2 Checking Design Rules

Docian Bula Sate

	Design Fluid Octo	3-12
•	Design Rule Types	3-17
•	Dracula® Rule Set to L-Edit Design Rule Conversion	n 3-30
	Checking the Layout	3-62
•	Correcting Errors	3-75
•	Optimizing Performance	3-81

2 12

Design rules, in their simplest form, are usually minimum allowable values for certain *widths*, *separations*, *extensions*, and *overlaps* of and between geometrical objects. The exact nature of design rules is dependent on specifications supplied by the foundry to which the design will be submitted for fabrication.

To check a layout for design rule violations, then, involves two basic steps:

- Defining the rules that are acceptable for your design.
- Running the design rule checker on the entire design or a portion of it.

The commands **Tools > DRC** (for whole cells) and **Tools > DRC Box** (for limited regions) run a *design rule checker*, which determines whether a design obeys a specific set of rules. Design rule violations can be reported directly on the layout, in a text file, or both.

L-Edit can check only boxes and $45^{\circ}/90^{\circ}$ polygons and wires for design rule violation. It cannot check circles or all-angle polygons and wires.

Design Rule Sets

It is not usually necessary to create sets of design rules from scratch. If you have a previous design file that uses a set of rules similar to those you want to employ in your current design, you can modify the rule set from the previous design.

In general, you must perform the following three steps to create or edit a design rule set:

- [1] Determine which rules must be specified. Fabrication services are typically able to provide design rule sets.
- [2] Determine which generated layers, if any, will be needed to implement each rule in the set. Define these layers using **Setup > Layers** (see Layer Setup on page 1-155).
- [3] Enter the rules in the **Setup Design Rules** dialog—see Specifying Design Rules on page 3-69.

Setups

The design rule set is part of the setup specification that characterizes every L-Edit design. This setup should be established before you start any new design. There are several ways of doing this, described below.

Copying Setup Information to a New File

There are three ways to copy setup information to a new file:

- Copy a design file. The copy will automatically contain the same setup information as the original design file.
- Create a new file while a file with the desired setup information is active. The new file will automatically contain the same setup information.
- Import from a TTX or TDB file. Use the command File > Replace Setup
 to copy the setup information from the specified file to the active file. See
 Replacing the Setup on page 1-82.

If necessary, you can modify design rules in the new or copied file using the **Setup Design Rules** dialog (see Specifying Design Rules on page 3-69).

Combining Rules from Different Files

You can combine design rules from TDB-format or TTX-format files as follows:

- Start by opening the design file into which you want to introduce additional rules. In that file, use the Setup Design Rules dialog to delete the design rules that you do not want to keep.
- Create any additional needed layers, including generated layers, and remove any unused layers. You should be careful in this step to create all and only

the required layers. Design rules associated with missing layers will not work properly. No design rules will be specified for extra layers, so layout errors on those layers will not be detected.

Specify the TDB or TTX file from which existing rules are to be taken in the Replace Setup Information dialog (see Replacing the Setup on page 1-82). Enter the name of the source file, uncheck all the options except DRC rules, and click OK. L-Edit reads the specified setup, including design rules, into the active file.

Generated Layers

You can specify design rules for generated layers just as you would for other layers. When you use generated layers in the specification of a design rule, L-Edit/ DRC automatically generates objects on those layers then deletes these objects when the check is complete.

Following the DRC run, L-Edit automatically deletes objects on generated layers that were created during the DRC run. If enable derivation is off, then DRC doesn't generate or delete the derived layer.]

Rule Lists

To import a set of design rules into an existing L-Edit design file, use **File > Replace Setup** and then choose the appropriate TDB or TTX file.

You can output an entire design rule set in list form to a text file. To create such a file, choose **Tools > DRC**, then click the **Setup** button. In the dialog **Setup Design Rules** (see Specifying Design Rules on page 3-69) click **Write to file**. The default name of the output file is *design*.rul, where *design* is the filename of the active design file.

Each design rule has an entry in the rule file, in the following form:

where the first two text lines specify the *name*, *type* (including any exceptions), and *distance* (with the associated *unit*) of the design rule, and the next lines specify the involved layers, one per line. Definitions for generated layers are also shown.

For example:

21.6 N Well Metal2 high density

Type: Density, Layer: density

Layer: N Well

DENSITY

Layer: Metal2

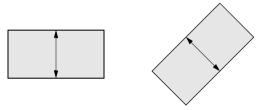
Design Rule Types

L-Edit supports seven types of design rules. Each of these rule types is described below, with specific examples shown in the accompanying figures.

- **Minimum Width** (page 3-18)
- **Exact Width** (page 3-19)
- **Not Exist** (page 3-19)
- **Spacing** (page 3-20)
- **Surround** (page 3-21)
- Overlap (page 3-21)
- **Extension** (page 3-23)
- Density (page 3-24)

Minimum Width

Minimum width rules specify the minimum width of all objects, in any direction, on the named layer.



Poly Minimum Width = 2 lambda

You can specify exceptions for this rule type as described in Rule Exceptions on page 3-25.

Exact Width

Exact width rules specify the exact width of all objects on the named layer. The width of octagons is measured between parallel sides.





Poly Contact Exact Width = 2 lambda

Not Exist

Not exist rules specify that no objects should exist on the named layer. Not exist rules are unique in having no associated distance.

Spacing

Spacing rules specify the minimum distance that should separate all pairs of objects, either on the same layer or two different layers.



Via to Poly Contact Spacing = 2 lambda

You can specify exceptions for this rule type as described in Rule Exceptions on page 3-25.

Surround

Surround rules specify that objects on one layer must be completely surrounded by objects on another layer.

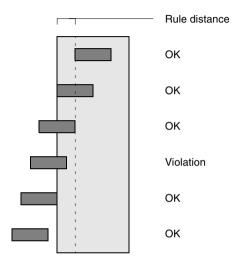


Metal2 Surround Via = 1 lambda

You can specify exceptions for this rule type as described in Rule Exceptions on page 3-25.

Overlap

Overlap rules specify the minimum amount that an object on one layer must overlap an object on another layer (when there is an overlap). Objects which overlap more than the specified distance or whose edges coincide are not considered in violation of overlap rules.

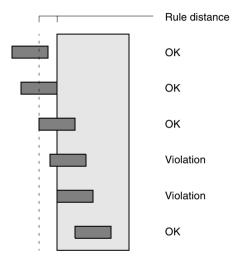


Poly Contact Overlap Poly = 2 lambda

Extension

Extension rules specify the minimum amount that an object on one layer must extend beyond the edge of an object on another layer. Objects are not considered in violation of extension rules when they:

- Extend more than the specified distance
- Have a coincident edge but are otherwise outside
- Are entirely surrounded



Poly Contact Extend Poly = 2 lambda

Density

The density rule finds and flags objects on the derived density layer specified in **Layer1**. The layer specified must be a **Density** type derived layer. Violations to the rule include any polygons output to a density layer.

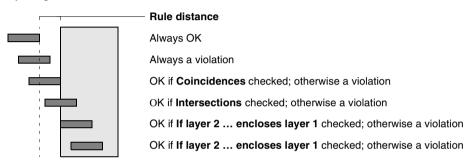
Rule Exceptions

You can fine-tune some rules by specifying particular layout conditions that are *not* to be reported as violations. These conditions are represented by the **Ignore** options in the dialog the dialog **Setup Design Rules** (see Specifying Design Rules on page 3-69).

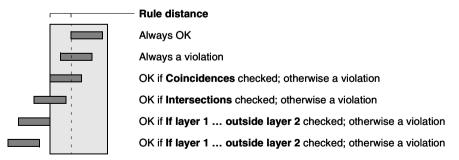
Condition	Description	Applicable rules
Coincidences	Coincident edges between objects are ignored.	Surround
Intersections	Intersections between objects are ignored.	Surround
If layer 2 completely encloses layer 1	Objects on one layer <i>entirely surrounded</i> by objects on another layer are ignored.	Spacing
If layer 1 completely outside layer 2	Objects on one layer <i>entirely outside</i> objects on another layer are ignored.	Surround
45 degree acute angles	Portions of objects containing angles of 45° (or less) are ignored.	Minimum width Spacing Surround

The following illustration shows exception conditions for spacing and surround rules.

Spacing



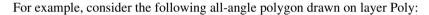
Surround

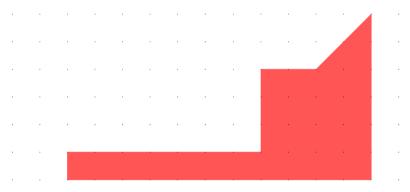


Acute Angles

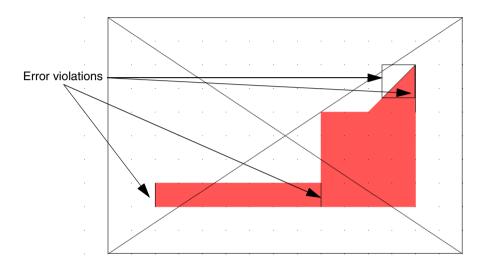
By checking the option **Ignore 45 degree acute angles**, you can instruct L-Edit to ignore acute angles while performing a design rule check.

This option does not exclude objects with angles of 45° or less from the design rule check. Instead, it excludes the specific portions of those objects that contain acute angles.

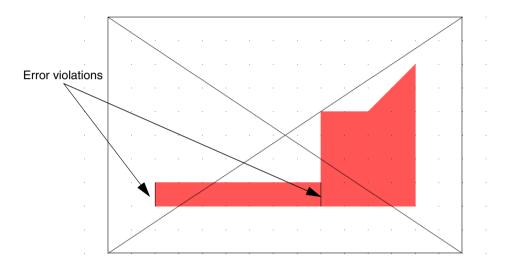




Checking this object for minimum width would ordinarily yield four violations, as marked here:



Running the same design rule check with the option **Ignore 45 degree angles**, however, excludes the upper right portion of this object, which contains an acute angle. With that portion excluded, L-Edit detects only two violations:



Dracula® Rule Set to L-Edit Design Rule Conversion

L-Edit provides a translation utility that converts a subset of the Cadence Dracula rule set into a TTX format text file that is imported into an L-Edit design file to define derived layer, technology, and DRC rule information.

The converter checks whether referenced layers used during design rule checking already exist in the TDB file, and creates them if they are not. In cases where a layer cannot be translated, all future instances of a command using that layer will not be translated. In cases where a rule in not translated, L-Edit creates a warning in the log file that is generated during conversion.

The converter generates other warning messages, particularly concerning rules or commands that are omitted from the translation process, or cases where the translation does not fully preserve the meaning of the corresponding Dracula command. Warnings are included as comments to the translated file in a separate log file.

The Rule Set Conversion Process

[1] To import a Dracula rule set into an L-Edit file, click the **Import Dracula** button on the **Setup Design Rules** dialog. (Use **Tools > DRC** and click the **Setup** button to open **Setup Design Rules**). Alternately, as the translator is a DOS program, you can open a DOS window and run the converter from the command line. (See Launching the Converter from DOS on page 3-42.)

- [2] Type in the name of the file to read as input, a name for the log file, and select the import options you would like to use.
- [3] L-Edit lauches and runs the converter to import layers and rules from the Dracula file. If there are any layer name conflicts between the Dracula and TDB files, L-Edit present a series of dialogs that you can use to resolve them.

Import Dracula



From file

Name of the Dracula input file. You can choose from available files and directories with the **Browse** button.

Merge Dracula setup into file < filename>

Creates a temporary TDB file containing the imported information, then merges the temporary file with the currently active TDB file.

If there are any conflicts, a three-step process allows you to resolve technology unit and setup mismatches, layer mapping, and layer name conflicts. (See Resolving Conflicts in Merged Files on page 3-35.)

Create new TDB file with Dracula settings

Creates a new TDB file containing Dracula rules.

Colorize non-existent layers

Adds a randomly selected outline color with solid fill to each imported layer.

Log file

Name of the log file in which translation warnings and omissions are recorded. You can choose from available files and directories with the **Browse** button.

If left blank, by default the name will be the name of the Dracula file appended with _Dracula.log, written to the directory of the TBD file.

Log Files

The log file contains the original Dracula file with comments on what was not translated.

Changes to layer names and new layers that result from merging files will be recorded at the end of the log file as shown below. The appended material will be the contents of the tables from the second and third steps of the **Import Conflict Resolution** dialog.

```
**** Layer name conflict resolution
                       **** L-Edit Win32 8.30.20001206.09:45:55
                        **** Target TDB file: "D:\My
Documents\LEv8.3\sirf ind v830.tdb"
Dracula input layer
                        "NWELi "(GDS=2) mapped into layer "NW"(GDS=2)
Dracula input layer
                       "HOTWL" (GDS=51) copied into layer "HOTWL"
Dracula input layer
                               "(GDS=3) mapped into layer "OD"(GDS=3)
                        "ODi
                       "PDIFi "(GDS=11) mapped into layer "ODP"(GDS=11)
Dracula input layer
Dracula input layer
                       "NDIFi "(GDS=12) mapped into layer "ODN"(GDS=12)
Dracula input layer
                        "OD2i
                               "(GDS=4) mapped into layer "OD2"(GDS=4)
. . .
```

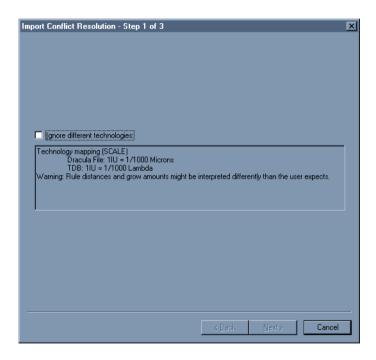
Resolving Conflicts in Merged Files

A three step process allows you to resolve conflicts between the Dracula rule set and critical settings in your TBD file. A record of all conflict resolution actions are appended to the log file once merging is complete.

Internal and Technology Units

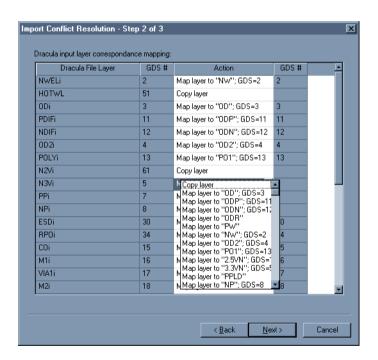
If the technology units in the Dracula file are different than those in the TDB file, rule distances and grow (or shrink) amounts might be incorrectly interpreted. When there is not a direct scaling relationship between technology units or the

technology setup of the two files, you will be warned as shown in the dialog below. You must enable the checkbox to proceed.



Input Layers

This step allows you to map drawn Dracula layers to L-Edit layers.



Dracula File Layer

Lists the layers from the Dracula Input-Layer block that were imported to the TDB file.

•

GDS # Displays the GDS number of the rule set layer.

Action A pull-down menu for each layer allows you to

either copy it into the TDB file or map it to an

existing L-Edit layer.

When L-Edit finds a GDS number match it will automatically map the rule set layer to the

L-Edit layer.

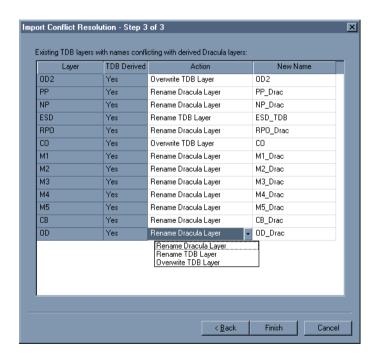
GDS # Displays the GDS number the layer will have in the merged TDB file. If the layer already has

a GDS number in L-Edit it will be entered

automatically.

Layer Names

Clicking on the **Action** field in this dialog allows you to resolve any layer name conflicts for derived Dracula layers.



There are three ways to resolve any layer name conflicts:

- Rename Dracula Layer—changes the name of the rule set layer and copies the renamed layer to the TDB file at the end of the layer list. This is the default action for drawn layer conflicts; L-Edit will append _Drac to the layer name.
- Rename TDB Layer—changes the name of the TDB layer and copies the rule set layer to TDB file at the end of the layer list.
- Overwrite TDB Layer—overwrites the TDB layer with the derivation for the rule set layer. This is the default action for derived layer conflicts; L-Edit will adopt the rule set layer name.

Dracula layers are added to the end of the layer list. L-Edit then checks the derivation dependancies for derived layers. If a layer depends on another that is after it in the list, the layer will be moved to be immediately after the last layer that it depends upon.

For example, if layer A depends on B and C, and the order of layers after merging files is:

then layer A will be moved so that it is immediately after C:

This operation is completed in layer list order and dependency problems are resolved as they appear. If, for example, layer Y depends on layer A, the previous move would create a problem because layer A is now after layer Y.

Because Y was initially after A, its dependencies are resolved after A is moved. The new order will therefore be:

X, B, Z, C, A, Y, T

After all layer order moves, derived layers are again chaecked for validity. If a lyer's derivation is invalid its derivation will be disabled and the layer will appear as derived but disabled in the **Setup Layers** dialog.

Note that these layer order changes and layer disabling due to invalid derivations are not recorded in the log file.

Rule Naming Convention

The naming convention for a rule, if it has the form **rule OUTPUT** coutput
name>, is that L-Edit name will be <output</pre>
coutput
name>rule
If the rule has no
output clause, then the rule will simply be **rule**.

A layer derivation or selection that has an output clause generates both the derived layer and a NOT_EXISTS rule for that layer. For example, **AND A B C OUTPUT D** generates the layer **C** and a rule **D AND A B C**, which tests **NOT_EXISTS C** to flag as an error anything that appears on layer **C**.

Substrate Layer

For files that have a "substrate" layer, a special layer (called "Empty Layer") is constructed for the derivation of an appropriate substrate layer. That layer should be kept empty.

Launching the Converter from DOS

To launch the rule set converter in a DOS window, enter the following command from the DOS prompt:

```
Dracula2ttx -i <input_dracula_file> -o <output_ttx_file.ttx>
    -e <error file.txt> [-s] [-c #]
```

The following command line option flags may be used:

-i	The path and filename of the Dracula input file.
-0	The path and filename of the TTX output file. You should use the extention .TTX.
-е	Creates a separate log file, for which you must enter a path and filename.

Special layers. Definitions of L-Edit special layers, which are used for drawing, display and error reporting, are included with the common default specification.

-c# Colorize. With this option selected, arbitrary color outlines with black fill are created for all layers defined in the output TTX file. # is the number of colors in the palette of the TDB file.

For example:

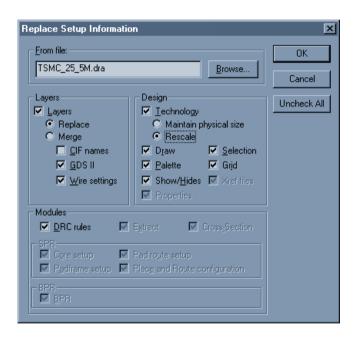
c:\mydocu~1\lev8.3>Dracula2ttx -i ruleset1.ttx -o
dracdrc1.ttx -e dr2ttx.err

Command Line Conversion Process

- [1] To translate a Dracula rule set to an L-Edit TTX file using the command line, open a DOS window and change to the directory in which you would like to create the text file.
- [2] Enter the conversion executable command, the name of the file to read as input and the name of the TTX file that will be output, and your choice of the command line options available. A brief copyright notice will appear when the translation process is complete.

[3] When replace setup is complete, L-Edit displays the log file in a text window. Note that existing TTX and log files from previous runs are overwritten without warning by the translator.

[4] In an L-Edit design file, use **File > Replace Setup** to import the derived layer and DRC rules information from the TTX file. Use the following options:



[5] Run **Tools > DRC**, using the **Setup** button to make any necessary design rule modifications.

Supported Translations and Naming Conventions

The converter translates only commands in the Operation Block (those between *OPERATION and the following *END directive), with the exception of input layers and some technology parameters.

Layers

All layer derivations and drawn layers are translated. Those which are not feasible because component layers could not be constructed are not translated. Each derived layer constructed by the translated rule set is given the same name as in the Dracula file.

Temporary layers are supported, but these are renamed, since L-Edit does not support layers which are periodically reused within the same rules file. Temporary layer names, where necessary, are derived from the full name of the operations performed to construct them.

Rules

DRC rules that cannot be translated or that involve undefined layers are not translated. Each of these situations yields a warning. Untranslated rules and derivations, except as noted in Supported Commands on page 3-47, yield warnings.

Since L-Edit does not have the facility for conjunctive rules, only the first part of any conjunctive rule is translated. Any command relying on the output of a DRC check (such as those using the [R] option) will not be translated.

Each L-Edit rule is given the corresponding Dracula rule as its name. In cases where a rule flag is omitted by the translator, the rule name shows the rule that most closely approximates it.

Supported Commands

A Dracula rule set file consists of four blocks, a Description block, an Input-Layer block, an Operation block and a Plotting block.

The commands that will be translated for each block, and their corresponding L-Edit settings, are given in the following sections.

Translated Description Block Commands

Command L-Edit Equivalence

Scale Sets the Technology Units to microns and

specifies the number of Microns per Internal

Unit (in the Setup Design dialog).

RESOLUTION = step-size

units

Specifies the **Grid Size** that will be used with **Flag off-grid objects** in **Design Rule Check—Advanced**. RESOLUTION should be an integer multiple of SCALE, since nothing can be represented with resolution finer than this amount.

In the L-Edit file, everything is expressed internally in terms of internal units, which are set by the SCALE command. If, for example, SCALE = 0.001 microns, the smallest representible unit is 1/1000 microns. If RESOLUTION = 0.05, then it is passed to L-Edit as 0.05/0.001, rounded up to the nearest integer.

FLAG-OFFGRID = YES / NO {grid-value}	Enables Flag off-grid objects in Design Rule Check—Advanced. If present, grid-value will overide resolution as the step size for off grid checking.
FLAG-NON45 = YES / NO	Enables Flag objects ignored by DRC in Design Rule Check—Advanced with respect to paths and wires.
FLAG-PTH-OFFGRID = YES / NO	Enables Flag off-grid objects in Design Rule Check—Advanced.
FLAG-SELFTOUCH = YES {YES DONUT} / NO	Enables the Polygons with ambiguous fills options in Design Rule Check—Advanced. NO corresponds to Don't Flag, YES corresponds to Flag.
FLAG-SELFINTERS = YES {FULL} / NO	Enables Flag self-intersecting polygons and wires in Design Rule Check—Advanced

Translated Input-Layer Block Commands

Command L-Edit Equivalence

Layer Name Definition

layer-name = layer-number

The layer-name variable assigns names and

should be created whose derivation is equal to

ger-name = layer-number GDSII numbers to layout layers.

SUBSTRATE

Generates a substrate layer you can use as a regular layer in an operation. This is used only to setup a layer name definition, no special handling of the substrate is done. If the substrate layer does not exist, then a layer

NOT (empty layer).

Translated Operation Block Commands

The following Boolean operations are supported: AND, OR, ANDNOT, NOT, and XOR. The following select operations are supported: INSIDE, OUTSIDE, HOLE, CUT, TOUCH, ENCLOSE, OVERLAP, and VERTEX. The area command is also supported. The following DRC commands are supported: ENC, INT, EXT, WIDTH (see table below).

The option that allows you to place the constructed polygons into particular cells and/or layers is universally ignored.

Command L-Edit Equivalence

AND

AND layer-a layer-b trapfile {OUTPUT c-name l-num {d-num}}

Creates a new layer from the region shared by two other layers. The OUTPUT option is interpreted as a "not exists" DRC rule and no geometry is written to the cell c-name.

The edge option (AND[E] layera layerb layerc) is ignored.

Trapfile = layer-a AND layer-b

AREA

AREA layer-a {RANGE/NE/ EQ} n1 n2 {layer} {OUTPUT c-name l-num {d-num}} Performs a minimum area check to determine if polygon areas are within an area range. This command also creates a layer with polygons that are within specified area boundaries. A NOT EXIST rule is always created for this rules, and no geometry is written to the cell cname.

trapfile = AREA layer-a RANGE n1 n2 trapfile = AREA layer-a EQ n1 trapfile = AREA layer-a NE n1

The following DRC rule is created: Not Exists: trapfile

ENC {[EOT]} layer-a layer-b LT n1

Determines the degree to which layer-b polygons partially or fully enclose layer-a polygons by measuring between the outer edges of layer-a polygons and the inner edges of layer-b polygons.

The following options are translated:

E - turn off ignore outside

O - turn off ignore intersections T - turn off ignore coincidences

The following command options are ignored: C, C', N, N', P, P', R, R', S, U, U', V, X, Y.

The commands with the following operators are ignored: LE/RANGE/SELLT/SELLE/SELGT/SELGE/SELRA/GT/GE/EQ/SELEQ/SELNE. Conjunctive layer renaming and the O option for layers are also ignored.

EXT

EXT{[H]} layer-a {layer-b} LT n1

Checks the external separation between outside edges of polygons. The outside edges of both layers are checked against each other if you specify two layers. The outside edges of a layer are checked against other edges on that layer when you specify one layer.

The EXT rule has an H option which flags the outside edges of notched layer-a polygons that fail the spacing check. Because L-Edit ignores notches in geometry, the EXT rule is translated into a rule that behaves the same whether or not the "H" flag is present. A warning is issued when an EXT command is translated to ndicate that notches are not considered during the DRC spacing check.

The following command options are ignored: E, C, C', G, N, N', O, P, P', R, R', S, T, T', U, U', V, X, Y, Z.

Spacing:
Rule Distance = n1,
Layer1 = layer-a,
{Layer2 = layer-b},
do not not ignore 45 degree acute angles,
ignore coincidences,
ignore intersections,
ignore if layer1 completely encloses layer2.

GROW input-layer dx dy {dx2 dy2} trapfile

SHRINK input-layer dx dy {dx2 dy2} trapfile

Oversizes or shrinks objects on a layer, where dx and dy determine the direction and amount of growth. This rule is only translated if dx = dy = dx2 = dy2.

trapfile = GROW input-layer by dx In the case of SHRINK, this is translated as trapfile = GROW input-layer by-dx

INT layer-a layer-b LT n1

Determines the depth to which polygons of two layers overlap by measuring between the inside edges of polygons. No options are translated.

Overlap: Rule Distance = n1, Layer1 = layer-b, Layer2 = layer-a. NOT layer-a layer-b trapfile

Creates a new layer from two others consisting of the first layer minus the region shared by both layers.

trapfile = layer-a AND NOT layer-b.

OR layer-a layer-b trapfile

Creates a new layer from two others consisting of the total region of both layers.

trapfile = layer-a OR layer-b.

SELECT layer-a INSIDE | OUTSIDE | HOLE layer-b trapfile

SELECT {NOT} layer-a CUT | TOUCH | ENCLOSE | OVERLAP {[n1:n2]} layer-b trapfile

SELECT layer-a VERTEX[n1:n2] trapfile An area-relation command that selects a group of polygons from a layer and creates a new layer with the results. If a polygon on layer-a is related to layer-b, the layer-a polygon is output.

The N option that turns on node-based rather than polygon-based selection is ignored. The ANGLE relation that lets you separate allangle geometry from Manhattan geometry is ignored.

Trapfile = layer-a [NOT] [INSIDE|OUTSIDE|CUT|TOUCH|ENCLOSE| OVERLAP] layer-b

Trapfile = layer-a [NOT] VERTEX MIN n1 MAX=n2

Output creates a NOT EXISTS rule in L-Edit.

SIZE input-layer BY n1 trapfile

Creates a new layer from another layer by oversizing or undersizing the new layer by a specified number of layout units and moving the sides of the polygon perpendicularly.

A positive value oversizes and a negative value undersizes the polygon.

The single letter options (SIZE[options]) are all ignored.

trapfile = GROW n1 input-layer.

WIDTH{[D]} layer-a LT n1

Checks the dimensions between the inside edges of polygons on the input layer. Only the D option which disable acute-angle checking is translated. This command is used to ignore 45 degree acute angles.

Min Width: Rule Distance = n1, Layer = layer-a,

XOR layer-a layer-b trapfile

Creates a new layer from two other layers where the new layer consists of the total region of both layers minus the region shared by both layers.

TempLayer1 = layer-a AND NOT layer-b. TempLayer2 = layer-b AND NOT layer-a. trapfile = TempLayer1 OR TempLayer2.

Conjunctive rules (those connected with the symbol "&") and conjunctive renaming are not supported; any rules occurring in a conjunctive sequence are not translated. Only the LT ("less than") comparison operator is translatable.

Additional CORNER-CORNER and CORNER-EDGE options are not supported. (L-Edit handles only "round" corners.) Optional arguments that send the resulting error flags to a particular cell on a particular layer are also not supported. However, in each of these cases, the closest translation is provided and a warning is emitted.

Ignored Flags

The following flags are not supported. In these cases, L-Edit/ DRC may produce additional errors that would otherwise be omitted. Warnings are emitted by the translator for these flags, if present.

- C (flag only parallels)
- C' (flag only non-parallel)
- N (don't flag connected)
- N' (flag connected)
- P (flag segments that project onto each other)
- P' (inverse of P)
- U (do not flag violations on same node)
- U' (flag only on same node)

The following flags produce additional geometric output indicating where errors were found. This geometry may then be further processed by other commands. Rules using these flags are translated, but the resulting layer containing error geometry is not produced, and subsequent rules using those layers are not translated.

- R (output trapezoids enclosing projections of violating pairs)
- R' (output errors as trapezoids)

The following flags are not supported.

- + S (square, rather than round, corners)
- T' (flag corners that touch)

- V (check against all inside edges, not just enclosing ones)
- X (check for this delta in x-direction if edges project onto each other)
- Y (similt to the X flag but applied to the y-direction)

The flags 'E' and 'O' are supported on EXT and ENC. The 'T' flag is supported in all of EXT, ENC and INT.

WIDTH checks are similar, although the options vary. Only the 'D' flag ('ignore acute angles') is translated. Width rules can use both the LT ('minimum width') and EQ ('exact width') comparisons.

Omitted Commands in the Operation Block

There is currently no L-Edit equivalent for the following commands:

- CORNER (select and output corner data for special checking)
- COVERAGE (report an error if area surrounding a point is not covered to a specified percentage)
- CUT
- EXPLODE
- OCTBIAS (cuts Manhattan corners into octagonal corners)
- FLOATCHK (finds floating contacts)

- NEIGHBOR (for improving performance on certain layouts)
- LENGTH, PLENGTH, RLENGTH
- RELOCATE (shift a layer by a specified x and y, and possibly rescale)
- SNAP (snap geometry to a specified grid)
- TRIANGLE (create triangles at corners)

Skipped Commands

The following operations are not required for correctness within a sequence of Tanner rules, and will be skipped with no change in semantics of the rules executed:

- FLATTEN
- FRACTURE
- OVERLAP (process overlapping polygons from graphics system)

Checking the Layout

When your layout is complete, you should check for design rule violations before sending the layout to the chip foundry for fabrication. If a chip is fabricated with design rule violations, it may fail to function as designed.

You can run DRC against an entire cell or a specific region ("DRC box") of the cell.

To perform a *full-cell* check on the active cell's layout, select **Tools > DRC**. To perform a region-only check, choose **Tools > DRC Box**.

Region-Only Design Rule Check

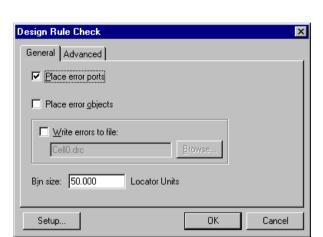
Tools > DRC Box performs *region-only* checking within the active cell's layout.

Run a region-only check when a restricted area or group of objects in the layout needs to be checked for design rule errors. A region-only check is useful for interim verification during layout creation or to confirm that a design rule violation in a specific region has been corrected.

Note that because the select and area operations require a full cell check, if either of these operations is defined and enabled, L-Edit will warn that checking has been confined to just the specified DRC box.

To perform a region-only check, select **Tools > DRC Box**. Then use the DRAW mouse button to drag a rectangular region around the area of the layout to be checked. When the DRAW button is released, L-Edit opens the **Design Rule Check** dialog, just as it would for the full-cell check. When the check is complete, a box on the Error layer indicates the region that was checked. Errors are reported just as they are for a full cell check.

Sporadic region-only checking is a good practice, but does not substitute for a full-cell check. You should always run a full-cell check after the last layout alteration and before submitting a design for fabrication.



L-Edit displays the **Design Rule Check** dialog for both DRC commands:

On the **General** tab, you can select the following options:

Place error ports

Instructs L-Edit to place an *error port* on the specified error layer at the location of each violation. For additional information on error ports, see Error Markers on page 3-75.

Place error objects Instructs L-Edit to place a marker object on the

specified error layer at the location of each violation. For additional information on error objects, see Error Markers on page 3-75.

Write errors to file Instructs L-Edit to write errors to a text file.

The default name of the file is *cell.drc*, where *cell* is the name of the current cell. For additional information on the format of this

file, see Error Files on page 3-77.

Bin size Use to specify the bin size in locator units.

Binning

When you run DRC, L-Edit divides the area to be checked into a grid of square *bins*. L-Edit then performs the check one bin at a time, beginning at the lower left corner of the cell or region and moving to the right. At the right edge of the area, L-Edit moves up to the next row, and proceeds again from the left.

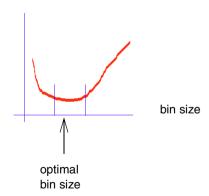
L-Edit marks each bin with a large X from corner to corner if there is a design rule violation within its boundaries. When the whole check is completed, L-Edit displays the total number of errors and warnings detected.

Bin Sizing

In order to ensure that layout is checked for violations at or between bin boundaries, DRC checks a small portion of the overlap between adjacent bins by "bloating" each bin as it is checked.

If you define a bin size that is too small, DRC can ultimately check each piece of the layout several times, causing an unnecessary increase in DRC processing time. If you define a bin size that is too large, DRC processing time can also increase significantly due to the longer checking time needed for larger bins (checking time is approximately proportional to bin area). The curve below shows the typical increase in processing time for different bins sizes:

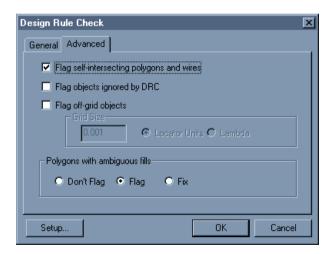
time



Therefore, you should define the largest possible bin size that will yield the shortest processing time. A good method for estimating is to run DRC with several bin sizes. Create an error file each time and use the DRC run time recorded in the error file to determine the optimum bin size.

Note that to keep polygon vertices on grid, the actual bin size used to generate layers is incremented to snap to a multiple of the mouse snap grid (set in **Setup Design > Grid > Mouse snap grid**).

To set options for polygon integrity checking, click the **Advanced** tab:



Set the following options:

Flag self-intersecting polygons and wires

Flags self-intersecting polygons and wires (described in Self-Intersecting and Ambiguous Fill Polygons on page 1-258).

Flag objects ignored by DRC

Flags objects that are ignored by DRC. Ignored objects include:

- all-angle wires and polygons
- circles, arcs, and tori
- polygons with curved edges
- wires with a round end or join style
- wires with a miter join style

Ignored objects are reported in the DRC log file but counted separately.

Flag off-grid objects

Flags objects that do not have all edges on the specified grid, as well as off-grid vertices and cells with off-grid placement. (This checking is performed before layout data is merged.) All errors are flagged with an error port and added to the DRC log.

Grid Size

Enter the grid size to use if off-grid flagging is enabled. Choose **locator units** or **lamba**.

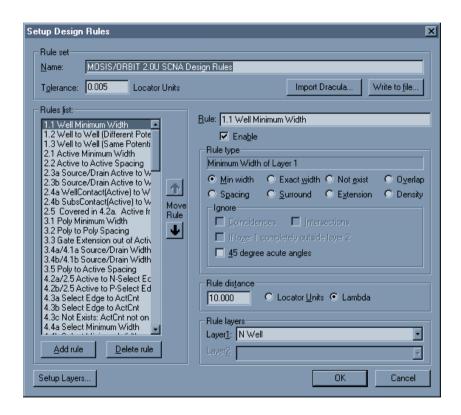
Polygons with ambiguous fills

Checks for polygons with ambiguous fill (as described in Self-Intersecting and Ambiguous Fill Polygons on page 1-258) and performs the following:

- Don't Flag—ignore polygons with ambiguous fill.
- Flag—Flag polygons with ambiguous fill.
- Fix—fix polygons with ambiguous fill by merging them.

Specifying Design Rules

Click **Setup** to open the **Setup Design Rules** dialog, which allows you to modify design rules.



Rule set

Name identifies the design rule set. **Tolerance** is a value T, common to all rules in the set, which together with the distance D for each rule (see below) determines the precision of error checking. A distance on the layout must be outside the range $D\pm T$ to be flagged as a violation. A minimum tolerance of 5 internal units is required for designs containing 45° geometry. The tolerance may be set to zero for pure orthogonal designs.

Import Dracula

Opens the **Import Dracula** dialog. See Import Dracula on page 3-32 for information on the fields in this dialog.

Write to file

Opens the **Write DRC Rules to File** dialog, which prompts for the name, path (location), and type (extension) for the text file. There is one predefined file type: **.rul**.

Rules list	The list of available rules. To add a rule, click Add rule and type the name of the new rule in the Rule field. To delete a rule, highlight it and click Delete rule . Rules can be repositioned in the list by highlighting the rule and clicking the up or down Move Rule arrows.
Rule	The name of the rule highlighted in the Rules list. Rule type, Ignore, Rule distance, and Rule layers all pertain to the uniquely named rule.
Enable	If this box is checked, the rule will be checked when DRC is run. If violations are detected, the names of the violated rules are indicated.
Rule type	Selected by clicking the appropriate option button. See Design Rule Types on page 3-17 for information on the supported types.

Ignore

Cases which will not be considered a design rule violation. Options include:

- Coincidences
- Intersections
- If layer 1 completely outside layer2
- 45 degree acute angles

See Rule Exceptions on page 3-25 and Acute Angles on page 3-27 for further information on the use of these options.

Rule distance

The distance value associated with a rule. Distances are measured either in locator units or in user units. User units are defined with the command **Setup Design—Technology**—see Technology Parameters on page 1-134.

Rule layers

DRC specifies which layers are involved in each of the design rules. For example, selecting the **Spacing** rule type automatically specifies **Minimum Layer[] to Layer[] spacing**. The appropriate layers can be selected from the drop-down menus.

Working with 45° Layout

Generation of layers containing 45° objects can produce off-grid vertices due to off-grid intersections of 45° polygons or conversion of 45° wires to polygons. For further information on this topic, see Working with 45° Objects on page 1-476.

Correcting Errors

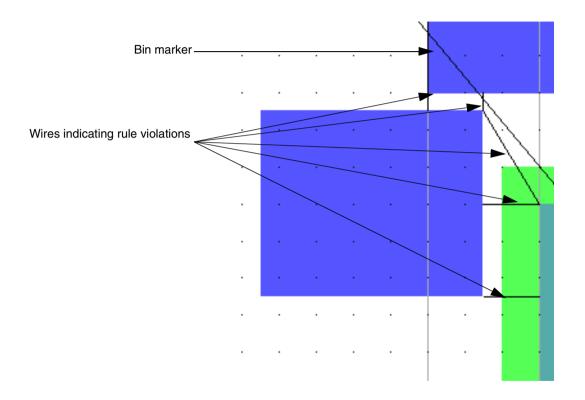
Error Markers

When you specify **Place error ports** or **Place error objects** in the **DRC** dialog, L-Edit places *error ports* and/or *error objects* on the Error layer at violation locations. These are placed in addition to the diagonally marked boxes that always appear, which represent bins containing violations.

An error port consists of the name of the violated design rule and a bracketed expression; the expression indicates the spacing or nature of the error and the rule distance. Error ports and objects can be moved, deleted, hidden, and shown in the same manner as other objects.

For example, an error port named **8.4c Via to Active Spacing [1 < 2]** shows that the associated violation involves a spacing of 1, when the minimum Via-Active spacing should be 2.

An error object is a wire that spans the distance between any objects violating a design rule. In the following illustration, for example, the prominent central Metal1 box creates five separate violations of the rule Metal1 to Metal1 spacing, and each violation is indicated by a wire. (Layers irrelevant for the purpose of this illustration are hidden for clarity).



Error Files

When you select **Write errors to file** in the **Design Rule Check** dialog, L-Edit writes design rule errors to a text file with the extension **.drc**. You can open these files with any text editor.

The default name of the error file is **cell.drc** (where **cell** is the name of the checked cell) in the following format:

```
DRC Errors in cell cell of file file.

rule = value unit; (x1,y1) \rightarrow (x2,y2)

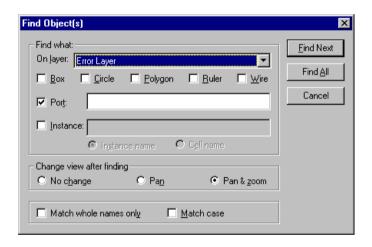
...

number errors.
...
```

The first line names the checked **cell** and **file**. The second line is repeated once for every error: the name of the **rule** violated, the required **value** for the specified rule, the **unit** of the specified value, and coordinates (**x1,y1**) and (**x2,y2**) indicating the location of the error on the layout. The last line shows the total **number** of errors reported. The error file also includes further information about timing.

Finding Error Markers

Edit > Find, along with **Edit > Find Next** and **Edit > Find Previous**, can locate reported rule violations.



Specify a **Wire** search (for error objects) or a **Port** search (for error ports) on the Error layer. For ease of viewing, specify **Pan** or **Pan & zoom** as the viewing option.

Because of the default rendering setup of the Error layer, a port's text is only visible when the port is selected. Thus, as the search commands go through ports on the Error layer, the names of the error ports become visible one at a time.

Clearing Error Markers

Tools > Clear Error Layer opens the following dialog, prompts for options, then removes error markers (ports and objects).



Options include:

This cell (*cell name***)** Removes all objects on the Error layer in the

active cell

This file (*file name***)** Removes all objects on the Error layer in the

active file

All files

Removes all objects on the Error layer in all open files

It is not necessary to run **Clear Error Layer** before running **DRC**. **DRC** automatically clears the Error layer in the active cell and in its instanced cells as its first step.

Note:

This command cannot be undone.

Optimizing Performance

Design rule checking is a complicated, computation-intensive process that involves large numbers of comparisons and measurements. Some DRC runs can result in very long execution times. This section outlines several ways to achieve faster results.

Reducing Distances

The time required to perform a design rule check is related to the greatest distance specified in the rule set.

When you run DRC on a region or bin, L-Edit *bloats*, or increases, its boundaries by a distance $B = \max(R, G)$, where R is the maximum distance specified in the design rule set and G is the sum of the absolute values of all Grow amounts specified in generated-layer definitions. You specify R in the dialog **Design Rule Setup** (see Specifying Design Rules on page 3-69) and G in the dialog **Setup Layers—Rendering** (see Defining Boolean Layer Derivations on page 1-441).

To optimize performance, you should do a series of DRC runs on a small sample cell using a range of bin sizes to find the bin size that yields the shortest runtime.

As the distances specified in design rules or Grow amounts increase, so does the time required for rule checking.

Disabling large distance rules, where appropriate, can produce significant performance improvements. For example, disabling Pad rules when checking a core cell would be a logical choice.

Checking Incrementally

The time required for checking also varies, roughly, with the square of the number of objects to be checked in each bin or region. Thus, as the design size increases, the time needed for a complete check increases dramatically.

Use region-only checks at convenient stopping points in the design process. Performing region-only checks will also help prevent compounding of errors which might require extensive layout modification to correct. Use the full-cell check for completed cells and at least once on the final design.

Hiding Layers

DRC does not check rules involving layers hidden at execution time. Once you have examined and repaired all violations involving particular layers, you can hide these layers and reduce execution times for subsequent DRC runs involving other layers.

Disabling Rules

You can use the **Enable** option in the **Setup Design Rules** dialog to turn off rules that do not need to be checked.

For example, pad spacing rules typically involve distances far greater than other rules. A rule like Pad Comment Width might be set for 100 internal units; if this rule were left active, *each bin* to be checked would be bloated by 100 internal units *on each side*, to nine times the original area. Effectively, nine bins would need to be checked for each bin in the original grid.

Since the time required to perform a design rule check varies roughly with the square of the number of objects to be checked, the performance time would increase (assuming constant object density) by approximately a factor of 81. A more efficient operation would be to run DRC separately for rules involving large distances, such as pad spacing rules.

* Dracula is a registered trademark of Cadence Design Systems.