SIEMENS EDA

Calibre® CellArray RET User's and Reference Manual

Software Version 2021.2

SIEMENS

Unpublished work. © 2021 Siemens

This material contains trade secrets or otherwise confidential information owned by Siemens Industry Software, Inc., its subsidiaries or its affiliates (collectively, "Siemens"), or its licensors. Access to and use of this information is strictly limited as set forth in Customer's applicable agreement with Siemens. This material may not be copied, distributed, or otherwise disclosed outside of Customer's facilities without the express written permission of Siemens, and may not be used in any way not expressly authorized by Siemens.

This document is for information and instruction purposes. Siemens reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Siemens to determine whether any changes have been made. Siemens disclaims all warranties with respect to this document including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement of intellectual property.

The terms and conditions governing the sale and licensing of Siemens products are set forth in written agreements between Siemens and its customers. Siemens' **End User License Agreement** may be viewed at: www.plm.automation.siemens.com/global/en/legal/online-terms/index.html.

No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Siemens whatsoever.

TRADEMARKS: The trademarks, logos, and service marks ("Marks") used herein are the property of Siemens or other parties. No one is permitted to use these Marks without the prior written consent of Siemens or the owner of the Marks, as applicable. The use herein of third party Marks is not an attempt to indicate Siemens as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A list of Siemens' trademarks may be viewed at: www.plm.automation.siemens.com/global/en/legal/trademarks.html. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Support Center: support.sw.siemens.com
Send Feedback on Documentation: support.sw.siemens.com/doc_feedback_form

Table of Contents

Chapter 1
Introduction to Calibre CellArray RET (CA-RET) 9
Calibre CellArray RET Application 9
Calibre CellArray RET Flow 11
Calibre CellArray RET Processing Requirements 13
Syntax Conventions 13

Chapter 2
Getting Started with Calibre CellArray RET 15
Running Calibre CellArray RET 15

Chapter 3
Calibre CellArray RET Command Reference 17
SVRF Commands 18
LITHO DENSEOPC CELLARRAY 19
LITHO FILE Parameters 21
RET CELLARRAY 25
RET SBAR 28

Appendix A
Calibre CellArray RET Example SVRF Files 31
Calibre CellArray RET SVRF File 31

Index

Third-Party Information

List of Figures

Figure 1-1. Calibre CellArray RET General Overview 10
Figure 1-2. Calibre CellArray RET Flow 11

List of Tables

Table 1-1. Syntax Conventions 13
Table 3-1. SVRF Commands 18
Table 3-2. MAP Keywords 22

Chapter 1
Introduction to Calibre CellArray RET (CA-RET)

Calibre® CellArray Resolution Enhancement Techniques (CA-RET) uses a flow of multiple tools through a defined methodology to insert sub-resolution assist features (SRAFs) and optical proximity correction (OPC) shapes into large, repetitive cell-based designs. The Calibre CellArray RET methodology includes pattern classification, SRAF generation, and OPC tools.

Memory device applications have evolved from simple devices to system-level advanced solutions. Advanced solution requirements have driven the transition from planar to 3D scaling and other emerging technologies to face unique manufacturing and process challenges.

Calibre CellArray RET Application 9
Calibre CellArray RET Flow 11
Calibre CellArray RET Processing Requirements 13
Syntax Conventions 13

Calibre CellArray RET Application

The Calibre® CellArray RET methodology is especially applicable for highly repetitive arrays.

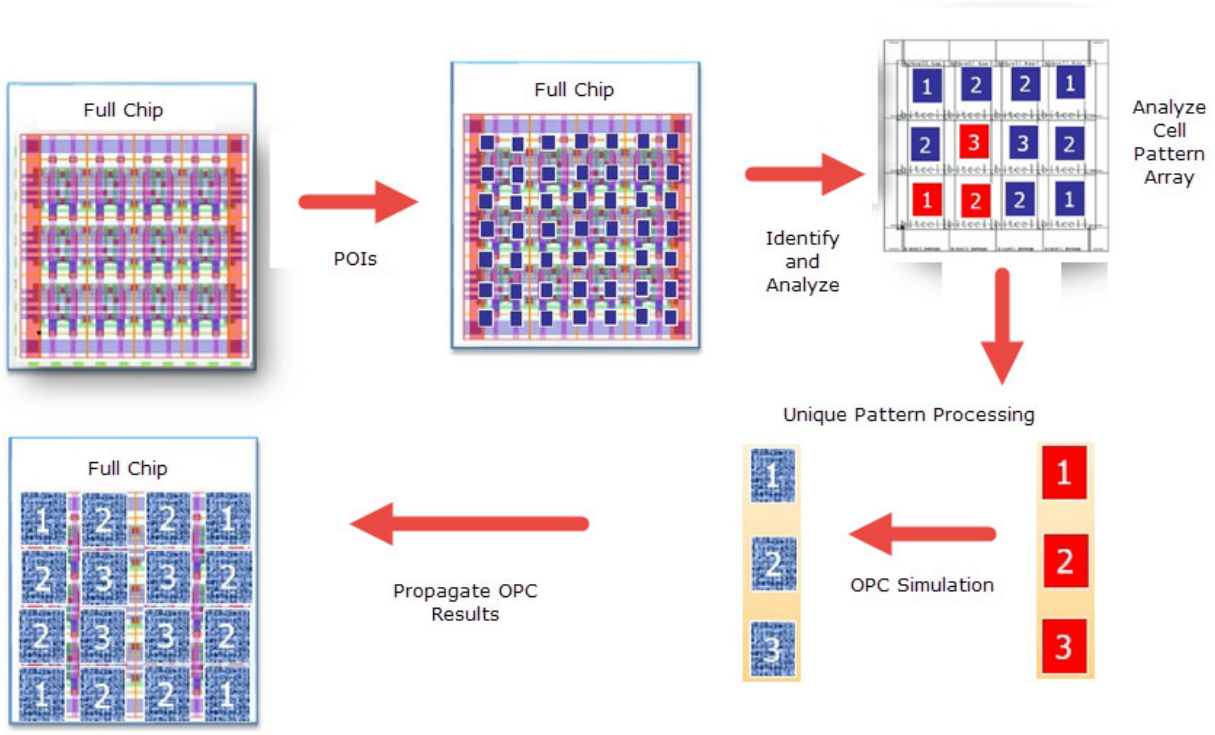
3D NAND technologies are widely used for storage applications on devices such as smart phones and solid-state storage devices (SSDs). These devices have highly repetitive array structures with repeated cores, requiring processing approaches with geometric consistency and acceptable processing times. These requirements are addressed by Calibre CellArray methodologies.

The Calibre CellArray RET methodology is a pattern-based approach, automatically determining unique representative locations known as points of interest (POIs) within the design. These are the patterns requiring lithographic simulation. Computation for OPC is performed only at selected unique pattern structures and the post-OPC results are then propagated to the rest of the pattern instances.

A generalized approach and its terms require some description.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Figure 1-1. Calibre CellArray RET General Overview



Points of Interest Collection and Halo Selection

To ensure full chip pattern coverage, points of interest must be derived within the target layer of all cells across the chip. In addition to intra-cell targets, SRAF generation is performed for both inter-cell and cell periphery areas. The POIs are considered the core locations for all the pattern instances. To detect the impact of context on each pattern, a proper halo size is identified to ensure lithography simulation output accuracy; this halo is dictated by the process and optical diameter of the simulation that leads to large and highly-overlapping windows when compared to the original cell dimensions.

Unique Pattern Identification

Identifying patterns based on all the automatically placed POIs and the large halo size ensures both full chip scanning and coverage in addition to having adequate context for each pattern. The patterns are then analyzed to identify a unique representative for each group, taking into account both orientation and transformations. Additional optimization techniques are used in the selection of the unique locations to improve the lithography simulation accuracy and enable uniform contexts.

OPC Simulation at Unique Locations

The number of unique locations that undergo OPC simulation represent a significantly small portion of the full chip layout. The capability of reducing the simulation computation portion of

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

the OPC flow to a few locations enables use of Calibre nmOPC providing higher resolution and accuracy with short run times.

Propagating OPC Simulation Results to all Pattern Instances

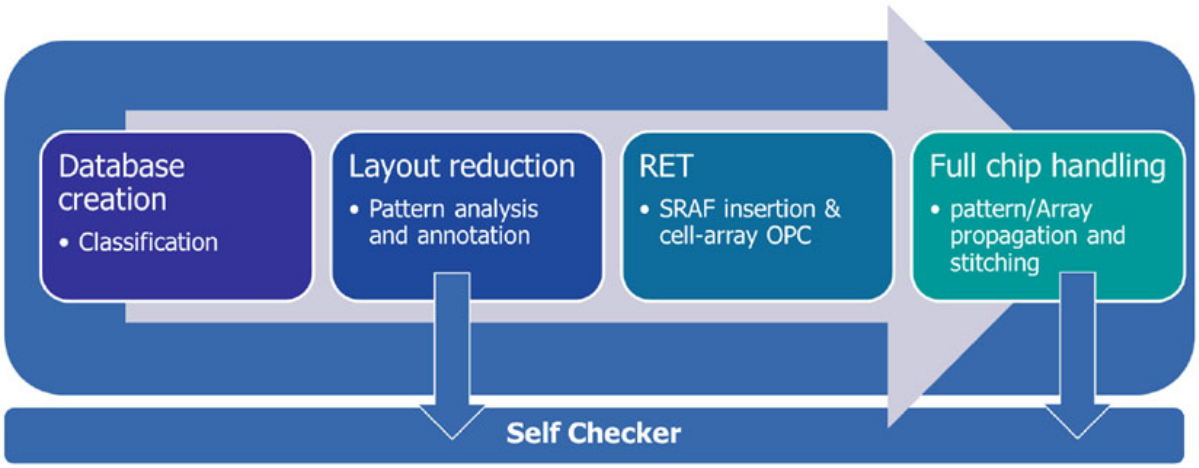
After OPC simulation is performed on unique pattern representatives, OPC results are then propagated to the pattern instances across the chip. The accuracy of the pattern identification and analysis are essential for stitching the results together.

Calibre CellArray RET Flow

Calibre CellArray RET employs a pattern-based SRAF insertion and OPC approach to ensure consistent full chip processing of memory devices.

The Calibre CellArray RET methodology is developed to meet the rigors of processing array structures. It uses four general phases, all supported by a self-validation mechanism. A description of the phases of the Calibre CellArray RET flow follows the diagram.

Figure 1-2. Calibre CellArray RET Flow



Calibre CA-RET comprises Pattern Property Analysis and Annotation (PA2)
With SRAF Insertion and Cell-Array OPC.

Database Creation

Creation of the hierarchical database (HDB) adds information about unit cells, mask layers, standard cells, and critical areas (core array, periphery and controlling logic). Coverage of the full chip is ensured by collecting points of interest (POI) that cover the center of 2D via structures, and 1D metal critical corner and line-end structures. The HDB holds the context used for classification information of each POI by its interaction distance, essential to ensure output accuracy of lithographic simulation.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Layout Reduction

After HDB generation and POI pattern classification, Calibre CellArray RET checks the orientation of patterns and groups them according to symmetry about the X- or Y-axes. Symmetry classification is based on the symmetry of the optical model used. Symmetric optical models provide solutions with symmetric OPC and SRAF shapes. Additionally, symmetry reduces the number of patterns during layout reduction, ensuring geometric consistency.

RET Processing of the Reduced Layout

Generation of SRAFs is performed on the reduced layout. The SRAFs are then included in a model-based OPC simulation on the reduced layout to generate a mask of target patterns. The generation of SRAFs and model-based OPC on the reduced layout reduces runtime compared to traditional methodologies.

Full Chip Results Propagation and Stitching

The generated SRAF and OPC results are then propagated from the representative patterns of the reduced layout to the locations across the full chip. A methodology challenge regarding memory applications is that matching may occur where nested patterns or a common sub-pattern are incorrectly identified. Calibre CellArray RET uses the classified metadata for property-based matching to improve accuracy and performance.

Self Checker

The Calibre CellArray RET methodology involves multiple steps. To verify results during processing, a self-checker using pattern- and property-aware analysis ensures each part of the process adheres to the periodicity, symmetry and consistency requirements of the flow. This helps achieve superior quality of results for image-on-target, edge-placement errors (EPE), and process variation (PV) bands. The self-checker enables attributes of design for testability (DFT):

- **Accessibility —** The self-checker uses meta-data generated in the HDB and performs classification for SRAF and OPC results for further analysis based on the same POIs. The self-checker differentiates between information that exists on the reduced layout and full chip.
- **Controllability —** The self-checker uses controls for interaction distance, SRAF existence distance, and orientation control.
- **Observability —** The self-checker processes the classification and periodicity of meta-data, and aligns them statistically to eliminate false violations and flag issues that may occur during processing. Symmetry checks use the same POIs with a modified halo to check for local symmetry.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Calibre CellArray RET Processing Requirements

Appropriate system allocation and suitable memory yields optimum processing speeds.

Siemens EDA recommends running RET CELLARRAY with a minimum of 20 CPUs, each supporting 256 Gigabytes of RAM. System requirements vary with chip size.

Syntax Conventions

The command descriptions use font properties and several metacharacters to document the command syntax.

Table 1-1. Syntax Conventions

Convention	Description
Bold	Bold fonts indicate a required item.
Italic	Italic fonts indicate a user-supplied argument.
`Monospace`	Monospace fonts indicate a shell command, line of code, or URL. A bold monospace font identifies text you enter.
Underline	Underlining indicates either the default argument or the default value of an argument.
UPPercase	For certain case-insensitive commands, uppercase indicates the minimum keyword characters. In most cases, you may omit the lowercase letters and abbreviate the keyword.
[ ]	Brackets enclose optional arguments. Do not include the brackets when entering the command unless they are quoted.
{ }	Braces enclose arguments to show grouping. Do not include the braces when entering the command unless they are quoted.
‘ ’	Quotes enclose metacharacters that are to be entered literally. Do not include single quotes when entering braces or brackets in a command.
| or ||	Vertical bars indicate a choice between items. Do not include the bars when entering the command.
…	Three dots (an ellipsis) follows an argument or group of arguments that may appear more than once. Do not include the ellipsis when entering the command.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Table 1-1. Syntax Conventions (cont.)

Example:

DEVice {***element_name*** [‘(’*model_name*‘)’]}
device_layer {***pin_layer*** [‘(’*pin_name*‘)’] …}
[‘<’*auxiliary_layer*‘>’ …]
[‘(’*swap_list*‘)’ …]
[BY NET | BY SHAPE]

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Chapter 2
Getting Started with Calibre CellArray RET

Calibre RET CELLARRAY employs standard execution methods.

Running Calibre CellArray RET 15

Running Calibre CellArray RET

Calibre RET CELLARRAY is executed from the Linux command line.

Prerequisites

- A Calibre installation with RET CELLARRAY
- Optical and resist models for Calibre nmOPC
- SVRF files for Calibre CA-RET, cnSRAF, and nmOPC
- Your full chip design data

Procedure

1. Set the MGC_HOME environment variable:
   setenv MGC_HOME /data/calibre
2. Change directories to your working directory:
   cd work_dir
3. Execute Calibre RET CELLARRAY:
   $MGC_HOME/bin/calibre -drc -hier -turbo -turbo_litho -hyper rules.svrf

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Chapter 3
Calibre CellArray RET Command Reference

Calibre CellArray RET uses SVRF commands for optimal pattern-based processing of large, repetitive designs.

Pattern identification, SRAF and OPC generation, and results propagation are dependent on control parameters you specify.

SVRF Commands **18**
LITHO DENSEOPC CELLARRAY 19
LITHO FILE Parameters 21
RET CELLARRAY 25
RET SBAR 28

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

SVRF Commands

Calibre CellArray RET commands adhere to SVRF standards. A typical SVRF file containing RET CELLARRAY-related commands may also contain other SVRF commands that specify input layers, perform preliminary layer manipulation, and save output layers.

These commands are documented in the *Standard Verification Rule Format (SVRF) Manual*.

Table 3-1. SVRF Commands

Command	Description
LITHO DENSEOPC CELLARRAY	Derives output target layers for SVRF. Used by Calibre RET CELLARRAY for reduced pattern OPC.
LITHO FILE Parameters	Specifies Calibre RET CELLARRAY parameters within the SVRF file. The LITHO FILE statement containing RET CELLARRAY-related parameters must be referenced only by RET CELLARRAY.
RET CELLARRAY	The RET CELLARRAY command performs pattern classification, SRAF and OPC generation, and propagates results to all pattern instances.
RET SBAR	Invokes Calibre cnSRAF. Used by Calibre RET CELLARRAY for generation of reduced pattern SRAFs.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

LITHO DENSEOPC CELLARRAY

SVRF Commands

Derives output target layers for SVRF. Used by Calibre RET CELLARRAY for reduced pattern OPC.

Usage

output_layer_name **= LITHO DENSEOPC CELLARRAY** ***layer*** [*layer*…]
FILE {***filename*** | ***name*** | **'['**
inline_file
']'}
MAP ***name***

Arguments

- ***output_layer_name***

  A required keyword specifying the name of the new output target layer. The new layer is created and the results of the operation are written to the layer.

- **CELLARRAY**

  A required keyword invoking OPC for RET CELLARRAY applications.

- ***layer***

  A required original or derived polygon layer upon which to perform dense OPC operations. You can specify ***layer*** any number of times in one statement.

- **FILE** {***filename*** | ***name*** | **'['**

  inline_file

  ']'}

  A required keyword followed by one of three arguments:

  A ***filename*** indicating the path to the setup file. The ***filename*** parameter can contain environment variables.

  The ***name*** parameter of a LITHO FILE specification statement. The setup file is specified in the Litho File statement.

  An ***inline_file*** between brackets ([ ]). Characters within the brackets and on the same line as them are ignored. Inline files contain setup instructions for OPC and span multiple lines. The instructions may not use environment variables.

- **MAP** ***name***

  A required keyword followed by the name of a layer called by the setlayer command. The specified layer may then be mapped to the layer in the command file to be output.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Description

The LITHO DENSEOPC command is an argument of the Calibre LITHO batch operation. This command is used in conjunction with the setlayer denseopc command to derive output target layers for SVRF.

For detailed nmOPC information, see the *Calibre nmOPC User's and Reference Manual*.

Examples

The following example performs OPC on the patterns reduced within a design processed by RET CELLARRAY. Critical FILE, OPTIONS, and MAP terms are colored similarly for clarity.

```
// Generate OPC on the reduced layout
reduced_opc = LITHO DENSEOPC CELLARRAY reduced_layout reduced_sraf
   FILE CA_nmopc_setup MAP CA_opc_out

LITHO FILE CA_nmopc_setup [
   modelpath $env(DESIGN_DIR)/models/CAOPC/
   layer target
   layer sraf
   tilemicrons 30
   denseopc_options nmOPC.opt {
      version 1
      algorithm 2
      fragment_min 0.025
      fragment_max 0.100
      mrc_rule external target sraf {
         use 0.025 euclidean
      }
      NEWTAG external target target opposite -projecting > 0.06
         -out tag_space_lt_75 range < 0.075
      ...
      FRAGMENT_MOVE
      OPC_ITERATION 15
   }
   setlayer CA_opc_out = denseopc target sraf curvetarget
      MAP target OPTIONS nmOPC.opt
]
```


Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

LITHO FILE Parameters

SVRF Commands

Specifies Calibre RET CELLARRAY parameters within the SVRF file. The LITHO FILE statement containing RET CELLARRAY-related parameters must be referenced only by RET CELLARRAY.

Usage

LITHO FILE ***name*** '**[**'
parameters
'**]**'

Arguments

- ***name***

  A required name allowing parameters to be referenced by the LITHO DENSEOPC CELLARRAY command. Each LITHO FILE statement must have a unique name.

- ***parameters***

  A required group of parameters specified one per line. All parameters must be specified between the opening and closing brackets ([ ]). Any term on the same line as the brackets is ignored.

  Parameters may span multiple lines specified by a backslash (\), which must not be followed by any characters or whitespace.

  influence_range ***range***

  A required parameter that defines the halo size in microns for the reduction stage.

  layer ***name***

  A required parameter that defines the layers specified in the RET CELLARRAY command. One layer must be specified for reduction applications and three layers must be specified for propagation applications. Only manhattan- and 45-degree input layer geometries are supported. No skew-edged geometries are valid target shapes for SRAFs.

  sraf_radius ***radius***

  A required parameter that defines the SRAF radius in microns used in the reduction stage.

  caret_exec *cellarray_output* = reduce *target_layer*

  An optional sub-command that performs pattern reduction on the design. All keywords and layers must be specified.

  cellarray_output — The output layer specified to the RET CELLARRAY command.

  reduce — A required keyword that triggers pattern reduction.

  target_layer — A required input target layer.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

caret_exec *cellarray_output* = propagate *opc_layer sraf_layer target_layer* MAP *map_keyword*

An optional sub-command that performs artifact propagation on the design. All keywords and layers must be specified.

cellarray_output — The output layer specified to the RET CELLARRAY command.

propagate — A required keyword that triggers artifact propagation.

opc_layer — The reduced OPC output layer. This must be the first specified layer.

sraf_layer — The reduced SRAF output layer. This must be the second specified layer.

target_layer — The input target layer. This must be the third specified layer.

MAP *map_keyword* — A required keyword pair. The map_keywords are pre-defined, and must match the map_keywords specified by the caret_map sub-commands.

caret_map *cellarray_output* = MAP *map_keyword*

An optional sub-command that links the output from the caret_exec sub-commands to the corresponding output of the RET CELLARRAY command in SVRF. The cellarray_output is the output of the caret_exec sub-command. The map_keyword corresponds to the map_keyword specified in the RET CELLARRAY command. All keywords and layers must be specified.

cellarray_output — The output layer specified to the RET CELLARRAY command.

MAP *map_keyword* — A required keyword and mapping layer pair. The map_keywords are pre-defined, and must match those specified by the caret_exec sub-commands.

The MAP keywords are predetermined and trigger various functions for the RET CELLARRAY command:

Table 3-2. MAP Keywords

Stage	Map Keyword	Keyword Application
reduce	reduced	Reduced target output layer
propagate	propagated_opc	Propagated OPC results
	propagated_opc_bbox	Propagated OPC bounding box results
	restored_sraf	Propagated SRAF results

litho_model *path*

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

An optional parameter that sets the path of the Lithomodel file relative to the directory specified by modelpath. You must specify modelpath and litho_model together:

```
modelpath ./models
litho_model .
```

modelpath *path*

An optional parameter that specifies an absolute or relative path to a directory containing optical and resist models. RET CELLARRAY checks the optical models defined in the Lithomodel file, and detects the source symmetry. Possible symmetries detected are no symmetry, or X- and Y-reflection.

The modelpath and litho_model parameters must be specified together and are mutually exclusive with the orientation_flags parameter. Where either X- or Y-symmetry reflection are required for patterns of the reduced layout (apart from the symmetry declared by the optical model or the optical kernels), you must specify orientation_flags with one of either xsym or ysym arguments, and not modelpath or litho_model.

Paths may be absolute or relative:

```
modelpath /Siemens/models/CARET/contact/models

modelpath ../models/CARET/contact/models
```

Environment variables may be specified within the path (csh shown):

```
setenv MODELS_DIR /Siemens/models
...

modelpath $env(DESIGN_DIR)/CARET/contact/models
```

orientation_flags {off | on | xsym | ysym}

An optional parameter that applies pattern rotation and reflection during layout reduction and artifact propagation stages. By default, orientation_flags is off, accepting the symmetry established by the optical model or optical kernels. The orientation_flags parameter is mutually exclusive with modelpath and litho_model.

off — Pattern rotation and reflection is disabled. This is the default setting.

on — Enables pattern rotation and reflection for X- and Y-axes.

xsym — Explicitly enables symmetric reflection of patterns across the Y-axis only.

ysym — Explicitly enables symmetric reflection of patterns across the X-axis only.

Description

The LITHO FILE statement contains RET CELLARRAY parameters. The LITHO FILE statement is called by the FILE argument of the RET CELLARRAY command. The LITHO FILE statement can be written within the SVRF file or referenced as an external file, and in either case, called by a symbolic name for multiple instantiations. The LITHO FILE

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

statement for RET CELLARRAY cannot share parameters with the LITHO FILE statements used by RET SBAR or LITHO DENSEOPC.

Examples

Example 1 — Layout Reduction

This example demonstrates the general format of the Calibre CellArray RET LITHO FILE statement for layout reduction.

```
LITHO FILE caret_reduce [
    layer target
    sraf_radius 0.3
    influence_range 1.0
    orientation_flags on
    small_distance 0.002
    target_space 0.03
    filter_size 3.0
    restore_sraf_halo 0.0015
    caret_exec x = reduce target
    caret_map x MAP reduced
]
```

Example 2 — Artifact Propagation

This example demonstrates the general format of the Calibre CellArray RET LITHO FILE statement for SRAF and OPC propagation.

```
LITHO FILE caret_propagate [
    layer reduced_opc
    layer reduced_sraf
    layer target
    sraf_radius 0.3
    influence_range 1.0
    small_distance 0.002
    target_space 0.03
    filter_size 3.0
    restore_sraf_halo 0.0015
    caret_exec x = propagate reduced_opc reduced_sraf
      MAP propagated_opc
    caret_exec y = propagate reduced_opc reduced_sraf
      MAP propagated_opc_bbox
    caret_exec z = propagate reduced_opc reduced_sraf
      MAP propagated_sraf
    caret_map x MAP propagated_opc
    caret_map y MAP propagated_opc_bbox
    caret_map z MAP propagated_sraf
]
```


Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

RET CELLARRAY

The RET CELLARRAY command performs pattern classification, SRAF and OPC generation, and propagates results to all pattern instances.

Usage

output_layer = **RET CELLARRAY**
target_layer [*target_layer* ...]
FILE {***filename*** | ***name***}
MAP ***map_name***

Arguments

- ***output_layer***

  A required output layer name. The results of the RET CELLARRAY command are written to the output_layer.

- ***target_layer***

  One or more required target_layers consisting of target shapes. The target_layers must be specified as the first arguments to RET CELLARRAY. The specified target_layers are order-dependent.

- **FILE** {***filename*** | ***name***}

  A required keyword with one of two arguments:

  filename

  Indicates the path to the LITHO FILE statement contained within an external file. The filename parameter can contain environment variables. For information regarding the use of environment variables in the filename parameter, refer to “Environment Variables in Pathname Parameters” in the *Standard Verification Rule Format (SVRF) Manual*.

  name

  The name of a LITHO FILE statement called by RET CELLARRAY.

- **MAP** ***map_keyword***

  A required keyword pair specifying the map_keyword triggering a RET CELLARRAY function that is mapped to the RET CELLARRAY output layer. Four MAP keywords are provided:

  reduced

  Triggers layout reduction.

  propagated_opc

  Triggers propagation of OPC results to the pattern instances within the full chip.

  propagated_opc_bbox

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Triggers propagation of OPC bounding boxes to the pattern instances within the full chip.

propagated_sraf

Triggers propagation of SRAFs to the pattern instances within the full chip.

Description

Calibre RET CELLARRAY comprises various tools which perform pattern classification, SRAF and OPC generation on unique patterns identified within the reduced layout, and subsequently propagates results to all pattern instances within the full chip.

RET CELLARRAY performs three stages of execution. RET CELLARRAY executes stages 1 and 3. The user must script SVRF commands for OPC and SRAF generation to complete stage 2:

1. Pattern reduction performed on the target data. A reduced layout is generated.
2. OPC and SRAF generation on the reduced layout.
3. Propagation of the SRAF and OPC results to the pattern instances comprising the full chip.

Additionally, RET CELLARRAY is compatible with hyper flex processing.

Examples

Example 1 — Layout Reduction

To perform pattern identification and layout reduction, the MAP keyword ‘reduced’ must be specified.

```
reduced_layout = RET CELLARRAY target
   FILE setup MAP reduced
reduced_layout { COPY reduced_layout } DRC CHECK MAP reduced_layout 15

LITHO FILE setup [
   layer target
   caret_exec x = reduce target
   caret_map x MAP reduced
]
```

Example 2 — OPC and SRAF Propagation

To perform the propagation of OPC and SRAFs identification and layout reduction, the MAP keywords propagated_opc, propagated_opc_bbox, and propagated_sraf must be specified.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

```
restored_opc = RET CELLARRAY reduced_opc reduced_sraf target
   FILE setup MAP propagated_opc
restored_opc_bbox = RET CELLARRAY reduced_opc reduced_sraf target
   FILE setup MAP propagated_opc_bbox
restored_sraf = RET CELLARRAY reduced_opc reduced_sraf target
   FILE setup MAP propagated_sraf

LITHO FILE setup [
   layer reduced_opc
   layer reduced_sraf
   layer target
   caret_exec x = propagate reduced_opc reduced_sraf
      MAP propagated_opc
   caret_exec y = propagate reduced_opc reduced_sraf
      MAP propagated_opc_bbox
   caret_exec z = propagate reduced_opc reduced_sraf
      MAP propagated_sraf
   caret_map x MAP propagated_opc
   caret_map y MAP propagated_opc_bbox
   caret_map z MAP propagated_sraf
]
```


Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

RET SBAR

SVRF Commands

Invokes Calibre cnSRAF. Used by Calibre RET CELLARRAY for generation of reduced pattern SRAFs.

Usage

output_layer = **RET SBAR**
target_layer
[INSIDE OF LAYER *enclosing_layer*]
[*offset_layer*]
[*sb_layer* [*sb_layer2*]]
[*tag_layer*]
FILE {***filename*** | ***name***}
MAP *name*

Arguments

- ***output_layer***

  A required output layer name. The results of the RET SBAR command are written to this layer.

- ***target_layer***

  A required input layer consisting of target shapes. The target_layer must be specified as the first argument to RET SBAR and only one target_layer may be specified.

- INSIDE OF LAYER *enclosing_layer*

  An optional keyword specifying a layer of shapes that enclose target shapes for which SRAFs are generated. Generated SRAF shapes intersecting the enclosing_layer are truncated.

- *offset_layer*

  An optional polygon layer containing preexisting polygons that define areas where scattering bars cannot be created. The target layer should never be used as the offset_layer. Only one offset_layer may be specified.

- *sb_layer* [*sb_layer2*]

  An optional polygon layer containing preexisting scattering bars to be factored into calculations when generating additional scattering bars. There is no default for this layer setting. If the sb_layer name is specified once, it may contain both positive and negative SRAFs. Alternatively, the sb_layer may be specified twice with no order dependency, one layer for positive SRAF and one layer for negative SRAF.

- *tag_layer*

  An optional polygon layer specifying a tag name representing imported layers or edges from SVRF. Any number of tag_layers may be specified.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

- **FILE** {***filename*** | ***name***}

  A required keyword with one of the following arguments:

  filename — Indicates the path to the LITHO FILE statement contained within an external file. The filename parameter can contain environment variables. For information regarding the use of environment variables in the filename parameter, refer to "Environment Variables in Pathname Parameters" in the *Standard Verification Rule Format (SVRF) Manual*.

  name — The name of a LITHO FILE statement.

- **MAP** ***name***

  A required keyword and layer pair specifying the name of a layer that is mapped to the RET SBAR output layer. The layer name must be generated by the setlayer command within the LITHO FILE statement that is called by RET SBAR.

Description

The RET SBAR command inputs a target layer and templates specified in the LITHO FILE statement to output a new layer containing SRAFs. This command can be specified any number of times in the recipe.

For detailed RET SBAR information, see the *Calibre nmSRAF User's and Reference Manual*.

Examples

Example 1 — LITHO FILE Organization

When the RET SBAR command is compiled, the SVRF file is searched for the LITHO FILE statement named cnsraf_LF. This search is case-insensitive. The cnsraf_LF LITHO FILE statement is used if found within the SVRF file. If it is not found, cnsraf_LF is searched for as an external file on disk. Critical FILE, OPTIONS, and MAP terms are colored similarly for clarity.

```
sraf_cnsraf = RET SBAR sraf_tgt sraf_OL sraf_tag_p1 sraf_tag_p2
   FILE sraf_setup MAP out_layer

LITHO FILE sraf_setup [
   // setup commands ...
   layer target
   tilemicrons 20
```


Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

```
    sraf_options sraf_opts {
       cleanupversion 1
       minoffset 0.025
       ...
       taglayer sraf_tag_p1
       taglayer sraf_tag_p2
       ...
       rectedge tag sraf_tag_p1
          extension -0.008 width 0.025 offset 0.050 prior 100
       rectedge tag sraf_tag_p2
          extension 0.053 width 0.029 offset 0.053 prior 110
    }
    setlayer out_layer = cnsraf sraf_tgt OPTIONS sraf_opts
]
```


Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Appendix A
Calibre CellArray RET Example SVRF Files

An SVRF file is an ASCII text file comprising SVRF commands. It contains everything needed to read layout data, process layer data, and apply Calibre CellArray RET (CA-RET) solutions.

You can create a new SVRF file from scratch using a text editor, or by copying an existing SVRF file and modifying it.

This section describes basic CA-RET SVRF files. It is intended to provide enough information for the creation of a simple SVRF file. For more information, refer to the *Standard Verification Rule Format (SVRF) Manual*.

All elements in an SVRF file are either operations or statements. The basic distinction between these is that operations manipulate layout data and statements prepare the environment in which the operations work.

Calibre CellArray RET SVRF File **31**

Calibre CellArray RET SVRF File

Use this example RET CellArray SVRF file to help develop your own.

```
LAYOUT PATH "./input.oas"
LAYOUT SYSTEM OASIS
LAYOUT PRIMARY "*"
DRC SUMMARY REPORT "./result.rep"
DRC RESULTS DATABASE "./result.oas" OASIS
FLAG OFFGRID YES
LAYOUT MAGNIFY AUTO
LAYOUT PRECISION 4000
PRECISION 4000
RESOLUTION 1

// Layer declaration
Layer target 1
Layer design 2
LAYOUT BASE LAYER design

// Reduction
reduced_layout = RET CELLARRAY target
   FILE caret_reduce MAP reduced
```


Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

```
// Propagation and restoration
restored_opc = RET CELLARRAY reduced_opc reduced_sraf target
   FILE caret_propagate MAP propagated_opc
restored_sraf = RET CELLARRAY reduced_opc reduced_sraf target
   FILE caret_propagate MAP propagated_sraf

LITHO FILE caret_reduce [
    layer target
    sraf_radius 0.3
    influence_range 1.0
    modelpath /user/T1/CA-RET/models
    litho_model .
    small_distance 0.002
    target_space 0.03
    filter_size 3.0
    restore_sraf_halo 0.0015
    caret_exec x = reduce target
    caret_map x map reduced
]
LITHO FILE caret_propagate [
    influence_range 1.0
    layer reduced_sraf
    layer target
    layer reduced_opc
    sraf_radius 0.3
    orientation_flags on
    small_distance 0.002
    target_space 0.03
    filter_size 3.0
    restore_sraf_halo 0.0015
    caret_exec x = propagate reduced_opc reduced_sraf
      MAP propagated_opc
    caret_exec y = propagate reduced_opc reduced_sraf
      MAP propagated_opc_bbox
    caret_exec z = propagate reduced_opc reduced_sraf
      MAP propagated_sraf
    caret_map x MAP propagated_opc
    caret_map y MAP propagated_opc_bbox
    caret_map z MAP propagated_sraf
]

// Declare layers for SRAF and OPC generation
sraf_OL = area target < 0.000001
sraf_pl = DFM PROPERTY target GLOBALXY [len=PERIMXP(target)/2] > 0.300
sraf_tag_1 = LENGTH (target COINCIDENT EDGE sraf_pl) > 0.080
sraf_tag_2 = LENGTH (sraf_tag_1 COINCIDENT EDGE sraf_pl) > 0.080
```


Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

```
// Generate SRAFs
sraf_cnsraf = RET SBAR target sraf_OL sraf_tag_1 sraf_tag_2
   FILE sraf_setup MAP sraf_cnsraf
// Litho FILE parameters
LITHO FILE sraf_setup [
   layer target
   layer sraf_OL
   layer sraf_tag_1
   layer sraf_tag_2
   tilemicrons 20
   # cnSRAF options and templates
   sraf_options srafOpts {
      offsetlayer sraf_OL
      # global cleanup settings
      cleanmode postSizing
      cleanupversion 1
      conflictmode bbox
      dynamicsizing 0.025
      minoffset 0.025
      # global mrc settings
      dynamicsizingrunlength 0
      maxconflict 0.500
      minwidth 0.015
      minarea 0.001
      minfeaturespace 0.050
      minlength 0.055
      taglayer sraf_tag_1
      taglayer sraf_tag_2

      rectedge tag sraf_tag_1 extension 0.092
         width 0.029 offset 0.227 prior 30
      rectedge tag sraf_tag_2 extension 0.092
         width 0.029 offset 0.227 prior 36
   }
   setlayer sraf_cnsraf = cnsraf target OPTIONS srafOpts
]

// Save SRAFs to the reduced layout
reduced_sraf = COPY sraf_cnsraf
```


Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

```
//Perform OPC on the reduced layout with reduced srafs
reduced_opc = LITHO DENSEOPC CELLARRAY reduced_layout reduced_sraf
   FILE RL_nmopc_setup MAP RL_opc_out
reduced_frag = LITHO DENSEOPC CELLARRAY reduced_layout reduced_sraf
   FILE RL_nmopc_setup MAP RL_frag_out
reduced_tile = LITHO DENSEOPC CELLARRAY reduced_layout reduced_sraf
   FILE RL_nmopc_setup MAP RL_tile_out
curvetarget = LITHO DENSEOPC CELLARRAY reduced_layout reduced_sraf
   FILE RL_nmopc_setup MAP curvetarget

LITHO FILE RL_nmopc_setup [
  modelpath /user/T1/CA-RET/models
  layer  target
  layer  sraf
  tilemicrons 30
  denseopc_options nmOPC.opt {
    version 1
    algorithm 2
    mrc_mode 2
    max_iter_movement 0.003
    max_opc_move 0.03
    opc_grid_multiplier on
    preclean off
    layer target opc mask_layer 0
    layer curvetarget hidden
    fragment_min 0.025
    fragment_max 0.100
    image .
    mrc_rule external target {
     use 0.025 euclidean
    }
    NEWTAG all target -out T1
   }
   setlayer curvetarget = curve_target target criticalDistance 0.050
      cornerRadius 0.0375 lineEndRatio 0.5
   setlayer RL_opc_out = denseopc target sraf curvetarget
      MAP target OPTIONS nmOPC.opt
]


//OPC Output of reduced layout
opc_final = COPY reduced_opc
```


Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Index

— Symbols —

[], 13
{}, 13
\, 23
|, 13

— B —

Backslash (\), 23
Bold words, 13

— C —

Command reference, 13
Courier font, 13

— D —

Double pipes, 13

— H —

Heavy font, 13

— I —

Italic font, 13

— L —

Line continuation, 23
LITHO DENSEOPC, 20
LITHO FILE command, 21

— M —

Minimum keyword, 13

— P —

Parentheses, 13
Pipes, 13

— Q —

Quotation marks, 13

— R —

RET SBAR, 28
Rule files
    creating, 31
    definition and contents, 31
    using, 31

— S —

SBAR, 28
Slanted words, 13
Square parentheses, 13
SVRF
    LITHO FILE, 21

— U —

Underlined words, 13

Third-Party Information

Details on open source and third-party software that may be included with this product are available in the *<your_software_installation_location>/legal* directory.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.