SmartDesign User Guide

Libero SoC v2025.1

Introduction (Ask a Question)

The SmartDesign tool is a visual block-based design creation and entry tool for instantiating, configuring, and connecting Microchip IPs, user-generated IPs, and custom and glue-logic HDL modules. The tool provides a canvas for stitching together the various design components.

The resulting HDL from the SmartDesign tool is a Design-Rule-Checked (DRC) and 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 SmartDesign canvas instantiates the following design objects:

- Microchip IP cores
- User-generated or third-party IP cores
- HDL modules
- HDL parameterized core modules
- Basic macros
- Other SmartDesign components
- Reusable design blocks published from the Libero® SoC Design Suite

The SmartDesign tool provides the following features for effortless design creation and visualization:

- Addition of synthesis attributes to design objects
- Visualization of the memory map of the design
- Smart Search and Connect tool for fast design look-up and connectivity for complex designs
- Smart Search and Filter to create focused cones of the design on the canvas

Table of Contents

1. Understanding the SmartDesign Canvas (Ask a Question)

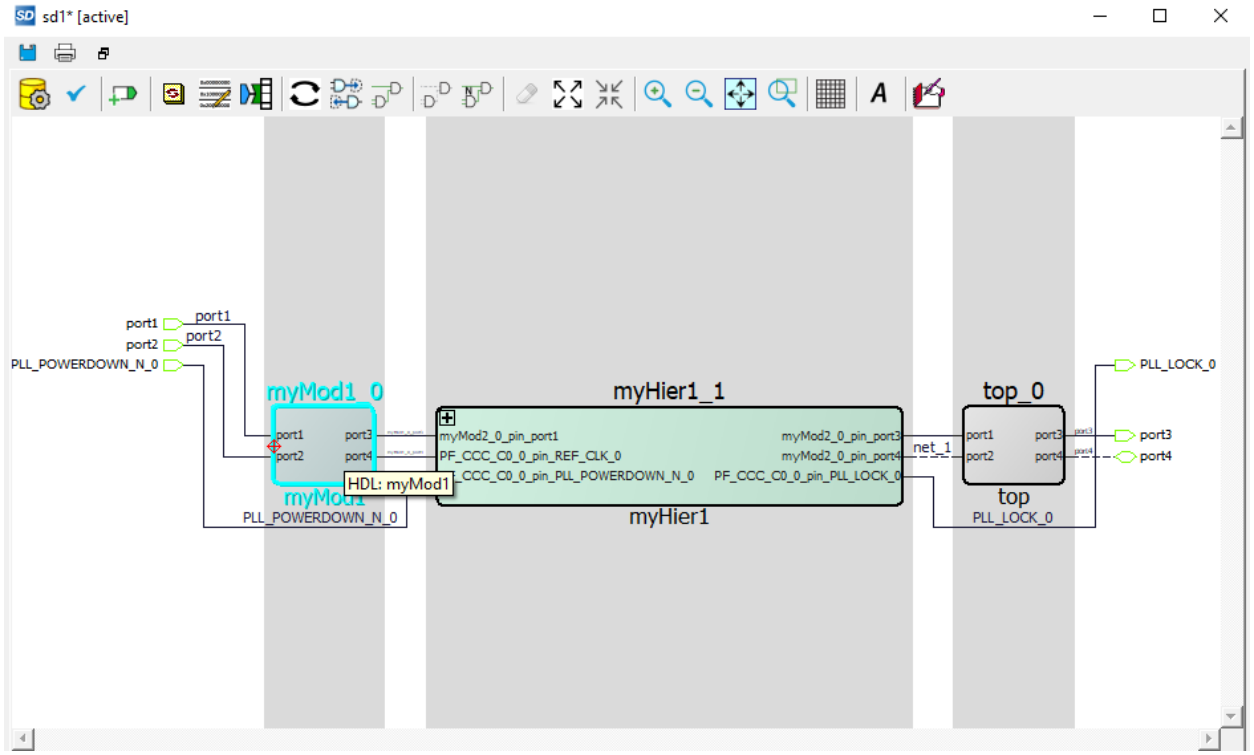
The SmartDesign canvas places all components in columns, with the nets vertically routed in the space between columns. Input ports are placed in the left-most column. Output ports and inout ports are placed in the right-most column. However, you can choose to move the ports or components to any location of your choice in the SmartDesign canvas.

When components are instantiated in the SmartDesign tool, they are either placed in an existing column or a new column is created for them. When you move the components vertically, you can see the column boundaries. You can also move the components horizontally to a new or an existing column.

The following figure shows the SmartDesign tool's canvas.

Tip: Click on specific area within the SmartDesign canvas image to know more about it.

Figure 1-1. SmartDesign Canvas at a Glimpse



1.1. Click and Drag Operations in the Canvas (Ask a Question)

The following table lists the mouse click and drag operations supported in the SmartDesign canvas.

Table 1-1. Click and Drag Operations in SmartDesign Canvas

Operation	Action	Description
Zoom in	Click and drag the mouse pointer towards the top-left in the canvas.	The distance your cursor is dragged determines the magnitude of the zoom-in and is indicated by a positive integer in red.

Table 1-1. Click and Drag Operations in SmartDesign Canvas (continued)

Operation	Action	Description
Zoom out	Click and drag the mouse point towards the top-right in the canvas	The distance your cursor is dragged determines the magnitude of the zoom-out and is indicated by a negative integer in red.
Zoom to fit	Click and drag the mouse pointer towards bottom-left in the canvas.	Changes the display to tightly fit the design in the canvas.
Select	Click and drag toward bottom-right in the canvas to select multiple instances.	Ensure that instances, pins, and ports you want select are fully contained inside the selection rectangle.

To move the viewpoint/perspective of specific component(s) of your design to a specific area in the canvas, hold the **Ctrl** key down, drag and drop the components of the design to the required area in the canvas, and release the **Ctrl** key. Alternatively, you can perform this action by holding the mouse scroll wheel down and drag and drop the design to the required area in the canvas.

1.2. Additional Canvas Operations (Ask a Question)

The following table lists additional SmartDesign toolbar canvas operations. Hover your cursor over the icon to view the operation name in a tool tip. You can perform any of these operations by right-clicking in the empty space inside the canvas.

Table 1-2. Additional Toolbar Options

Operation	Action	Description
Reset Layout	Resets the layout view using the built-in optimization algorithms	You can retain the presentation model (instance, port coordinates, and highlight colors) while saving your SmartDesign. However, if you choose to clear the current presentation model, click on **Reset Layout**. This action removes all the presentation information (position, size, highlights, and modified pin orders) and recreates canvas layout using the built-in optimization algorithms.
Auto-arrange Layout	Auto arranges the components in the layout using the built-in optimization algorithms	Only the location (X-Y coordinates) of the instances and ports are changed. All presentation information remains intact.
Compress Layout	Compresses display of the complete design using the built-in optimization algorithms	Click to push the instances and ports towards each other to remove the 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.
Unhighlight All	Removes all highlights in the design	Highlighting of all the design objects (nets, pins, ports, and instances) on the canvas are removed. This action is active only if design objects are already highlighted.
Zoom In	Zooms in to the canvas	Zoom in on the canvas.
Zoom Out	Zooms out of the canvas	Zoom out on the canvas.
Zoom to Fit	Optimizes the design to the fit into the canvas	Adjust the zoom and viewpoint, so that everything on the canvas fits within the visible view port with no extra empty space around the design.
Zoom to Selection	Zooms in to the selected area in the canvas	Click this action and drag the mouse to draw a rectangle which is when released, causes a zoom in, so that the visible view port area is approximately the size of the drawn rectangle.
Show Grid	Makes the gridlines visible	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 might not show if the canvas is zoomed too far out.
Add Note	Allows you to add notes in your design	Click to enter the **Add Note** mode. During the next mouse click, the canvas opens a dialog box for entering the text and font size of the text (anchored at the mouse click location).

Table 1-2. Additional Toolbar Options (continued)

Operation	Action	Description
Save to PDF	Saves the design to a `.PDF` file.	This action opens a dialog box that allows you to save a picture of all or part of the design to a `.PDF` document.

2. Creating a Synthesizable SmartDesign (Ask a Question)

In the design process, the first step is to create a SmartDesign component after creating a project. Create your SmartDesign using the following procedures.

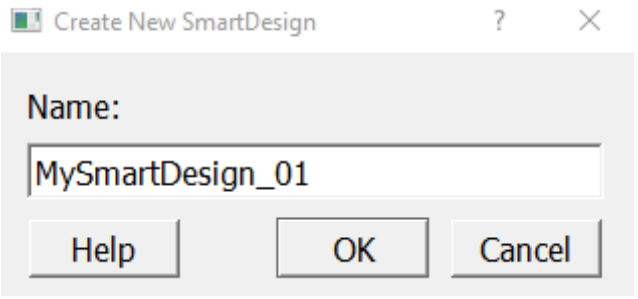
2.1. Creating a SmartDesign Component (Ask a Question)

To create a SmartDesign component using the SmartDesign canvas, perform the following steps:

1. Launch the SmartDesign tool in either of the following ways.
   - On the menu, click **File** > **New** > **SmartDesign**
   - On the **Design Flow** tab, double-click **Create SmartDesign**

   Result: The **Create New SmartDesign** dialog box appears.

   Figure 2-1. Create New SmartDesign Dialog Box



2. Enter an appropriate and unique name for your SmartDesign component in the **Name** box.
3. Click **OK** to create the SmartDesign component. The SmartDesign component appears in the **Design Hierarchy** tab. The newly created SmartDesign appears as a central tab in the IDE main window.

Result: The design file is saved in a folder with the same name as that of the design itself and is placed in your `<project>\component\work\` folder.

2.2. Importing an Existing HDL Source File(s) or Folder(s) (Ask a Question)

To import an HDL design source file(s) or folder(s) use the following procedures.

1. Import your design sources in any of the following ways:
   - On the menu, click **File** > **Import** and select **HDL Source Files**. The **Import Files** dialog box appears.
   - On the menu, click **File** > **Import**, and select **HDL Source Folder**. The **Import Folders** dialog box appears.
   - In the **Design Flow** tab, right-click on **Create HDL** option, and select **Import Files**. The **Import Files** dialog box appears.
2. Navigate to the location where your design source file(s) or folder(s) are located, select the file(s) or folder(s), and click **Open**.

> **Important:** When you use the **Design Flow** import option, the selection option changes from **Open** to **Choose**.

Result: The selected file(s) or folder(s) are copied to your `<project>\hdl\` folder.

For more information, see Libero Design Flow User Guide.

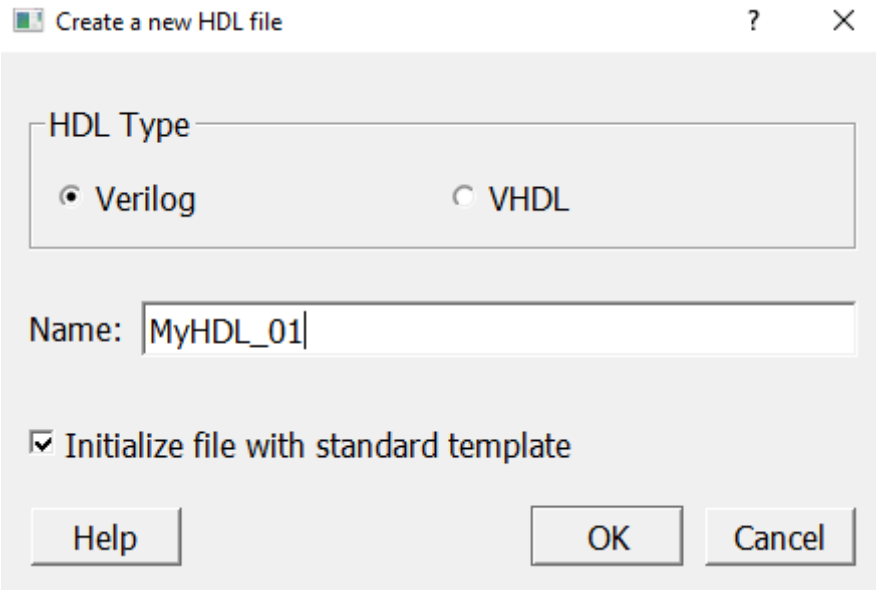
2.3. Creating a New HDL File Using the IDE (Ask a Question)

To create a new HDL file using the IDE, perform the following steps:

1. Launch the IDE in either of the following ways:
   - On the menu, click **File** > **New** > **HDL**
   - On the **Design Flow** tab, double-click **Create HDL**

   Result: The **Create a new HDL file** dialog box appears.

Figure 2-2. Create new HDL file Dialog Box



2. By default, the preferred language to write the design file is set to **Verilog**. If you choose to use VHDL, select the **VHDL** option.
3. Enter an appropriate and unique name for your design file in the **Name** box.
4. By default, the design file is initialized with the built-in standard design file template. If you choose to not use the default template, deselect the **Initialize file with standard template** option.
5. Click **OK** to create the HDL file. The newly created HDL file appears in the **Design Hierarchy** tab and is displayed in the central tab of the IDE main window.

Result: The newly created HDL file is copied to your `<project>\hdl\` folder.

3. Instantiating User HDL Modules, IP Cores, and Components in SmartDesign (Ask a Question)

You can add one or more HDL modules, building blocks, and components from the **Design Hierarchy** tab to your design. The components can be IP cores from the **IP Catalog** tab, basic macros, design blocks, and other SmartDesign components available from the **Components** node in the **Design Hierarchy** tree or from the **Components** tab.

3.1. Instantiating a User HDL Module in your Design (Ask a Question)

To instantiate a user HDL source module, use either of the following procedures:

- On the **Design Hierarchy** tab, select the user HDL module of interest, drag and drop it onto the canvas.
- On the **Design Hierarchy** tab, right-click the user HDL module of interest and select **Instantiate in <active_SD_name>** from the right-click menu.

Result: The selected module is instantiated in the SmartDesign canvas.

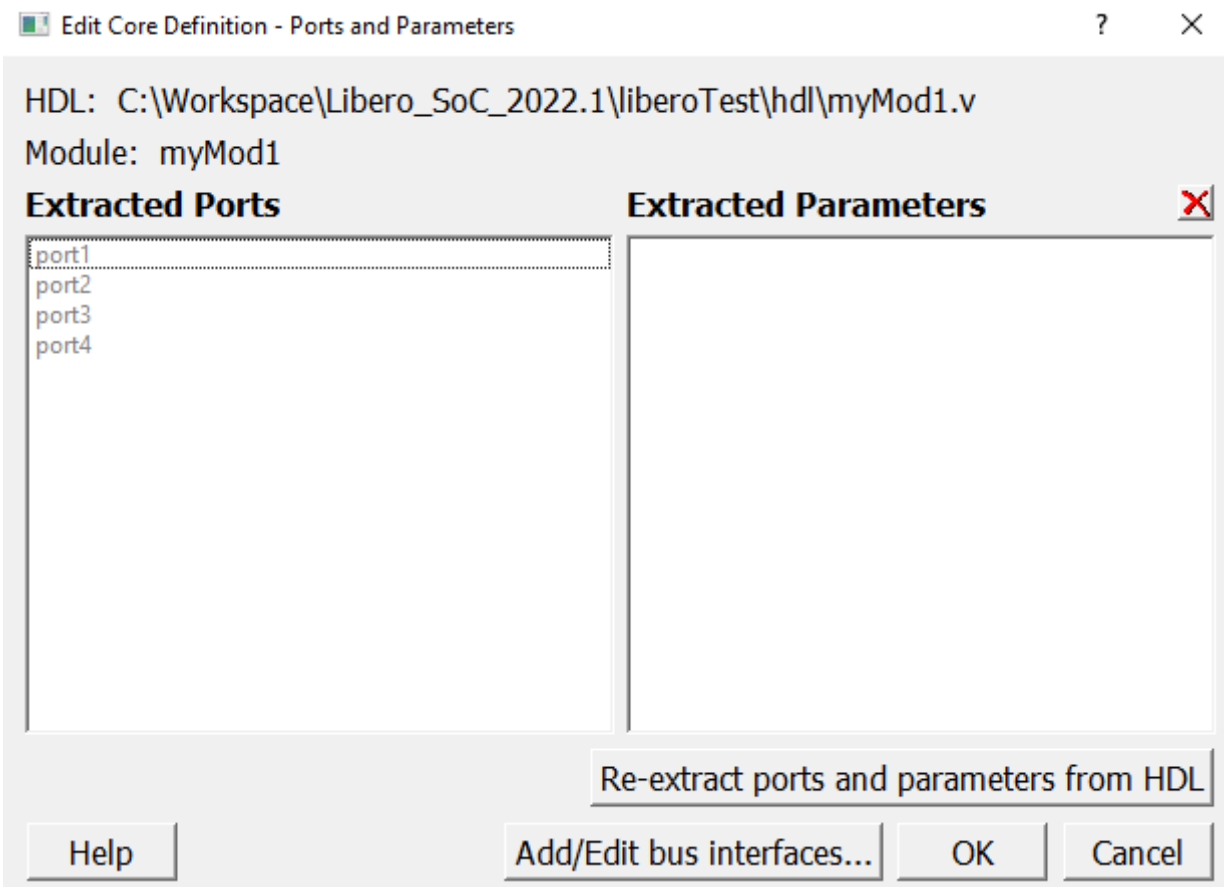
Tip: Repeat these steps to instantiate all the user HDL source modules of interest in your SmartDesign canvas.

3.2. Creating a HDL+ Core from an User HDL Source Module (Ask a Question)

To create a HDL+ core from a user HDL source module, perform the following steps:

1. On the **Design Hierarchy** tab, right-click the user HDL file and select **Create Core from HDL** from the right-click menu. The **Edit Core Definitions – Ports and Parameters** dialog box appears.

Figure 3-1. Edit Core Definitions – Ports and Parameters



2. Specify the ports and parameters as required and click **OK**. A BIF might also be added to the HDL core. It shows up in the SmartDesign with a BIF pin that can easily connect to the compatible instance BIF pins.

Result: The selected user HDL module is converted to an HDL+ core and can be instantiated in your SmartDesign canvas.

3.3. Configuring and Instantiating Instances in your Design (Ask a Question)

You can choose to configure and instantiate the following components in your design:

- SmartDesign component
- IP Core components
- User HDL cores
- Design blocks

Important: Only IP core components can be configured.

3.3.1. Configuring and Instantiating an IP Core Component (Ask a Question)

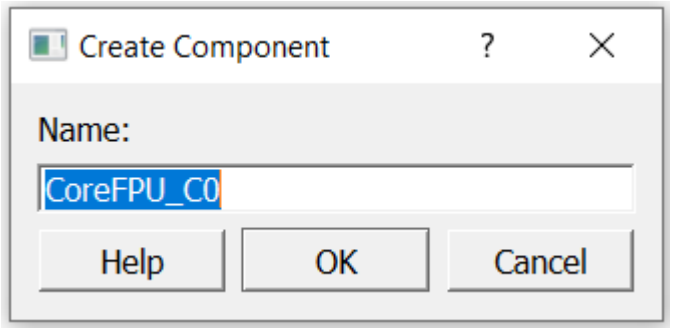
Microchip offers a large portfolio of configurable cores. These cores are available from the **Catalog** tab. To instantiate an IP core component in your design, you must first configure the component.

1. To configure and instantiate an IP core component, use either of the following procedures:

- On the **Catalog** tab, select the IP core component of interest, drag and drop it onto the canvas.
- On the **Catalog** tab, right-click the IP core component of interest and select **Instantiate in <active_SD_name>** from the right-click menu.
- On the **Catalog** tab, right-click the IP core component of interest and select **Configure core** from the right-click menu.

Result: The **Create Component** dialog box appears.

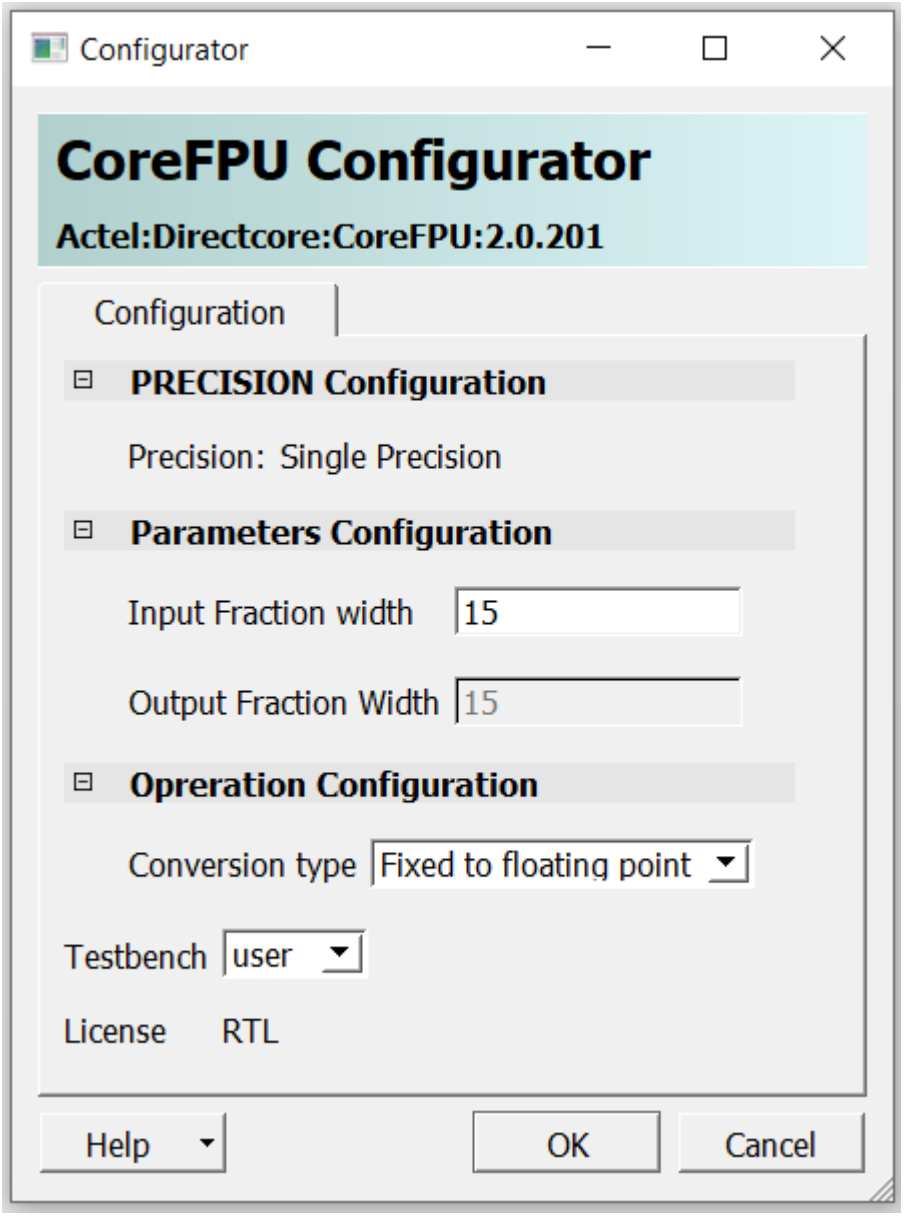
Figure 3-2. Create Component Dialog Box



By default, the selected IP core's name is displayed in the **Name** box. You can use the next step to change the core's name to a name of your choice.

2. Enter an appropriate and unique name for your core component in the **Name** box and click **OK**. The **Configurator** window appears.

Figure 3-3. Configurator Window



3. Configure the core component as per your design requirements and click **OK**.
4. (Optional) To re-configure an existing component, right-click the component and choose **Configure**. The **Configurator** window appears. Proceed with step 3 to modify the existing configuration.

Result: The configured IP core component is made available in the **Design Hierarchy** tab, **Components** tab, and is instantiated in your SmartDesign canvas.

Tip: Repeat these steps to instantiate the necessary IP core components of interest in your SmartDesign.

Important: If you observe that the core component that you are looking for appears in italicized-gray color in the **Catalog** tab, it means that the core component is yet to be downloaded from the Microchip IP Core Repository to your hard disk (vault).

Download the core component of interest from the **Catalog** tab, in either of the following ways.

- Double-click the core component of interest.
- Right-click the core component of interest and click **Download**.

Result: The core component of interest is downloaded from the Microchip vault to your local vault and is made available for configuration and instantiation.

For information on the types of available core components and its color representation, see Appendix D – Component Types.

3.3.2. Instantiating SmartDesign Components, User HDL Cores, and Design Blocks in your Design (Ask a Question)

To instantiate a SmartDesign component, a user HDL core, or a design block use either of the following procedures:

- On the **Design Hierarchy** tab, select the SmartDesign component, user HDL core, or the design block of interest, drag, and drop it onto the canvas.
- On the **Design Hierarchy** tab, right-click the SmartDesign component, user HDL core, or the design block of interest, and select **Instantiate in <active_SD_name>** from the right-click menu.

Result: The selected components are instantiated in your SmartDesign.

Tip:

- Repeat these steps to include the necessary SmartDesign components, user HDL cores, or design blocks of interest in your design.
- A component is either created within the existing project or imported into the existing project. Once in the project, the method for instantiating a component in your SmartDesign is the same regardless of the component type.

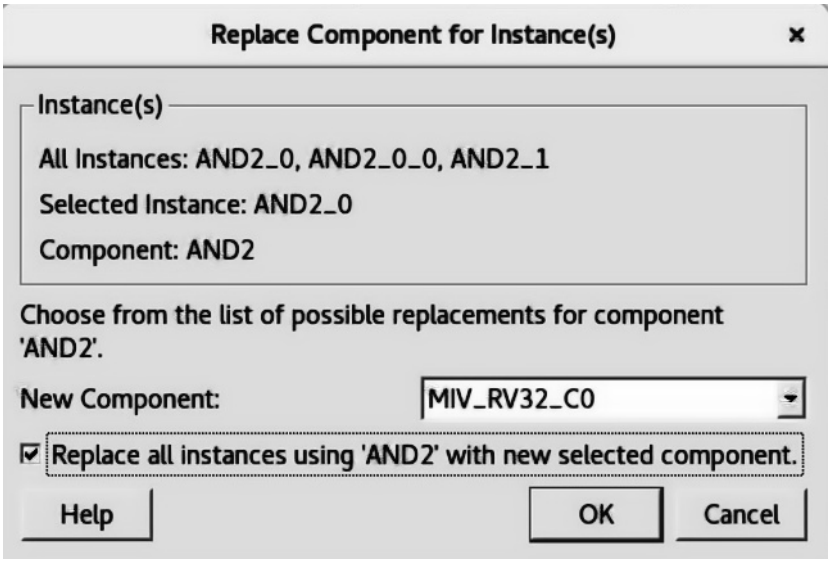
3.4. Replacing an Instance Component with a Different Component (Ask a Question)

You can choose to replace an instance(s) component with a different component or a different implementation while preserving the current instance's interface (port) connections.

To replace an instance component with a different component, perform the following steps:

1. Right-click the instance component and select **Replace Component** from the right-click menu. The **Replace Component for Instance(s)** dialog box appears.

Figure 3-4. Replace Component for Instance(s) Dialog Box



The instance(s) section shows the following information:

- **All Instances**: Lists all instances of the component.
- **Selected Instance**: Shows the selected instance that will be replaced with the new component.
- **Component**: The component of the selected instance.

2. From the **New Component** dropdown list, select the replacement component of your choice.
3. By default, the **Replace all instances using <component_name> with new selected component** option is selected. When this option is selected, every instance of the selected component in the current design (the list of instances shown under the **All Instances** group) are replaced with the new component specified in step 2.

Important:

- As a result of the replace action, if the instance port list changes, then some pins connections might be dropped. The information related to the dropped pins connection in printed in the **Log** window.
- If the **Replace all instances using <component_name> with new selected component** option is deselected, only the selected instance is replaced in the design.
- If you select an instance and click **OK** for a non-valid component, the dialog box closes automatically, and an error message is printed in the **Log** window.

3.5. Updating a Component Core Version (Ask a Question)

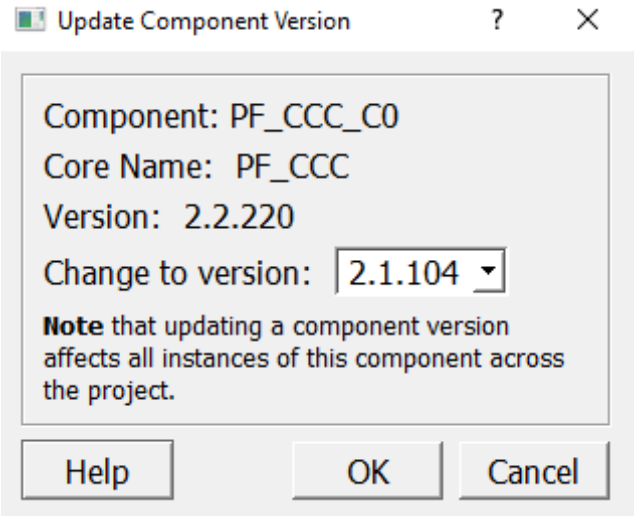
The **Update Component Version** functionality enables you to update a component of an instance with another version. You can restore or update your component without creating a new instance or losing your connections.

1. You can update a component version in your design in either of the following ways:
   - On the **Design Hierarchy** tab, right-click the component and select **Update Component Version** from the right-click menu.

- On the SmartDesign canvas, right-click the component and select **Update Component Version** from the right-click menu.

Result: The **Update Component Version** dialog box appears.

Figure 3-5. Update Component Version Dialog Box



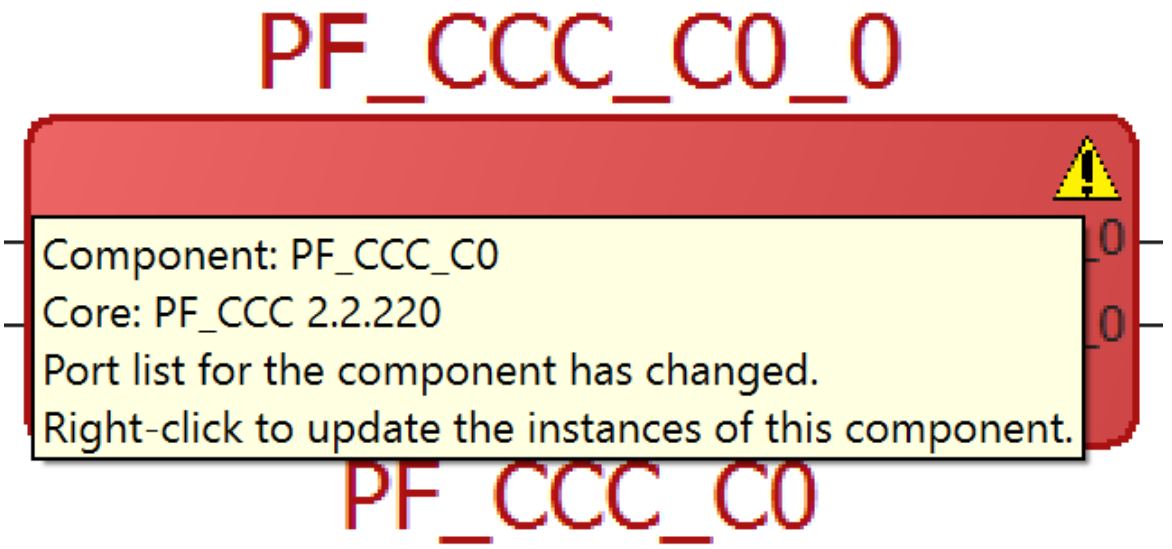
2. The **Change to version** dropdown list contains all the versions of the selected core available in the vault. In the **Change to version** dropdown box, select the version you want to upgrade to and click **OK**.
   Result: The core is automatically downloaded to the vault if it does not already exist in the vault. If the core exists in the vault, the component in the SmartDesign canvas is upgraded with the selected version.

Important: You cannot upgrade to a different version of the core that is not in your vault and if the vault is read-only.

3.6. Updating the Component Instance Port List (Ask a Question)

When the lower-level component port list is modified, the higher-level component is highlighted in red color and an exclamation symbol in shown within a red colored triangle at the top right corner of the component as shown in the following figure. This means that the instance port list needs to be updated.

Figure 3-6. Higher-level Component Highlighted in Red Color with Tooltip for User Action



Perform the following steps in the recommended order when the port list of the lower-level component is modified:

1. Go to the next higher-level of the design hierarchy containing lower-port modifications.
2. Right-click the lower-level component and select **Update Instance** from the right-click menu.
3. Generate the higher-level component in which the instance is updated.

Result: The red colored triangle at the top right corner of the component disappears if the component update is successful.

3.7. Performing Additional Operations on Components (Ask a Question)

When one or more instances are selected, you can use the right-click menu to perform the following operations.
Note: Some operations are applicable only on a single instance.

Table 3-1. Performing Additional Operations on Components

Operation	Action
Configure	If the selected instance is a configured core component or parametrized HDL core, the **Configure** option opens the **Configurator** dialog box for the core to be configured. If the instance is a SmartDesign component, the **Configure** option brings up the SmartDesign canvas for edits. If the selected instance is an HDL module, the **Configure** option brings up the HDL code editor to edit the instance module. The configure operation is equivalent to double-clicking the instance. The **Configure** option is available only when a single instance is selected. For more information, see Instantiating SmartDesign Components, User HDL Cores, and Design Blocks in your Design
Modify HDL	The **Modify HDL** option brings up the HDL code editor to edit the HDL source file of the instance module. This option is available only when an instance of an HDL core is selected. **Note:** The **Modify HDL** option is only available for HDL modules. For an HDL module, double-click the HDL component in the SmartDesign canvas or right-click the HDL component and choose **Configure** to open the HDL file in the HDL code editor.

Table 3-1. Performing Additional Operations on Components (continued)

Operation	Action
Replace Component	The **Replace Component** option allows you to replace an instance component with a different component or a different implementation while preserving the current instance's interface (port) connections. For more information, see Replacing an Instance Component with a Different Component
Update Component Version	The **Update Component Version** option allows you to update a component of an instance with another version. For more information, see Updating a Component Core Version
Rename	The **Rename** option prompts you to modify the selected instance name. Provide a new unique name for the instance and then click **OK.** **Note:** An error message is shown in the **Log** window if the instance name does not follow the HDL naming rules.
Delete	The **Delete** option deletes the selected item. When multiple items are selected, all of them are deleted. **Note:** Not all design objects can be deleted.
Create Hierarchical SmartDesign and **Flatten Hierarchical SmartDesign**	You can choose to select one or more instances in your SmartDesign and create a new SmartDesign out of it (all the nets and connections from the instance are retained). or you can choose to flatten a hierarchical SmartDesign component. For more information, see Working with Hierarchical SmartDesigns
Remove Connections	The **Remove Connections** option disconnects all pins that can be disconnected from nets. Pins that cannot be disconnected (for example, pins connected to pads) are logged in the **Log** window.
Highlight	The **Highlight** option opens a menu with multiple highlight color options. Select a color of your choice and the selected instances are highlighted with the chosen color. If any selected instances already have a different highlight, then such instances are highlighted with the chosen new color. This option is available when a single or multiple instances, nets, or ports are selected. If you highlight an instance, it automatically highlights the non-highlighted pins of the instance. Click **Unhighlight all** in the SmartDesign toolbar to remove the highlight color of all highlighted design objects, including highlighted nets. The **Highlight** option is also available in the right-click menu of the low-level instances in the **Expanded Inplace** view.
Help	This option brings up the handbook, the release notes, or the configuration user guides for the core.

4. Creating Pins and Ports for Connecting Instances in the SmartDesign (Ask a Question)

After placing the components in your SmartDesign canvas, you now need to create pins and ports to connect the various instances to complete your design.

Each pin or port has a direction and a type. Direction of the regular ports (non-Bus Interface ports) can be input, output, and bidirectional (inout).

The following is a list of types of pins or ports:

- **Scalar**: Single unit-level signal
- **Bus**: Array of scalar ports. The range of the bus ports are indicated by square brackets `[nFirst:nLast]`. For Example, `busName[3:0]`.
- **Slice**: Slice is a part of a bus port. A bus slice is of any range size within the bus range. For example, `sliceName[0]`, `sliceName[2:0]`.
- **Bus Interface (BIF)**: A Bus Interface is a bundle of scalar or bus ports that has functional meaning to it. BIF ports have characteristics such as functional types and roles that define how two Bus Interfaces can connect to one another. SmartDesign provides a rich set of AMBA Bus Interfaces–`AXI`, `AXI4`, `AXI4Stream`, `AHB`, `AHBLite`, and `APB3` – to help create easily AMBA sub-systems. It also provides Microchip specific Bus Interface types for easy connectivity between Microchip hardware components.

Scalar ports or pins in SmartDesign may have a PAD characteristic property. A pin or port with a PAD property must be connected to a top-level port of the design. PAD ports eventually assign to a package pin. In SmartDesign, these ports are automatically promoted to the top-level and can be modified, if needed.

For better organization of the instance pins, you can create groups. Groups are displayed on an instance with a Group pin that is just a visual representation and cannot be connected. Any pin can be added or removed from the group. Groups can be collapsed to hide member pins.

4.1. Creating Pins and Ports for your Design (Ask a Question)

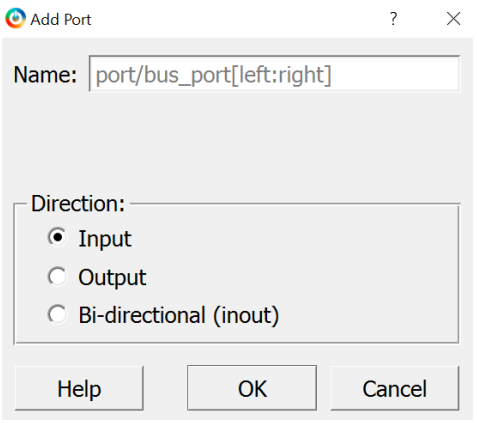
After placing the components in your SmartDesign canvas, you now need to create pins and ports to connect the various instances to complete your design.

Each pin or port has a direction and a type. Direction of the regular ports (non-Bus Interface ports) can be input, output, and bidirectional (inout).

1. You can create a pin or port in either of the following ways:
   - On the menu, click **SmartDesign** > **Add Port**
   - On the SmartDesign toolbar, click **Add Port**

   Result: The **Add Port** dialog appears.

Figure 4-1. Add Port Dialog



2. Enter an appropriate and unique name for the port in the **Name** box. You can specify a bus port by indicating the bus range directly into the name using brackets [ ]. For example, `mybus[3:0]`.

Important: If the port name violates HDL naming rules, an error message is printed in the **Log** window, and the new port is not created.

3. Choose the pin or port type that you want to create as per the following table and then click **OK**. The chosen port type is created in the SmartDesign canvas.

Table 4-1. Pin and Port Type Description

Pin or Port Type	Representation	Purpose
Input	- Unconnected - Connected	Input port for operating with inputs. By default, the input pins and ports are placed on the left side of the canvas.
Output	- Unconnected - Connected	Output port for operating with outputs. By default, the output pins and ports are placed on the right side of the canvas.
Bi-directional (inout)	- Unconnected - Connected	Bi-directional port for operating as an input or an output pin or port. By default, the bi-directional pins and ports are placed on the right side of the canvas.

Important: You can choose to move the ports to any location of your choice in the SmartDesign canvas.

Tip: To remove a port from the top-level, right-click the port, and select **Delete** from the right-click menu or select the port and press the **Delete** key.

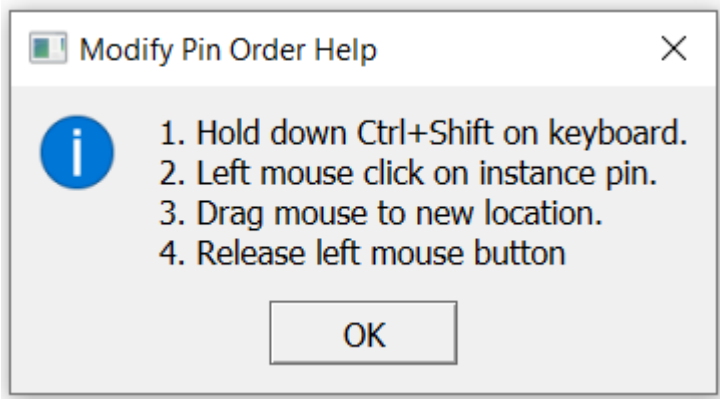
4.2. Modifying the Pin Order (Ask a Question)

If the outputs of one component (instance A) communicate with the inputs of another component (instance B) and otherwise, nets may intersect with each other and the view may be cluttered. To prevent this, the **Modify Pin Order** functionality allows you to modify the default pin placement of the instance.

To modify the default pin placement of the instance:

1. Right-click the component and select **Modify Pin Order** > **Modify Pin Order** from the right-click menu. The **Modify Pin Order Help** dialog box appears.

Figure 4-2. Modify Pin Order Help Dialog Box



2. Follow the instructions in **Modify Pin Order Help** dialog box to move the pin.
A pin that has been moved away from default locations is identified by a bold arrowhead. An inward-pointing arrowhead indicates an input pin and an outward-pointing arrowhead indicates an output pin. Inout pins do not have an arrowhead when they are moved away from the default locations (right side of instance).

Figure 4-3. Connections Between Two Instances with Regular Pin Order

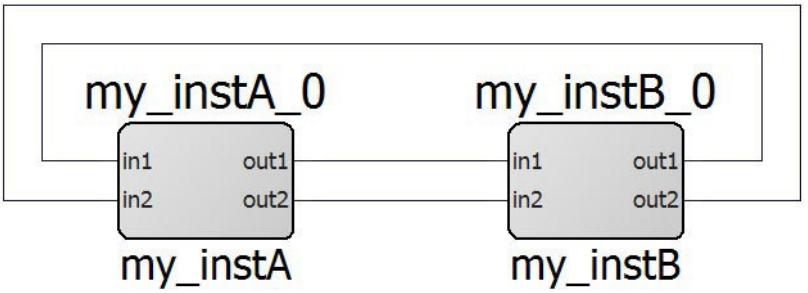
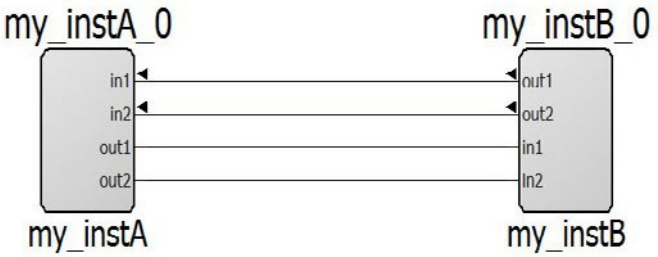


Figure 4-4. Connections Between Two Instances with Modified Pin Order



3. To reset the instance pin order to its default order, right-click the instance, and select **Modify Pin Order** > **Reset Pin Order** from the right-click menu.

Important: The **Modify Pin Order** operation is unavailable when the instance is expanded in place. The modified pin order might not be preserved when an instance is expanded but retains the set order when folded.

4.3. Performing Operations on Pins or Ports (Ask a Question)

When one or more pins or ports are selected, you can use the right-click menu to perform the following operations on the ports and pins.

Table 4-2. Possible Operations on Pins and Ports

Operation	Action
Connect	The **Connect** command connects the selected pins and ports with a net. If there is a net selected, it is used to make the connection. This is the only pin or port action that takes selected nets into account. If the connection is not possible, an error message is printed in the **Log** window. A connection is established only if all the selected objects can be connected.
Disconnect	The **Disconnect** command disconnects all the selected non-pad pins or ports from their attached net.
Promote to Top-Level	The **Promote to Top-Level** command is available to all non-PAD pins. It creates a port and a net connecting the port to the pins or slices. If a port with the same name already exists, a new unique port name is created. For example, if a BIF (`myBIF`) pin contains a pin (`myPort`), then the top-level `myPort` is named `myBIF_myPort` after the `myBIF` pin is promoted to the top level.
Go to Driver	The **Go to Driver** command zooms onto the driver of the selected pin or port. The **Go to Driver** command is not available for output pins and input ports. The driver cannot be an inout.
Magnify Pin	Double-click a pin/port or right-click and select **Magnify Pin** to zoom into the pin/port connection. The **Magnify Pin** window shows the specified pin/port connections. If the pin has a fanout of more than one, the number beside the **+** sign on the right shows the total fanout count. Click the **+** sign to see all the fanouts of the pin. You can double-click the net, pin, port, or instance inside the **Magnify** window to zoom and select the item.
Modify/ Rename	The **Modify** or **Rename** command opens a **Modify Port** dialog box. You can change the port name and the range of the port. **Notes:** • All slices of the bus are deleted if the range is changed. • When renaming BIF port, all the member ports that have BIF name prefix are renamed.
Delete	The **Delete** command deletes all the selected items that can be deleted: slices, user created groups, group members, and ports. The **Delete** command deletes all the selected items, 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.
Mark Unused	This option is available to the output pins (scalar, bus, and BIF) of an instance. The **Mark Unused** command allows you to show the output pins that are not being used in the design and must not be flagged as a warning when generating the design or running the DRC operation.
Invert	The **Invert** command inverts the input or output scalar pin and port. A bubble is added to indicate inversion.
Tie High	The **Tie High** command connects the pin (scalar and bus) to a logical 1. 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.
Tie Low	The **Tie Low** command connects the pin (scalar and bus) to a logical 0. 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.
Tie Constant	The **Tie Constant** command 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.
Clear Attributes	The **Clear Attributes** command clears the pin attributes (Tie to High, Low, Constant, Inversion, or Marked Unused).

Table 4-2. Possible Operations on Pins and Ports (continued)

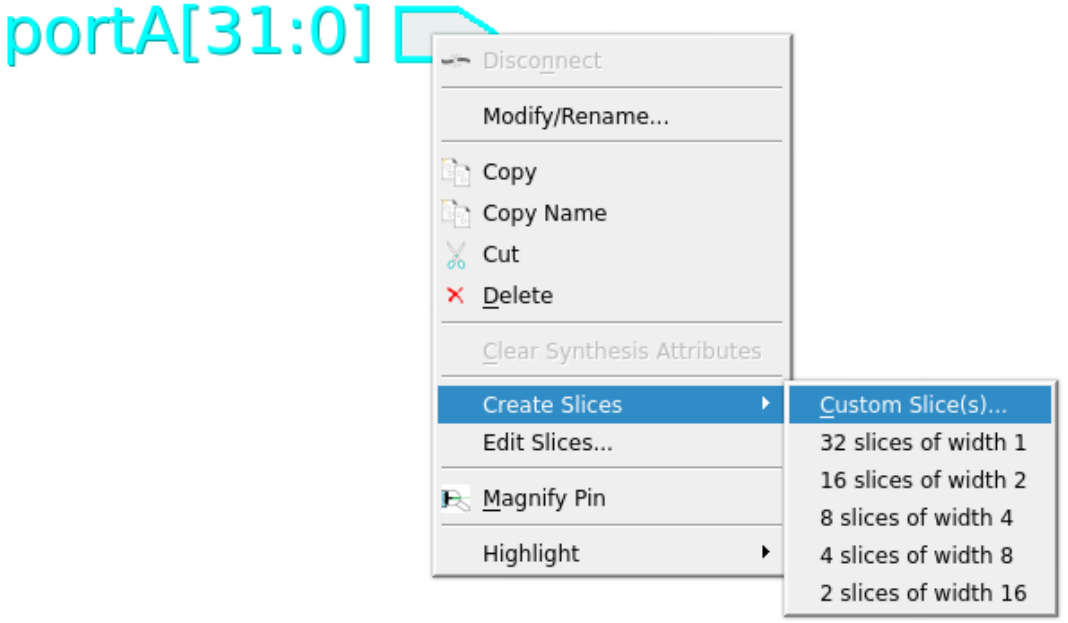
Operation	Action
Highlight	The **Highlight** option opens a menu with multiple highlight color selections. Select a color to highlight the selected items. If any items are already highlighted, then choosing a different color highlight changes from the previous highlight color. This option is available when a single or multiple instances, nets, and ports are selected. If you highlight an instance, it automatically highlights the non-highlighted pins of the instance. Click the **Unhighlight all** icon ( ) in the toolbar to remove the highlight color of all highlighted design objects, including highlighted nets. The **Highlight** option is also available in the right-click menu of the low-level instances in the **Expanded Inplace** view.
Bus and Slice operations	When a bus, slice pins, or ports are selected, the right-click menu in addition to regular pin actions have additional commands. For more information, see Working with Bus and Slice.
Add Pin to New Group and **Add Pin to Group**	This menu item is available to instance pins. When a group is selected, right-click a pin and choose **Add Pin to Group** to add the pin to the selected group. If no group is selected, the **Add Pin to New Group** command is available and creates a new group with the default group name, such as **Group**, **Group_1**, **Group_2**, **Group_3**, and all the selected pins are added to the newly created group. A pin group is expanded to display the member pins or is collapsed to hide the member pins. If a group is collapsed, pins that are not connected to nets or have attributes (tied low or high, tied to constant, and marked unused) are hidden.
Rename (Group)	Group pins, unlike other instance pins, can also be renamed. Right-click **Group** and choose **Rename**. The **Rename Group** dialog box appears. You can choose to change the name of the group in this dialog box and click **OK** to apply the changes. If an invalid name is entered or a pin with that name already exists, then an error message is printed in the **Log** window.

4.4. Working with Bus and Slice (Ask a Question)

You can perform the following operations on a Bus or Slice using the right-click menu options:

- **Flip Bit Order**: This command is available only for slices. This option allows you to flip the slice range. For example, `mySlice[10:0]` is flipped to `mySlice[0:10]` of the slice. All connection or tieoff information and presentation information is retained.
- **Create Slice**: This command opens a menu of slice options that can be created from the bus pin or ports. The custom slice(s) option can be used to create any slice or bit combination of your choice. For example, using a 32-bit bus with the Custom Slices option allows you to create any slice or bit combinations (For example, a slice of 10 bits and another slice of 22 bits). To make it convenient to create slices, common slice ranges for the selected bus are listed in the list. For example,
  - 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 4-5. Slice Creation for a 32-Bit Bus

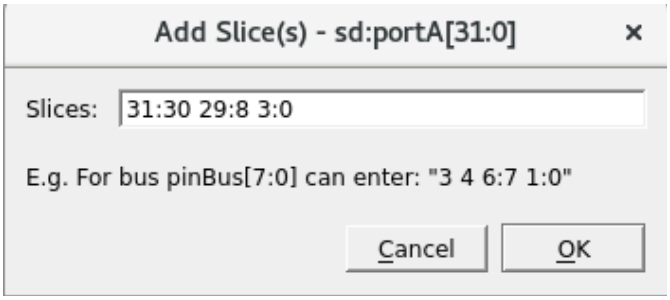


The directions of the slices (input or output) are indicated by an arrowhead and match the bus pin or port direction. On a bus port, the slices are placed in a column behind the bus port.

Important: If slices existed before, the predefined slices are created and the existing slices are deleted before creating the new ones.

- **Custom Slices:** This command opens the **Add Slice(s)** dialog box for entering a list of slices.

Figure 4-6. Add Slices Dialog Box

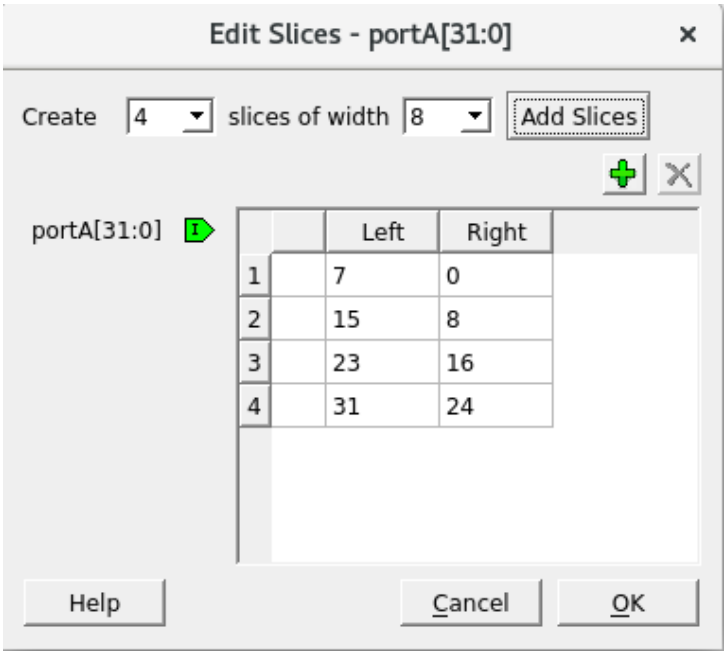


If these slices are all valid, they are added to the bus. If the slices are not valid (For example, out-of-range slices, overlapping slices for input pins (output ports), or existing slices), the error is printed in the **Log** window. The dialog box supports any separator character except colon because the colon is used to specify a range. No characters other than the colon are allowed to be adjacent to the two-range indices.

Tip: Creating a **Custom Slice** does not delete any pre-existing slices.

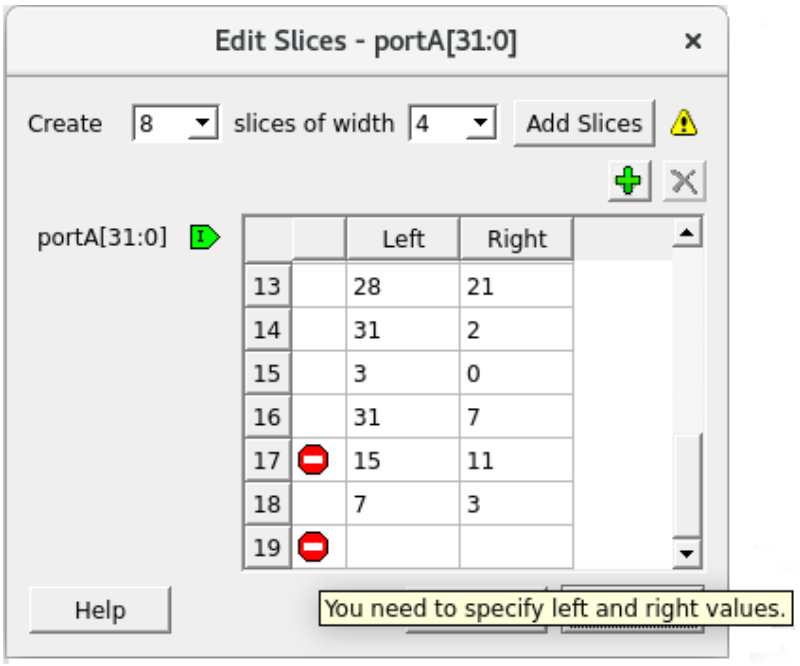
- **Edit Slice**: This command opens the **Edit Slices** dialog box. You can choose to delete, modify, and create a new slice, if required.

Figure 4-7. Edit Slices Dialog Box



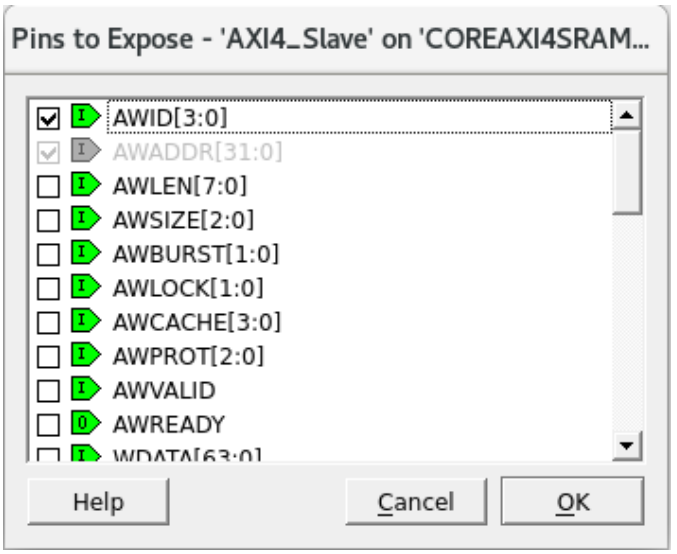
Use the dialog box to change the range of the bits, add a slice, or delete a slice. If you specify invalid range values, an error icon appears in the **Edit Slices** dialog box. Hover your cursor over the error icon to display the error message.

Figure 4-8. Edit Slices - Tooltip and Error Message



- **Pins to Expose:** This command opens the **Pins to Expose** dialog box displaying all the **Bus Interface** pins that can be exposed or hidden.

Figure 4-9. Pins to Expose Dialog Box



In the dialog box, you can check the pins in the BIF that you want to expose and deselect the ones that you want to be hidden. Hidden BIF pins are not visible in the SmartDesign canvas.

Important: Not all pins can be exposed. If a BIF pin is already connected, none of the input pins can be exposed. If a BIF pin is not connected, every item in the menu can be exposed.

5. Connecting Instances in your SmartDesign (Ask a Question)

In a SmartDesign, pins and ports are connected by nets. You can create manual connections between pins of the instances and ports in your design using the canvas.

Unused input scalar pins and output ports are set to high (1) or low (0). Unused input bus pins and output ports are set to a constant. Unused output pins and input ports are left unconnected when marked as unused.

Important: The nets that connect PAD pins and ports are called PAD nets. Such PAD nets cannot be deleted.

For more information one pins and ports, see .Creating Pins and Ports for Connecting Instances in the SmartDesign

To connect instances in your design, use either of the following methods.

5.1. Method 1: Using the Drag and Drop Feature (Ask a Question)

Hovering your cursor over a pin or port activates the connection mode automatically and the cursor changes to a crosshair. Click and drag the mouse pointer to the target pin or port and release the mouse button to make the connection.

- For pins, the connection mode activates when you hover over a pin icon or a pin name.
- For nets, the connections mode activates when you hover over any part of the net.
- The connection mode does not activate when you hover over pins, ports, or nets in the following cases:
  - When in a hierarchical view
  - When **Zoom to Selected Area** mode is activated
  - When **Add Note** mode is activated
  - When the canvas is in a read-only **Filtered** mode

5.2. Method 2: Using the Right-click Menu (Ask a Question)

You can also connect multiple pins or ports through the right-click menu options.

To connect the selected pins, perform the following steps:

1. Select a pin
2. Press the **Ctrl** key and select the required pins
3. Right-click on any of the selected pins and click **Connect**

Note: The **Connect** menu option is enabled only if the connection operation between the selected pins is logically feasible.

5.3. Showing or Hiding Nets and Net Names (Ask a Question)

You can choose to show or hide nets and net names on the canvas.

To show or hide nets, click **Show or Hide Net** toggle icon in the SmartDesign toolbar. 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.

To show or hide net names, click **Show or Hide Net Names** toggle icon in the SmartDesign toolbar. The net names are displayed alongside the net. Hiding net names makes the canvas less cluttered for big designs.

Tip: Net names always display in a tooltip when the cursor is hovered over the net.

5.4. Performing Operations on Nets (Ask a Question)

When a net or multiple nets are selected, the following commands are available in the right-click menu.

Operation	Action
Connect	The **Connect** command combines all the selected pins or ports to form a connection. Selecting a net is functionally equivalent to selecting all pins and ports that this net is connected to. Therefore, if you connect a net and a pin, it is equivalent to connecting all pins/ports of the net and the newly selected pin/port. The **Connect** command can also be used to connect a net to another net if one of the nets is not driven.
Go to Driver	When a net is selected, the **Go to Driver** command centers the view on the net's driver pin or port, zooms away, and selects the net driver. **Go to Driver** traces the net to the driver at the local level of hierarchy. It does not traverse hierarchy. The **Go to Driver** command is not available when multiple nets are selected, or the selected net has no driver. In the latter case, the net displays as a dotted line.
Highlight	The **Highlight** command highlights all the selected nets with the picked color. If any design objects are already highlighted, then highlighting a net with a different color overwrites the previous highlight color. The highlight action is available when a single or multiple design objects are selected. **Tip:** Highlighting a net, highlights the net and all the pins or ports (through the hierarchy) connected to the net. To remove the highlight on a single design object, right-click the design object and click **Remove** . To remove the highlight on all design objects, click **Unhighlight All** in the SmartDesign toolbar.
Rename	The **Rename** command opens a dialog box for a new net name to be entered to replace the current name. Enter a unique and valid net name and click **OK**. If the name already exists, an error message is printed in the **Log** window, the **Rename** dialog box closes, and the name is not changed.
Delete	The **Delete** command deletes the selected net. When multiple nets are selected, all are deleted. **Note:** Not all design objects can be deleted. For example, PAD ports and nets cannot be deleted.

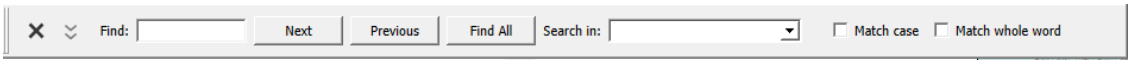
6. Finding Objects in Your Project (Ask a Question)

You can quickly and easily find objects in your project using the **Find bar** functionality.

To find for objects in your project, perform the following steps:

1. On the menu, click **View** and select **Find Bar**. The **Find** dockable dialog box appears at the bottom the Libero main window.

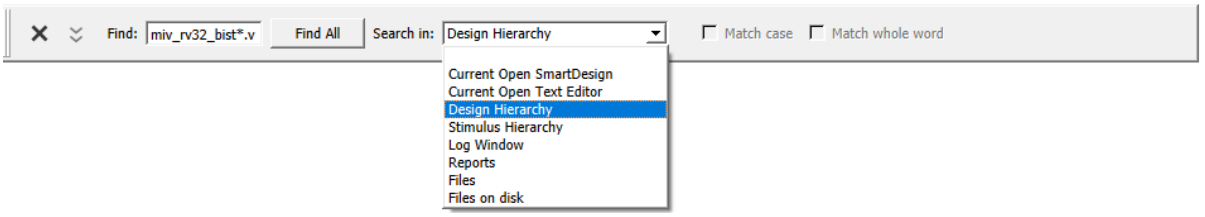
Figure 6-1. Find Dialog Box



Tip: Ensure to retain the **Find Bar** option enabled to avoid performing step 1 and directly continue with step 2 whenever Libero is launched next.

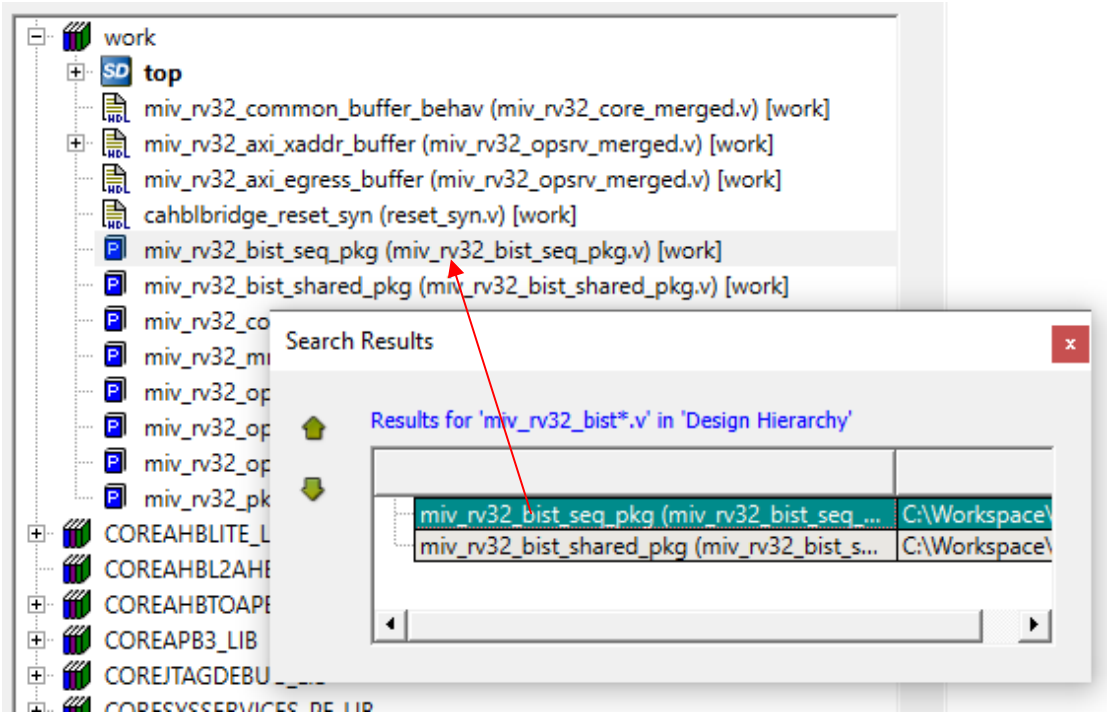
2. In the **Find** box, specify the object type you want to find.
3. In the **Search in** box, select the location in which you want to search the object.

Figure 6-2. Find Dialog Box with Find and Find in Filled



4. Click **Find All**. If the objects you are trying to find are available in the specified search location, the search results are populated in the **Search Results** dockable dialog box. When you select a search result, it is highlighted and displayed in the library.

Figure 6-3. Search Results Dialog Box with Results Populated



7. Using the Smart Search and Connect Tool (Ask a Question)

The **Smart Search and Connect** tool manages large designs with many pins, ports, and nets to connect. Design objects can be searched and connected easily without looking at the SmartDesign canvas. You can see the connections on the SmartDesign canvas as you make them. Any errors in the design are printed in the **Log** window.

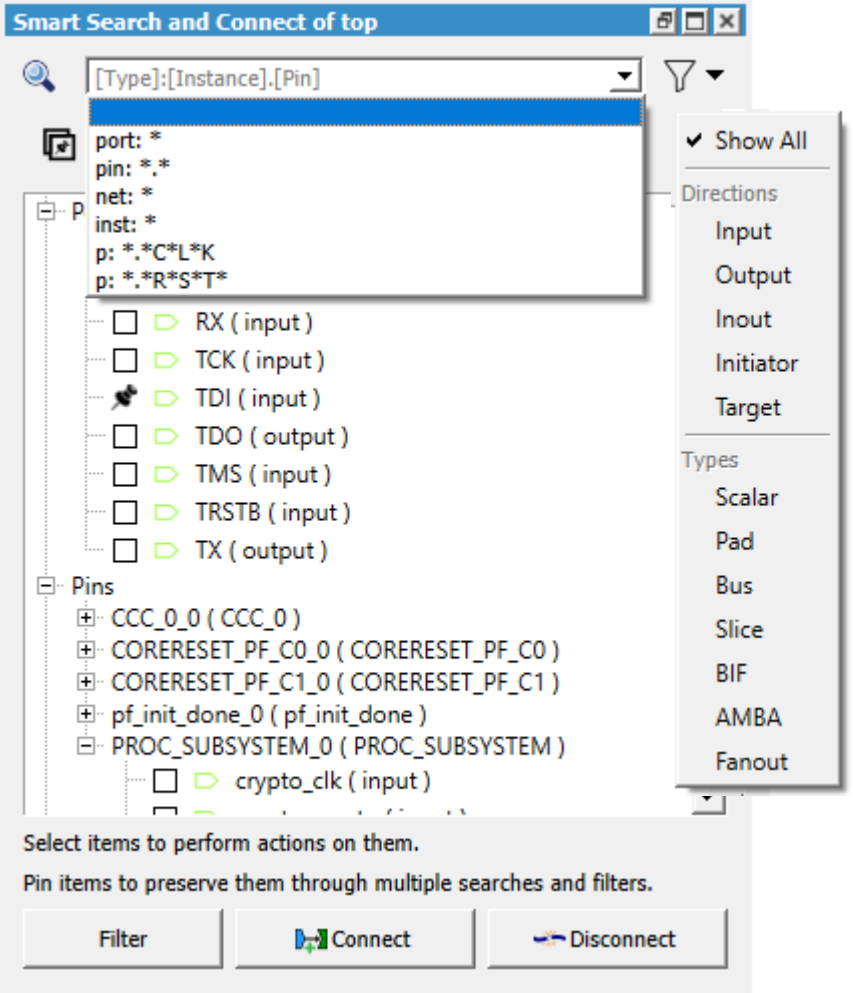
The **Smart Search and Connect** tool also help with complex searching and efficiently connecting multiple objects (clocks, resets, and so on) at the same time. You can search items using keywords and filters, select them, and then connect them using the right-click menu options. Alternatively, you can click **Connect** or **Disconnect** in the **Smart Search and Connect** tool window. It is possible to keep the result of a search by pinning the desired items and performing multiple searches. The pinned items are retained and available in search results, even if they do not match search patterns.

To launch the **Smart Search and Connect** tool, use either of the following methods:

1. On the menu, click **Edit** and select **Find**.
2. On the SmartDesign toolbar, click the **Smart Search and Connect** toggle icon.

Result: By default, the dockable **Smart Search and Connect** dialog box appears to the right side of the canvas. You can undock the tool window and move it a different location in the SmartDesign tool window as needed.

Figure 7-1. Smart Search and Connect Tool



7.1. Searching for Design Objects in the Canvas (Ask a Question)

You can use the **Search** box in the **Smart Search and Connect** tool to search and connect design objects. The **Search** box uses the following search pattern: **[Type]:[Instance].[Pin]**.

Important: You can keep searched items by pinning them. Pinned items remain visible through multiple searches.

You can do a simple search when your search consists of only alphanumerical characters. All pins, ports, instances, and nets that match the search pattern appears in the search results tree view automatically. If the search pattern has the "." separator symbol, SmartDesign tool treats the part before the separator as an instance name and the part after as a pin name.

You can also search by the following keywords: **port:**, **pin:**, **instance:**, **net:**, and ***:**. Alternatively, you can use their abbreviated versions. You must specify the keywords at the beginning of the search pattern.

The SmartDesign canvas zooms in and highlights 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.

7.2. Narrowing the Search for Design Objects Using Filters (Ask a Question)

You can use the narrow search filter button ( ) to filter the design objects based on either pin or port directions or pin or port types or with a combination of both in the Smart Search and Connect tool. The selected filter is applied on the search results and are displayed in the treeview.

The following table depicts the filter states.

Table 7-1. Design States on Design Objects

Icon State	Description
	Filter is selected
	No filter is selected

7.3. Applying the Selected Filters on the Design Objects in the Canvas (Ask a Question)

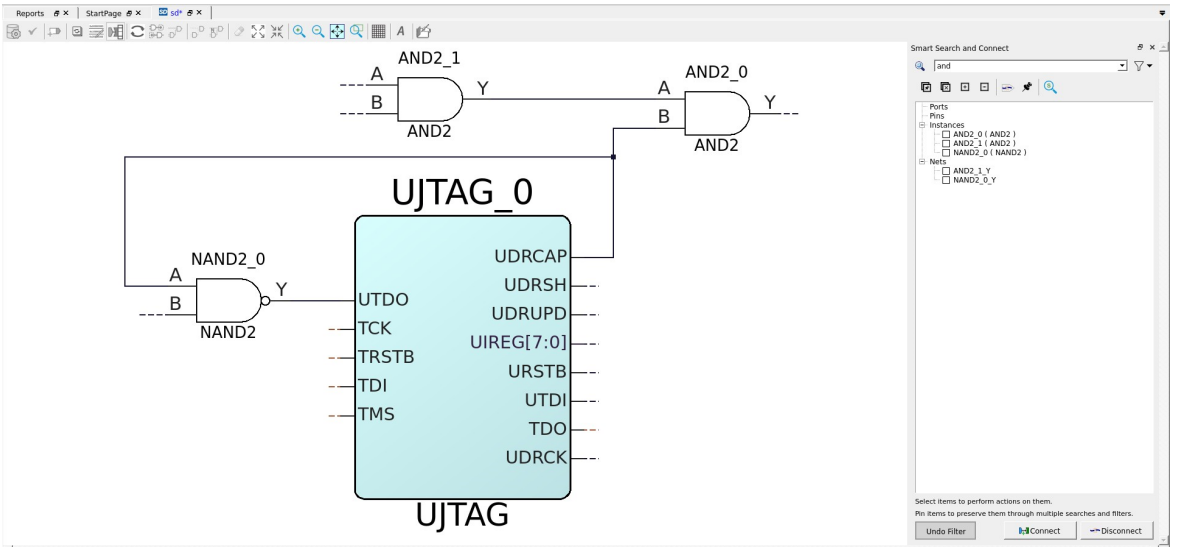
You can apply the selected filters and display only a subset of the SmartDesign objects using the **Filter** toggle button in the **Smart Search and Connect** tool. This view is suitable to use with large designs to focus on a particular structure of the design, such as the clock, reset structure, or AMBA sub-systems for instance.

To create a filtered view in the SmartDesign canvas, click **Filter** in the **Smart Search and Connect** tool. The SmartDesign canvas applies the filters and redraws the canvas with the filtered design objects visible in the **Smart Search and Connect** tools treeview.

Important: The filtered SmartDesign canvas with the object(s) are ***read-only***.

In the **Filtered** canvas view, except for the navigation commands all other editorial commands and right-click options are disabled. The **Filtered** canvas view is created with a cone structure. To add new objects to the filtered canvas, double-click the dotted nets to see all the connected nets to these nets in an expanded view. The net appears dotted if it is connected to an instance that is not included in the filtered view. You can double-click the dotted net to load all the instances connected to that net.

Figure 7-2. Filtered Canvas View



To add new objects to the filtered view from the Smart Search and Connect tool's treeview results, right-click the object of interest and select **Add to Filtered View**. When you add a pin or port to the filtered view, the tool adds the net connected to the object(s).

To exit the read-only filtered view to a normal SmartDesign canvas, perform the following steps:

- Toggle the **Undo Filter** button.
- Close the **Smart Search and Connect** window using **X** button.
- Press **Ctrl+Z** or **Undo** button.

7.4. Connecting and Disconnecting the Ports or Pins (Ask a Question)

You can choose to connect or disconnect pins and ports in your design.

Connecting Pins or Ports

You can connect the unconnected pins or ports in either of the following ways:

- In the Smart Search and Connect tool's treeview, hold the **Ctrl** key pressed, select the unconnected pins or ports that you want to connect, and click the **Connect** button in the Smart Search and Connect tool window.
- In the SmartDesign canvas, hold the **Ctrl** key pressed, select the unconnected pins or ports that you want to connect, right-click on any of the selected pin or port, and select the **Connect** option.

Result: Connections are made between the selected pins or ports.

Disconnecting Pins or Ports

You can disconnect the connected pins or ports in either of the following ways:

- In the Smart Search and Connect tool's treeview, hold the **Ctrl** key pressed, select the connected pins or ports that you want to disconnect, and click the **Disconnect** button in the Smart Search and Connect tool window.
- In the SmartDesign canvas, hold the **Ctrl** key pressed, select the connected pins or ports that you want to disconnect, right-click on any of the selected pin or port, and click the **Disconnect** option.

Result: Connections between the selected pins or ports are disconnected.

8. Working with Hierarchical SmartDesigns (Ask a Question)

A hierarchical SmartDesign is a better way to organize your main top level design. In the hierarchical approach, your top level design can have multiple sub-components or blocks. Each sub-component in turn can be expanded or collapsed to view the additional sub-components of your design with their own building blocks.

The nested structure allows you to traverse up or down your main design. The SmartDesign tool automatically takes care of any data conflicts and makes it easier to manage the instances in your design.

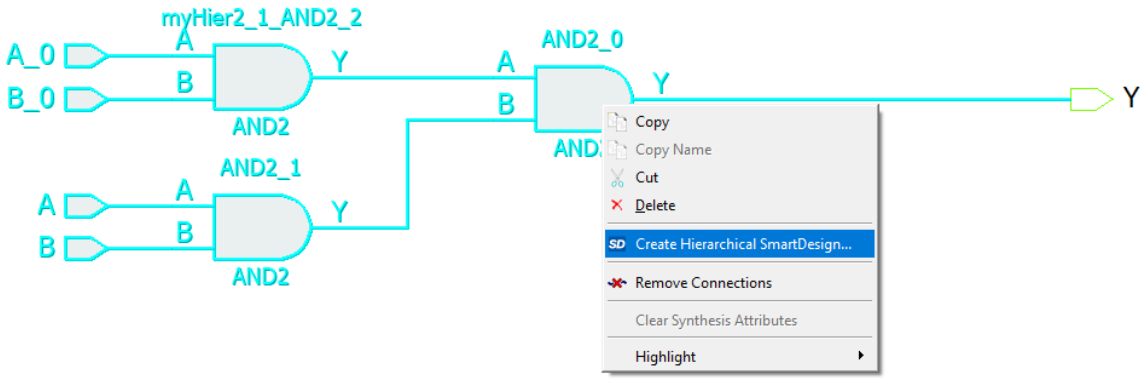
8.1. Creating a Hierarchical SmartDesign (Ask a Question)

This feature allows you to select one or more instances in your SmartDesign and create a new SmartDesign out of it (all the nets and connections from the instance are retained). This feature helps you organize a complicated design by creating sub-components on the fly from within the current SmartDesign canvas.

To create a new hierarchical SmartDesign, perform the following steps:

1. Select one or more design object(s) in your SmartDesign.
2. Right-click any selected design object(s) and select **Create Hierarchical SmartDesign** from the right-click menu. The **Create New Hierarchical SmartDesign** dialog box appears.

Figure 8-1. SmartDesign View with Create Hierarchical SmartDesign (Option for Selected Portion)



Important: The selected portion must contain at least one instance.

3. In the **Name** box, enter an appropriate unique name for the hierarchy and click **OK**.
   Result: A new SmartDesign component is created with the specified name and all the selected items are cut and pasted in the new SmartDesign.
   The newly created SmartDesign is automatically instantiated in the SmartDesign canvas, the sub-design is replaced with the new SmartDesign component, and all the connections are restored.

A error message is printed in the **Log** window depending on the following scenarios.

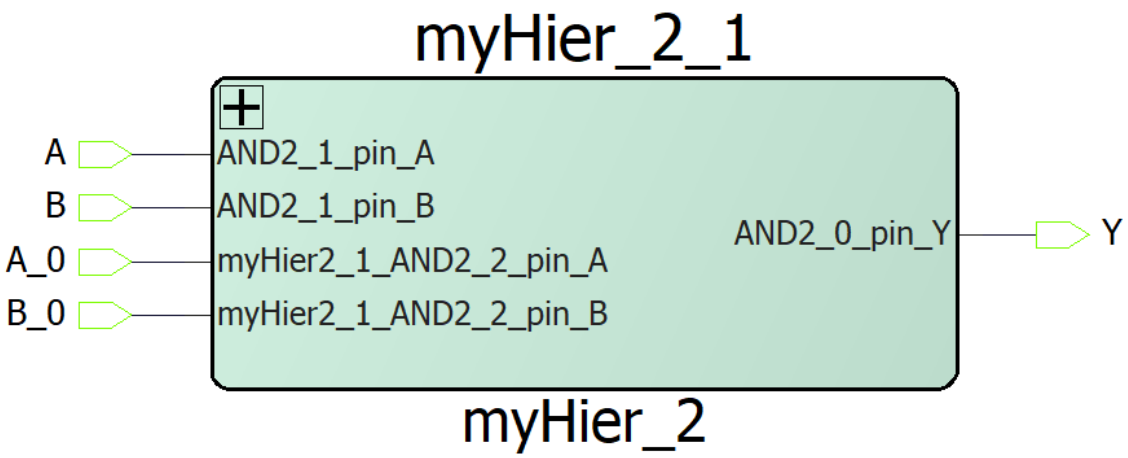
Scenario	Message
If another component or HDL module with the same name already exists in the project.	`Error: SmartDesign with the specified name already exists. Please specify a different name.`

Creating a Hierarchical SmartDesign (continued)	
Scenario	**Message**
If the specified name violates any HDL naming rules.	`Error: Cannot create hierarchical SmartDesign '[specified_name]' because it violates HDL naming rules. Please specify a different name.`

Important: The following conventions are followed while creating a hierarchical SmartDesign:

- Default name of a hierarchical SmartDesign is: *hier_[index]*.
- Ports are created in the hierarchical SmartDesign for all the unconnected instance pins.
- Port names are created for the Hierarchical SmartDesign using the following syntax: *[instance_name] + _pin_ + [pin_name]*.
- PAD ports do not follow the same naming rules and are promoted to top level automatically.
- If a port name already exists, index suffix is added to the name.
- Sliced ports are created similarly to the regular bus ports.
- A hierarchical SmartDesign can be created out of a single instance.
- Creating a hierarchical SmartDesign out of the ports is not allowed. At a minimum, one instance must be selected.

Figure 8-2. Hierarchical SmartDesign (Port Names Folded Inside the Block)

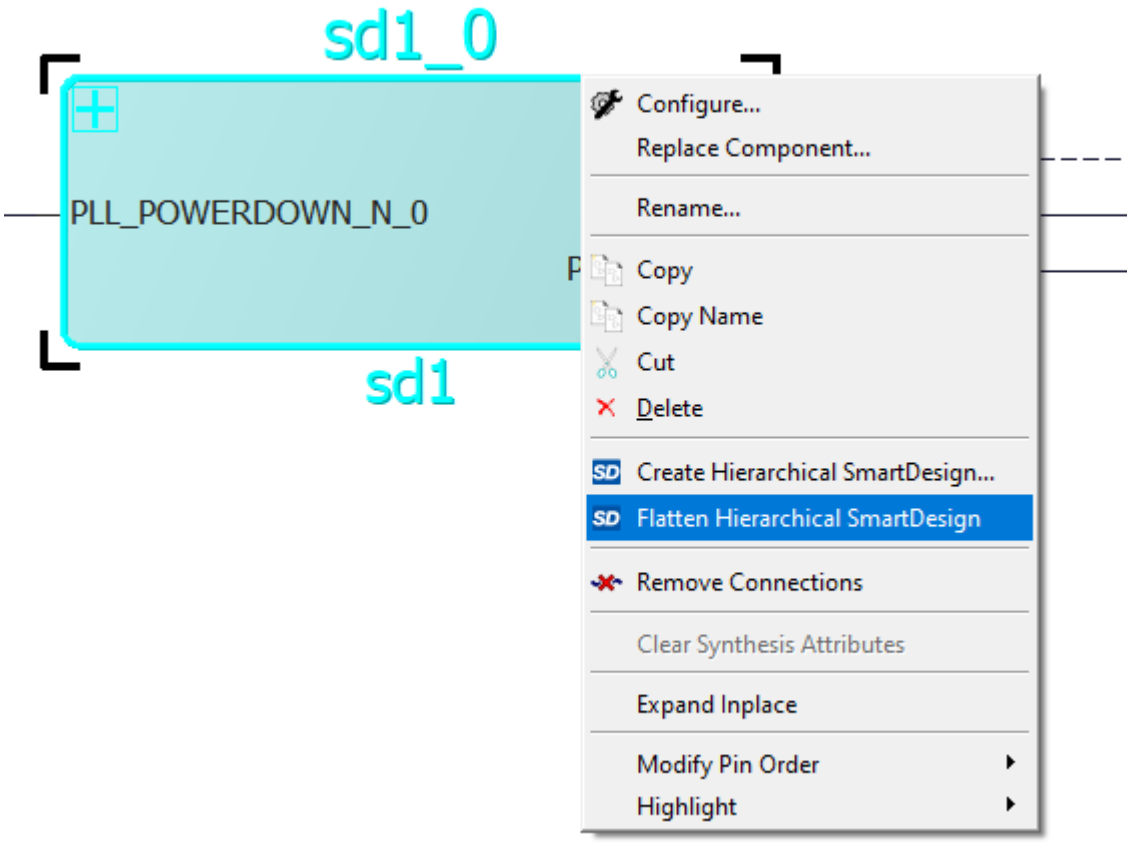


8.2. Flattening a Hierarchical SmartDesign (Ask a Question)

The **Flatten Hierarchical SmartDesign** operation extracts all the objects from the hierarchical SmartDesign and places them at the top-level while keeping all inner and outer connections, attributes, slices, and exposed pins.

To flatten a hierarchical SmartDesign, right-click any hierarchical SmartDesign instance and select **Flatten Hierarchical SmartDesign**.

Figure 8-3. SmartDesign View with Flatten Hierarchical SmartDesign



Result: All the objects from the selected hierarchical SmartDesign are extracted and placed at the at the top-level while keeping all inner and outer connections, attributes, slices, and exposed pins.

The following modifications are applied to the flattened design:

- If there is an inverted pin in the hierarchical instance and a corresponding port or pin (or both) from the hierarchical SmartDesign is also inverted, the resulting pin is inverted if inversion count is odd else it is not inverted.
- When there is an inverted pin in the hierarchical SmartDesign and the corresponding pin from the hierarchical instance has another attribute (tied high or low), the inversion count is odd and the tied attribute flips.
- Internal nets and inner instances are renamed as `<Hierarchical instance name> + _ <Internal net name or inner instance name>`.
- After flattening hierarchical SmartDesign, synthesis attributes of the outer and inner nets are merged. If both have the same attribute, the value of the outer net is used.

Important: The following conventions are followed while flattening a hierarchical SmartDesign:

- The selected instance is removed
- All the instances are copied from the hierarchical SmartDesign component to the current SmartDesign
- All the connections, attributes, slices, and exposed pins are restored automatically

8.3. Expanding and Folding Instances (Ask a Question)

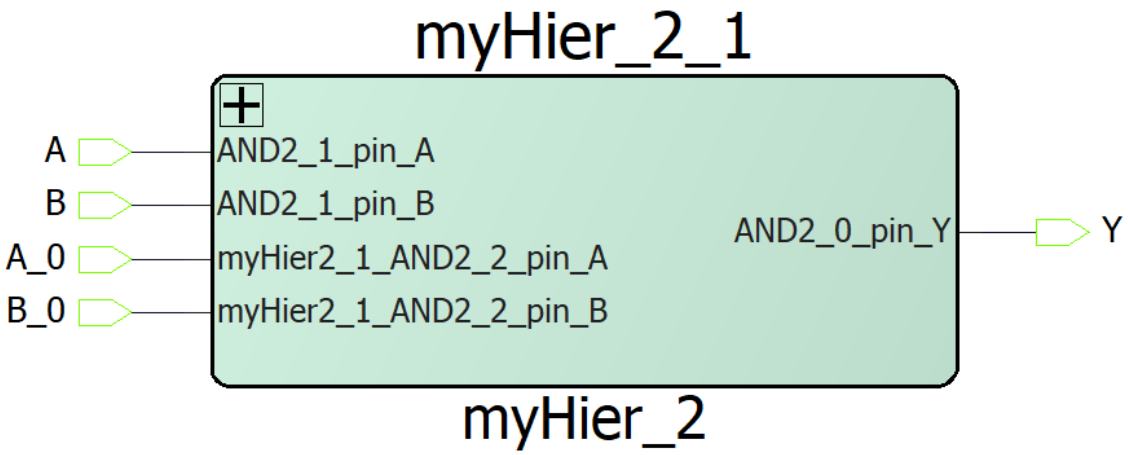
The **Expand Instance** and **Fold Instance** functionality enables you to traverse up or down the design hierarchy of a SmartDesign component.

The **Expand Instance** functionality enables you to traverse one level down of the hierarchy for viewing, while the **Fold Instance** functionality allows you to collapse to the next higher-level. Expanding and folding of an instance is executed in place within the canvas.

Important: Instance is read-only in the expanded view.

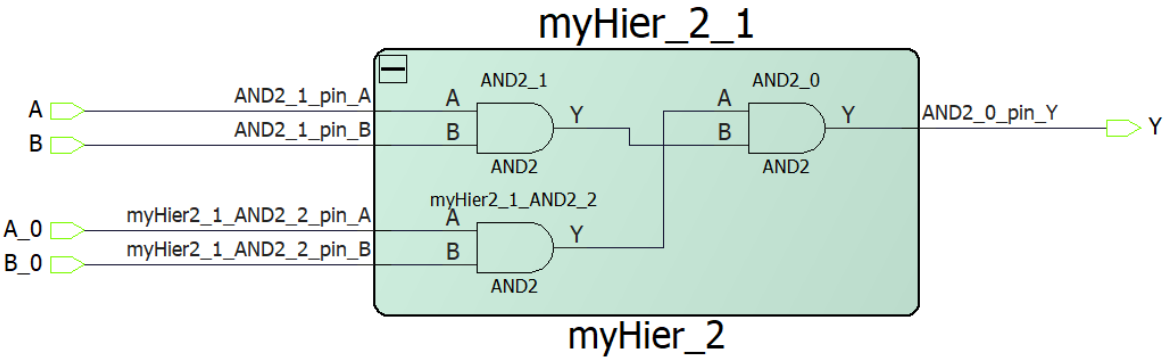
A folded design is indicated by a **+** symbol and an expanded design is indicated by a **-** symbol at the top-left corner of the instance.

Figure 8-4. Top-Level Design–Folded



To expand a folded design, click the **+** symbol. The folded instance is expanded to the next lower level of the hierarchy. Alternatively, right-click the instance and select **Expand Instance** in the right-click menu.

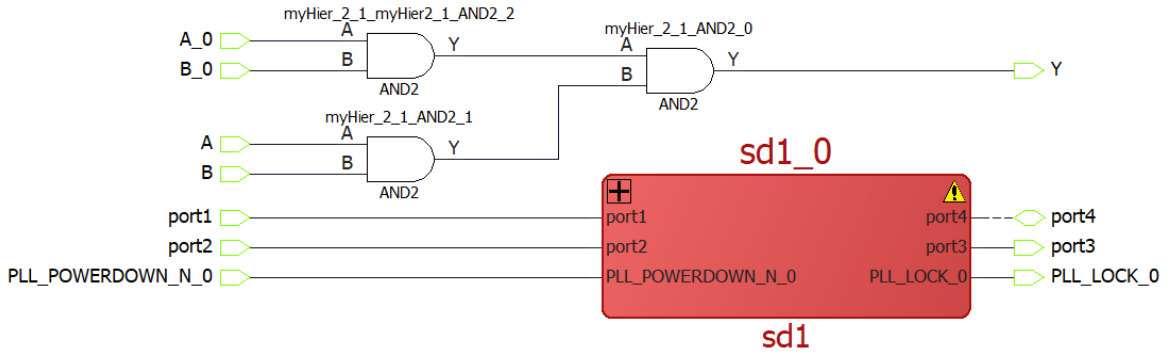
Figure 8-5. Top-Level Design–Expanded



To fold and expanded design, click the **-** symbol. The expanded instance is folded to the next higher-level of the hierarchy. Alternatively, right-click the instance and select **Fold Instance**.

When a lower-level component is modified, all the parent components containing modified component are marked with an asterisk **(*)** adjacent to their names. The modified component is highlighted in red, and a status icon is displayed in the instance's top-right corner.

Figure 8-6. Red Highlight and Changed Module Identification



Important:

- The red highlight and the **warning icon** appears only after the higher level component(s) are folded or expanded to expose the changed lower-level component.
- Instead of **ports changed**, the error becomes a **missing module** when a lower-level SmartDesign component instantiated in a higher-level module or component is deleted from the project.

If a changed, missing module, or component is at the lowest level of the hierarchy, the **Update Instance** and **Generate Components** steps must be repeated at every level of hierarchy on the top of the changed module or component until the top-level hierarchy is reached, updated, and regenerated.

The **Expanded View** does not reflect changes to the lower-level design in real time. To see the updated view, you must collapse the hierarchy and expand it again.

The top-level ports in the low-level components are shown differently in the **Expanded Inplace View** if the following are top-level ports:

Figure 8-7. Low-Level Component View with Inverted and Sliced Port Sliced

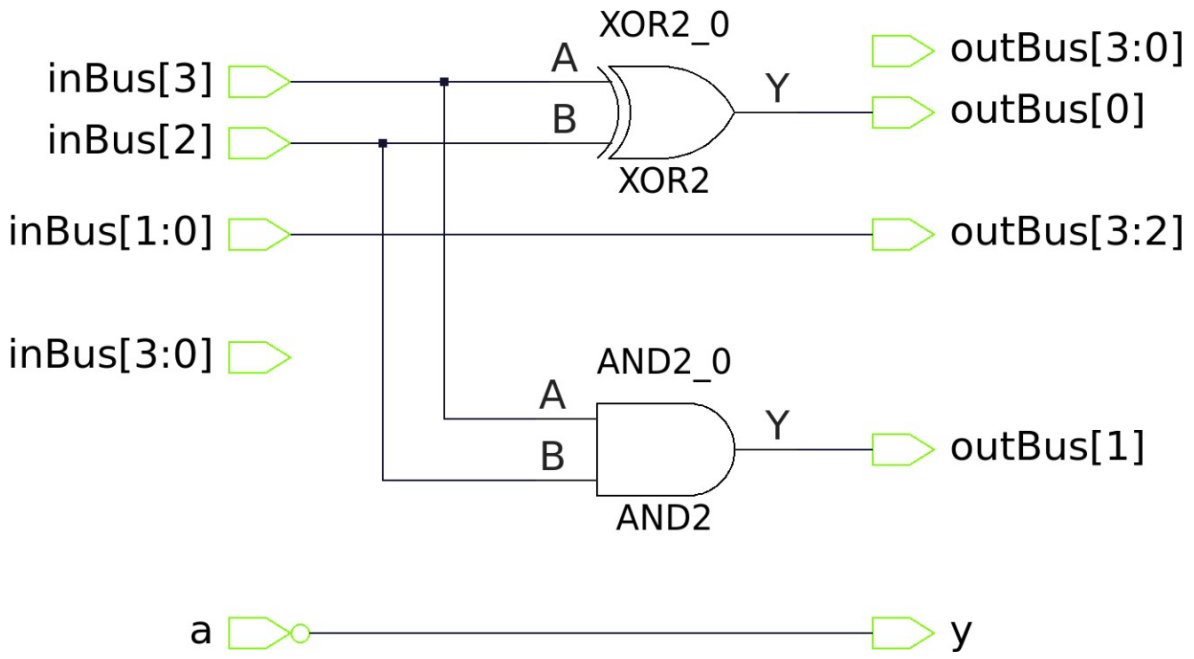
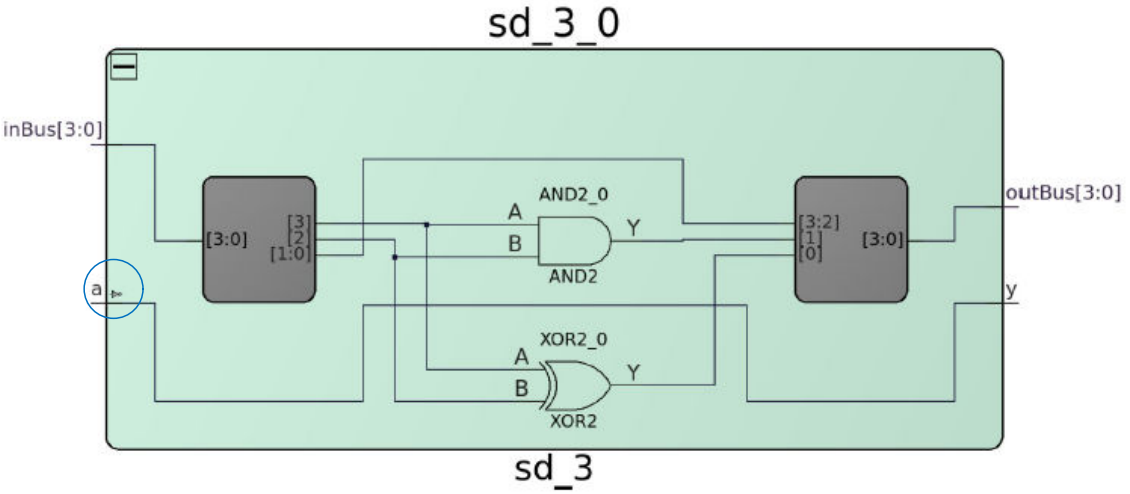


Figure 8-8. Expanded Inplace View with Inverter Symbol and Slicer Ports



In the **Expanded Inplace View**, an inverter symbol is displayed next to the port (circled in the preceding figure) for inverted ports. For the slice ports in this view, a slicer instance is inserted for each sliced bus.

9. Validating the SmartDesign Using the DRC (Ask a Question)

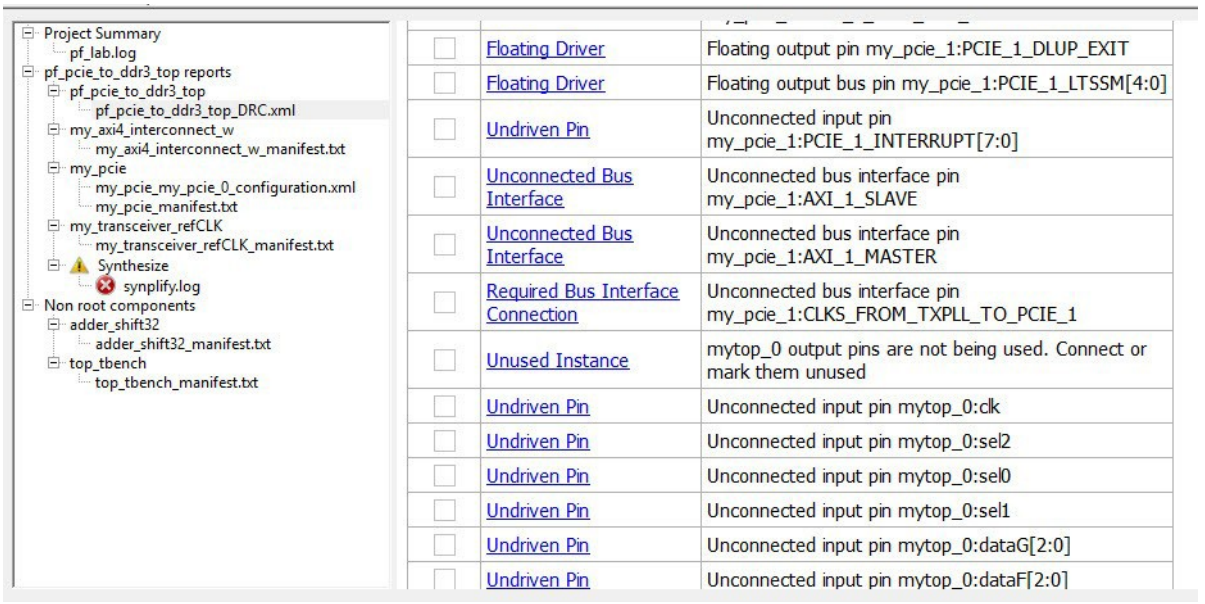
After instantiating all the components and connecting them together, the SmartDesign component must be validated and generated.

Click **Design Rule Check** in the SmartDesign toolbar to generate the SmartDesign component HDL file. This component can be used by downstream processes such as synthesis and simulation. You can also instantiate the SmartDesign component into another SmartDesign. When you generate your SmartDesign, the tool invokes the Design Rules Checker (DRC) to verify the connectivity of your design. Undriven input pins (output ports), floating output pins (input ports), and any other DRC violations are reported.

These errors are printed in the **Log** or **Message** window. Any errors must be addressed before a component can be generated successfully.

Tip: You can choose a generated HDL file language from the **Project** > **Project Settings** > **Design flow** tab.

Figure 9-1. DRC Report



The following table list the DRC errors and the corresponding solutions to fix the errors.

Table 9-1. DRC Errors and Solutions

DRC Error	Solution
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 interface – ports and/or parameters, referenced by this instance.
Undriven Pin	You must connect the pin to a driver or tie it to "1" (tie high), "0" (tie low) or, a constant (tie to constant for bus\slice pins).
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 a required connection.

Table 9-1. DRC Errors and Solutions (continued)

DRC Error	Solution
Required Bus Interface Connection	You must connect this Bus Interface before you 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 in a design. For example, there can only be one Arm® Cortex®-M1 or CoreMP7 in a design due to silicon constraints. 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 Microchip SoC to obtain the necessary license.
No Top level ports	There are no ports in this design. Use the **Create Port** menu or toolbar command to create ports and connect these ports to the intended pins in the design.
Self-Instantiation	A component cannot instantiate itself. This message is reported only in the **Log** window.
Bus interface data width mismatch	There is a data width mismatch between the `<instance1>:<port1>[left1:right1]` and `<instance1>:<port1>[left2:right2]`, ports implicitly connected through a Bus Interface net, which might result in loss of data. To correct an error, make the initiator (driver) BIF pin write port width less than or equal to the target BIF pin write port width as per the port name shown in the DRC message. Another way to correct the error is to make the target BIF pin read port width less than or equal to the initiator (driver) BIF pin read port width as per the port name shown in the DRC message. Preceding types of corrections require to reconfigure the Initiator or the Target core component.
Bus interface ID width mismatch	There is an ID width mismatch between the `<instance1>:<port1>[left1:right1]` and `<instance1>:<port1>[left2:right2]`, ports implicitly connected through a Bus Interface net, which might result in loss of data. To correct an error, make the initiator (driver) BIF Pin AWID, WID, or ARID port width less than or equal to the target BIF pin AWID, WID, or ARID port width as per the port name shown in the DRC message. Another way to correct an error is to make the target BIF pin RID or BID port width less than or equal to the initiator (driver) BIF pin RID/BID port width as per port name shown in the DRC message. Preceding types of corrections require to reconfigure the initiator or target core component.

10. Managing Synthesis Attributes (Ask a Question)

This feature allows you to add or modify the synthesis attributes of the SmartDesign objects (nets, ports, and instances) directly from the SmartDesign canvas.

Important: Synthesis attributes are not allowed to be set on BIF pins, BIF nets, pin groups, and slices. User-defined attributes are not supported.

Synthesis attributes that apply to the component can only be set on SmartDesign canvas. In the generated HDL file, those attributes are added to the SmartDesign module or architecture.

The following table lists the synthesis attributes available in the Synopsys® FPGA synthesis tool that are supported by SmartDesign.

Table 10-1. List of Synthesis Attributes

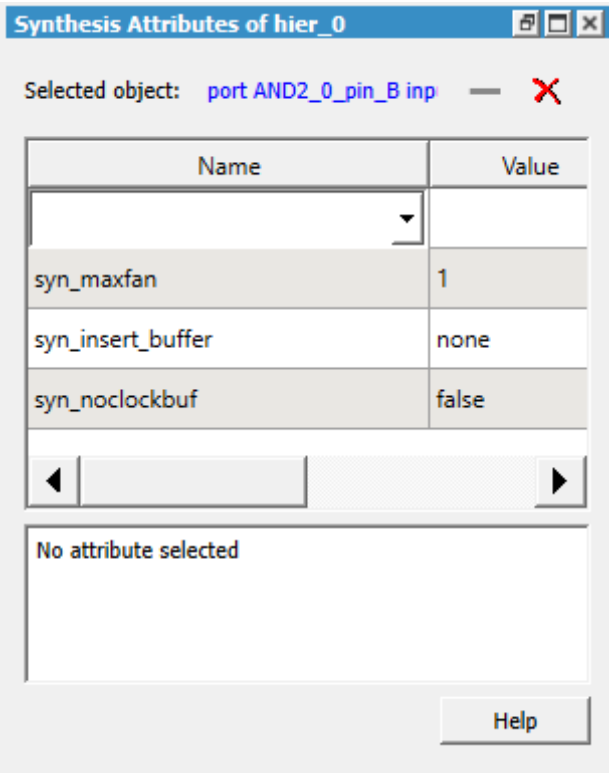
Attribute	Object
syn_insert_buffer	Port, instance
syn_keep	Net
syn_maxfan	Port, net, instance
syn_no_compile_point	Module or architecture
syn_noclockbuf	Port, net, module or architecture
syn_noprune	Instance, module or architecture
syn_preserve	Port, module or architecture
syn_hier	Module or architecture

Important: For more details for each attribute, see Synplify Pro® ME.

To add, modify, or remove synthesis attributes of the SmartDesign objects, perform the following steps:

1. Click **Manage Synthesis Attributes** icon in the SmartDesign toolbar. The **Synthesis Attributes** window appears.

Figure 10-1. Synthesis Attributes



2. Select any object of interest such as a port, net, or instance in the SmartDesign canvas, the attributes of the selected object is displayed in the **Synthesis Attributes** window.
3. You can perform the following actions:

Action	Steps
Add attributes	From the **Name** dropdown list box, select the synthesis attribute of your choice. The synthesis attribute is automatically added with a default value. The selected synthesis attribute description is displayed at the bottom of the window.
Modify attributes	Double-click the `Value` field of the synthesis attribute and change the default value as required.
Remove attribute	Click to select the synthesis attribute you want to remove and click **–**. To remove all synthesis attributes, click **X**. Alternatively, right-click on the port, net, or instance that has synthesis attributes and select **Clear Synthesis Attributes**.

11. Generating the SmartDesign Component (Ask a Question)

After creating your visual design, you need to convert your design into an HDL file that is used during synthesis and/or simulation of the design. Click **Generate Component** in SmartDesign toolbar to automatically invoke the DRC and check for any errors.

If there are any DRC errors, the error messages are printed in the **Message** window. All errors must be fixed for a successful component generation. Once all the errors are fixed, the component can be generated either recursively or non-recursively. Non-Recursive generation is enabled by default.

Important: Generating a SmartDesign component invalidates the synthesis state of the design if the component is used in that design.

11.1. Generating the SmartDesign Component Recursively (Ask a Question)

In recursive generation mode, the **Generate** functionality generates all non-generated components, including the SmartDesign and the configured core components in a bottom-up fashion. The parent SmartDesign components generates only if all the components it uses are generated successfully.

To enable recursive generation of designs, perform the following steps:

1. On the menu, click **Project** > **Preferences**. The **Project Preferences** dialog box appears.
2. In the **Project Preferences** dialog box, select **Design Flow** from the preferences list on the left side. All the design flow related options are displayed on the right side.
3. In the **SmartDesign generation options** group, select **Generate recursively**. Recursive generation of the SmartDesign is enabled.

11.2. Generating the SmartDesign Component Non-recursively (Ask a Question)

In non-recursive generation mode, the **Generate**command generates the active SmartDesign. If the generation is successful, SmartDesign is marked as successfully generated even if a sub-component is un-generated (either never attempted or unsuccessfully attempted). An un-generated component is annotates with an exclamation symbol circled in blue color in the **Design Hierarchy** tab.

To enable non-recursive generation of designs, perform the following steps:

1. On the menu, click **Project** > **Preferences**. The **Project Preferences** dialog box appears.
2. In the **Project Preferences** dialog box, select **Design Flow** from the preferences list on the left side. All the design flow related options are displayed on the right side.
3. In the **SmartDesign generation options** group, select **Generate non-recursively**. Non-recursive generation of the SmartDesign is enabled.

12. Creating a SmartDesign Testbench (Ask a Question)

After importing or creating and generating your design, create testbenches to simulate your design. You can create testbenches for your design in either of the following ways.

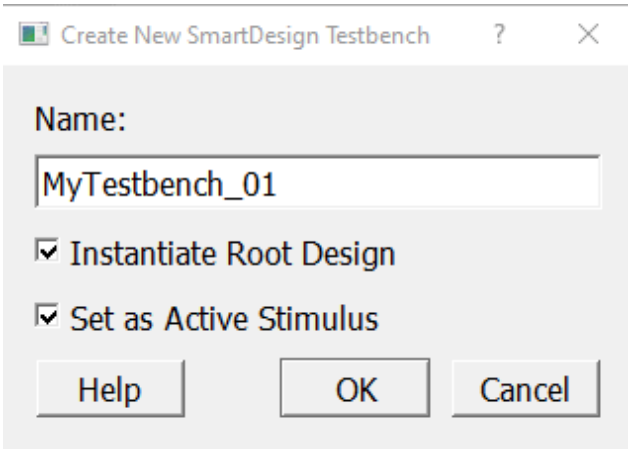
12.1. Creating a SmartDesign Testbench (Ask a Question)

To create a SmartDesign testbench using the SmartDesign canvas, perform the following steps:

1. Launch the **Create New SmartDesign Testbench** dialog box in either of the following ways:
   - On the menu, click **File** > **New** > **SmartDesign Testbench**.
   - On the **Design Flow** tab, double-click **Create SmartDesign Testbench**.

   Result: The **Create New SmartDesign Testbench** dialog box appears.

   Figure 12-1. Create New SmartDesign Testbench Dialog Box



2. Enter an appropriate and unique name for your design file in the **Name** box.
3. By default, the root design is instantiated in the testbench and is set as the active stimulus for the design.
   a. If you do not want to instantiate the root design in the testbench, deselect the **Instantiate Root Design** option. When you deselect the **Instantiate Root Design** option the **Set As Active Stimulus** option is automatically deselected and disabled.
   b. If you do not want the testbench to serve as the active stimulus, deselect the **Set As Active Stimulus** option.
4. Click **OK** to create the testbench. The newly created testbench appears in the **Stimulus Hierarchy** tab along with other testbenches and is displayed in the central tab of the IDE main window.
   Result: The testbench file is saved in a folder with the same name as that of the testbench itself and is placed in your `<project>\component\work\` folder.

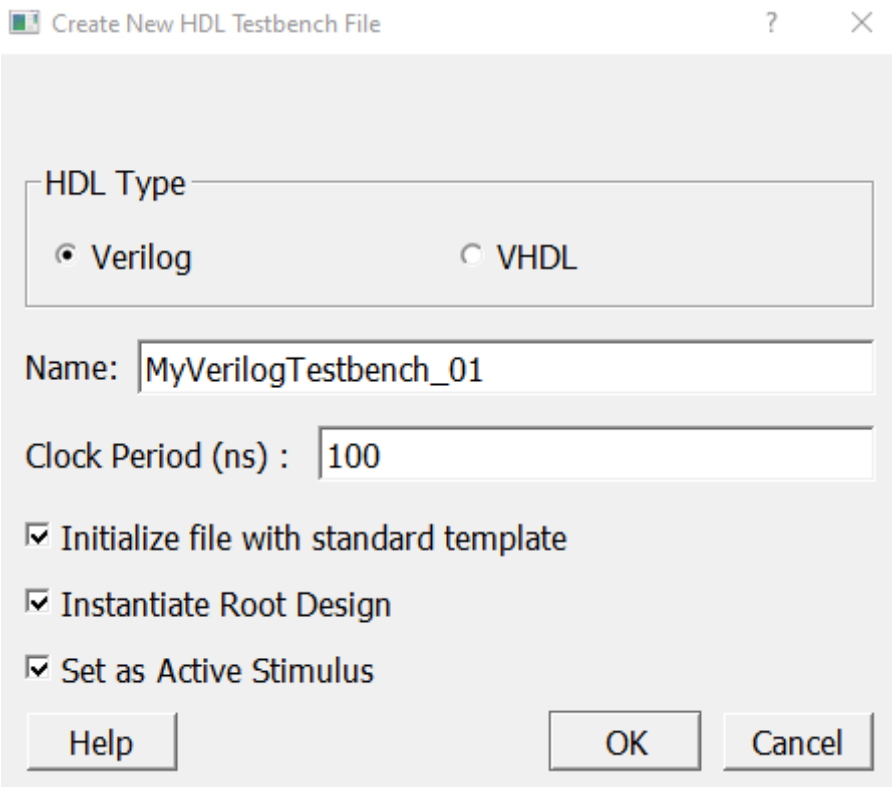
12.2. Creating a SmartDesign HDL Testbench Using the IDE (Ask a Question)

To create a HDL testbench using the IDE, perform the following steps:

1. Launch the **Create New HDL Testbench file** dialog box in either of the following ways:
   - On the menu, click **File** > **New** > **HDL Testbench**.
   - On the **Design Flow** tab, double-click **Create HDL Testbench**.

   Result: The **Create New HDL Testbench File** dialog box appears

Figure 12-2. Create New HDL Testbench file Dialog Box



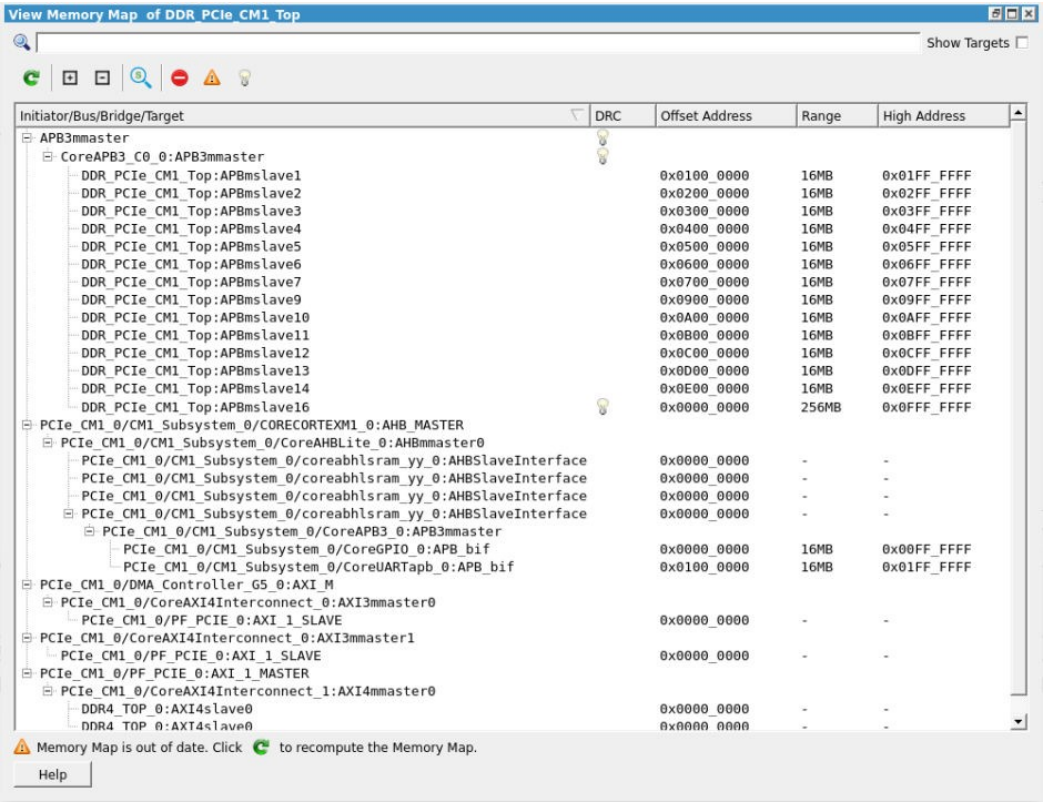
2. By default, the preferred language to write the testbench file is set to **Verilog**. If you choose to use VHDL, select the **VHDL** option.
3. Enter an appropriate and unique name for your testbench file in the **Name** box.
4. Enter the clock speed you want to set for the testbench in nanoseconds (ns) in the **Clock Period (ns)** box. By default, the value is set to *100ns*.
5. By default, the testbench is initialized with the built-in standard testbench file template, the root design is instantiated in the testbench, and the testbench is set as the active stimulus for the design.
   a. If you choose to not use the default standard template, deselect the **Initialize file with standard template** option.
   b. If you choose to not use the testbench to serve as the active stimulus for your design, deselect the **Set As Active Stimulus option**.
   c. If you choose to not instantiate the root design in the testbench, deselect the **Instantiate Root Design** option. When you deselect the **Instantiate Root Design** option the **Set As Active Stimulus** option is automatically deselected and disabled.
6. Click **OK** to create the testbench. The newly created SmartDesign HDL testbench appears in the **Stimulus Hierarchy** tab along with other testbenches and is displayed in the central tab of the IDE main window.

Result: The newly created SmartDesign HDL testbench is copied to your `<project>\hdl\` folder.

13. Viewing the Design Memory Map (Ask a Question)

The **View Memory Map** toggle window displays the memory map corresponding to various initiators in the SmartDesign component and displays each initiator to target(s) connectivity path in a tree format. To open the window, click **View Memory Map** icon in the SmartDesign toolbar. The dockable **View Memory Map** window appears to the right side of the canvas.

Figure 13-1. View Memory Map Window



This window shows the memory map starting from an initiator in the design to targets connected through bus and bridge cores. There can be various types of bus and bridge cores in the path from initiators to targets.

- **Initiator Node** > **Bus** > **Bridge** > **Bus** > **Targets**
- **Initiator Node** > **Bus** > **-Targets**

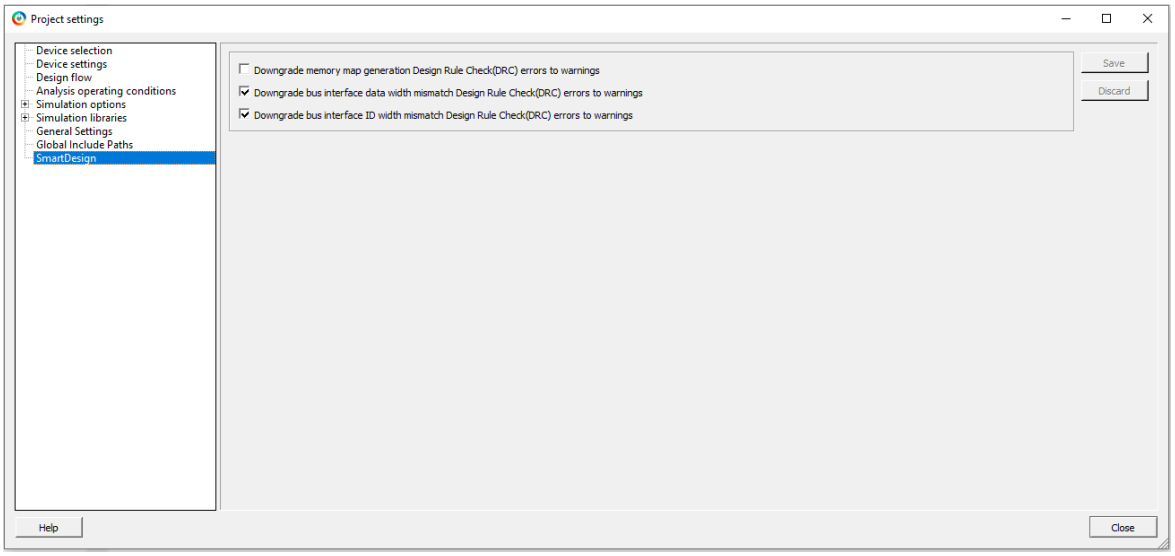
Important: The memory map also considers initiators, bus and bridge cores, and targets that are present in other SmartDesign components, which are instantiated under the current SmartDesign's hierarchy.

Each target in the memory map is shown with an Offset Address, Range, High Address, and DRC. If the Memory Map DRC detects a partially invalid or invalid structure, an Error or a Warning icon with a tooltip message is shown. DRC is flagged if a target cannot be accessed completely or partially by the initiator's address space.

DRC errors in the Memory Map corresponding to a SmartDesign component and its hierarchy are flagged when the SmartDesign component is generated and are printed as messages to the **Log** window. If DRC are warnings, the SmartDesign component generation passes. However, if the DRCs

are errors, the SmartDesign component generation fails. There is an option to downgrade Memory Map DRC errors to warnings and let the SmartDesign component generation to go through without failing. To set these options, open the **Libero SoC Project Settings SmartDesign** page.

Figure 13-2. Downgrade Memory Map Generation Design Rule Check (DRC) Errors to Warnings



Important: If Memory Map errors are downgraded to warnings using the preceding preference check box, the targets flagged by the DRC error cannot be accessed by the Initiator they are connected to. Check each DRC warning and ensure that there are no issues before going further in the design flow.

The **View Memory Map** window also provides buttons and filters at the top of the window for easy navigation and ease of use.

Figure 13-3. View Memory Map Window Actions and Filters

- **Search**: Searches for a specific initiator, bus, bridge, and target in the memory map by specifying a full or partial name in the search box.
- **Show Targets**: Shows all the initiator to target(s) paths without the buses and bridges that connect them with their start addresses, ranges, and DRCs if any. The initiators, bus, and bridge cores are not shown.
- **Refresh**: Updates the content of the Memory map. It becomes active only when something is changed on canvas.
- **Expand**: Expands and shows the full path of all the initiators in the memory map starting from the initiator to the targets.
- **Collapse**: Collapses the memory map tree and only shows the initiators in the window.

- **Zoom and Center**: Toggle action. If checked, the canvas zooms to the selected item after every selection.
- **DRC Filters**: It filters the DRC messages, if any. It categorized the message on the basis of severity: error, warning, and information.

Additional Memory Map functions allow you to sort items by their offset or high addresses and ranges.

1. Click the columns of the header to sort the items in ascending or descending order.
2. Click the first column of the header to see the initial view of the table.

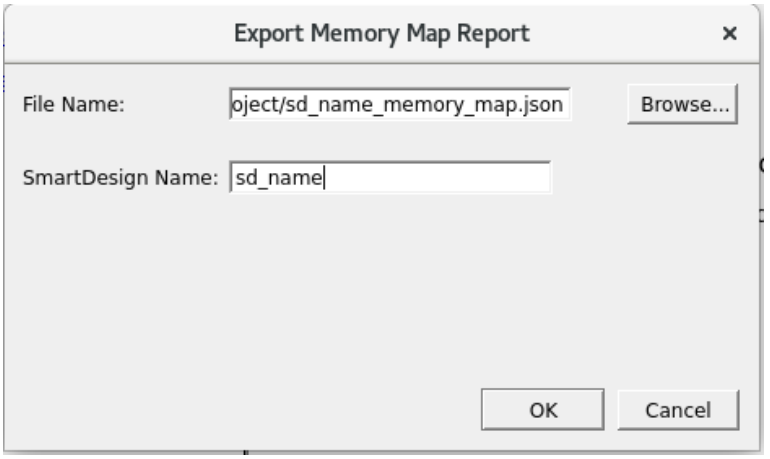
14. Exporting Memory Map Report (Ask a Question)

The **Export Memory Map Report** functionality enables exporting all necessary data about BIF connections in a `.json` or `.html` file. It contains the data about the initiator pins, target pins, and their respective connections on the SmartDesign canvas. It also includes information about specified start address and range of every target BIF connected to respective initiator BIF in the processor.

To export the memory map report, perform the following steps:

1. On the menu, click **File** > **Export** > **Memory Map Report**. The **Export Memory Map Report** dialog box appears.

   Figure 14-1. Export Memory Map Dialog Box

2. Click **OK**. A `.json` formatted memory map report file is created, which you can open in any text editor.
   Note: You can choose to export the memory map to a different location and with a different filename in either `.json` or `.html` format as required.

Sample Memory Map Report

```
{
    "title": "Memory Map Report",
    "date": "Tue Feb  17 00:00:17 2023",
    "project_name": "MSS_ICICLE",
    "project_location": "/home/user/MSS_ICICLE",
    "SmartDesign name": "FIC_1_PERIPHERALS",
    "Initiator/Bus/Bridge/Target OffsetAddress Range HighAddress":[
    {"Node name":"AXI4mmaster0",
    "Type":"Initiator",
        "Connected Node":[
            {"Node name":"FIC_1_INITIATOR_0:AXI4mmaster0",
            "Component name":"FIC_1_INITIATOR",
            "Type":"Bus",
                "Connected Node":[
            {"Node name":"PCIE:AXI_1_SLAVE",
            "Component name":"PF_PCIE_C0",
            "Offset Address":"0x0_0000_0000",
            "Range":"256GB",
            "High Address":"0x3F_FFFF_FFFF",
            "Type":"Target"
            },
            {"Node name":"FIC_1_PERIPHERALS:AXI4mslave0",
            "Component name":"PCIE_INITIATOR",
            "Offset Address":"0x0_6000_0000",
            "Range":"126GB",
            "High Address":"0x1F_FFFF_FFFF",
            "Type":"Target"
            }               ]           }        ]
    }
```

```
,
    {"Node name":"PCIE:AXI_1_MASTER",
    "Component name":"PF_PCIE_C0",
    "Type":"Initiator",
        "Connected Node":[
            {"Node name":"AXI_ADDRESS_SHIM_0:AXI4_TARGET",
            "Component name":"AXI_ADDRESS_SHIM",
            "Type":"Bus"
            }        ]
    }
    ]
}
```

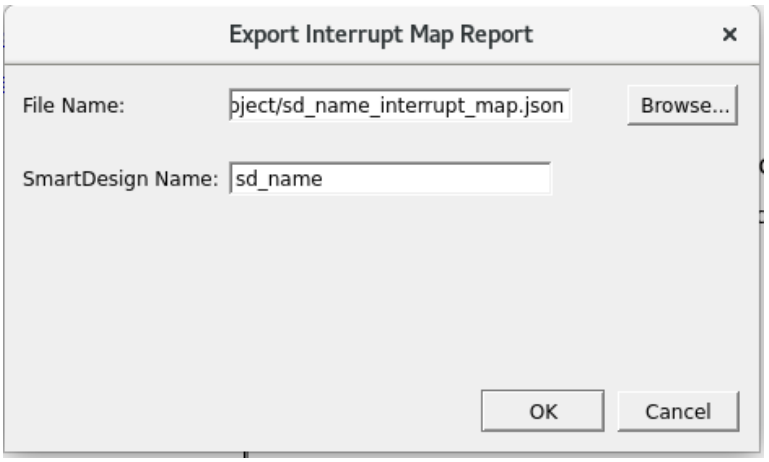
15. Exporting Interrupt Map Report (Ask a Question)

The **Export Interrupt Map Report** allows exporting the necessary data about the interrupt pin connections. It contains the data about MSS and MIV interrupt pins and their drivers. In particular, the hierarchical path to both pins, their instances, and the components where they are located. Besides for the connectivity information, the Platform Level Interrupt Controller (PLIC) number is also exporting for each MSS interrupt pin. The **Interrupt Map Report** functionality supports the **PFSOC_MSS** components and the **MIV** processors.

To export the interrupt map report, perform the following steps:

1. On the menu, click **File** > **Export** > **Export Interrupt Map Report**. The **Export Interrupt Map Report** dialog box appears.

   Figure 15-1. Export Interrupt Map Report Dialog Box



2. Click **OK**. A `.json` formatted interrupt map report file is created, which you can open in any text editor.
   Note: You can choose to the export the interrupt map to a different location and with a different filename as required.

Sample Interrupt Map Report

```
{
   "version":"0.1",
   "title":"Interrupt Map Report",
   "date":"Mon Jan 01 00:00:00 2022",
   "project_name":"my_project",
   "project_location":"/home/user/Projects/my_project",
   "SmartDesign name":"MPFS_ICICLE_KIT_BASE_DESIGN", /* The SmartDesign for which the report
is being exported. */
   "Processor interrupt map":[
      {
         "Processor instance":"ICICLE_MSS_0", /* If the MSS instance is located in the top
level SmartDesign the name of the instance should not be hierarchical. */
         "Processor component":"ICICLE_MSS",
         "Component type":"MSS",
         "Interrupt connection map(Interrupt pin name - Hierarchical driver pin name)":[
            {
               "Interrupt pin":"MSS_INT_F2M[0]",
               "PLIC":"118",
               "Driver pin":"Y", /* If the driver pin is located in the top-level
SmartDesign, the name of it should not be hierarchical. */
               "Driver component":"", /* If the driver pin is located in the top-level
SmartDesign, need to leave this field empty. */
               "Driver instance":"" /*  If the driver pin is located in the top-level
SmartDesign, need to leave this field empty. */
            },
```

```
            {
               "Interrupt pin":"MSS_INT_F2M[1]",
               "PLIC":"119",
               "Driver pin":"PCIE_1_INTERRUPT_OUT",
               "Driver component":"",
               "Driver instance":""
            },
            {
               "Interrupt pin":"MSS_INT_F2M[2]",
               "PLIC":"120",
               "Driver pin":"INTERRUPT",
               "Driver component":"",
               "Driver instance":""
            },
            {
               "Interrupt pin":"MSS_INT_F2M[3]",
               "PLIC":"121",
               "Driver pin":"mBUS_INT",
               "Driver component":"",
               "Driver instance":""
            },
            {
               "Interrupt pin":"MSS_INT_F2M[4]",
               "PLIC":"122",
               "Driver pin":"INT",
               "Driver component":"",
               "Driver instance":""
            },
            {
               "Interrupt pin":"MSS_INT_F2M[59]",
               "PLIC":"177",
               "Driver pin":"IHC_SUBSYSTEM_0/U54_4_IRQ_AGGREGATOR:IRQ",
               "Driver component":"MIV_IHCIA",
               "Driver instance":"IHC_SUBSYSTEM_0/U54_4_IRQ_AGGREGATOR" /* The hierarchical
path to the driver instance. */
            },
            {
               "Interrupt pin":"MSS_INT_F2M[60]",
               "PLIC":"178",
               "Driver pin":"IHC_SUBSYSTEM_0/U54_3_IRQ_AGGREGATOR:IRQ",
               "Driver component":"MIV_IHCIA",
               "Driver instance":"IHC_SUBSYSTEM_0/U54_3_IRQ_AGGREGATOR"
            },
            {
               "Interrupt pin":"MSS_INT_F2M[61]",
               "PLIC":"179",
               "Driver pin":"IHC_SUBSYSTEM_0/U54_2_IRQ_AGGREGATOR:IRQ",
               "Driver component":"MIV_IHCIA",
               "Driver instance":"IHC_SUBSYSTEM_0/U54_2_IRQ_AGGREGATOR"
            },
            {
               "Interrupt pin":"MSS_INT_F2M[62]",
               "PLIC":"180",
               "Driver pin":"IHC_SUBSYSTEM_0/U54_1_IRQ_AGGREGATOR:IRQ",
               "Driver component":"MIV_IHCIA",
               "Driver instance":"IHC_SUBSYSTEM_0/U54_1_IRQ_AGGREGATOR"
            },
            {
               "Interrupt pin":"MSS_INT_F2M[63]",
               "PLIC":"181",
               "Driver pin":"IHC_SUBSYSTEM_0/E51_IRQ_AGGREGATOR_0:IRQ",
               "Driver component":"MIV_IHCIA",
               "Driver instance":"IHC_SUBSYSTEM_0/E51_IRQ_AGGREGATOR_0"
            }
         ]
      }
   ]
}
```

16. Appendix A - Glossary (Ask a Question)

BIF	Abbreviation for Bus Interface. Logical grouping of ports or pins that represent a single functional purpose. A Bus Interface is a specific mapping of a bus definition onto a component instance. It may contain both input and output, scalars, or buses.
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 an initiator, target, 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 two or more compatible Bus Interfaces Canvas A visual representation of the canvas for placing components and stitching the components to create a functional design.
Driver	A driver is the origin of a signal on a net. The input and target BIF ports of the top-level or the output and Initiator BIF ports from instances are drivers.
Instance	A block-like item with pins on either side of it. These are connected to create designs. You may have multiple instances of a single component in your design. You usually have custom connections for each instance that differ from other instances of the same component.
Net	A wire that connects pins or ports in a design PAD The property of a port that must be connected to a design's top- level port. PAD ports eventually assigns to a package pin. In SmartDesign, these ports are automatically promoted to the top-level and cannot be modified.
Pins	Pins are the inputs or outputs or 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	A port 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, and bidirectional) 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 it connects 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 the designs you create in a standard way.
Viewport	The rectangular view area of the canvas that is visible to you. You can move the viewport around on the canvas or zoom in or zoom out to view your design. Showing the whole canvas would be too large in most cases.
Initiator BIF	The Bus Interface that initiates a transaction (such as a read or write request) on a bus.
Target BIF	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 initiator nor target; 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.
Port	An external interface connection to the outside world. Scalar if a 1-bit port, bus if a multiple-bit port or a Bus Interface (BIF). These are connected to the pad or package pins of the FPGA device.
Pin Group	A grouping of pins (scalar or bus) you create for easy connection or identification.

17. Appendix B - Available Cores in Libero SoC Catalog (Ask a Question)

The following is a consolidated table of all the available initiators, bus, and bridge cores supported by the memory map feature. The last column specifies, if needed, the minimum version of the core required for the memory map feature to work as expected.

Table 17-1. Initiators, Bus, and Bridge Cores Available in Libero SoC Catalog

Classification	Core	Core Type	Core Versions for Correct Memory Map Generation
Initiators	MIV_RV32IMC	AXI, AHBLite and APB3 Initiators	All available production versions supported
	MIV_RV32	AXI, AHBLite and APB3 Initiators	All available production versions supported
	MIV_RV32IMA_L1_AXI	AXI Initiator	All available production versions supported
	MIV_RV32IMA_L1_AHB	AHBLite Initiator	All available production versions supported
	MIV_RV32IMAF_L1_AHB	AHBLite Initiator	All available production versions supported
	CORERISCV_AXI4	AXI4 Initiator	All available production versions supported
	COREAXI4DMACONTROLLER	AXI4 Initiator	All available production versions supported
	COREABC	APB3 Initiator	All available production versions supported
Family-specific Initiator cores	PolarFire® SoC Standalone MSS (PolarFire SoC only)	AXI4 Initiator	All available production versions supported
	System Builder and SmartFusion® 2 MSS (SmartFusion2 and IGLOO® 2)	AHBLite and APB3 Initiators	All available production versions supported
	CORECORTEXM1(PolarFire, PolarFire SoC and RTG4™)	AHBLite Initiator	All available production versions supported
	PF_PCIE (PolarFire only)	AXI Initiator	All available production versions supported
	SERDES_IF, SERDES_IF2, SERDES_IF3 (SmartFusion 2 and IGLOO 2)	AXI and AHBLite Initiators	All available production versions supported
	PCIE_SERDES_IF (RTG4 only)	AXI and AHBLite Initiators	All available production versions supported
	COREHPDMACTRL (SmartFusion 2 and IGLOO 2)	AHBLite Initiator	All available production versions supported
	CORESYSSERVICES (SmartFusion 2 and IGLOO 2)	AHBLite Initiator	All available production versions supported
	CoreConfigMaster (SmartFusion 2 and IGLOO 2)	AHBLite Initiator	All available production versions supported
Bus cores	COREAXI4INTERCONNECT	AXI bus	Version v2.5.100 and above supported
	CoreAHBLite	AHBLite bus	All available production versions supported
	CoreAPB3	APB3 bus	All available production versions supported
	CoreAXI	AXI bus	All available production versions supported

Table 17-1. Initiators, Bus, and Bridge Cores Available in Libero SoC Catalog (continued)

Classification	Core	Core Type	Core Versions for Correct Memory Map Generation
Family-specific bus type cores	PF_DRI (PolarFire and PolarFire SoC)	APB bus	Support for generating Memory Maps is planned for a future release.
	CoreConfigP (SmartFusion 2 and IGLOO 2)	APB bus	All available production versions supported
Bridge cores	COREAXITOAHBL	AXI to AHBLite bridge	Version v3.5.100 and above
	COREAHBL2AHBL_BRIDGE	AHBLite to AHBLite bridge	Support for generating Memory Maps is planned for a future release.
	COREAHBLTOAXI	AHBLite to AXI bridge	All available production versions supported
	COREAHBTOAPB3	AHBLite to APB bridge	All available production versions supported
	COREAXITOAXICONNECT	AXI to AXI bridge	Support for generating Memory Maps is planned for a future release.

18. Appendix C - Keyboard Shortcuts (Ask a Question)

The following table lists the keyboard shortcuts available in the SmartDesign canvas.

Table 18-1. SmartDesign Keyboard Shortcuts

Shortcut Key	Description
Ctrl+w	Maximize or Minimize the canvas work area.
Ctrl+ + or Ctrl+scroll wheel up	Zoom in (same as clicking the Zoom In button in the toolbar or pressing Ctrl+ mouse).
Ctrl+- or Ctrl+scroll wheel down	Zoom out (same as clicking the Zoom Out button in the toolbar or pressing Ctrl+mouse scroll wheel down).
Ctrl+c	Copy selected instances, ports, and nets with all their properties to the clipboard.
Ctrl+v	If there is a copy or cut information of an instance or port available in clipboard, paste them. Cut data is removed from the source SmartDesign after paste.
Ctrl+x	Cut selected instances, ports, and nets.
Ctrl+h	Creates **Hierarchical SmartDesign** out of the selected portion.
Ctrl+J	Flattens the selected **Hierarchical SmartDesign**.
Ctrl+k, Ctrl+f	Open or close the **Smart Search and Connect** tool.
Ctrl+t	Open or close the **Manage Synthesis Attributes** tool.
Ctrl+z	Undo the last operation.
Ctrl+y	Redo the last operation.
Shift+click	This command is helpful when multiple items are selected for the same command (For example, promotion to top-level or Add to the group).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 the second clicks are selected automatically.
Ctrl+click	Toggle switch that selects or de-selects an item hovering your mouse cursor.
Ctrl+Shift+click and drag	Selects a pin of an instance and drag it to a new location. This is not available for macros and expanded in place instances.

19. Appendix D – Component Types (Ask a Question)

A component instance is a block-like item with pins on either side, connected to create designs. You might have multiple instances of a single component in your design. You usually have custom connections for each instance that differ from other instances of the same component.

The following table lists the component types and its diagrammatic representation in the SmartDesign canvas.

Table 19-1. Component Types in the SmartDesign Tool and Tooltip

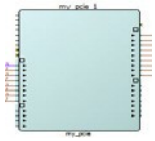
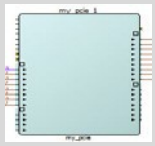
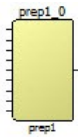
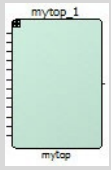
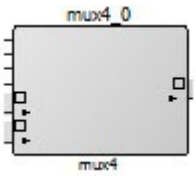
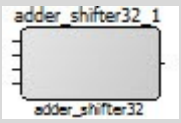
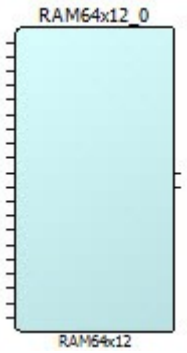
Icon	Types of Design Components	Tooltip Information
	Configured IP Core Component	Component: my_pcie Core: PF_PCIE 1.0.217
	IP Core directly instantiated from the Catalog	Core: PF_PCIE 1.0.217
	Block Component	Block: prep1
	SmartDesign Component	SmartDesign: mytop
	Parameterized HDL Core Modules	HDL Core: mux4
	HDL Module	HDL:adder_shifter32

Table 19-1. Component Types in the SmartDesign Tool and Tooltip (continued)

Icon	Types of Design Components	Tooltip Information
	Library Macro	Macro: RAM64x12

20. Appendix E – Operations Native to SmartDesign (Ask a Question)

This section describes the operations/functions native to the SmartDesign tool.

- Pad ports and pad nets cannot be copied or pasted
- Hierarchy instances can also be copied or pasted
- All the port attributes (tie-off values, inversions) are kept after pasting
- Port or pin slices are preserved. Slice port cannot be copied individually
- In the **Copy and Paste** or **Cut and Paste** Tcl commands, bus port names must be specified without a range. Names such as `bus_name[10:1]` are treated as invalid
- **Cut and Paste** do not have an Undo option
- **Cut and Paste** do not work within the same SmartDesign
- **Copy** or **Cut** or **Paste** are not supported across different working Libero projects

21. Revision History (Ask a Question)

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

Revision	Date	Description
N	05/2025	The following changes are made in this revision: • In the Replacing an Instance Component with a Different Component section, updated Figure 3-4 . • In the Creating Pins and Ports for your Design section, updated all the three unconnected icons in the Table 4-1. • In the Performing Operations on Pins or Ports section, updated the **Unhighlight all** icon in Table 4-2.
M	08/2024	This document is released with Libero SoC Design Suite v2024.2 without changes from v2024.1.
L	02/2024	This document is released with Libero SoC Design Suite v2024.1 without changes from v2023.2.
K	08/2023	This document is released with Libero SoC Design Suite v2023.2 without changes from v2023.1.
J	06/2023	Updated Table 17-1 table to remove the column that had the version numbers of the IP cores.
H	04/2023	The following is a summary of the changes made in this revision: • The topics in the user guide have been enhanced for clarity and reorganized for easy accessibility. • The memory map report has been updated to include the Component name.
G	12/2022	The complete document has been reorganized and few sections has been re-written to enhance user's experience.
F	08/2022	Removed section Appendix: FAQ.
E	04/2022	Updated section: "Instantiate."
D	12/2021	Removed figures "Example of Zoom In" and "Example of Zoom Out."
C	08/2021	The following is a summary of the changes made in this revision: • Added note related to Port or Net or Instance naming rules in the "SmartDesign User Actions" section. • Updated "VHDL Construct Support in SmartDesign."
B	04/2021	Updated "How do I make manual connections?" section with information related to the Connection Tool.
A	11/2020	Document converted to Microchip template. Initial Revision.

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

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

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-1256-5

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.