Allegro 16.0 Constraints: Effective Use and 15.x Migration Tips

John Schiavone
Cadence Design Systems, Inc.
270 Billerica Road
Chelmsford, MA 10824
1-978-262-6319
johnsc@cadence.com

ABSTRACT

The Allegro 16.0 release contains the most extensive changes to Physical and Spacing (P&S) Constraints since version 6.0. The P&S constraint system has a well-defined hierarchy and introduces the NetClass object and extends existing Electrical constraint objects (Bus, DiffPair, and PinPair) to P&S constraints. This release also introduces layer-based constraint regions. This new system improves the user's ability to manage a design's constraints by eliminating unnecessary assignments and by employing a concept of constraint by exception. At the highest level (the design) the baseline constraints are established. As you progress down the hierarchy, constraints that are exceptions to the baseline values are specified through overrides.

In this paper, readers will learn how to effectively use the P&S constraints system. The paper will present the constraint system hierarchy. The system objects and their relationships will be briefly discussed. The major portion of the paper will be used to show how a real-world design can be constrained in Allegro 16.0. This customer design was created using Allegro 15.x. The constraints will be re-applied to take advantage of the v16.0 hierarchical system.

1. INTRODUCTION

The Allegro platform has a collection of constraints that the designer can use to control all aspects of the design, including electrical, physical, spacing, and manufacturing parameters. Electrical constraints are usually determined during the circuit design process and passed to the physical design process. Physical, spacing and manufacturing constraints are imposed during the physical design.

A constraint is a design requirement which is validated using a Design Rule Check (DRC). The designer sets the value of the constraints and applies them to the design objects through a hierarchical system of constraint objects. Each constraint has an associated DRC; this constraint-DRC association is built into Allegro.

A constraint belongs to one of four domains, based on the nature of the requirement. Constraints in the Electrical, Physical, and Spacing domains are applied to the design using the hierarchical constraint system. Constraints in the Design domain are global; one requirement (value) exists for the entire design.

The Electrical, Physical and Spacing domains support constraint sets — a named, reusable collection of constraint values. Either constraint sets or individual constraint values may be applied to constraint objects.

This paper will focus exclusively on Physical and Spacing constraints.

2. ALLEGRO CONSTRAINT SYSTEM

The Allegro platform supports a constraint hierarchy, which allows the designer to manage constraints on the appropriate objects in the design. Table I details the hierarchical constraint system. This table represents all system objects applicable to the Physical and Spacing Domains. Constraint information specified on an object is inherited by all subordinate objects in the hierarchy. For example, a constraint defined on the Design will be inherited by all objects in the design. If an object needs a different constraint value (either a tighter or looser) from its inherited constraint value, you can set an override on that object and its constraint value will take precedence.

Table I - Constraint System Object Hierarchy

Physical Domain	Spacing Domain
Design	Design
NetClass	NetClass
Bus	Bus
Differential Pair	Differential Pair
Xnet	Xnet
Net	Net
Pin Pair	Pin Pair
---	Class-Class
Region	Region
Region-Class	Region-Class
---	Region-Class-Class

NOTE: The term “Class” in Class-Class, Region-Class, and Region-Class-Class refers to a NetClass object. Class is substituted for convenience, only.

2.1 TERMS and DEFINITIONS

Constraint Sets: The Electrical, Physical and Spacing domains support Constraint Sets (CSets). A CSet is a named, reusable collection of constraint values. CSets are not supported in the Design domain.

Physical and Spacing CSets: A Physical CSet consists of one value per layer for each physical constraint. A Spacing CSet consists of one value per layer for each spacing constraint. In all designs, one Physical- and one Spacing-CSet (named “DEFAULT”) are provided, which you

cannot delete or rename; its values are predefined, but you can modify them to suit your design requirements.

Constraint Override: A constraint override is an individual constraint value applied to a member of a constraint object that overrides the inherited constraint value. A constraint override has higher precedence than a CSet.

2.2 Constraint System Objects

The Design object is constrained by a physical constraint set and a spacing constraint set in those respective domains. In all designs, the Allegro Physical Editor provides one physical constraint set and one spacing constraint set. These sets are named “DEFAULT” and each constraint is set to a pre-defined value. You may set the value of any constraint within a physical or spacing constraint set, but you cannot remove or add constraints to the constraint set. The “DEFAULT” constraint sets cannot be deleted. You may also reference the Design object to another constraint set. Constraint overrides are not permitted on the Design object in the Physical and Spacing domains.

The NetClass, Bus, Differential Pair, and Xnet objects are similar in that they are all collections of Nets. The Bus, Differential Pair and Xnet objects existed prior to 16.0 in the Electrical domain. Their application has been extended to the Physical and Spacing domains. The NetClass is a new constraint system object for 16.0. It can contain any Net-based object (Net, Xnet, Differential Pair, and Bus) as a direct member. Constraints referenced through these objects are inherited by their members.

The Pin Pair object existed prior to 16.0 in the Electrical Domain. Its application has been extended to the Physical and Spacing Domains for 16.0. Due to the inherent ambiguity of implicit Pin Pairs, constraining a Pin Pair with P&S constraints works best when the Pin Pair is explicitly created, and the portion of the net between the two pins of the pair are user-scheduled.

The Region object is a collection of Constraint Region class shapes. In 16.0, a new Allegro class is defined: the Constraint Region class. The subclasses of the Constraint Region class include Through All, Outer Layers, Inner Signal Layers, Inner Plane Layers, and all the named layers of the stack up. Physical design objects (pins, vias, clines, shapes, etc.) inherit the constraints of a Region by being located within the boundary of a Constraint Region shape.

The remaining objects are used to manage the relationship between constraint system objects. These are the Class-Class, Region-Class, and Region-Class-Class objects.

3. DESIGN MIGRATION FROM ALLEGRO 15.X.

Many readers may be familiar with the constraint system from 15.x. This section will help them understand how the previous system data is translated to the new system.

The 15.x constraint system was driven by two fundamental properties: the Net_Physical_Type and the Net_Spacing_Type. These properties were attached to nets with name values, which were used to identify the type, or classification, of the net. These properties were also attached to shapes that were drawn on the Board Geometry / Constraint Area subclass, thereby declaring the type of the constraint area. Two tables existed in each design: the Physical Assignment Table and the Spacing Assignment Table. The Physical Assignment Table was automatically populated with all the possible combinations of the Net_Physical_Type properties attached the nets and constraint areas in a format of Net_Type – Area Type. The Spacing Assignment Table was also automatically populated with all possible combinations in the format of Net Type – Net Type – Area Type. The last entry in each row of the tables contained a constraint set name. This entry is initially set to "Default" and can be user-modified.

When a 15.x design is opened in 16.0, the P&S constraints are automatically revised to the new system. NetClass objects are created from Net_Physical_Type and Net_Spacing_Type values attached to nets and the nets are added to the NetClass. Regions are created from the Net_Physical_Type and Net_Spacing_Type values attached to shapes and the shapes are added to the Region. Class-Class, Region-Class, and Region-Class-Class objects are subsequently created per the 15.x Assignment Tables. The new 16.0 objects reference the same constraint sets that were assigned to their 15.x equivalent row in the Assignment Table. See Tables II and III for a summary of this translation from 15.x to 16.0.

Table II – Physical Constraints

16.0 Constraint Object	15.x Equivalent
Design	No_Type - No_Type
Net Class	Net_Type - No_Type
Bus	(none)
Differential Pair	(none)
Xnet	(none)
Net	Property on a net
Pin Pair	(none)
Region	No_Type - Area_Type
Region-Class	Net_Type - Area_Type

Table III – Spacing Constraints

16.0 Constraint Object	15.x Equivalent
Design	No_Type - No_Type - No_Type
Net Class	Net_Type - No_Type - No_Type
Bus	(none)
Differential Pair	(none)
Xnet	(none)
Net	(none)
Pin Pair	(none)
Class-Class	Net_Type - Net_Type - No_Type
Region	No_Type - No_Type - Area_Type
Region-Class	Net_Type - No_Type - Area_Type
Region-Class-Class	Net_Type - Net_Type - Area_Type

4. CONSTRAINING A DESIGN EFFECTIVELY

Constraining a design effectively means applying constraints in a way that enforces the design intent and yet is still obvious and manageable. As you move from the top to the bottom of the constraint hierarchy in Table I, fewer constraint overrides should be needed.

As mentioned previously, all new designs have at least one Physical constraint set and one Spacing constraint set; both are name “Default”. When a new design is started, the “Default” constraint sets are populated with default values. These values are predetermined and coded into Allegro. For a board with design units of mils, these values are mostly 5, which are based on a design technology of “5 mil lines, 5 mil spaces”. This is a starting point and probably not optimal for any specific design or manufacturing technology, especially when you consider that these values are initially the same for all layers and between different pin and via types (thru pins and vias, SMD pins, test points, etc). The Default constraint set should be used to establish the baseline constraints. The other constraint system objects in Table I should be used to apply exceptions to those baseline constraints.

For this section we’ll look at how a customer’s design has been constrained in version 15 and how the constraints could be applied differently and more effectively in version 16.

This test case is fully placed and routed with a few DRC errors remaining. The design contained nearly 5000 packages and 7000 nets. There are 24 layers. There are 15 Physical constraint sets and 24 Spacing constraint sets. There are 121 shapes on the Constraint Region / All subclass that are managed by 10 Constraint Regions. (If you are familiar with v15 terminology, this would be 121 shapes on the Board Geometry / Constraint Area subclass with a total of 10 unique combinations of the Net_Physical_Type and Net_Spacing_Type properties attached to those shapes.)

Allegro v15 required that all possible combinations of the Net_<x>_Type properties attached to shapes and nets be referenced to a constraint set. In the Spacing domain alone, the sum of the Region, Region-Class, Region-Class-Class and Class-Class constraint objects is greater than 900. Each one references a constraint set. In many cases, the referenced constraint set is “Default” which is the same constraint set referenced by the Design and in many other cases, the Region-Class and Region-Class-Class objects reference the same constraint sets as the Region. Both conditions add complexity without benefit.

Of the 121 Constraint Region shapes in the design, only 37 appear to be specifically associated with a package symbol. The rest seem to be used to suppress DRC errors. We will delete these shapes and keep the ones associated with package symbols. These package symbols are BGAs, QFPs and connectors. These kinds of packages typically require exceptions to the Default constraints to complete the routing under and around the package. It is recommended that the constraint region shape be included in the symbol drawing. In the symbol drawing, the Region_Name property should be attached to the shape. When the symbol is placed on the board, a Constraint Region will be created with the value of the Region_Name property and the shape will be added to that Constraint Region. Successive placement of the same symbol will add the new shapes to the existing Constraint Region. Here, the symbol drawings were changed

to include the constraint region shape and the Region_Name property was set with a descriptive name such as “BGA_256”.

To start the process of re-constraining this design, we’ll start at the bottom of the Table I hierarchy. All the existing Region, Region-Class, Region-Class-Class and Class-Class objects are deleted in Constraint Manager. All of the shapes on the Constraint Region subclasses are deleted. Next the package symbols are refreshed, which creates new Constraint Regions. There are 16 Constraint Regions that manage the constraints of 37 shapes. At this point, however, these new Constraint Regions are unconstrained. It’s not known, yet, if any constraints are necessary.

In version 16, constraints must be explicitly applied; therefore, if a Constraint Region is unconstrained, it’s invisible to the DRC system. This applies to all constraint system objects and applies on a constraint by constraint basis. If a Region (or NetClass, etc.) has a Line-to-Line spacing constraint attached to it, the Region is still invisible for all other spacing constraints.

When the “drc update” command is run at this point, not unexpectedly, many DRC errors show up within the boundaries of the Constraint Region shapes. Nearly all of these are spacing violations involving SMD pins and Thru Vias. The Default constraint between thru vias is 15 mils (all layers) so a Thru Via to Thru Via override of 12 mils is added to about half of the regions. The Default constraint between SMD Pins and Thru Vias is 8 mils. An SMD Pin to Thru Via override of 5 mils is added to nearly all of the regions.

There are also many Line to Line DRC errors. The Default Line to Line spacing is 5 mils on the Top and Bottom and 4 mils on all internal layers. Adding a 4 mil overriding Line to Line constraint to five Constraint Regions restores the design to compliance. A few spacing DRC errors remained. Two of the regions needed an exception to the SMD Pin to Thru Pin constraint and one region needed an exception to the SMD Pin to Line constraint.

The design has now been brought back into compliance with respect to the spacing constraints. This was accomplished without any Class-Class, Region-Class, nor Region-Class-Class constraints. Also, unreferenced Spacing Constraint Sets are deleted which reduces the number of constraint sets from 24 to 8.

The shapes of all the Constraint Regions are on the Constraint Region / Through All subclass. This is still necessary since we needed to allow exceptions to the Default Thru Pin and Thru Via constraints. If, however, the Default constraints could be reduced to match the exceptions, then the Constraint Region shapes could be relocated to the Top, Bottom, or Outer Layers subclass. Some Constraint Regions (and their shapes) could be eliminated altogether since now the constraint overrides on these regions are identical to the Default constraints. A lower quantity of Constraint Regions shapes will reduce the time to perform DRC checking.

Figures 1 thru 3 show the view in Constraint Manager of the Regions’ Spacing constraints.

Figures 1-3 : Constraint Manager view of Spacing Constraints on Regions. Unconstrained columns have been hidden for easier reading.

1	Type	Objects	Referenced Spacing CSet	Line To		
				Line	SMD Pin	Shape
2				mil	mil	mil
3	Dsn	⊟ cns_v16_l	DEFAULT	5.00:4.0...	5.00	7.00:5.0...
4	Rgn	CONN_200				
5	Rgn	BGA_144				
6	Rgn	BGA_165		4.00		8.00
7	Rgn	BGA_256_1				
8	Rgn	BGA_256_2				
9	Rgn	BGA_364			4.00	
10	Rgn	BGA_676				
11	Rgn	BGA_860				
12	Rgn	⊞ BGA_1096		4.00		
13	Rgn	BGA_1152		4.00		
14	Rgn	⊞ BGA_1356		4.00		
15	Rgn	BGA_1517				
16	Rgn	LED				
17	Rgn	SMD2VIA				
18	Rgn	QFP_128		4.00		
19	Rgn	⊞ QFP_176				

Figure 1 - Line Spacing Constraints on Regions

1	Type	Objects	Referenced Spacing CSet	SMD Pin To				
				Line	Thru Pin	SMD Pin	Thru Via	Test Via
2				mil	mil	mil	mil	mil
3	Dsn	⊟ cns_v16_l	DEFAULT	5.00	14.00	7.00	8.00:8.0...	15.00:15...
4	Rgn	BGA_144					5.00	
5	Rgn	BGA_165						
6	Rgn	BGA_256_1					5.00	
7	Rgn	BGA_256_2					5.00	
8	Rgn	BGA_364		4.00		6.00	5.00	
9	Rgn	BGA_676				6.00	5.00	
10	Rgn	BGA_860					5.00	
11	Rgn	⊞ BGA_1096					5.00	
12	Rgn	BGA_1152					5.00	12.00
13	Rgn	⊞ BGA_1356					5.00	
14	Rgn	BGA_1517			2.50	6.00	2.50	
15	Rgn	CONN_200					5.00	
16	Rgn	LED			8.00			
17	Rgn	QFP_128						
18	Rgn	⊞ QFP_176						
19	Rgn	SMD2VIA					5.00	

Figure 2 – Pin Spacing Constraints on Regions

	Type	Objects	Referenced Spacing CSet	Thru Via To	
1				SMD Pin	Thru Via
2				mil	mil
3	Dsn	⊟ cns_v16_l	DEFAULT	8.00:8.0...	15.00
4	Rgn	BGA_144		5.00	12.00
5	Rgn	BGA_165			
6	Rgn	BGA_256_1		5.00	12.00
7	Rgn	BGA_256_2		5.00	
8	Rgn	BGA_364		5.00	12.00
9	Rgn	BGA_676		5.00	12.00
10	Rgn	BGA_860		5.00	
11	Rgn	⊞ BGA_1096		5.00	
12	Rgn	BGA_1152		5.00	
13	Rgn	⊞ BGA_1356		5.00	
14	Rgn	BGA_1517		2.50	12.00
15	Rgn	CONN_200		5.00	12.00
16	Rgn	LED			
17	Rgn	QFP_128			
18	Rgn	⊞ QFP_176			12.00
19	Rgn	SMD2VIA		5.00	

Figure 3 – Via Spacing Constraints on Regions

Since the uprev'd Regions and Region-Classes were deleted, there are a number of Physical DRC errors that need to be resolved.

There may be a number of NetClasses in the Physical domain that have only one member net. This is a result of the uprev process which is designed to maintain the v15 constraint resolution. In v15, constraint overrides existed for many of the physical constraints. However, in v15 these overrides had precedence over the constraint area. In order to maintain the constraint resolution, a net with an override was placed into a NetClass by itself, and a Region-Class was created between that NetClass and each Region. The name of the NetClass was formulated from the net's Net_Physical_Type and the net name. Although this process maintains the v15 constraint resolution, the result is a more complex constraint application and is not recommended for new work.

Since the Regions and Region-Classes were deleted, these single-net NetClasses are no longer necessary. The nets should be moved back into their original NetClasses. You can identify the original NetClass from the current NetClass name or the name of the referenced Physical Constraint Set. The override constraint should remain on the net.

Three Region-Classes are needed to bring the design into DRC compliance with respect to the Physical domain. The Min Line Width constraint is overridden for these three Region-Classes. Notice that Regions themselves are unconstrained in the Physical domain. Also, unreferenced Physical Constraint Sets are deleted which reduces the number of sets from 15 to 10.

Figure 4 shows the view in Constraint Manager of the Regions' Physical constraints.

1	Type	Objects	Referenced Physical CSet	Line Width	
				Min	Max
2				mil	mil
3	Dsn	⊟ cns_v16_l	DEFAULT	5.00:4.00:4.00...	0.00
4	Rgn	CONN_200			
5	Rgn	BGA_144			
6	Rgn	BGA_165			
7	Rgn	BGA_256_1			
8	Rgn	BGA_256_2			
9	Rgn	BGA_364			
10	Rgn	BGA_676			
11	Rgn	BGA_860			
12	Rgn	⊟ BGA_1096			
13	RCls	POWER		10.00	
14	Rgn	BGA_1152			
15	Rgn	⊟ BGA_1356			
16	RCls	RAM		10.00	
17	Rgn	BGA_1517			
18	Rgn	LED			
19	Rgn	SMD2VIA			
20	Rgn	QFP_128			
21	Rgn	⊟ QFP_176			
22	RCls	DIFFPAIRS		4.00	

Figure 4 – Min Line Width Constraints on Region-Classes

5. TECHNOLOGY FILES

One purpose of a technology file is to capture and/or specify the design independent constraints. The technology file supports Constraint Set, NetClass, Region, Class-Class, Region-Class, and Region-Class-Class entries. The constraints of NetClasses and Regions can be specified in technology file, but these objects will not have any members (nets and shapes, respectively). These member objects are dependent on the logical and physical part of the design. Constraints applied to nets, xnets, differential pairs, and buses are dependent on the specific design and are not captured in a technology file. This is an important point to consider when promoting a consistent constraint process within a design group.

6. CONCLUSION

This paper explained Allegro's 16.0 constraint system hierarchy. This system improves the user's ability to manage a design's constraints by eliminating unnecessary assignments and by employing a concept of constraint by exception. Techniques for constraining a design effectively were discussed in the context of reapplying the constraint to a design completed with Allegro 15.x. At the highest level (the design) the baseline constraints are established. Constraints that are exceptions to the baseline values are specified through overrides on lower constraint system objects. The results were that over 900 Class-Class, Region, Region-Class, and Region-Class-Class constraints were replaced with 16 Regions and 3 Region-Class constraints without affecting the design's DRC compliance. Additionally, 21 constraint sets were no longer necessary. It is possible that further consolidation within the constraint sets themselves may be achieved; however, this was not investigated for this paper.