

Allegro® X System-Level Design Methodology Guide

Product Version 23.1
September 2023

© 2023 Cadence Design Systems, Inc. All rights reserved.

Portions © Apache Software Foundation, Sun Microsystems, Free Software Foundation, Inc., Regents of the University of California, Massachusetts Institute of Technology, University of Florida . Used by permission. Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Allegro System Design Authoring tool contain technology licensed from, and copyrighted by: Apache Software Foundation, 1901 Munsey Drive Forest Hill, MD 21050, USA © 2000-2005, Apache Software Foundation. Sun Microsystems, 4150 Network Circle, Santa Clara, CA 95054 USA © 1994-2007, Sun Microsystems, Inc. Free Software Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA © 1989, 1991, Free Software Foundation, Inc. Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, © 2001, Regents of the University of California. Daniel Stenberg, © 1996 - 2006, Daniel Stenberg. UMFPACK © 2005, Timothy A. Davis, University of Florida, (davis@cise.ulv.edu). Ken Martin, Will Schroeder, Bill Lorensen © 1993-2002, Ken Martin, Will Schroeder, Bill Lorensen. Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts, USA © 2003, the Board of Trustees of Massachusetts Institute of Technology. All rights reserved.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission. All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information. Cadence is committed to using respectful language in our code and communications. We are also active in the removal and/or replacement of inappropriate language from existing content. This product documentation may however contain material that is no longer considered appropriate but still reflects long-standing industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

1

Designing PCB Systems with Allegro X System Capture	5
<u>System-Level Designing</u>	5
<u>Benefits of PCB System Design</u>	6
<u>Supported Flows of System Design in System Capture</u>	7
<u>System-Level Design Terminology</u>	8
<u>License Requirements for System Design</u>	8
<u>Setting up the Workshop Database</u>	10
<u>Launching System Capture</u>	12
<u>Setting up Color Coding for Matching Nets</u>	14
<u>System Design-Specific UI Additional Features</u>	14

2

Designing a System using Functional Blocks	21
<u>Top-Down Design Flow Diagram</u>	22
<u>Creating a New Project</u>	23
<u>Adding Functional Blocks</u>	24
<u>Creating Subsystems</u>	38
<u>Creating Physical Boards for the Subsystems</u>	41
<u>Printing System-Level Designs</u>	44

3

Building a System using Multi-Board Connectivity	47
<u>Flow Diagram</u>	48
<u>Creating a New Project</u>	49
<u>Adding Existing Designs</u>	49
<u>Creating System-Level Design</u>	53
<u>Adding Connectivity Between Subsystems</u>	61
<u>Verifying NetGroup Objects and Connectivity</u>	63
<u>Generating Connectivity Reports</u>	67

Allegro X System-Level Design Methodology Guide

<u>System Connectivity Report</u>	67
<u>End-to-End Connectivity Report</u>	69
 <u>4 Updating Subsystems in a Design</u>	
	73

Designing PCB Systems with Allegro X System Capture

In the context of PCB system design, a system is any electronic hardware system. That can range from a large, complete system to a chip inside an end-user product. Creating any electronic system involves handling exact specifications as well as formalizing the abstract ideas in a designer's mind. In most cases, such as a new model of a cellphone, music player, or any household product, many existing components are used along with new components specifically designed just for that product. In most of these systems, the hardware and physical requirements are known and standard. The missing pieces are filled up as the design and planning of a new electronic product progresses.

This reality of designing is in contrast with the way traditional schematic applications work. The one board per design paradigm falls short and after the boards are created and ready for manufacturing, reusing them with other logical designs, or schematics, is not an easy task. All this while chasing aggressive time to market deadlines.

System-Level Designing

System design involves building an electronic hardware system, from conceptual ideas, through implementation to manufacture. It combines the logical design components and the physical design parts in one application.

- System Capture aims at enabling the design teams to put together parts of a familiar system, create logical parts of the design, and distribute those parts to individual teams of design experts. So, system designing is part of the design process right from the block diagrams, where the architects put down their initial thoughts using graphical tools, experiment with the way things are placed, add whatever is known at that time, and add information as the design progresses.
- The other requirement that System Capture addresses is to have a way to consolidate existing parts, or subsystems that will plug into the larger system. With system design, these subsystems are connected and a final product is ready for manufacturing.

Benefits of PCB System Design

A system usually has multiple fabrics. These are developed in isolation. Integrating these fabrics, or boards to complete the system can lead to interconnection issues. Forcing the designers to go back and redo the design results in loss of time and early mover advantage.

System architectures can be:

- Standard form factors with pre-defined form factors and protocols
- Distributed systems with different subsystems

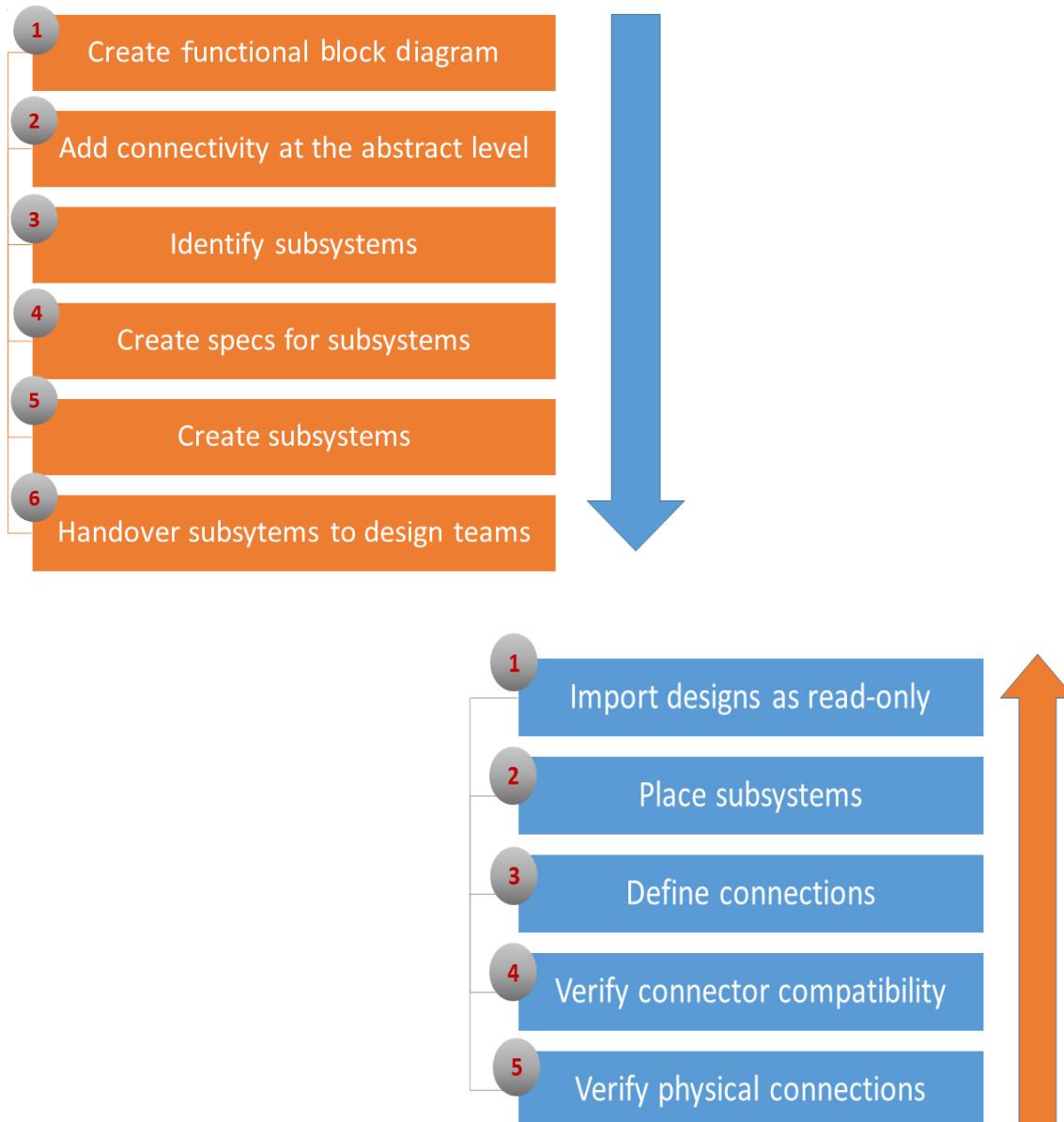
These subsystems can have common functionality across models, but each might require unique physical implementation

The benefits of using system design over traditional design methods are:

- Support for multiple boards
- Parallel designing
- Reuse of existing parts or boards
- Assembling a new system with interconnected subsystems
- Easy navigation across design hierarchies

System design enables designs to be electrically and physically connected while managing the pins and ports connectivity.

Supported Flows of System Design in System Capture



System-Level Design Terminology

The commonly used system-level design terms and their descriptions are listed in the following table:

Term	Description
System	An electronic hardware product which has electronic connectivity and hardware.
Subsystem	Each component of a system is a subsystem.
System design	A geometric layout which defines the product subsystems and components at an abstract level. Specs are created for each subsystem and connectivity information is captured in a system design.
Top-down design	A process flow diagram which captures the high-level abstracted design architecture.
Bottom-up design	A design which combines multiple designs into an electronic product.

License Requirements for System Design

Complete system design functionality is available with the following enterprise licenses:

- Allegro PCB Venture
- Allegro X EE
- Allegro X Designer
- Allegro X Venture
- Allegro System Capture Venture
- Allegro Design Entry HDL XL (Concept HDL Expert)
- Allegro System Architect GXL
- Allegro Managed Library Authoring
- Allegro Enterprise Authoring Solution
- Allegro Enterprise System Design Authoring
- Allegro Enterprise PCB Designer Suite

Allegro X System-Level Design Methodology Guide

Designing PCB Systems with Allegro X System Capture

Limited system-design features are available with the following licenses:

- Allegro PCB Designer (Schematic)
- Allegro X Artist
- Allegro Design Authoring
- Allegro System Capture Designer

With these licenses, you can:

- Create designs and add:
 - Unlimited number of functional blocks
 - Maximum three logical boards
- Open designs created with higher licenses that contain:
 - A maximum of three logical boards
 - Any number of functional blocks
- Export a subsystems project only for designs with a maximum of one default and two more subsystems.

Setting up the Workshop Database

This document provides hands-on exercises to create system-level designs using Allegro X System Capture. The flows present suggested ways of working with the tools. It is recommended that you follow the steps in the sequence presented in the workshop documentation to obtain an understanding.

Prerequisites

Before you begin, perform the following prerequisite tasks described in this section:

- [Extracting the Supporting Files](#)
- [Setting up Environment Variables](#)

Extracting the Supporting Files

1. Navigate to the location of the tutorial database:

`<installation_directory>\doc\scap_flow\tutorial_database`

For example, on a Windows computer:

`C:\Cadence\SPB_23.1\doc\scap_flow\tutorial_database\`

2. Depending on the operating system, extract one of the following files:

- [tutorial_database.zip](#) on Windows
- [tutorial_database.t.Z](#) on Linux

The following folder structure is created:

- SLD_SITE**
Contains the libraries and site-level information.
- Multi-board_system**
Contains the sample designs and data used later in this document.

This location has files required for **CDS_SITE** as well as folders that will be used for various tasks covered in this document.

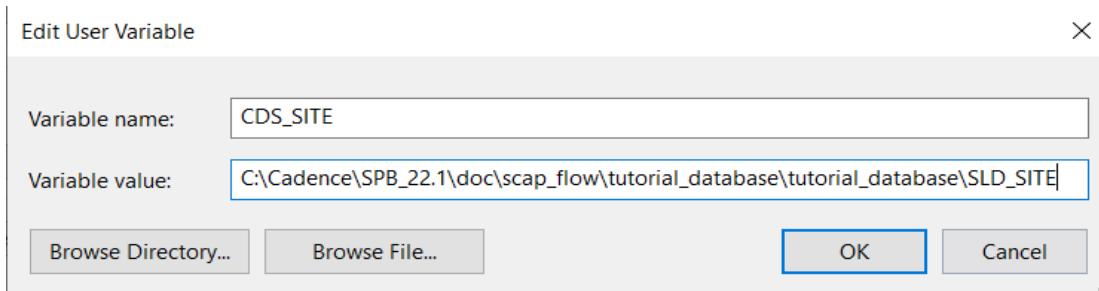
3. Record the location where the files are extracted.

Setting up Environment Variables

Add the following system variables:

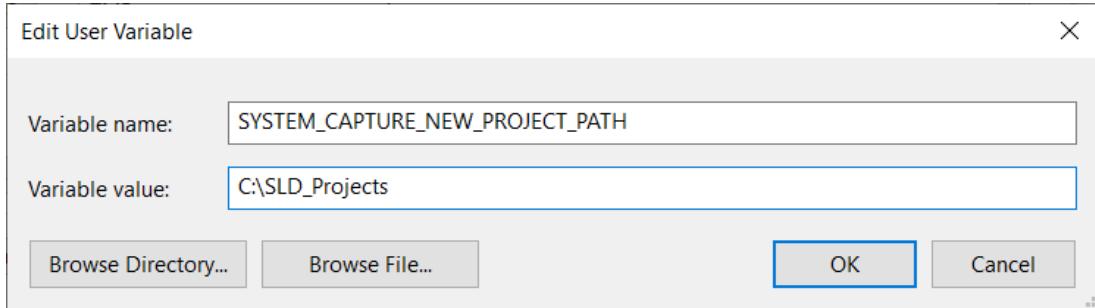
- CDS_SITE

Point to the SLD_SITE folder extracted in the previous section.



- SYSTEM_CAPTURE_NEW_PROJECT_PATH

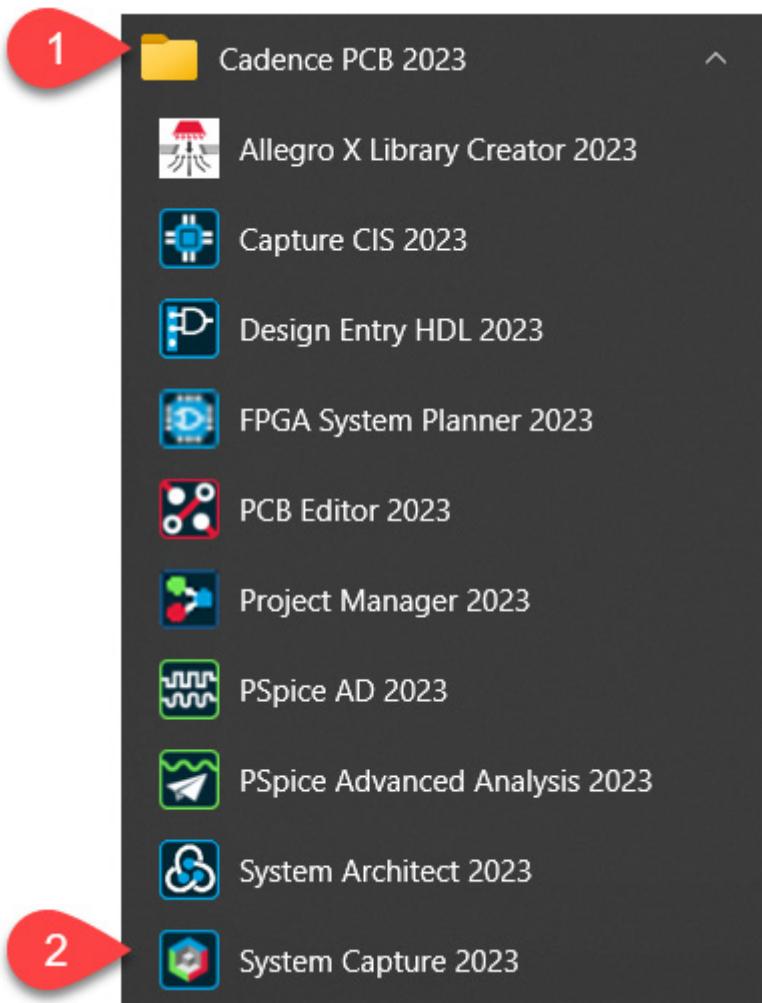
Point to a folder where all the System Capture projects that you create are saved. For example, C :\SLD_Projects.



Ensure that the folders are local directories where you have the write permission.

Launching System Capture

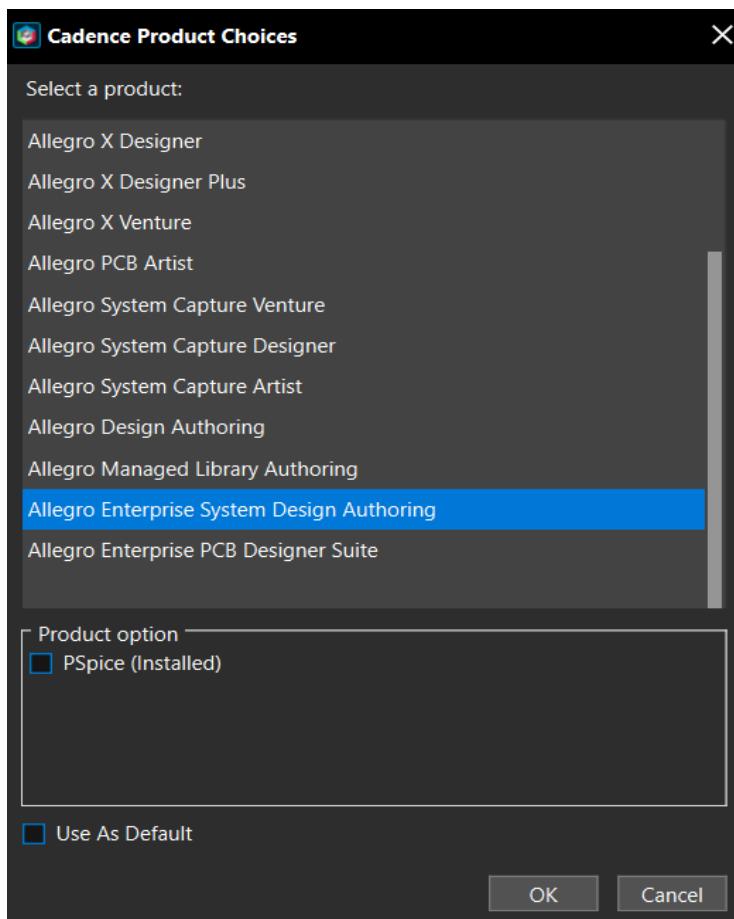
1. Start System Capture.



Allegro X System-Level Design Methodology Guide

Designing PCB Systems with Allegro X System Capture

When you launch Allegro X System Capture, the *Cadence Product Choices* license selection dialog box is displayed.



2. Choose a system design authoring license.
3. Click *OK*.

System Capture starts.

Setting up Color Coding for Matching Nets

To help quickly identify the nets that are perfect matches or partial matches, you can enable the color coding by setting these directives:

Directive	Value	Description
PIN_ASSIGNMENT_DIALOG_SHOW_CO LORED_NET_MISMATCH	Yes	Enables the feature
PIN_ASSIGNMENT_DIALOG_MINIMUM _CHARACTER_MATCH_COUNT	<number>	The number of characters that should match, such as 3

The next section lists the system design-specific changes in the System Capture user interface.

System Design-Specific UI Additional Features

With a system design license, many additional features are included in the System Capture user interface. This section lists these features and describes what they are used for.

View Menu

The following two options are added to the *View* menu which are displayed when a system block or *Subsystem* (which means a logical board or functional block) is added to a design with an authoring license in use:

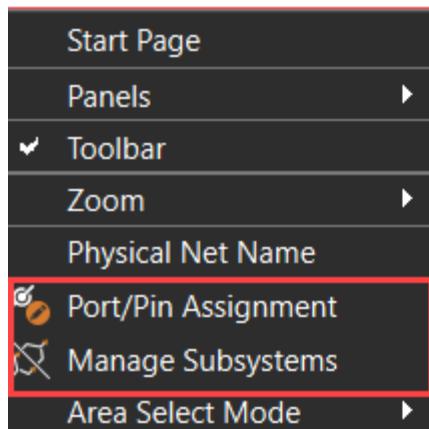
- Port/Pin Assignment*
- Manage Subsystems*

Allegro X System-Level Design Methodology Guide

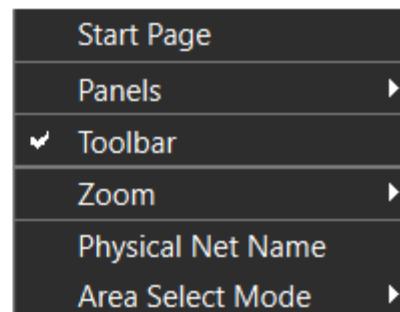
Designing PCB Systems with Allegro X System Capture

These options are used and explained later in [Building a System using Multi-Board Connectivity](#).

With system block

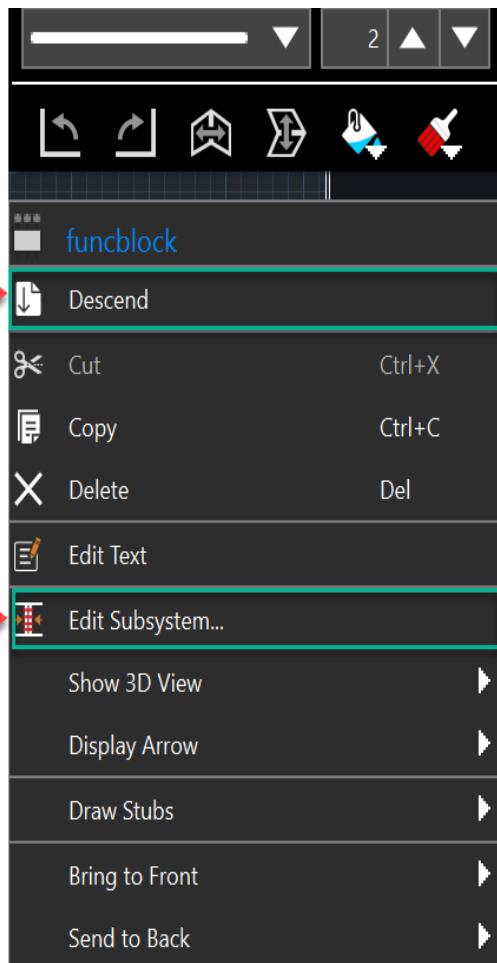


Without system block



Shortcut Menu

For system design blocks, the following options are added to the shortcut menu when you right-click a system block:

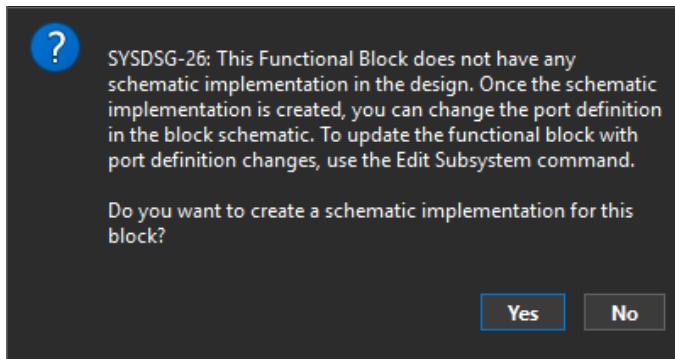


■ *Descend*

Allegro X System-Level Design Methodology Guide

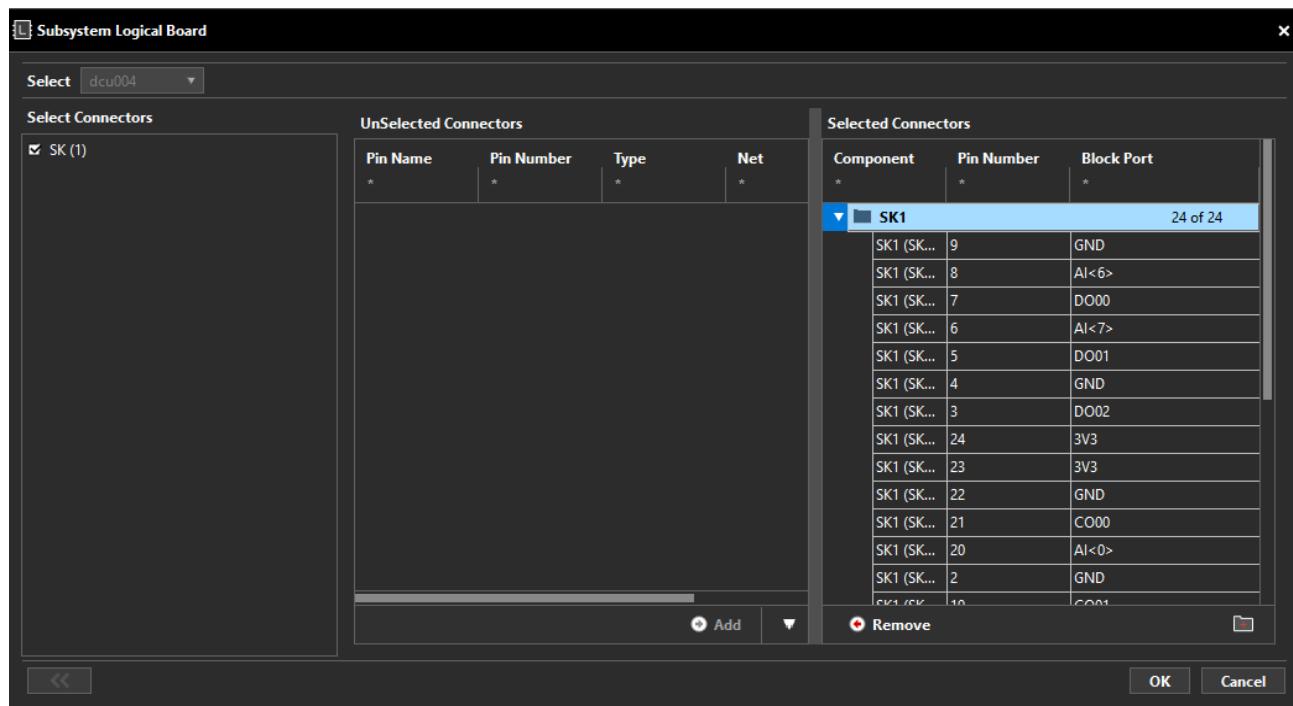
Designing PCB Systems with Allegro X System Capture

Opens the child block. In case there is no child block, on choosing *Descend* you are prompted if you need to create a schematic implementation.



■ *Edit Subsystem*

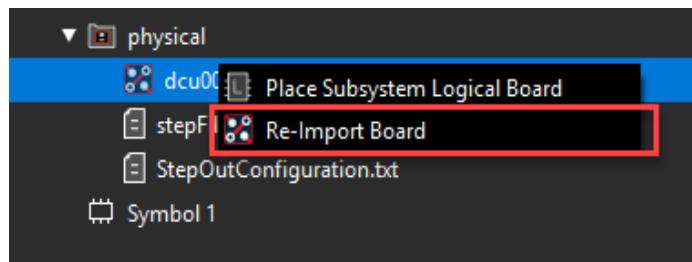
Displays the dialog box used to edit the subsystem, which specifies the connector details.



This is explained in [Designing a System using Functional Blocks](#).

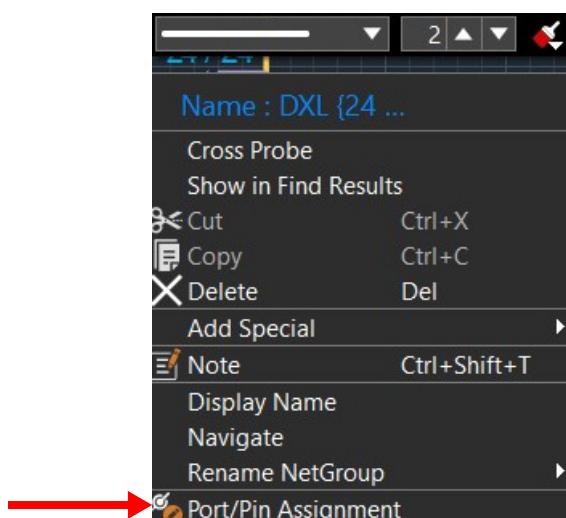
Re-Import Board Menu Command

After importing a logical board, right-click it in the *Project Explorer*. This shows an option for re-importing the logical board.



Port/Pin Assignment

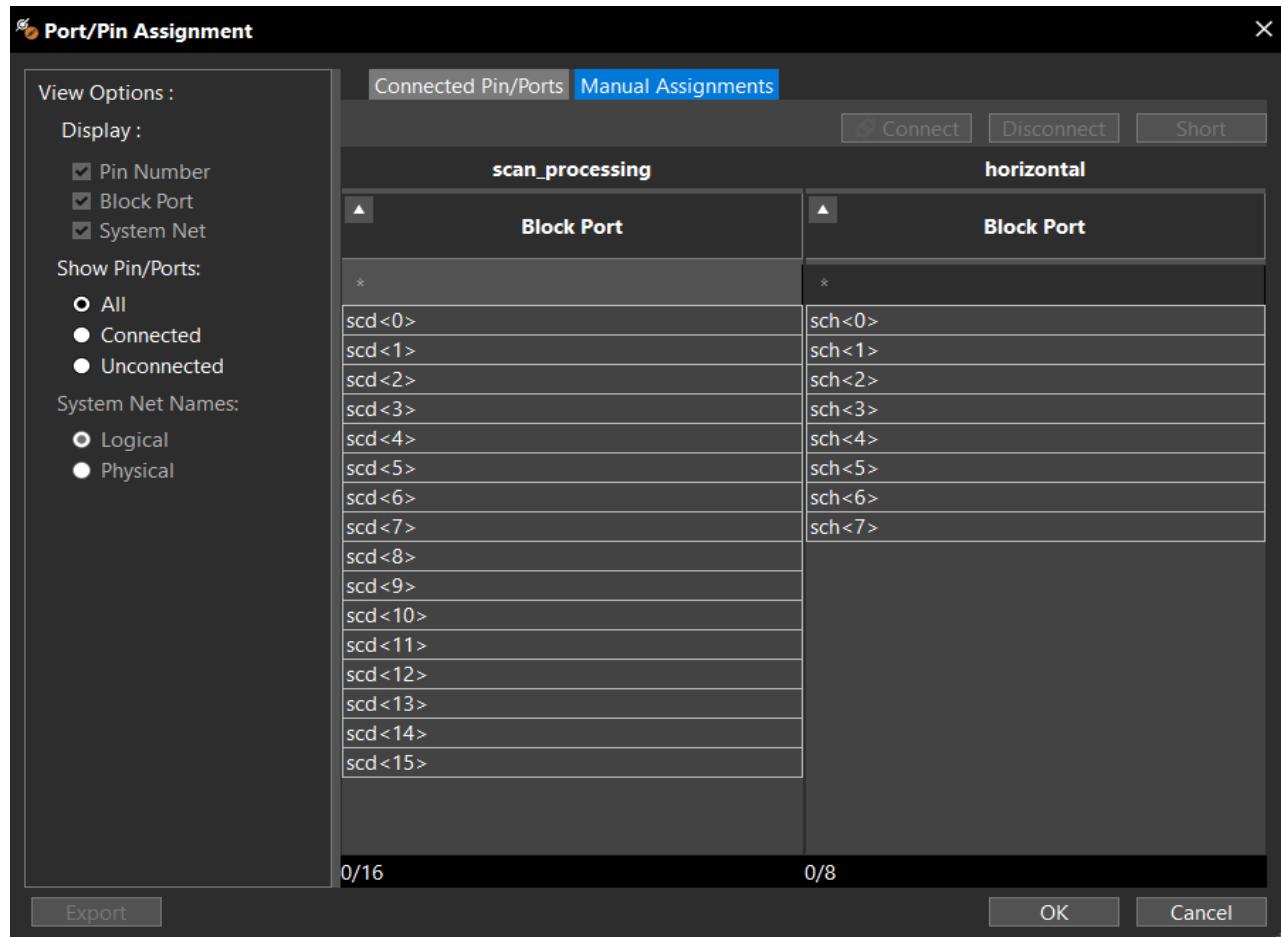
Right-clicking a NetGroup, shows the *Port/Pin Assignment* option:



The specific connections across pins and blocks are specified in the *Port/Pin Assignment* dialog box.

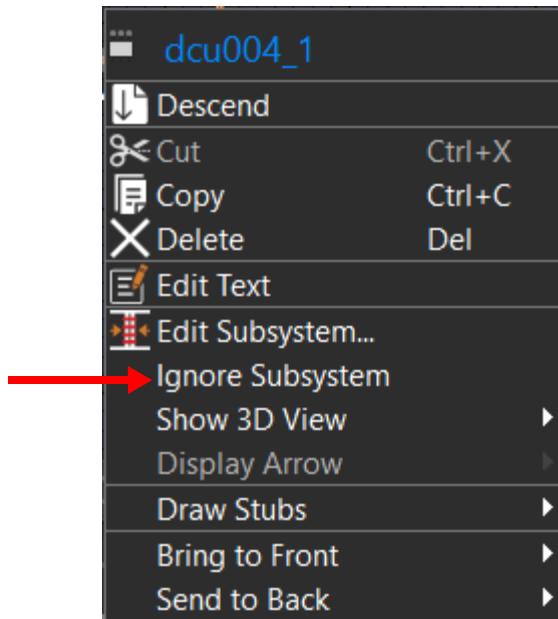
Allegro X System-Level Design Methodology Guide

Designing PCB Systems with Allegro X System Capture



Ability to Ignore Logical Boards

To exclude logical boards from navigation, *Port/Pin Assignment* dialog box, and System Connectivity report, right-click the logical board and select *Ignore Subsystem*.



The SLD_BLOCK_IGNORE property gets applied to blocks when this option is selected.

Designing a System using Functional Blocks

Diagram and graphics creation tools, such as Visio or PowerPoint are commonly used to define and develop an electronic hardware system. Typically, a diagram created with these tools is used for visual reference and remains independent of design data or tools. Allegro X System Capture provides equivalent functionality to enhance Electronic Hardware System exploration and development in several areas including the following:

- Quick creation of a functional block diagram using drawing tools that matches the ease-of-use in Visio and PowerPoint.
- Graphical display of data transfer between different functions within a system
- Addition of design intent through notes, tables, and graphics
- Establishing interconnectivity between functional blocks using ports and netgroup objects
- Assigning and managing a single or multiple functions to subsystems
- Exporting subsystem design projects with interconnectivity details for the Electronic Engineering teams to implement both logical (schematic) and physical (layout) designs

This chapter provides an overview of the tasks involved in creating a new system in System Capture using a top-down approach in which you create functional blocks and subsystems. It is not intended to cover the process of inheriting detailed design information in a bottom-up flow. As you will see, in this flow, the designer starts with a block diagram of the logical components and, at the end, has a set of functional, physical subsystems in the form of boards.

The product architecture is captured as a geometric diagram. In other words, the ideas visualized by an architect are drawn and saved within the schematic-capture software and not in an external flowchart application, or white board.

Allegro X System-Level Design Methodology Guide

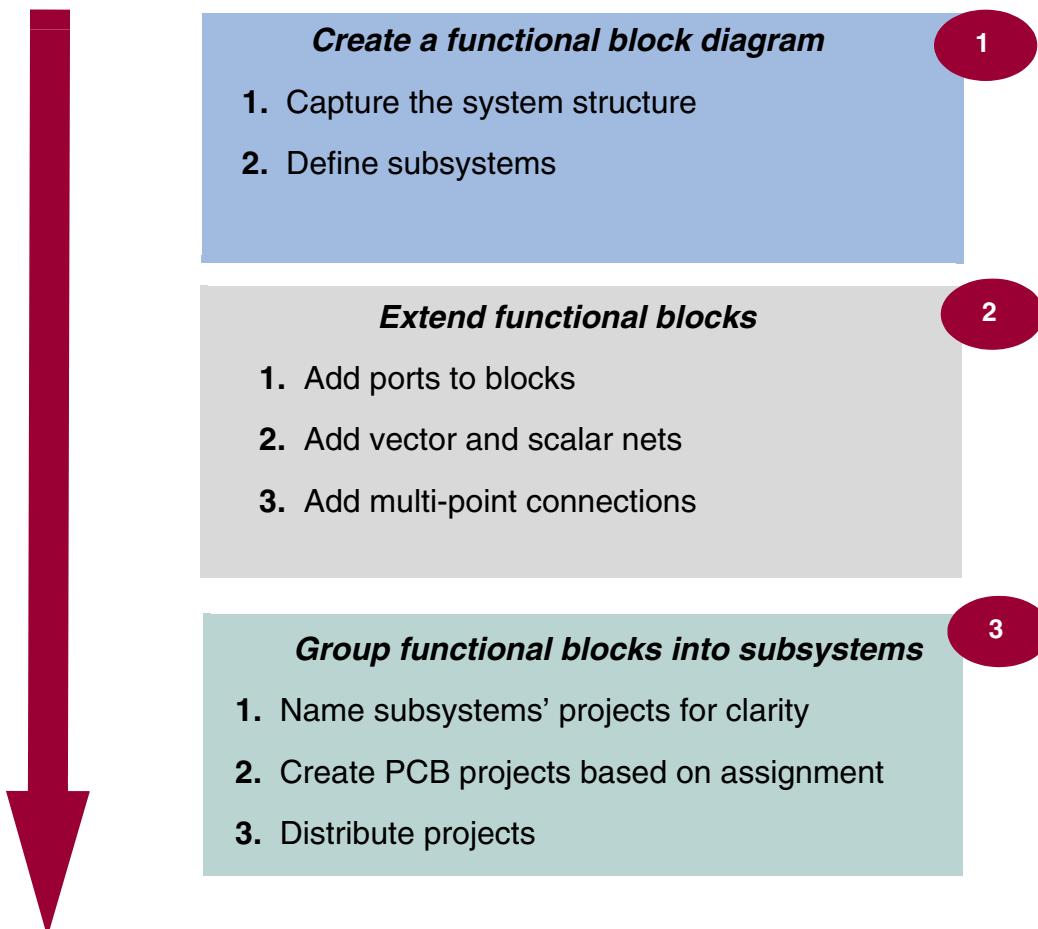
Designing a System using Functional Blocks

In this flow, engineers start with high-level abstracted design architecture and drill-down into the specifics of the blocks and interconnectivity. Therefore, this is also called the top-down design approach.

These top-down design flow tasks are commonly performed by a system architect or senior hardware engineer when defining an electronic hardware system structure:

- [Creating a New Project](#)
- [Adding Functional Blocks](#)
- [Creating Subsystems](#)
- [Creating Physical Boards for the Subsystems](#)
- [Printing System-Level Designs](#)

Top-Down Design Flow Diagram

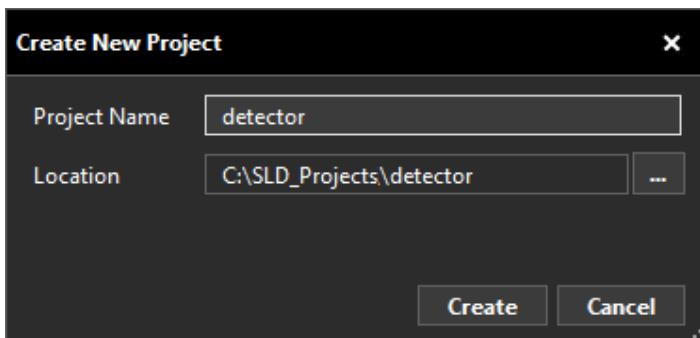


Creating a New Project

1. Start Allegro X System Capture.
2. Select a system design license.
3. Choose *New* in the *Start Page*.



4. Specify the project name and location.



5. Click *Create*.

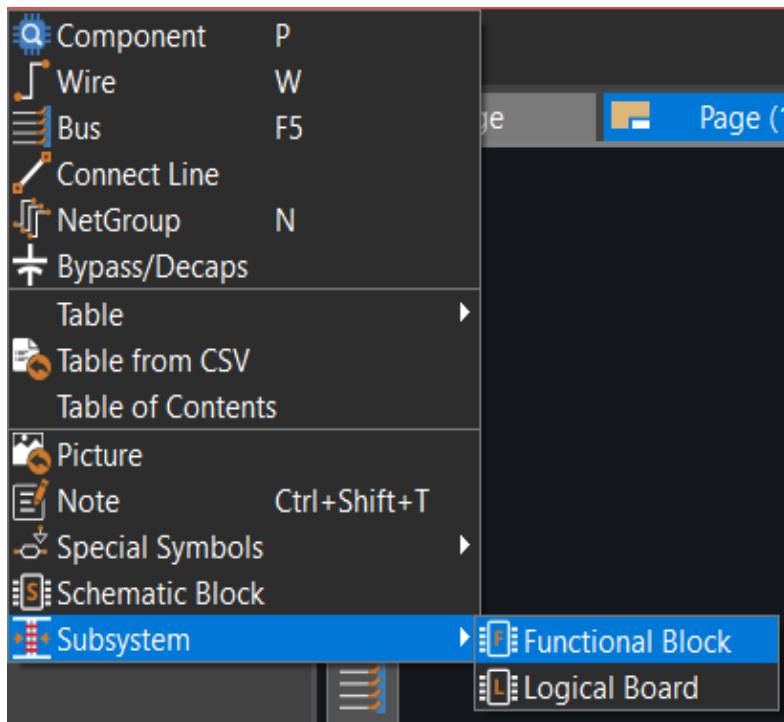
A blank canvas opens with an empty 'A' size page. You can now start creating the system design.



Ensure that the system architecture diagram is on the first page of the design.

Adding Functional Blocks

1. Choose *Place – Subsystem – Functional Block*.



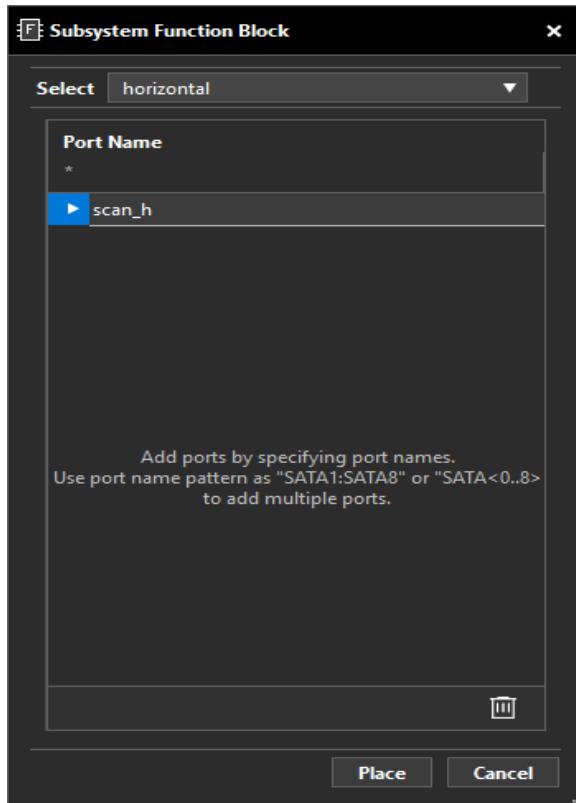
The *Subsystem Functional Block* dialog box is displayed.

2. Specify a name for the block. For example: *horizontal1*

Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

3. Enter a *Port Name* for the functional block. For example, *scan_h*



4. Click *Place*.
5. Place the block on the canvas.
6. Press the `Esc` key to end the command.

The functional block is placed on the canvas.

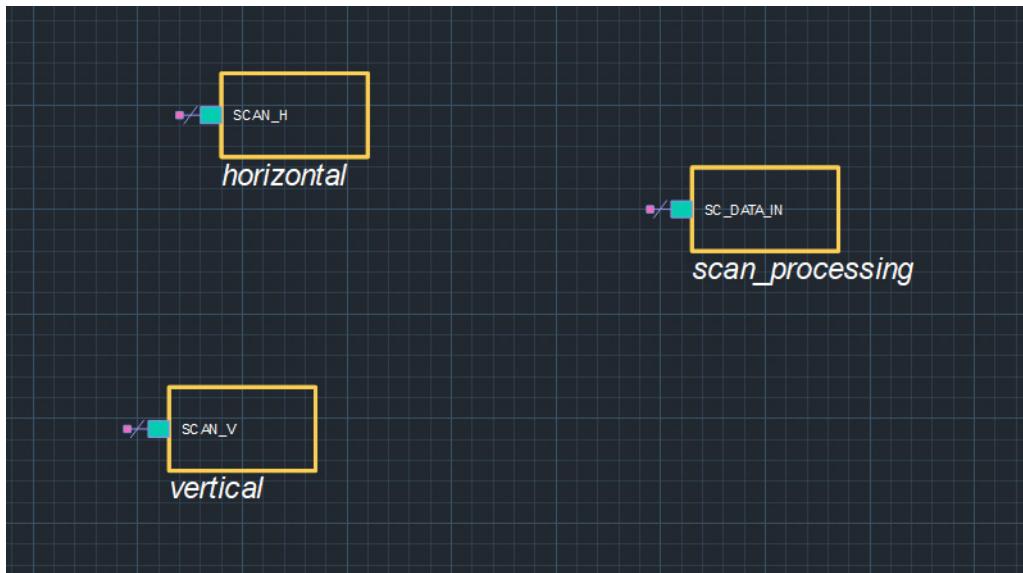


7. Repeat steps 1-6 and add the following blocks:
 - Block *vertical* with Port Name *scan_v*

Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

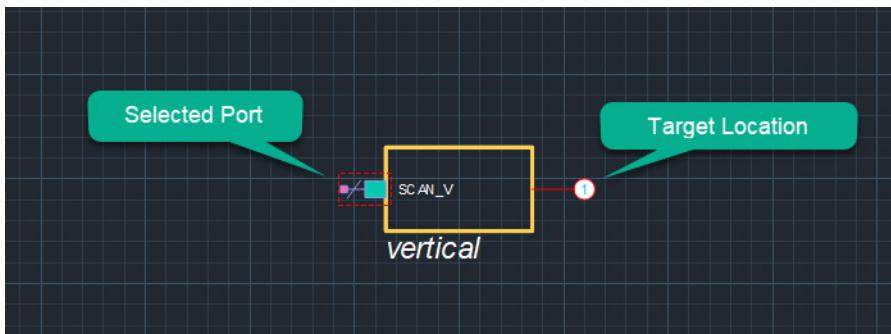
- Block *scan_processing* with Port Name *sc_data_in*



8. Arrange the blocks and port symbols as required.

The port symbols for *vertical* and *horizontal* will be moved to the opposite side of the block symbols.

- Click and drag the port symbols.

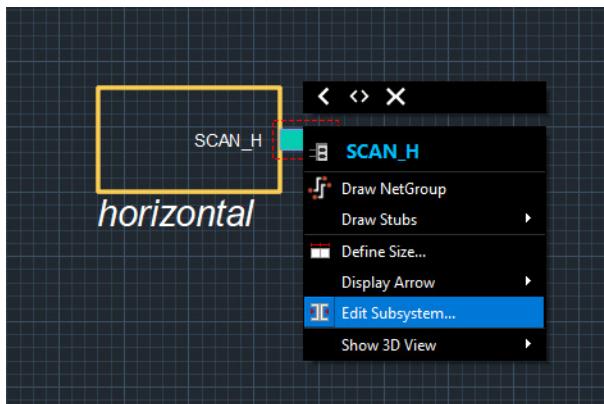


9. Add the port information for the blocks:

Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

- a. Right-click the *horizontal* block and choose *Edit Subsystem*.



- b. Add port details.

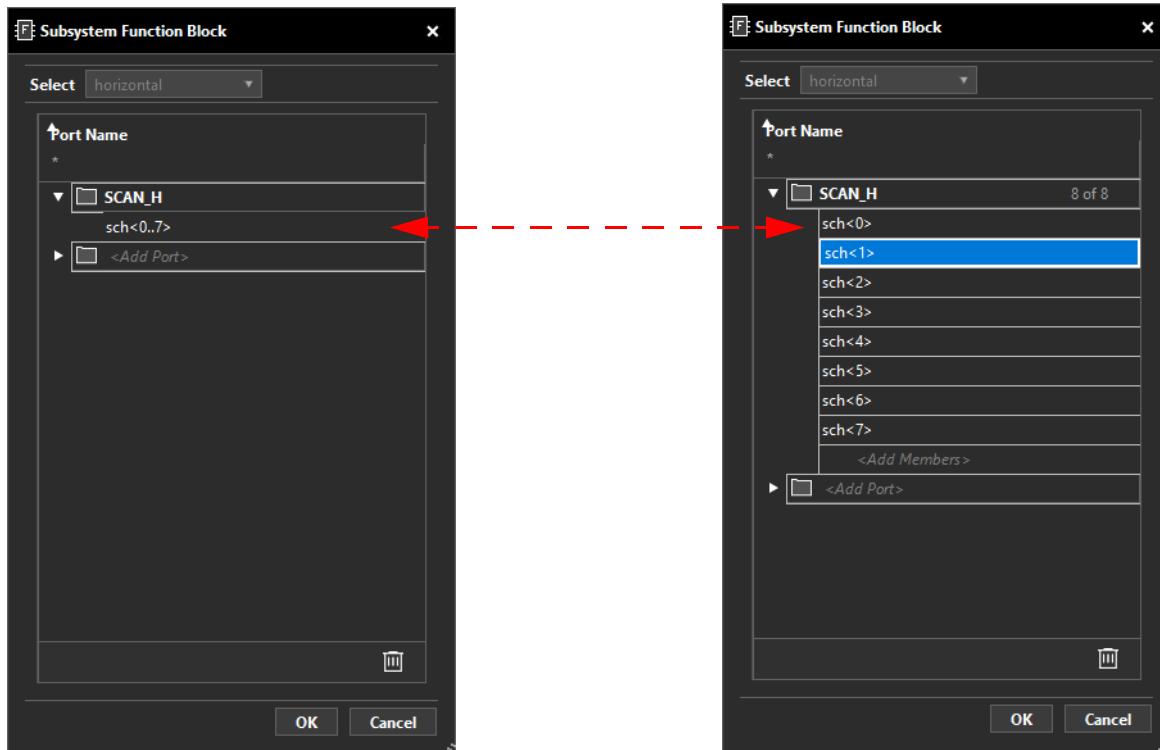
Ensure that the port, *SCAN_H*, is 8 bits wide.

10. In the *Subsystem Functional Block* dialog box, choose the *SCAN_H* port.
11. Enter `sch<0..7>` below the port name.
12. Press `Enter`.

Allegro X System-Level Design Methodology Guide

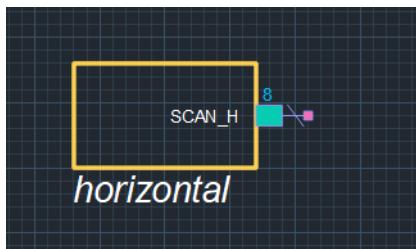
Designing a System using Functional Blocks

The port details expand to display each interconnected pin within the port group.



13. Click *OK*.

The block symbol gets updated to confirm the bit width.



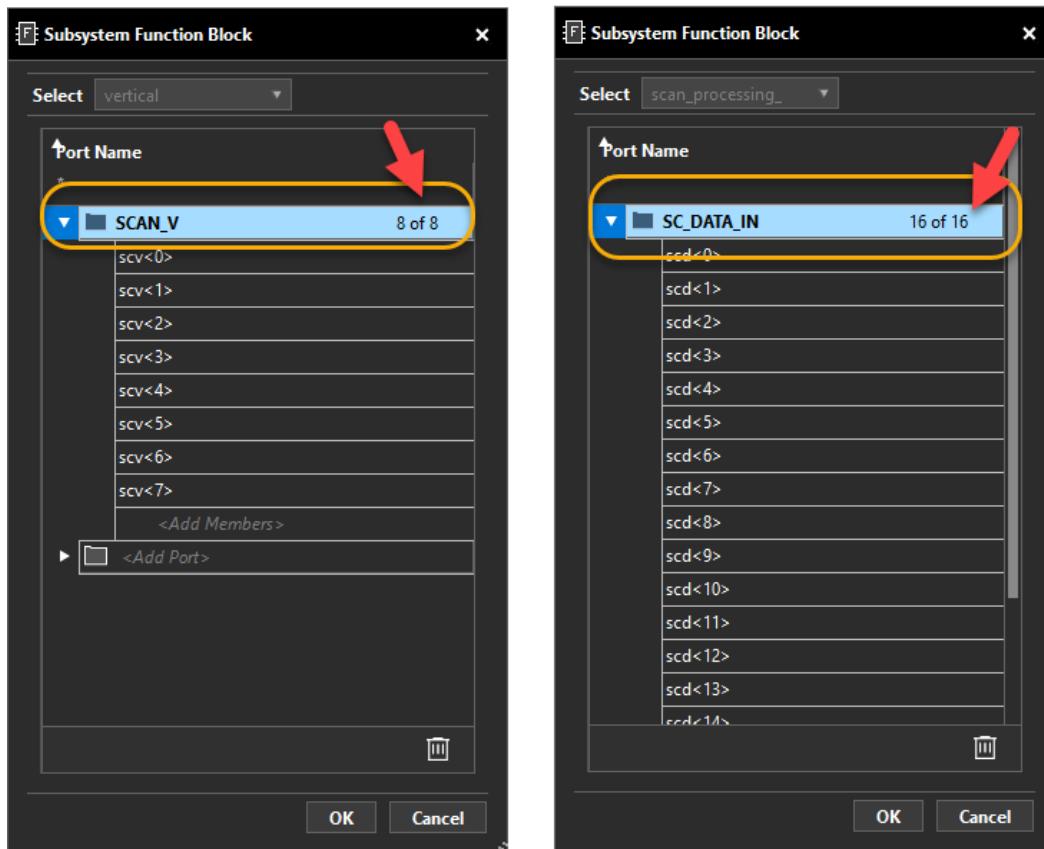
14. Add the port details for the following ports:

- Port *SCAN_V* - *scv<0..7>* (8 bits)
- Port *SC_DATA_IN* - *scd<0..15>* (16 bits)

Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

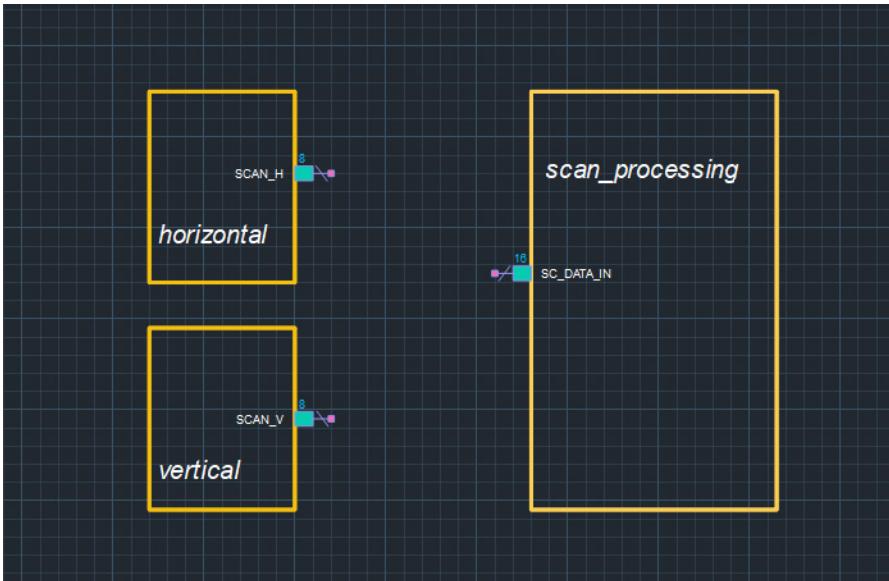
Here is how the blocks appear at this stage.



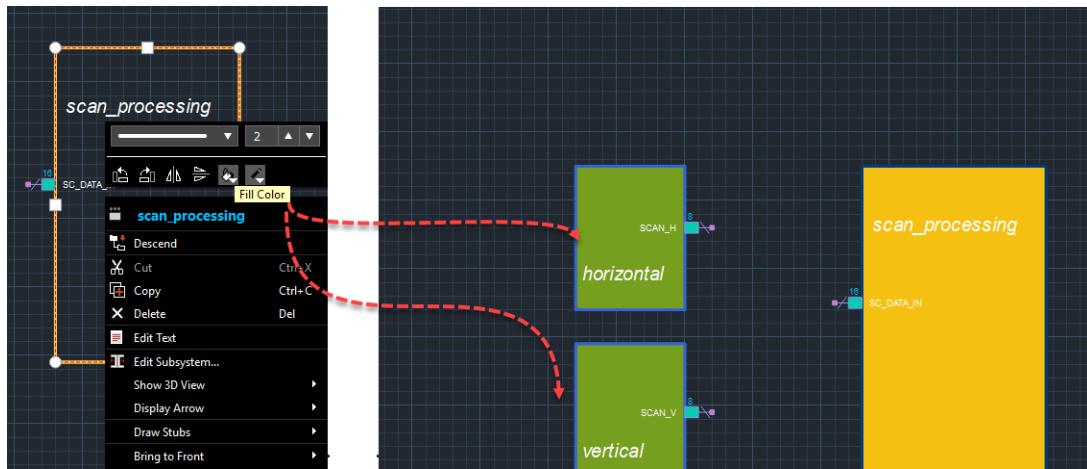
Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

Here is how the functional blocks appear. The sizes have been adjusted.



The context-menu has options for formatting these functional block symbols to change their appearance as required. Here is a sample:



With ports defined, high-level connectivity can now be added.

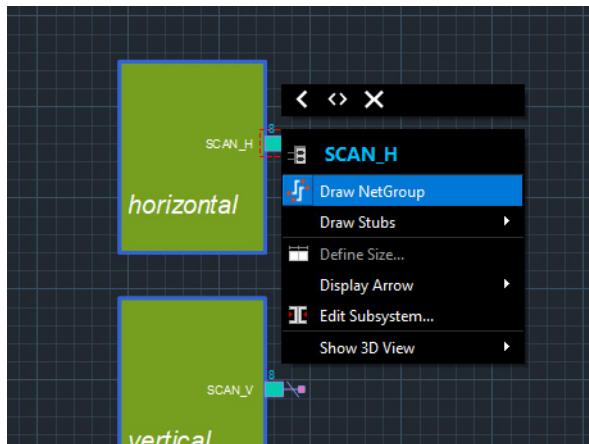
15. Add the high-level connectivity to the design.

Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

- Right-click the port on the `horizontal` block and choose *Draw NetGroup*.

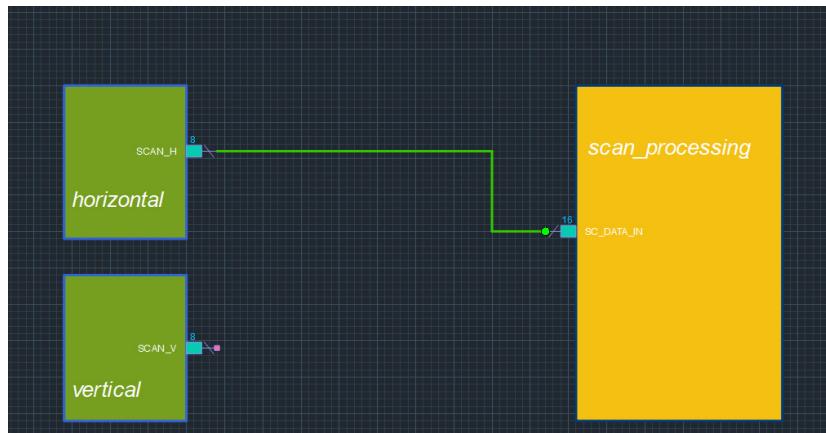
You can also click the NetGroup icon in the toolbar and click the port hotspot.



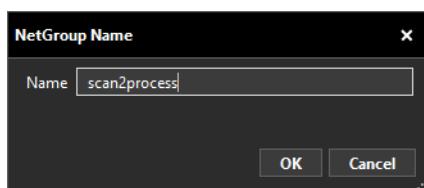
Important

A NetGroup name cannot start with `_N` or `_NG` because these prefixes are used by System Capture internally.

- Select the corresponding port on the `scan_processing` block to complete the connection.



- Enter a name for the NetGroup, `scan2process`.



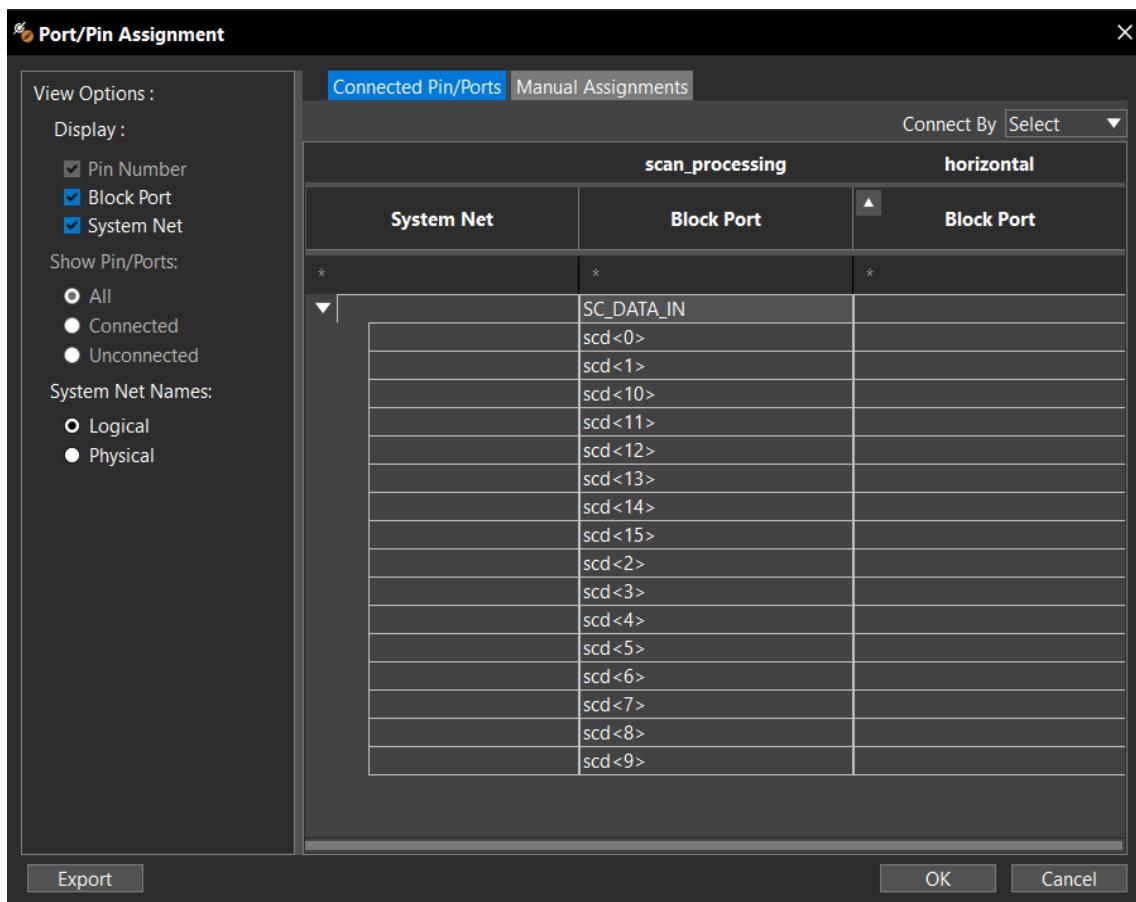
Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

After the NetGroup is added, the The *Port/Pin assignment* dialog box is displayed.

The *Connected Pin/Ports* tab displays any connections that could be made automatically. If a pattern between source and target pins is found, for example same width ports with matching pin numbers or signal names, System Capture makes the connections automatically.

As the bit names do not match, no connections have been made as visible in the *Connected Pin/Ports* tab.



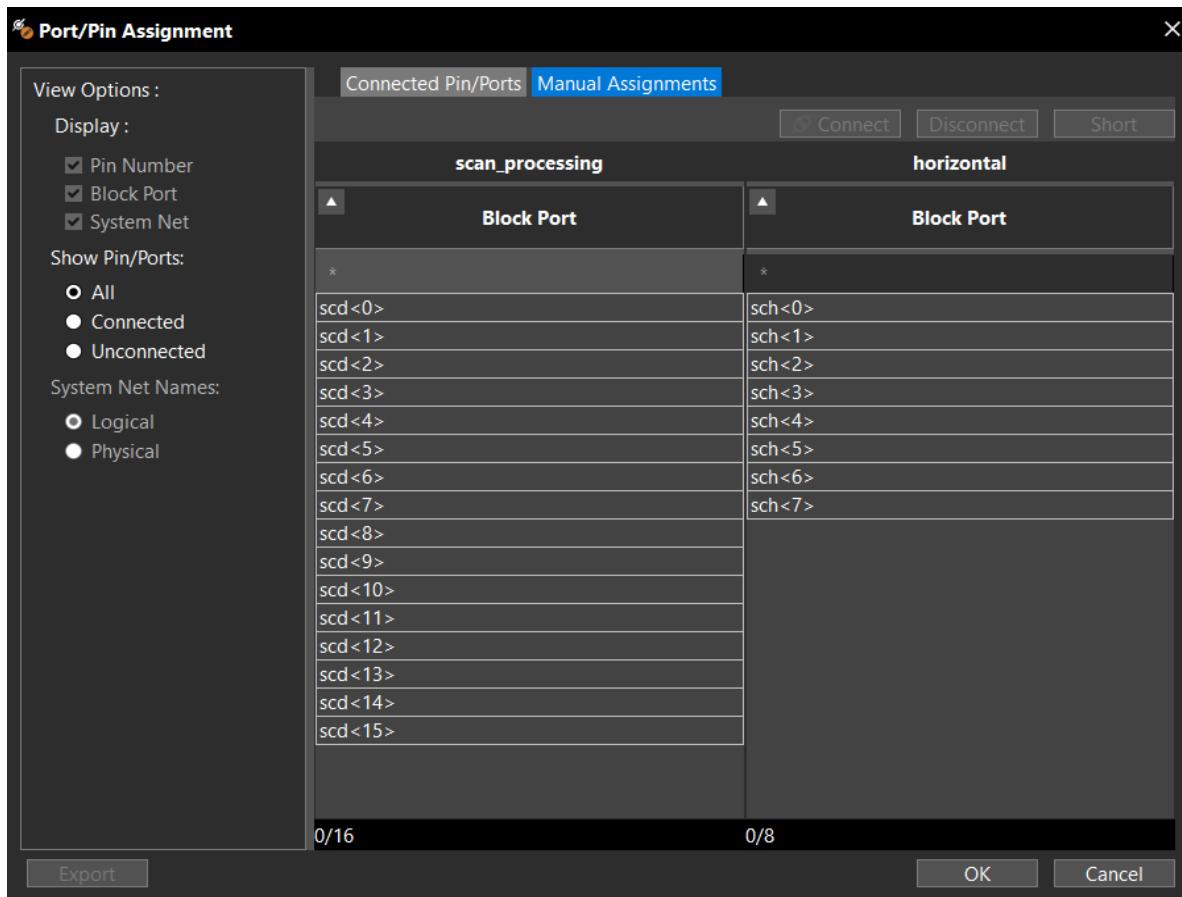
Connectivity will be added between two blocks in the next step.

16. Click the *Manual Assignments* tab.

Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

The available bits for both the blocks are listed.

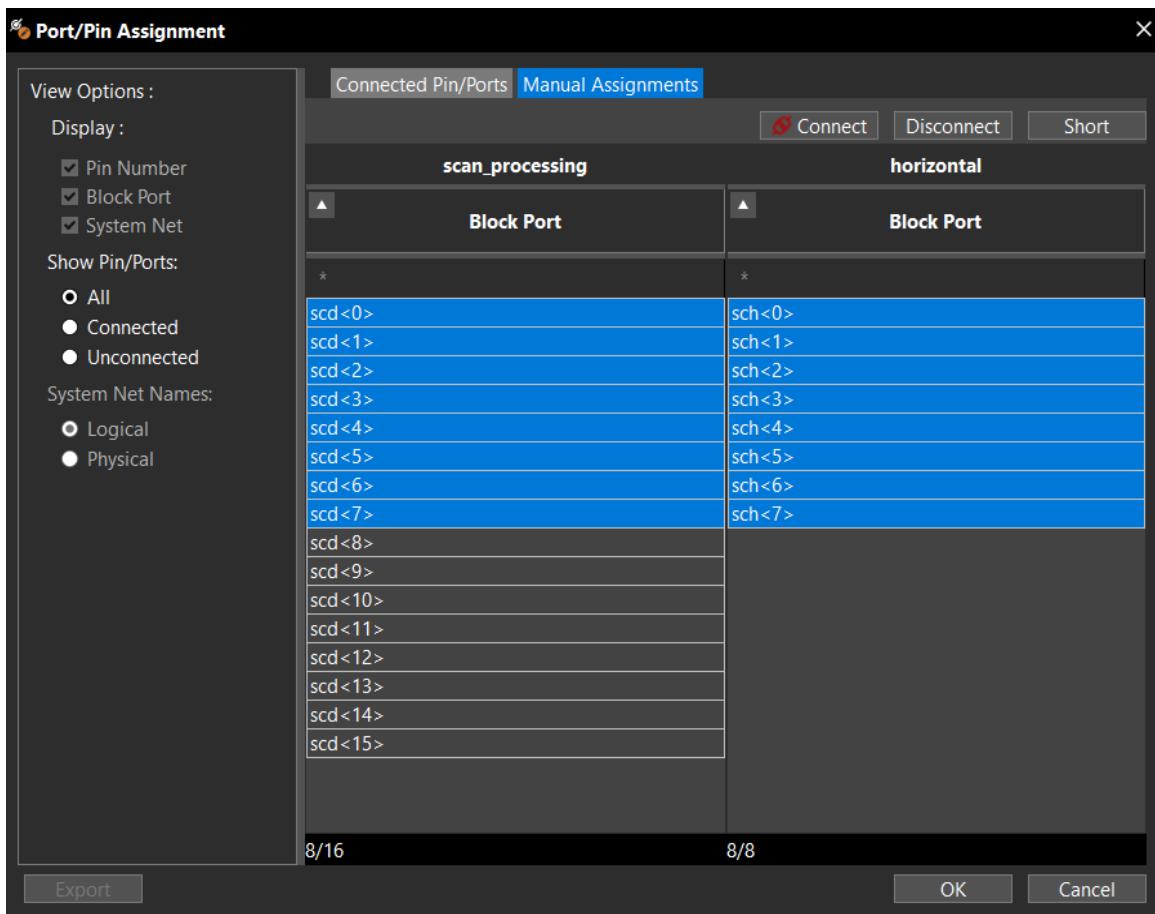


- a. Select the top eight bits of the *scan_processing* block, from *scd<0>* through *scd<7>*.

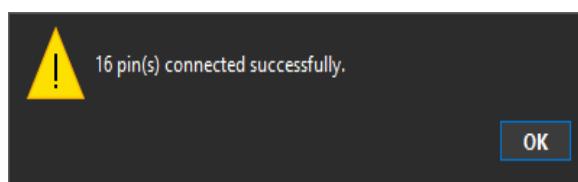
Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

- b. Select the eight bits of the *horizontal* block.



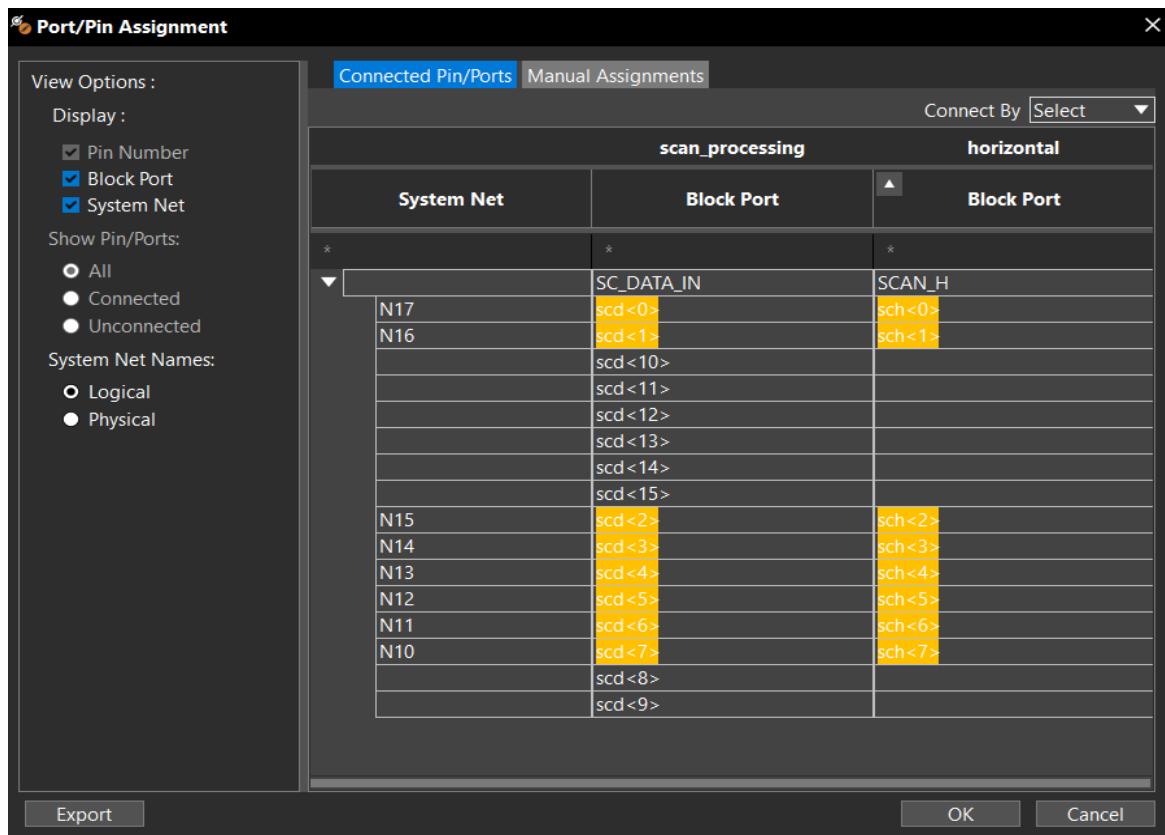
- c. Click *Connect*.
- d. Click *OK* on the confirmation message that the pins have been connected.



Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

- e. Click the *Connected Pin/Ports* tab.



- f. Review the results and click *OK*.

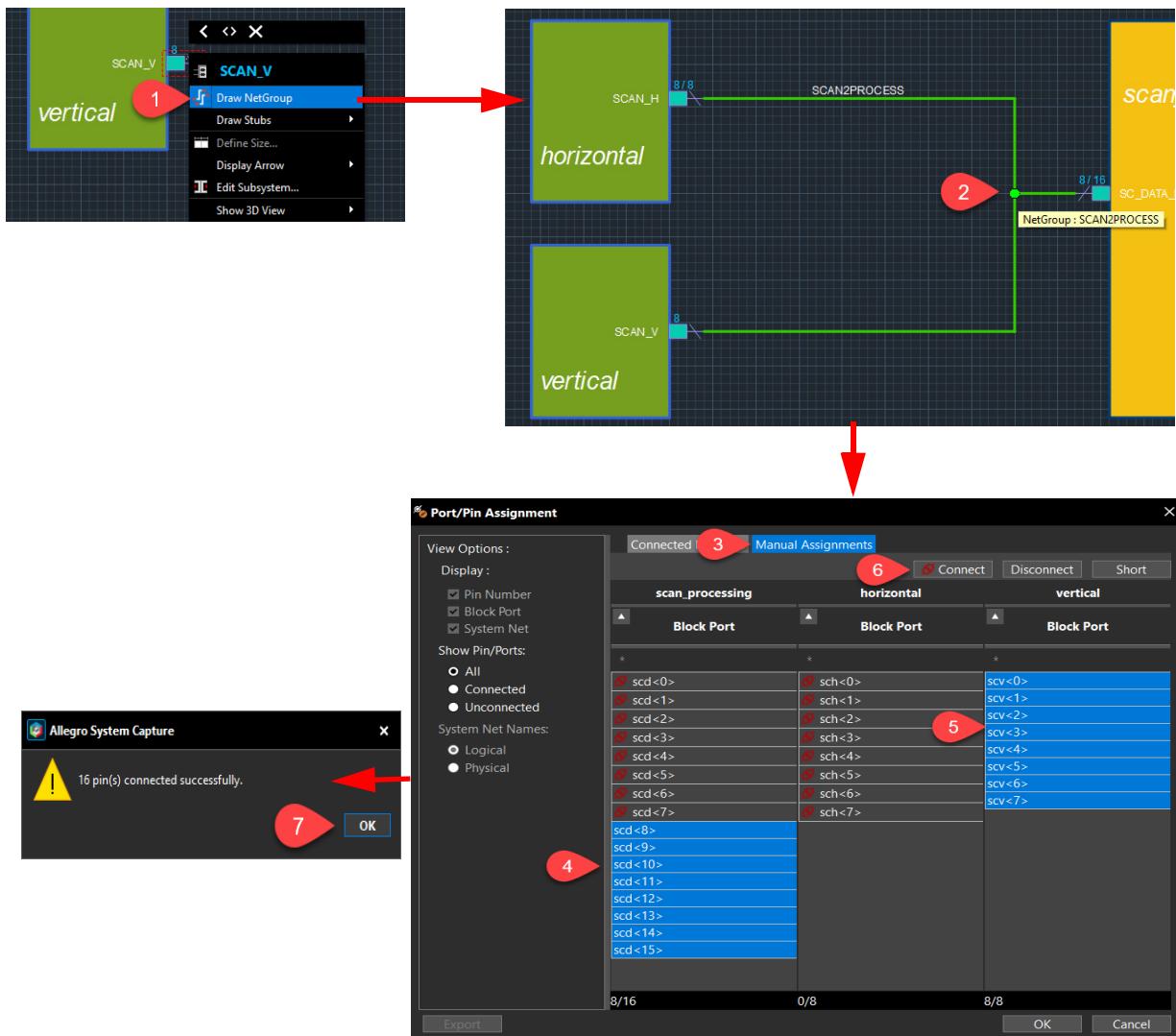
The canvas updates to show that eight pins of the block, *horizontal* are connected to eight bits of *scan_processing*.



Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

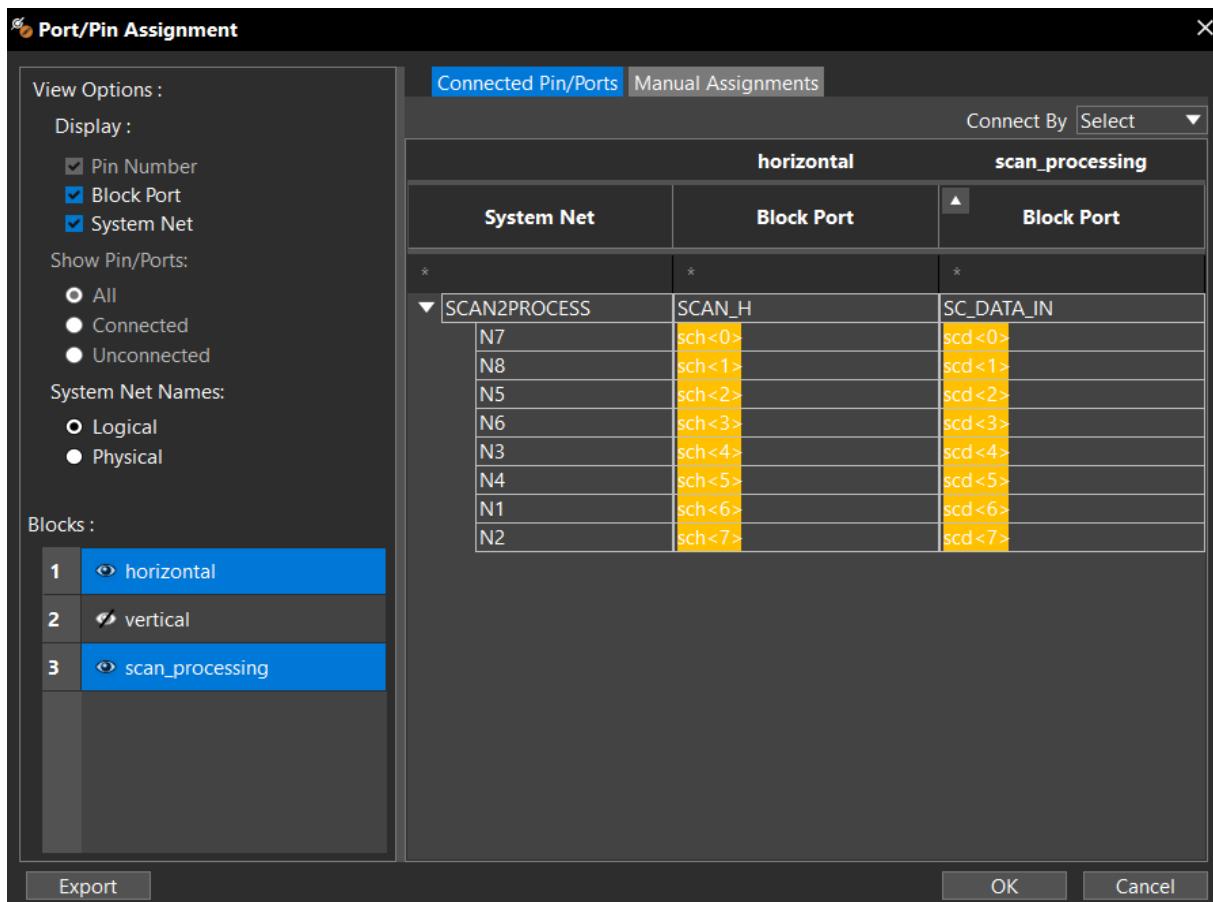
Similarly, connect the *vertical* block pins to the remaining 8 bits of the *scan_processing* block.



Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

Note: To verify the connectivity with color coding when more than two system blocks are connected, right-click on the canvas, and select *Port/Pin Assignment*. Deselect the vertical block by clicking the eye icon next to it to view the connectivity of other two functional blocks.



After establishing the connectivity, the next step is to define the physical subsystems within the system design. At this stage, you can consider how the system-level functions should be defined, based on the specific fabrics to be used.

Note: Color coding feature will work only when the directives given in [Chapter 1, “Setting up Color Coding for Matching Nets”](#) are set.

In this example, two boards will be created.

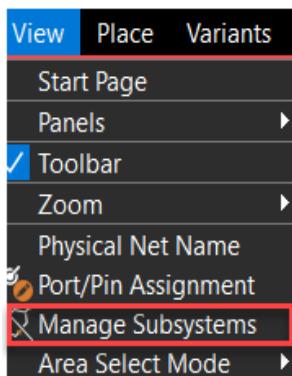
Related Topic

- [Creating Subsystems](#)

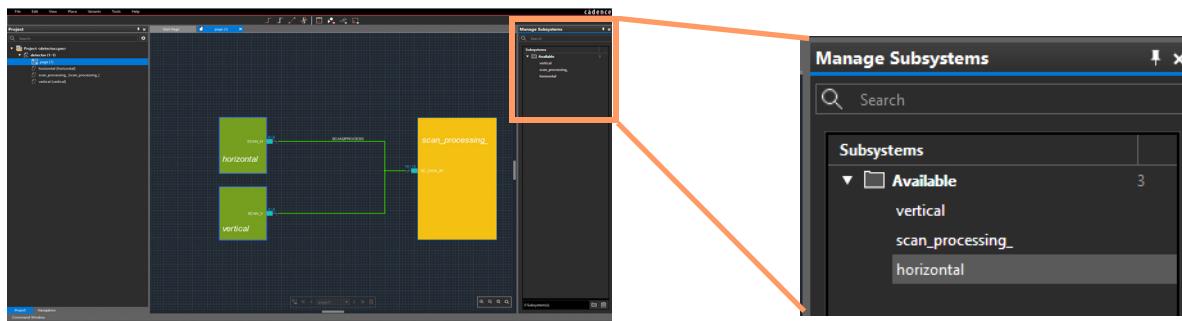
Creating Subsystems

After adding the blocks, specifying their ports, establishing the connectivity across components, you can divide the system design into subsystems.

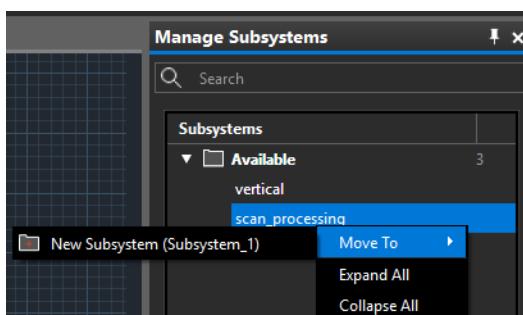
1. Choose *View – Manage Subsystems*.



The *Manage Subsystems* window opens to the right of the canvas.



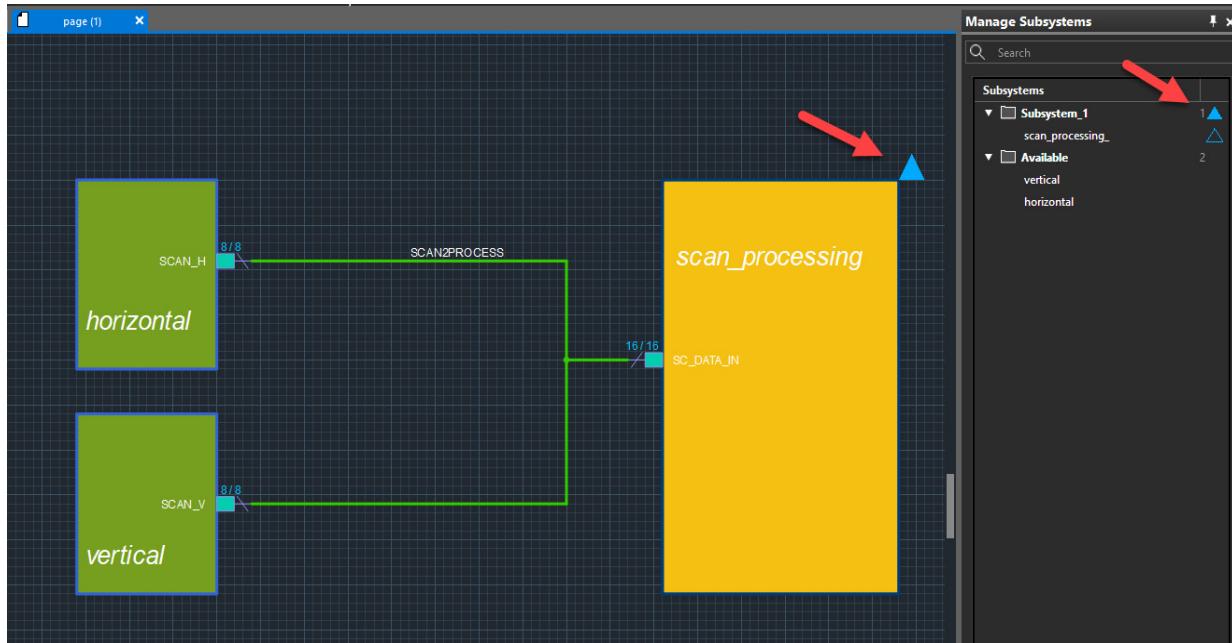
2. Right-click the `scan_processing` block and choose *Move To – New Subsystem (Subsystem_1)*.



Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

A new subsystem is displayed and it contains the `scan_processing` block. This is confirmed with a triangular marker both in the *Manage Subsystems* panel and on the canvas as illustrated in the following image:

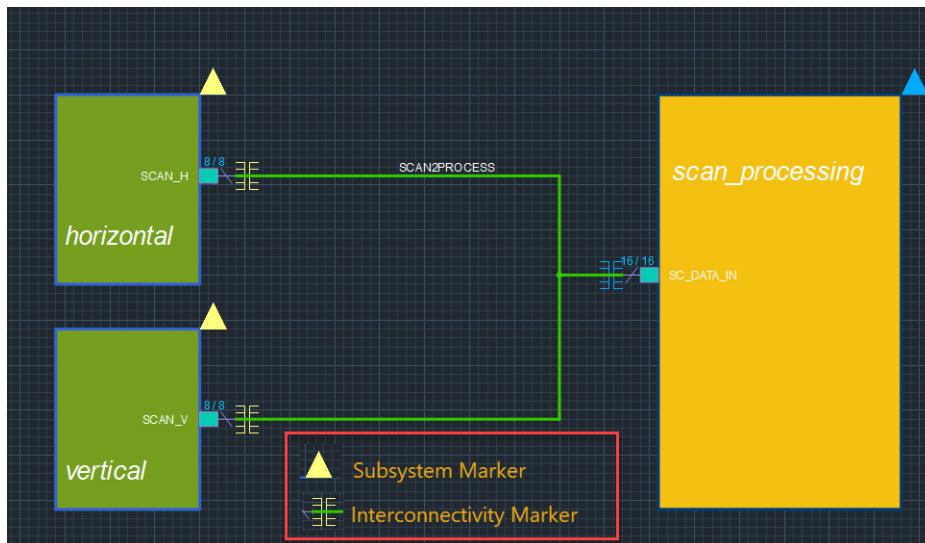


3. Click the name of the subsystem.
4. Rename it as `PCB1`.
5. Select both the `horizontal` and `vertical` blocks and assign them to a common subsystem in the *Manage Subsystems* panel:
 - a. Click `vertical` and `horizontal`.
 - b. Right-click the `controller` block and choose *Move To – New Subsystem*.
 - c. Rename the subsystem to `PCB2`.

Allegro X System-Level Design Methodology Guide

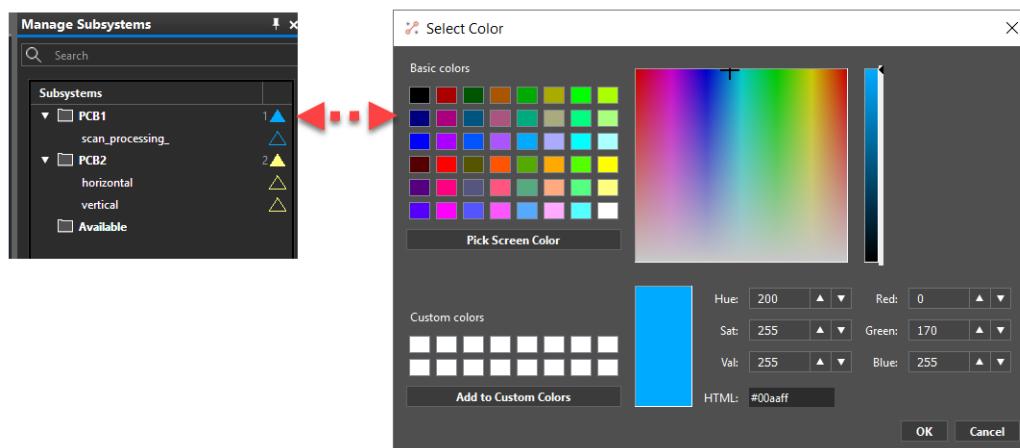
Designing a System using Functional Blocks

Triangular markers are placed on the canvas to indicate the subsystems that have been created.



The indicators on the net group show that the interconnectivity between the two blocks requires a definition, such as a connector, cable, or flex.

6. To differentiate between subsystems, you can assign a unique color to the triangular indicator representing a subsystem in the *Manage Subsystems* panel. To change the color, click the triangle in the *Manage Subsystems* panel.

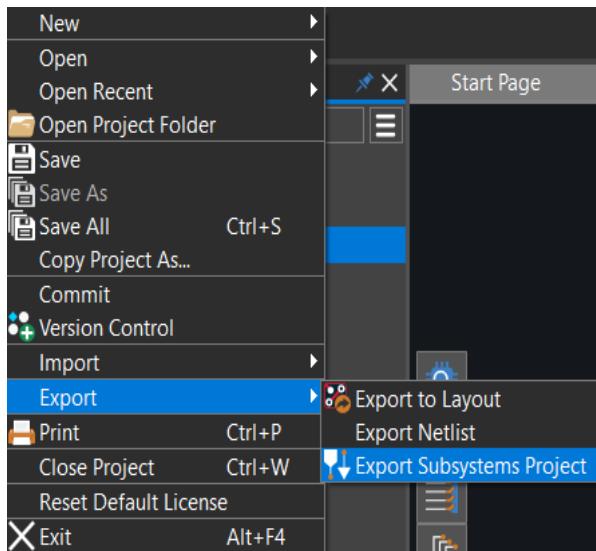


7. After defining subsystems, you can export them to create individual design projects that can be distributed to design teams.

Creating Physical Boards for the Subsystems

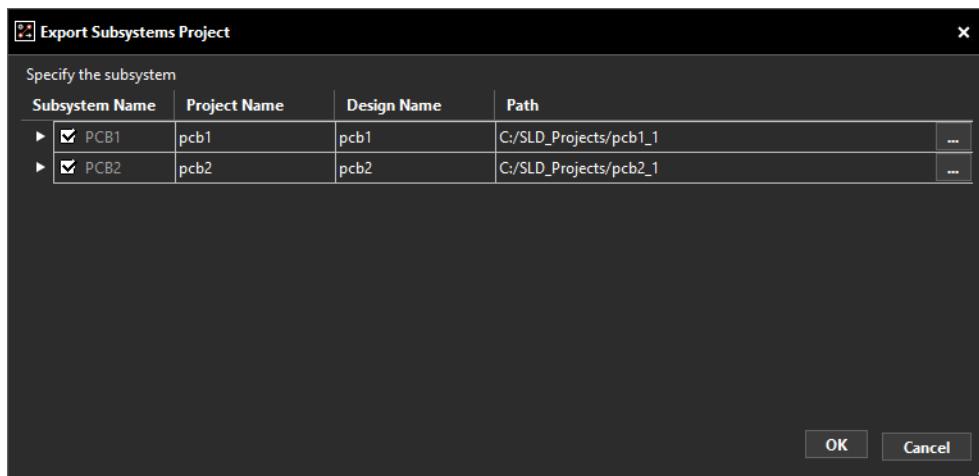
After adding connectivity information between components and identifying the subsystems, you can create the board files, and hand them over to the design teams.

1. Choose *File – Export – Export Subsystems Project*.



2. Save the design, if prompted.

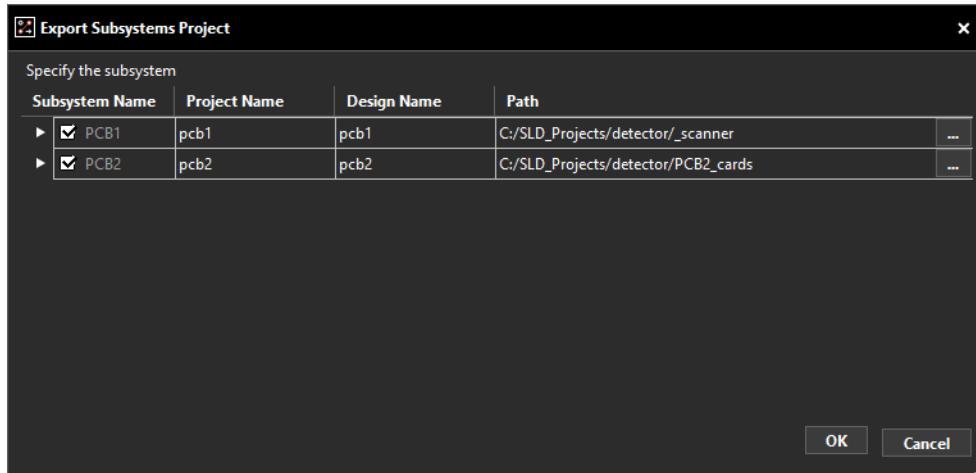
The *Export Subsystems Project* dialog box opens, where the two subsystems you identified are listed.



Allegro X System-Level Design Methodology Guide

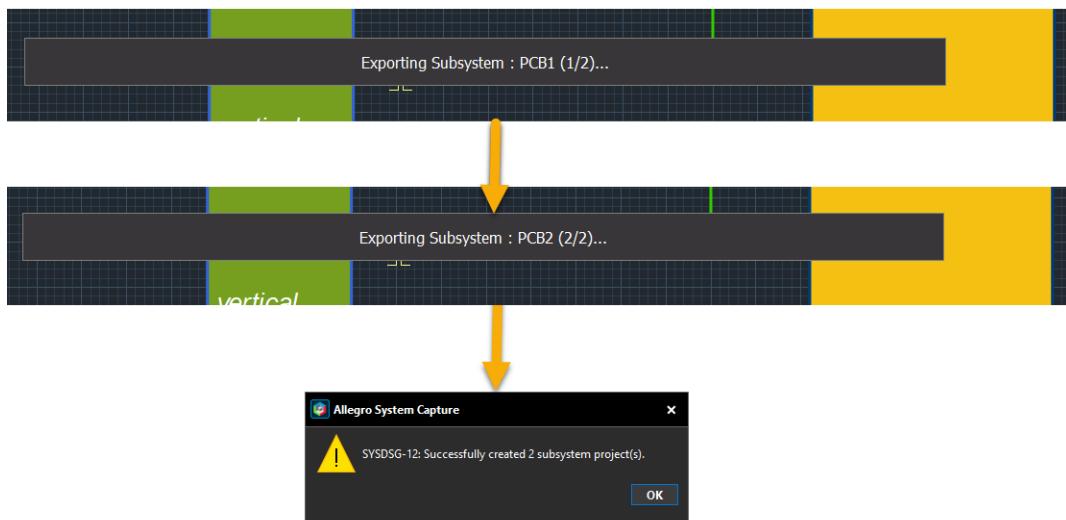
Designing a System using Functional Blocks

3. Change the default project names, design names, or locations, if required.



4. Click *OK*.

After showing the progress, a message confirms the creation of subsystems.

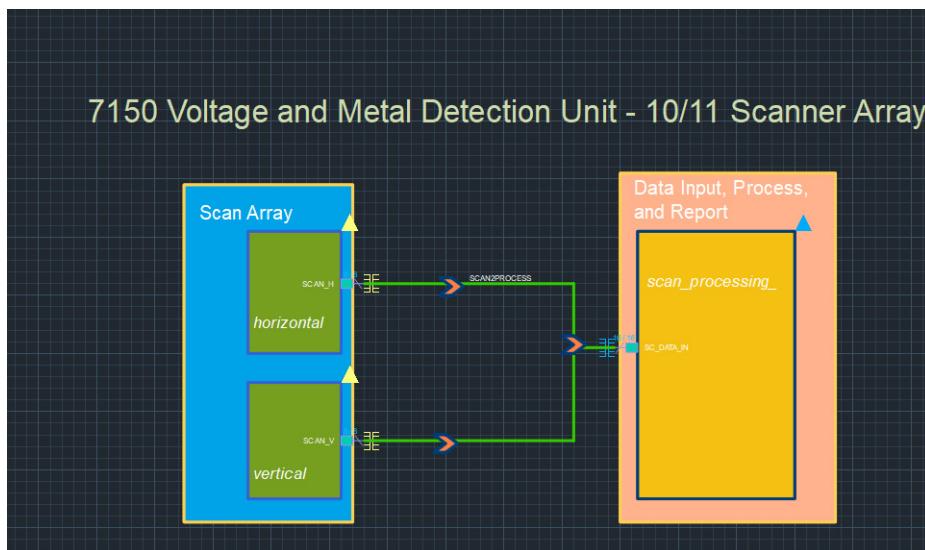


The subsystems are now available as independent projects and can be shared with hardware design teams.

Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

To add visual representation of the system design, you can use the various drawing tools, available images, and so on. Here is a sample:



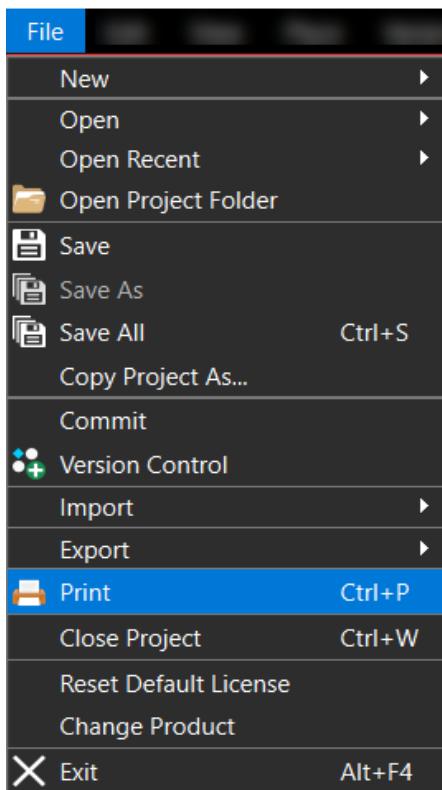
You can now save the project.

Printing System-Level Designs

After creating a system-level design, you can print it as a hard copy, PDF, Smart PDF, and so on, just like schematic designs. Blocks are printed in the top-down order, without any repetition.

To print a system-level design, follow these steps:

1. Choose *File – Print*.



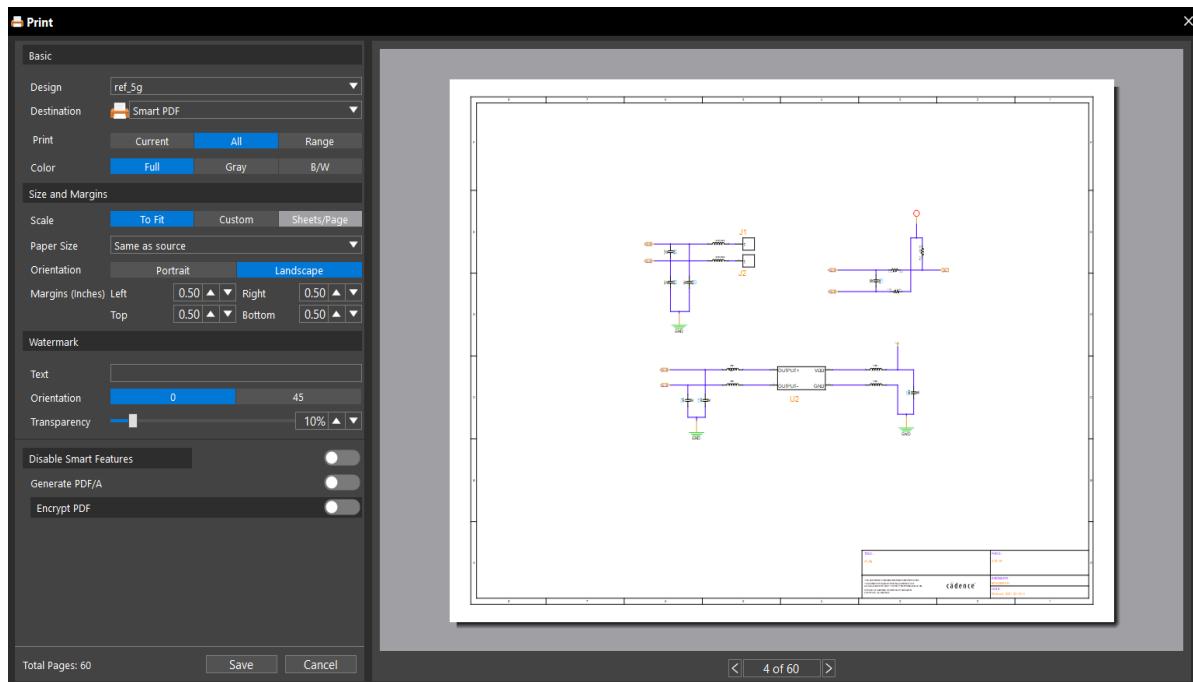
2. Select the *Destination*.

You can select various printing options such as *Smart PDF*, *OneNote (Desktop)*, *Save as PDF* and so on, as per your requirement.

Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

3. Make the required changes in the settings. For details, see [Print and Generate PDFs of a design](#).



4. Click *Save* or *Print*.

This will vary based on the selected *Destination*. For *Smart PDF* and *Save as PDF* options, *Save* will be displayed. For the rest of the options, *Print* will be displayed.

Allegro X System-Level Design Methodology Guide

Designing a System using Functional Blocks

Building a System using Multi-Board Connectivity

This chapter gives you an overview of the tasks involved in assembling a new subsystem in System Capture using existing designs and boards, also called the bottom-up approach.

Benefits

- Existing boards and designs can be reused for a system design
- Multi-board projects can handle inter-design connectivity

Prerequisites

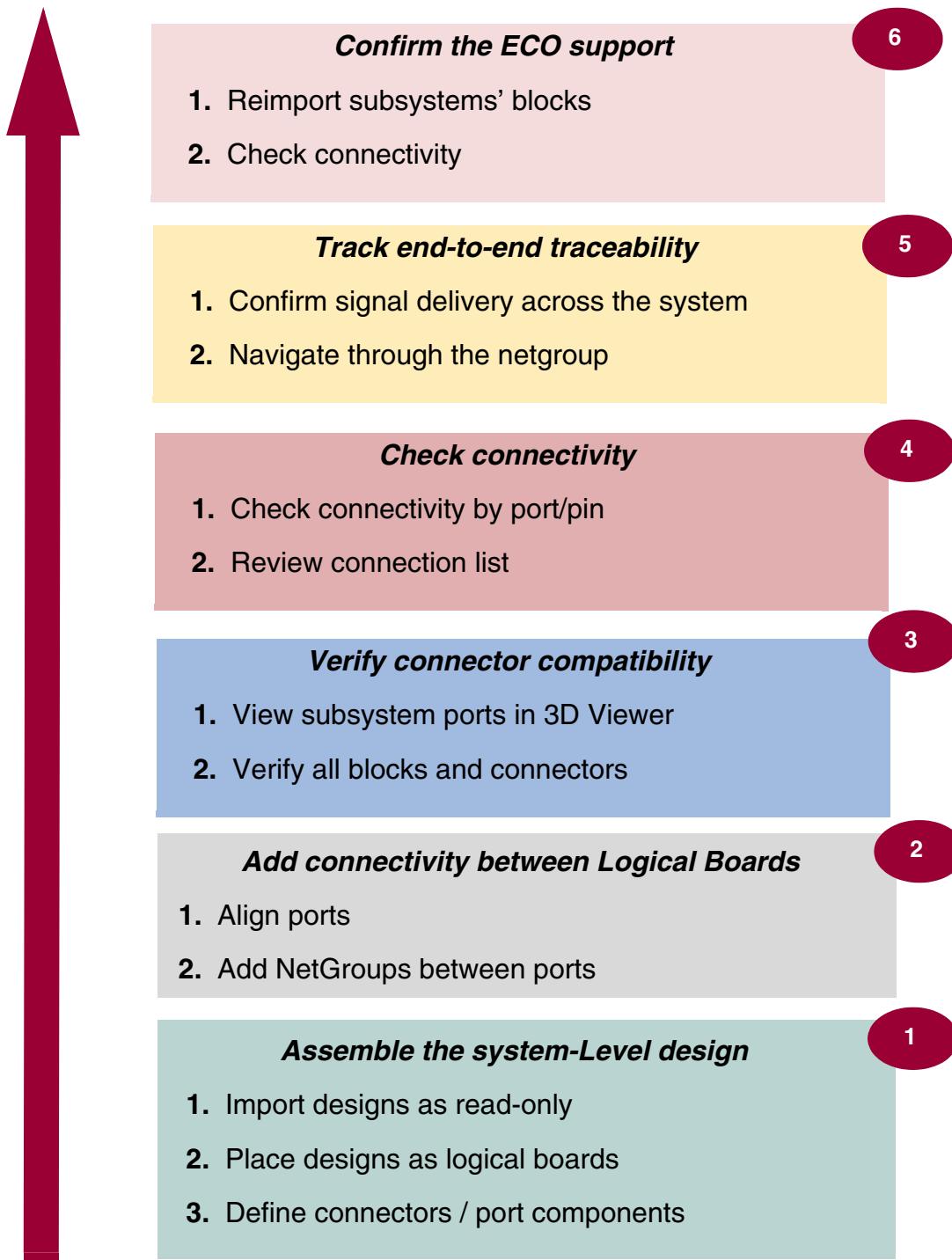
Schematics and board files that are ready for integration.

Tasks involved

In the bottom-up approach, read-only copies of existing boards or projects are reused to create an integrated higher system. To build a system using multi-board connectivity:

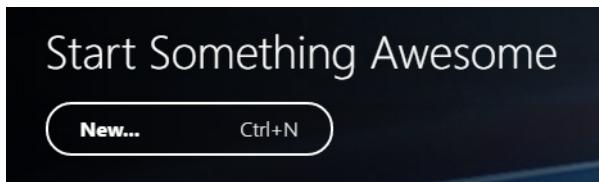
- [Creating a New Project](#)
- [Adding Existing Designs](#)
- [Creating System-Level Design](#)
- [Adding Connectivity Between Subsystems](#)
- [Verifying NetGroup Objects and Connectivity](#)
- [Generating Connectivity Reports](#)

Flow Diagram

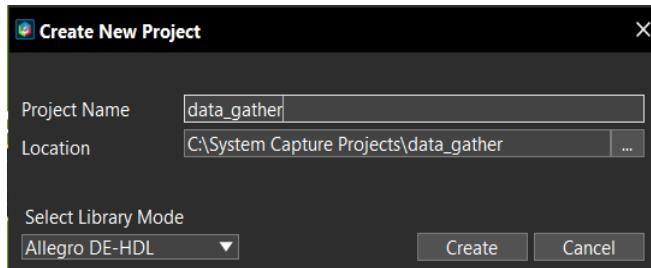


Creating a New Project

1. Start Allegro X System Capture.
2. Select a system design license.
3. Choose *New* on the *Start Page*.



4. Specify the project name, location, and design name.



5. Click *Create*.

An empty page opens.

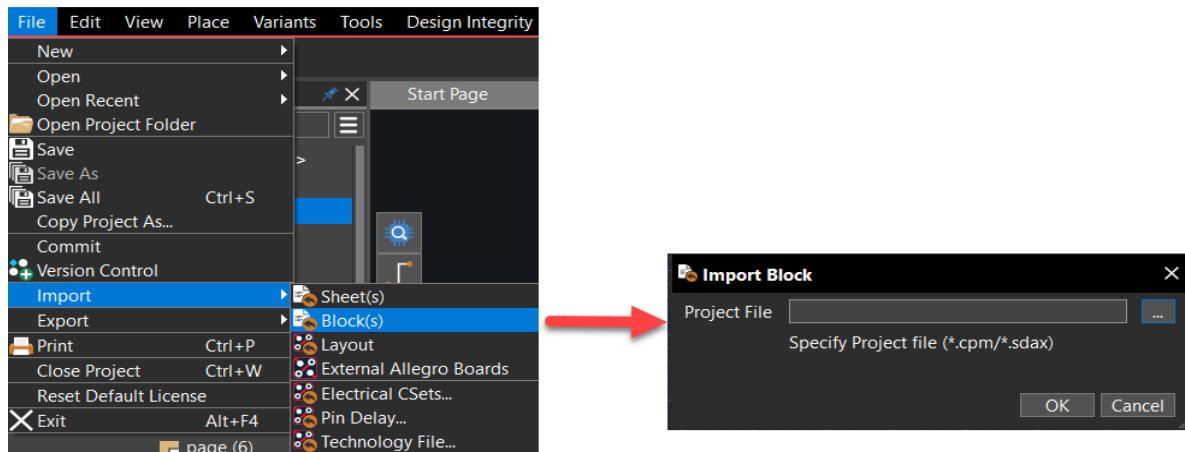
Adding Existing Designs

To add existing designs:

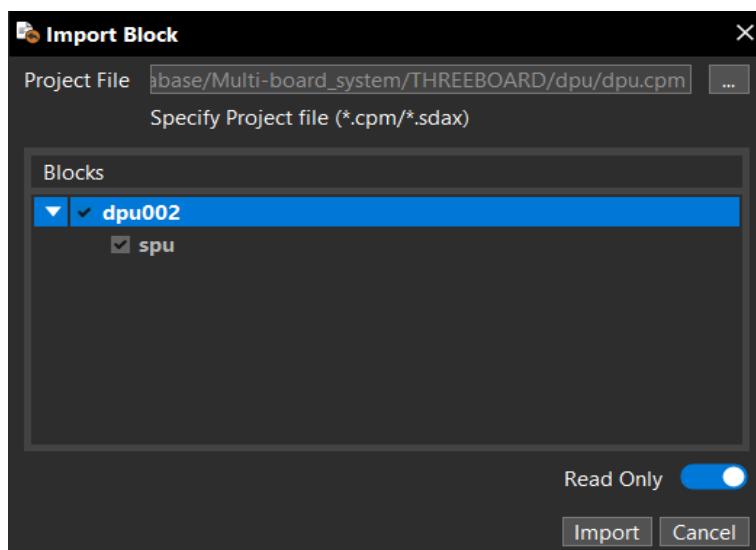
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

1. Choose *File – Import – Block(s)*.



- Ensure that *Allegro X System Capture* option is selected.
- Navigate to the folder that contains the design to import.
- Select *<project.cpm>, dpu.cpm*.
- Click *Open*.
- Select the block(s) to import. For example, *spu*.



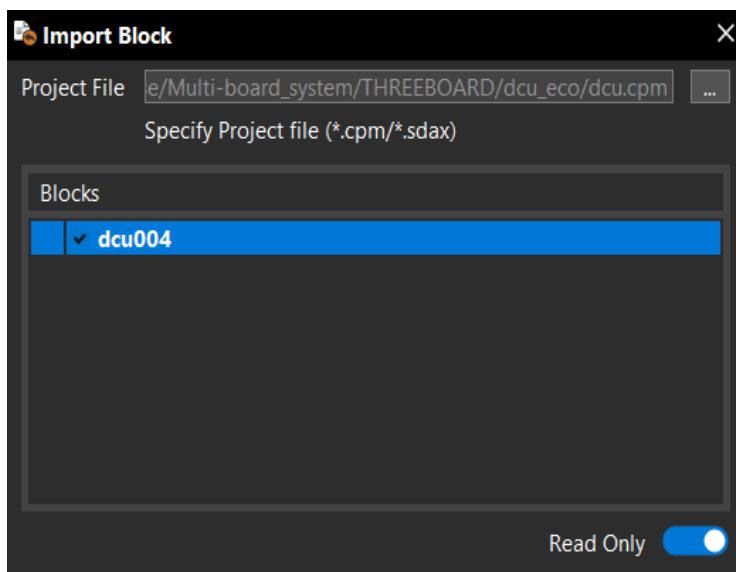
- Select the *Read Only* option.
- Click *Import*.

Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

- h. The block symbol gets attached to the cursor. For now, press Esc. Block will be added with connectivity later.
2. Repeat step 1 for each design to be added.

For example, add *dcu* also to the system design.

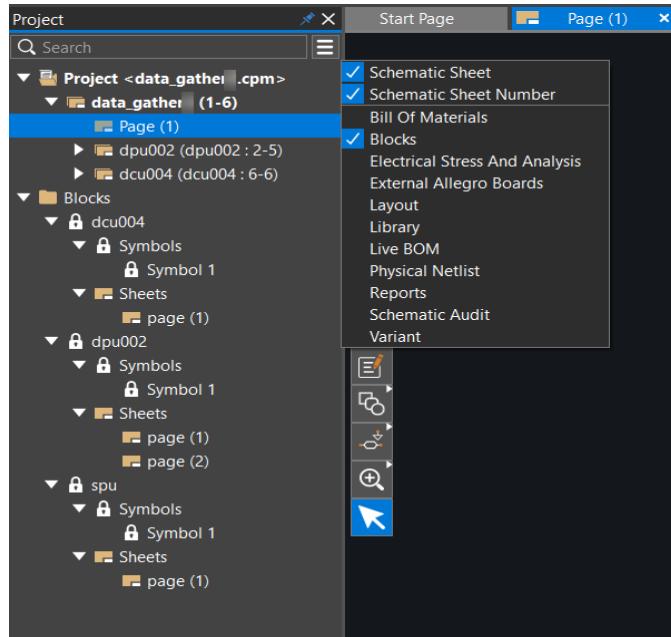


Press Esc to remove the block from the cursor.

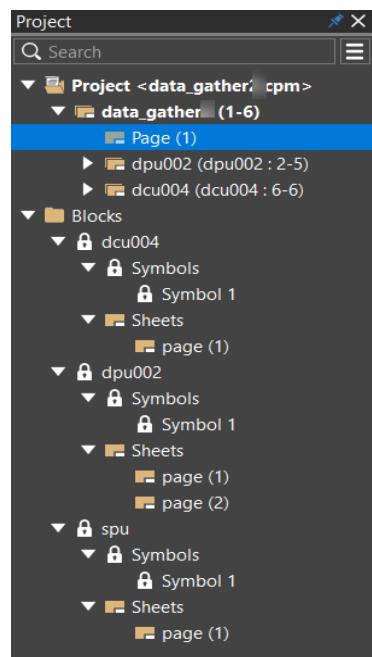
3. Check the project in *Project Viewer*.
4. Confirm the blocks are displayed by clicking the gear icon.

Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity



The imported designs have been added to the design as blocks.



You can create system-level design with the designs you imported as blocks.

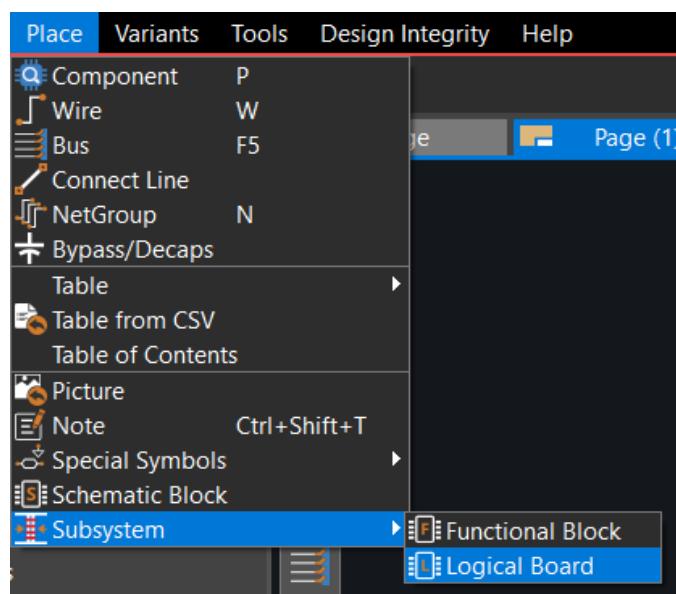
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

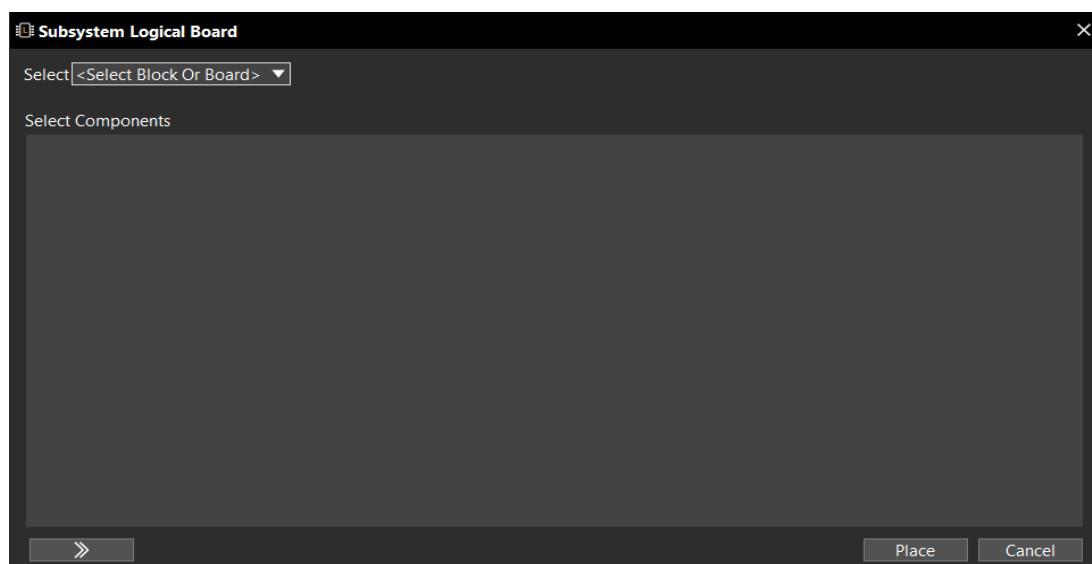
You can also add existing PCB boards with the *File – Import – External Allegro Boards* menu command.

Creating System-Level Design

1. Click *Place – Subsystem – Logical Board*.



The *Subsystem Logical Board* dialog box opens.



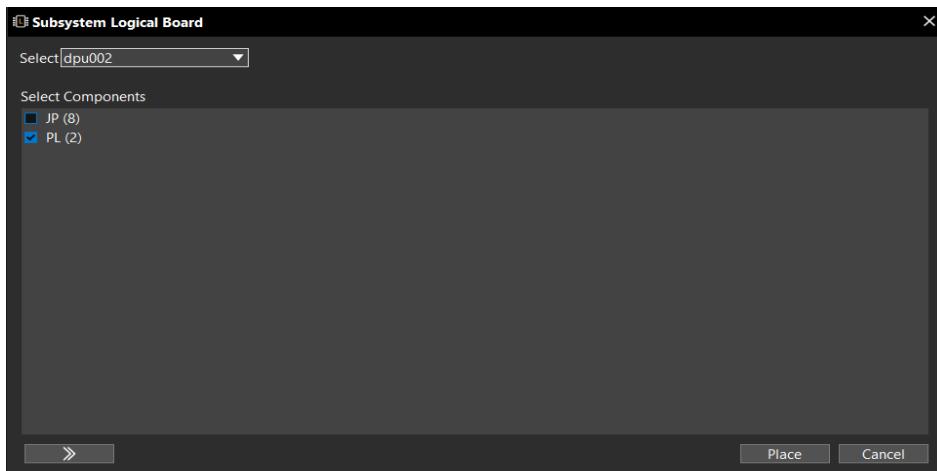
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

2. Add the components from the designs to the system design.

- Select the blocks to be added.
- Select the component from the list of connector references.

In this example, the component PL2 is selected from the design dpu002.



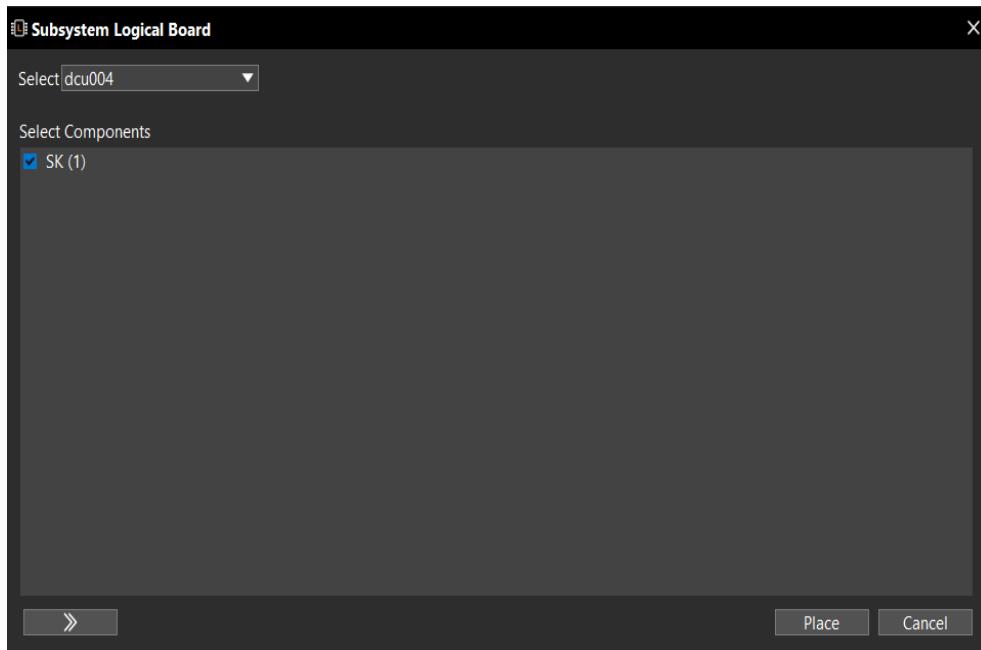
- Click *Place*.



Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

- d. Repeat for the *SK(1)* component from the design *dcu004* in *tutorial_database*.



- e. Place two instances of *dpu002* on each side.

3. Align the connector ports.



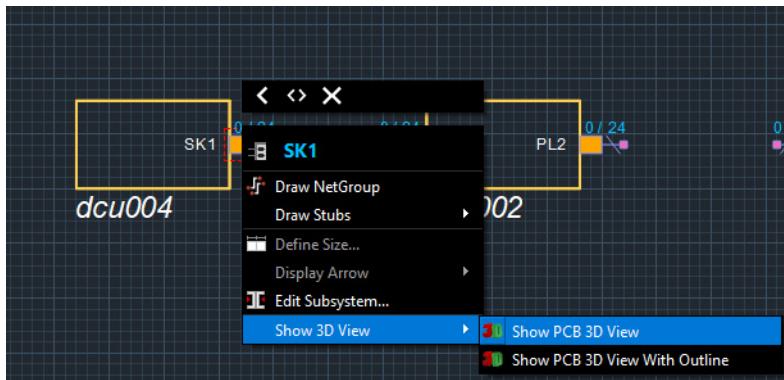
Both the block ports show 0 / 24 which indicates that no pins on either connector are connected to any other pin.

4. Preview the physical connectors used in the subsystems for alignment and compatibility checking.

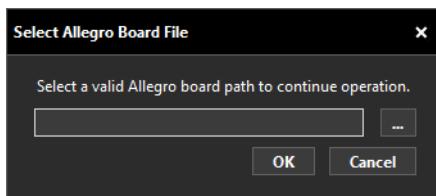
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

- a. Right-click the port on the component, such as *dcu004*, and choose *Show 3D View – Show PCB 3D View*.



You are prompted for the board file location.



- b. Specify a board file, such as *dcu004.brd* and click *OK*.

This file is available at the following location:

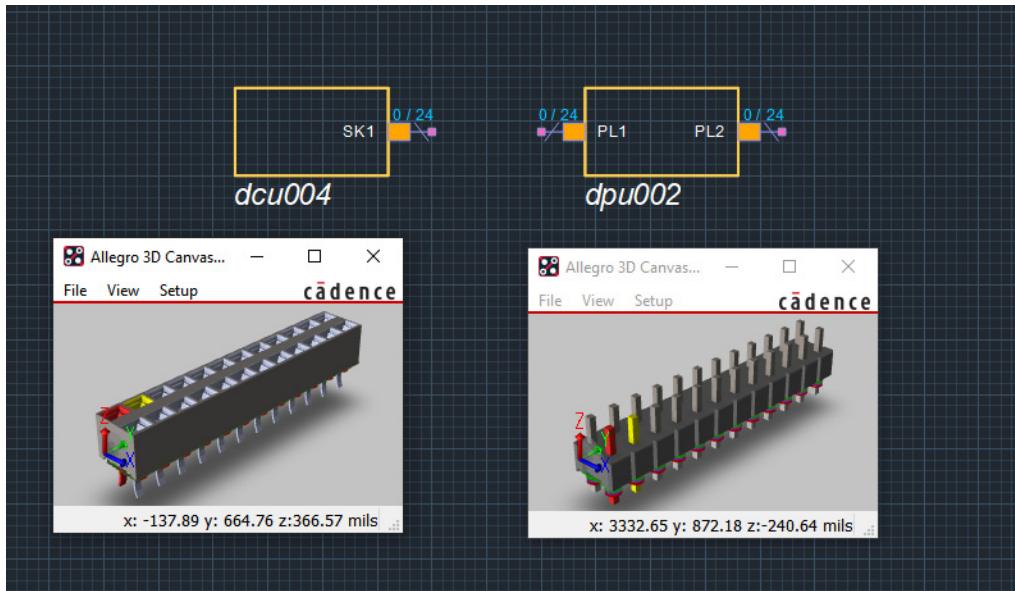
...\\Multi-board_system\\THREEBOARD\\dcu\\output\\dcu004\\physical
3D Viewer opens.

- c. Position the *Allegro 3D canvas* window above the block.
- d. Repeat steps a-c for the port on the block, *dpu002*.

Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

- e. In the *Select Allegro Board File* window, browse to `dpu002.brd`

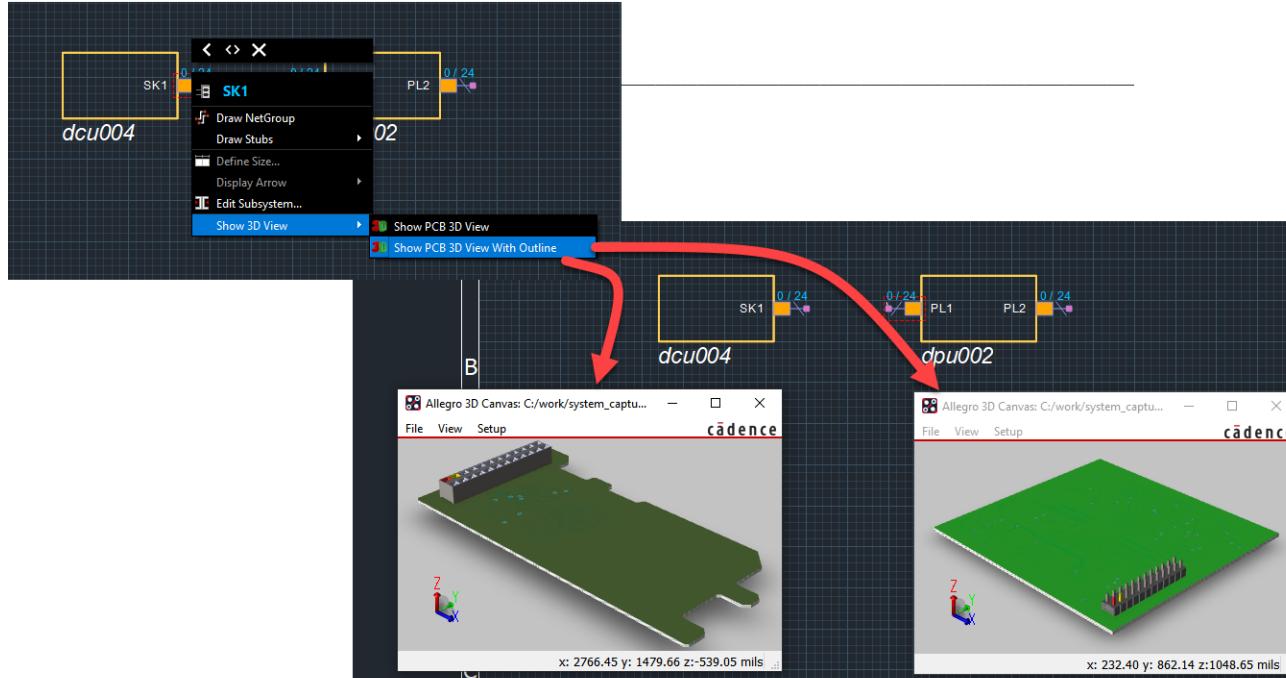


- f. To visualize the position and orientation of connectors in relation to the board, use the *Show – 3D View – Show PCB 3D View with Outline* menu command available in the shortcut menu for the subsystem.

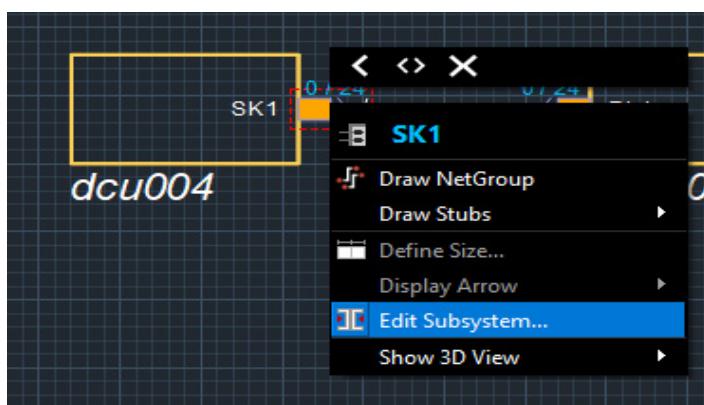
This launches the 3D Viewer and extracts the connector and bare board data.

Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity



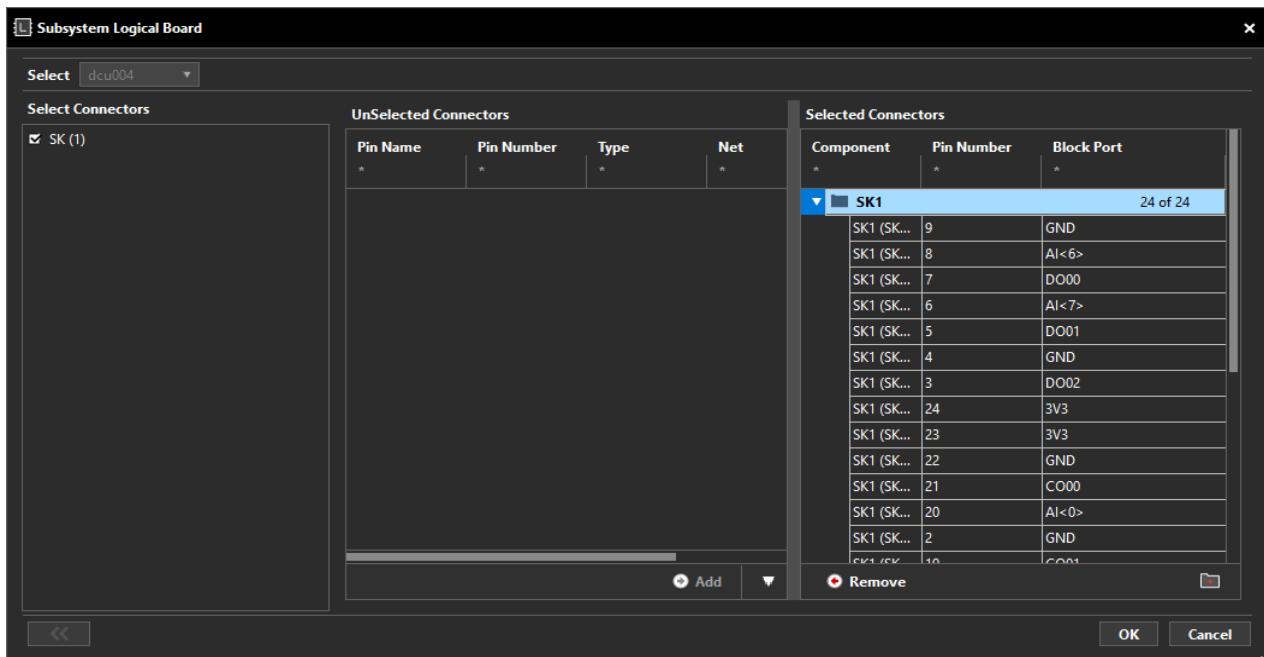
5. Close the Allegro 3D canvas windows after visual verification of the ports.
6. Check the connectivity status of the subsystem ports.
 - a. Right-click the port on the block and choose *Edit Subsystem*.



Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

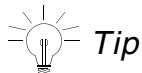
The *Subsystem Logical Board* dialog box opens.



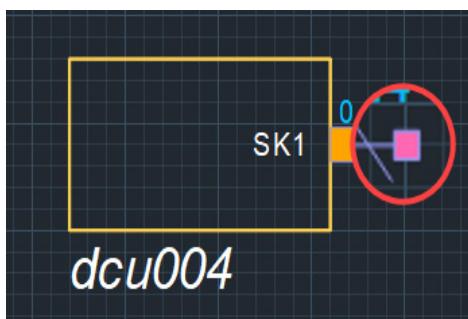
b. Review the information displayed:

- *Component Reference Designator*
- *Pin Number*
- *Signal Names*

7. Click *OK*.



A pink square on the port is a visual cue that the port is unconnected.



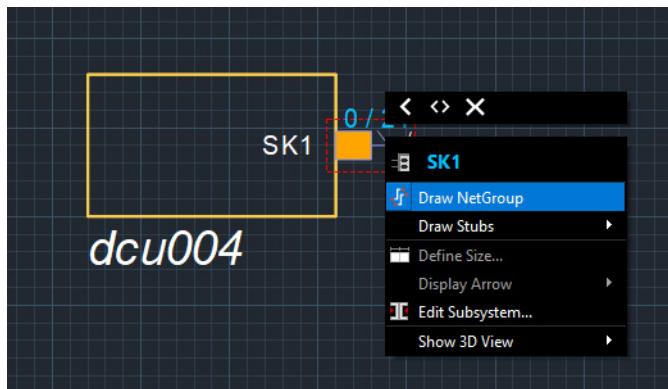
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

After placing the subsystems and verifying the ports, the next task is to build connectivity between the subsystems.

Adding Connectivity Between Subsystems

1. Right-click the port on a subsystem and choose *Draw NetGroup*.

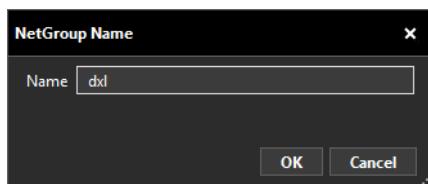


2. Attach the wire to the *PL 1* port.



You will be prompted to enter a name for the NetGroup.

3. Specify a name for the NetGroup.



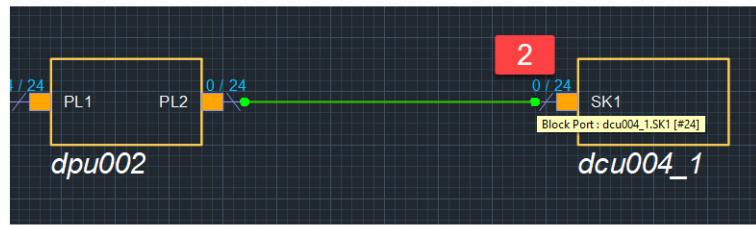
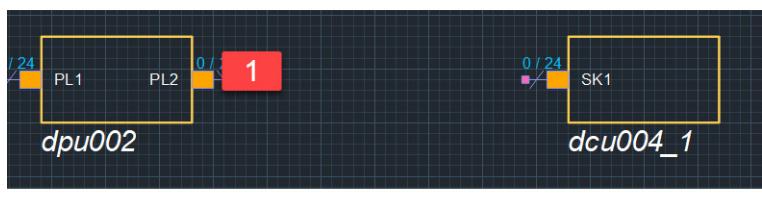
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

- Click OK.



- Add another NetGroup DXR to connect DCU004_1 to DPU002.



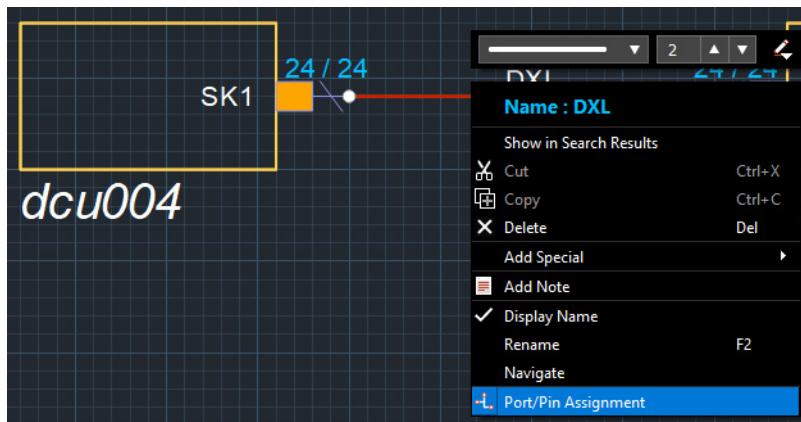
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

Verifying NetGroup Objects and Connectivity

1. Check the connections between the blocks.

- Right-click a NetGroup between two subsystems and choose *Port/Pin Assignment*.



The *Port/Pin Assignment* form displays.

dcu004					dpu002
System Net	Pin Number	Block Net	Pin Number	Blo	
*	*	*	*	*	
DXL	SK1	SK1	PL1	PL1	
GND_1	SK1.1	GND	PL1.1	GND	
GND_2	SK1.2	GND	PL1.2	GND	
R_D002	SK1.3	DO02	PL1.3	R_D002	
GND_3	SK1.4	GND	PL1.4	GND	
R_D001	SK1.5	DO01	PL1.5	R_D001	
R_ARRAY<...	SK1.6	AI<7>	PL1.6	R_ARRAY<7>	
R_D000	SK1.7	DO00	PL1.7	R_D000	
R_ARRAY<...	SK1.8	AI<6>	PL1.8	R_ARRAY<6>	
GND_4	SK1.9	GND	PL1.9	GND	
R_ARRAY<...	SK1.10	AI<5>	PL1.10	R_ARRAY<5>	
WRITE	SK1.11		PL1.11	WRITE	
R_ARRAY<...	SK1.12	AI<4>	PL1.12	R_ARRAY<4>	
CLK	SK1.13		PL1.13	CLK	
R_ARRAY<...	SK1.14	AI<3>	PL1.14	R_ARRAY<3>	
GND_5	SK1.15	GND	PL1.15	GND	
R_ARRAY<...	SK1.16	AI<2>	PL1.16	R_ARRAY<2>	
R_CO02	SK1.17	CO02	PL1.17	R_CO02	
R_ARRAY<...	SK1.18	AI<1>	PL1.18	R_ARRAY<1>	
R_CO01	SK1.19	CO01	PL1.19	R_CO01	

Note: Color coding feature will work only when the directives given in [Chapter 1, “Setting up Color Coding for Matching Nets”](#) are set.

Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

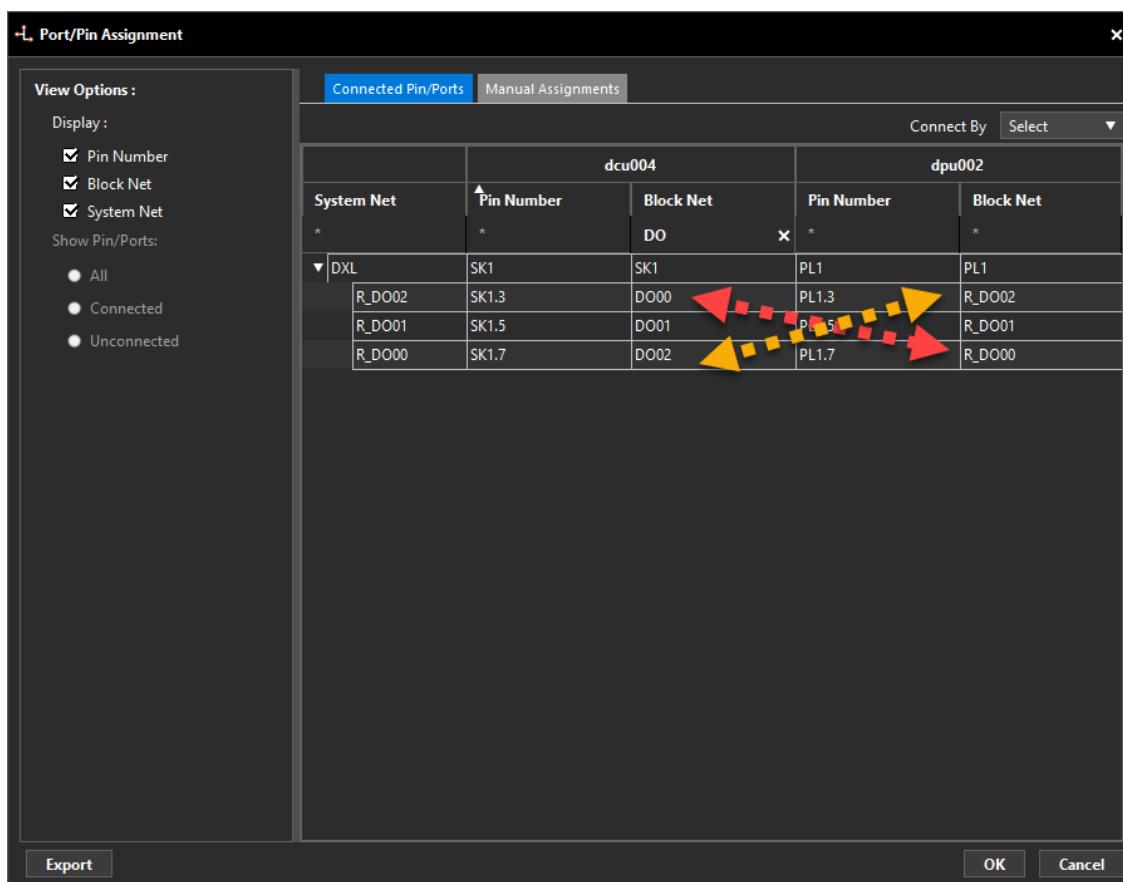
- a. Control the columns to display with *View Options*.
- b. Verify the connectivity between the blocks with NetGroup in the *Connected Pin/Ports* tab.
- c. Check the unconnected pins and make the connections, if any, in the *Manual Assignments* tab.

In this example, there are no unconnected pins.

You can use wildcards to filter nets and check the connections. For example:

- GND nets
- Nets with a voltage value

- d. Check *Block Net DO**



dcu004					dpu002	
System Net	Pin Number	Block Net	Pin Number	Block Net		
*	*	DO	*	*		
DXL	SK1	SK1	PL1	PL1		
R_D002	SK1.3	D000	PL1.3	R_D002		
R_D001	SK1.5	D001	PL1.5	R_D001		
R_D000	SK1.7	D002	PL1.7	R_D000		

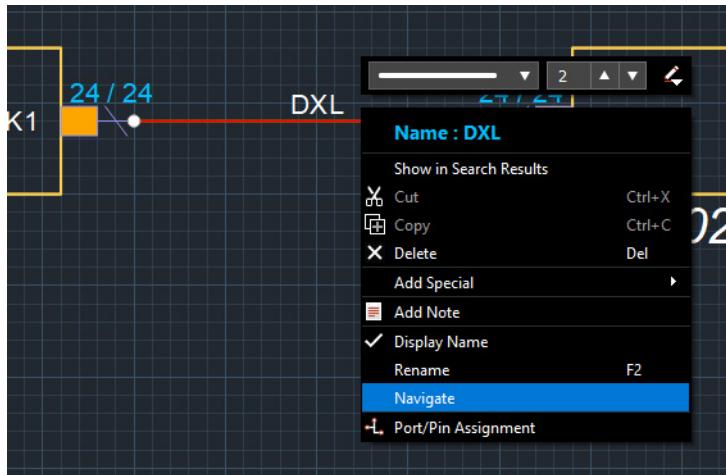
There appears to be a potential error. To confirm, the signal can be traced.

2. Check the end-to-end traceability of a signal.

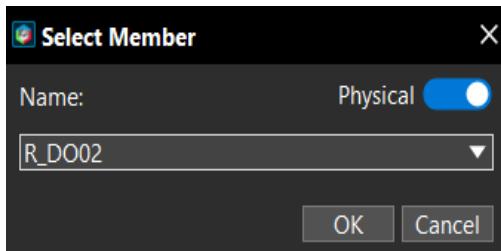
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

- a. Right-click a NetGroup and choose *Navigate*.



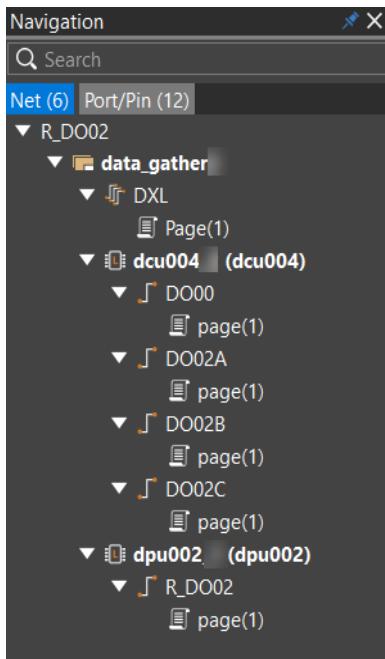
- b. Choose the R_DO02 signal.



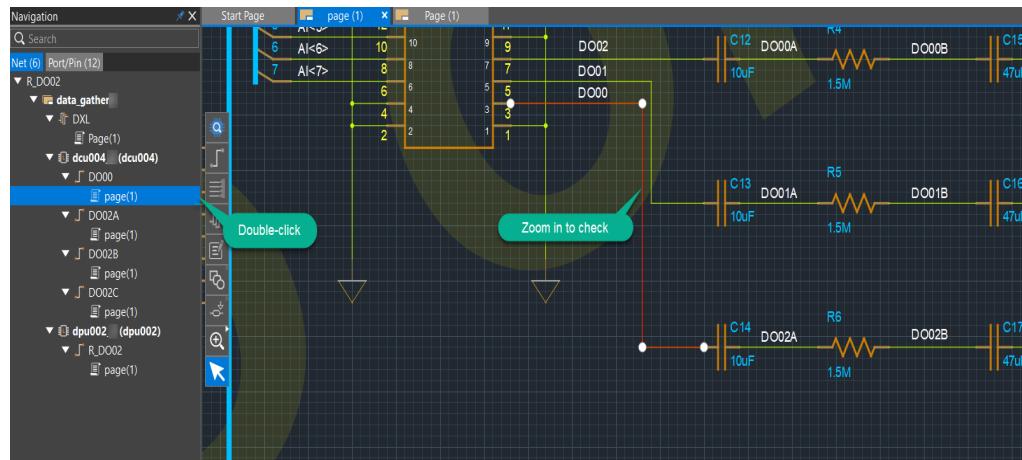
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

The *Navigation* tab in *Project Explorer* displays the locations the signal passes through. You can see the path of the net/signal in the tree view.



- In the *Net* tab, double-click the page entries one by one to trace the signal across the design.



- In the *Port/Pin* tab, double-click each entry to traverse the design.

After verifying the signal path, and making changes, as needed, close the subsystem tabs.

Generating Connectivity Reports

To ascertain the scope of nets and connectivity across the designs, System Capture provides the following types of reports:

- System Connectivity Report

Lists the connections between two subsystems.

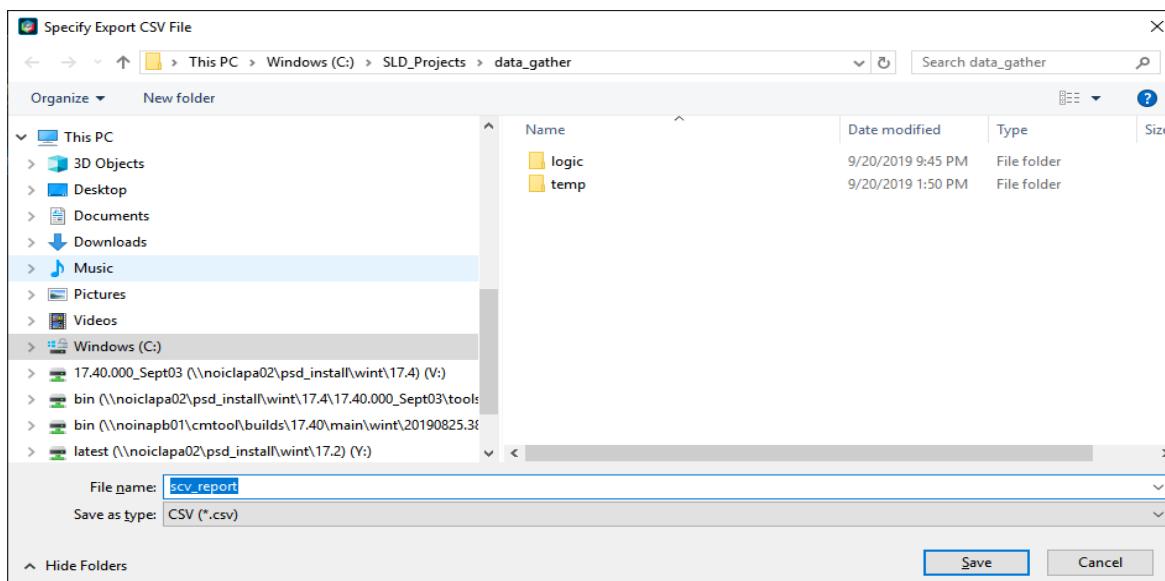
- End-to-End Connectivity Report

Lists the nets and connections throughout the design.

System Connectivity Report

You can create a report for the connectivity in a design. System Capture creates a CSV file that you can open in an external application, such as MS Excel.

1. Choose *Tools – Generate System Design Report*.
2. Specify a name and location for the CSV file.

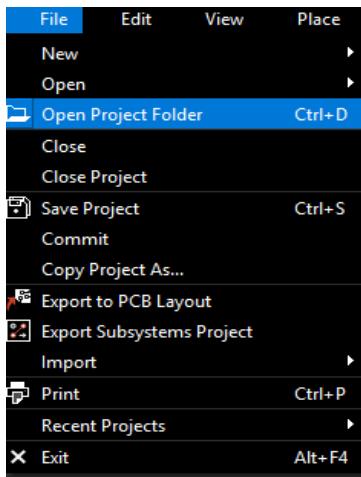


3. Click Save.

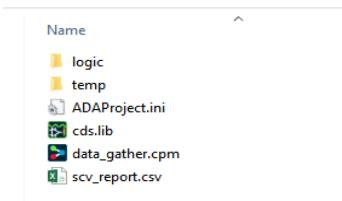
Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

4. Choose *File – Open Project Folder*.



5. Open the report file from Windows Explorer.



Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

6. Review the System Connectivity Report in Excel.

A	B	C	D	E	F	G	H	I
1	dcu004		dcu004_1		dpu002			
2	System Net	System Net	Pin Number	Block Net	Pin Number	Block Net	Pin Number	Block Net
3	DXL	3V3	SK1.23	3V3		PL1.23	3V3	
4	DXL	3V3_1	SK1.24	3V3		PL1.24	3V3	
5	DXR	3V3_2		SK1.23	3V3	PL2.23	3V3	
6	DXR	3V3_3		SK1.24	3V3	PL2.24	3V3	
7	DXR	AI<0>		SK1.20	AI<0>	PL2.20	L_ARRAY<0>	
8	DXR	AI<1>		SK1.18	AI<1>	PL2.18	L_ARRAY<1>	
9	DXR	AI<2>		SK1.16	AI<2>	PL2.16	L_ARRAY<2>	
10	DXR	AI<3>		SK1.14	AI<3>	PL2.14	L_ARRAY<3>	
11	DXR	AI<4>		SK1.12	AI<4>	PL2.12	L_ARRAY<4>	
12	DXR	AI<5>		SK1.10	AI<5>	PL2.10	L_ARRAY<5>	
13	DXR	AI<6>		SK1.8	AI<6>	PL2.8	L_ARRAY<6>	
14	DXR	AI<7>		SK1.6	AI<7>	PL2.6	L_ARRAY<7>	
15	DXL	CLK	SK1.13			PL1.13	CLK	
16	DXR	C000		SK1.21	C000	PL2.21	L_C000	
17	DXR	C001		SK1.19	C001	PL2.19	L_C001	
18	DXR	C002		SK1.17	C002	PL2.17	L_C002	
19	DXR	D000		SK1.7	D000	PL2.7	L_D000	
20	DXR	D001		SK1.5	D001	PL2.5	L_D001	
21	DXR	D002		SK1.3	D002	PL2.3	L_D002	
22	DXL	GND	SK1.1	GND		PL1.1	GND	
23	DXL	GND_1	SK1.2	GND		PL1.2	GND	
24	DXL	GND_2	SK1.4	GND		PL1.4	GND	
25	DXL	GND_3	SK1.9	GND		PL1.9	GND	
26	DXL	GND_4	SK1.15	GND		PL1.15	GND	
27	DXL	GND_5	SK1.22	GND		PL1.22	GND	
28	DXR	GND_6		SK1.1	GND	PL2.1	GND	
29	DXR	GND_7		SK1.2	GND	PL2.2	GND	
30	DXR	GND_8		SK1.4	GND	PL2.4	GND	

End-to-End Connectivity Report

The end-to-end connectivity report will include any Xnets or any schematic in the given logical boards.



Ensure that the system architecture diagram is on the first page of the design for generating the end-to-end connectivity report.

You can create customized reports using the `generateEndToEndConnectivityReport` Tcl command. The syntax of the command is:

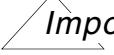
```
generateEndToEndConnectivityReport -file <file_path> [-net <net_pattern>] [-netgroup <netgroup_pattern>] [-power_net_only] [-io_net_only] [skip_dup_block_nets] [-report_sld_block_ports] [-report_netgroup] [-block_nets] [-pinname] [-system_aliases] [-block_order <list>]
```

Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

```
[-ic_prefix <prefix>]
```

where:

Field	Description
-file <file_path>	The output file location and name
[-net <net_pattern>]	The nets to be included in the report, such as *CLK*
[-netgroup <netgroup_pattern>]	The NetGroups to be included in the report, such as DDR*
[-power_net_only]	Extracts power and ground signals only
[-io_net_only]	Extracts IO nets only
 Important	
	In case few nets from one connector of a card have the VOLTAGE property, whereas the connected net on the mated card does not have any VOLTAGE property, both the nodes with the VOLTAGE property and the other end card connector nodes without the VOLTAGE property are skipped. For system aliases, the net connected to the third card using another connector are also skipped. Only the nets with the <code>-power_net_only</code> option get reported.
[skip_dup_block_nets]	Suppresses duplicate block nets in the report. If the block net for node is same as the previous node added to the report, the block net is not added.
[-report_sld_block_ports]	Includes system block ports
[-report_netgroup]	Reports NetGroups
[-block_nets]	Reports functional nets
[-pinname]	Shows pin names in the report instead of pin numbers. By default, pin numbers are displayed.

Allegro X System-Level Design Methodology Guide

Building a System using Multi-Board Connectivity

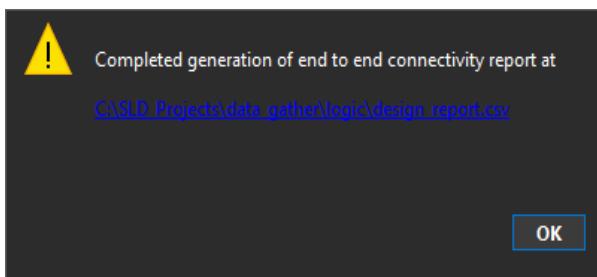
Field	Description
<code>[-system_aliases]</code>	Includes signal aliases across system in the report
<code>[-block_order <block1, block2, block3>]</code>	Sorts the blocks in the given order. In case there are many blocks and only a few are listed, the rest of the blocks are not included in the report.
<code>[-ic_prefix <prefix>]</code>	Sorts the connectors in the report to show the connectors with the specified prefix first, followed by the other connectors.

Example

1. Type the following command in the command window:

```
generateEndToEndConnectivityReport -file design_report.csv -  
io_net_only -report_netgroup -block_nets -system_aliases
```

The location of the report file is shown in a message.



2. Click the link.

The folder containing the report opens.

Updating Subsystems in a Design

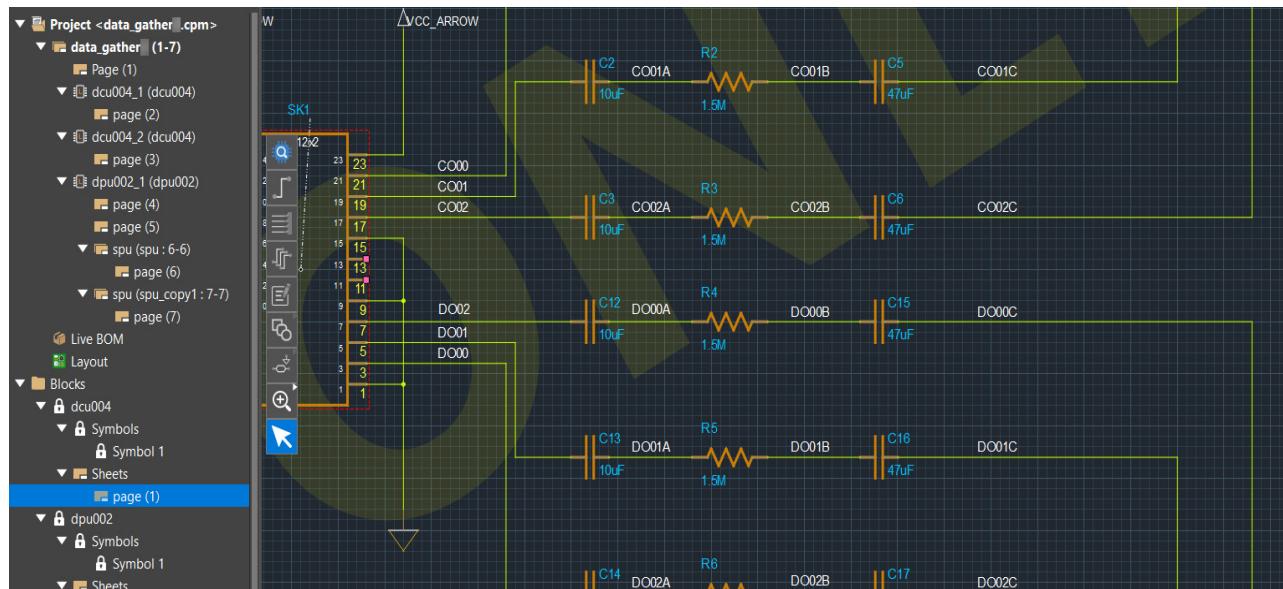
While developing of an electronic hardware system, some connectivity changes might be required. Some of these changes might affect signals that interconnect subsystems. It then becomes necessary to plan and advise changes to the design teams working on the designs that have been imported into a system design in the read-only mode. After the changes are done, the system needs to be updated and the connection integrity verified again.

In this section, an Engineering Change Order (ECO) is shown, which requires resolution at the system-level.

1. View the *dcu004* schematic.

- a. Double-click *Blocks* – *<dcu004>* – *Sheets* – *Page (1)* in the *Project Explorer*.

The schematic for the subsystem opens.



- b. Check the *DO00* and *DO02* signals that are connected to *SK1*.

Allegro X System-Level Design Methodology Guide

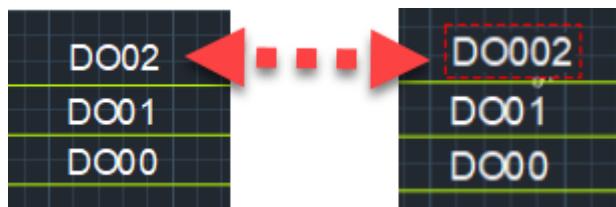
Updating Subsystems in a Design

These are incorrect assignments and need to be swapped. As the `dCU004` block was originally imported as a read-only block, the change to the schematic needs to be made at the source.

2. Close the `dCU004` schematic tab in the system design project.
3. Open the `dCU004` design.
4. Select `dCU004` from the recently used project list, or browse to the `dCU004` project location.
5. Zoom into the two signals that need to be changed, `DO00` and `DO02`.

Note: System Capture packages in real-time, therefore it is necessary to provide an unused name as a temporary placeholder when renaming signals, as described in the following steps.

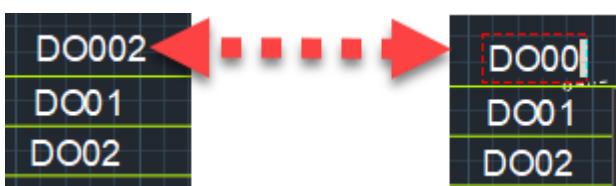
- a. Double-click the top signal `DO02` and temporarily rename it as `DO002`.



- b. Rename signal `DO00` to `DO02`.



- c. Rename the original signal `DO02` that is currently `DO002` to `DO00`.



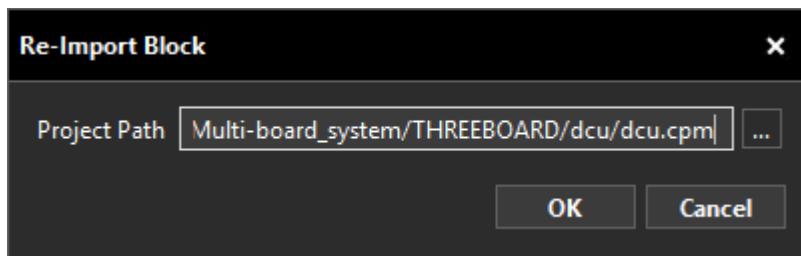
- d. Save the project.

This completes the required change. Switch to the system design project.

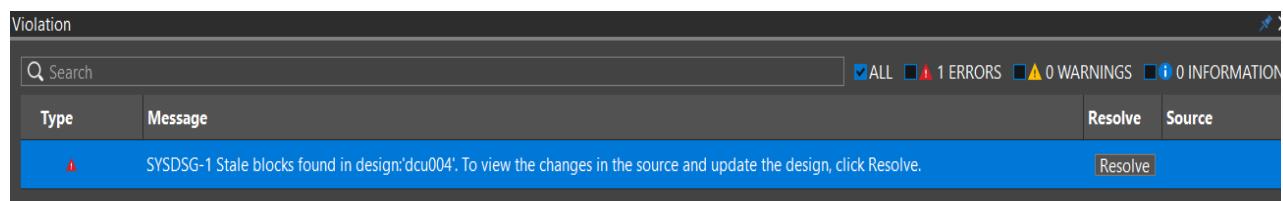
Allegro X System-Level Design Methodology Guide

Updating Subsystems in a Design

6. In *Project Explorer*, right-click `dcu004` and choose *Re-Import Block*.

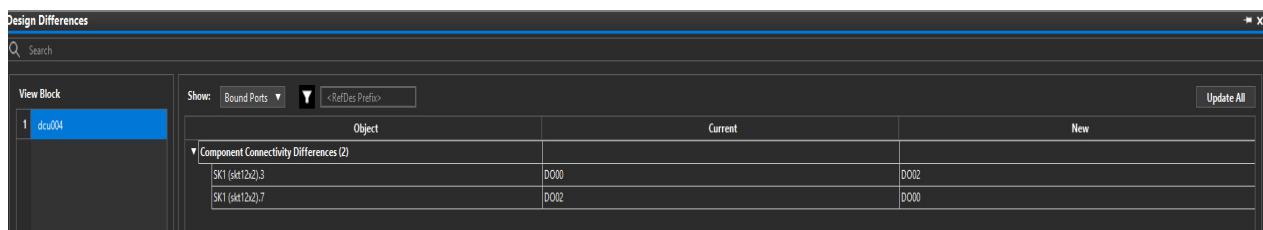


7. Click *OK*.
8. Check the *Violation* window.

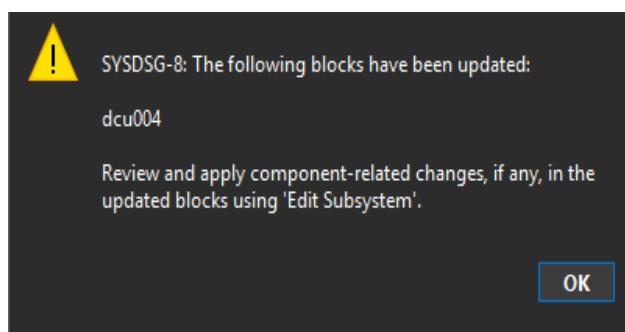


9. Click *Resolve*.

In the *Design Differences* window, two differences are identified and displayed.



10. Click *Update All*.



11. Click *OK*.

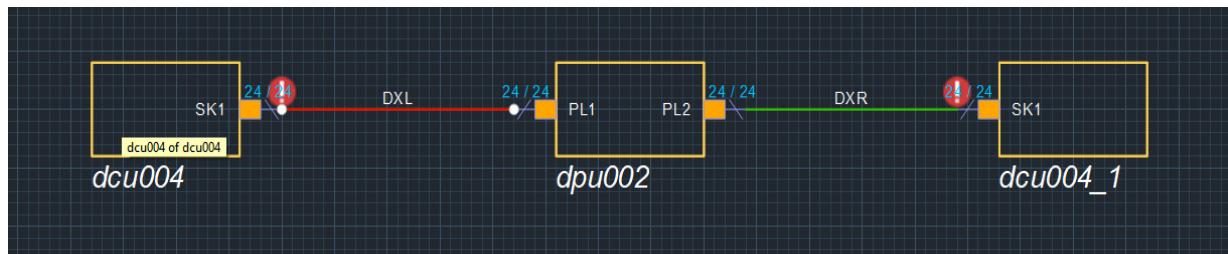
Allegro X System-Level Design Methodology Guide

Updating Subsystems in a Design

In case there are connection problems, red icons appear on the ports. This indicates that there are differences between the original System-level connections and the newly updated version of the block.

Graphical warning markers are displayed at both *dcu004* and *dcu004_1* ports.

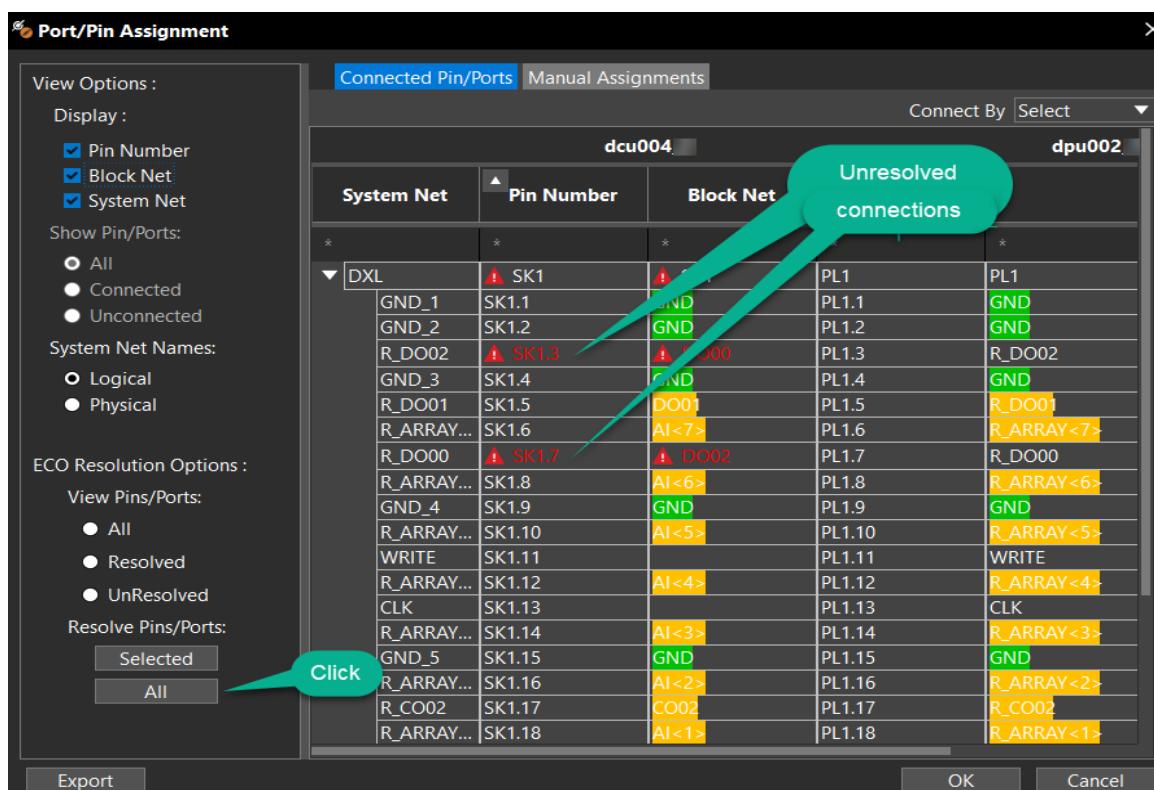
First, resolve *dcu004*.



12. Right-click the *DXL* NetGroup and choose *Port/Pin Assignment*.

13. Check the *Connections* tab in the *Port/Pin Assignment* dialog box.

Mismatches are shown in red.



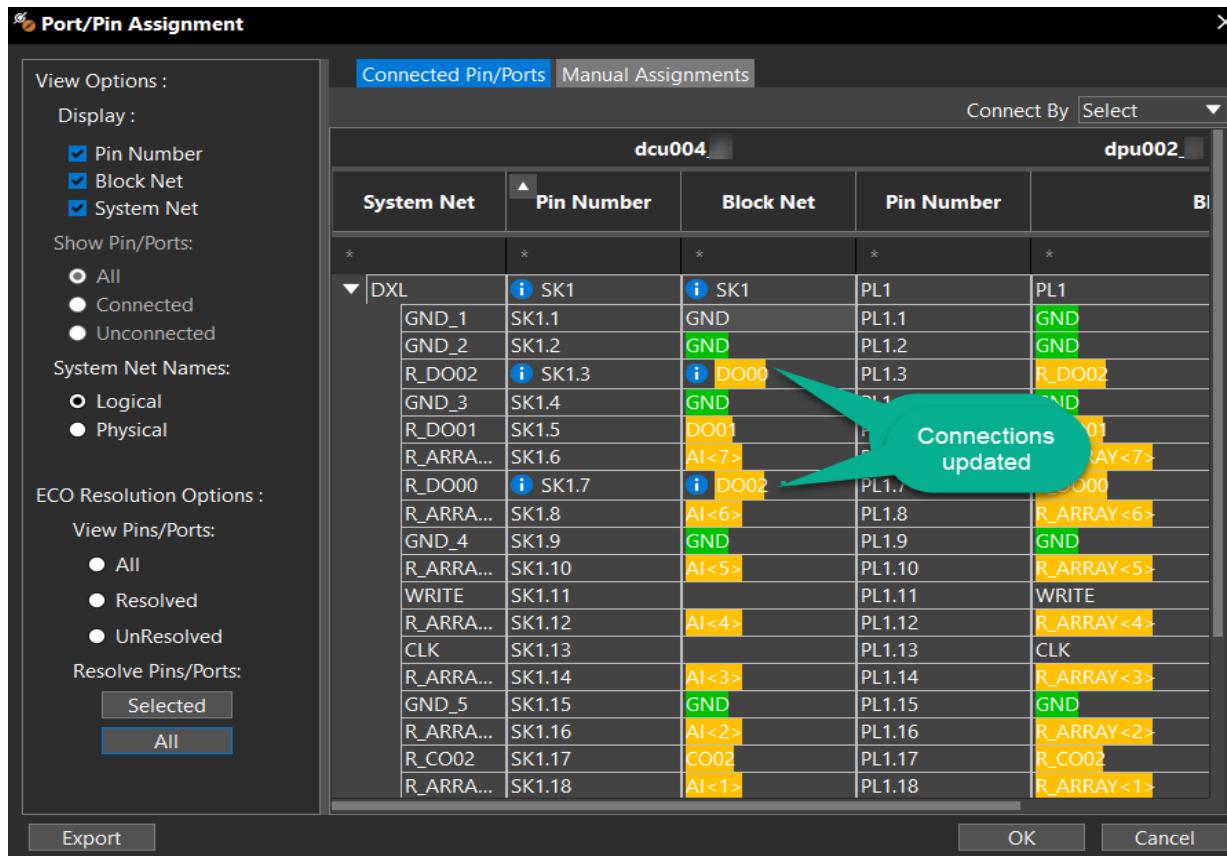
Allegro X System-Level Design Methodology Guide

Updating Subsystems in a Design

You can select mismatched connections one at a time, or select all and resolve them in one go.

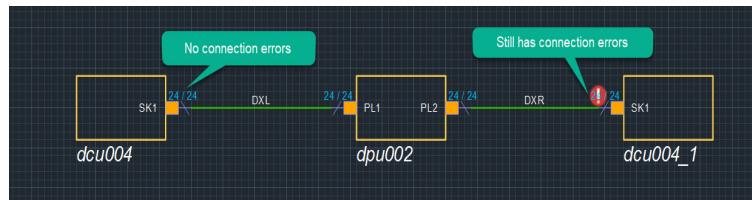
14. Click *All* in the *Resolve Pins* field.

The *Port/Pin Assignment* form updates.



15. Click *OK*.

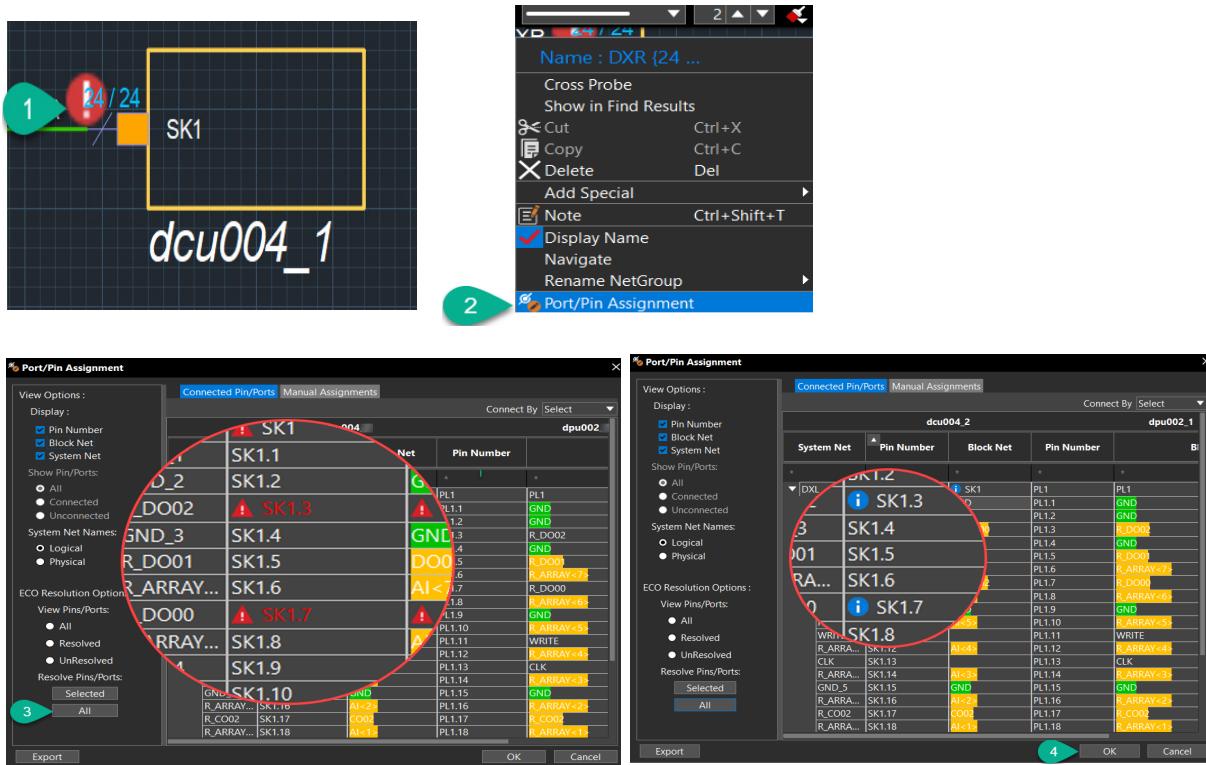
The canvas has been updated.



Allegro X System-Level Design Methodology Guide

Updating Subsystems in a Design

16. Similarly, correct the connections for the *DXR* NetGroup.



17. Ensure all connectivity changes have been resolved in the design.



18. Save the project.

This completes updating a subsystem or design, after it is added to a system design.