# Allegro Design Entry HDL-FPGA System Planner Flow Guide

**Product Version 23.1**
**September 2023**

**Last Updated: April 2021**

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

**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 longstanding industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

# Contents

Preface . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

Related Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

Typographic and Syntax Conventions . . . . . . . . . . . . . . . . . . . . . . . . . 2

Working with Allegro FPGA System Planner . . . . . . . . . . . . . . . . . . 1

FPGAs - The Problem Scenario . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

Allegro FPGA System Planner - The Solution . . . . . . . . . . . . . . . . . . . 2

- FPGAs in FSP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

FSP – Supported Flow Methodology . . . . . . . . . . . . . . . . . . . . . . . . . 3

Hierarchical Method . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

- ECOs in Hierarchical Method . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
- Is This the Right Solution for Me? . . . . . . . . . . . . . . . . . . . . . . . . . 5

FSP - DE-HDL Front and Back Flow . . . . . . . . . . . . . . . . . . . . . . . . 1

Overview to Front and Back Flow . . . . . . . . . . . . . . . . . . . . . . . . . . 1

Tasks Involved in the Front and Back Flow . . . . . . . . . . . . . . . . . . . . 2

- Tasks to Perform in FSP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
- Tasks to Perform in Design Entry HDL . . . . . . . . . . . . . . . . . . . . . . 4
- Tasks to Perform in PCB Editor . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

Tasks to Perform in Front and Back Flow . . . . . . . . . . . . . . . . . . . . 1

Project Creation and Setup (Front Flow) . . . . . . . . . . . . . . . . . . . . . . 1

- Creating a New Project . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
- Setting-up Library for the FSP Design . . . . . . . . . . . . . . . . . . . . . . . 3

Placing Components and Setting Targets . . . . . . . . . . . . . . . . . . . . . . 4

- Adding Interface to the FSP Design . . . . . . . . . . . . . . . . . . . . . . . . . 4
- Adding Device to the FSP Design . . . . . . . . . . . . . . . . . . . . . . . . . . 7
- Setting Target to Device Instance . . . . . . . . . . . . . . . . . . . . . . . . . . 7

Running Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

Adding and Mapping Power Regulators . . . . . . . . . . . . . . . . . . . . . . . . 9

- Defining Power Mapping . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

Defining Terminations, Decoupling Capacitors and External Ports . . . . . . . . . 13

- Defining Terminations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
- Applying Terminations to Instance Pins . . . . . . . . . . . . . . . . . . . . . . 14
- Defining Decoupling Capacitors . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
- Defining External Ports . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
- Defining External Connections for Virtual Interface . . . . . . . . . . . . . . . 18

Generating Symbols and Schematics . . . . . . . . . . . . . . . . . . . . . . . . 19

Importing FSP Design in DE-HDL 23

- Integrating FSP-Generated Schematic in a PCB Design Project 23

Updating PCB Editor Board 24

- Updating the Board File with Schematic Changes 25
- Importing Placement File 25

Updating FSP Design from Allegro Board (Back Flow) 25

Preserving Schematic in the Front and Back flow 26

FSP – Allegro Integration Flow1

FPGA Design Flow – The Problem 1

FSP - Allegro Integration Flow – The Solution 1

- The Beginning Phase 3
- The Last Phase 5

Tasks to Perform in Integration Flow1

Project Creation and Setup 1

- Launching the FPGA System Planner 1
- Setting up the License 1
- Creating a New Project 2
- Reading Config.ini file and Rules files 2
- The Project Directory 3
- Setting Up Libraries 3
- Defining NetGroups for Nets 4

Placing Component and Setting Target 6

- Adding Interface to the FSP Design 6
- Adding Device to the FSP Design 10
- Setting Target to Device Instance 10

Running Design 11

Adding and Mapping Power Regulators 12

Defining Terminations, Decoupling Capacitors, and External Ports 14

Generating DE-HDL Symbols and Schematics 14

Preparing FSP Design for Integration 14

- Creating a Copy of FSP Design 14
- Packaging the Design 15
- Launching the PCB Editor 16
- Setting up the License 16
- Loading FSP Design in PCB Editor 17

Synchronizing Design between FSP and Allegro 17

Swapping FPGA Pins in PCB Editor 19

Setting up the Design for Optimization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
Displaying Bundles . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
Performing Pin Swaps on Bundles . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
Synchronizing Design between Allegro and FSP . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
Merging Changes with the FSP Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
Regenerating Symbols and Schematics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
Setting up ECO Mode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
Selecting a Lower Product Options . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
Synchronizing Design between FSP and Allegro . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
Swapping FPGA Pins in PCB Editor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
Synchronizing Design between Allegro and FSP . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
Merging Changes with the FSP Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
Regenerating Symbols and Schematics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

Index . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

# Preface

## About Allegro® Design Entry HDL-FPGA System Planner Flow Guide

The *Allegro® Design Entry HDL-FPGA System Planner Flow Guide* provides you easy access to information about the flow of information between Allegro FPGA System Planner, Allegro Design Entry HDL, and Allegro PCB Editor. The topic-related help allows you to get information about the topic you are working on.

## Finding Information in This Flow Guide

This flow guide covers the following topics:

| **See...** | **For Information About...** |
|---|---|
| Chapter1 “Working with Allegro FPGA System Planner” | This chapter gives an introduction about FSP and the supported flow methodology of the FSP solution i.e. Hierarchical method. |
| Chapter2 “FSP - DE-HDL Front and Back Flow” | This chapter describes the procedure for creating the design in FSP, integrating the FSP design in DE-HDL and importing the changes back in FSP. |
| Chapter3 “Tasks to Perform in Front and Back Flow” | This chapter describes step-by-step procedure involved in the Front and Back flow. |
| Chapter4 “FSP - Allegro Integration Flow” | This chapter introduces FSP - Allegro Integration flow and describes the methodology. |
| Chapter5 “Tasks to Perform in Integration Flow” | This chapter describes step-by-step procedure involved in the FSP - Allegro Integration flow. |

## Related Documentation

To see a list of important CCRs defining the known problems in the FPGA System Planner solution, and the solutions or workarounds for the problems, refer to *Allegro FPGA System Planner User Guide*.

You can also refer the following documentation to know more about related tools and methodologies:

- *Allegro FPGA System Planner User Guide (<your_install_folder>/doc/fsp_ug)*
- *FSP – Capture Flow Guide (<your_install_folder>/doc/fsp_capture)*

## Typographic and Syntax Conventions

This list describes the syntax conventions used for this user guide:

`literal` Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names.

*`argument`* Words in italics indicate user-defined arguments for which you must substitute a name or a value.

| Vertical bars (OR-bars) separate possible choices for a single argument. They take precedence over any other character.

`[ ]` Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices. You can choose one argument from the list.

`{ }` Braces are used with OR-bars and enclose a list of choices. You must choose one argument from the list.

# 1

# Working with Allegro FPGA System Planner

## FPGAs - The Problem Scenario

FPGAs are becoming more and more prevalent in today's PCB designs. As their complexity and pin count increases, so do the problems encounteredwhile incorporating the FPGA onto the PCB. With FPGAs that have hundreds to thousands of pins along with increased number of pin assignment rules, the time taken for initial pin assignment also increases.

Besides, FPGA pin assignment is a multi-dimensional and multi-domain task:

| | |
|---|---|
| **Logical Constraints** | The schematic engineer defines the connectivity, which is the logical relationship between signals. |
| **PCB Electrical and Physical Constraints** | The layout designer places components on the PCB, and specifies signal timing and relative propagation delay. |
| **FPGA I/O Pin Usage Constraints** | The FPGA designer specifies the FPGA I/O pin usage constraints including SSO considerations and banking rules. |

Therefore, creating optimal assignments and performing optimization at layout is a significant challenge. If the pins are assigned without considering the exact impact on PCB routing, users are forced to work with the sub-optimal pin assignment.

FPGA design-in is also an iterative process, which includes:

- Assigning FPGA pin locations
- Creating front end symbols
- Capturing schematic
- Creating PCB Footprint

- Routing the PCB

This process increases the number of iterations between the layout engineer and the FPGA designer to finalize optimal pin assignment to improve routing on PCB, as minor layout optimizations or addition of even a single interface to the FPGA impacts the PCB, schematic, and the FPGA designer. Also, changes made late in the design cycle cause significant schedule slips.

In the classic FPGA-based design flow, FPGA tools have no awareness of the PCB. The focus is on the logic inside the FPGA and on how to design the FPGA, and not the board topology. FPGA designers view design from pins-inward and do not always consider board-level signal integrity, routing, system timing, EMI, or manufacturing concerns. The schematic and PCB designers, on the other hand, view design from pins outward. Their focus is on PCB component to FPGA connections. They understand the board topology but not the FPGA I/O rules, pin types, and FPGA constraints and are concerned with signal integrity at the board level.

## Allegro FPGA System Planner - The Solution

The Allegro FPGA System Planner (FSP) solution provides an integrated environment for FPGA-on-Board design, simplifying the entire process through design abstraction. It helps you visualize the design from the PCB perspective, even before the detailed PCB designing starts.

The FSP solution speeds up the FPGA-PCB co-design to integrate large pin count and complex FPGAs in a production-ready PCB Design flow. It synthesizes optimal, up-front, pin locations reducing tedious cleanup of inefficient pin assignments. The end result is reduction in design layers and adherence to critical FPGA and PCB constraints.

In a nutshell, the Cadence FPGA System Planner solution provides:

- Pin assignment feasibility based on the knowledge of the floor-plan. The tool includes I/O synthesis capability that optimizes pin assignment locally and globally across FPGAs.
- Automatic generation of schematic, which can be netlisted to PCB layout. The tool creates library symbols and generates the schematic, which can be opened in front-end schematic tools, such as Design Entry HDL and Design Entry CIS (OrCAD Capture). In the front-end tools, the design can be completed and then exported to PCB Editor where the complete layout can be done.
- Ability to import placement changes from PCB layout. The solution has the capability to import placement and optimization changes from the PCB Editor layout to reassign the FPGA pins in FPGA System Planner and drive back the front to back flow.

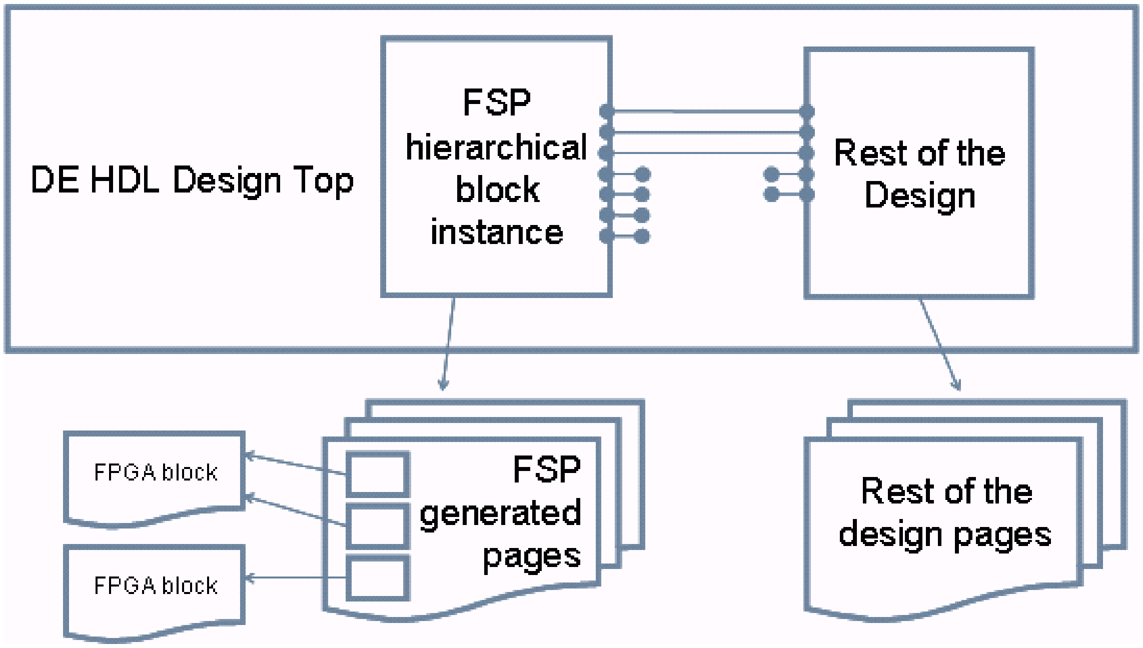
### FPGAs in FSP

In FSP, FPGAs are encapsulated inside a hierarchical block. FPGAs can be divided into multiple split symbols, either existing library parts (standard components) or newly generated parts in FSP, which could be split by bank or custom split symbols. All the split symbols are part of the hierarchical block. All the signals connected to the FPGA pins are interface signals and come out as ports on the hierarchical block symbol. Connection to a specific pin number is hidden. Therefore, you need not bother about pin assignments in FSP. Even when a pin number changes, the FPGA hierarchical block symbol does not change.

DE-HDL schematic can be generated from the FSP design, by using the Hierarchical Methodology.

# FSP – Supported Flow Methodology

# Hierarchical Method

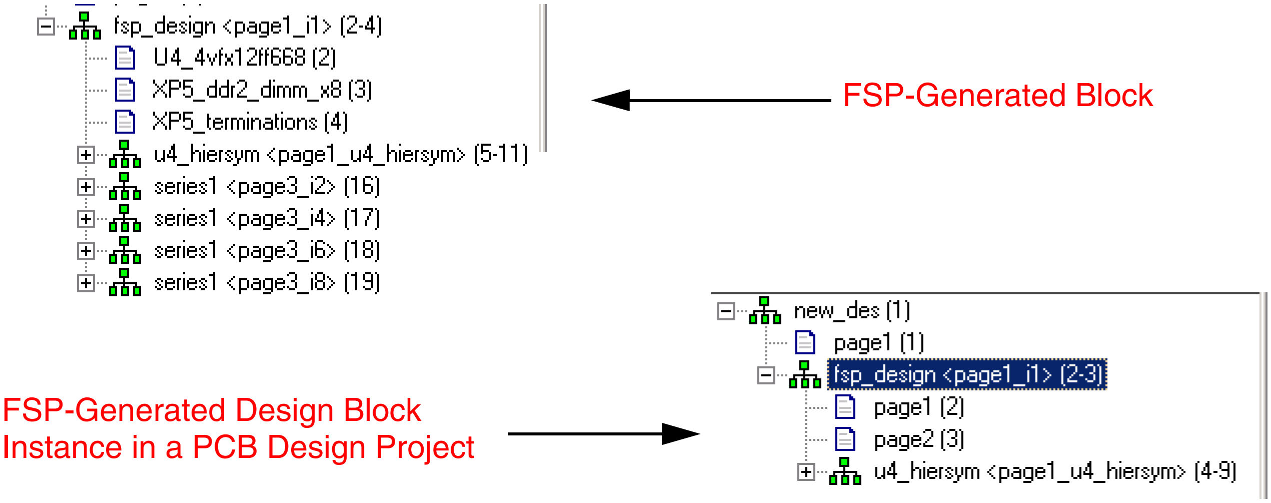


In the hierarchical method, the design intent is captured in FSP and the schematic generated is encapsulated inside a hierarchical schematic block. FSP manages the schematic block entirely and the you need not be concerned about the contents of the block. You integrate the schematic block in a PCB design project by importing the block. You can then manage the interfaces on the symbol connected to other circuitry.

This approach is best suited when the connectivity between the FPGA and the interface components is not frozen and is likely to undergo some iterations.

*Important*

You are not allowed to make any changes to the FSP schematic block in a schematic editor.



```
⊟ fsp_design <page1_i1> (2-4)
    U4_4vfx12ff668 (2)
    XP5_ddr2_dimm_x8 (3)
    XP5_terminations (4)
  ⊞ u4_hiersym <page1_u4_hiersym> (5-11)
  ⊞ series1 <page3_i2> (16)
  ⊞ series1 <page3_i4> (17)
  ⊞ series1 <page3_i6> (18)
  ⊞ series1 <page3_i8> (19)
```

FSP-Generated Block

```
⊟ new_des (1)
    page1 (1)
  ⊟ fsp_design <page1_i1> (2-3)
      page1 (2)
      page2 (3)
    ⊞ u4_hiersym <page1_u4_hiersym> (4-9)
```

FSP-Generated Design Block Instance in a PCB Design Project

**Figure 1-1 FSP Design - Hierarchical Method**

There are two levels of hierarchy, the FPGA block and the rest of the FSP design. FPGA block symbols and other interface components are placed at the same level of hierarchy and are interconnected.

## Integration with the PCB Design Project

If you are using standard components from a corporate library, the `cds.lib` file must contain the reference to the library. If you are generating symbols, you need to copy primitives used in FSP schematic to the PCB design to avoid dynamic changing of schematic symbols. When you import the FSP-generate schematic in a PCB design project, the blocks of FPGA are automatically imported.

## ECOs in Hierarchical Method

In the ECO changes in the FSP design can include connectivity changes between the FPGA component and the interface components, and changes in pin assignment in the FPGA block.

- The complete FPGA schematic block needs to be regenerated. The schematic block owned by FSP is updated without impacting the PCB design project.

- The schematic below the hierarchical block for the FPGA is completely regenerated.
- Wherever new signals are connected to the FPGA, the interface of the hierarchical block must also be updated.
- You need to copy the primitive symbols again and re-import the FSP-generated schematic block.
- Any changes in the external nets is visible on the FSP-generated block symbol.

## Is This the Right Solution for Me?

The hierarchical method is easy to understand and offers almost complete isolation of FPGA changes to schematic. The ECO process is also simple to understand. This method also facilitates quick capture of terminations in FSP.

On the flip side, the hierarchical method creates a potentially large hierarchical block symbol. Also, the schematic generated by FSP cannot be modified in a schematic editor. Additional work is required for defining the terminations in FSP and the implementation in the schematic.

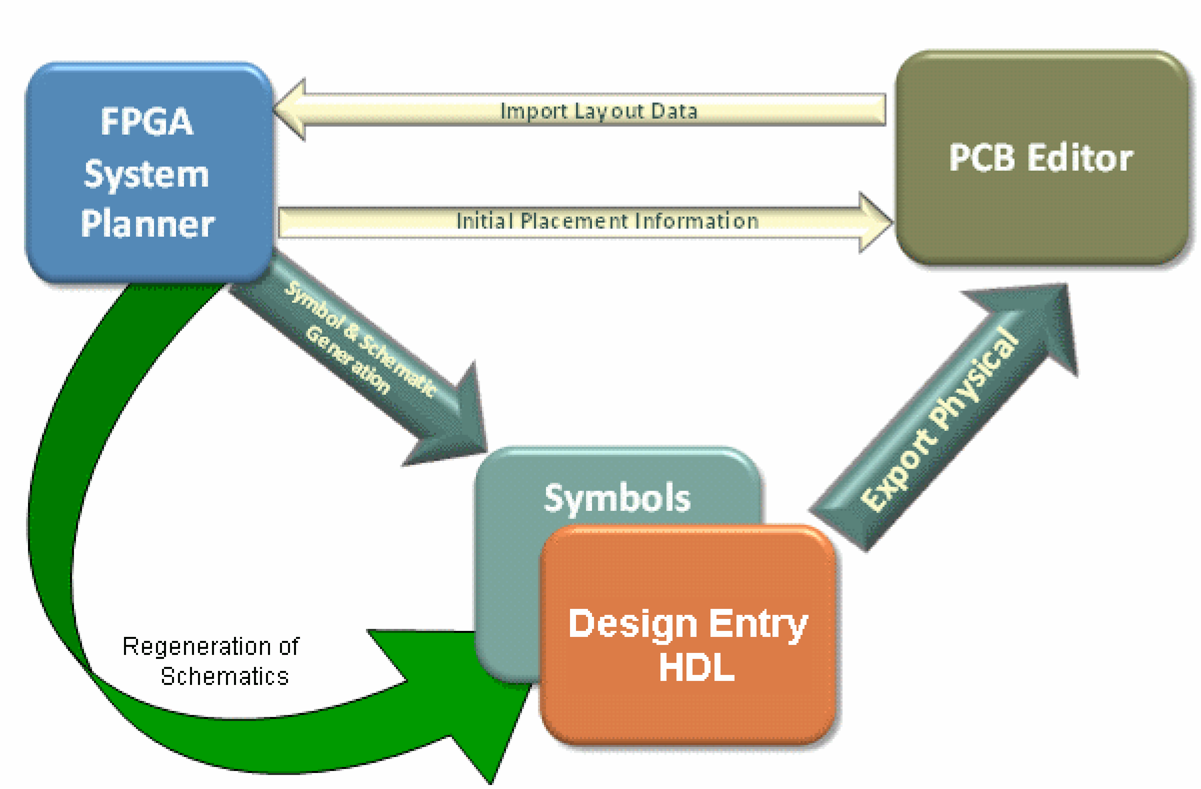
In the hierarchical method, you need to make all the nets as external nets in order to make connections between the interface components and the schematic components in the PCB design project.

# 1

# FSP - DE-HDL Front and Back Flow

## Overview to Front and Back Flow

The flow diagram below depicts the flow of designing an FPGA-based PCB using FPGA System Planner.



The initial FPGA interface is captured in FSP, where power nets are mapped to the voltage requirements of each interface. Symbols and schematics are generated in Design Entry HDL or Design Entry CIS format. The schematic sheets are then integrated into a PCB design project. The text file of initial placement data is also generated in FSP. The placement data is used to recreate floor plan captured in FSP. The FSP design is opened in DE-HDL. Schematic related changes such as adding and removing nets, adding terminations and properties and many more are performed in DE-HDL. Once the changes are done in DE-HDL the design is exported to PCB Editor. You can perform the layout related changes such as instance placement changes in PCB Editor. The changes made in the DE-HDL and PCB Editor are integrated with the existing FSP design by importing the board file in FSP. The symbols/ schematics and layout changes are done in FSP and the schematic is regenerated.

The FSP solution supports the **Hierarchical** methodology. Learn more about the methodology in FSP – Supported Flow Methodology.

# Tasks Involved in the Front and Back Flow

In the flow depicted above, the following tasks are performed in each of the three tools FPGA System Planner, Design Entry HDL, and PCB Editor:

## Tasks to Perform in FSP

See Creating a New Project on page 1 and Updating FSP Design from Allegro Board (Back Flow) on page 25 for details.

FPGA System Planner

1. Create an FSP design
2. Capture FPGA interface - add and connect components on FSP canvas
3. Define power nets and power mappings
4. Define nets interfacing with circuits outside the FSP canvas (external ports)
5. Define and add terminations to instance pins
6. Define decoupling capacitors and add to the power and ground pins
7. Generate symbols and schematics for DE-HDL
8. Generate board file (`.brd`) from FSP
9. Update the FSP design with the layout changes made in the PCB Editor from Allegro board.
10. Re-generate symbols and schematics for DE-HDL

## Tasks to Perform in Design Entry HDL

See Importing FSP Design in DE-HDL on page 23 for details

**Design Entry HDL**

1. Import the generated schematic block/sheets in an existing PCB design project.
2. Package the design using the board file of the PCB design project

## Tasks to Perform in PCB Editor

See Updating PCB Editor Board on page 24 for details.

**PCB Editor**

1. Open the board file.

1

# Tasks to Perform in Front and Back Flow

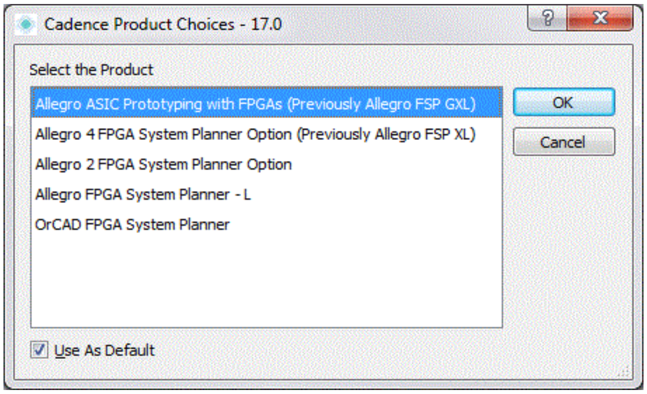
## Project Creation and Setup (Front Flow)

To start working with the FSP solution, the first step is to create an FSP design.

### Creating a New Project

1.Specify *fpgasysplanner* in the *Run* window and click *OK*.

The Cadence Product Choices - <*Release Version*> dialog box is displayed.



Cadence Product Choices - 17.0

Select the Product

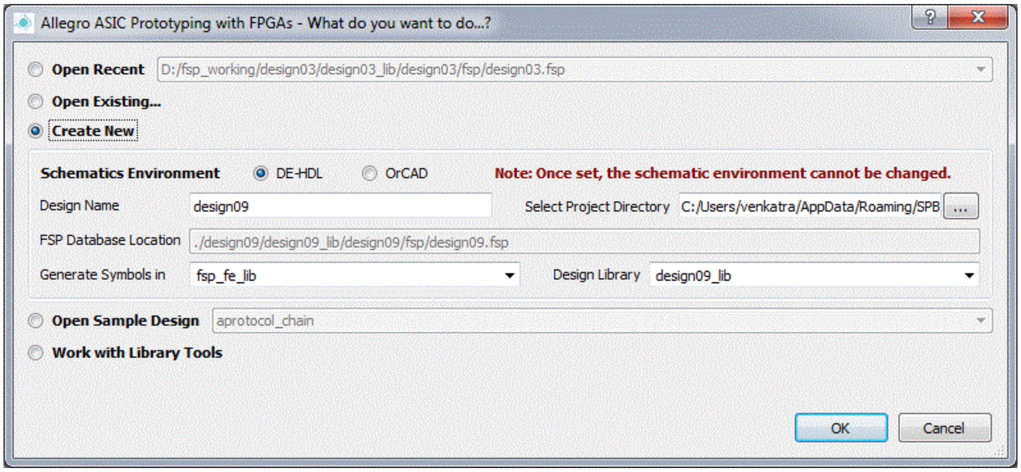
- Allegro ASIC Prototyping with FPGAs (Previously Allegro FSP GXL)
- Allegro 4 FPGA System Planner Option (Previously Allegro FSP XL)
- Allegro 2 FPGA System Planner Option
- Allegro FPGA System Planner - L
- OrCAD FPGA System Planner

☑ Use As Default

2.Choose the appropriate license string in the Cadence Product Choices <*Release Version*> and click *OK*.

**Note:** Select the *Use As Default* option to invoke the selected product license every time you invoke FSP.

The *<Cadence Product String> What do you want to do?* dialog box is displayed.


Allegro ASIC Prototyping with FPGAs - What do you want to do...?
OK
Cancel

○ Open Recent D:/fsp_working/design03/design03_lib/design03/fsp/design03.fsp

○ Open Existing...

● Create New

Schematics Environment ● DE-HDL ○ OrCAD **Note: Once set, the schematic environment cannot be changed.**

Design Name design09 Select Project Directory C:/Users/venkatra/AppData/Roaming/SPB ...

FSP Database Location ./design09/design09_lib/design09/fsp/design09.fsp

Generate Symbols in fsp_fe_lib Design Library design09_lib

○ Open Sample Design aprotocol_chain

○ Work with Library Tools

3.Select the *Create New* option to create a new design.

4. Select DE-HDL as schematic environment.
5. In the *Design Name* field, enter a name for the new project.
6. In the *Select Project Directory* field, type the complete path of the folder in which you want to create the new project.

   Or

   Click *browse(...),* select a folder in *Choose Folder Path* dialog box, and then click *OK.*

7.Click *OK.*

## Reading Config.ini file and Rules files

The `config.ini` file is located at `$CDSROOT/share/cdssetup/fsp` contains tool configuration settings information and rules file path variable. The configuration settings is useful during working with features such as create new project, generate schematic/ symbols and more. The rules file path variable helps FSP to identify and locate the interface library rules files and display it on the Library Explorer. FSP reads these `config.ini` file entries from different locations such as `CDSROOT` and `CDS_SITE` level. The default `config.ini` file is overwritten when the tool is re-installed. It is recommended that you copy and customize the `config.ini` file outside the installation area. Set the `CDS_SITE` environment variable to point to the new location. FSP reads the available customizations from site-level `config.ini` file and remaining settings from installation level `config.ini` file. The order in which FSP looks for the `config.ini` file is:

- FSP_CONFIG_FILE
- CDS_SITE
- CDSROOT

For detailed information about the `config.ini` file, refer to the see the Working with FSP Template Files section and setting up rules file path, see the Set up Rules File Search Path section in *Allegro® FPGA System Planner User Guide*.

## Setting-up Library for the FSP Design

The `config.ini` file contains the lrfpath as variable name and location of the interface rules file as variable values. FSP reads this variable to access the interface library files. You can add libraries in FSP by specifying their logical names and physical locations in the `config.ini` file using any text editor. Do not modify the `config.ini` file located at $CDSROOT. Create a site `config.ini` file and then do the neccessary changes.

You can also use the *Rules File Path Editor* to add, modify or delete the libraries of the project. Both the rules and mapping files are fetched only from the directories specified in the *Rules File Path Editor*. The order in which the libraries are listed in the *Rules File Path Editor* determines their search order.

When you create a project, FSP creates a default `lrf` directory in the project directory and sets it as the working directory. You can set any existing library listed in the *Rules File Path Editor* dialog box as working directory. A working directory is also included as a part of search mechanism.

For detailed information about the setting up the rules file search path, see the Set up Rules File Search Path section in *Allegro® FPGA System Planner User Guide*.

1. Choose *Library – Edit Rules File Path.*

   The *Rules File Path Editor* dialog box is displayed.
2. Click *Add* and select a folder where the rules file exists from *Browse For Folder* dialog box.
3. Click *OK* of *Browse For Folder* dialog box.
4. Select a library row and click *Set Working Dir* to set as the working directory.
5. Click *OK* of *Rules File Path Editor* dialog box.

The new library name appears in Library Explorer.

**Note:** By default, a cpm file is created in project directory. The cpm file contains minimum settings required for the project are stored in cpm file. You can also point your own cpm file of the master board schematics in FSP using Settings dialog box. Click *browse (...)* of Project CPM File field and select .cpm file. While specifying the cpm file, the associated cds.lib file should also be available in the same directory as cpm file.

In *Settings* dialog box, you also have the option to change the symbol generation directory through Generate Symbols In option. Click and select the library name.

# Placing Components and Setting Targets

After adding libraries, you start adding components to your design.

## Adding Interface to the FSP Design

There are two methodologies available for interface component placement. You can use any of the following methodology to start creating your design:

- Capturing FSP Design Using Real Components
- Capturing FSP Design Using Rules File or Virtual Interfaces

## Capturing FSP Design Using Real Components

Component Browser is invoked from FSP to select a symbol from the central library. If the PTF property is set for the symbols then available interface rules file for the selected symbol is displayed. You select a specific PTF row for the symbol. The selected PTF row is used to derive a proper rules file for the chosen part/primitive. After selecting FSP automatically determines and select the appropriate mapping file and rules file combination for the selected symbol. Once the exact interface rules file is found the mapping file is searched in the symbols `library:cell:view` directory. Once the interface rules file and mapping file combination is selected the associate dra file is used to draw the footprint on the canvas.

## Capturing FSP Design Using Rules File or Virtual Interfaces

When you do not have the write permission to access the central library or cannot find the required symbol using Component Browser you can continue to use Library Explorer to place the rules file on canvas. You can even create a virtual interface if you still do not find your desired part in library explorer. Note that the rules file placed using Library explorer or the virtual interface is just a logical rules file which means they are not mapped with any of the front-end symbol or footprint. You can continue your design by capturing the connectivity. Once you complete your design you can convert the interface rules file or virtual interface to real component using mapping file before generating schematics.

This flow describes the Capturing FSP Design Using Real Components steps. For detailed information on the Mapping File and Component Placement methdology and various scenarios (such as mapping file, PTF selection) involved in both the flows refer to *Working with Components* chapter in *Allegro® FPGA System Planner User Guide.*

To place the interface component on canvas you can choose any of the following forms:

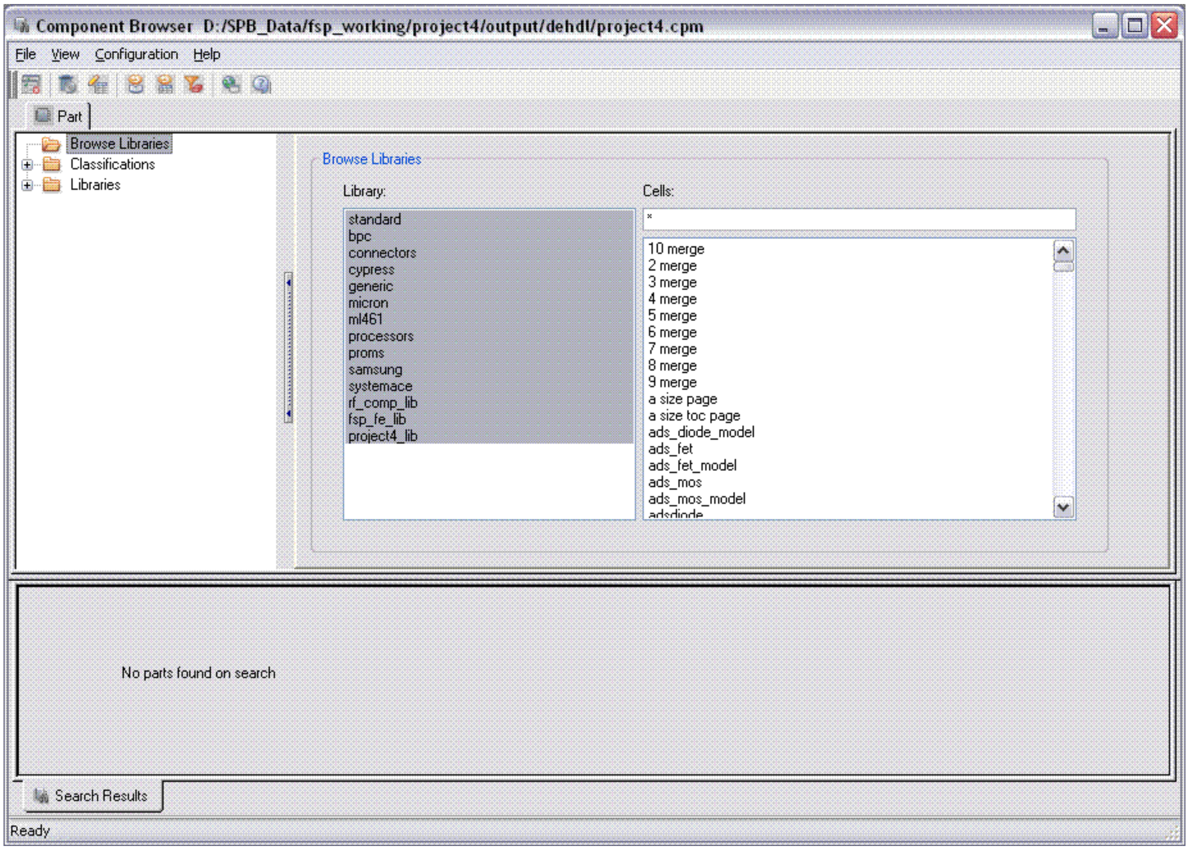
- Using Create/Select Component Placement Rules and Mapping Information Wizard
- Using Add Part dialog box.

The steps below are described using Add Part dialog box. It is also assume that you have both mapping file and interface rules file for the design. Incase if you do not have the mapping file you need to create a new mapping file, for more information see *Working with Components* chapter in *Allegro® FPGA System Planner User Guide.*

To place the component on FSP canvas perform the following steps:

1. To invoke Component Browser click *Add Part* icon in toolbar.

   The Component Browser is displayed.



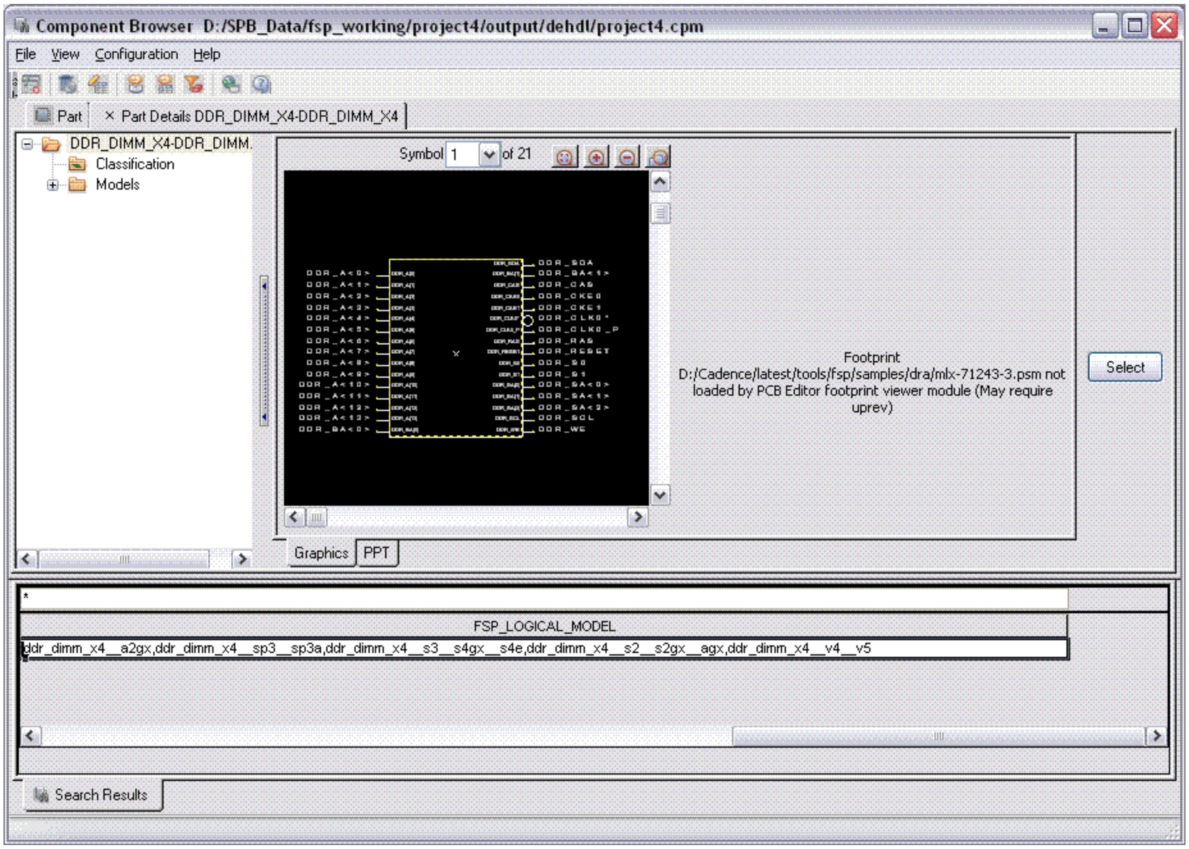
Component Browser D:/SPB_Data/fsp_working/project4/output/dehdl/project4.cpm

No parts found on search

Ready

2. Select a library name in Library pane, whose component you want to bind with the FSP logical model.
3. Select a cell name in Cells pane or enter the name of cell in Cells text box.
4. In Search Results pane, click the row corresponding to the physical part you want to add.

   The symbol and footprint name for the component is displayed in Part Name tab.


Component Browser D:/SPB_Data/fsp_working/project4/output/dehdl/project4.cpm

Part Details DDR_DIMM_X4-DDR_DIMM_X4

Symbol 1 of 21

Footprint
D:/Cadence/latest/tools/fsp/samples/dra/mlx-71243-3.psm not loaded by PCB Editor footprint viewer module (May require uprev)

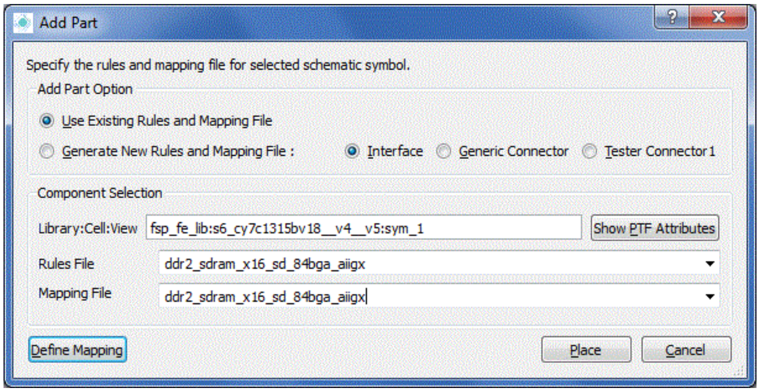
FSP_LOGICAL_MODEL

ddr_dimm_x4__a2gx,ddr_dimm_x4__sp3__sp3a,ddr_dimm_x4__s3__s4gx__s4e,ddr_dimm_x4__s2__s2gx__agx,ddr_dimm_x4__v4__v5

5.Select a symbol view from Symbol drop down list.

6.Click *Select*.

The Add Part dialog box is displayed. Since you have both rules file and mapping file by default the rules file name and mapping file name is displayed in respective fields.


Add Part

Specify the rules and mapping file for selected schematic symbol.

Add Part Option

● Use Existing Rules and Mapping File

○ Generate New Rules and Mapping File : ● Interface ○ Generic Connector ○ Tester Connector1

Component Selection

Library:Cell:View fsp_fe_lib:s6_cy7c1315bv18__v4__v5:sym_1

Rules File ddr2_sdram_x16_sd_84bga_aiigx

Mapping File ddr2_sdram_x16_sd_84bga_aiigx

7.Click *Place* to place the interface component on canvas.

## Adding Device to the FSP Design

Device rules file are treated differently compare to interface rules file. Before you begin the placement, you need to specify the device device rules file name in `cell/library/PTF` row as `FSP_FPGA` property value. The `FSP_FPGA` property helps the Component Browser to identify the component as device component otherwise the component will be treated as interface which leads you to error prone state.

For detailed information on device placement methdology see *Working with Component* chapter in *Allegro® FPGA System Planner User Guide.*

To place the device rules file on canvas using Component Browser perform the following steps:

1. Click *Add Part* icon in toolbar.

   The Component Browser dialog box is displayed.

2. Browse for the library whose component you want to add in Library Pane.
3. Select a cell name in Cells pane or enter the name of cell in Cells text box.
4. In Search Results pane, click the row corresponding to the physical part you want to add.

   The symbol and footprint name for the component is displayed in Part Name tab.

5. Select a symbol view from *Symbol* drop down list.
6. Click *Select*.

   After clicking Select, a graphical view of device is displayed. Left click to drop the device rules file on canvas and right click to disable the graphical view.

**Note:** After clicking Select in Component Browser if you unexpectedly see the Add Part dialog box, then you must specify the `FSP_FPGA` attribute and re-invoke the Component Browser to place the device rules file on canvas.

## Setting Target to Device Instance

After placing the interface and device instance on canvas, target need to be specified. Targeting device decides which interface will connect to which device.

To target the device, perform the following steps:

1. Right-click on the interface instance and do any of the following:

   ❑ Choose *Target To Device – <Instance Name>*.

   After you choose this option, all the groups of the interface are targeted to device at one go.

   ❑ Click *Instance Properties.*

2. Click the *Group Settings* button in the *Properties window.*

The *Group Settings for Interface Instance<inst_name>* is displayed.

3. To target all the groups, click the *Connect to Device* column name, click on any one of the drop-down button, and select the device instance name from the drop-down list.
4. To target a single groups, click the drop-down button in the first group under *Connect to Device* column and select the instance name.

   **Note:** You can perform the same step for other groups.
5. To target the interface groups to a specific bank of the targetted device, do the following:

   **a.** To target all the interface groups to one bank, click on the *Use Bank* column name and click on any one of the drop-down button.

   A pop-up menu is displayed with package view of the targeted device and the list of banks available in the device.

   **b.** Select a bank number from the list.

   **c.** Click *OK.*

   **Note:** To select a bank from the package view, select a pin of a bank to which you want to target the interface group. After you select the pin, the remaining pins of the bank are automatically selected and highlighted.
6. Click *OK* in the Group Settings for Interface Instance <inst_name> dialog box.

The interface instance group (s) is targeted to the device instance.

## Running Design

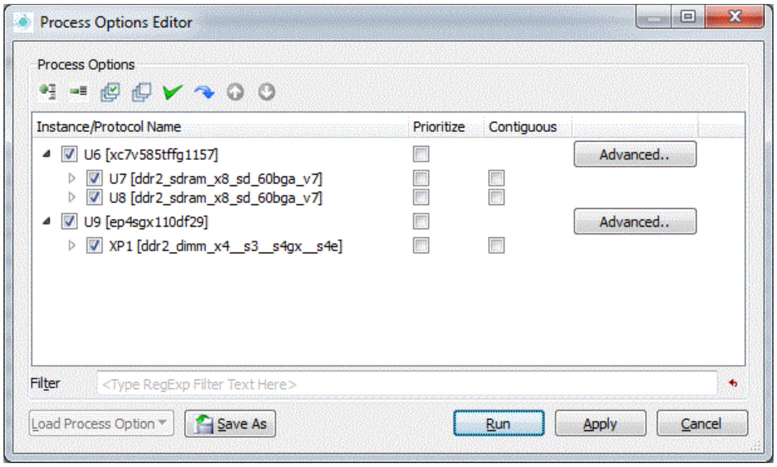
Once the interface components are targeted to respective devices, you are ready to make the connections between them. Before running a design you set the preferences such as setting an order to an interfaces and its groups such as selecting few device and interface instances for pin assignments and defining the advance options for better results.

For detailed information on how to run the design per instance and various process options see *Running a Design* chapter in *Allegro® FPGA System Planner User Guide*.

To make connections between the components perform the following steps:

1. Choose *Tools – Run Design.*

   The Process Options Editor dialog box is displayed.



| Instance/Protocol Name | Prioritize | Contiguous | |
|---|---|---|---|
| ☑ U6 [xc7v585tffg1157] | □ | | Advanced.. |
| ☑ U7 [ddr2_sdram_x8_sd_60bga_v7] | □ | □ | |
| ☑ U8 [ddr2_sdram_x8_sd_60bga_v7] | □ | □ | |
| ☑ U9 [ep4sgx110df29] | □ | | Advanced.. |
| ☑ XP1 [ddr2_dimm_x4__s3__s4gx__s4e] | □ | □ | |

2.Click *Advance* to specify different proximity options as per required.

3.Click *Run* to run the design.

The pin assignments and connections between the instances are automatically established.

# Adding and Mapping Power Regulators

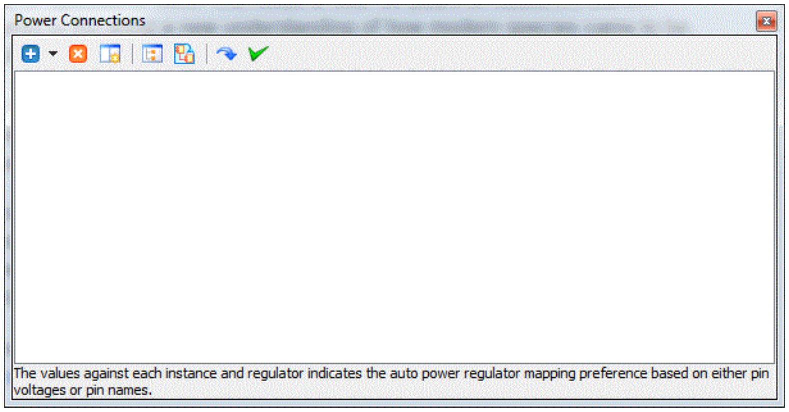
Once the connections are made, you can define new power regulators. Power regulators can be edited at anytime during the design. You define new regulators and assign power voltage to each regulators. Regulators can be defined manually of your own choice or can be added automatically. After defining the power regulators you map them with the associated FPGA power pins. For detailed information, see the *Adding and Mapping Power Regulators* chapter in *Allegro® FPGA System Planner User Guide.*

You can add power regulators and corresponding voltage values of your choice.

To add a regulator, perform the following steps:

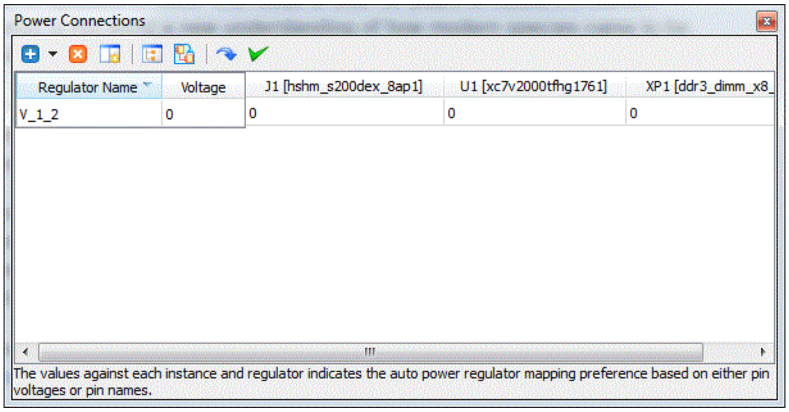
1. Choose *Window – Power Connections.*

   The *Power Connections* pane is displayed.



The values against each instance and regulator indicates the auto power regulator mapping preference based on either pin voltages or pin names.

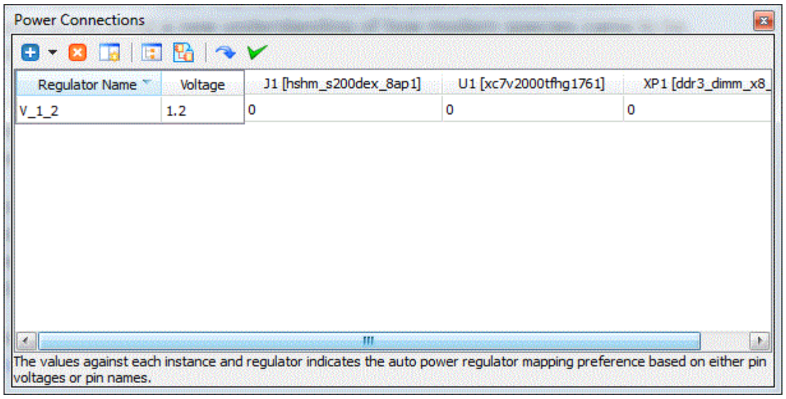
2. Click the **+** icon to add a new row.
3. Type a name under the *Regulator Name* column.



| Regulator Name | Voltage | J1 [hshm_s200dex_8ap1] | U1 [xc7v2000tfhg1761] | XP1 [ddr3_dimm_x8_ |
|---|---|---|---|---|
| V_1_2 | 0 | 0 | 0 | 0 |

The values against each instance and regulator indicates the auto power regulator mapping preference based on either pin voltages or pin names.

4. Enter a numerical value under the *Voltage* column.



| Regulator Name | Voltage | J1 [hshm_s200dex_8ap1] | U1 [xc7v2000tfhg1761] | XP1 [ddr3_dimm_x8_ |
|---|---|---|---|---|
| V_1_2 | 1.2 | 0 | 0 | 0 |

The values against each instance and regulator indicates the auto power regulator mapping preference based on either pin voltages or pin names.

To define regulators automatically, perform the following steps:

1. Click the *Auto Add Regulator* option.

   A confirmation window is displayed about adding new regulators in the design.

2. Click *Yes* to proceed further.

   The new regulator names and values are listed in the power connections window.

## Defining Power Mapping

After adding the power regulators and corresponding voltage values, you define power mapping. You use this feature to define a mapping between a power regulator and a power/ground pin name or its voltage value. For example, if you want to connect a power regulator to a power/ground pin, you first need to define the mapping between them. After defining the mapping, these mapping inputs are considered by FSP when you Automap Power Regulators.

You can define power mapping using the following methods:

- Mapping Voltage Value
- Mapping Power Pins

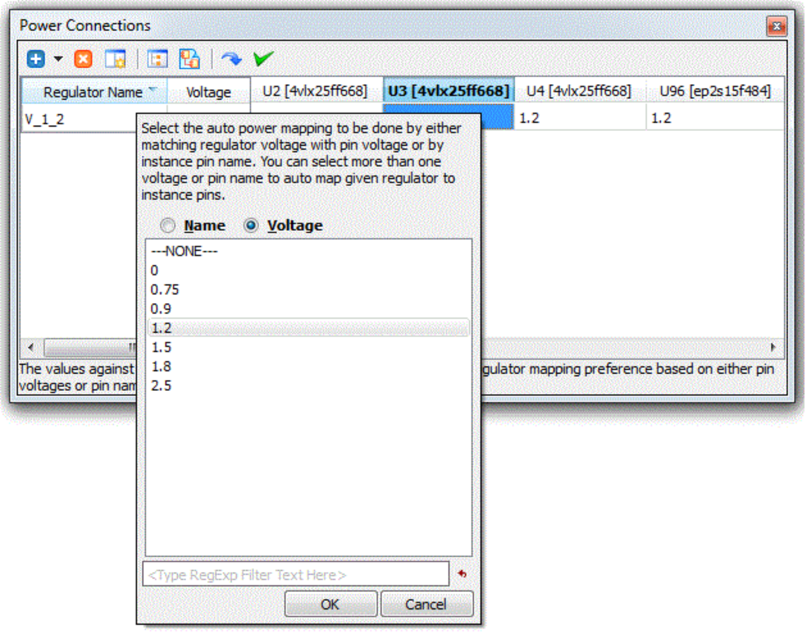
For detailed information about the different types of power mapping, see the *Working with Power Regulators* chapter of the *Allegro FPGA System Planner User Guide.*

This section provides the steps to define power mapping using power pins.

To map a power pin name to a power regulator, do the following:

1. Under the instance name column, click on the cell for which you want to define power mapping.

   A dialog box appears.



Power Connections

| Regulator Name | Voltage | U2 [4vlx25ff668] | U3 [4vlx25ff668] | U4 [4vlx25ff668] | U96 [ep2s15f484] |
|---|---|---|---|---|---|
| V_1_2 | | | | 1.2 | 1.2 |

The values against … voltages or pin nam… …gulator mapping preference based on either pin

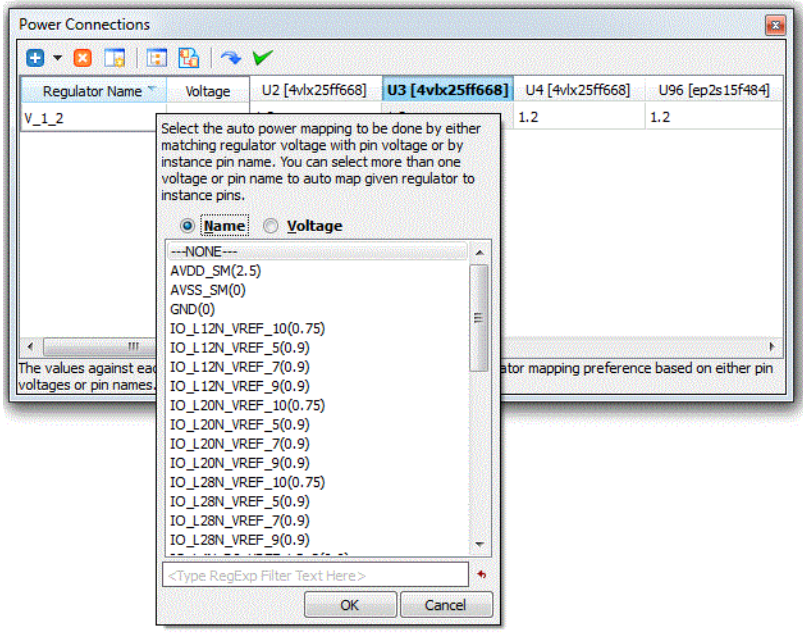
Select the auto power mapping to be done by either matching regulator voltage with pin voltage or by instance pin name. You can select more than one voltage or pin name to auto map given regulator to instance pins.

○ Name ● Voltage

---NONE---
0
0.75
0.9
1.2
1.5
1.8
2.5

2. Select *Name.*

You will see a list of power pin names of the current instance.



Power Connections

| Regulator Name | Voltage | U2 [4vlx25ff668] | U3 [4vlx25ff668] | U4 [4vlx25ff668] | U96 [ep2s15f484] |
|---|---|---|---|---|---|
| V_1_2 | | | | 1.2 | 1.2 |

The values against eac… voltages or pin names. …ator mapping preference based on either pin

Select the auto power mapping to be done by either matching regulator voltage with pin voltage or by instance pin name. You can select more than one voltage or pin name to auto map given regulator to instance pins.

● Name ○ Voltage

---NONE---
AVDD_SM(2.5)
AVSS_SM(0)
GND(0)
IO_L12N_VREF_10(0.75)
IO_L12N_VREF_5(0.9)
IO_L12N_VREF_7(0.9)
IO_L12N_VREF_9(0.9)
IO_L20N_VREF_10(0.75)
IO_L20N_VREF_5(0.9)
IO_L20N_VREF_7(0.9)
IO_L20N_VREF_9(0.9)
IO_L28N_VREF_10(0.75)
IO_L28N_VREF_5(0.9)
IO_L28N_VREF_7(0.9)
IO_L28N_VREF_9(0.9)

3. Select a name.

**Note:** You can also select multiple power pins names.

4. Click *OK.*

# Defining Terminations, Decoupling Capacitors and External Ports

## Defining Terminations

Once the power mapping is done, you add termination in your FSP design. Two most common types of termination are supported in FSP are:

1.Split Termination

2.Series Termination

Once you select and specify the termination types, you need to place the primitive and non primitive components in the terminations block and map the ports with selected terminations.

**Note:** For detailed information on the different types of terminations FSP supports and on how to define and map terminations with primitive and non-primitive components, refer to the *Define Termination* section of the *Allegro® FPGA System Planner User Guide*.

To add terminations in your design first you need to specify the termination type. The below described is the steps required to define a termination for single ended signals.

To define a series termination for single ended signals complete the following steps:

1.Choose *Tools – Define Terminations*.

The Define Termination dialog box is displayed.

2.Click *Add* to add a blank row to define the termination in Define Termination dialog box.

3.In Name column, enter a termination name.

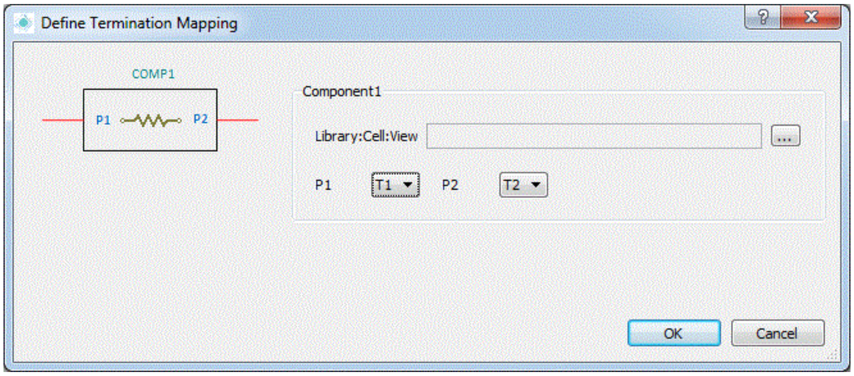
4.In Termination Type column, click the cell. The Termination Type drop down pane displays the available termination types. Select the appropriate type and click *OK.*

Important

Do not define anything in Diff.Termination column since you are defining the termination for single ended signal.

5. Click *browse (...)* in Termination Mapping column.

   The Define Termination Mapping dialog box is displayed.



   Since you have selected the Series Termination it may include one resistor or one capacitor. You need to map the termination ports with components pins.

6. Click and select the library name from *Library Name* drop down list.
7. Click and select the part name (with two ports) from *Part Name* drop list.
8. In Symbol Port Mapping pane do the following:

   **a.** Click and select the component port name from P1 drop down list.

   **b.** Click and select the other side of component port name from P2 drop down list.

9. Click *OK* to apply the termination mapping.
10. Click *OK* of Define Termination dialog box.

## Applying Terminations to Instance Pins

After defining the terminations, you can start applying terminations to the appropriate pins. Terminations can be applied to both the device and interface instance pins. You apply the terminations to the instance pins using the Design Connectivity window. Design Connectivity window gives you a spreadsheet view that helps you to apply the termination on each pin of the instances. Several quick and right mouse button options are also available in the editor to quickly apply the termination on all the pins in the design. The Properties window can also be used as an alternative for applying terminations to the pins. Invoke and arrange the Design Connectivity window and Properties window side-by-side. Click on a pin in the Design Connectivity window, the properties of the selected pin is displayed in the Properties window. In the Properties window, use the *Pin Termination* and *FPGA Ext Termination* cells to apply the termination.

**Note:** The described below are the steps to apply series termination to both single ended and differential pair pins. You can follow the same steps for other terminations also.

## Applying Series Termination to Single Ended Pins (Interface/Device)

To apply the series termination to single ended pins, perform the following steps:

1. Invoke the *Design Connectivity window.*

   The Design Connectivity window is displayed.

   **Note:** In the Design Connectivity window, the Pin Termination and FPGA Ext. Termination columns are used to apply terminations for instance pins.

2. In the Pin Termination column, double-click on a cell and select a termination name from the drop-down list.

**Note:** When you double click on a cell, a list of termination names are displayed. These termination names are defined for the single ended pins in the Define Termination dialog box are displayed.

3. In the FPGA Ext Termination column, the termination name is automatically applied as defined in the Define Termination dialog box.
4. Apply the termination to other pins if required, by performing the steps 1 to 4.

*Important*

In the PCB design project in which you will integrate the FSP generated schematic, you need to manually add the terminations.

*Important*

You can only use discrete components in the termination circuit. Usage of active components, such as buffer ICs is not allowed. You must also ensure that in Design Entry HDL, SI (signal integrity) models are assigned to the discrete components, which are connected in series mode in the termination circuit. The design with SI models will guide FSP to identify connectivity differences between FSP design and the layout while importing the board file back in FSP.

## Defining Decoupling Capacitors

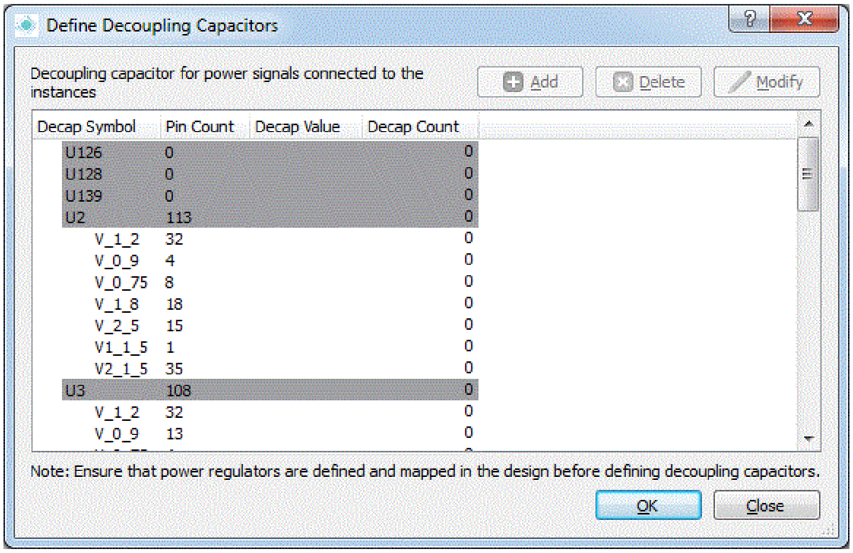
Decoupling capacitor is a special kind of pull up/down termination provided by FSP which is placed between power regulator and ground connections. Placing a capacitor between power regulator and ground connection maintains the power supply voltage at the device. See Defining Decoupling Capacitors section in Allegro FPGA System Planner User Guide.

FSP outputs the decoupling capacitor topology as bypass termination in schematic and places in a different schematic page.

To add a decoupling capacitor in your design, perform the following steps:

1. Choose *Tools – Decoupling Capacitors*.

   The Define Decoupling Capacitors dialog box displays a list of instance names with associated power regulators and voltage values in tree view structure.



Decoupling capacitor for power signals connected to the instances

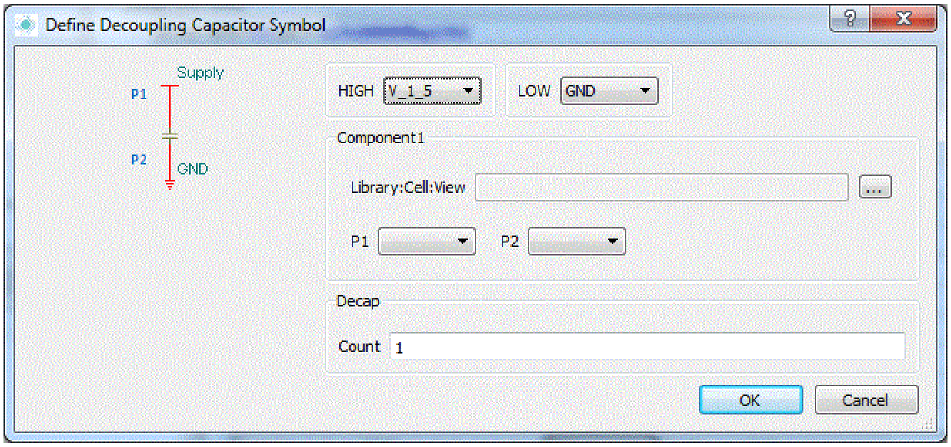
| Decap Symbol | Pin Count | Decap Value | Decap Count |
|---|---|---|---|
| U126 | 0 | | 0 |
| U128 | 0 | | 0 |
| U139 | 0 | | 0 |
| U2 | 113 | | 0 |
| V_1_2 | 32 | | 0 |
| V_0_9 | 4 | | 0 |
| V_0_75 | 8 | | 0 |
| V_1_8 | 18 | | 0 |
| V_2_5 | 15 | | 0 |
| V1_1_5 | 1 | | 0 |
| V2_1_5 | 35 | | 0 |
| U3 | 108 | | 0 |
| V_1_2 | 32 | | 0 |
| V_0_9 | 13 | | 0 |

Note: Ensure that power regulators are defined and mapped in the design before defining decoupling capacitors.

**Note:** After power mapping is done the regulator names and values are displayed in this dialog box.

2. Click a regulator name from list to which you want to connect capacitor.
3. Click *Add*.

   The Define Decoupling Capacitor Symbol dialog box is displayed.



HIGH V_1_5

LOW GND

Component1

Library:Cell:View

P1

P2

Decap

Count 1

4. Click *browse (...)* to invoke Component Browser.

   **a.** Select a library name in Library pane.

**b.** Select a cell name in Cells pane or enter the name of cell in Cells text box.

**c.** In Search Results pane, click the row corresponding to the physical part you want to add.

**d.** The symbol and footprint name for the component is displayed in Part Name tab.

**e.** Select a symbol view from Symbol drop down list.

**f.** Click Select.

After selecting the symbol from Component Browser the `lib:cell:view` structure name is displayed in `Library:Cell:View` text box.

5. Click and select the component port name from *P1* drop down list.
6. Click and select the other side of component port name from *P2* drop down list.
7. In High field, click and select a port name from drop down list that you want to connect to regulator.
8. In Low field, click and select the other port name to that you want to connect to ground.
9. Enter the number of capacitors you want to connect to regulators in Count field.

   For example, if you have fifteen `V_1_5` power regulators in component you choose fifteen capacitors.
10. Click *OK* to save settings.

    Once you click OK, the selected capacitor value is displayed in Decap Value column and number of capacitors is displayed in Decap Count.
11. You can add capacitors for other regulators also, if required by performing steps 1 to 16.
12. Click *OK* to add capacitors in your design.

Clicking OK indicates that capacitors is added in your design. When you generate the schematics FSP output these capacitors as Bypass termination in DE-HDL and will be displayed in separate schematic page.

## Defining External Ports

Some nets created on the FSP canvas need to communicate with the components which are not captured on the FSP canvas. These nets are external to the FSP canvas and known as External Ports. In case the top-level design is encapsulated in a hierarchical block, external ports become interfaces of the hierarchical block and function as pins of the block symbol. Connections to these pins establish connectivity with the design captured in the schematic. FSP provides you a convenient way to define ports. You can quickly define the export ports for routed and unrouted nets in the Design Connectivity window. The Design Connectivity window also provides various quick options to define external port to multiple pins, thereby saving time and effort. When you generate the schematics, FSP automatically creates a high-level port for these nets so that you can easily connect the FSP-generated design with the user-created design while keeping the optimized portion separate.

For detailed information, see the *Defining External Connections* section in *Allegro® FPGA System Planner User Guide*.

The different types of port connections, are available as a drop-down list options in the cell(s) of the *Connection Type* column in Design Connectivity window. However, these options are filtered and displayed based on the pin connections. For example, if a pin is connected to a net(or the pin has *Allocated* value in the *Status* column) then the *Extend as External Port* option is available in the *Connection Type* column.

To define a port connection type, perform the following steps:

1. Invoke the *Design Connectivity window.*
2. For a routed net:

    **a.** In the *Connection Type* column, click on the cell and select the *Extend as External Port* option from the drop-down list.

    **b.** In the *Net Name* column, type a name in the cell next to the cell (*Connection Type* column) and press *Enter.*
3. For an unrouted net:

    **a.** In the *Connection Type* column, click in the cell and select an option from the drop-down list that displays the available connection types, then press *Enter.*

    **b.** In the *Net Name* column, type a name in the cell next to the cell (*Connection Type* column) and press *Enter.*

    **Note:** To define the same connection type to multiple pins or signals, press and hold Ctrl and select the cells, and click on the last selected cell and select an option from the drop-down list and press *Enter.*

    **Note:** To define the same connection type for vectored pins or signals, define a connection type for any one of the bit, and right-click on the cell and choose the *Apply to Bus <Net Name>* option.

    **Note:** You can also define the connection type in the *Properties window.*

## Defining External Connections for Virtual Interface

FSP provides you support for Virtual Interface, which are created on the FSP canvas as place holders for real component interfaces. A virtual interface becomes an interface to the FSP hierarchical block. Therefore, the ports on the FSP hierarchical block symbol can be used to connect to a real component in the schematic.

After running the design, the virtual interface nets are automatically set as *Extend as External Port* in the *Connection Type* column in Design Connectivity window. These nets are displayed in the disabled mode in Design Connectivity window.

# Generating Symbols and Schematics

After creating the basic FSP design, you need to generate the symbol data for the components used in the design, also the schematic, and the placement data for the PCB board. Since the components are already linked with the associated symbol and footprint (using Add Part dialog box) you do not need to generate the symbol, the symbol data will be reuse by FSP.

In case if you have not placed the component using Add Part dialog or Wizard you can follow the below section descriptions.

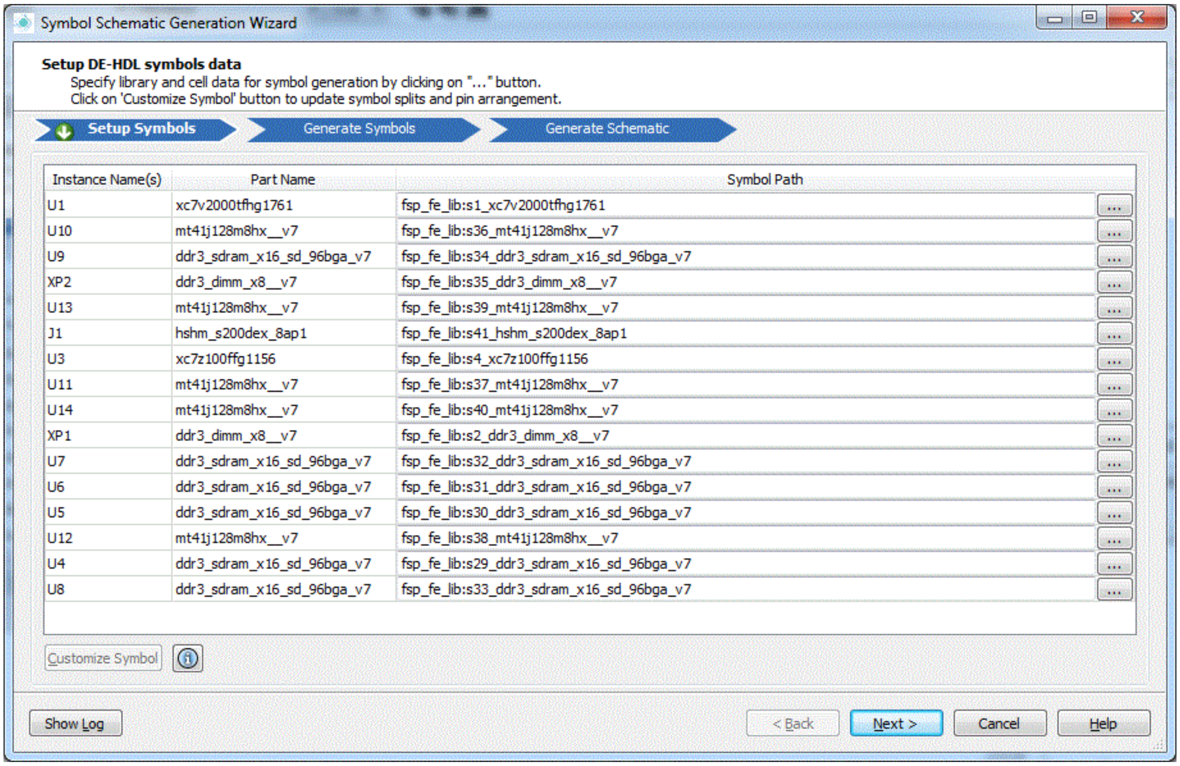
Symbols, schematics and placement data can either be generated individually by invoking the respective forms using separate menu commands or using *Symbol Schematic Generation Wizard*. The *Symbol Schematic Generation Wizard* guides you through a series of steps to define the required attributes to generate the symbols and schematics for DE-HDL, and to generate the initial placement data for PCB Editor. The wizard takes you through the following steps:

- Setting up Symbol Data
- Generating DE-HDL Symbols
- Generating DE-HDL Schematics
- Generating PCB Placement Data

To generate the symbols and schematics perform the following steps:

1. Invoke the Symbol Schematic Generation Wizard by choosing *Generate – Schematic Generation Wizard.*

   The Symbols Schematic Generation Wizard is displayed with Setup DE-HDL Symbols to Use/Generate page.



**Setup DE-HDL symbols data**

Specify library and cell data for symbol generation by clicking on "..." button.
Click on 'Customize Symbol' button to update symbol splits and pin arrangement.

| Instance Name(s) | Part Name | Symbol Path | |
|---|---|---|---|
| U1 | xc7v2000tfhg1761 | fsp_fe_lib:s1_xc7v2000tfhg1761 | ... |
| U10 | mt41j128m8hx__v7 | fsp_fe_lib:s36_mt41j128m8hx__v7 | ... |
| U9 | ddr3_sdram_x16_sd_96bga_v7 | fsp_fe_lib:s34_ddr3_sdram_x16_sd_96bga_v7 | ... |
| XP2 | ddr3_dimm_x8__v7 | fsp_fe_lib:s35_ddr3_dimm_x8__v7 | ... |
| U13 | mt41j128m8hx__v7 | fsp_fe_lib:s39_mt41j128m8hx__v7 | ... |
| J1 | hshm_s200dex_8ap1 | fsp_fe_lib:s41_hshm_s200dex_8ap1 | ... |
| U3 | xc7z100ffg1156 | fsp_fe_lib:s4_xc7z100ffg1156 | ... |
| U11 | mt41j128m8hx__v7 | fsp_fe_lib:s37_mt41j128m8hx__v7 | ... |
| U14 | mt41j128m8hx__v7 | fsp_fe_lib:s40_mt41j128m8hx__v7 | ... |
| XP1 | ddr3_dimm_x8__v7 | fsp_fe_lib:s2_ddr3_dimm_x8__v7 | ... |
| U7 | ddr3_sdram_x16_sd_96bga_v7 | fsp_fe_lib:s32_ddr3_sdram_x16_sd_96bga_v7 | ... |
| U6 | ddr3_sdram_x16_sd_96bga_v7 | fsp_fe_lib:s31_ddr3_sdram_x16_sd_96bga_v7 | ... |
| U5 | ddr3_sdram_x16_sd_96bga_v7 | fsp_fe_lib:s30_ddr3_sdram_x16_sd_96bga_v7 | ... |
| U12 | mt41j128m8hx__v7 | fsp_fe_lib:s38_mt41j128m8hx__v7 | ... |
| U4 | ddr3_sdram_x16_sd_96bga_v7 | fsp_fe_lib:s29_ddr3_sdram_x16_sd_96bga_v7 | ... |
| U8 | ddr3_sdram_x16_sd_96bga_v7 | fsp_fe_lib:s33_ddr3_sdram_x16_sd_96bga_v7 | ... |

*Generate Symbol* check boxes are disabled for the instances which are linked to the symbols and footprint, and enabled for the instances which are not linked and symbols need to be generated for them.

2. Click *Next* to advance to the Generate DE-HDL Symbols page.

The Generate DE-HDL Symbols page is displayed.

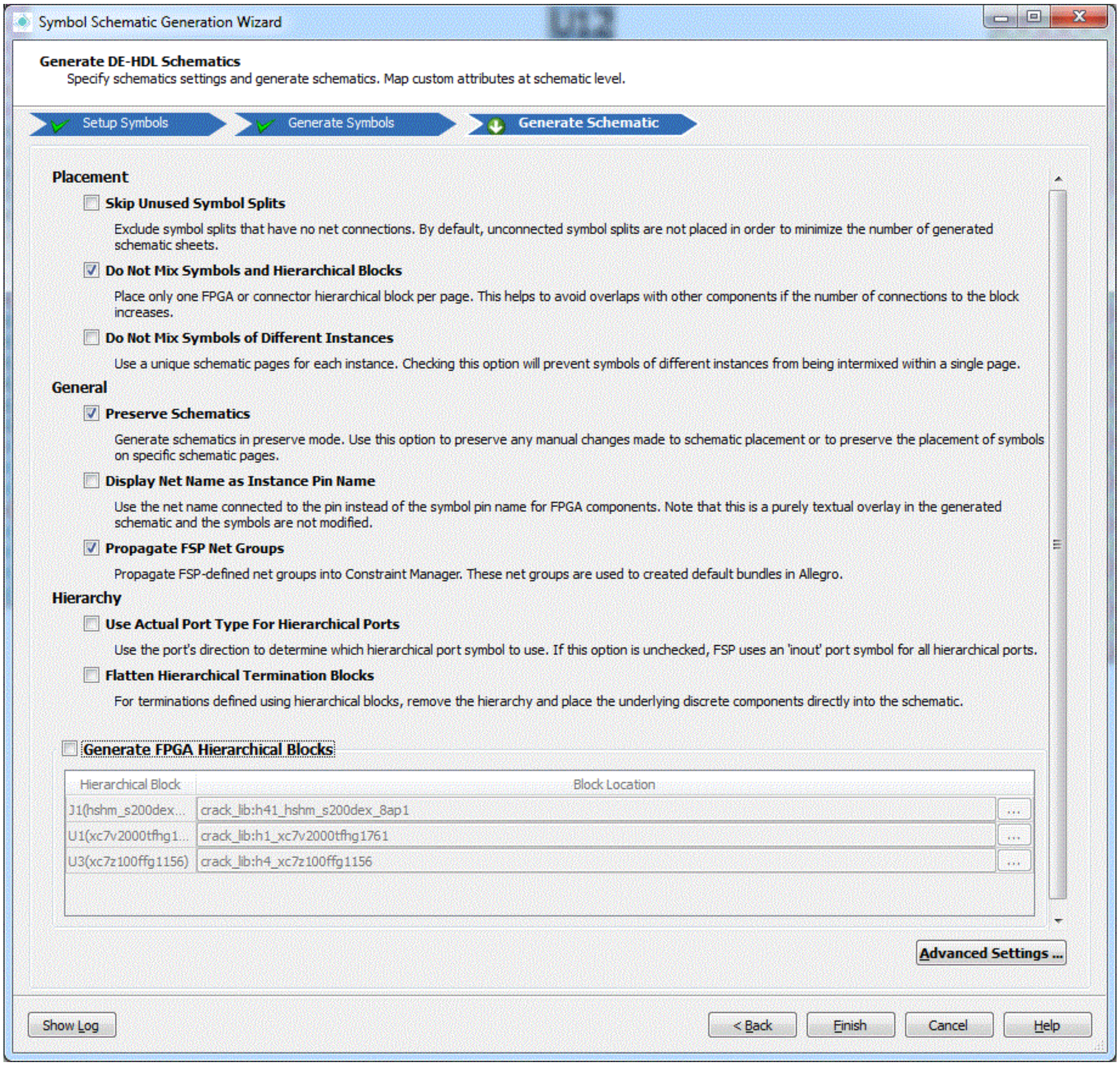
This page lets you to generate the DE-HDL symbols for the parts (which are not linked) used in the design. The global power properties for the part can also be generated through this page. After generating the symbols, these symbols will be used to capture schematics. The Generate/Reuse Allegro Symbols page displays only parts (without existing symbols or not linked). You can regenerate symbols for parts, which already have symbols.The existing symbols can also be reused by FSP for the parts in your design.

*Caution*

***The Global Power Prop. check box is used to specify whether power pins are included as pins on the symbol or are added to the*** `chips.prt` ***file as implicit connections. However, it is recommended that you do NOT enable the Global Power Prop check box. You should not generate power pins as implicit power pins in the*** `chips.prt` ***file, as this could result in loss of global signal assignments data while importing the schematic into another design.***

3.Click *Next* to generate the symbols and advance to the Generate DE-HDL Schematics page.

The Generate DE-HDL Schematics page is displayed.



**Generate DE-HDL Schematics**

Specify schematics settings and generate schematics. Map custom attributes at schematic level.

**Placement**

☐ **Skip Unused Symbol Splits**

Exclude symbol splits that have no net connections. By default, unconnected symbol splits are not placed in order to minimize the number of generated schematic sheets.

☑ **Do Not Mix Symbols and Hierarchical Blocks**

Place only one FPGA or connector hierarchical block per page. This helps to avoid overlaps with other components if the number of connections to the block increases.

☐ **Do Not Mix Symbols of Different Instances**

Use a unique schematic pages for each instance. Checking this option will prevent symbols of different instances from being intermixed within a single page.

**General**

☑ **Preserve Schematics**

Generate schematics in preserve mode. Use this option to preserve any manual changes made to schematic placement or to preserve the placement of symbols on specific schematic pages.

☐ **Display Net Name as Instance Pin Name**

Use the net name connected to the pin instead of the symbol pin name for FPGA components. Note that this is a purely textual overlay in the generated schematic and the symbols are not modified.

☑ **Propagate FSP Net Groups**

Propagate FSP-defined net groups into Constraint Manager. These net groups are used to created default bundles in Allegro.

**Hierarchy**

☐ **Use Actual Port Type For Hierarchical Ports**

Use the port's direction to determine which hierarchical port symbol to use. If this option is unchecked, FSP uses an 'inout' port symbol for all hierarchical ports.

☐ **Flatten Hierarchical Termination Blocks**

For terminations defined using hierarchical blocks, remove the hierarchy and place the underlying discrete components directly into the schematic.

☐ **Generate FPGA Hierarchical Blocks**

| Hierarchical Block | Block Location |
|---|---|
| J1(hshm_s200dex... | crack_lib:h41_hshm_s200dex_8ap1 |
| U1(xc7v2000tfhg1... | crack_lib:h1_xc7v2000tfhg1761 |
| U3(xc7z100ffg1156) | crack_lib:h4_xc7z100ffg1156 |

The FSP solution supports the Hierarchical methodology. In the **hierarchical method**, the design intent is captured in FSP and the schematic generated is encapsulated inside a hierarchical schematic block. FSP manages the schematic block entirely and the you need not be concerned about the contents of the block. You integrate the schematic block in a PCB design project by importing the block. You can then manage the interfaces on the symbol connected to other circuitry.

This approach is best suited when the connectivity between the FPGA and the interface components is not frozen and is likely to undergo some iterations.

In an FSP design, pins of the FPGA symbol are connected by a wire stub with the signal name and offpage connector symbols and port symbols are placed wherever required. Each interfacing component can have one or more split symbols.

4.Use various schematic related setting tabs such as *Symbol, Placement , Routing* ..so on to specify the schematic settings for generating the DE-HDL schematic.

5.Click *Finish* to generate the DE-HDL schematics.

**Note:** You can generate the PCB placement data by using the *GenerateLayoutData* tcl command to recreate the floor plan captured in FSP and to generate the initial board, the `placement.xml file` is generated.

This file contains the following information:

- Identification of FSP instances (instance ids) and their location on the board (x, y)
- Mirror to identify if the instance is to be placed on top or bottom layer on the board
- Rotation angle

# Importing FSP Design in DE-HDL

After the schematic is generated, the Message Log section displays the successful creation message and path of the schematic. You can also click the link to directly open the location. You import the generated schematic to a PCB design project and proceed with the rest of the board designing process.

## The cds.lib File

The FSP generated schematic project's `cds.lib` file contains the following entries:

1.Reference to the cds.lib file is available at the `CDS_SITE/CDSROOT`. To achieve this, the tools follows this process:

- ❑ It looks for the environment variable `CDS_SITE`. If this variable is set, it locates the file `$CDS_SITE/cdssetup/cds.lib`. If the file is available, the following statement is added to the file:

```
INCLUDE "$CDS_SITE/cdssetup/cds.lib"
```

- ❑ If the file is not found in the previous step, the environment variable `CDS_INST_DIR` is located. If this variable is set, the following statement is added to the file:

```
INCLUDE "$CDS_INST_DIR/share/cdssetup/cds.lib"
```

2.The statement for defining the worklib to the file is added:

```
DEFINE projectname_lib worklib
```

## Integrating FSP-Generated Schematic in a PCB Design Project

Perform the following tasks to include the FSP-generated schematic in your PCB design:

1.Copy the *local* library in the FSP-generated project to the project folder of the PCB design project and add the following to the `cds.lib` file:

```
DEFINE local local
```

FSP generates the instance-specific FPGA and interface instances in a library named *local*, which is not part of the standard (corporate) library. Therefore, these cells are not available automatically to the PCB design project where the FSP generated blocks are imported. You can include the *local* library in the PCB design project. However, in the complete FSP flow, the FPGA cells are likely to be modified, in which case the already imported FSP blocks will not be in sync with the updated FPGA cells. Therefore, you need to keep these cells (the *local* library) isolated in the PCB design project.

2. In Design Entry HDL, import the block using the *File – Import Design* command.

   **Note:** Ensure that you point to the `.cpm` file of the FSP-generated block.

*Important*

For the **hierarchical method**, import the FSP-generated schematic block to the PCB design project. If you import the block as a read-only block, the FSP block in the PCB design project will be dynamically updated if any changes are made to the FSP design.

3. Instantiate the block in the PCB design project.

*Important*

After importing the FSP design in the PCB design project, you need to set the `PAGE_NAME_PROP` directive to the property name, which holds the name of a page (sheet). Otherwise, page names will not be visible in the hierarchy viewer. To set the `PAGE_NAME_DIRECTIVE`, run the following commands at the DEHDL command console:

a. Type set *PAGE_NAME_DIRECTIVE* <property name>.
b. Type *hier_write* to save the hierarchy.

4. After adding the FSP block, complete the connectivity with the other components of the main design.

*Important*

You CANNOT reuse an FSP block in a Design Entry HDL schematic. If you want to replicate a design, you should do so strictly in FSP.

**Note:** FSP does not support cross probing between schematic canvas and the FSP canvas.

# Updating PCB Editor Board

Once the schematic changes are done in DE-HDL, the design need to run the Export Physical to Physical to synchronize the schematic and the board for the design.

### Updating the Board File with Schematic Changes

1. In Design Entry HDL, choose *File – Export Physical*.
2. Specify the `.brd` file of the PCB design project as the input board file.
3. Specify the output board file name.
4. Click *OK*.

### Importing Placement File

Launch PCB Editor by typing *Allegro* in Run command window. In PCB Editor, you will first import the placement data file which contains placement information for all the instance in the FSP canvas:

1. At the command console, type *place fsp* and press Enter.
2. Browse to the `placement.xml` file and click *Open*.

The fsp board is placed.

FSP generates placement data up to six places of decimal when specifying the x and y positions of components in inches, while PCB Editor supports a maximum of four decimal places. When placing components on the board, PCB Editor rounds off the values to four places of decimal. Therefore, when the FSP instances are placed in PCB Editor, you might notice a slight shift in the position of the components. In the round trip flow, the placement of FSP instances change in the FSP canvas as the coordinates are already rounded off to four places of decimals.

Now you can make the layout changes such as placement change as required. For detailed information on PCB Editor see *PCB Editor User Guide*.

## Updating FSP Design from Allegro Board (Back Flow)

Modifications to the board file, such as redrawing the board outline, changing components placement and renaming reference designators, need to be communicated back to the logical design. In FSP, you can only back annotate changes to board outline, component placements and the reference designators.

**Note:** Changes such as creating, renaming or removing nets and swap pins cannot be back annotate to the logical design.

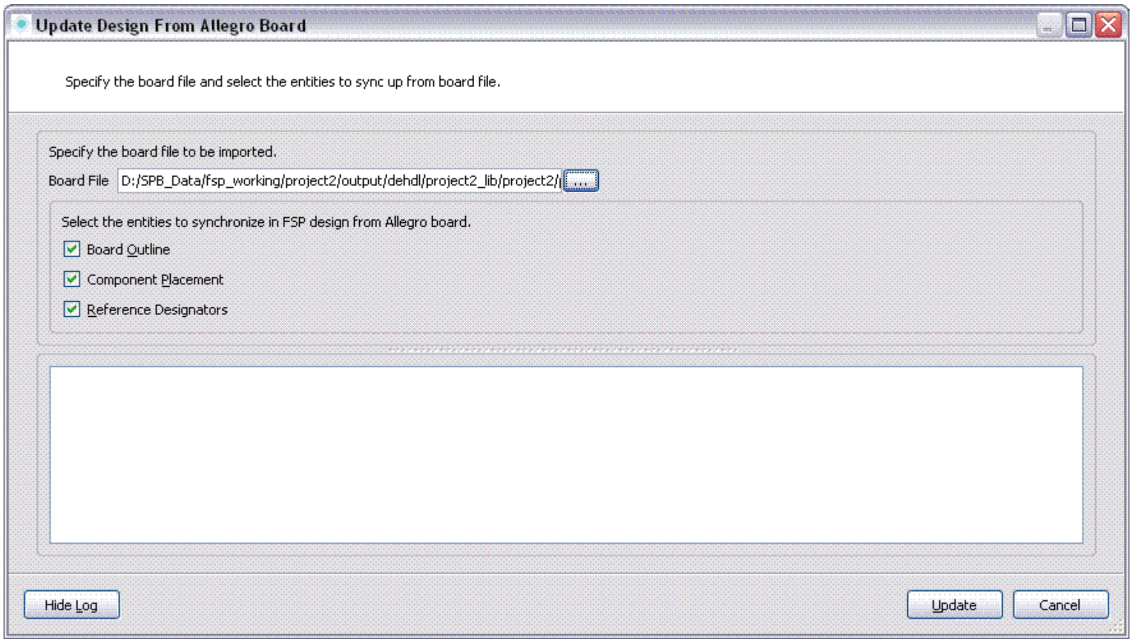
**Note:** You can also import an FSP initiated or non-FSP initiated board in FSP using *File – Import Allegro Design* menu command.

For more information about updating the layout changes or importing an Allegro design in FSP, see the Working With Board Files section of the *Allegro FPGA System Planner User Guide*

To update the FSP design with the changes made in Allegro board file, perform the following steps:

1.Choose *File - Update Design from Allegro Board*.

The Update Design from Allegro Board dialog box is displayed.



Update Design From Allegro Board

Specify the board file and select the entities to sync up from board file.

Specify the board file to be imported.

Board File D:/SPB_Data/fsp_working/project2/output/dehdl/project2_lib/project2/

Select the entities to synchronize in FSP design from Allegro board.

☑ Board Outline

☑ Component Placement

☑ Reference Designators

2. Specify the board file name and path to the board file in Board File field or click *Browse* to select the board file.
3. By default all check boxes are selected to ensure that changes to the board outline, component placement, and reference designators are imported in the FSP design. However you can modify the selection to import selective data.
4. Click *Update* to update the design.

The progress of updating design is displayed in Log window.

# Preserving Schematic in the Front and Back flow

In the FSP - Allegro flow, whether you make changes in FPGA System Planner, Allegro Design Entry, or in Allegro PCB Editor, it is important that the schematics are regenerated in the Preserve mode in order to preserve the placement of components in the generated schematic. The following two entities are preserved in the generated schematic sheet:

- Component and associated components reference designator
- Component placement in schematic hierarchy

## Recommendation for Preserving Schematic

To preserve the component reference designator and placement in the schematics, the following sequence of tasks is recommended:

After generating the schematics from FSP, you may continue to make changes in DE-HDL such as rearranging, adding or deleting passive and active. After modifying the design, you can package the design and open the Allegro board. In Allegro PCB Editor, you can perform the reference designator and routing changes.

**Note:** You may also synchronize the placement of passive and active components from FSP to Allegro, using the `place fsp` command in Allegro.

If you want to propagate the changes made in FSP to downstream tools, you must first bring FSP and Allegro designs in sync. You can synchronize the designs by importing the Allegro board in FSP. After synchronizing the FSP design with the Allegro board, you can propagate the FSP design changes to the schematic by regenerating the schematic with the *Preserve Schematics* option selected in the *Generate Allegro DE-HDL Schematics* dialog box.

You need to perform the following tasks to preserve the modifications done to the schematic generated initially.

**a.** Update the board by running *Export Physical* command from Allegro Project Manager.

**b.** Update the logical design using board file. Choose *File – Update Design from Board* in FSP.

**c.** Regenerate the schematic in the Preserve mode. Choose *Generate – Schematics* with *Preserve Schematics* option selected in FSP.

You can make the layout changes in Allegro PCB Editor and preserve the original changes by performing the following steps:

**a.** Update the logical design using the board file. Choose *File – Update Design from Board* in FSP.

**b.** Regenerate the schematics in the Preserve mode. Choose *Generate – Schematics* with *Preserve Schematics* option selected in FSP.

You can make the changes in FSP and preserve the modification done to the generated schematic and layout.

**a.** Regenerate the schematic in the preserve mode. Choose *Generate – Schematics* with *Preserve Schematics* option selected in FSP.

1

# FSP – Allegro Integration Flow

## FPGA Design Flow – The Problem

FPGA-based PCB design is an iterative and time-consuming process. This process increases the number of iterations between the FPGA engineer, the schematic engineer, and the layout engineer to finalize optimal pin assignments that improve routing on the PCB. A minor layout optimization to the FPGA design impacts the logic, schematics, and also layout. Also, changes made late in the design cycle cause significant schedule slips. Besides, during the design stage all the engineers focus on their specific areas or modules instead of considering the entire design as a single project. This results in longer design cycles.

## FSP - Allegro Integration Flow – The Solution

The FSP – Allegro Integration flow provides an intuitive environment between the different tools. It enables a non-FPGA engineer to swap FPGA pins without the knowledge of FPGA IO DRCs. The FSP – Allegro Integration flow solution keeps the design database and the engineers in sync throughout the design cycle. The end result is shorter design cycles, improved schedules and minimized possibility of errors.

In the FSP – Allegro integration flow, FSP runs as an engine under the Allegro PCB Editor. The FSP engine running in the background in Allegro PCB Editor guides you through the valid pin swaps, that conform to the FPGA DRCs. Pin swaps minimize the length of rats and crossovers.

The FSP - Allegro Integration provides two commands, Auto Pin Swap and Manual Pin Swap. These two commands provides you with the right level of control on pin swaps depending on the phase of the design. The Auto Pin Swap command absorbs the layout-specific details such as bundle flow path, breakout direction, breakout traces and automatically performs pin swaps to improve the routing scenario and provides a router-friendly pin optimization environment in Allegro PCB Editor. It’s bundling and flow planning capability enables you to visualize bundle flow patterns in Allegro PCB Editor and lets you perform pin swaps in large volumes to clean up the crossovers. On the other hand, the Manual Pin Swap command provides you the flexibility to take a finer control of your design and manually change the pin assignments to match your routing needs.

The FSP - Allegro Intergration flow provides you with maximal flexibility to adapt it to match your requirements. For example, you can choose to follow a two database approach, where you create a copy of the FSP database, associate it with the Allegro database to make changes and backannotate the changes into the original FSP database. You can also choose to use a single database and directly associate the original FSP database with the Allegro database.

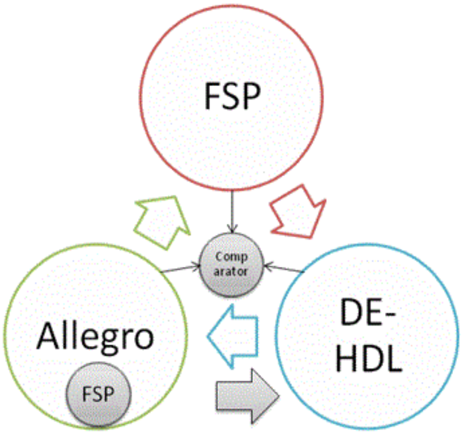
Depending on the phase of the design, you may want to allow the FSP engine to make changes that require schematic generation or set the ECO flag on the FSP datbase to restrict the FSP engine in Allegro from making any changes that require a schematic generation.

The FSP – Allegro integration flow includes the sequences of tasks performed by different engineers in different tools to design a printed circuit board (PCB). FSP engineer creates the initial design in FSP, schematic engineer performs the schematic changes in Design Entry HDL, and layout engineer targets optimizing FPGA pins in Allegro PCB Editor.

*Important*

> All the engineers need not to work on the complete flow. An engineer experienced with one or more tools used in the flow can work on the FSP – Allegro Integration flow.

The FSP – Allegro integration flow supports the FSP design database across different tools and domains. The following figure depicts how FSP design data is propagated between FPGA System Planner, Design Entry HDL, and Allegro PCB Editor.



The FSP – Allegro integration flow involves the following high-level tasks:

1. Capture the initial design in FSP, synthesize the FPGA IO pin assignments, and generate symbols and schematics.
2. Update the design in DE-HDL, perform schematic changes, and package the design.
3. Associate the FSP design in Allegro and perform manual and auto pin swaps.
4. Backannotate the changes made in the board in FSP.

The development of any design requires ECO changes or incremental changes and design synchronization between the FSP design, the schematic, and the board. Based on how you prepare your design, you make and synchronize the changes in one of the following two ways:

- The Beginning Phase

  In this phase, you may want to plan your design and would have the fexibility to make major changes in the design that require incremental schematic changes.

- The Last Phase

  In the last phase of the design cycle, you may want to restrict the FSP engine to make only ECO changes to the design that do not require a schematic regeneration.

## The Beginning Phase

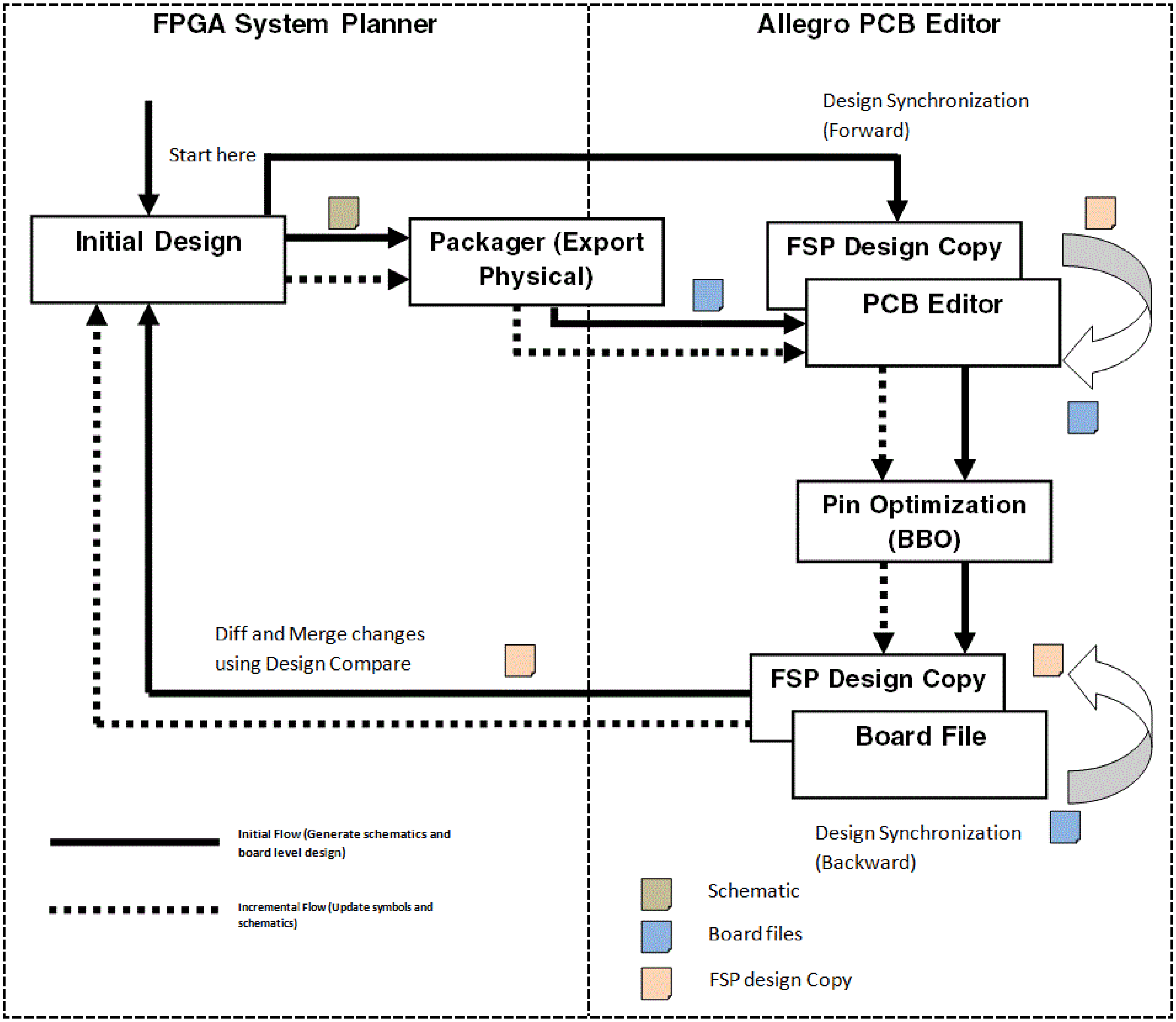
In the beginning phase, you may want to make large volume changes to the design. These changes may not be backannotated from the Allegro board to schematics and hence require schematics rengeneration. In this phase, you would not concern about schematics going out of sync with FSP and Allegro databases. You use both the pin swap commands to make connectivity changes that improve the routing scenario on the Allegro board. The changes made in Allegro are automatically backannotated to the associated FSP database.

If you are using a two database methodology, you use Design Comparison form to synchronize the changes between FSP original and copy databases. To make any further iterations, you can forward the changes in FSP copy database to Allegro PCB Editor by reassociating it with the Allegro database. Once you are satisfied with the changes you have made, you can regenerate schematics to bring back the schematics in sync with the FSP and Allegro database.

If you are using a single database methodology, you can open the FSP database in FSP and make further changes.

The following figure depicts the beginning phase of the design cycle.

The Beginning Phase


FPGA System Planner
Allegro PCB Editor
Start here
Design Synchronization (Forward)
Initial Design
Packager (Export Physical)
FSP Design Copy
PCB Editor
Pin Optimization (BBO)
Diff and Merge changes using Design Compare
FSP Design Copy
Board File
Design Synchronization (Backward)
Initial Flow (Generate schematics and board level design)
Incremental Flow (Update symbols and schematics)
Schematic
Board files
FSP design Copy

In the beginning phase, you:

1. Capture the initial design in FSP, generate symbols and schematics and take the backup of the FSP design.

   **Note:** Ensure that ECO flag is not set, unless if you want to proceed to the last phase of the design cycle.

   For detailed information, see the Project Creation and Setup (Front Flow)section the Generating Symbols and Schematics section.

2. Package the design.

For detailed information, see the Packaging the Design section.

3. Load FSP design in PCB Editor.

   For detailed information, see the Loading FSP Design in PCB Editor section.

4. Synchronize FSP design with board.

   For detailed information, see the Synchronizing Design between FSP and Allegro section.

5. Perform manual pin swaps or auto pinswaps on PCB Editor board.

   For detailed information, see the Swapping FPGA Pins in PCB Editor section.

6. Synchronize board with FSP design.

   For detailed information, see the Synchronizing Design between Allegro and FSP section.

7. Merge the changes in the original FSP design.

   For detailed information, see the Merging Changes with the FSP Design section.

8. Make changes in the FSP design and regenerate the schematic.

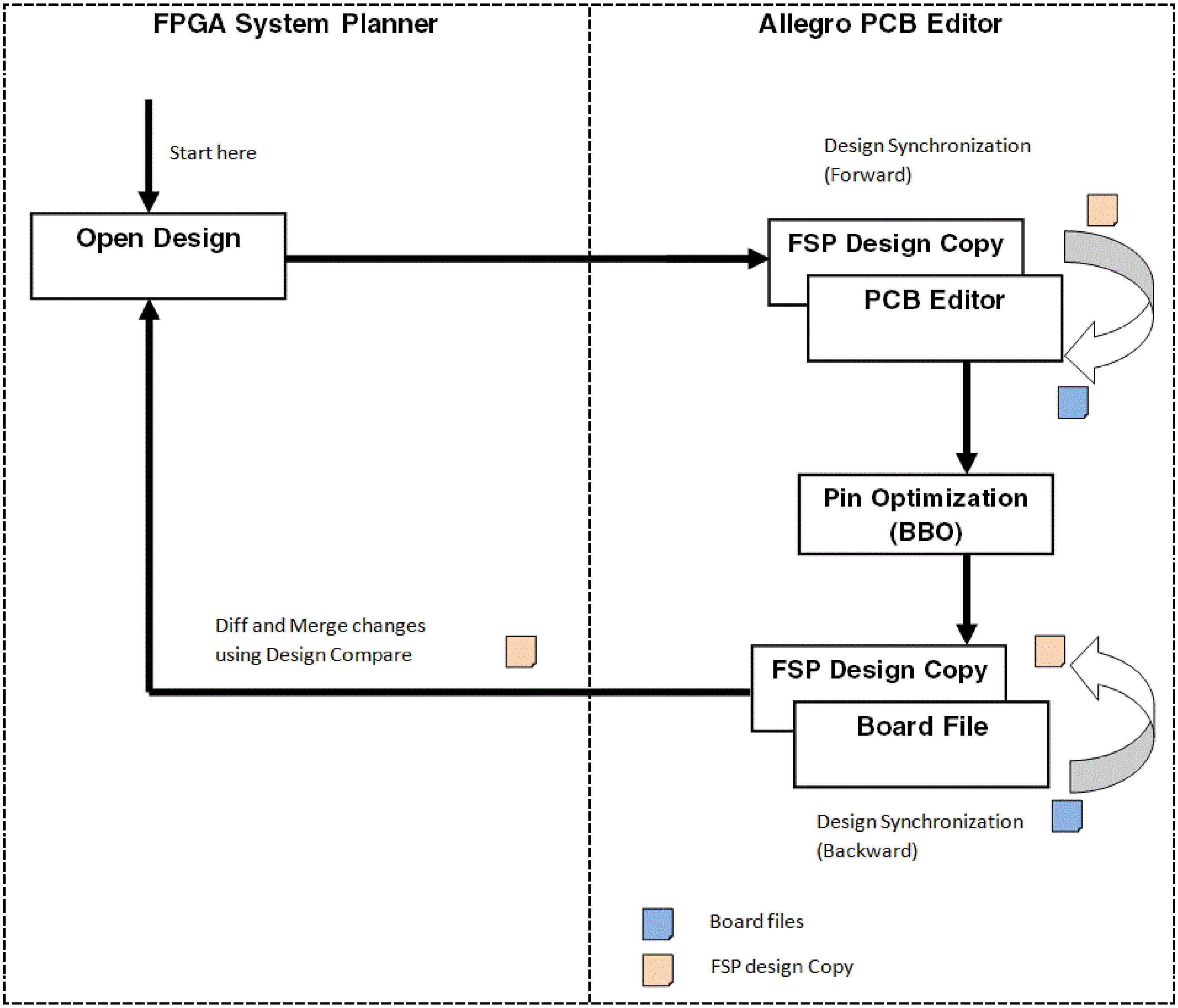
   For detailed information, see the Regenerating Symbols and Schematics section.

9. Repeat the steps from two and seven for further connectivity changes.

## The Last Phase

In the last phase, an ECO mode flag is set in the FSP design. An FSP design with ECO mode flag running in the background in Allegro PCB Editor allows pin swaps that do not require schematic regeneration. The following figure depicts the last phase of the design cycle.

## The Last Phase



In the last phase you:

1. Open the existing design and set the ECO flag in FSP.

   For more information about how to set the ECO flag, see the Setting up ECO Mode section.

2. Load FSP design in PCB Editor.

   For detailed information, see the Loading FSP Design in PCB Editor section.

3. Synchronize FSP design with board.

For detailed information, see the Synchronizing Design between FSP and Allegro section.

4. Perform manual pin swaps or auto pinswaps on PCB Editor board.

   For detailed information, see the Swapping FPGA Pins in PCB Editor section.

5. Synchronize board with FSP design.

   For detailed information, see the Synchronizing Design between Allegro and FSP section.

# 1

# Tasks to Perform in Integration Flow

## Project Creation and Setup

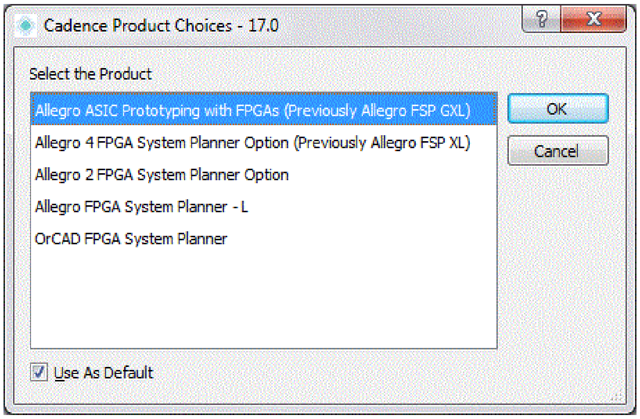
In the FSP – Allegro Integration flow, the first task you perform is creating a new project. You then begin the design process by creating a logic design and then generate a board-level design.

### Launching the FPGA System Planner

Launch FSP by typing `fpgasysplanner` in the *Run* command window or in the Command Prompt.

### Setting up the License

The *Cadence Product Choices - <Release Name>* dialog box appears, when you invoke the FPGA System Planner.



Cadence Product Choices - 17.0

Select the Product

- Allegro ASIC Prototyping with FPGAs (Previously Allegro FSP GXL)
- Allegro 4 FPGA System Planner Option (Previously Allegro FSP XL)
- Allegro 2 FPGA System Planner Option
- Allegro FPGA System Planner - L
- OrCAD FPGA System Planner

☑ Use As Default

The *Cadence Product Choices - <Release Name>* dialog box allows you to choose between different products. Depending on the product, different components and features are available.

At the beginning stage of the design cycle, you can select a higher product option. Selecting a higher product option enables the Auto Pin Swap commands in Allegro PCB Editor. The Auto Pin Swap commands are available only in the *Allegro ASIC Prototyping with FPGAs* and *Allegro 4 FPGA System Planner Option* products. At later stage of the design cycle, you can select a lower product option for the ECO changes for the same design. For example, Select the *Allegro 4 FPGA System Planner Option* product for a two FPGA design that needs to be planned and later in the design cycle when same design is targeted for ECO pin swaps you may select *Allegro 2 FPGA System Planner Option*.

To select a product from *Cadence Product Choice - <Release Name>* dialog box, perform the following steps:

1. Select a product.
2. Select the *Use As Default* option.

**Note:** Select the *Use As Default option* to invoke the selected product license every time you invoke FSP.

3. Click *OK.*

The <*product_name*>- *What do you want to do...?* dialog box is displayed. You can use this dialog box to create a new project or open an existing design.

## Creating a New Project

To create a new project, perform the following steps:

1. Select a *Create New* option.
2. Enter the design name in the *Name* field and path of the directory where you want to save the design in the *Path* field.
3. Click *OK.*

## Reading Config.ini file and Rules files

The `config.ini` file located at `$CDSROOT/share/cdssetup/fsp` contains tool configuration settings information and rules file path variable. The configuration settings is useful during working with features such as create new project, generate schematic/ symbols, and more. The rules file path variable helps FSP to identify and locate the interface library rules files and display it on the *Library Explorer*. FSP reads the `config.ini` file entries from different locations such as `CDSROOT` and `CDS_SITE` levels. The default `config.ini` file is overwritten when the tool is re-installed. It is recommended that you copy and customize the `config.ini` file outside the installation area. Set the `CDS_SITE` environment variable to point to the new location. FSP reads the available customizations from site-level `config.ini` file and remaining settings from installation level `config.ini` file. The order in which FSP looks for the `config.ini` file is:

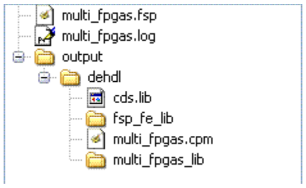
- FSP_CONFIG_FILE

- CDS_SITE
- CDSROOT

For detailed information about the `config.ini` file, refer to the see the Working with FSP Template Files section and setting up rules file path, see the Set up Rules File Search Path section in *Allegro® FPGA System Planner User Guide*.

## The Project Directory

The following project directory is created.



```
multi_fpgas.fsp
multi_fpgas.log
output
    dehdl
        cds.lib
        fsp_fe_lib
        multi_fpgas.cpm
        multi_fpgas_lib
```

## Setting Up Libraries

The `config.ini` file contains the lrfpath as variable name and location of the interface rules file as variable values. FSP reads this variable to access the interface library files. Using any text editor you can manually add libraries in FSP by specifying their logical names and physical locations in the `config.ini` file. Do not modify the `config.ini` file located at $CDSROOT. It is recommended that you create a site `config.ini` file and then do the neccessary changes.

You can also use the *Rules File Path Editor* to add, modify, or delete the libraries of the project. Both the rules and mapping files are fetched from the directories specified in the *Rules File Path Editor*. The order in which the libraries are listed in the *Rules File Path Editor* determines their search order.

When you create a project, FSP creates a default `lrf` directory in the project directory and sets it as the working directory. You can set any existing library listed in the *Rules File Path Editor* dialog box as working directory. A working directory is also included as a part of search mechanism.

For detailed information about the setting up the rules file search path, see the Set up Rules File Search Path section in *Allegro® FPGA System Planner User Guide*.

1. Choose *Library – Edit Rules File Path.*

   The *Rules File Path Editor* dialog box is displayed.

2. Click *Add* and select a folder where the rules file exists from *Browse For Folder* dialog box.

3.Click *OK* of *Browse For Folder* dialog box.

4. Select a library row and click *Set Working Dir* to set as the working directory.

5.Click *OK* of *Rules File Path Editor* dialog box.

The new library name appears in *Library Explorer*.

**Note:** By default, a cpm file is created in project directory. The cpm file contains minimum settings required for the project are stored in cpm file. You can also point your own cpm file of the master board schematics in FSP using *Settings* dialog box. Click *browse (...)* of Project CPM File field and select .cpm file. While specifying the cpm file, the associated cds.lib file should also be available in the same directory as cpm file.

In the *Settings* dialog box, you have the option to change the symbol generation directory through the *Generate Symbols In* option. Click and select the library name.

## Defining NetGroups for Nets

*Important*

> This feature is useful when you plan to perform optimization on bundles in Allegro PCB Editor.

In the FSP – Allegro integration flow, the Auto Pin Swap optimization is performed on bundles in Allegro PCB Editor. Bundles allow you to associate multiple connections and manipulate them as a single entity within the design. They enhance your visual understanding of the routing stratergy for complex designs in Allegro PCB Editor.

In Allegro PCB Editor, rats are combined into bundles based on the NetGroup definitions. A NetGroup is a group of nets that you treat as a single entity in FSP. A net can belong to only one NetGroup at a time. These NetGroups are propagated via schematics to Allegro PCB Editor.

*Important*

> The NetGroups propagated from FSP are locked in Allegro PCB Editor for editing. To make changes, you need to come back to FSP and make the required changes. After making the changes, regenerate the schematics and update the board to reflect the NetGroup changes on the Allegro board.

**Note:** The steps for creating bundles using IFP and Constraint Manager are not covered in this flow. See the, Working with Global Route Environment User Guide for more information.

In FSP, nets are allocated together based on the interface logical groups. A NetGroup can be automatically created for nets using interface logical groups. NetGroups can be created automatically or manually in FSP. Automatic creation of NetGroups is recommended over the manual method. Manually creating NetGroups is a tedious task as it requires naming hundreds of nets. You can also combine automatic and manual grouping methods. For example, you first auto-create NetGroups and then edit some of NetGroups manually.

To set up your design for NetGroups, perform the following steps:

1. Choose *File – Settings.*

   The Settings dialog box is displayed.
2. Click the *NetGroups* tab.
3. Select *first* option to auto create NetGroups on device protocol signals.

   When selected, the NetGroups are automatically created on device protocols, when no NetGroups are existing.
4. Select *second* option to auto create NetGroups for interface signals.

   When selected, the NetGroups are automatically created on interfaces based on their logical group names.
5. Specify the maximum number of nets allowable in a group.

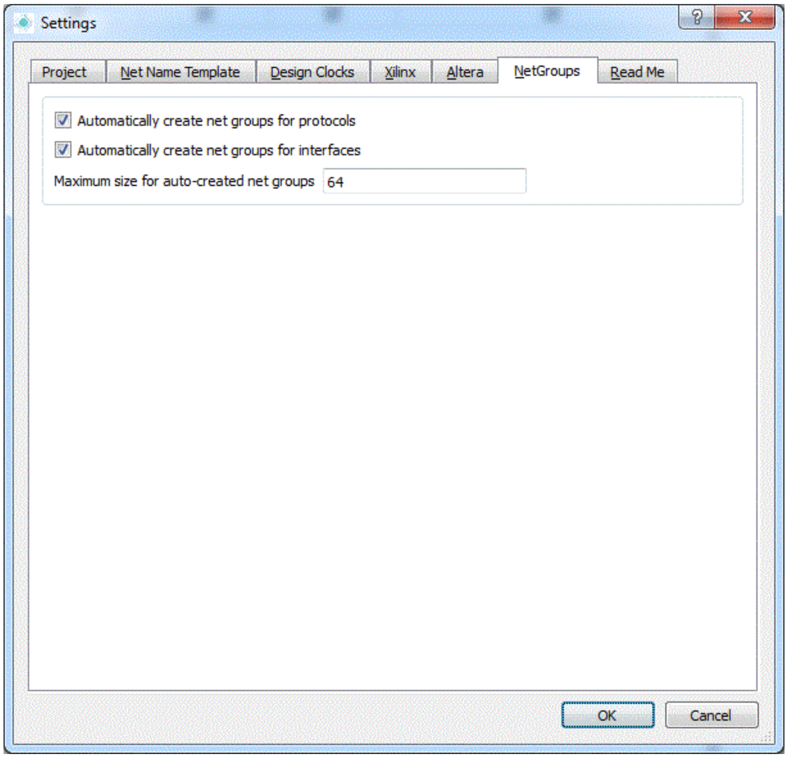
   For example, for a 64-bit size of interface group U2.Data<0>…..<63>, select the *first* option and specify the NetGroup size as 16. The NetGroups for the interface group nets are automatically defined as.

   ❑ U2.Data1<0>...<15>

   ❑ U2.Data2<16>…<30>

   ❑ U2.Data3<31>...<45>

   ❑ U2.Data4<46>...<63>



☑ Automatically create net groups for protocols

☑ Automatically create net groups for interfaces

Maximum size for auto-created net groups 64

6. Click *OK* to save the settings.

When you place the interface components on canvas or create device protocol, net groups are automatically created based on the specified parameters.

# Placing Component and Setting Target

After adding libraries, you start adding components to your design.

## Adding Interface to the FSP Design

There are two methodologies available for interface component placement. You can use any of the following methodology to start creating your design:

- Capturing FSP Design Using Real Components
- Capturing FSP Design Using Rules File or Virtual Interfaces

## Capturing FSP Design Using Real Components

Component Browser is invoked from FSP to select a symbol from the central library. If the PTF property is set for the symbols then available interface rules file for the selected symbol is displayed. You select a specific PTF row for the symbol. The selected PTF row is used to derive a proper rules file for the chosen part/primitive. After selecting FSP automatically determines and select the appropriate mapping file and rules file combination for the selected symbol. Once the exact interface rules file is found the mapping file is searched in the symbols `library:cell:view` directory. Once the interface rules file and mapping file combination is selected the associate dra file is used to draw the footprint on the canvas.

## Capturing FSP Design Using Rules File or Virtual Interfaces

When you do not have the write permission to access the central library or cannot find the required symbol using Component Browser you can continue to use Library Explorer to place the rules file on canvas. You can even create a virtual interface if you still do not find your desired part in library explorer. Note that the rules file placed using Library explorer or the virtual interface is just a logical rules file which means they are not mapped with any of the front-end symbol or footprint. You can continue your design by capturing the connectivity. Once you complete your design you can convert the interface rules file or virtual interface to real component using mapping file before generating schematics.

This flow describes the Capturing FSP Design Using Real Components steps. For detailed information on the Mapping File and Component Placement methdology and various scenarios (such as mapping file, PTF selection) involved in both the flows refer to *Working with Components* chapter in *Allegro® FPGA System Planner User Guide.*

To place the interface component on canvas you can choose any of the following forms:

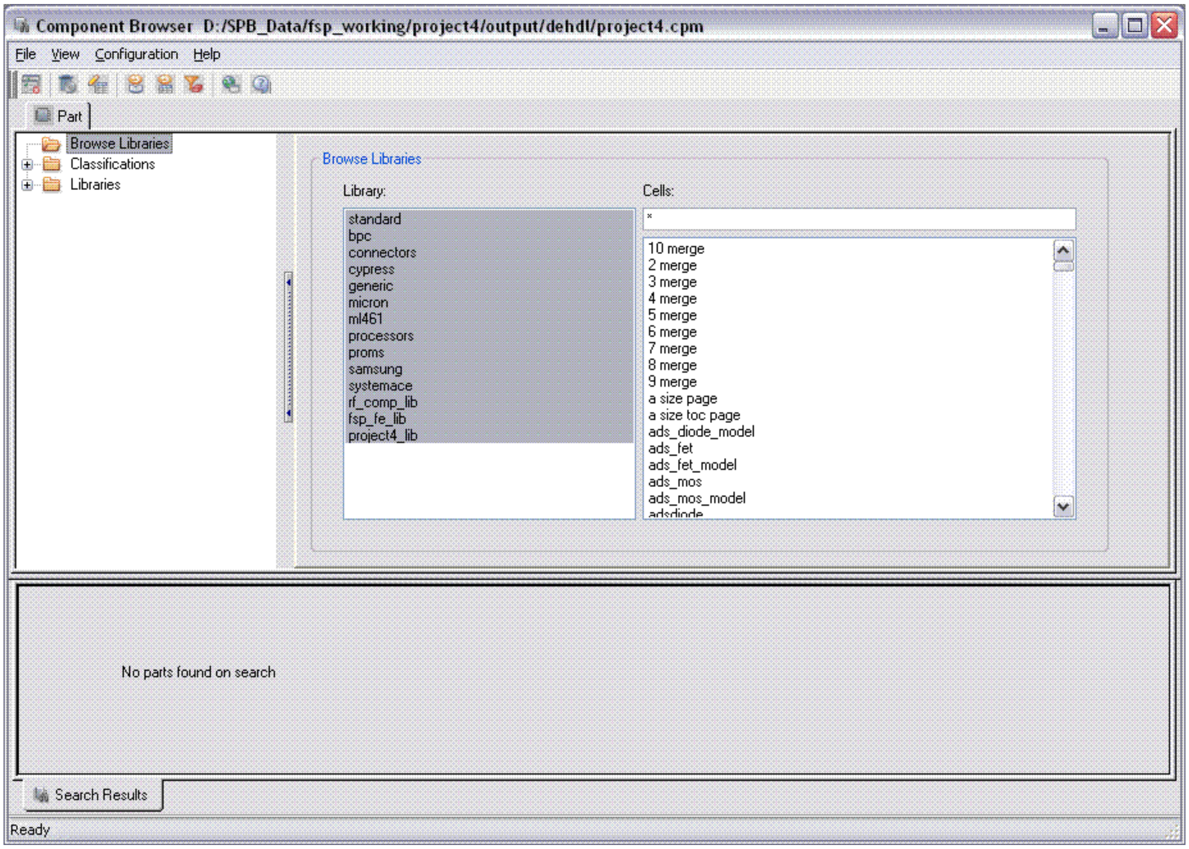
- Using Create/Select Component Placement Rules and Mapping Information Wizard
- Using Add Part dialog box.

The steps below are described using Add Part dialog box. It is also assume that you have both mapping file and interface rules file for the design. Incase if you do not have the mapping file you need to create a new mapping file, for more information see *Working with Components* chapter in *Allegro® FPGA System Planner User Guide.*

To place the component on FSP canvas perform the following steps:

1. To invoke Component Browser click *Add Part* icon in toolbar.

   The Component Browser is displayed.



Component Browser D:/SPB_Data/fsp_working/project4/output/dehdl/project4.cpm

Library:

standard
bpc
connectors
cypress
generic
micron
ml461
processors
proms
samsung
systemace
rf_comp_lib
fsp_fe_lib
project4_lib

Cells: *

10 merge
2 merge
3 merge
4 merge
5 merge
6 merge
7 merge
8 merge
9 merge
a size page
a size toc page
ads_diode_model
ads_fet
ads_fet_model
ads_mos
ads_mos_model
adsdiode

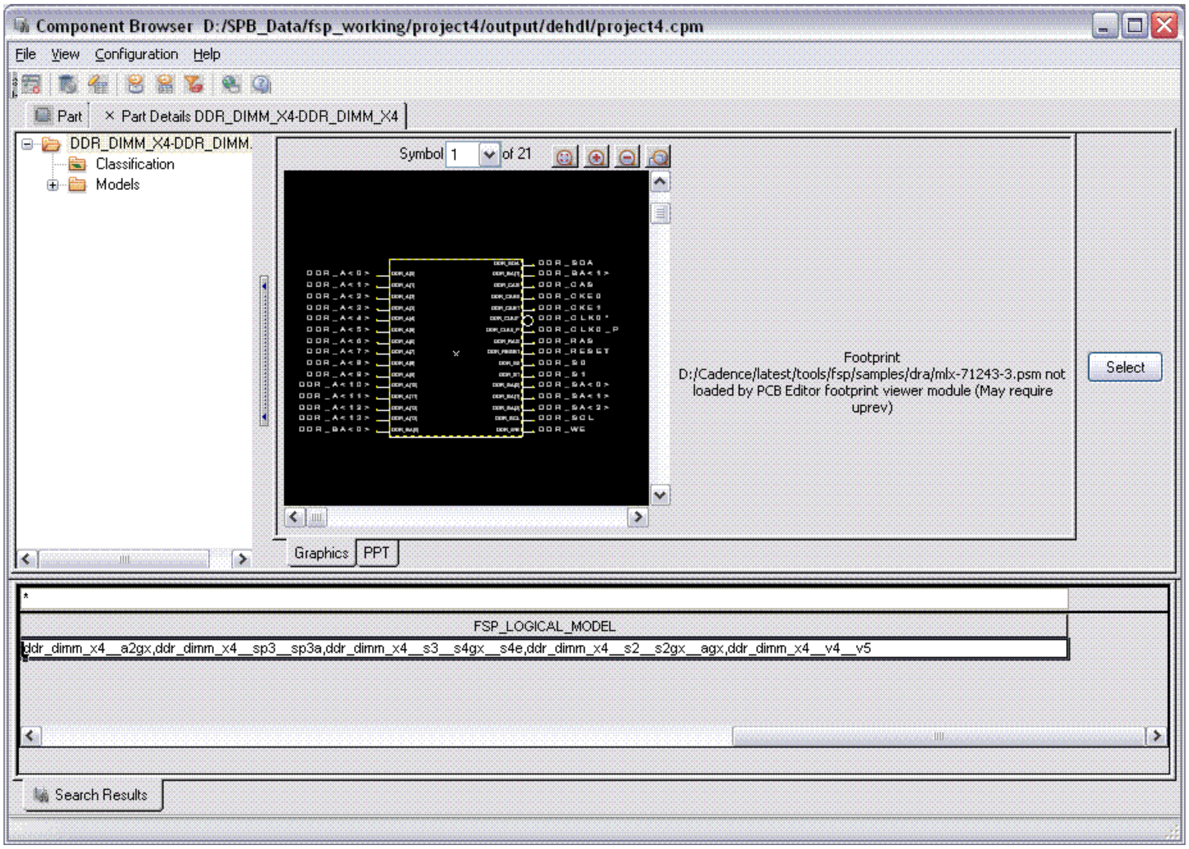
No parts found on search

2.Select a library name in Library pane, whose component you want to bind with the FSP logical model.

3.Select a cell name in Cells pane or enter the name of cell in Cells text box.

4.In Search Results pane, click the row corresponding to the physical part you want to add.

The symbol and footprint name for the component is displayed in Part Name tab.



Component Browser D:/SPB_Data/fsp_working/project4/output/dehdl/project4.cpm

Symbol: 1 of 21

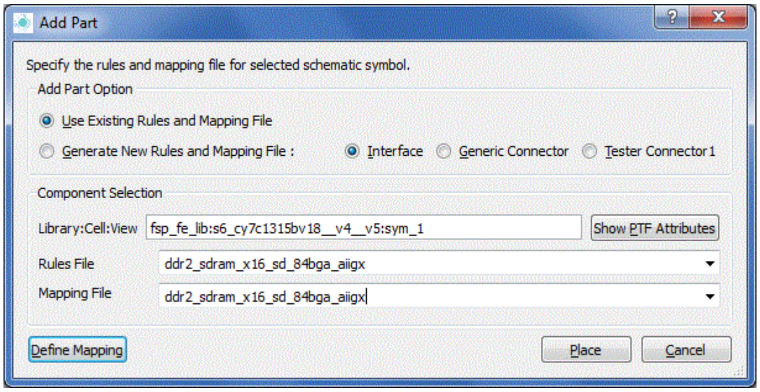
Footprint D:/Cadence/latest/tools/fsp/samples/dra/mlx-71243-3.psm not loaded by PCB Editor footprint viewer module (May require uprev)

| FSP_LOGICAL_MODEL |
|---|
| ddr_dimm_x4__a2gx,ddr_dimm_x4__sp3__sp3a,ddr_dimm_x4__s3__s4gx__s4e,ddr_dimm_x4__s2__s2gx__agx,ddr_dimm_x4__v4__v5 |

5.Select a symbol view from Symbol drop down list.

6.Click *Select*.

The Add Part dialog box is displayed. Since you have both rules file and mapping file by default the rules file name and mapping file name is displayed in respective fields.



Add Part

Specify the rules and mapping file for selected schematic symbol.

Add Part Option

● Use Existing Rules and Mapping File

○ Generate New Rules and Mapping File : ● Interface ○ Generic Connector ○ Tester Connector1

Component Selection

Library:Cell:View: fsp_fe_lib:s6_cy7c1315bv18__v4__v5:sym_1

Rules File: ddr2_sdram_x16_sd_84bga_aiigx

Mapping File: ddr2_sdram_x16_sd_84bga_aiigx

7.Click *Place* to place the interface component on canvas.

## Adding Device to the FSP Design

Device rules file are treated differently compare to interface rules file. Before you begin the placement, you need to specify the device device rules file name in `cell/library/PTF` row as `FSP_FPGA` property value. The `FSP_FPGA` property helps the Component Browser to identify the component as device component otherwise the component will be treated as interface which leads you to error prone state.

For detailed information on device placement methdology see *Working with Component* chapter in *Allegro® FPGA System Planner User Guide.*

To place the device rules file on canvas using Component Browser perform the following steps:

1. Click *Add Part* icon in toolbar.

   The Component Browser dialog box is displayed.
2. Browse for the library whose component you want to add in Library Pane.
3. Select a cell name in Cells pane or enter the name of cell in Cells text box.
4. In Search Results pane, click the row corresponding to the physical part you want to add.

   The symbol and footprint name for the component is displayed in Part Name tab.
5. Select a symbol view from *Symbol* drop down list.
6. Click *Select*.

   After clicking Select, a graphical view of device is displayed. Left click to drop the device rules file on canvas and right click to disable the graphical view.

**Note:** After clicking Select in Component Browser if you unexpectedly see the Add Part dialog box, then you must specify the `FSP_FPGA` attribute and re-invoke the Component Browser to place the device rules file on canvas.

## Setting Target to Device Instance

After placing the interface and device instance on canvas, target need to be specified. Targeting device decides which interface will connect to which device.

To target the device, perform the following steps:

1. Right-click on the interface instance and do any of the following:

   ❑ Choose *Target To Device – <Instance Name>*.

   After you choose this option, all the groups of the interface are targeted to device at one go.

   ❑ Click *Instance Properties.*
2. Click the *Group Settings* button in the *Properties window.*

   The *Group Settings for Interface Instance<inst_name>* is displayed.

3. To target all the groups, click the *Connect to Device* column name, click on any one of the drop-down button, and select the device instance name from the drop-down list.
4. To target a single groups, click the drop-down button in the first group under *Connect to Device* column and select the instance name.

**Note:** You can perform the same step for other groups.

5. To target the interface groups to a specific bank of the targetted device, do the following:

   **a.** To target all the interface groups to one bank, click on the *Use Bank* column name and click on any one of the drop-down button.

   A pop-up menu is displayed with package view of the targeted device and the list of banks available in the device.

   **b.** Select a bank number from the list.

   **c.** Click *OK.*

**Note:** To select a bank from the package view, select a pin of a bank to which you want to target the interface group. After you select the pin, the remaining pins of the bank are automatically selected and highlighted.

6. Click *OK* in the Group Settings for Interface Instance <inst_name> dialog box.

The interface instance group (s) is targeted to the device instance.

## Running Design

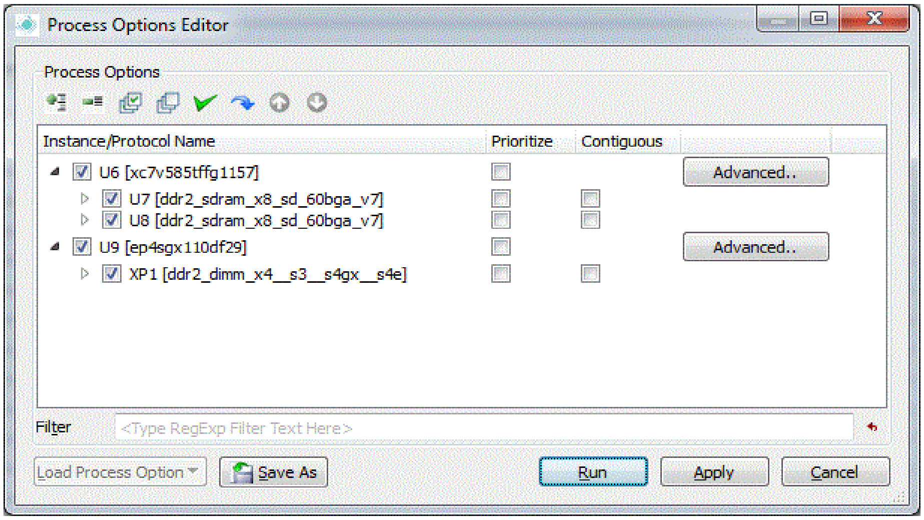
After you target the interface components to respective devices, you are ready to make the connections between them. Before running a design you set the preferences such as setting an order to an interfaces and its groups for example, selecting few device and interface instances for pin assignments, and defining the advance options for better results.

For detailed information, on how to run the design per instance and various process options see the *Running a Design* chapter in *Allegro® FPGA System Planner User Guide*.

To make connections between the components perform the following steps:

1. Choose *Tools – Run Design.*

The Process Options Editor dialog box is displayed.



| Instance/Protocol Name | Prioritize | Contiguous | |
|---|---|---|---|
| ☑ U6 [xc7v585tffg1157] | □ | | Advanced.. |
| ☑ U7 [ddr2_sdram_x8_sd_60bga_v7] | □ | □ | |
| ☑ U8 [ddr2_sdram_x8_sd_60bga_v7] | □ | □ | |
| ☑ U9 [ep4sgx110df29] | □ | | Advanced.. |
| ☑ XP1 [ddr2_dimm_x4__s3__s4gx__s4e] | □ | □ | |

2.Click *Advance* to specify different proximity options as per required.

3.Click *Run* to run the design.

The pin assignments and connections between the instances are automatically established.

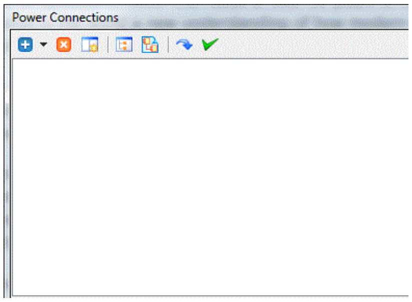
# Adding and Mapping Power Regulators

After making connections, you can start adding power regulators. Power regulators can be edited at anytime during the design. You define new regulators and assign power voltage to each regulators. Regulators can be defined manually of your own choice or can be defined automatically. After defining the power regulators you map them with associated FPGA power pins.

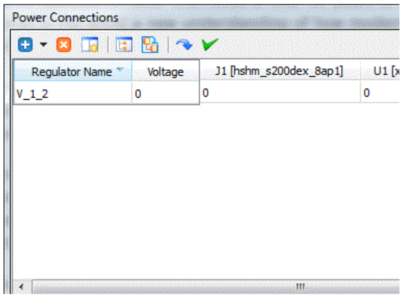
To define the power regulators, perform the following steps:

1.Choose *Tools – Power Connections.*

The *Power Connections* pane is displayed.



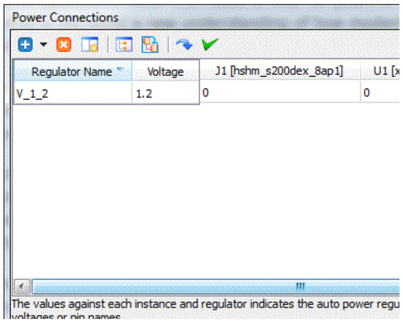
2. Click the **+** icon to add a new row.
3. Type a name under the *Regulator Name* column.



Power Connections

| Regulator Name | Voltage | J1 [hshm_s200dex_8ap1] | U1 [x |
|---|---|---|---|
| V_1_2 | 0 | 0 | 0 |

4. Enter a numerical value under the *Voltage* column.



Power Connections

| Regulator Name | Voltage | J1 [hshm_s200dex_8ap1] | U1 [x |
|---|---|---|---|
| V_1_2 | 1.2 | 0 | 0 |

The values against each instance and regulator indicates the auto power regu
voltages or pin names

To define regulators automatically, perform the following steps:

1. Click the *Auto Add Regulator* option.

   A confirmation window is displayed about adding new regulators in the design.

2. Click *Yes* to proceed further.

   The new regulator names and values are listed in the power connections window.

To avoid the manual mapping process, FSP provides an option to automatically map the regulators for the complete design.

To map power regulators automatically, perform the following steps:

1. Select the check boxes under the instance name columns for which you want to map regulators.
2. Click in the *Power Connections* window.

   A confirmation window is displayed about mapping power regulators.

3. Click *Yes* to reset.

The power regulators are mapped to the power pins in the *Regulator Name* column of *Design Connectivity window.*

# Defining Terminations, Decoupling Capacitors, and External Ports

For more information, see the Defining Terminations, Decoupling Capacitors and External Ports section in the *Tasks to Perform in Front and Back flow* chapter.

# Generating DE-HDL Symbols and Schematics

For more information, see the Generating DE-HDL Symbols and Schematics section in the *Tasks to Perform in Front and Back flow* chapter.

# Preparing FSP Design for Integration

FSP stores the complete database as a single file. You may choose to work directly on the FSP design or take a backup of the FSP design and directly work on the original FSP design. You can also work on the backup design and later merge the changes with the original FSP design.

A single FSP design (.fsp) can be send to the layout engineer for optimization. The layout engineer can use the FSP design file (.fsp) as FSP engine, by loading FSP design file in Allegro PCB Editor to perform pin swaps. All swap and optimization changes made within Allegro PCB Editor are updated simultaneously in the FSP design file. This FSP design file can be sent back to the FSP engineer to merge it with the FSP design.

**Note:** If the engineers are working on the same network, they can use a single FSP design file for integration or they can share the FSP design file if they are at geographically separate locations.

## Creating a Copy of FSP Design

After completing your design project, you create a backup of your design by using any one of the following main menu options.

- Using *File – Create Design Copy*.
- Using *File – Save As.*

This design copy is used as an FSP engine in Allegro PCB Editor to perform pin swaps. In the beginning phase, it is recommended that you take a backup of your design and work on the design copy. However, in the last phase of the design cycle, you do not need to create a backup of your design if you plan to work directly on the master design. You can directly synchronize your master design with the Allegro database and perform pin swaps.

To create a backup of your design, perform the following steps:

1. Choose *File – Create Design Copy.*

   The Specify Design Copy Path dialog box is displayed and, by default, displays the output folder of the current project and file name with format `<design_name>_<copy>.fsp` in *File Name* field.

2. Click *Save* to save the design in the same directory with the default name or browse to the different directory and enter a new name.

The Log window displays the successful creation message and path of the design copy. You can also click the link to directly open the directory.

## Packaging the Design

After you have created the design copy, run the *Export Physical* command from Project Manager to synchronize the schematic with the board design. However, you can also make changes in the schematic before packaging the design. Changes such as signal name and reference designator changes can be updated easily in the board. Any other changes made in the schematic cannot be backannotated to the FSP design in the FSP – Allegro integration flow.

To update the board with the schematic, perform the following steps:

1. Open the *Run* window and enter *projmgr*.

   The Cadence Product Choices dialog box is displayed.

2. Select a suitable product and click *OK.*

   The Project Manager window is displayed.

3. Choose the *Design Sync* icon from the Project Manager window and click *Export Physical.*

   The Export Physical dialog box appears.

4. Enter the name of the existing PCB Editor file that needs to be updated in the *Input Board File* field or click *Browse* to browse to the input board file.
5. Enter the name of the resulting updated file in the *Output Board File* field or click *Browse* to browse to the output board file.

   If you specify the output board file as the same as the input board file, Packager-XL overwrites the existing file. If you specify a new file (<any_name>.brd), a new board file is created.

6. Click *OK.*

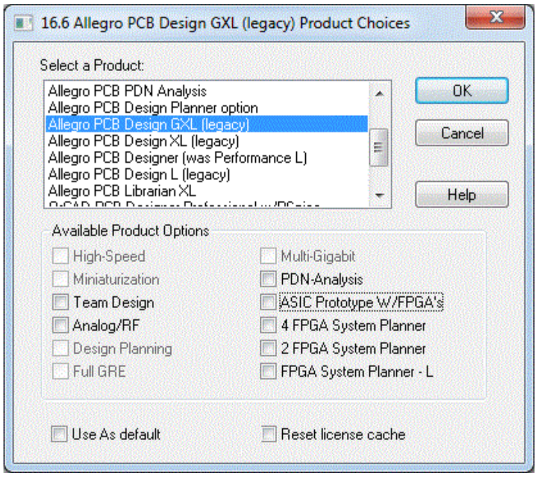
The Progress Window appears displaying the progress of the packaging process. The design is considered packaged when the packaging process is completed.

## Launching the PCB Editor

Enter *Allegro* at run command and click *OK,* to start the PCB Editor*.* The *<Release Name> <Product Name> Product Choices* dialog box is displayed.

## Setting up the License

All the FPGA System Planner product options are only available in the *Allegro PCB Design GXL, XL, L (Legacy)* series products. You may choose between different *Allegro PCB Design* products and the available FSP product options. Depending on the selected products different features and options are available in the Allegro PCB Editor canvas. For example, Interconnect Flow Planner (IFP) mode is not available in the lower capability product (*Allegro PCB Design L*) and *Auto Pin Swap* feature is not available when you choose *2 FPGA System Planner* or *FPGA System Planner - L* product options.



16.6 Allegro PCB Design GXL (legacy) Product Choices

Select a Product:

Allegro PCB PDN Analysis
Allegro PCB Design Planner option
Allegro PCB Design GXL (legacy)
Allegro PCB Design XL (legacy)
Allegro PCB Designer (was Performance L)
Allegro PCB Design L (legacy)
Allegro PCB Librarian XL

Available Product Options

| | |
|---|---|
| □ High-Speed | □ Multi-Gigabit |
| □ Miniaturization | □ PDN-Analysis |
| □ Team Design | □ ASIC Prototype W/FPGA's |
| □ Analog/RF | □ 4 FPGA System Planner |
| □ Design Planning | □ 2 FPGA System Planner |
| □ Full GRE | □ FPGA System Planner - L |

□ Use As default □ Reset license cache

To choose a product, perform the following steps:

1. Select a product *Allegro PCB Design GXL, XL or L (Legacy)* options in the *Select a Product* pane.
2. Select any one from the following options based on the number of components used in your design.
   - ❑ ASIC Prototype W/FPGA’s
   - ❑ 4 FPGA System Planner
   - ❑ 2 FPGA System Planner
   - ❑ FPGA System Planner - L
3. Click *Use as Default* if you want to use the selected product option as the default product choices when every time you invoke the PCB Editor.
4. Click *OK.*

The Allegro PCB Editor canvas is displayed. At anytime during the design cycle, you may change the product options based on your preferences.

### Loading FSP Design in PCB Editor

Initially you need to load the FSP design in Allegro PCB Editor using *FSP Load database* dialog box. When the load process completes, FSP design starts working as backend engine in the background in Allegro PCB Editor. With FSP design running in the background, Allegro PCB Editor provides an interactive environment allowing you to make pin swaps in real time ensuring that the pin swaps that you make are correct for FPGAs.

*Important*

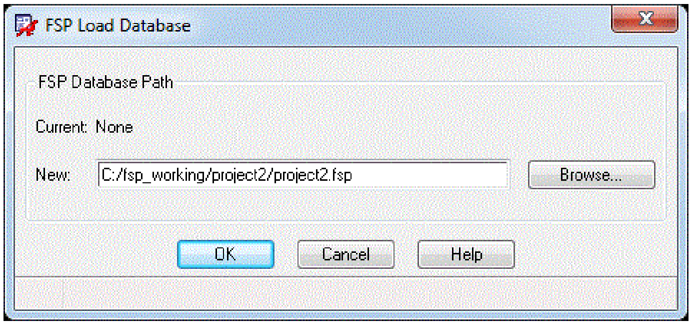
> During the design cycle, you perform this step only once at the initial stage. From next time, you can directly invoke Allegro PCB Editor and synchronize the design.

To load the FSP design in PCB Editor, perform the following steps:

1. Choose *Place – FPGA System Planner – Load Database.*

   The FSP Load Database dialog box is displayed.

2. Click *Browse...* to choose the location where the design copy is stored.



FSP Load Database

FSP Database Path

Current: None

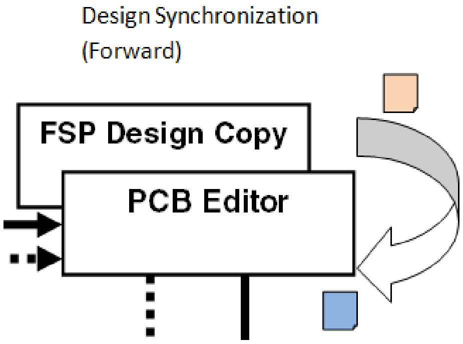
New: C:/fsp_working/project2/project2.fsp

3. Click *OK* to import the design copy.

The FSP Engine Status window is displayed mentioning that FSP engine is starting and connectivity is being verified. The *FSP Synchronize* dialog box is automatically displayed, when loading completes.

## Synchronizing Design between FSP and Allegro

In the FSP – Allegro integration flow, at this stage the board and the FSP designs may be out of sync. To start making pin swaps, you need to synchronize the FSP design with the board.



The following three changes may occur during the transfer of the design from FSP to board:

1. Pin Swaps

   These are the connectivity differences between FSP and Allegro databases.

2. Schedules

   These are net schedule differeces on the multi-segment nets such as deep and wide or multi-point connections between FSP and Allegro databases.

3. Placement

   These are placement changes between FSP and Allegro databases.

After the FSP design is loaded, the databases are verified for their compatibility.

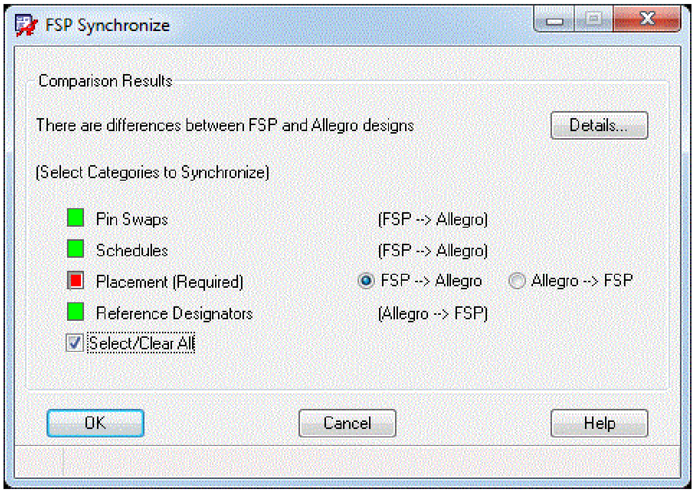
If the databases have irreconcilable differences that cannot be merged, you will not be allowed to proceed further or use any of the pin swap commands. In such case, you will have to regenrate the schematics, package the design, and update the board to bring back FSP and Allegro databases in sync.

If the databases have differences that can be merged, the *FSP Synchronize* dialog box is displayed. In the *FSP Synchronize* dialog box, any category with red color indicates that differences were found in the corresponding category. The green color indicates no differences found, and the yellow color indicates that differences can be ignored during synchronization.

To synchronize the FSP design with the Allegro database, perform the following steps:

1. Select *Select All* option.

   For *Placement (Required)* option, the *FSP –> Allegro* sub option is selected by default if no components are placed on the Allegro canvas. Otherwise, the *Allegro –> FSP* option is selected by default.



FSP Synchronize

Comparison Results

There are differences between FSP and Allegro designs

(Select Categories to Synchronize)

🟩 Pin Swaps (FSP --> Allegro)

🟩 Schedules (FSP --> Allegro)

🟥 Placement (Required) ● FSP --> Allegro ○ Allegro --> FSP

🟩 Reference Designators (Allegro --> FSP)

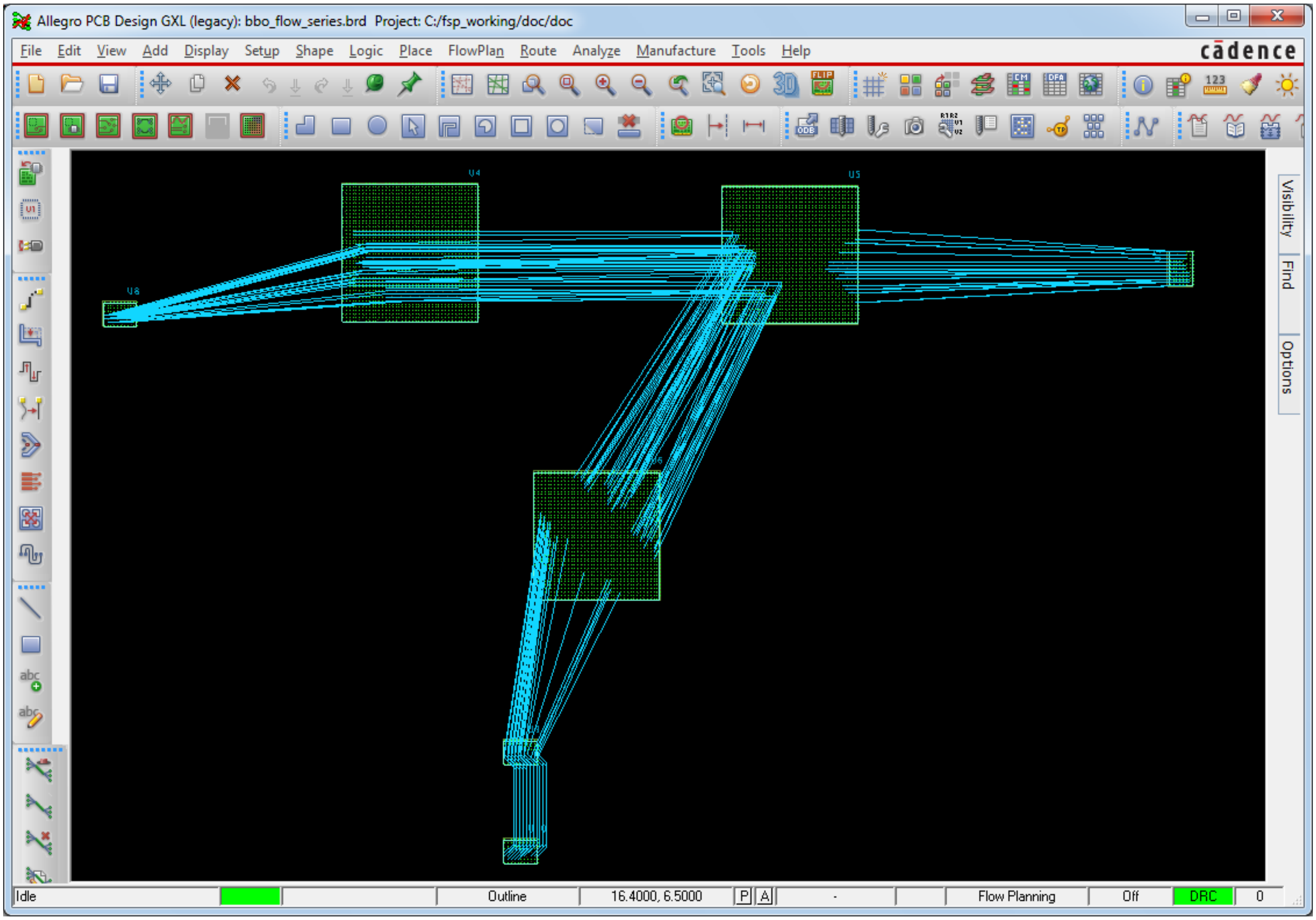
☑ Select/Clear All

2. Click *OK.*

The components with ratsnests are displayed on the Allegro PCB Editor canvas.

## Swapping FPGA Pins in PCB Editor

When synchronization completes, the design somewhat looks as shown in the following figure. The design shown in the following figure is used as an example to explain different types of optimization commands in this section.


Allegro PCB Design GXL (legacy): bbo_flow_series.brd Project: C:/fsp_working/doc/doc
File Edit View Add Display Setup Shape Logic Place FlowPlan Route Analyze Manufacture Tools Help
cādence
U4
U5
U8
Visibility
Find
Options
Idle
Outline
16.4000, 6.5000
P
A
-
Flow Planning
Off
DRC
0

Before using different optimization commands, it is important that you understand and gain more experience with different optimization commands. This may help you to adopt a more custom approach based on your specific designs and work environment.

## Setting up the Design for Optimization

You will now learn to set up the design for optimization in the IFP environment. The IFP environment exists within Allegro PCB Editor as an application mode. Activate the IFP mode before you set up the design.

The IFP application mode can be activated by performing any one of the following steps:

- Enter `ifp` in the Command Console window.
- Choose *Setup – Application Mode – Planning Mode.*
- Click the IFP Application Mode toolbar button.

## Displaying Bundles

Once the IFP application mode is activated, bundles need to be displayed to perform optimization. Bundles are created by IFP based on the NetGroup definitions. The NetGroups may be either those defined by you in Allegro or those are propagated from FSP through schematics. If you do not want to consider bundling using NetGroup property, you can bundle the rats using Constraint Manager. For detailed information about bundling the rats using Constraint Manager, refer to the Global Route Environment User Guide*.*

To view bundles, perform the following step:

- Right-click on the canvas and choose *Show All – Bundles.*

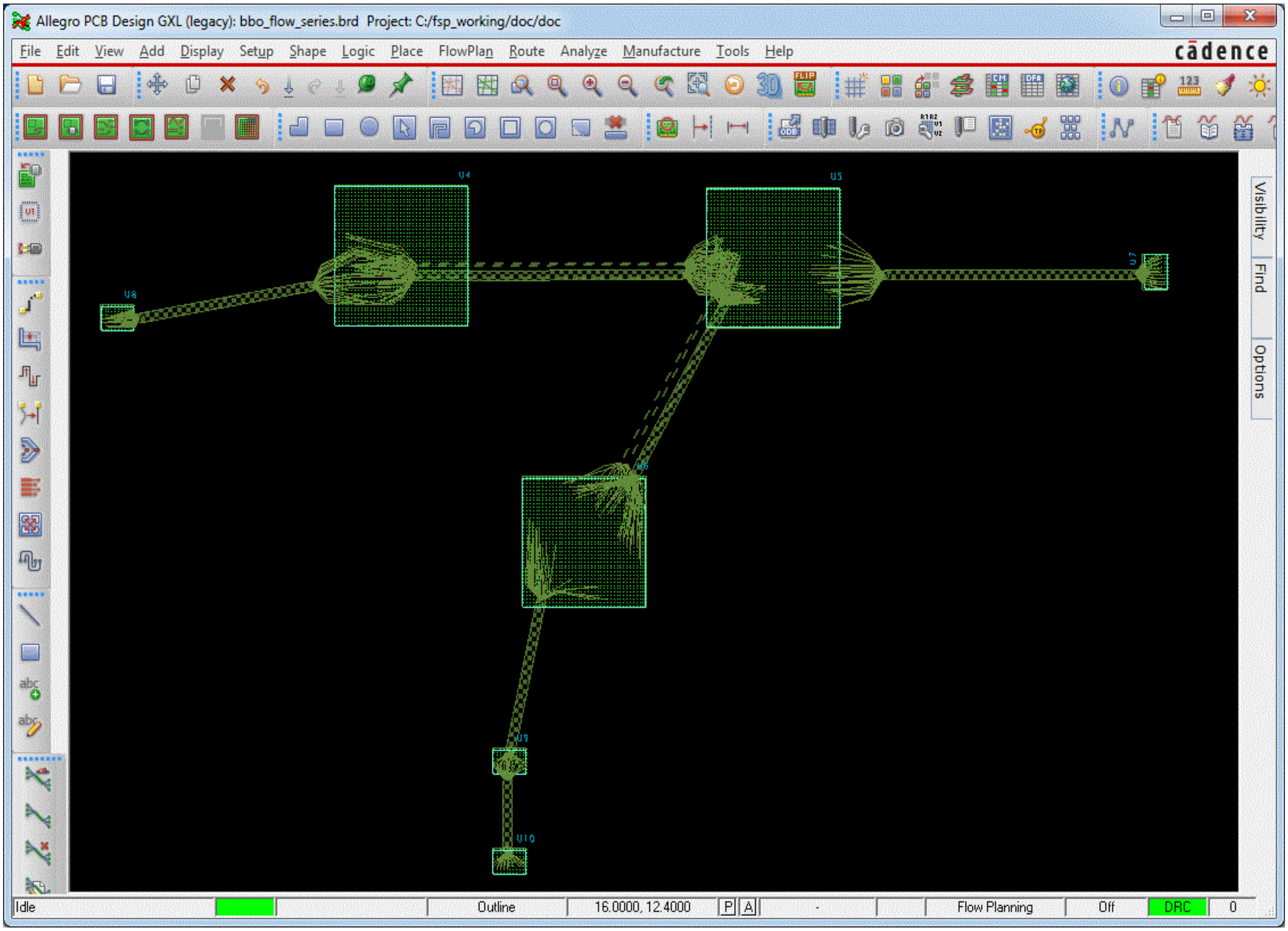
**Note:** Make sure that nothing is selected on the canvas, before performing the step.

To remove the ratsnests from the canvas, perform the following step:

- Right-click on the canvas and choose *Blank All – Rats.*

The following figure depicts the example design with bundles. You are now ready to perform optimization on bundles.

**Note:** At any time during the design cycle, you can modify the bundle definitions. Bundle modification tasks are available in the right mouse button menu options. For detailed information, refer to the Global Route Environment User Guide.


Allegro PCB Design GXL (legacy): bbo_flow_series.brd Project: C:/fsp_working/doc/doc
File Edit View Add Display Setup Shape Logic Place FlowPlan Route Analyze Manufacture Tools Help
cādence
U4
U5
U7
U8
U6
U9
U10
Visibility
Find
Options
Idle
Outline
16.0000, 12.4000
P
A
Flow Planning
Off
DRC
0

## Performing Pin Swaps on Bundles

This section describes different types of optimizations and the tasks associated with each type of optimization. The following are two different pin swaps methods.

- Auto Pin Swap
- Manual Pin Swap

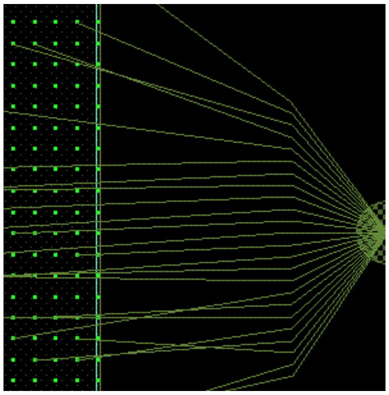
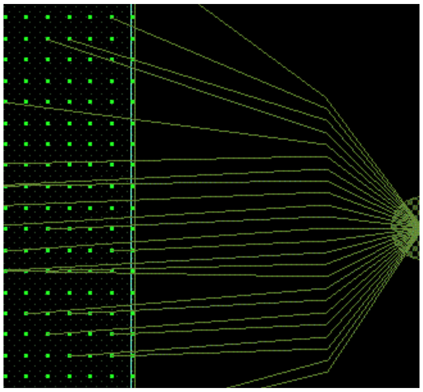
**Note:** This section does not include a detailed description of the optimization tasks and other IFP tasks which are required while routing the design. It is assumed that you are familiar with graphic user interface of the IFP within Allegro PCB Editor. For more information about the different types of optimizations, see the Allegro PCB and Physical Layout Command Reference: F Commands Guide and about the IFP tasks, see the Working with Global Routing Environment User Guide.

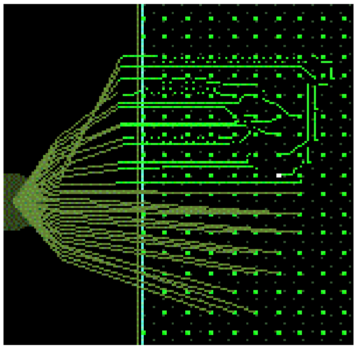
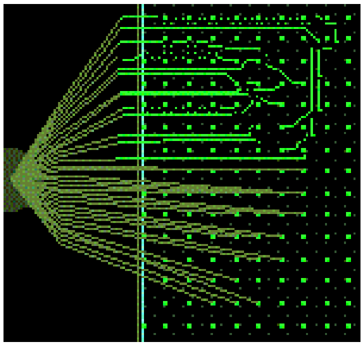
## Auto Pin Swap

The Auto Pin Swap command is also called as Breakout Based Optimization (BBO) because breakout locations are considered during optimizations. The Auto Pin Swap commands are the design solution designed to reduce the gap between layout and IO synthesis. The command enables a router-friendly pin assignment that minimizes the length of the rats and number of crossovers on the PCB. The command has an inbuilt ability to assess the routability of the assigned pins and swap pin to achieve minimum crossovers. This helps in minimizing the number of routing layers. Auto Pin Swap command restricts its scope based on the phase of the design. In the beginining phase, i.e., when the associated FSP design does not have an ECO flag, the command might perform pin swaps that require schematic regeneration to update schematics. In the last phase, i.e., when the associated FSP design has the ECO flag set, the command will make only those swaps that can be backannotated to the schematics from Allegro PCB Editor.

In Allegro PCB Editor, the Auto Pin Swap command provides three options. Depending on the state of routing on the bundle, you may choose one of the options. To optimize a bundle using Auto Pin Swap command indicates that, you wish the FSP engine to consider the layout specific data such as gather points and breakout/fanout locations to figure out the best connections for the bundle on the selected FPGA.

The following table outlines the advantages of considering the breakout locations.

| Scenario | Before Optimization | After Optimization |
|---|---|---|
| Rats from bundle gather point to pinout locations (without breakout) |  |  |

| | | |
|---|---|---|
| Rats from bundle gather point to pinout locations (with breakout) |  |  |

In Allegro PCB Editor, you use *Auto Pinswap* dialog box to perform optimization. Three different options are available for pin swaps in the Auto Pinswap dialog box.

**Note:** Before performing optimization using Auto Pin Swap commands, make sure that all the fanouts or breakouts for the respective components are properly drawn. You can do this by choosing *Route – Connect* option from main menu*.*

The following section helps you in selecting an appropriate Auto Pin Swap command based on the routing state of the bundle and on the type of changes you want to make.

### *Rake Order Based Optimization*

The rake order specifies the order of rakes drawn from bundle gather point to the routing end point.

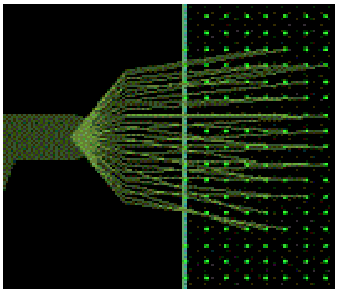
To perform rake order optimization, perform the following steps:

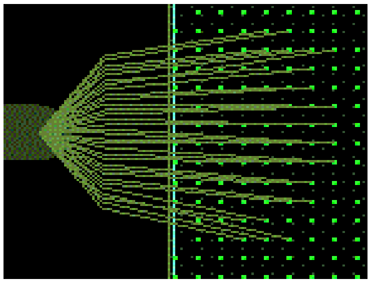
1. Click on the bundle*.*
2. Choose *Place – FPGA System Planner – Auto Pinswap.*

   The *FSP Auto Pinswap Option* dialog box is displayed.
3. Select *Rake Order* option and click *OK.*

The rats from the selected bundle gather point to the rakes are cleaned up. See the following figure.

| **Rats before optimization** | **Rats after optimization** |
|---|---|





### *Breakout Order Based Optimization*

The Breakout order is the radial order of the breakout etches from the center of the BGA. The breakout order optimization should be used after breaking out the pins on both sides of the bundle.

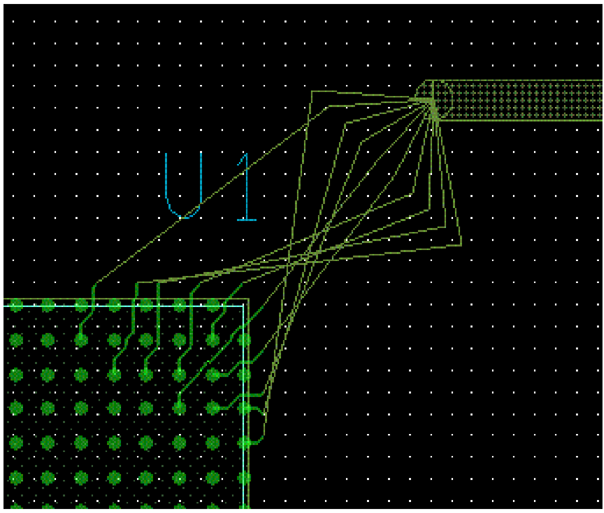
To perform breakout order optimization, perform the following steps:

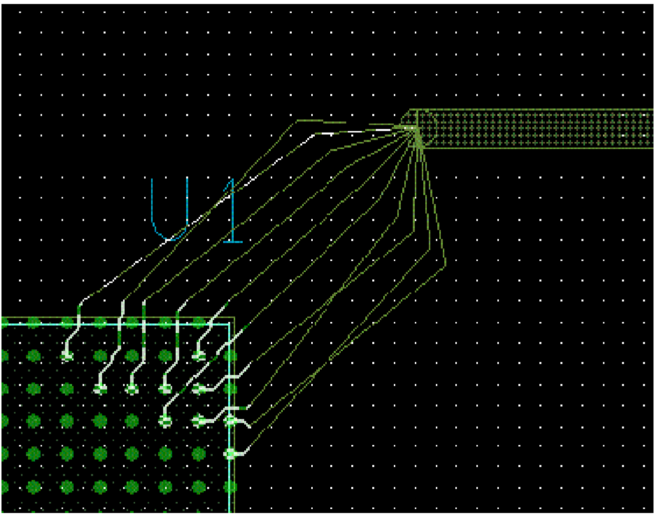
1. Click on the bundle with left mouse button.
2. Choose *Place – FPGA System Planner – Auto Pinswap.*

   The *FSP Auto Pinswap Option* dialog box appears.
3. Select *Breakout Order* option and click *OK.*

The breakout order optimization yields better results over the rake order optimization for cases where breakout is done on the BGA corner. See the following figure.

| **Rats before optimization** | **Rats after optimization** |
|---|---|



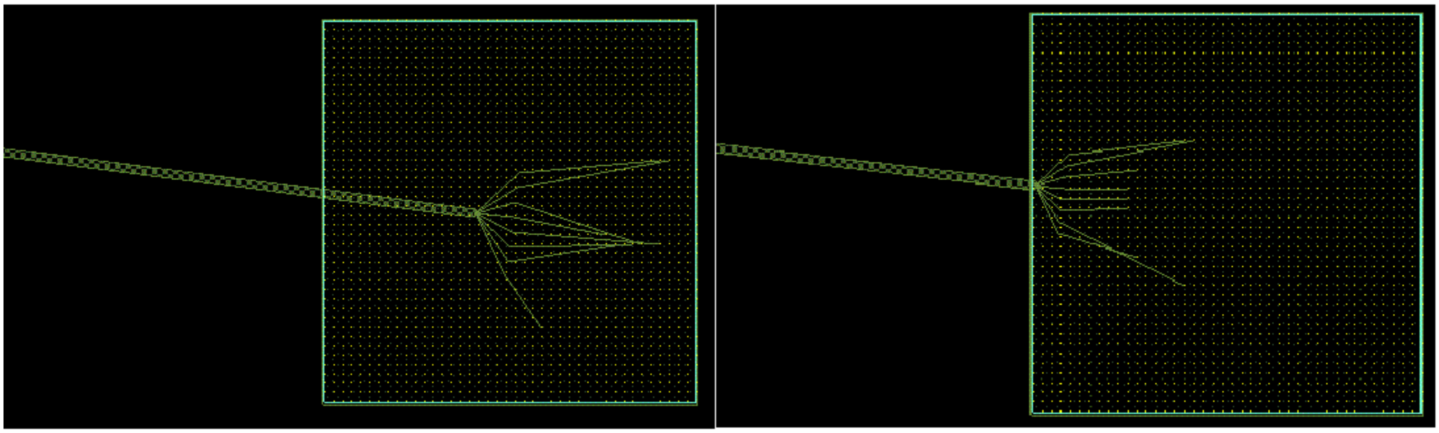


### *Re-assign Bundle Pins*

The Re-assign Bundle Pins command lets you reassign the bundle pins to a new set of pins that are neared to the bundle gather point. This is useful when you see breakout from the pin locations towards the bundle gather point occupies most of the BGA or board space as shown below.

**Rats assigned to Farthest Bank**

**Rats reassigned to Nearest Bank**



The Reassign Bundle Pins option in the *Auto Pinswap* dialog box lets you reassign the bundle pins to the banks that are close to the bundle gather point(s).

**Note:** In some cases, the reassignment of bundled pins is not guaranteed due to various reasons such as non availability of free banks near bundle gather points and conformity of bundle definitions based on FSP logical groups.

**Note:** Reassign bundle pins feature is aligned with the logical groups definition and design settings such as Use Banks setting specified in FSP. Before you use the feature it is important that you understand the methodology.

To perform reassign bundle optimization, perform the following steps:

1. Select on the bundle.
2. Choose *Place – FPGA System Planner – Auto Pinswap.*

   The *FSP Auto Pinswap Option* dialog box appears.
3. Select *Reassign Bundle Pins* option and click *OK.*

The bundle pins are moved to the bank that are near to bundle gather point.

**Note:** You can also use the *Auto Pin Swap* command to optimize the bundle nets that are assigned to multiple layers. In Allegro PCB Editor, you can assign individual nets that belongs to the same bundle to different layers. After you define the layers and assign nets to the layers, you can perform optimization using any of the options of the *Auto Pin Swap* command.

This section does not include a detailed description about defining layers and the tasks that are required to assign nets to the layers.

## Manual Pin Swap

The Manual Pin Swap command is useful, when major part of the design is completed and you wish to make final changes such as pin swaps and net moves with minimal volumes. The

FSP design running in the background in Allegro PCB Editor, provides an intuitive environment for manual pin swapping. It automatically recommends pins for you on the FPGA component to swap to reduce crossovers.

In the last phase, you set the ECO flag in the design. The FSP engine with an ECO flag, running in the background in Allegro PCB Editor allows restricted pin swaps. That indicates pin swaps that require an extra connection or removal of existing connection is ignored during swapping.

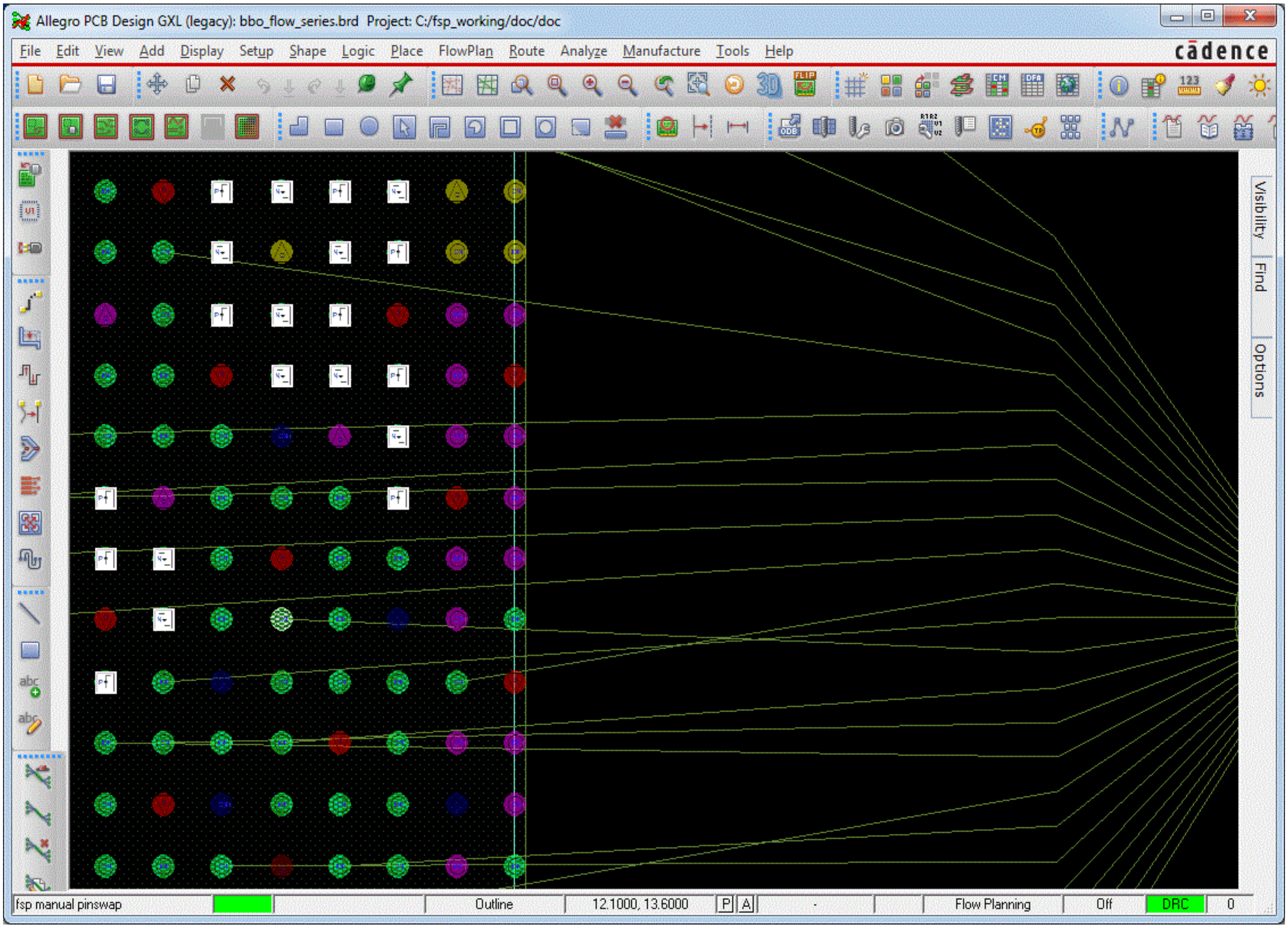
In this phase, since you have not set the ECO flag in the design, both the Auto Pin Swap and the Manual Pin Swap commands allows unrestricted pin swaps.

To perform manual pin swap, perform the following steps:

1. Select a pin on the FPGA component for which you want to perform swap and zoom to the selected pin.
2. Choose *Place – FPGA System Planner – Manual Pinswap.*

   Allegro PCB Editor displays the similar view of FSP canvas with all the FSP FPGA legends.
3. Right-click on the pin and choose *Show Swappable Pins* option.

   Allegro PCB Editor highlights the FPGA pins on the canvas that are available to be swapped with the current selected pin.

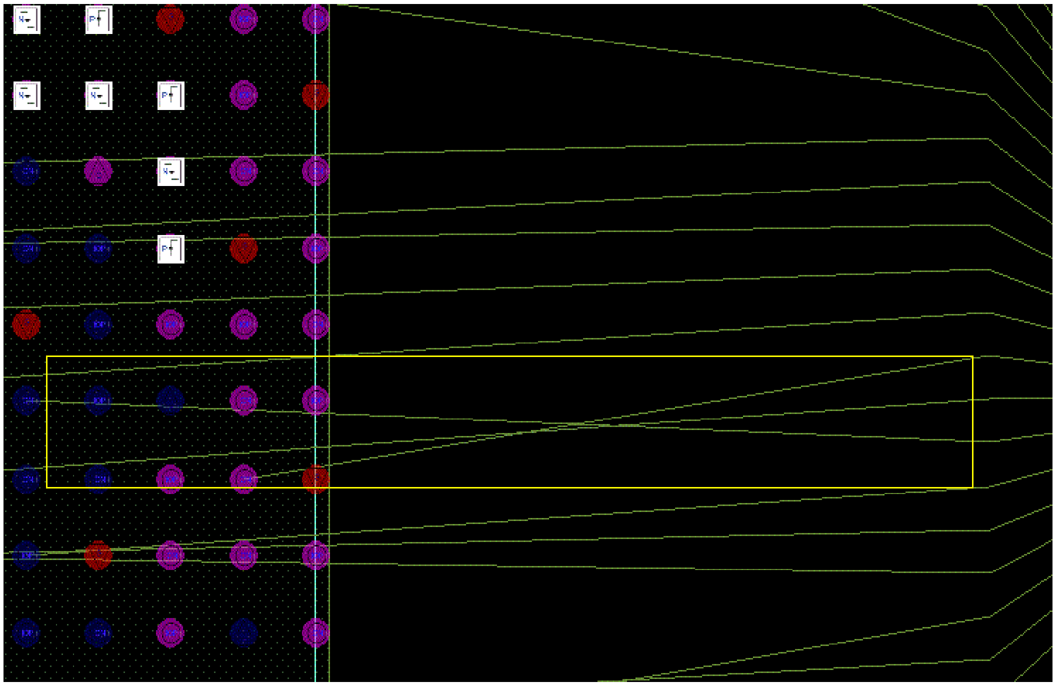


Allegro PCB Design GXL (legacy): bbo_flow_series.brd Project: C:/fsp_working/doc/doc
File Edit View Add Display Setup Shape Logic Place FlowPlan Route Analyze Manufacture Tools Help
cādence
Visibility
Find
Options
fsp manual pinswap
Outline
12.1000, 13.6000
P
A
Flow Planning
Off
DRC
0

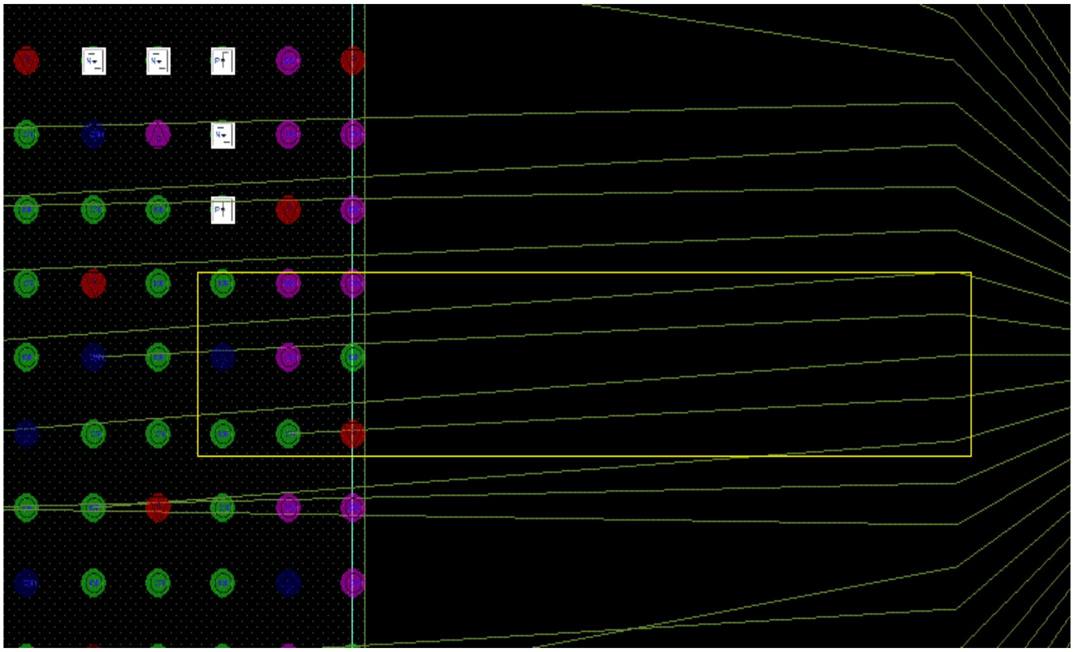
4. Click a pin from the highlighted pins that you want to swap with the selected pin.

**Note:** For differential pairs, quad signal, and other pin(s) belonging to the signal group are also highlighted and all the signals are moved accordingly when you select a destination pin.

### *Rats before optimization*



### *Rats after optimization*



The rats are redrawn and saved in the design database. You can continue to perform the same steps for other rats.

## Synchronizing Design between Allegro and FSP

After completing the optimization tasks, changes caused by the ECOs that are made in the board file such as reference designator, placement and pin swap changes have caused the FSP design and the Allegro database to go out of sync. You need to backannotate these changes from the board to the FSP design to bring them in sync.

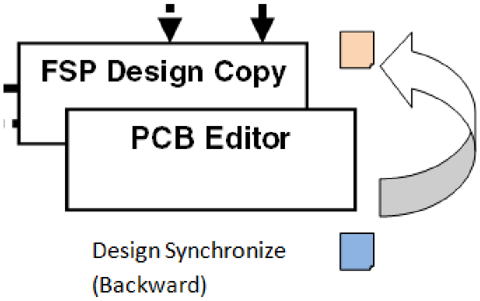
The following two changes occur during the transfer of design from Allegro PCB Editor to FSP:

1. Reference Designator

   You might make reference designator changes in Allegro PCB Editor.
2. Placement

   You might make placement changes in Allegro PCB Editor to facilitate routing.

The Allegro PCB Editor detects the placement and reference designator changes between the Allegro database and the FSP design and displays the results in the *FSP Synchronize* dialog box. The pin swaps and net schedules changes are never higlighted in the *FSP Synchronize* dialog box because these changes are saved instantly in the Allegro database.



To synchronize the Allegro database with the FSP design, perform the following steps:

1. Choose *Place – FPGA System Planner – Synchronize.*

   The FSP Synchronize dialog box appears.

   **Note:** The FSP Synchronize dialog box is not displayed, if no differences are found between the Allegro database and the FSP design.
2. Click *Details* to invoke *Design Compare* dialog box to view the differences in detail.
3. Click the *Select All* option.
4. Click *OK.*

The FSP design and the Allegro database are synchronized.

## Merging Changes with the FSP Design

**Note:** In this section, the *FSP design* term is referred to as the FSP design associated with FPGA System Planner and *FSP design copy* term is referred to as the FSP design associated with Allegro PCB Editor.

The changes you make in the FSP design copy may cause the FSP design and the FSP design copy to go out-of-sync. To merge the changes made in the FSP design copy with the FSP design, you the use Design Comparison dialog box in FSP.

The Design Comparison dialog box provides a sophisticated difference reporting and merging capabilities between FSP design and FSP design copy. You import the FSP design and the FSP design copy files in FSP’s Design Comparison dialog box to compare and generate a list of differences. The Design Comparison dialog box supports various controls to view, filter, and merge the differences. For more information on the various fields and options of the Design Comparison dialog box, see the Allegro FPGA System Planner User Guide.

The Design Comparison dialog box provides support for merging the following changes in FSP design:

- Pin swaps
- Placement changes
- NetGroups
- Reference designators

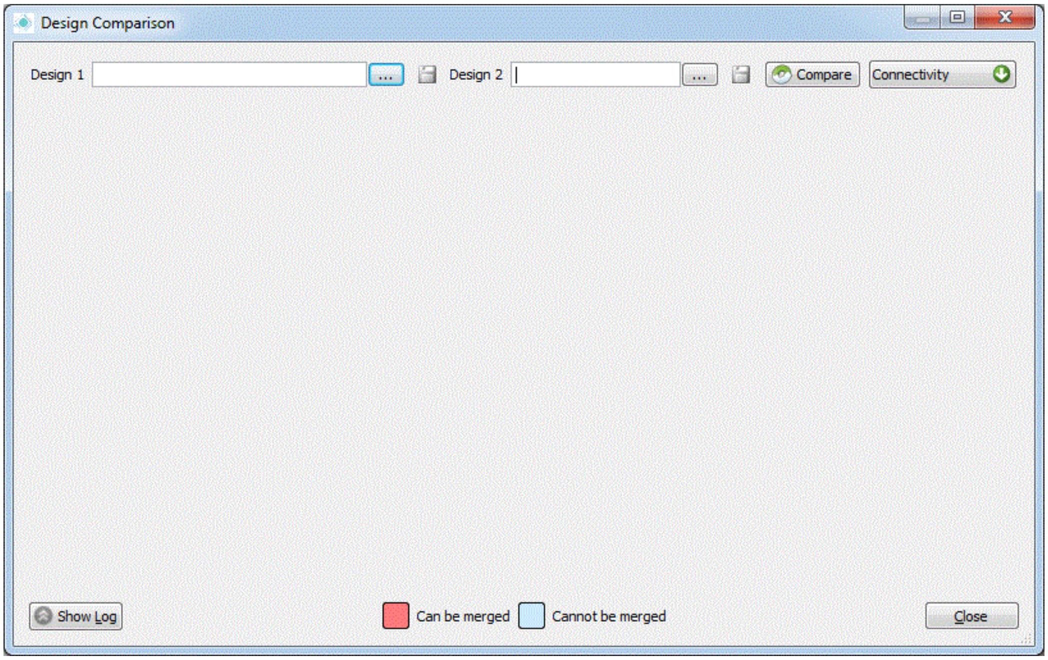
The Design Comparison dialog box does not provides support for merging the following changes in FSP design.

- Nets added or deleted
- Terminations
- Power regulators
- New component added or removed

To merge the changes, open the FSP design and perform the following steps:

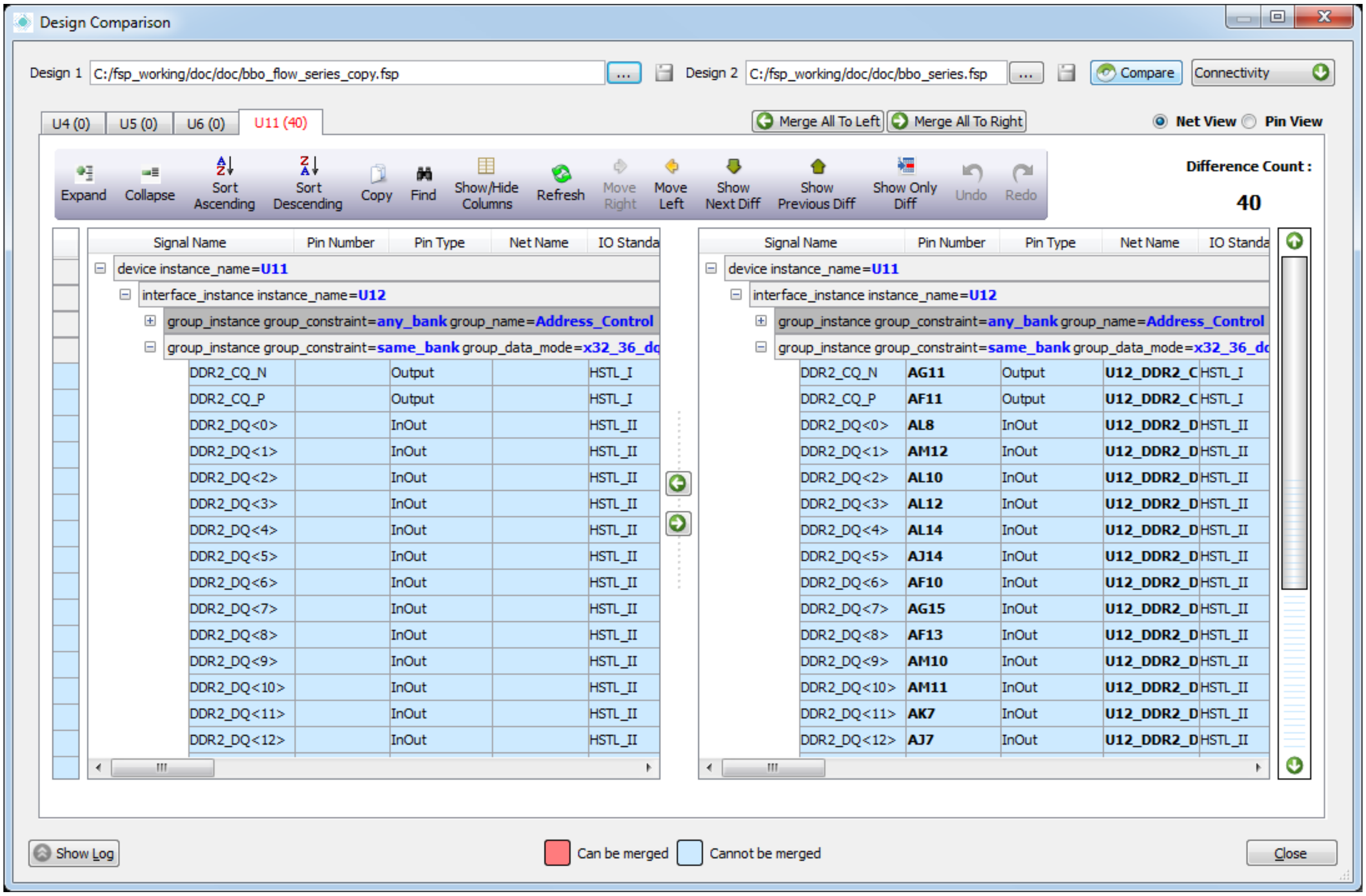
1. Choose *File – Design Compare* or click *Design Compare* icon from toolbar.

   The Design Comparison dialog box is displayed.



2. To select the first design,click the *browse (...)* next to the *Design1* text box and browse to the design file to import.
3. To select the second design, click the *browse (...)* next to the *Design2* text box and browse to the design file to import.
4. Click the *Design Compare* icon.

The differences between the two designs are displayed in the dialog box in two separate panes, one for each design. These differences can be filtered based on the category selection.



Design 1 C:/fsp_working/doc/doc/bbo_flow_series_copy.fsp

Design 2 C:/fsp_working/doc/doc/bbo_series.fsp

● Net View

Difference Count : 40

| Signal Name | Pin Number | Pin Type | Net Name | IO Standa |
|---|---|---|---|---|
| ⊟ device instance_name=**U11** | | | | |
| ⊟ interface_instance instance_name=**U12** | | | | |
| ⊞ group_instance group_constraint=**any_bank** group_name=**Address_Control** | | | | |
| ⊟ group_instance group_constraint=**same_bank** group_data_mode=**x32_36_dc** | | | | |
| DDR2_CQ_N | | Output | | HSTL_I |
| DDR2_CQ_P | | Output | | HSTL_I |
| DDR2_DQ<0> | | InOut | | HSTL_II |
| DDR2_DQ<1> | | InOut | | HSTL_II |
| DDR2_DQ<2> | | InOut | | HSTL_II |
| DDR2_DQ<3> | | InOut | | HSTL_II |
| DDR2_DQ<4> | | InOut | | HSTL_II |
| DDR2_DQ<5> | | InOut | | HSTL_II |
| DDR2_DQ<6> | | InOut | | HSTL_II |
| DDR2_DQ<7> | | InOut | | HSTL_II |
| DDR2_DQ<8> | | InOut | | HSTL_II |
| DDR2_DQ<9> | | InOut | | HSTL_II |
| DDR2_DQ<10> | | InOut | | HSTL_II |
| DDR2_DQ<11> | | InOut | | HSTL_II |
| DDR2_DQ<12> | | InOut | | HSTL_II |

| Signal Name | Pin Number | Pin Type | Net Name | IO Standa |
|---|---|---|---|---|
| ⊟ device instance_name=**U11** | | | | |
| ⊟ interface_instance instance_name=**U12** | | | | |
| ⊞ group_instance group_constraint=**any_bank** group_name=**Address_Control** | | | | |
| ⊟ group_instance group_constraint=**same_bank** group_data_mode=**x32_36_dc** | | | | |
| DDR2_CQ_N | **AG11** | Output | **U12_DDR2_C** | HSTL_I |
| DDR2_CQ_P | **AF11** | Output | **U12_DDR2_C** | HSTL_I |
| DDR2_DQ<0> | **AL8** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<1> | **AM12** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<2> | **AL10** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<3> | **AL12** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<4> | **AL14** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<5> | **AJ14** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<6> | **AF10** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<7> | **AG15** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<8> | **AF13** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<9> | **AM10** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<10> | **AM11** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<11> | **AK7** | InOut | **U12_DDR2_D** | HSTL_II |
| DDR2_DQ<12> | **AJ7** | InOut | **U12_DDR2_D** | HSTL_II |

5. Select a category from drop-down list next to the *Design Compare* icon.

   The differences are displayed based on the selected category. For example, select *Net Groups* to display net differences in the pane.
6. Click -> to merge the nets from the right side pane to the left side pane or click <- to merge the nets from the left side pane to the right side pane.
7. Click the *Merge All to Left* icon to merge all the changes from the right pane to the left pane and click the *Merge All to Right* icon to merge the changes from left to right.
8. Click the *Save* icon next to *Design 1* browse (...) button to save the changes in the FSP design file.
9. Click the *Save* icon next to *Design 2* browse (...) button to save the changes in the FSP design copy file.
10. Click *Close* to exit the form.

The changes made in the FSP design copy are merged with the FSP design. You can view the changes in the FSP design.

## Regenerating Symbols and Schematics

Once the changes are merged in the FSP design, you can regenerate or update the schematics to bring the FSP design and the schematic in sync.

At this phase, you can continue to make ECO and non-ECO changes in the FSP design. After you have made the changes in the design, you can regenerate the symbols and the schematics. Symbols and schematics can be generated multiple times to bring the schematic and logical design in sync.

To regenerate the schematics, perform the following steps:

1. Choose *Generate – Schematics.*

   The Generate Allegro DE-HDL Schematics dialog box is displayed.
2. Specify the options required for generating the schematics in the Generate Allegro DE-HDL Schematics dialog box.
3. Click *OK.*

The Message window displays the successful creation message and the path of the directory where the schematic files are generated.

*Important*

The steps explained in the sections above, can be performed multiple times till the major changes in the design are accomplished.

## Setting up ECO Mode

Once the major changes are accomplished, you can now set the ECO flag in the design. The ECO flag is set in the last phase of the design cycle. During last phase of the design cycle, if you plan to take a backup of the design, then you must set the flag first before taking the backup of your design. You set the ECO flag in the Project tab of the *Settings* dialog box.

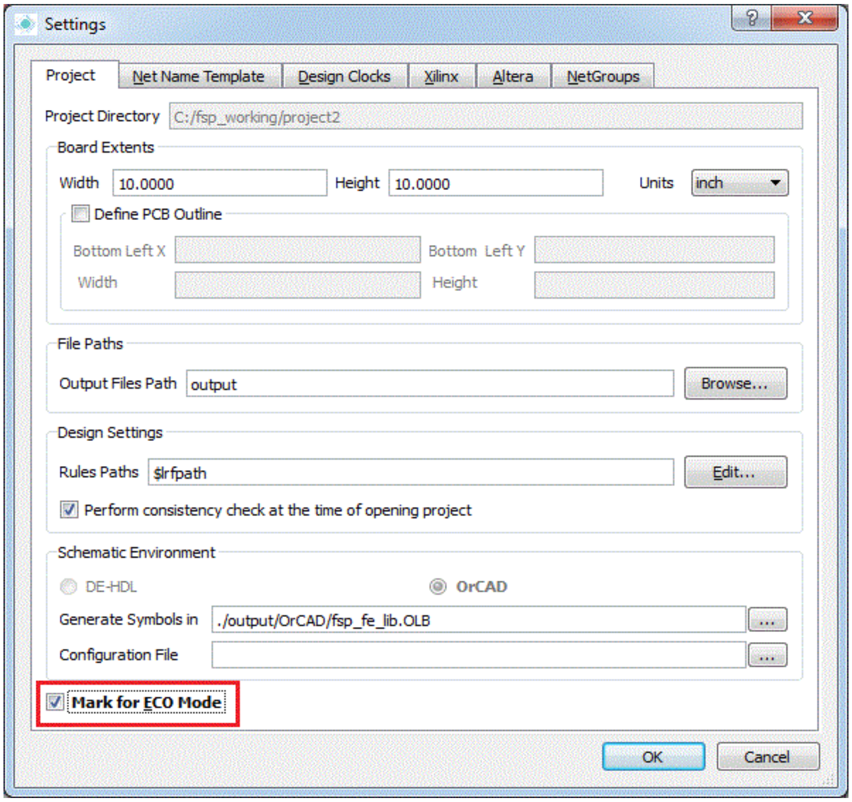
*Important*

The design with ECO flag, allows the Auto Pin Swap and the Manual Pin Swap commands to perform restricted pin swaps in Allegro PCB Editor.

To set the ECO flag:

1. Choose *File – Settings.*

   The Settings dialog box is displayed.
2. Select *Mark for ECO Mode* to set the ECO flag.



Project Directory: C:/fsp_working/project2

Board Extents

Width: 10.0000 Height: 10.0000 Units: inch

□ Define PCB Outline

Bottom Left X: Bottom Left Y:

Width: Height:

File Paths

Output Files Path: output

Design Settings

Rules Paths: $lrfpath

☑ Perform consistency check at the time of opening project

Schematic Environment

○ DE-HDL ● OrCAD

Generate Symbols in: ./output/OrCAD/fsp_fe_lib.OLB

Configuration File:

☑ Mark for ECO Mode

3. Click *OK.*

# Selecting a Lower Product Options

To perform ECO changes in the design, you can select a lower product options. Selecting a lower product option, enables the optimization commands to perform restricted pin swaps in Allegro PCB Editor.

To select a lower product, perform the following steps:

1. Choose *File – Close,* to close the project.
2. Choose *File – Change Product.*

   The *Cadence Product Choice - <Release Name>* dialog box is displayed.
3. Select a lower product among the last three options.
4. Select the *Use As Default* option.
5. Click *OK* of the *Cadence Product Choice - <Release Name> dialog box.*
6. Choose *File – Open,* to open the project.

Before synchronizing the FSP design with the Allegro board, you can perform ECO changes in FSP or in schematic, based on your preferrences.

## Synchronizing Design between FSP and Allegro

To synchronize the FSP design with the Allegro board file, enter *Allegro* in the *Run* command window to launch Allegro PCB Editor.

**Note:** You do not need to load the design in Allegro PCB Editor at this stage, since you have already performed at the beginning phase of the design cycle.

After Allegro PCB Editor is invoked, the *FSP Synchronize* dialog box is automatically displayed, if any differences are found between the FSP design and the Allegro board file. The categories listed in the *FSP Synchronize* dialog box are highlighted by default depending on the changes made in FSP and in schematic tool.

**Note:** Besides the listed changes, any other changes such as adding new components, adding or deleting existing nets, terminations, reference designators made in FSP or in schematic tool, are ignored during design synchronization.

For more information about the steps to synchronize the FSP design with the Allegro board, see the Synchronizing Design between FSP and Allegro section.

## Swapping FPGA Pins in PCB Editor

The steps to perform for displaying bundles, activating IFP mode, and performing different types of optimization are explained in the Swapping FPGA Pins in PCB Editor section. With only one exception, the Auto Pin Swap and the Manual Pin Swap commands perform restricted pin swaps. That indicates pin swaps that require an extra connection or removal of existing connection is ignored during swapping.

## Synchronizing Design between Allegro and FSP

The steps to synchronize the Allegro board with the FSP design are explained in the Synchronizing Design between Allegro and FSP section.

## Merging Changes with the FSP Design

The steps to merge the changes made in the FSP design copy with the FSP design are explained in the Merging Changes with the FSP Design section.

# Regenerating Symbols and Schematics

In the last phase of the design cycle, it is not required to regenerate schematic. The existing back annotation process between FSP, DE-HDL, and Allegro PCB Editor is sufficient to keep the schematic design in sync.

# Index

## Symbols

[] in syntax 2
{} in syntax 2
| in syntax 2

## B

braces in syntax 2
brackets in syntax 2

## C

cds.lib file 23
Components
    adding device 7, 10
    adding real components 4, 6
    adding rules file or virtual interface 4, 7
conventions
    user-defined arguments 2
    user-entered text 2

## E

environment variable
    FSP_CONFIG_FILE 2

## F

files
    placement.xml 23, 25
FPGA System Planner
    overview 2
FPGAs
    in FSP 3
    problem scenario 1
FSP
    flow 1
    front and back flow 2
    overview 2
    tasks in the flow
        in DEHDL 4

    - in FSP 3
    - in PCB Editor 4
- FSP methodology flow
  - hierarchical method 3, 24
    - ECOs 4
    - integrating with PCB design 4
    - pros and cons 5
- FSP_CONFIG_FILE 2

## I

- integration flow
  - auto pin swap 23
    - breakout order 25
    - rake order 24
    - reassign bundle pins 26
  - creating design copy 14
  - design comparison 32
  - ECO flag 35
  - license set up 1
  - loading design 17
  - manual pin swap 27
  - netgroups 4
  - synchronizing design
    - Allegro and FSP 31
    - FSP and Allegro 17
  - the beginining phase 3
  - the last phase 5
- integration flow-about 1
- italics in syntax 2

## K

- keywords 2

## L

- literal characters 2
- local library 23

## O

- or-bars in syntax 2

## T

- Terminations
  - defining 13

## V

vertical bars in syntax 2