Right-click the Canvas and choose **Modify Memory Map** to invoke the **Modify Memory Map** dialog that enables you to set a peripheral to a specific address on the bus.

The bus core will show the slot numbers on the bus interface pins. These slot numbers correspond to a memory address on the bus.

Verify that your peripheral is mapped to the right bus address by viewing the Memory Map.

How do I view the Memory Map of my design?

Generate your project and open datasheet in the **Report View**.

The memory map section will also show the memory details of each peripheral, including any memory mapped registers.

I have my own HDL block that I want to connect as a peripheral on the AMBA bus. How can I do that?

SmartDesign supports automatic creation of data driven configurators based on HDL generics/parameters.

If your block has all the necessary signals to interface with the AMBA bus protocol (ex: address, data, control signals):

1.Right-click your custom HDL block and choose **Create Core from HDL**. The Libero SoC creates your core and asks if you want to add bus interfaces.

2.Click **Yes** to open the Edit Core Definition dialog box and add bus interfaces. Add the bus interfaces as necessary.

3.Click **OK** to continue.

Now your instance has a proper AMBA bus interface on it. You can manually connect it to the bus or let Auto Connect find a compatible connection.

How do I start writing my application code for my design?

Libero SoC simplifies the embedded development process by automatically creating the workspace and project files for the Software IDE that you specify in the Tools profile.

Once you have generated your design, the firmware and workspace files will automatically be created. Click Write Application Code in the Design Flow tab and the Software IDE tool will open your design's workspace files.

How do I simulate my processor design?

SmartDesign automatically generates the necessary Bus Functional Model (BFM) scripts required to simulate your processor based design. A top-level testbench for your SmartDesign is generated automatically as well.

Create your processor design, generate it, and you will be able to simulate it in ModelSim.

VHDL Construct Support in SmartDesign

What VHDL constructs do you support?

VHDL types Record, Array, Array of Arrays, Integer and Unsigned are supported on entity ports of imported VHDL files - these are treated as special types in Libero SoC.

How can I import files with VHDL Special Types into SmartDesign?

To work with a VHDL file with Special Types you must:

1.Drag and drop the entity into SmartDesign and connect it just as you would with any other SmartDesign instance.

2.Generate the Mapping File (meta.out):

- Navigate to the Design Hierarchy view, under the current SmartDesign.
- Right-click every VHDL file or every top hierarchical file and choose **Create Mapping File** (VHDL).

3.Generate the SmartDesign

4. Continue with the Libero SoC Design Flow steps (Synthesis, Simulation, etc.)

If you do not generate the Mapping File, and try to Generate your SmartDesign, you will see the following error in the log window:

Error: Select the HDL file in the Design Hierarchy and right-click the HDL file and choose Create Mapping File(VHDL) because at least one entity port is of type Array or Record.

The above is reported only if the entity port is of type Record, Array, Array of Array, or Unsigned.

What is the purpose of the mapping file?

The mapping file contains the mapping information between the SmartDesign ports and original user-specified data types of ports in design files, and is used for type casting of signals during design generation.

Where will the mapping file meta.out be generated?

The file is generated in your $project_dir/hdl folder. This file will be used during SmartDesign generation.

What are the VHDL special types that are not generated automatically?

The following types are not automatically generated from the right-click menu option Create Mapping File(VHDL):

- Array of array is not supported
- Array of record is not supported
- Enum in range of array is not supported.
- Constants are not supported.
- Buffer output ports are not supported

What do I do if I am using VHDL types that are not generated automatically?

You must manually write the mapping information in the meta.out file for unsupported types (types which are not generated automatically) in the prescribed format.

- Integer
- Unsigned
- Array and Array of Arrays
- Record

What is the meta.out file format?

The meta.out File Format:

MetaFile: MetaLibraryItem | MetaPackageList | MetaEntityList

MetaLibraryItem: library <lib_name>

MetaPackageList: MetaPackageItem MetaPackageList

MetaPackageItem: package <package_name> MetaItemDeclarationList end

MetaItemDeclarationList: MetaItem MetaItemDeclarationList

MetaItem: (MetaRecordItem | MetaArrayOfArrayItem | MetaIntegerType | MetaArrayItem)

MetaIntegerItem: (MetaIntegerType | MetaIntegerWithoutType)

MetaIntegerType: integer <integer_name> NumericRange

MetaIntegerWithoutType: integer NumericRange

MetaUnsignedItem: unsigned <name> NumericRange

MetaArrayOfArrayItem: array_of_array < MetaArrayOfArrayName> Range [MetaArrayItem] end

MetaRecordItem: record <record_name> RecordItemList end

RecordItemList: RecordItem RecordItemList

RecordItem: <Inst_name> (MetaArrayOfArrayName | MetaIntegerItem | MetaUnsignedItem | MetaSimpleArray)

MetaEnumuratedItem: enum <enum_name> (Item_name{,Item_name})

Range: [NumericRange | MetaEnumuratedItem]

NumericRange: lsd: msd

MetaArrayItem:array <array_name> [<record_name>] end

MetaEntityList: entity <entity_name> MetaEntityItemList end

MetaEntityItemList: MetaEntityItem MetaEntityItemList

MetaEntityItem: (RecordEntityItemList | IntegerEntityItemList | ArrayEntityItemList | ArrayOfArrayEntityItemList | UnsignedEntityItemList | BufferPortItemList)

RecordEntityItemList: RecordEntityItem RecordEntityItemList

RecordEntityItem: (RecordNormalItem | RecordArrayOfArrayItemList)

```
RecordNormalItem: <user_port_name>. RecordItem <record_name>
RecordArrayOfArrayItemList: <record_port_name>[index]. RecordItem <record_name>
BufferPortItemList: BufferPortItem BufferPortItemList
BufferPortItem: buffer <buffer_name>
IntegerEntityItemList: IntegerEntityItem IntegerEntityItemList
IntegerEntityItem: <user_port_name> (MetaIntegerType | MetaIntegerWithoutType)
ArrayEntityItemList: ArrayEntityItem ArrayEntityItemList
ArrayEntityItem: <user_port_name> MetaArrayItem
ArrayOfArrayEntityItemList: ArrayOfArrayEntityItem ArrayOfArrayEntityItemList
ArrayOfArrayEntityItem: <port_name> < MetaArrayOfArrayName>
UnsignedEntityItemList: UnsignedEntityItem UnsignedEntityItemList
UnsignedEntityItem: <user_port_name> MetaUnsignedItem
```

Meta.out file Example:

```
package record_pkg
array array1
end
array_of_array array2 [3:0]
end
record test
test_integer integer [0:127]
test_array array1
test_array_of_array array2
test_unsigned unsigned [2:0]
end
end
entity MUX
mux_in1.test_std_logic test
mux_in1.test_std_logic_vector test
mux_in1.test_integer test
mux_in1.test_array test
mux_in1.test_array_of_array[0] test
mux_in1.test_array_of_array[1] test
mux_in1.test_array_of_array[2] test
mux_in1.test_array_of_array[3] test
mux_in1.test_unsigned test
mux_in2.test_std_logic test
mux_in2.test_std_logic_vector test
mux_in2.test_integer test
mux_in2.test_array test
mux_in2.test_array_of_array[0] test
mux_in2.test_array_of_array[1] test
mux_in2.test_array_of_array[2] test
mux_in2.test_array_of_array[3] test
mux_in2.test_unsigned test
mux_out.test_std_logic test
mux_out.test_std_logic_vector test
mux_out.test_integer test
mux_out.test_array test
mux_out.test_array_of_array[0] test
mux_out.test_array_of_array[1] test
mux_out.test_array_of_array[2] test
mux_out.test_array_of_array[3] test
mux_out.test_unsigned test
mux_array array1
end
```

Making the Design Look Nice

Can the tool automatically place my instances on the Canvas to make it look nice?

Yes. Right-click the Canvas white space and choose **Auto Arrange Instances**.

My design has a lot of connections, and the nets are making my design hard to read. What do I do?

You can disable the display of the nets in the toolbar (**Hide Nets**). This automatically hides all the nets in your design.

You can still see how pins are connected by selecting a net stub, the net will automatically be visible again as a RAST net. The tooltip also shows the drivers of the pin.

You can also show only BIF Nets. Check the box at the far right of the SmartDesign Canvas window.

My instance has too many pins on it; how can I minimize that?

Try grouping functional or unused pins together. For example, on the CoreInterrupt there are 8 FIQ-Source* and 32 IRQSource* pins, group these together since they are similar in functionality.

To group pins: Select all the pins you want to group, then right-click one of the pins and choose Add pins to group.

If a pin is in a group, you are still able to use it and form connections with it. Expand the group to gain access to the pin. All Member pins of the group are shown.

Pins in a group can be hidden and collapsed into the group. If a pin (in a group) is connected to a net, then the pin cannot be hidden and collapsed into the group. Pins (in a group) can be hidden and collapsed if the pins are tie-off/marked unused/disconnected.

Oops, I missed one pin that needs to be part of that group? How do I add a pin after I already have the group?

Select the group, right-click the pin you want to add and select Add Pin to Group.

I have a pin that I don't want inside the group, how do I remove it?

Right-click the pin and choose Ungroup Pin or select the pin and press the Delete key on the Keyboard.

How can I better see my design on the Canvas?

There are zoom icons in the Canvas toolbar. Use them to **Zoom in, Zoom out, Zoom to fit, and Zoom selection**. You can also maximize your workspace with CTRL-W.

Generating the Design

Ok, I'm done connecting my design, how do I 'finish' it so that I can proceed to synthesis?

In the SmartDesign Canvas toolbar, click the **Generate Component** icon.

I get a message saying it's unable to generate my SmartDesign due to errors, what do I do? What is the Design Rules Check?

The Design Rules Check is included in your Report View. It lists all the errors and warnings in your design, including unconnected input pins, required pin connections, configuration incompatibilities between cores, etc.

Errors are shown with a small red stop sign and must be corrected before you can generate; warnings may be ignored.

For details See “Appendix C - DRC Check” on page 50.

What does this error mean? How do I fix it?

Review the Errors in Appendix C - DRC Check for how to correct DRC violations.

8 Appendix B - Glossary

Table 6 • **SmartDesign Glossary**

Term	Description
BIF	Abbreviation for bus interface. Logical grouping of ports or pins that represent a single functional purpose. May contain both input and output, scalars or buses. A bus interface is a specific mapping of a bus definition onto a component instance.
Bus	An array of scalar ports or pins, where all scalars have a common base name and have unique indexes in the bus
Bus Definition	Defines the signals that comprise a bus interface. Includes which signals are present on a master, slave, or system interface, signal direction, width, default value, etc. A bus definition is not specific to a logic or design component but is a type or protocol
Bus Interface Net	A connection between 2 or more compatible bus interfaces
Canvas	A visual representation of the breadboard for placing components and stitching the components to create a working design.
Driver	A driver is the origin of a signal on a net. The input and slave BIF ports of the top-level or the output and Master BIF ports from instances are drivers.
Instance	A block like item with pins on either side of it. These are connected together to create designs. You may have multiple instances of a single component in your design. For each specific instance, you usually will have custom connections that differ from other instances of the same component.
Net	A wire that connects pins/ports in a design
PAD	The property of a port that must be connected to a design's top-level port. PAD ports will eventually be assigned to a package pin. In SmartDesign, these ports are automatically promoted to the top-level and cannot be modified.
Pin	Pins are the inputs/outputs/inouts of an instance that a net can be attached to for connection with other components in the design. By default, pins are placed on either the left (inputs) or the right side (outputs and inouts) of the instance. Pin order can be modified for a cleaner, less cluttered connection.
Port	It is like a pin, but it is not attached to an instance. It acts as a way of letting a net connect to the outside world. A port has a direction (input, output, bi-directional) and may be referred to as a 'scalar port' to indicate that only a single unit-level signal is involved. In contrast, a bus interface on an instance may be considered as a non-scalar, composite port.
Macro	A type of very basic instance that typically has a special well-known shape associated with it. Inside of more complicated instances will be connected macros to do a more complicated function.Macros are specific to the technology family. Macros are listed in the Basic Macro group in the Catalog.
HDL File	A specially formatted text-file that describes that design created by the user in a standard way

Table 6 • **SmartDesign Glossary**

Term	Description
Viewport	The rectangular view area of the canvas that is visible to the user. The user can move the viewport around on the canvas or zoom in/out to view their design. Showing the whole canvas would be too large in most cases.
Master BIF	Master Bus Interface. The bus interface that initiates a transaction (such as a read or write request) on a bus.
Slave BIF	Slave Bus Interface. The bus interface that responds to a transaction (such as a read or write request) on a bus.
System Bus Interface	Interface that is neither master nor slave; enables specialized connections to a bus.
Slice	A slice is created from a bus. It is a portion of the bus and it contains some but not all scalar members of the bus.
Top-Level Port	An external interface connection to the outside world. Scalar if a 1-bit port, bus if a multiple-bit port or a BIF (Bus Interface). These are connected to the pad/package pins of the FPGA device.
Pin Group	A grouping of pins (scalar or bus) the user creates for easy connection or identification.

9 Appendix C - DRC Check

When SmartDesign components are generated, the tool automatically enforces Design Rule Checks. The component is generated if the check passes.

Alternatively, to invoke DRC checks without generating the component, click the DRC check icon ✓ in the toolbar.

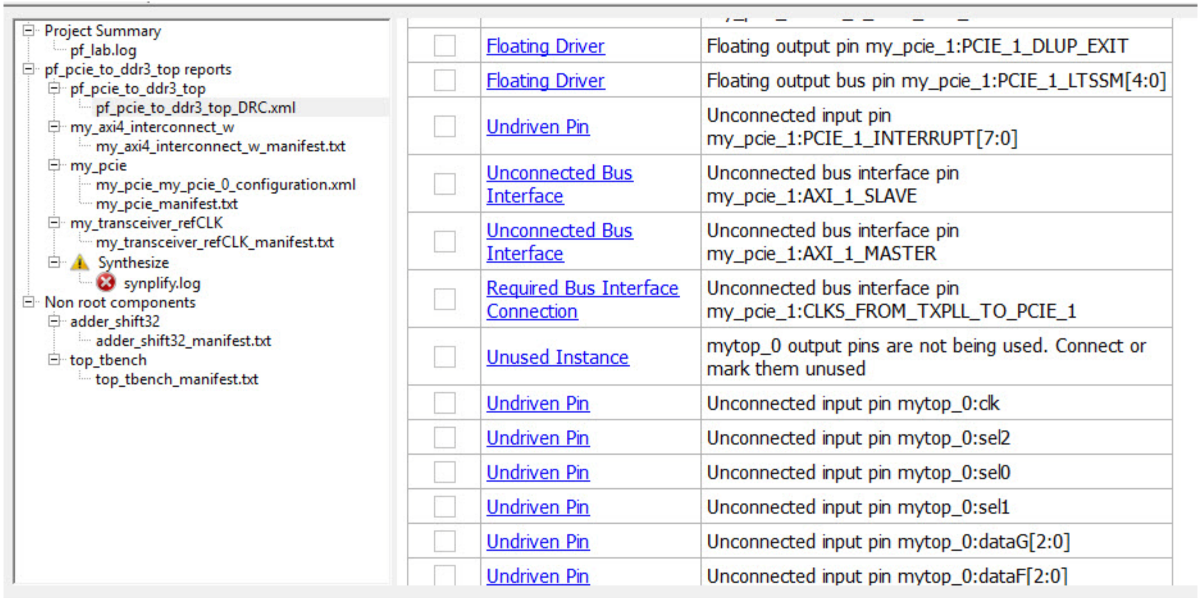
To view the results of the DRC check, from the **Design** menu, choose **Reports.** The Report is named <Design_component>_DRC_xml in the Reports tab.

DRC errors are also reported in the Log/Message Window.

Click the links on the DRC report. SmartDesign zooms in and highlights the design object that is the subject of the DRC violation.

You must correct all DRC errors before you can generate the SmartDesign component and continue with the design flow.

Figure 40 • **DRC Report**



- Project Summary
  - pf_lab.log
- pf_pcie_to_ddr3_top reports
  - pf_pcie_to_ddr3_top
    - pf_pcie_to_ddr3_top_DRC.xml
  - my_axi4_interconnect_w
    - my_axi4_interconnect_w_manifest.txt
  - my_pcie
    - my_pcie_my_pcie_0_configuration.xml
    - my_pcie_manifest.txt
  - my_transceiver_refCLK
    - my_transceiver_refCLK_manifest.txt
  - ⚠ Synthesize
    - ☒ synplify.log
- Non root components
  - adder_shift32
    - adder_shift32_manifest.txt
  - top_tbench
    - top_tbench_manifest.txt

☐	Floating Driver	Floating output pin my_pcie_1:PCIE_1_DLUP_EXIT
☐	Floating Driver	Floating output bus pin my_pcie_1:PCIE_1_LTSSM[4:0]
☐	Undriven Pin	Unconnected input pin my_pcie_1:PCIE_1_INTERRUPT[7:0]
☐	Unconnected Bus Interface	Unconnected bus interface pin my_pcie_1:AXI_1_SLAVE
☐	Unconnected Bus Interface	Unconnected bus interface pin my_pcie_1:AXI_1_MASTER
☐	Required Bus Interface Connection	Unconnected bus interface pin my_pcie_1:CLKS_FROM_TXPLL_TO_PCIE_1
☐	Unused Instance	mytop_0 output pins are not being used. Connect or mark them unused
☐	Undriven Pin	Unconnected input pin mytop_0:clk
☐	Undriven Pin	Unconnected input pin mytop_0:sel2
☐	Undriven Pin	Unconnected input pin mytop_0:sel0
☐	Undriven Pin	Unconnected input pin mytop_0:sel1
☐	Undriven Pin	Unconnected input pin mytop_0:dataG[2:0]
☐	Undriven Pin	Unconnected input pin mytop_0:dataF[2:0]

9.1 Message Types and Corrective Actions

The following are the more common DRC errors and the corrective actions you need to take:

Unused Instance - You must remove this instance or connect at least one output pin to the rest of the design.

Out-of-date Instance - You must update the instance to reflect a change in the component referenced by this instance.

Undriven Pin - To correct the error you must connect the pin to a driver or change the state, i.e. tie low (GND), tie high (VCC), tie to Constant or marked unused.

Floating Driver - You can mark the pin unused if it is not going to be used in the current design. Pins marked unused are ignored by the Design Rules Check.

Unconnected Bus Interface - You must connect this bus interface to a compatible port because it is required connection.

Required Bus Interface Connection - You must connect this bus interface before you can generate the design. These are typically silicon connection rules.

Exceeded Allowable Instances for Core - Some IP cores can only be instantiated a certain number of times for legal design. For example, there can only be one CortexM1 or CoreMP7 in a design because of silicon limitations. You must remove the extra instances. This check is technology-dependent.

Incompatible Family Configuration - The instance is not configured to work with this project's Family setting. Either it is not supported by this family or you need to re-instantiate the core. This DRC check is family/technology-dependent.

No RTL License, No Obfuscated License, No Evaluation License - You do not have the proper license to generate this core. Contact Microsemi SoC to obtain the necessary license.

No Top level Ports - There are no ports on the top level. To auto-connect top-level ports, right-click the Canvas and choose Auto-connect

Self-Instantiation. A component cannot instantiate itself-This is reported only in the Log/Message Window.