cadence®

Allegro® Front-to-Back User Guide

Product Version 17.4-2019
October 2019

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

Contents

Preface

This chapter contains the following information:

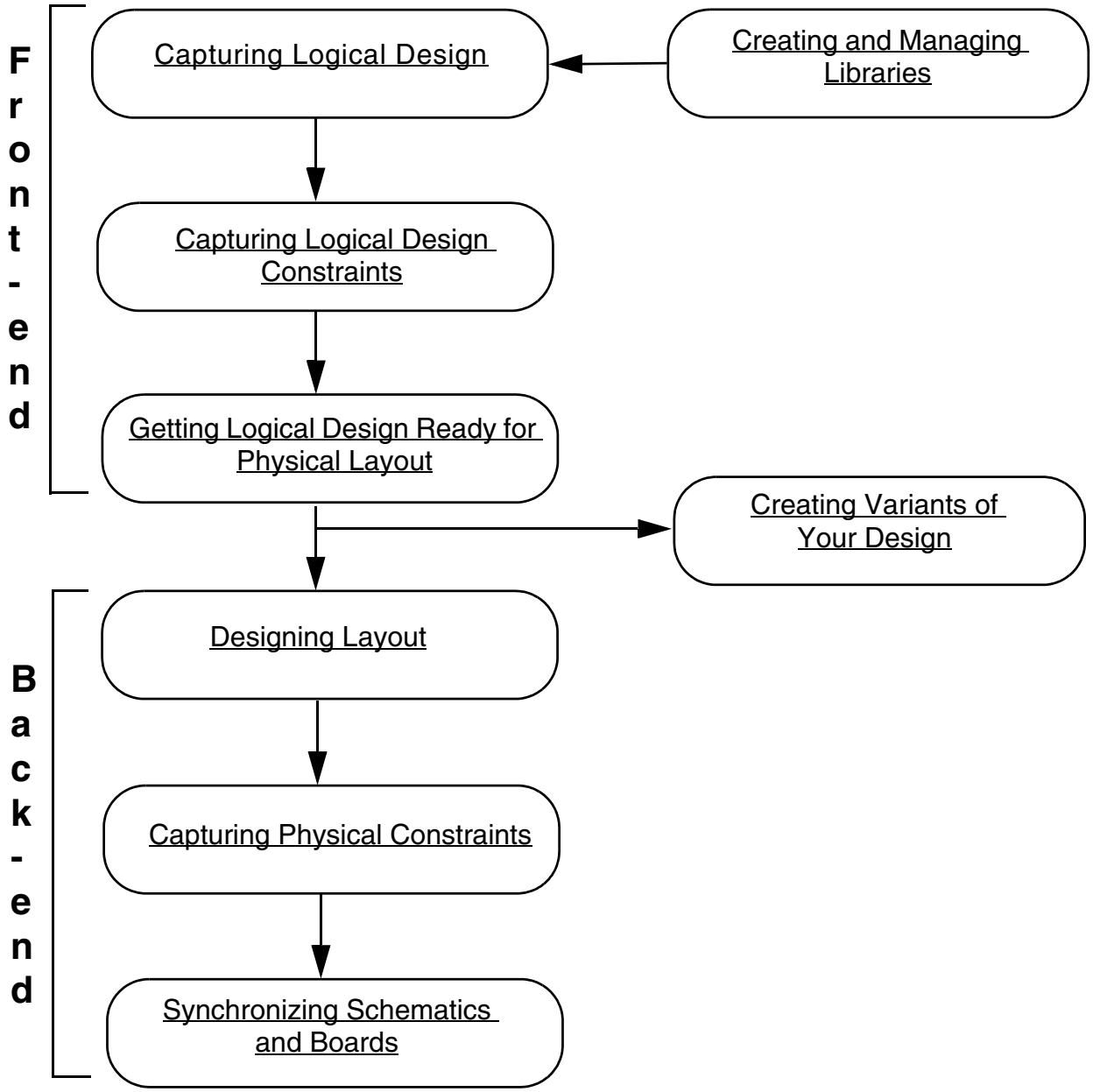
About the Front-to-Back Flow

The front-to-back flow is the sequence of tasks you perform using Cadence tools to design a Printed Circuit Board (PCB). Depending of the stage in the PCB design cycle, these tasks are grouped as front-end and back-end tasks.

PCB Designing is a design solution that integrates PCB tools for creating projects, managing libraries, capturing schematics, packaging, physical placement and routing, and producing manufacturing output.

The front-to-back flow process is depicted in Figure 1-1 on page 8. This flow lists the common design tasks that are required for designing any PCB that are involved in designing a PCB.

Figure 1-1 Front-to-Back Flow



How This Book is Organized

Chapter 1: Creating and Managing Libraries

This chapter explains the basic concepts of libraries. You also learn to create and manage a libraries in Design Entry HDL and OrCAD Capture.

Tools used in this chapter:

- Allegro Part Developer (PDV)

  Part Developer is a tool for creating, editing, and verifying part data.

- Library Explorer

  Library Explorer lets you create, view, and maintain part libraries.

Chapter 2: Capturing Logical Design

This chapter describes how to create a schematic.

Tools used in this chapter:

- System Capture
- OrCAD Capture

  OrCAD Capture lets you draft schematics and produce connectivity and simulation information for PCBs.

- Allegro Design Entry HDL

  Design Entry HDL helps you capture the design of a PCB in a schematic form.

Chapter 3: Capturing Logical Design Constraints

This chapter describes about capturing constraints on a design using Constraint Manager.

Tools used in this chapter:

- Constraint Manager”

  Allegro Constraint Manager is used to manage constraints across all tools in the front-to-back flow.

Chapter 4: Creating Variants of Your Design

This chapter describes how to create variants of your design.

Tools used in this chapter:

- System Capture
- Variant Editor

  Variant Editor lets you create and manage designs that are different from each other by minor differences.

Chapter 5: Getting Logical Design Ready for Physical Layout

This chapter describes how to transfer your logical design to the physical layout tools. You also learn how to update the logical design with the changes on the board.

Tools used in this chapter:

- Packager-XL

  Packager-XL is an interface between the logical design and the physical layout for the Cadence Board Design Solution.

Chapter 6: Designing Layout

This chapter describes how to place and route your board-level design and generate manufacturing output using PCB Editor. This chapter also provides information on how to automatically route your board.

Tools used in this chapter:

- Allegro PCB Editor

  Lets you create graphic symbols that represent packages, mechanical elements, drawing formats, and custom pads

- Allegro PCB Router

  A tool that handles high-density designs requiring complex design rules.

Chapter 7: Capturing Physical Constraints

This chapter describes how to capture constraints on a physical board.

Tools used in this chapter:

- System Capture
- Constraint Manager”

Allegro Constraint Manager is used to manage constraints across all tools in the front-to-back flow.

Chapter 7: Synchronizing Schematics and Boards

This chapter describes how to synchronize your schematic and board-level designs.

1

Creating and Managing Libraries

An electronic design is created by connecting components. The most difficult and time consuming part for designers is creating components. Before adding these components to your schematic, you need to design these components. After you design components, you need to store these into libraries.

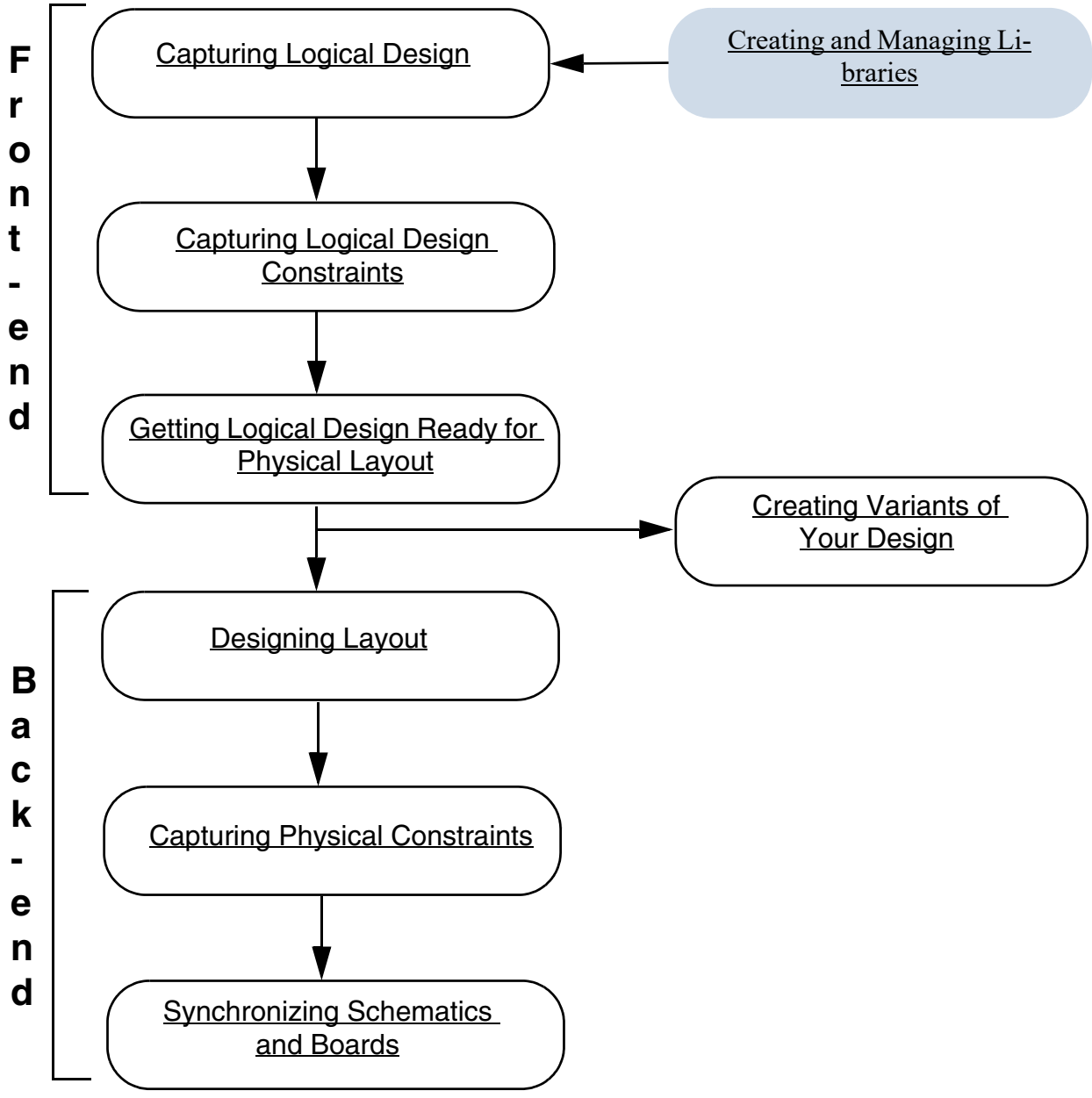
Libraries are a collection of components that enable you to successfully design a schematic using schematic editors such as Design Entry HDL. The libraries consist of a collection of cells that describe:

- Components of a single design.
- Components of the same technology or family. For example, lsttl.
- Common components potentially used in many designs.

The libraries are further organized into separate directories, one for each technology (for example, HCMOS components are in a directory called hcmos). Each library contains many subdirectories, one for each of the components (for example, hc00, hc02). Under each component, there are further subdirectories, such as entity, chips etc. which describes the component in a unique manner.

This chapter covers:

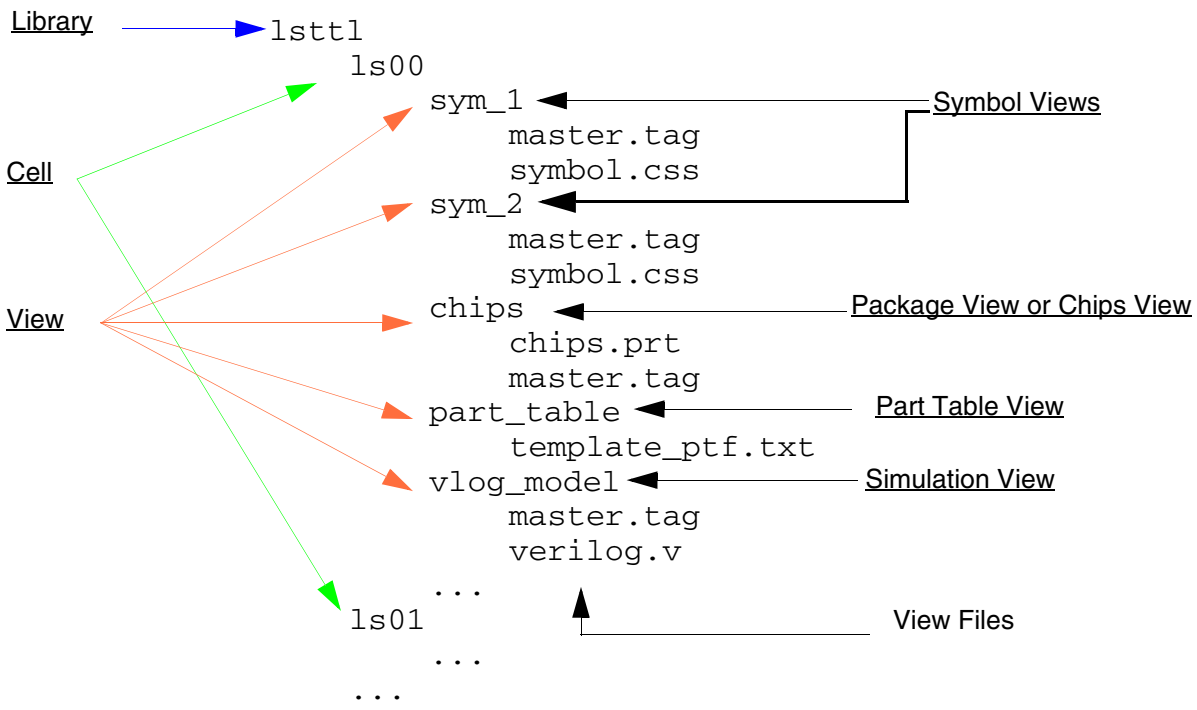
The following figure displays the location of the library management phase in the front-to-back flow.



Allegro Front-end Tools Libraries

The design data is arranged in a *Library – Cell – View* hierarchy.

Library Structure



This figure shows the Library Structure where:

- Each individual library is stored in a directory bearing its name.

  If you have Cadence supplied libraries, the `standard` library resides in the directory `<your_install_dir>/share/library/standard`.

- Under each library, there are one or more cells, each residing in a separate file system directory.

  For example, the files of cells `inport` and `ioport` under the `standard` library reside in directories `<your_install_dir>/share/library/standard/inport` and `<your_install_dir>/share/library/standard/ioport` respectively.

- Under each cell, there are files of different views, each set residing in a separate file system directory.

For example, the files related to the symbol view `sym_1` reside in directory *<your_install_dir>*`/share/library/standard/ioport/sym_1`.

Library

A library is a set of cells that are related in any of these ways:

- Components of a single design (a design library)
- Components of same technology or family
- Common components potentially used in many designs (a reference library)

Cell

A cell is the basic building block of a design. It is a collection of views that describe the functionality and properties of an individual building block of a chip or system.

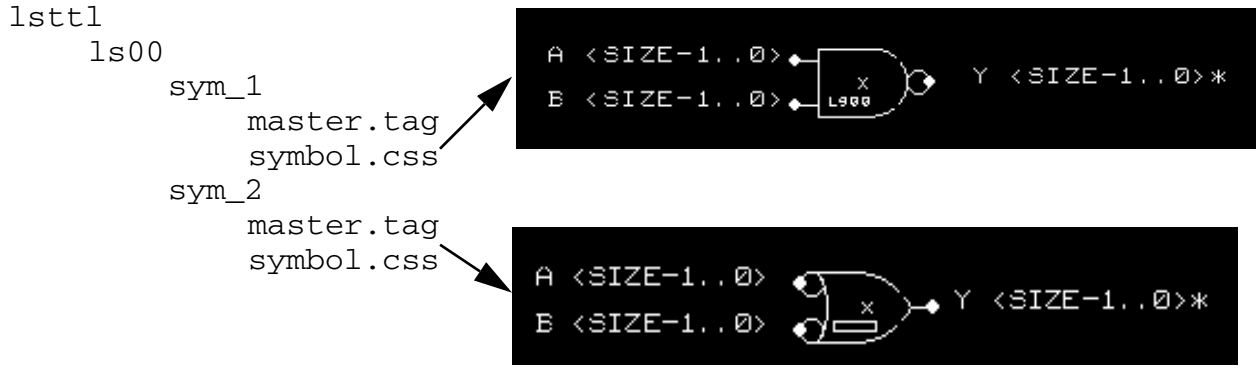
View

A view is a collection of related files that contain information about one type of representation, such as symbolic, schematic, simulation or layout. A cell may not contain all possible types of views.

Symbol Views

The symbol view is the drawing of the logical part. Each part can have one or more symbol views that are in effect different versions of the logical representation.

Figure 1-1 Example of Symbol Views



You need to create these versions when:

- You need different graphical representations as shown in above example.
- You need scalable symbols.

Different versions or symbol views are stored under directories named `sym_1`, `sym_2`, and so on.

The symbol views of a part are created by using Part Developer:

Part Developer lets you edit the schematic symbols, physical pin data, and part table data and create simulation views for multiple parts - all within a single user interface.

Part Developer provides the ability to import part information from a variety of input sources, such as EDAXML and Si2PinPak, to create or modify Design Entry HDL and Capture parts, and to export part data in EDA XML and Capture format. Also, with built-in error checking, Part Developer ensures that symbols are consistent and error-free.

Package View or Chips View

The `package` view or the chip view stores the package information like pin names and electrical information for the part. This view connects the logical view of a component to its physical view.

The pin information like pin names, types, loading and physical numbers is stored in the `chips.prt` file located in `chips` directory.

Part Table View

The part table view has additional physical part information that can be used to customize a part and is located in `ptf` directory. This view is used along with the package view or the chips view while packaging the part.

Simulation View

You can also associate a Verilog or VHDL module with each part used in a design and simulate the entire design using Cadence Simulation products. The simulation view contains all files required for simulating the part.

Creating and Managing Libraries

The Design Entry HDL libraries are maintained using the PCB Librarian toolkit. Along with other tools, this toolkit comprises:

- **Library Explorer**—Lets you create and maintain libraries. Using the familiar Windows NT Explorer interface, you can manage library build areas and launch applications to edit parts and views in a library.
- **Part Developer**—Lets you create and edit schematic symbols, package information (chips), and physical part tables. You can also create parts from XML files and create Verilog and VHDL wrappers/map files.
- **Part Table Editor**—Lets you create and maintain Part Table files.

Library Explorer

The Library Explorer tool lets you create, view, and maintain part libraries. You can use Library Explorer to:

- Create and maintain build areas
- Browse reference libraries
- Import and export files, parts, and libraries to and from the build area
- Create new libraries and cells

  You can create new libraries within the build area and new cells under the library. You can then edit the cells using Part Developer.
- Create and maintain library category files
- Launch Part Developer and other tools to create and edit library parts

  You can select a part or view and launch Part Developer to edit that part. You can also launch Design Entry HDL for editing or viewing symbols, or use the Open function to display a file in the text editor of your choice.

For more information about creating libraries, refer to *Library Explorer User Guide*.

Part Developer

Part Developer is a tool for creating, editing, and verifying part data. It can be launched from Library Explorer once the user has selected an existing part or named a new part. It lets you edit the schematic symbol, physical pin data, and part table data and create simulation views

all within a single user interface. This allows information such as pin names and common properties to be shared across the different views of the component. With built-in error checking, Part Developer ensures that parts are consistent and error-free. Part Developer also has the ability to handle asymmetrical, large pin count and technology-independent parts. Part Developer supports automatic part data entry using XML and adherence to company guidelines using part templates.

For more information about creating and managing parts in Part Developer, refer to the *Part Developer User Guide*.

Part Table Editor

PTF Editor is shipped as part of PCB Librarian, PCB Librarian XL, and the latest packaging configurations of Allegro Design Entry HDL. You can use the Part Table Editor to:

- Create a part table
- Modify a part table
- Verify the part table
- Add new parts to the part table

Physical Part Table File Uses

The Physical Part Table (.ptf) file stores the packaging properties for a part in the library. This file contains part information such as package types, manufacturer names, part numbers, and any custom properties. Each physical part must have an entry in the.ptf file in order to be packaged properly.

In a part table file:

- A unique part number is assigned based on the package style.
- An Allegro package symbol name is assigned based on the package style.

Note: The JEDEC_TYPE property may also be defined in the chips.prt file. However, the part_table view has the priority.

- A part description is added.

For more information about creating part table files using Part Table Editor, refer to *Part Table Editor User Guide*.

OrCAD Libraries

Capture provides more than 80 libraries; in addition, you can create custom libraries. If you edit a library provided by Capture, you should give it a custom name so that you do not copy over your changes when you receive updated libraries. You can, for example, create a library to hold all your programmable logic devices, or hold schematic folders that you use often. There is no need to create a library for a particular project, because the design cache holds all the parts and symbols used in the project.

Since an OrCAD library is a file, you can work with it in the Windows File Manager as well as in Capture. It is recommended that, rather than editing parts in libraries provided by OrCAD, you copy the part and make the changes in a custom library.

Creating a Library

In Capture, you can add as many libraries as required by specifying a name and storage location for each library. Each library is available to each project. The library size is limited only by the amount of space on your system's hard disk; however larger libraries take longer to load.

When you create a new library, project manager adds an empty library to the project. To populate the library, you can create your own parts, or you can move or copy parts from another library.

Saving a Library

The changes you make to a part are temporary until you save the part or the library. When you save a library, you save all the parts and symbols residing in the library. If there are several parts or symbols opened in the part editor, the changes you make to any of them are saved.

For more information about creating and managing libraries in OrCAD, refer to *OrCAD Capture User Guide*.

2

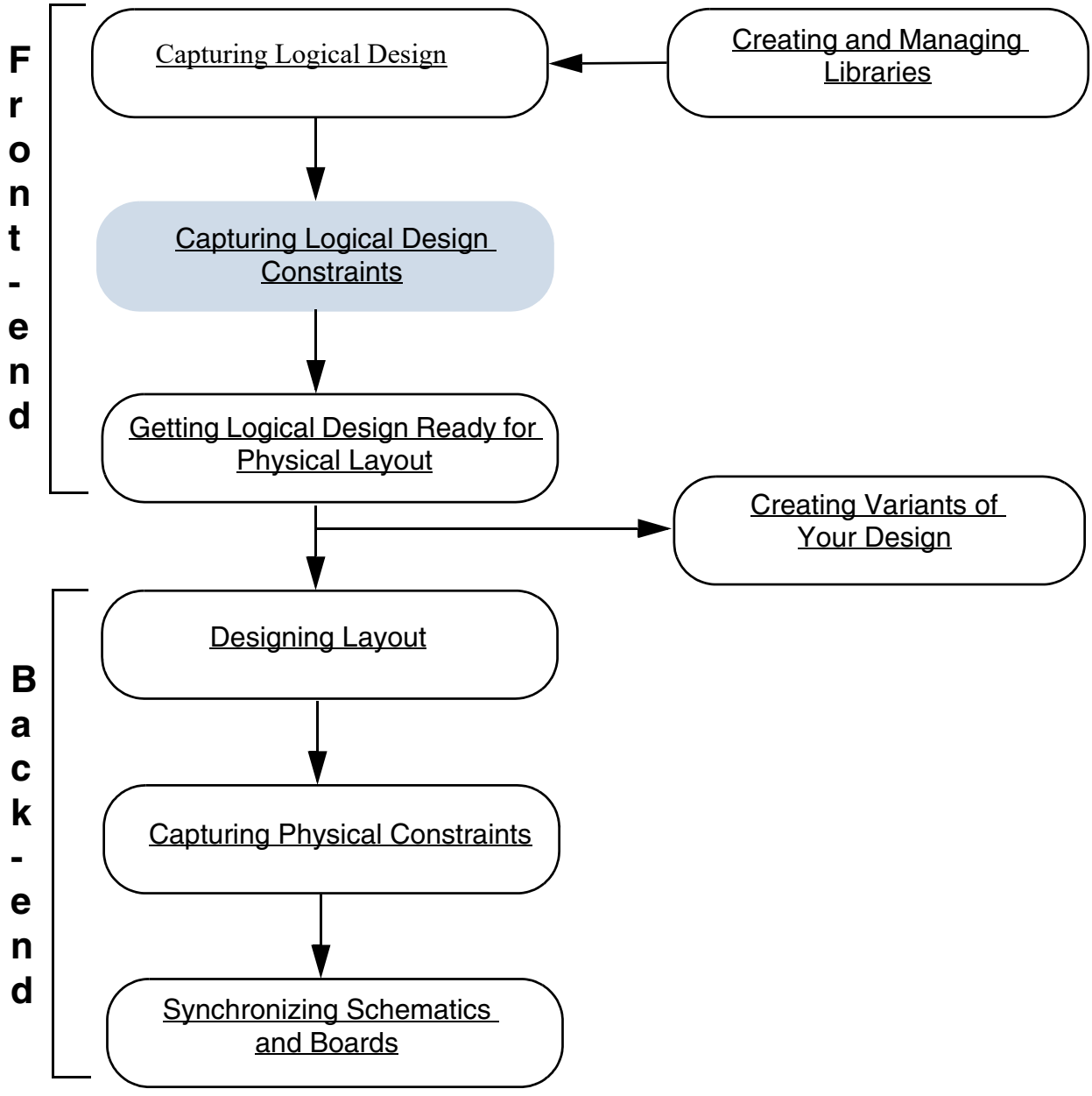
Capturing Logical Design

Capturing a logical design or schematic is representing an electronic circuit as a logical design. When you create a logical design on a design software, you connect a components available in the library which represents a design. After creating a schematic, you perform design related procedures, such as simulate the design and analyze the design simulation results.

Creating Project Using Allegro Tools

Irrespective of the schematic tool you are working with, you start the PCB design process by creating a project. A design project includes paths to libraries, part tables, tool settings, global settings, view directory names, and other related settings for designing a PCB to required specifications. You create project in the schematic tool that is being used to capture the

schematic, Allegro System Capture, OrCAD Capture, and Allegro Design Entry HDL (DE-HDL) to create and set up a new project.



System Capture

Before you start designing schematic, you create a new System Capture project. Projects created in System Capture consist of the same files as in DE-HDL.

When you create a project, a default schematic page is added and displayed. You can add or delete one or multiple sheets simultaneously. When you click a schematic sheet, the sheet opens in the canvas and you can view its contents. You can also preview the contents of a schematic page in a thumbnail without opening the page.

You can also perform other page manipulation operations, such as creating a new page or deleting an existing page from the Project viewer. You can drag and move the pages up and down to change their order in the Project viewer.

Allegro Design Entry HDL

Design Entry HDL is a legacy tool that provides same functionalities as System Capture. It organizes schematic information into pages. It captures and displays only one page of schematic information at a time, and is a by-reference editor because it references all parts in a schematic from various libraries that reside at the reference or local area.

When you create a project using Design Entry HDL or System Capture, a project file called `<projectname>.cpm` is created in the project directory. To create a project, you can also use Project Manager. This project consists of the following:

- Reference libraries
- Local libraries (design libraries)
- `cds.lib` file
- Project file (`.cpm`)

Reference Libraries

Allegro System Capture and DE-HDL references all parts in a schematic from various libraries that are located in the reference or local area. These libraries are called reference libraries. For example, the standard library is a reference library.

The cds.lib File

The `cds.lib` file defines all the libraries used in your schematic design and maps them to their physical locations. When you create a project, a `cds.lib` file is also created which contains:

- A directive to include the installed Cadence libraries. (For example: `INCLUDE <your_install_dir>/share/cdssetup/cds.lib`)
- A define statement that maps the logical project library (projectname_lib) to its physical name (worklib). (For example: `DEFINE myproject_lib worklib`)

The following example shows the contents of a typical `cds.lib` file:

```
DEFINE lsttl ../../library/lsttl
DEFINE memory ../../library/memory
DEFINE 54alsttl ../../library/54alsttl
DEFINE 54fact ../../library/54fact
```

Project File

When you create a new project, a project file called <*projectname*>.cpm in the project directory is created. The <*projectname*>.cpm file includes the following setup information for your project:

- The name of the top-level design and the library in which it is located
- The list of project libraries
- The name and location of the text editor for editing text files from Cadence tools
- The location of the temporary directory where tools generate intermediate data
- Setup directives for individual tools such as Design Entry HDL and Packager-XL, Programmable IC, and PCB Editor
- Directives for customizing the Project Manager (a customized Tools menu or customized flows)
- The current session name

For more information about project creation and setup in Allegro Design Entry HDL, refer to the *Project Creation and Setup* section of *Allegro Design Entry HDL User Guide*.

Creating Project Using OrCAD Capture

OrCAD Capture is a schematic design tool set for the Windows environment. With this tool, you can draft schematics and produce connectivity and simulation information for printed circuit boards and programmable logic designs. It is fully integrated with OrCAD PSpice and other PCB board layout tool set.

A project in OrCAD Capture refers to the collection of design file, part libraries, report files, and other associated materials that exist, as a set, within the environment.

When you create a project using OrCAD Capture, a design is immediately created with a project file `.opj` which contains details about the design. However, you also have the option of creating a design without first creating a project.

The project file consists of:

- Pointers for interacting with the design file (`.dsn`) file
- Other referenced files
- Outputs reports associated with the design file
- Information about libraries and VHDL files.

When the project is first created, the project manager creates a design file with the same name as the project. It also creates a schematic folder within the design file, and a schematic page within the folder. You can create a new design to replace the design created by the project manager.

For more information about project creation and setup in OrCAD Capture, refer to the *Working with Projects* section of *OrCAD Capture User Guide*.

Creating a Schematic

Using Cadence's schematic design tools, you can easily create a logical design and apply design procedures. The tools are used for capturing a logical design in the printed circuit board (PCB) design flow are:

- Allegro System Capture
- OrCAD Capture
- Design Entry HDL (DE-HDL)

With these tools, you can create a project, place parts (components), connect parts, name signals, add ports, and generate parts.

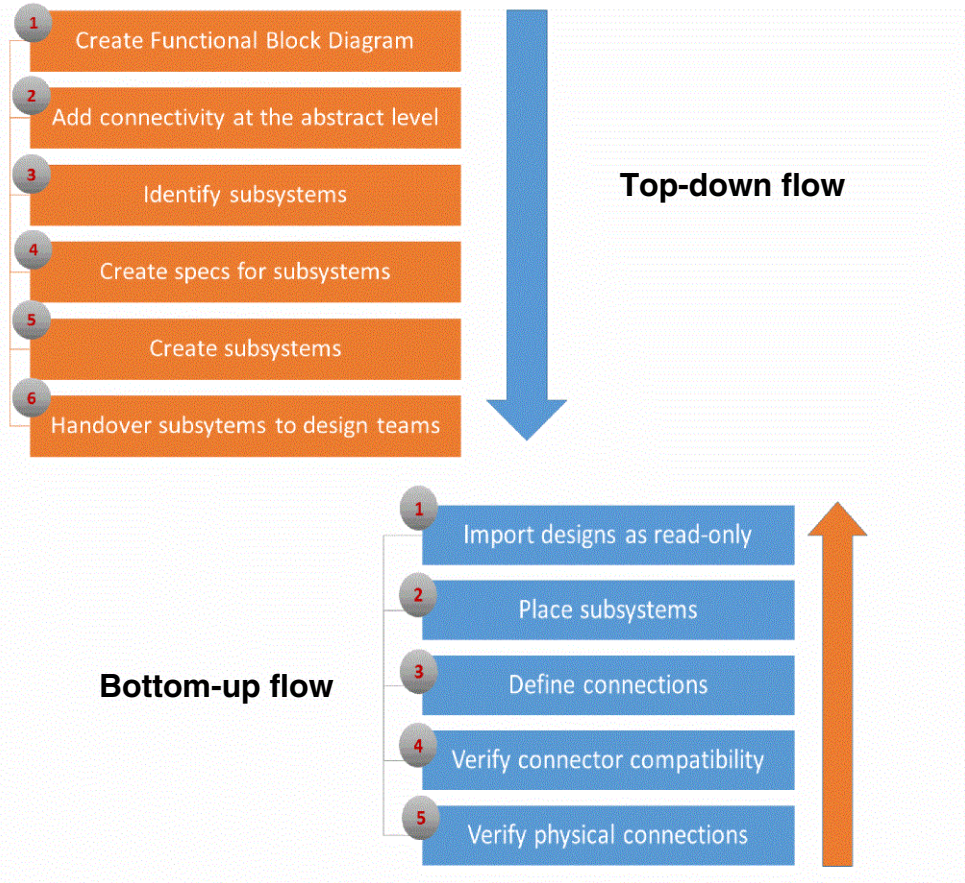
Designing Schematic Using System Capture

Allegro System Capture helps you design a PCB in a schematic form and add constraints on it. The schematic design is passed to the layout tool, Allegro PCB Editor to place and route the board-level design. Any changes made in the layout are then brought back to the schematic.

System Capture for PCB system design

System Capture can also be used for system-level designing. The objective of system design with System Capture is to enable design teams to quickly put together parts of the system that they know, create logical parts of the design, 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.

System Capture supports following system design flows:

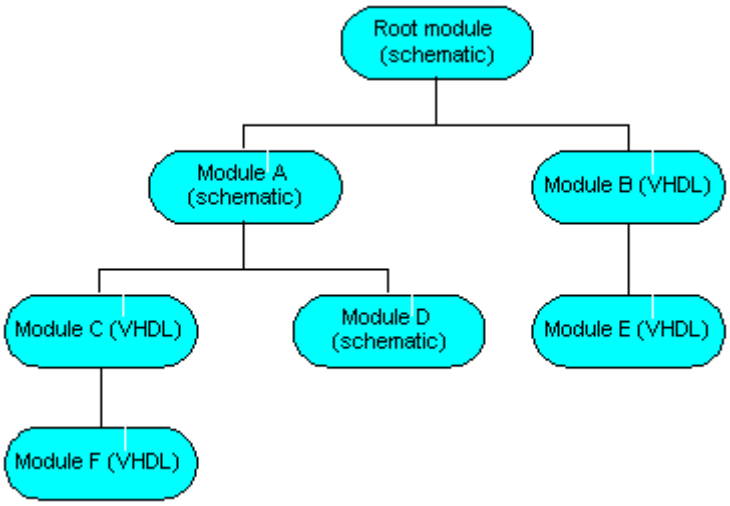


Creating Schematic Using OrCAD Capture

Most of the electronic design projects have multiple PCB design. The multiple designs are connected to each other and create a complete electronic design.

OrCAD Capture provides the means to create electronic designs in two media: as schematics or as VHDL models. Schematic designs can include VHDL or Verilog models (one or the other, not both) as lower level hierarchical modules, but these models can only instantiate other models (of the same type) at lower levels in the hierarchy.

Consider the following illustration:



Any schematic design module can include either schematics or VHDL/Verilog models as instantiated components. However, VHDL/Verilog design modules are limited to other modules of the same type as instantiated components. Hence, if the root module of your design is a VHDL model, all lower level modules must also be VHDL models.

A design file also contains a design cache, which is like an embedded library — it contains a copy of all the parts and symbols used on the schematic pages. You can create a new design file to replace the design created by the project manager.

Using the design variants capability of OrCAD Capture, you can also manage unlimited board assembly variations without having to maintain duplicate schematics or manually edit individual BOMs. This capability reduces the number of files by maintaining all design assembly variations within a single design and outputs. On the schematic canvas, substituted and/or unplaced components within each assembly are displayed through graphical indicators for easy reference.

For more information about creating logical design using OrCAD Capture, refer to the *Working with Designs* section of *OrCAD Capture User Guide*.

Capturing Logical Designs using DE-HDL

Design Entry HDL supports creating Flat, Structured, and Hierarchical designs. Design Entry HDL allows you to:

- Create a schematic (Flat, Structured, or Hierarchical)
- Manage a design with multiple users

Note: For detailed information about Design Entry HDL, refer to *Allegro Design Entry HDL User Guide* and *Allegro Design Entry HDL Tutorial*.

Usually, the creating a small design is done using the flat design technique.

Flat Designs

The flat design method is the most straightforward technique for creating a design with the Cadence system design tools. In a flat design, all parts on the drawing come from Design Entry HDL or user-defined libraries and are one-to-one logical representations of the physical parts. The entire interconnecting wiring within the design is entered pin-to-pin.

Structured Designs

The structured design method facilitates the entry and analysis of sophisticated designs that make use of bused signals, memory and, register depth. A structured design minimizes the number of interconnections and parts on the schematic.

Hierarchical Designs

The hierarchical design technique is an efficient approach to developing complex designs that can be organized into modules. This method is useful for designs that re-use many of the same circuit functions and for isolating portions of the design for teamwork assignments.

Managing Designs with Multiple Users in DE-HDL

Using the Design Management option in DE-HDL, you can enable structured team design for a design to shorten your design cycle. You can manage the design on a file system, in a folder on SharePoint, or in a folder in PTC Windchill.

This functionality enables designers to manage a multiblock hierarchical, or flat, design with multiple users concurrently modifying portions of the logical and physical design. This helps design teams reduce the overall design time. Groups of engineers work together on individual

portions of the design to develop a system and control how the portions of the system are integrated together.

With this feature, you can also:

- Prevent unintended modification of a design (control access)
- Provide version control and version history for all design changes
- Manage design modifications from multiple sources/sites
- Facilitate communication/notification among the design team
- Maintain the design data at a central location
- Integrate all design components into a released design

For more information about capturing a logical design using Design Entry HDL, refer to *Allegro Design Entry HDL User Guide*.

3

Capturing Logical Design Constraints

A constraint is a user-defined requirement applied to a net or pin-pair in a design. For example, you can capture a constraint to define the maximum voltage overshoot tolerated by a net and capture the minimum first switch delay for a driver-receiver pin-pair in your design.

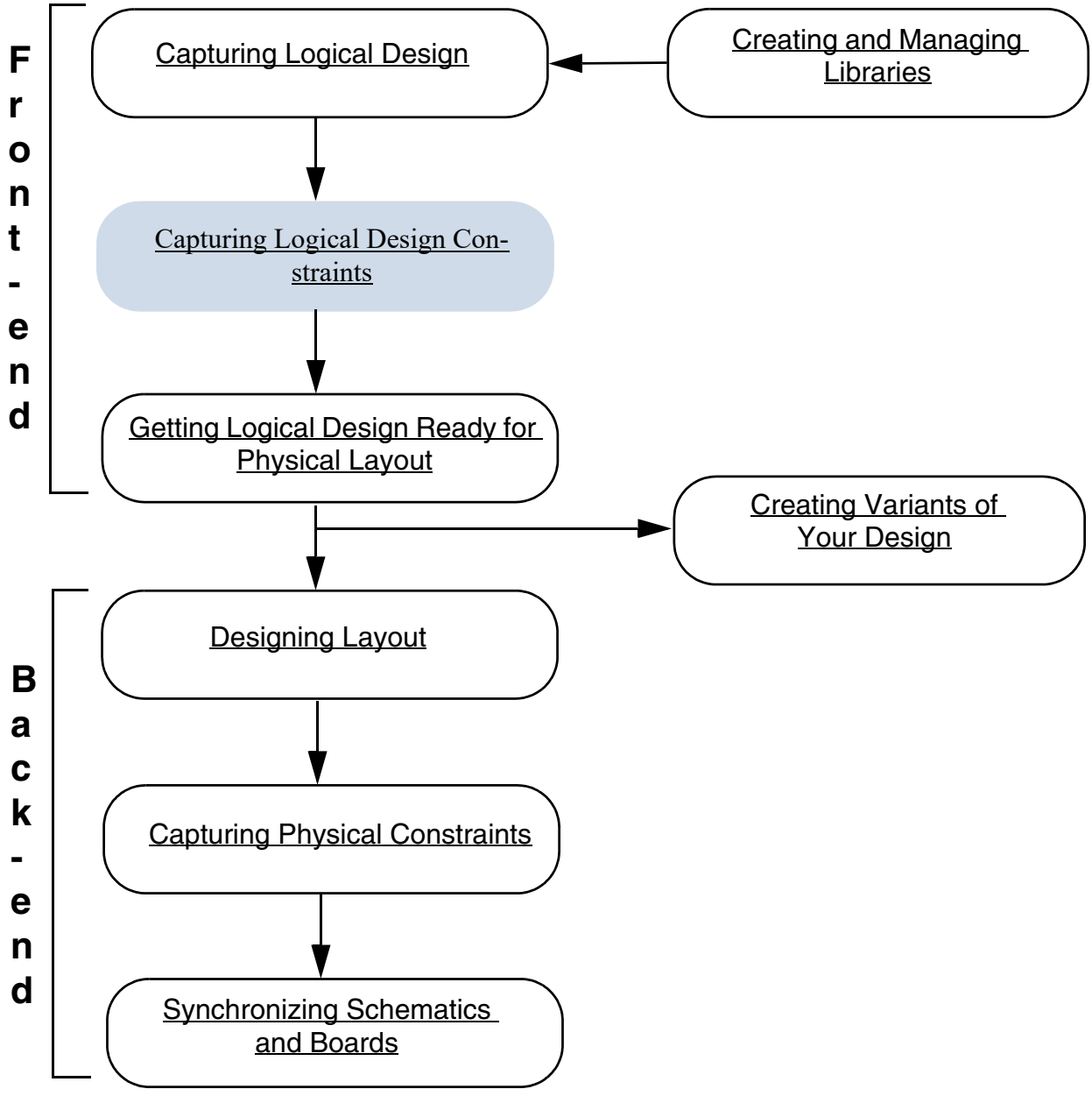
Cadence PCB design flow, provides a dedicated tool, Constraint Manager, for capturing and managing constraints. This tool is seamlessly integrated with schematic design tools as well as physical layout design tools. You can use Constraint Manager with design capture tools, Design Entry HDL or OrCAD Capture, to capture and manage constraints as you implement logic. Constraint Manager is well integrated with these tools, therefore, the changes that you make to constraint information are displayed in these tools. Similarly, the changes that you make to constraint information in these tools are displayed in Constraint Manager.

Why Constraint Manager?

Using Constraint Manager to capture design constraints has the following advantages:

- It provides a spread-sheet based user interface that allows you to quickly capture, modify, and delete constraints.
- It supports syntax checking for all constraints.
- It supports constraint inheritance. The constraints captured on a schematic block are inherited by the design in which the block is instantiated.
- It lets you create Electrical Constraint Sets (ECSets) - collection of electrical constraints that define a particular design requirement and assign them to objects on which you want to capture the same set of constraints. For example, you can create an ECSet to define the default timing and noise tolerance for a net. An Electrical Constraint Set applies to an

individual net although other nets may contribute to the measure of the constraint (for example, to crosstalk).



Using Constraint Manager with Design Capture Tools

You can create and modify electrical constraints in Constraint Manager connected to System Capture, DE-HDL or OrCAD Capture.

All the constraints are stored in the Constraint Manager Database (CMDB) and saved in the dictionary and constraints file *designname*.dcf.

Constraints in System Capture

Allegro System Capture provides a docked Constraint Manager that uses the same database as Allegro Constraints Manager. You can edit constraints on the currently selected object(s) on the canvas, and capture all the Electrical constraints, as well as apply all types of Constraints Sets from the docked Constraint Manager.

You can also access Allegro Constraint Manage from System Capture and:

- Enable Auto-Creation of Differential Pairs
- Automatically Create XNets without DML Models
- Extract Topology for an XNet in SigXplorer
- Create ECSets based on Extracted Topology
- Import ECSets

For more information about constraints in System Capture, refer to the Managing the High Speed Constraints Flow section of the *Schematic Design using System Capture* guide.

Capturing Constraints in Design Entry HDL

Sharing Constraint Information with Physical Design Tools

The data flow from the logic design capture tool, to physical layout tool is referred to as front -to-back flow. For sharing data with the physical layout tool, design capture tools use the *Export Physical* command. Using Export Physical command generates a set of package files that contain information about electrical constraints and the netlist.

In this flow, Constraint Manager stores information about constraints in a file named <*root_design_name*>.dcfx (<*root_design_name*>.dcf), which is created in the sch_1 folder in a ZIP format. The .dcf file is a binary file, which, when extracted, creates

multiple XML and text files that contain information about constraints and properties that are applied to objects in a design. For more information about viewing a `.dcfx` (`.dcf`) file, refer to the *Viewing a DCF File* section of *Allegro Design Entry HDL Reference Guide*.

For more information, refer to the *Allegro® Design Entry HDL - Constraint Manager User Guide* for Constraint Manager connected to Design Entry HDL

Capturing Constraints in OrCAD Capture

To specify constraints, you can use the existing mode to manage a subset of constraints using the property editor.

The Constraint Manager-enabled mode is new and is optional. You can enable this mode at any phase of the PCB design flow and on any of the following:

- New schematic design
- Existing schematic design
- Existing schematic design with PCB layout

In the Capture-Constraint Manager flow, Constraint Manager is used to define, manage, and assign constraints on the Capture schematic.

The recommended sequence of tasks to manage design constraints using Constraint Manager is:

1. Complete the logical design.
2. Add electrical constraints in Constraint Manager.
3. Create or update the PCB layout.
4. Update electrical, physical, and spacing constraints in the PCB layout.
5. Backannotate logical design to synchronize constraints.

You can now specify the physical and spacing constraints in Constraint Manager in Capture.

Note: You should avoid simultaneous editing of design objects, such as nets with while assigning and modifying constraints.

For more information, refer to:

- *Constraint Manager with OrCAD Capture* for Constraint Manager connected to OrCAD Capture.

4

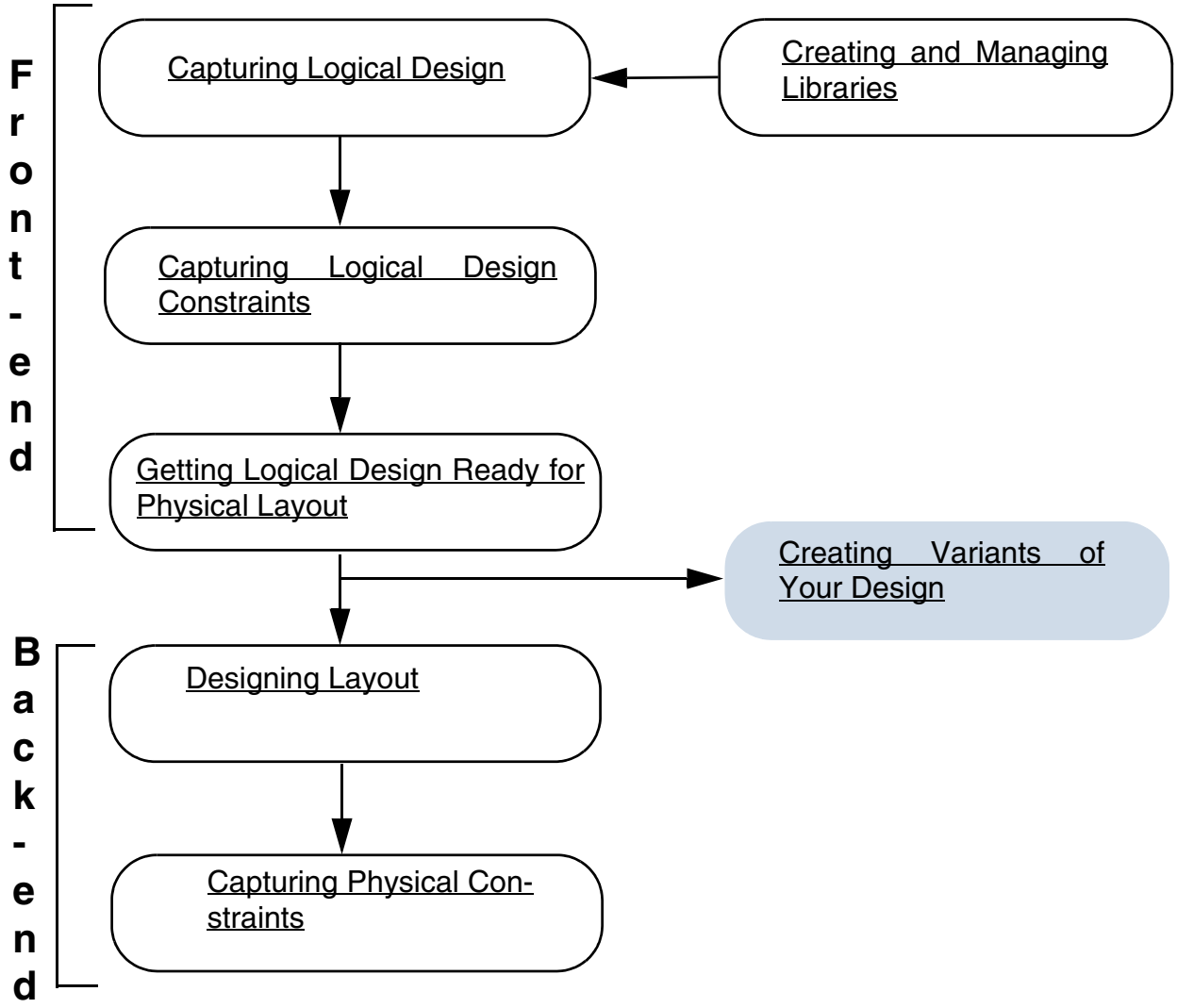
Creating Variants of Your Design

In today's market-place, there exists a need to create designs that share a common set of core elements and that vary because of minor differences. Requirements of targeted market segments or destination country or small changes in feature set often cause these differences. To understand these differences, let's consider two examples.

- Example 1: Two designs, Europe and US, share a common set of components. The only difference is in the resistor R1 that has different values, 10K for Europe and 5K for US.
- Example 2: Two designs, Japan and India, share a common set of components. The only difference is that IC U5 is present in Japan and is absent in India.

Even if there is a difference in only one component in two designs, each design is considered a new product. The individual designs require a new assembly with a unique bill of materials and documentation. If these designs have a change in footprints, they may require separate assembly process.

To manage such variants in the PCB design, you use System Capture and Variant Editor. These tools let you create and manage different variants of a base design that are different from each other by small differences.



Managing Variants in System Capture

System Capture supports creation and management of design variants on schematic sheets. You can create multiple variants of a base design and modify the components in the base schematic for use in the variants.

When you create a new variant in System Capture, the tool automatically switches to the Variant view for the newly created variant so that the variant data can be edited.

For more information about creating and managing variants using System Capture, refer to the Managing Variants in System Capture in the *Schematic Design using System Capture*

Creating Variants using Variant Editor

Using the design variance solution is simple. All you have to do is to create the base design in Design Entry HDL and then define the variant component in Variant Editor.

Variant Editor supports an intuitive user interface (UI). Without resorting to complex editing of text files, you can define variant components, generate Bill of Materials (BOM) reports, annotate special designators to any components, annotate variant data, and merge variant databases using Variant Editor.

Variant Editor allows you to:

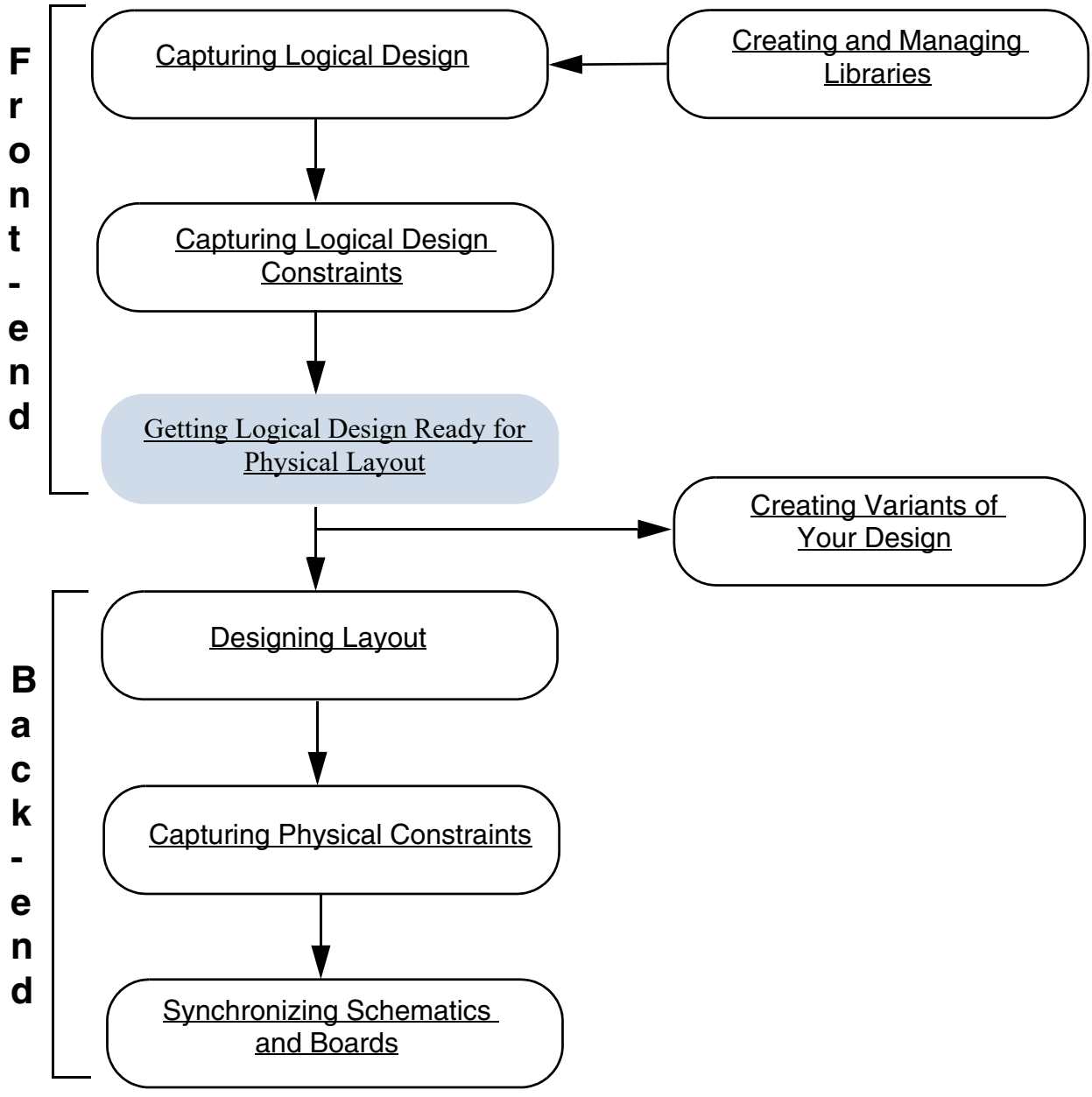
- Use the Physical Part table (PPT) driven data for defining variant component values.
- Generate the Bill of Material (BOM) that reflects the electrical *stuff list* for a variant.
- Generate a delta list of components from the base design for a variant.
- Generate a comparative BOM of different variants.
- Generate BOM reports in multiple formats, such as spreadsheet format and HTML format.
- Annotate variant data from the variant database into the schematic.
- Generate the interface file that is read by PCB Editor to create variant assembly drawings.
- Cross-probe with the Design Entry HDL.
- Support associated mechanical parts, callouts, and global find for specific components.
- Synchronize the changes made in the variant database with the changes made in the original schematic.
- Replace an existing component with another component that has a different name or a non-compatible footprint.

5

Getting Logical Design Ready for Physical Layout

Once the schematic capture process is completed, the next step is to get the design ready for PCB layout. Before pushing the design into the PCB domain, it is important to verify the design and ensure that everything runs smoothly in PCB Editor.

For this, you run Design Rules Check (DRC) to check for proper placement of elements on the drawing, consistency between the logic and body drawing, properties and property values, unconnected elements, and invalid names.



Preparing a System Capture Design for Physical Layout

System Capture lets you run design rule checks (DRCs) to identify connectivity and other errors in the design.

System Capture provides a standard set of DRCs, and also lets you write your own custom DRCs in the Tcl language and run the custom DRCs. DRCs, such as the property overlap DRC, are controlled by the values set for the variables in the overrides value TCL file.

You can enable or disable individual DRCs. When you run DRCs, the DRC errors are reported in the Violations window. You can also set the error severity level -- Error, Warning or Information -- to be reported in the Violations window if an enabled DRC fails. For example, if you set the error severity level for the Unconnected Nets DRC as Warning, System Capture displays warning messages for each unconnected net in the Violations window.

Preparing a DE-HDL Design for Physical Layout

Allegro Design Entry HDL Rules Checker lets you check for violations of design-related rules in a design as well as obtain information about various aspects of your design. It includes fifteen sets of rules and design checks that let you know early in the design cycle if you are violating design rules.

Using this tool, you can also check rules in logical and physical environments. These rules check for fan-in and fan-out errors, load errors, required properties and property values, unconnected elements, naming requirements, power requirements, cost requirements, and so on.

The first time you run Rules Checker on your design it opens with the default settings. When you exit Rules Checker, it creates an initialization file (`CheckPlus.ini`) that contains the following information:

- Last selected environment
- Selected and deselected rules
- Expanded and unexpanded rules

Note: For detailed information about Rules Checker, refer to *Allegro Design Entry HDL Rules Checker User Guide*.

Exporting a DE-HDL Design

After performing design rules checks, you need to package your design to translate the schematic into a physical design. Packaging involves converting a logical design into a physical layout and vice versa. The utility that completes packaging is Packager-XL. Packager-XL is the interface between the schematic and the board for the front-to-back flow.

You can use this utility to do the following:

- Translate the schematic into a physical design
- Backannotate the changes made in the board to the schematic
- Update the changes made in the schematic after initial packaging to the board

Packager-XL works in the following two modes:

Forward Mode

In the Forward mode, Packager-XL translates a logical design entered in Design Entry HDL into a physical design ready for layout in PCB Editor. To run the Forward mode, you need to run the Export Physical command.

In the Forward mode, you enter a design in Design Entry HDL, and then run Packager-XL to translate the logical design into a physical design. This process is also known as packaging the design into physical parts. To incorporate incremental design changes into the existing physical design, you can use subsequent Packager-XL runs. To import the packaged design into the PCB Editor environment, you use the PCB Editor Import Logic program.

Preparing an OrCAD Capture Design for Physical Layout

Before converting your design for PCB Editor, it is a good design practice to verify your design by running Design Rules Check for validating physical rules.

When you run the Design Rules Check tool, errors are marked on your schematic pages. Warnings are also marked, provided, the Create DRC markers for warnings check box was selected before running the design rules check.

When you run the Design Rules Check tool, Capture creates a report (.DRC) of warning and error messages. You can view the report in a text editor. These messages also appear in the session log.

In addition to the report, the Design Rules Check tool places error and warning markers on the schematic pages as well.

Note: For detailed information about Design Rules Checker in OrCAD Capture, refer to *OrCAD Capture User Guide.*

Exporting an OrCAD Capture

Annotating the Design

By annotating your design (that is, by assigning reference designators and net names to unnamed parts and electrical connections in your design), you provide the means by which to pass it "downstream" to other layout design tools (PCB Editor, for example) that take it beyond the schematic capture phase of the design.

The procedures involved in converting your logical design to physical layout are:

- Customizing Part References in a Design

  You can customize the way Capture assigns part references in your design. You can specify a range of part reference values that Capture will use to annotate a schematic page or a hierarchical block in your design.

- Backannotating

  When you need to transfer packaging information to your schematic folder from other EDA tools, use the backannotate tool. When you need to backannotate properties, use the Update Properties tool (see To update part or net properties). Using Backannotate, you can import changes created by external tools such as PCB layout packages. Capture uses a simple file format to provide support for gate swapping, for pin swapping, and for changing or adding properties on parts, pins, or nets. If the external tool creates a backannotation file, edit the file to match the format described in Designating pins, gates, or packages for swapping.

  Backannotating board file information to your schematic design is a matter of creating a report file and reading it back into Capture.

- Annotating Schematic Information

  If you make changes to your design in Capture, you can bring those changes into PCB Editor. In addition, you must save your Capture design before you can create a netlist.

  To forward, annotate schematic information from Capture. Capture includes functionality with which you can forward annotate your schematic data, such that it can be included in a PCB Editor board design.

You can choose the sequence in which the components of your design are annotated.

The Annotation Sequence list contains three options that you can use to decide the sequence in which the objects on your design are annotated — Default, Left to Right, and Top to Bottom.

- Designating Pins, Gates, or Packages for Swapping

  For PCB designs, a swap file is a text file containing old and new part references for use with the Backannotate command. Swap files are typically created by another application, such as PCB Editor. You can also create a swap (.SWP) file using any text editor that saves files in the ASCII format.

  When you are creating a swap file, include only the changes from the present state of the design to the state you want it to have. For example, you might place a part as U1 in the design, and change it in a PCB layout package first to U2, then to U3. The swap file should reflect the change from U1 to U3; do not include the intermediate step involving U2.

  For gate swaps, ensure that the gates being swapped are of the same type. If they are not, you may get incorrect results.

  For pin swaps, an additional element — the part reference — must be specified before the old and new values. Pin swap is limited to pins of the same type and shape on the same part. For example, you can swap data pins on U5B, but you cannot swap a pin on U5B with a pin on U5C.

- Creating an update file

  The update file is used by the Update Properties tool to determine which objects to change, which of the objects' properties are affected, and what values those properties receive. You can create an update (.UPD) file using any text editor that saves files in ASCII format. The file can include comments; any text to the right of a semicolon is ignored by the Update Properties tool. Strings in the update file (except for comments) must be enclosed in quotation marks and cannot exceed 124 characters. You can use spaces and tab characters to format the update file in rows and columns, as shown in the example below.

- Creating a Combined Swap and Update File

  You can create a file that combines the swap file and update file information. Run Backannotate to use a combined swap and update file. Swap and update files should have the same `.SWP` file extension as normal swap files.

Packaging Your Design (Advanced)

This section contains the following information which you can use to package your design in System Capture, Design Entry HDL as well as OrCAD Capture:

Generating a Bill of Materials

You can use the BOM-HDL tool to generate bill of materials reports. A bill of materials report consists of three basic sections:

- Report parameters

  Contains parameters, such as the report title, date, and template file used to generate the report.

- Physical parts

  Contains properties for each physical part in the schematic. Any property attached to an instance or package can be included in the bill of materials report.

- Callouts

  Contains callouts that describe any parts, such as ejectors, stiffeners, or mounting hardware that are required on the manufactured product, but are not graphically represented on the logical schematic.

Updating the Schematic With the Changes in the Board

You can use the Import Physical dialog box to update the schematic with the changes in the board.

Import Physical by default runs in Constraint Manager enabled flow. If Constraint Manager has not been used to edit electrical constraints in Design Entry HDL.

You can overwrite all existing electrical constraint information in the *schematic* with the electrical constraint information currently available in the *PCB Editor Board File*. You can also import only the electrical constraint information that has changed in the *PCB Editor Board File* since the last import and overwrite such constraints in the schematic.

The constraints in the schematic are synchronized with the constraints in the board. If you now start Constraint Manager from Design Entry HDL, all the electrical constraints that you captured in Allegro PCB and Package or Allegro SI will appear in Constraint Manager.

Passing Properties from the Layout to Schematic

While updating the schematic in Design Entry HDL with the changes in the board file in PCB Editor, you can use Packager Setup to specify which of the properties need to be modified and which properties need to be added to the schematic.

You can make the following changes:

1. Define the feedback properties.

   By default, all feedback properties replace the corresponding properties in the schematic. You can, however, specify that certain feedback properties will not replace (win) the properties in the schematic.

   - Add a property in the *No Feedback Properties* list box to insure that it will not replace the schematic property.
   - Click *Remove* in the *No Feedback Properties* list box to delete any property from it. A property removed from the *No Feedback Properties* list box is fed back to the schematic.

2. Run Packager-XL in the feedback mode.

   By default, the *None* radio button is selected signifying that Packager-XL will run only in the forward mode.

   To run Packager-XL in the feedback mode, click either the *Allegro PCB Editor* radio button or the *3rd Party* radio button to specify the source of feedback files.

3. Annotate properties.

   You can define the objects in the design that should be backannotated. You can select body, pin, net, or physical net name for backannotation. To choose any object for backannotation, select the respective check boxes under the *Options* radio button.

4. Manage hard properties.

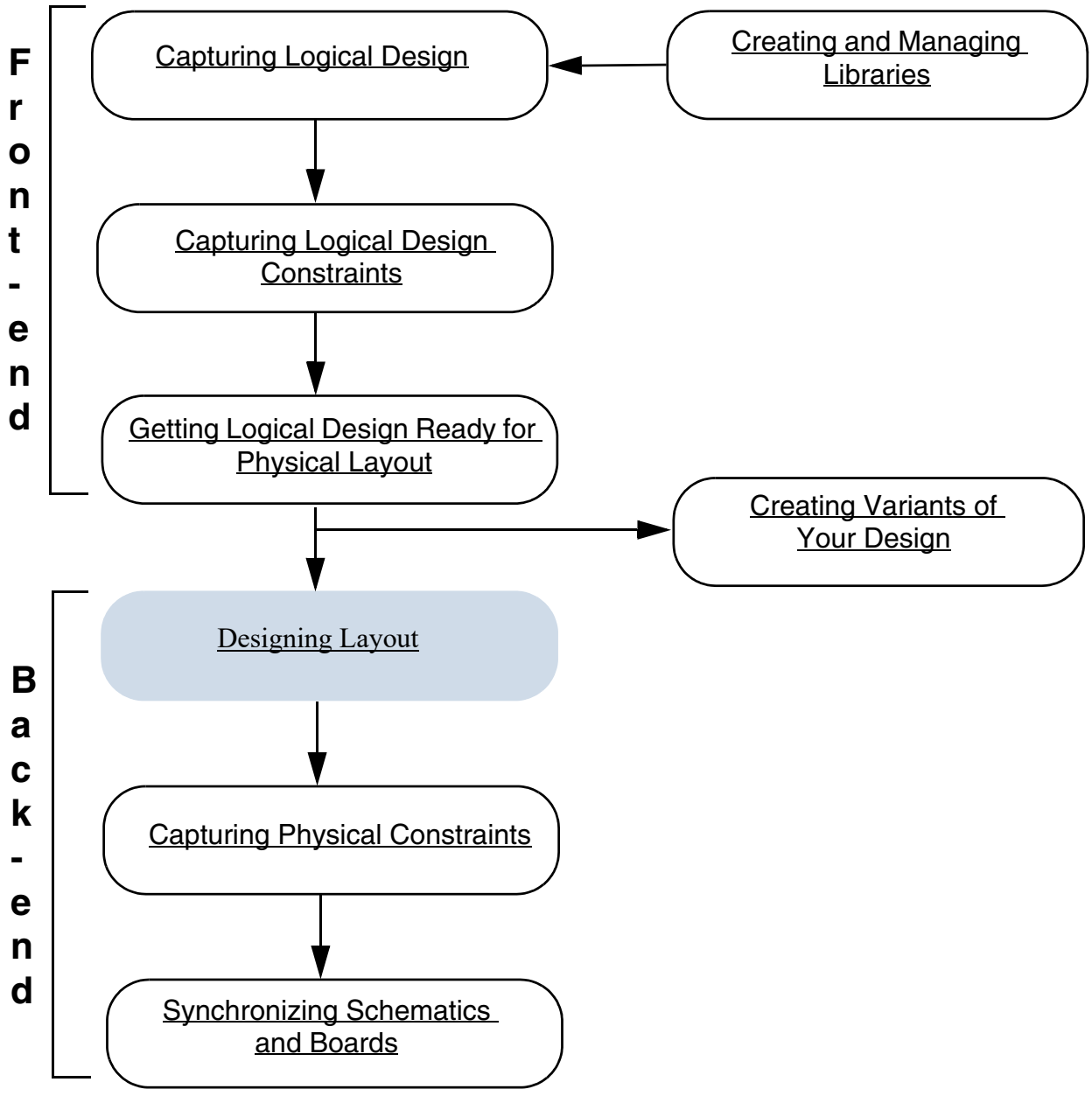
   You can manage the packaging of hard properties (user-defined properties) by selecting the *Do not Update Hard Location, Section and Pin numbers on schematic* check box. By default, Packager-XL updates only soft properties in the feedback mode.

5. Accept the changes.

You have specified the properties that will be passed from the layout to the schematic.

6

Designing Layout



PCB Editor is an intuitive, easy-to-use, constraint-driven environment for creating and editing simple to complex PCBs. Its extensive feature set addresses a wide range of design and manufacturability challenges.

Cadence's Allegro PCB Editor helps you perform the major board tasks of design, including:

- Floor planning and Placement
- Routing
- Constraint Management
- Placement Replication
- Multi-Line Routing

Physical Layout using OrCAD PCB Editor

After packaging the logical schematic, the next step is to design the physical layout. Native design logic refers to design data extracted from the logic design tools that create schematics or logic design files, that interface with the layout editor.

Following are the tasks that are required for designing the physical layout phase of PCB Flows:

- Creating a Board Outline
- Creating Symbols and Padstacks
- Importing Logic
- Defining Layers
- Defining Constraints
- Placement
- Routing
- Design Synchronization

Creating a Board Outline

The board outline defines the area within which the components in your board will be placed and routed. A board outline can be of any closed shape.

Creating Symbols and Padstacks

Before you import your logic design from logic design tools into Allegro PCB and Package, you need to create symbols and padstacks. A symbol is a set of data that can be used to represent any design element. A padstack is a file that contains information for each layer in a design.

- Package/Part (.psm)
- Mechanical (.bsm)
- Format (.osm)
- Shape (.ssm)
- Flash (.fsm)

Once you create a symbol, you can save it in a library so that you can reuse the symbol. After you load a library symbol into your layout, Allegro PCB and Package uses that symbol definition for future instantiations. If that symbol does not exist in your layout, Allegro PCB and Package looks for the symbol in the library.

Importing Logic

The Import Logic dialog box is displayed when you run the netin command. It is the interface from which you load the logic for your design into the Allegro PCB and Package database and establish the operating characteristics for the netrev utility.

The transfer of logic from logic design tools is governed by the constraint Manager-enabled flow.

Defining Layers

A layer is an insulated plane in the design that contains lines of etch. The ordered stack of layers in your design is also called the cross section.

Before you begin placement and routing, you define layers and their various characteristics as part of setting up your layout. While you place and route, you might need to insert extra

routing layers if the design is too dense to complete or you might need to delete layers because of an Engineering Change Order (ECO).

The layout cross section consists of the ordered layers of your layout, including the information about their type, thickness, electrical behavior, and shielding. You also specify whether to photoplot positively or negatively when you set up your cross section.

For more information, refer to the Capturing Physical Constraints chapter.

Defining Constraints

When you start a new design, you set design rules and constraint values that conform to your design rules. You can set the values using the constraint dialog boxes.

A constraint is a user-defined limit applied by design rule checking (DRC) to one or more physical elements in a design. When you define and apply a constraint value, Allegro PCB and Package adheres to that constraint in automatic and interactive processing and flags violations with DRC markers. You can think of a constraint as a bundled set of properties.

Allegro PCB and Package provides a set of predefined design rule types, each with its own descriptive name; for example, Line to Pin Spacing and Minimum Line Width.

Placement

Allegro PCB and Package provides a variety of interactive and automatic features for placing components and swapping pins, functions (gates, inverters, or logical elements within a packaged component), and components.

You can use interactive placement to place all components individually, or you can place components of the same type during one pass.

In automatic placement, Allegro PCB and Package places components automatically based on controls that you set before starting automatic placement and by assigning certain placement properties that restrict or influence component positioning and part packaging.

Routing

The next phase in the PCB flow after placing the components on the board is routing the connections.

The Allegro PCB Designer Routing Option is tightly integrated with the PCB Editor. Through the Routing Option interface, all design information and constraints are automatically passed

from the PCB Editor. Once the route is completed, all route information is automatically passed back to the PCB Editor.

Routing is the process of making electrical connections in your design. You can route interactively or automatically. You can also specify the order in which pins are routed on a particular net.

Increased design complexity, density, and high-speed routing constraints make manual routing of PCBs difficult and time-consuming. The challenges inherent in complex interconnect routing are best addressed with powerful, automated technology. The robust, production proven autorouter includes a batch routing mode with extensive user-defined routing strategy control as well as built-in automatic strategy capabilities

Design Synchronization

In the design synchronization phase, you bring the schematic and the PCB Editor board in sync. You resolve both property and connectivity changes between the schematic and the board.

Need for Synchronization

You need to synchronize the changes that occur on the board or in the schematic after the initial transfer of packaged information to the board.

The changes that occur in the board after the initial transfer of packaged information from the schematic are of the following four types:

Types of Changes	Description
Component changes	You can add new components in the design to handle signal integrity and electromagnetic compatibility problems.
Connectivity changes	You can make connectivity changes to facilitate routing after the initial placement of components. Connectivity changes may be caused by pin swaps, section swaps, and reference designator (refdes) swaps.
Reference designator changes	You can change reference designators to debug board problems.

Property changes	You can modify certain components in the board. These modifications will cause property changes.

Generating Manufacturing Output

This section highlights the tasks and Allegro PCB and Package features used to generate output for the manufacturing process:

- Creating NC Drill data
- Generating silkscreens
- Generating pen plots
- Creating artwork
- Drafting and dimensioning

Physical Layout using OrCAD Capture

After creating a schematic and verifying the logic, the next step in the design process is to create the physical layout of the PCB board in PCB Editor; the Cadence tool for designing physical layout of a PCB board.

Capture offers full integration with Cadence® PCB Editor tool suite, allowing you to use all of Capture schematic design capabilities to enter your PCB projects, then export the information to PCB Editor for layout and routing.

While the actual board design tasks are performed in PCB Editor, there are a few tasks that must be performed in Capture to prepare the schematic for the layout.

Following are the design tasks and the best practices must be followed during the schematic capture stage to ensure that process of exporting data to PCB Editor is completed smoothly:

- Preparing the Schematic for Layout
- Property Flow from Capture to PCB Editor
- Generating Initial Board File
- Backannotation from PCB Editor
- Running Design Rules Check - Physical Rules

Preparing the Schematic for Layout

Before you design the physical layout of your schematic in PCB Editor, you should validate your design to ensure the that the object names used in schematic follow the object naming convention required in PCB Editor. This section list some of the recommendations or best practices to be followed in Capture to ensure that schematic is successfully exported to PCB Editor.

Property Flow from Capture to PCB Editor

When you netlist a Capture schematic, not all properties defined in Capture are transferred to PCB Editor. For a property to flow from Capture to PCB Editor it needs to be included in the configuration (.cfg) file.

The configuration file specifies net, part (function), and component instance and component definition properties. This mapping determines what properties may be netlisted from Capture to PCB Editor or back annotated from PCB Editor to Capture. If a Capture property is not included in the configuration file it is not passed to PCB Editor. Similarly, if an PCB Editor property is not listed in the file, it does not get back annotated to Capture.

How properties are netlisted from Capture to PCB Editor

Not all properties in the configuration file show up as properties in PCB Editor. Some of these properties are used in generating portions of the netlist PST*.DAT files.

In PCB Editor, component properties (package properties in Capture) take precedence over function properties (part properties in Capture). So in the netlist, a package property value is used if both a part and package have values for the same property. Capture always uses the occurrence values in the netlist. For a design, you can have multiple configuration files.

Generating Initial Board File

While netlisting a Capture schematic, if required, you can also generate initial board file by selecting the Netrev option in the PCB Editor tab of the Create Netlist dialog box.

In order to generate PCB Editor board file, perform the steps listed in the Generating PCB Editor Netlist section to launch the Create Netlist dialog box and specify netlisting options.

On successful netlisting, blank board file is opened in PCB Editor where you can place the parts and route your ratsnest.

Cross Probing for PCB Editor

After creating the board file, you place and route the board. This includes placing the parts in PCB Editor/Allegro SI/PCB Editor, APD and routing the nets. Sometimes, you may also require to swap pins or sections/functions to make routing easier. You can select the components from the Select elements for placement list in the Placement dialog box and then place them directly on the board. You can also place the components directly from the Capture schematic design. This feature is called cross probing. Between Capture and PCB Editor, there are two cross probing functions: cross highlighting and cross selection.

Cross selecting between Capture and Allegro PCB Editor

If you are placing parts in PCB Editor using Place - Manually command, then select one or more parts in Capture and the corresponding parts will be selected in the Placement dialog box in PCB Editor. This option is only available when PCB Editor is active (running) and Intertool Communication (ITC) is enabled in Capture.

Cross highlighting between Capture and PCB Editor

Cross highlighting applies to three different types of objects: parts, nets and pins. Following are the general rules of cross probing:

- If PCB Editor is in highlight mode, you can select an object in PCB Editor, and the corresponding logical element in Capture is highlighted.
- If PCB Editor is in dehighlight mode, when you dehighlight a physical object, the corresponding logical element is dehighlighted in Capture. Deselecting an element in Capture dehighlights the corresponding element in PCB Editor.
- In Capture, when you select a component, its corresponding physical part is only highlighted in PCB Editor if you are in PCB Editor highlight mode. Otherwise, selection in Capture has no effect in PCB Editor, unless you are using cross selection.

Locking Components during Cross-Probing

When you cross probe between Capture and PCB Editor, you need to keep selecting components in your design to place them on your board.

In many cases, you create elaborate design with a large number of components and intricate connectivity. So when you keep selecting the components and nets on your design, you might inadvertently shift a component. This shift, in some cases, might even cause issues of connectivity.

Backannotation from PCB Editor

The Back Annotate dialog box appears when you choose Back Annotate from the Tools menu after selecting the design folder of a Capture project. The back annotation process generates a Capture compatible swap file, which is based on the differences between the logical view and the physical view.

You use back annotation to synchronize the design file with the changes done in the board file. Changes in the PCB Editor board need to be back annotated to the Capture schematic to ensure the physical board design is consistent with the logical schematic design.

Running Design Rules Check - Physical Rules

Before generating a physical netlist for exporting to PCB Editor, it is a good design practice to verify your design by running Design Rules Check for validating physical rules.

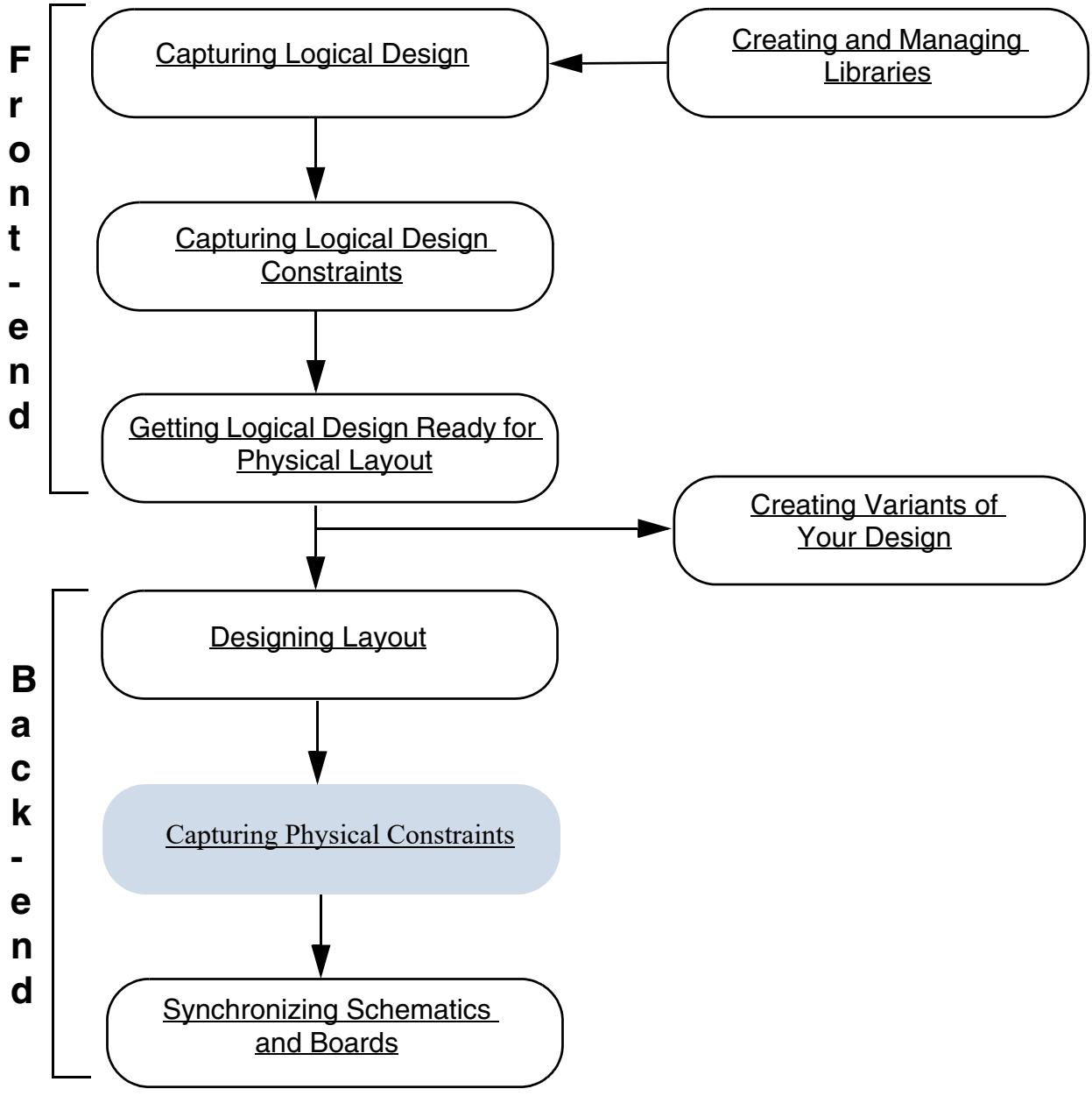
The Design Rules Check tool scans schematic folders to verify that a design conforms to design rules; it generates a report of error and warning messages and places markers on the schematic page to help you locate problems.

7

Capturing Physical Constraints

A constraint is a user-defined rule applied by Design Rule Check (DRC) to a physical element in a design. When you define and apply a constraint, the layout editor adheres to that constraint during automatic and interactive processing and flags violations with DRC markers.

Design rules must be followed while routing the design. You can define spacing and physical design rules within the PCB Editor user interface using the Constraint Manager.



Using Constraint Manager with PCB Editor

Constraint Manager is a spreadsheet-based application with an easy-to-use interface for entering constraints. Another advantage of using Constraint Manager is that it allows you to create generic constraints that you can apply to multiple nets or Xnets at the same time. These reusable constraints are called CSets (Constraint Sets). At a later point in time, if your design requirements change, you can edit the generic rule. The updated rule will be automatically applied to the nets or Xnets that refer to the rule.The existing routes will not modify, but may show DRCs.

Sharing Constraint Details with Schematic Design Tools

If you have made changes to the design in physical layout tool, the logical design needs to be updated with these changes, to ensure that the designs are synchronized. This flow of constraints and other data from Allegro PCB Editor to logic design tools is referred to as back-to-front flow. To update the logical design with the modifications in the physical layout of the design, use *Import Physical* command from the logic design tools.

The files that are read by the logic design tools, while importing changes communicate component, part, function, pin, and electrical constraint information.

For more information on the files used in the back-to-front flow, see the section *Back to Front Constraint Flow* in *Allegro Constraint Manager User Guide*.

Constraint Manager provides the following functionality:

- Creating topology files to use with electrical constraint sets
- Importing electrical constraint sets
- Assigning electrical constraint sets to buses, differential pairs, and XNets

The layout editor designs begin with default constraint sets (named DEFAULT) for spacing and physical constraints. However, electrical constraint sets do not have a default.

You can edit the spacing and physical default constraint sets and specify where and to what elements each constraint applies. You can also assign height information to package symbol files (.dra) and to package keepin and package keepout areas of a board file (.brd) or substrate design file (.mcm).

For more information about constraints in PCB Editor, *Allegro Constraint Manager User Guide* for Constraint Manager connected to Allegro PCB Design.

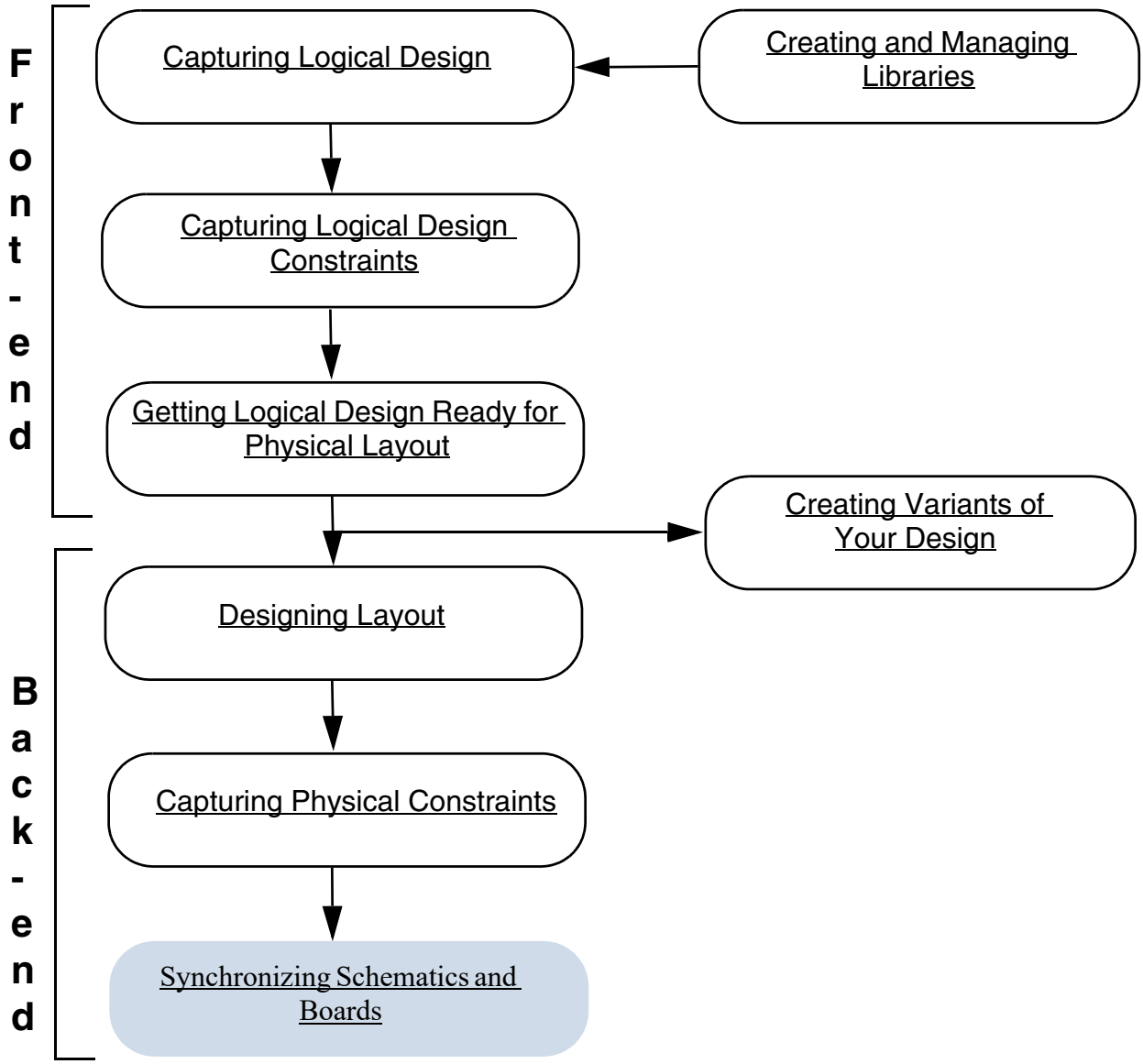
While designing a schematic in Capture, you can specify high-speed constraints as net properties and take them through a complete front-to-back flow. As net properties are passed to the physical netlist generated by Capture, these constraints are also passed to the PCB Editor. In PCB Editor, you can modify these constraints by launching Constraint Manager. Following figure shows the flow of signal properties.

For more information about capturing physical constraints in OrCAD PCB Editor, refer to *OrCAD Capture User Guide*.

8

Synchronizing Schematics and Boards

In the design synchronization phase, you bring the schematic and the board in sync. You resolve both property and connectivity changes between the schematic and the board.



Need for Synchronization

The primary need for synchronization is caused by changes that occur either in the board or in the schematic after the initial transfer of packaged information to the board.

The changes that occur in the board after the initial transfer of packaged information from the schematic are of the following four types:

1. Component changes

   You may add new components in the design to handle signal integrity and electromagnetic compatibility problems. These components may include termination resistors, series or shunt buffers, and bypass capacitors.

2. Connectivity changes

   You may make connectivity changes to facilitate routing after the initial placement of components. Connectivity changes may be caused by pin swaps, section swaps, and reference designator (refdes) swaps.

3. Reference designator changes

   You may change reference designators to debug board problems.

4. Property changes

   You may modify certain components in the board. These modifications will cause property changes.

Design Synchronization Tasks

The entire Design Synchronization process can involve the following tasks:

1. Package and export the schematic design to the layout editing tool by running Packager-XL in the Forward mode.
2. Compare the schematic and layout designs.
3. Package the design for feedback by running Packager-XL in the Feedback mode.
4. Backannotate the physical connectivity changes to the schematic.
5. Backannotate the schematic based on information in the board.
6. Run the Packager utilities to complete any or all of the following steps:

   a. Generating the Bill of Materials

b. Performing electrical rule checks

c. Generating netlist reports

Feedback Mode

In the Feedback mode, Packager-XL receives changes made in PCB Editor and incorporates these changes into the logical design. To run the Feedback mode, you need to run the Import Physical command.

After you have packaged the design and prepared the board, you may add new components, or make property, connectivity, or reference designator changes. These changes cause the schematic and the board to go “out of sync”. You can use the Feedback mode to incorporate the logical changes and assignments made in the physical layout back to the design. When Packager-XL completes packaging the design, a confirmation message is displayed and you want to view the results.