

# **A Quick Start to Calibre<sup>®</sup> Tools**

## **Validated with 2019.2**

May 2019

---

**© 2010-2019 Mentor Graphics Corporation**  
**All rights reserved.**

This document contains information that is proprietary to Mentor Graphics Corporation. The original recipient of this document may duplicate this document in whole or in part for internal business purposes only, provided that this entire notice appears in all copies. In duplicating any part of this document, the recipient agrees to make every reasonable effort to prevent the unauthorized use and distribution of the proprietary information.

This document is for information and instruction purposes. Mentor Graphics 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 Mentor Graphics to determine whether any changes have been made.

The terms and conditions governing the sale and licensing of Mentor Graphics products are set forth in written agreements between Mentor Graphics and its customers. 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 Mentor Graphics whatsoever.

MENTOR GRAPHICS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MENTOR GRAPHICS SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF MENTOR GRAPHICS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

**U.S. GOVERNMENT LICENSE RIGHTS:** The software and documentation were developed entirely at private expense and are commercial computer software and commercial computer software documentation within the meaning of the applicable acquisition regulations. Accordingly, pursuant to FAR 48 CFR 12.212 and DFARS 48 CFR 227.7202, use, duplication and disclosure by or for the U.S. Government or a U.S. Government subcontractor is subject solely to the terms and conditions set forth in the license agreement provided with the software, except for provisions which are contrary to applicable mandatory federal laws.

**TRADEMARKS:** The trademarks, logos and service marks ("Marks") used herein are the property of Mentor Graphics Corporation or other parties. No one is permitted to use these Marks without the prior written consent of Mentor Graphics or the owner of the Mark, as applicable. The use herein of a third-party Mark is not an attempt to indicate Mentor Graphics as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A current list of Mentor Graphics' trademarks may be viewed at: [mentor.com/trademarks](http://mentor.com/trademarks).

The registered trademark Linux<sup>®</sup> is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

**End-User License Agreement:** You can print a copy of the End-User License Agreement from: [mentor.com/eula](http://mentor.com/eula).

Mentor Graphics Corporation  
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777  
Telephone: 503.685.7000  
Toll-Free Telephone: 800.592.2210  
Website: [mentor.com](http://mentor.com)  
Support Center: [support.mentor.com](http://support.mentor.com)

Send Feedback on Documentation: [support.mentor.com/doc\\_feedback\\_form](http://support.mentor.com/doc_feedback_form)

# Table of Contents

---

<b>Chapter 1</b>	
<b>Getting Started</b>	<b>9</b>
IC Design Flow Using Calibre Tools	9
About the Quick Start Guide and Example Design	10
Installing the Quick Start Example Data	11
Global Prerequisites	13
<b>Chapter 2</b>	
<b>Performing Physical Verification</b>	<b>15</b>
Physical Verification Design Flow	15
Calibre Physical Verification Tool Reference	16
Performing Design Rule Checks with Calibre nmDRC-H	21
Reviewing DRC Results	24
Performing Layout Versus Schematic Comparison with Calibre nmLVS-H	27
Reviewing LVS Results	31
<b>Chapter 3</b>	
<b>Designing for Manufacturability</b>	<b>37</b>
Design for Manufacturability Design Flow	37
Calibre Design for Manufacturability Tool Reference	38
Performing a Critical Area Analysis Using Calibre YieldAnalyzer	41
Performing a CMP Hotspot Analysis Using Calibre CMPAnalyzer	46
Performing an LFD Analysis	52
<b>Chapter 4</b>	
<b>Extracting Parasitics</b>	<b>59</b>
Parasitic Extraction Design Flow	59
Calibre Parasitic Extraction (PEX) Tool Reference	60
Extracting Parasitics for Digital Designs Using Calibre xACT	62
Calculating Parasitics Using Calibre xRC	67
<b>Chapter 5</b>	
<b>Performing RET/OPC</b>	<b>73</b>
RET/OPC Design Flow	73
Calibre Resolution Enhancement Technology (RET) Tool Reference	74
Sparse OPC and Dense OPC Process Flows	77
Performing Sparse OPC	79
The VT5 Resist Modeling Flow	79
Performing OPC and Simulation Using Calibre OPCpro	81
Performing Sparse OPC Verification Using Calibre OPCverify	86
Performing Dense OPC	87

---

CM1 and VEB Resist Modeling Flow .....	87
Performing OPC and Simulation Using Calibre nmOPC .....	89
Performing OPC Verification Using Calibre OPCverify .....	95
<b>Chapter 6</b>	
<b>Preparing Mask Data .....</b>	<b>99</b>
Mask Data Preparation Design Flow .....	99
Calibre Mask Data Preparation (MDP) Tool Reference .....	100
Generating a Pattern File Using Calibre FRACTURE .....	105
Evaluating Fractured Data Using Calibre MDPverify .....	107
<b>Chapter 7</b>	
<b>Performing Electrostatic Discharge Checks with Calibre PERC .....</b>	<b>111</b>
ESD Checks in the Design Flow .....	111
Performing Electrostatic Discharge Checks Using Calibre PERC .....	113
Reviewing ESD Results .....	117
<b>Third-Party Information</b>	
<b>End-User License Agreement</b>	

# List of Figures

---

Figure 1-1. IC Design and Manufacturing Design Flow with Calibre .....	10
Figure 2-1. Physical Verification Design Flow .....	16
Figure 2-2. Viewing Rule Check Details in Calibre RVE.....	25
Figure 2-3. Viewing DRC Errors in Calibre RVE.....	25
Figure 2-4. Selecting Columns to Display in the Calibre RVE Result View .....	26
Figure 2-5. Setting DRC Browser Options in Calibre RVE .....	26
Figure 2-6. Viewing LVS Discrepancies in Calibre RVE.....	33
Figure 2-7. Viewing the Layout and Source Netlists in Calibre RVE.....	34
Figure 3-1. Design for Manufacturability Flow.....	38
Figure 3-2. CAA Settings Tab Calibre Interactive .....	43
Figure 3-3. CAA Analysis Tab Calibre Interactive.....	44
Figure 3-4. CAA Chip Summary Tab Calibre RVE .....	45
Figure 3-5. CMP Tab Calibre RVE.....	50
Figure 3-6. Viewing LFD PV Bands in Calibre DESIGNrev .....	54
Figure 3-7. LFD Spacing Hotspot .....	55
Figure 3-8. Selecting Columns in Calibre RVE to Display LFD Results .....	56
Figure 3-9. Displaying LFD PVI Results Column in Calibre RVE.....	56
Figure 3-10. Grouping by Cell/Check in Calibre RVE .....	57
Figure 3-11. Displaying LFD DVI Results Column in Calibre RVE .....	57
Figure 4-1. Parasitic Extraction Design Flow .....	60
Figure 5-1. RET/OPC Design Flow.....	74
Figure 5-2. OPC Process Flow .....	78
Figure 5-3. OPCpro Input and Output .....	85
Figure 5-4. Original Design Versus OPC Output .....	93
Figure 5-5. Original Design Versus Printed Image .....	95
Figure 6-1. Mask Data Preparation Design Flow .....	100
Figure 6-2. Fracture Results.....	107
Figure 7-1. ESD Verification in the Design Flow.....	112
Figure 7-2. Viewing ESD Errors in Calibre RVE .....	118



## List of Tables

---

Table 1-1. Calibre Tools Used in the Quick Start Design Flow .....	11
Table 1-2. Files in <i>/Shared/Design</i> Directory .....	12
Table 2-1. Calibre Physical Verification Tool Reference .....	17
Table 3-1. Calibre Design for Manufacturability Tool Reference .....	39
Table 4-1. Calibre Parasitic Extraction Tool Reference .....	61
Table 5-1. Calibre Resolution Enhancement Technology Tool Reference .....	75
Table 6-1. Calibre Mask Data Preparation Tool Reference .....	101





# Chapter 1

## Getting Started

---

To get started with the Calibre Quick Start Guide and the accompanying data, review the following topics.

---

### Note



This version of the Calibre Quick Start was validated with the 2019.2 release of Calibre.

---

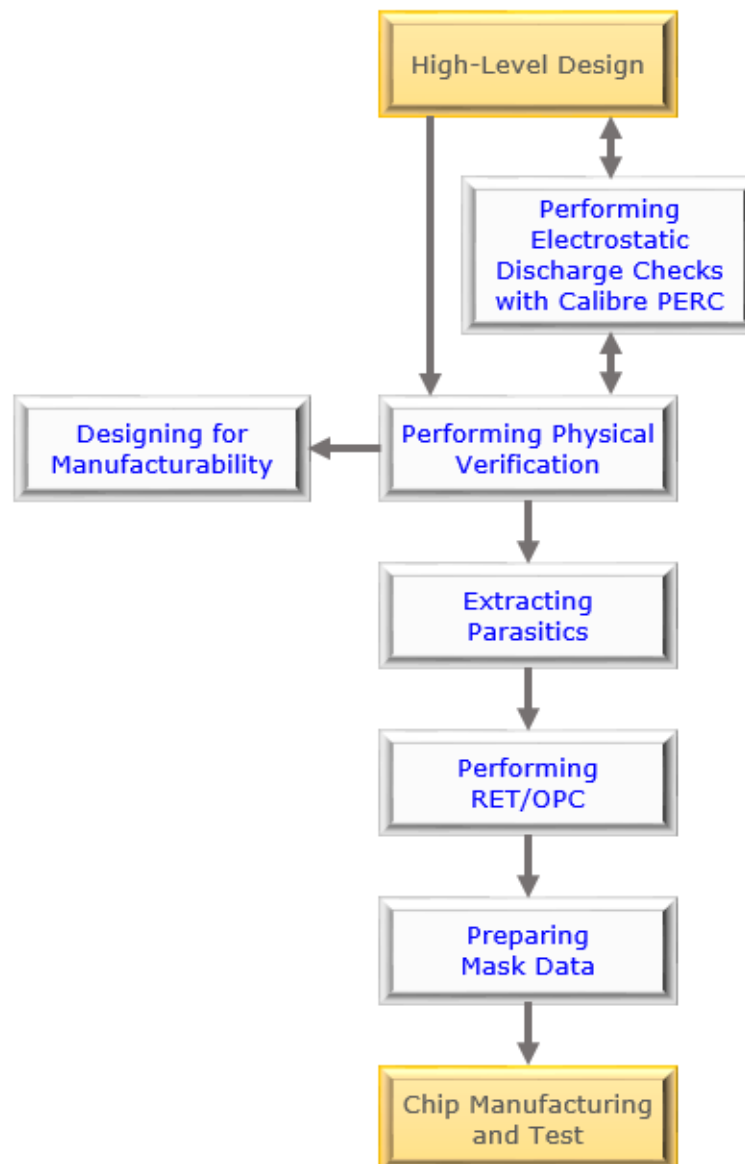
<b>IC Design Flow Using Calibre Tools .....</b>	<b>9</b>
<b>About the Quick Start Guide and Example Design .....</b>	<b>10</b>
<b>Installing the Quick Start Example Data .....</b>	<b>11</b>
<b>Global Prerequisites .....</b>	<b>13</b>

## IC Design Flow Using Calibre Tools

While design flows can vary, the primary tasks performed and the tools used are the same.

[Figure 1-1](#) illustrates a high-level sequence of steps required to complete a chip design process from concept to production, also referred to as a *design flow*. Each step in the figure links to a module containing more detailed information about the design flow, in addition to supporting examples.

**Figure 1-1. IC Design and Manufacturing Design Flow with Calibre**



## About the Quick Start Guide and Example Design

The purpose of this document and the example design data is to introduce you to Calibre tools and to demonstrate how these tools fit into an IC design flow.

An example flow is shown in [Figure 1-1](#). The examples are designed as a quick introduction to performing some of the basic functions found in the Calibre tools. Each module in this guide maps to a step in the IC design flow and covers the Calibre tools listed in [Table 1-1](#).

**Table 1-1. Calibre Tools Used in the Quick Start Design Flow**

Module	Design Flow Step	Calibre Tools Used
2	Performing Physical Verification	Calibre® nmDRC™, Calibre® nmLVS™, Calibre® RVE™
3	Designing for Manufacturability	Calibre® YieldAnalyzer, Calibre® CMPAnalyzer, Calibre® DESIGNrev™, Calibre® LFD™
4	Extracting Parasitics	Calibre® xACT SOC, Calibre® xRC™
5	Performing RET/OPC	Calibre® OPCpro™, Calibre® OPCverify™, Calibre® nmOPC™, Calibre® WORKbench™
6	Preparing Mask Data	Calibre® FRACTURE, Calibre® MDPverify™
7	Performing Electrostatic Discharge Checks with Calibre PERC	Calibre® PERC™, Calibre® RVE™

## Example Design Data

The design data package is for a signal conditioning IC developed by Oklahoma State University. This design is based on the 45nm FreePDK technology developed by North Carolina State University (<http://www.eda.ncsu.edu/wiki/FreePDK>). The contents of this package are described in “[Installing the Quick Start Example Data](#).”

# Installing the Quick Start Example Data

Before getting started with the examples used in *A Quick Start to Calibre Tools*, you must install the data.

## Prerequisites

- You have downloaded the *calbr\_quickstart\_ekit.zip* file containing both this document and the design data from Mentor Graphics Support Center. This file can be accessed from the [Documentation](#) tab (select the Getting Started Guide document type) or the [Calibre Utilities](#) page on Support Center.

## Procedure

- In a UNIX or Linux terminal window, browse to or create the directory from which you will run the examples.
- If you have not already done so, uncompress the *calbr\_quickstart\_ekit.zip* file. Otherwise, skip to Step 3.

Place the *calbr\_quickstart\_ekit.zip* file in the working directory and enter the following command to uncompress the file:

```
unzip calbr_quickstart_ekit.zip
```

3. Change directory to the example directory:

```
cd Calbr_qs_ekit
```

4. List the contents of the example directory:

```
% ls -al Calbr_qs_ekit
drwxrwxr-x    2 owner group    4096 Mar  4 14:19
  calibre_ekit_license_1.0.pdf
drwxrwxr-x    2 owner group    4096 Mar  4 14:19 Docs
drwxrwxr-x    2 owner group    4096 Mar  4 14:19 Module1
drwxrwxr-x    2 owner group    4096 Mar  4 14:19 Module2
drwxrwxr-x    4 owner group    4096 Mar  4 14:19 Module3
drwxrwxr-x    3 owner group    4096 Mar  4 14:19 Module4
drwxrwxr-x    4 owner group    4096 Mar  4 14:19 Module5
drwxrwxr-x    2 owner group    4096 Mar  4 14:19 Module6
drwxrwxr-x    2 owner group    4096 Mar  4 14:19 Module7
-rw-rw-r--    1 owner group    180 Feb 21 13:49 README
drwxrwxr-x    3 owner group    4096 Mar  4 14:19 Shared
```

These directories contain the following:

- *Docs* — Contains the user documentation, *A Quick Start to Calibre Tools*. The documentation is provided in PDF format.
- *Module[2-7]* — Contains data used to perform the examples in the corresponding Module of the user documentation. For example, the *Module2* directory contains files that are specific to performing the procedures in Module 2, “Performing Physical Verification.”
- *Shared* — Contains the IC design data and files that are used to perform the procedures. The */Shared/Design* directory includes the following files that are used in the examples:

**Table 1-2. Files in /Shared/Design Directory**

Design Files/Directories	Description	Calibre tool usage
<i>DEF</i>	Directory containing DEF design data file	xACT SOC
<i>drc.rules</i>	DRC rule file	nmDRC
<i>fullchip.oas</i>	OASIS layout file	nmDRC, nmLVS, YA, CMPAnalyzer, OPCpro, nmOPC, FRACTURE, MDPverify, xRC
<i>fullchip.oas.layerprops</i>	Layer properties file	YA, CMPAnalyzer

**Table 1-2. Files in /Shared/Design Directory (cont.)**

Design Files/Directories	Description	Calibre tool usage
<i>layer.inc</i>	Layer definition file	nmDRC, nmLVS, xACT SOC, xRC
<i>LEF</i>	Directory containing the LEF cell library, macro files, and technology file	xACT SOC
<i>lvs.rules</i>	LVS rule file	nmLVS, xRC
<i>README</i>	Describes contents of / <i>Shared/Design</i> directory	N/A
<i>source.net</i>	Source SPICE netlist	nmLVS, xRC

As you work through the examples in each module, you will set your working directory to the directory with the corresponding module name. The output generated from running the examples is output to the current working directory.

## Results

You have now installed and reviewed the Calbr\_qs\_ekit design data. Before starting the procedures, be sure to perform the items listed in the “[Global Prerequisites](#).”

# Global Prerequisites

All procedures in this document require certain items to be installed and properly configured.

These are:

- A Mentor Graphics software tree for the release specified on [page 9](#) must be installed and the \$CALIBRE\_HOME environment variable set to the location of this tree.
- The Calibre documentation must be installed and the appropriate variables must be set.
- Software licenses must be installed and license variables must be set.

Refer to the *Calibre Administrator's Guide* for configuration and licensing information.



# Chapter 2

## Performing Physical Verification

---

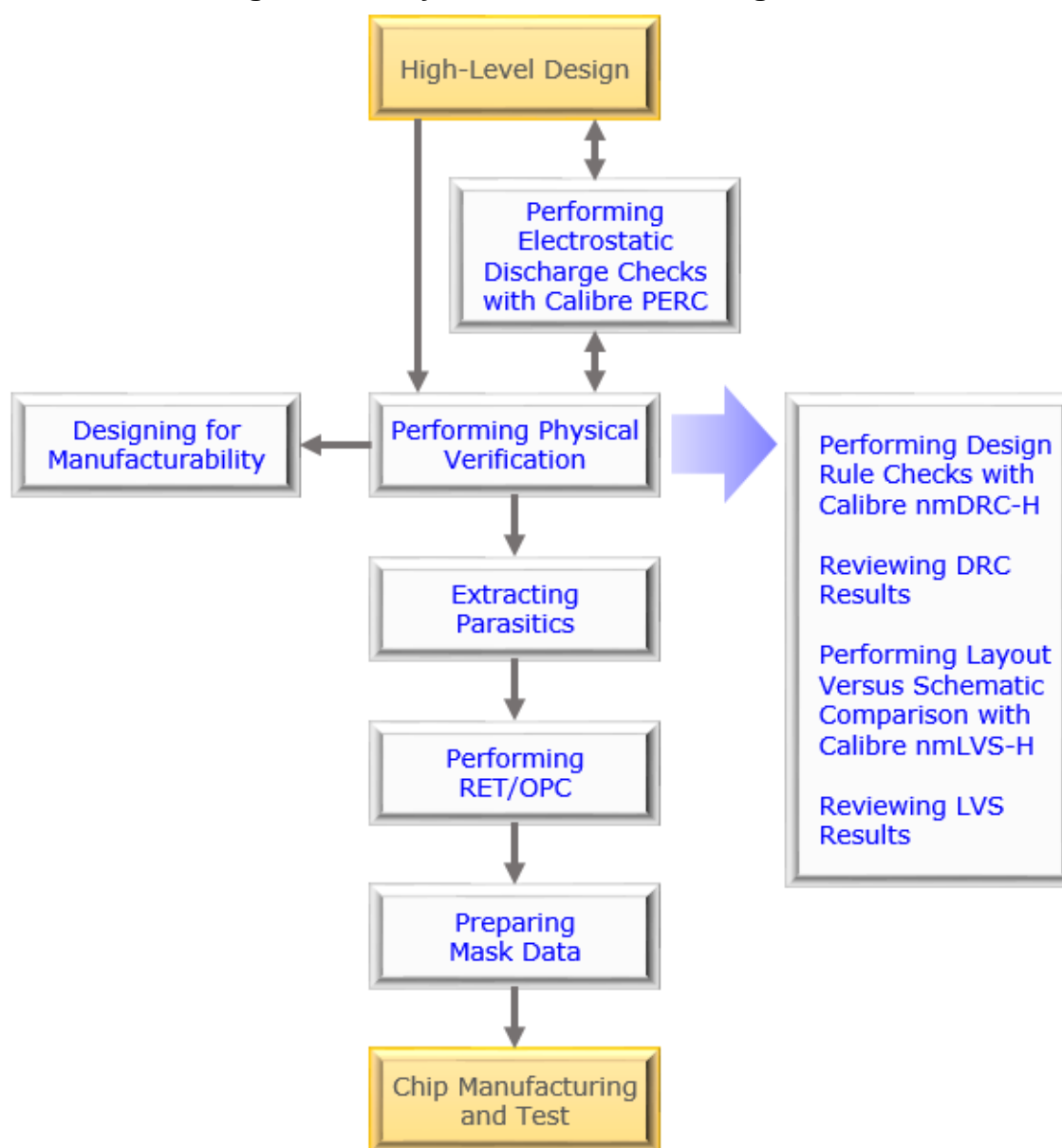
Physical Verification is a process whereby an IC layout design is checked by software tools to see if it meets certain physical design criteria. The verification process involves performing various checks.

<b>Physical Verification Design Flow .....</b>	<b>15</b>
<b>Calibre Physical Verification Tool Reference.....</b>	<b>16</b>
<b>Performing Design Rule Checks with Calibre nmDRC-H.....</b>	<b>21</b>
<b>Reviewing DRC Results.....</b>	<b>24</b>
<b>Performing Layout Versus Schematic Comparison with Calibre nmLVS-H .....</b>	<b>27</b>
<b>Reviewing LVS Results .....</b>	<b>31</b>

### Physical Verification Design Flow

Some of the possible physical verification checks are shown in Figure 2-1 and described in the procedures in this module. This design flow may involve performing electrical rule checks (ERC), short isolation, and antenna checks.

**Figure 2-1. Physical Verification Design Flow**



## Calibre Physical Verification Tool Reference

Supporting documentation is available for each licensed Calibre product and, in some cases, training is also available. There may be other products that are closely related to the use of that tool.

[Table 2-1](#) provides a comprehensive list and short description of the licensed Calibre products associated with performing physical verification, in addition to information on related documentation, training, and applications.



**Table 2-1. Calibre Physical Verification Tool Reference**

Tool	Description	Related Documentation	Related Training	Related Products
Calibre® 3DSTACK	A tool used to verify designs containing flip chips, TSVs, and other 2.5D and 3D-IC configurations.	<i>Calibre 3DSTACK Quick Reference</i> ----- <i>Calibre 3DSTACK User's Manual</i>		
Calibre® Auto-Waiver™	A tool used to manage waivers of rule check violations.	<i>Calibre Auto-Waiver User's and Reference Manual</i>		
Calibre® DESIGNrev™	A layout viewer used for full-chip design viewing and basic editing.	<i>Calibre DESIGNrev Key Definitions Quick Reference</i> ----- <i>Calibre DESIGNrev Layout Viewer User's Manual</i> ----- <i>Calibre DESIGNrev Reference Manual</i>	<a href="#">Calibre Fundamentals: DESIGNrev</a>	Calibre WORKbench ----- Calibre LITHOview ----- Calibre MDPview
Calibre® Interactive™	A GUI interface for DRC, LVS, PERC, PEX, and DFM. Also calls Calibre RVE.	<i>Calibre Interactive User's Manual</i>	<a href="#">Calibre Fundamentals: Performing DRC/LVS</a>	Incremental DRC ----- Calibre RVE

**Table 2-1. Calibre Physical Verification Tool Reference (cont.)**

<b>Tool</b>	<b>Description</b>	<b>Related Documentation</b>	<b>Related Training</b>	<b>Related Products</b>
Calibre® nmDRC/ Calibre nmDRC-H™	A tool used to perform design rule checking of integrated circuit layout designs.	<i>Calibre eqDRC Quick Reference</i> ----- <i>Calibre Layout Comparison and Translation Guide</i> ----- <i>Calibre Solutions for Physical Verification</i> ----- <i>Calibre Verification User's Manual</i> ----- <i>SVRF Manual</i>	<a href="#">Calibre Fundamentals: Performing DRC/ LVS</a> ----- <a href="#">Calibre Fundamentals: Writing DRC/ LVS Rules</a> ----- <a href="#">Calibre Advanced Topics: Mastering Calibre eqDRC</a>	Incremental DRC ----- eqDRC ----- Calibre Auto-Waiver ----- Fast XOR
Calibre® nmLVS/ Calibre nmLVS-H™	A tool used to perform layout-versus-schematic comparison and netlist extraction.	<i>Calibre Verification User's Manual</i> ----- <i>Calibre Solutions for Physical Verification</i> ----- <i>SVRF Manual</i>	<a href="#">Calibre Fundamentals: Performing DRC/ LVS</a> ----- <a href="#">Calibre Fundamentals: Writing DRC/ LVS Rules</a> ----- <a href="#">Calibre Advanced Topics: nmLVS Debug Case Studies</a>	Calibre PERC ----- Calibre Query Server

**Table 2-1. Calibre Physical Verification Tool Reference (cont.)**

Tool	Description	Related Documentation	Related Training	Related Products
Calibre® Pattern Matching	A tool used to locate specific polygons in a design from a pattern you specify.	<i>Calibre Pattern Matching Reference Manual</i> ----- <i>Calibre Pattern Matching User's Manual</i>	<a href="#">Calibre Pattern Matching</a>	
Calibre® PERC™	A tool used to perform programmable ERC and ESD checks.	<i>Calibre PERC User's Manual</i> ----- <i>Calibre Solutions for Physical Verification</i> ----- <i>SVRF Manual</i>	<a href="#">Calibre Fundamentals: Working with PERC</a> ----- <a href="#">Calibre Advanced Topics: Writing PERC Rules</a>	Calibre nmLVS/ Calibre nmLVS-H
Calibre® Query Server	A licensed database server used to examine the contents of the database generated by LVS.	<i>Calibre Query Server Manual</i>		Calibre nmLVS/ Calibre nmLVS-H

**Table 2-1. Calibre Physical Verification Tool Reference (cont.)**

<b>Tool</b>	<b>Description</b>	<b>Related Documentation</b>	<b>Related Training</b>	<b>Related Products</b>
Calibre® RealTime	Integrates flat Calibre nmDRC with supported layout design tools allowing Calibre nmDRC to be run directly from the design tool.	<i>Calibre RealTime Digital in Cadence Innovus Quick Reference</i> ----- <i>Calibre RealTime Digital User's Manual</i> ----- <i>Calibre RealTime Integration to Cadence Virtuoso Quick Reference</i> ----- <i>Calibre RealTime Integration to Synopsys Laker Quick Reference</i> ----- <i>Calibre RealTime User's Manual</i>		Calibre nmDRC
Calibre® RVE™	A GUI used to view and debug results output from Calibre tools.	<i>Calibre RVE User's Manual</i> ----- <i>Calibre RVE for DRC Quick Reference</i> ----- <i>Calibre RVE for LVS Quick Reference</i> ----- <i>Calibre RVE for PERC Quick Reference</i>	<a href="#">Calibre Fundamentals: Performing DRC/LVS</a>	Calibre Interactive

# Performing Design Rule Checks with Calibre nmDRC-H

A design has been placed and routed. In the following example, you run Calibre nmDRC-H to detect design rule errors.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in [“Installing the Quick Start Example Data”](#) and set up your environment according to the information in [“Global Prerequisites.”](#)
- Licenses for the following products: Calibre nmDRC-H.
- The following inputs are required to perform this example:
  - SVRF rule file, *Module2/drc.control*
  - GDS layout, *Shared/Design/fullchip.oas*

## Procedure

1. In a UNIX or Linux terminal window, change directory to *Calbr\_qs\_ekit/Module2*:

```
cd <path>/Calbr_qs_ekit/Module2
```

2. Open the DRC control file, *drc.control*, in an ASCII editor and notice the following SVRF rule file statements:

```
LAYOUT SYSTEM OASIS
LAYOUT PATH "../Shared/Design/fullchip.oas"
LAYOUT PRIMARY "top"

DRC SUMMARY REPORT drc.report HIER
DRC RESULTS DATABASE drc.db
```

All of these statements except DRC Summary Report are required to run DRC.

3. Notice the following two statements:

```
INCLUDE "../Shared/Design/layer.inc"

source "../Shared/Design/drc.rules"
```

These statements point to other rule files that are used for the run. The *layer.inc* file contains layer definitions and other layout-directed statements. The *drc.rules* file contains the DRC rule checks. Briefly review the contents of these files.

4. Close the *drc.control* file.
5. In the *Module2* working directory, enter the following command at a shell prompt:

```
$CALIBRE_HOME/bin/calibre -drc -hier -turbo drc.control |tee drc.log
```

The `-hier` argument specifies to perform design rule checking hierarchically, which reduces processing time, memory use, and DRC result counts compared to flat DRC applications. The `-turbo` argument, which specifies to use multithreaded, parallel processing, must be specified with the `-hier`. When `-turbo` is specified with no value, Calibre runs on the maximum number of CPUs available.

6. Open the *drc.log* file.

The header in the transcript includes the following information:

- The Calibre version used for the run:

```
// Calibre <version> <time stamp>
```

- The operating system, hostname, OS version, and build:

```
// Mentor Graphics software executing under <architecture>  
...  
// Running on <OS> <hostname> <version> <build>
```

- The path of the executable, the command line options, and the PID are shown on these lines:

```
// Running <version>/pkgs/icv/pvt/calibre -drc -hier -turbo  
drc.control  
// Process ID: <pid>
```

- The start time of the run and the number of CPUs used for making runtime calculations (those for which licenses are required) are shown on these lines:

```
// Starting time: <timestamp>  
//  
// Running on <N> CPU[s] (pending licensing)
```

- Calibre nmDRC automatically compiles the rule file at the beginning of the run. The transcript includes the pathname of the rule file, the contents of the rule file, and the amount of CPU and real time required for the compilation.

```
-----  
----- STANDARD VERIFICATION RULE FILE COMPILATION MODULE -----  
-----  
--- RULE FILE = drc.control  
...
```

- The “CALIBRE LAYOUT DATA INPUT MODULE” section in the transcript provides information on the layout system, magnification, and layout statistics.

```
...
--- LAYOUT SYSTEM = OASIS
--- LAYOUT MAGNIFICATION = 1
...
OASIS FILENAME:      ../Shared/Design/fullchip.oas
OASIS VERSION:       1.0
DATABASE PRECISION:  2000
MAGNIFICATION:       1
```

- During the run, rule checks are run mostly in the order they appear in the rule file. However, Calibre nmDRC locates all required layer operations that can run concurrently and executes them as a single group. Operations that are executed concurrently are grouped together in the transcript as shown here:

```
OX.2::<1> = EXT OX < 0.08 ABUT < 90 SINGULAR
OX.1::<1> = INT OX < 0.075 ABUT < 90 SINGULAR
-----
OX.2::<1> (HIER TYP=3 HGC=0 FGC=0 HEC=0 FEC=0)
OX.1::<1> (HIER TYP=3 HGC=0 FGC=0 HEC=0 FEC=0)
CPU TIME = 0   REAL TIME = 0   LVHEAP = 12/33/34   OPS COMPLETE = 49
OF 334   ELAPSED TIME = 11
```

Layers that are stored in memory are deleted from memory when they are no longer needed for the run.

- The tail of the transcript contains this summary information:

```
--- CALIBRE::DRC-H EXECUTIVE MODULE COMPLETED.   CPU TIME = 47
      REAL TIME = 27
--- TOTAL RULECHECKS EXECUTED = 152
--- TOTAL RESULTS GENERATED = 17 (192)
--- DRC RESULTS DATABASE FILE = drc.db (ASCII)

--- CALIBRE::DRC-H COMPLETED - <date>
--- TOTAL CPU TIME = 51   REAL TIME = 31
--- PROCESSOR COUNT = 2
--- SUMMARY REPORT FILE = drc.report
```

7. Close the *drc.log* file.
8. Open the *Module2/drc.report* file in an ASCII editor and review the results.

Calibre outputs a DRC Summary Report that includes the following information:

- **Header information** — The first part of the DRC Summary Report lists general information about the run.
- **Runtime Warnings** — This section lists any warnings that were generated during the DRC run. See “Runtime Messages” in the *SVRF Manual* for a listing.
- **Original Layer Statistics** — This section lists the original layers and the number of original shapes processed for that layer. The first number is a hierarchical count and the number in parentheses is an estimated flat count.

- **Rule Check Results Statistics** — This section lists the rule checks and the number of results generated. The DRC Summary Report also lists the rule checks that were not performed. Rule check results are shown per rule check (hierarchical count followed by estimated flat count).

9. Close the *drc.report* file.

## Results

You have now run Calibre nmDRC which performed design rule checking between the layout and the rule file. The outputs from Calibre nmDRC include:

- Run transcript (*drc.log*)
- Summary report (*drc.report*)
- Results database (*drc.db*)

If errors were found during the run, you would normally fix any errors and run DRC again. In the next procedure, you use Calibre RVE to review the results of the Calibre nmDRC run.

# Reviewing DRC Results

You can review the DRC Results Database using Calibre RVE.

Ordinarily, Calibre RVE is invoked either from a layout editor or from Calibre Interactive. If your tool environment is configured with Calibre RVE integrated into such tools, then call Calibre RVE from the appropriate menu items. In layout viewers like Calibre DESIGNrev, you can start Calibre RVE from the **Verification** menu. In third-party layout editors, you frequently start Calibre RVE from the Calibre menu. In this procedure, no layout editor is assumed to be configured, and Calibre Interactive is not assumed to be in your environment.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- Licenses for the following products: Calibre RVE.
- An ASCII DRC Results Database such as *drc.db* that was generated by performing the procedure “[Performing Design Rule Checks with Calibre nmDRC-H](#).”

## Procedure

1. Verify your working directory is set to *Calbr\_qs\_ekit/Module2*.
2. Invoke Calibre RVE using the *drc.db* file as input.

```
$CALIBRE_HOME/bin/calibre -rve drc.db
```



The tree view (left-hand pane) of the Calibre RVE interface shows the results. You can select a rule check to view the details of the check in the check text pane (lower pane) as shown in [Figure 2-2](#).

**Figure 2-2. Viewing Rule Check Details in Calibre RVE**

Check / Cell /	Results	Flat
⊕ × Check OX.3	2	176
⊕ × Check NW.2	1	2
⊕ × Check M6.3	2	2
⊕ × Check M5.3	12	12

Rule File Pathname: drc.control  
Min NW enclosure of OX = 0.025

3. Select a check in the tree view and a result to view in the detailed result view (right-hand pane) as shown in [Figure 2-3](#).

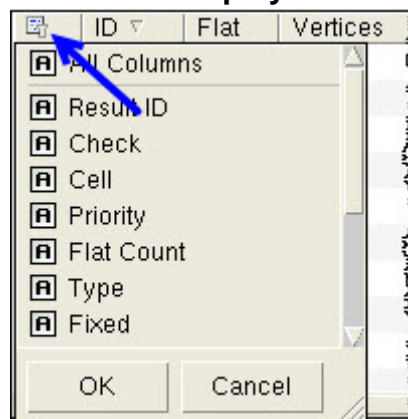
**Figure 2-3. Viewing DRC Errors in Calibre RVE**

Check / Cell /	Results	Flat	Flat	Vertices
⊕ × Check OX.3	2	176	1	4
⊕ × Check NW.2	1	2	2	4
⊕ × Check M6.3	2	2	3	4
⊕ × Check M5.3	12	12	4	4
			5	4
			6	4
			7	4
			8	4
			9	4
			10	4
			11	4
			12	4

4) Check M5.3, Cell top: 2 Edges  
2 Edges. Coordinates in cell top  
(138.3450 336.8950) (138.3450 336.8975)  
(138.3425 336.8950) (138.3450 336.8950)

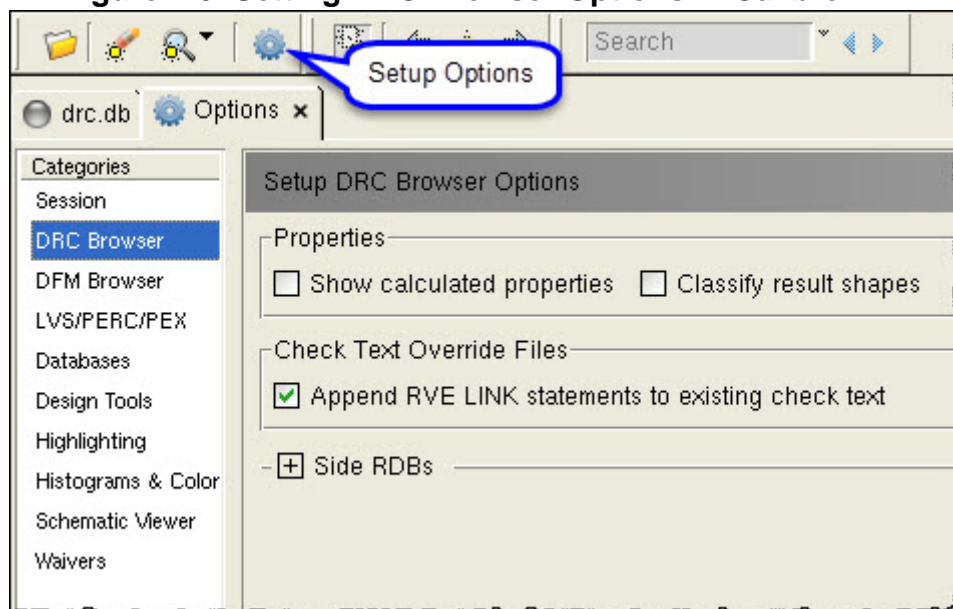
4. Click the icon in the detailed result view to display a menu from which you can select the columns to display in the result view as shown in [Figure 2-4](#).

**Figure 2-4. Selecting Columns to Display in the Calibre RVE Result View**



5. Click the **Setup Options** icon to display the **Options** tab and configure DRC Browser options as shown in [Figure 2-5](#).

**Figure 2-5. Setting DRC Browser Options in Calibre RVE**



6. Click the **drc.db** tab to return to viewing the results.
7. If a supported layout viewer (for example, Calibre DESIGNrev) is connected to Calibre RVE, you can highlight your results in the viewer by selecting the appropriate objects in the interface.
8. Close Calibre RVE when you are finished by selecting **File > Exit**.

## Results

You have now run Calibre RVE and viewed the results of the *drc.db* file that was generated by the Calibre nmDRC run. In the next procedure, you run Calibre nmLVS to perform layout versus schematic comparison.

# Performing Layout Versus Schematic Comparison with Calibre nmLVS-H

After a design has been placed and routed, you run Calibre nmLVS-H to perform layout netlist extraction followed by netlist-to-netlist comparison.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in [“Installing the Quick Start Example Data”](#) and set up your environment according to the information in [“Global Prerequisites.”](#)
- Licenses for the following products: Calibre nmLVS-H.
- The following inputs are required to perform this example:
  - SVRF rule file, *Module2/lvs.control*
  - OASIS layout, *Shared/Design/fullchip.oas*
  - Source SPICE netlist, *Shared/Design/source.net*
  - Hcell file, *Module2/cells*

## Procedure

1. In a UNIX or Linux terminal window, change directory to *Calbr\_qs\_ekit/Module2*:

```
cd <path>/Calbr_qs_ekit/Module2
```
2. Open the LVS control file, *lvs.control*, in an ASCII editor and notice the following SVRF rule file statements:

```
LAYOUT SYSTEM OASIS
LAYOUT PATH "../Shared/Design/fullchip.oas"
LAYOUT PRIMARY top

SOURCE SYSTEM SPICE
SOURCE PATH "../Shared/Design/source.net"
SOURCE PRIMARY top

LVS REPORT lvs.report
LVS SUMMARY REPORT lvs.summary
LVS REPORT OPTION EB EC EO ES F FX NOK R S V W

LVS EXECUTE ERC YES
ERC SUMMARY REPORT erc.report HIER
ERC RESULTS DATABASE erc.db
ERC KEEP EMPTY NO

GROUP supply_checks "?_no_supply_path"
ERC SELECT CHECK supply_checks

MASK SVDB DIRECTORY svdb QUERY
```

Everything up to the LVS Report statement is required in order to run LVS. The remaining statements are optional, but often used.

3. Notice the following two statements:

```
INCLUDE "../Shared/Design/layer.inc"  
INCLUDE "../Shared/Design/lvs.rules"
```

These point to other rule files that are used for the run. The *layer.inc* file contains layer definitions and other layout-directed statements. The *lvs.rules* file contains the LVS rules. Briefly review the contents of the files.

4. Close the *lvs.control* file.
5. In the *Module2* working directory, enter the following command at a shell prompt:

```
$CALIBRE_HOME/bin/calibre -spice layout.sp -turbo lvs.control \  
|tee extract.log
```

The *-spice* argument extracts a SPICE netlist from the layout and outputs the netlist to *layout.sp*. The *-turbo* argument specifies to use multithreaded, parallel processing. Calibre nmLVS-H generates a transcript of the netlist extraction run, which is output to *extract.log*.

6. Open the *extract.log* file.

The transcript contains similar information to the DRC transcript as shown under Step 6 on page 22.

The path of the executable, the command line options, and the PID are shown on these lines:

```
// Running <version>/pkgs/icv/pvt/calibre -spice layout.sp  
-turbo lvs.control  
// Process ID: <pid>
```

The tail of the run transcript shows the following:

```
--- SPICE NETLIST FILE = layout.sp  
--- CIRCUIT EXTRACTION REPORT FILE = lvs.report.ext  
--- PERSISTENT HIERARCHICAL DATABASE (PHDB) = svdb/top.phdb  
--- QUERY DATABASE = svdb TOP CELL = top  
--- TOTAL RULECHECKS EXECUTED = 6  
--- TOTAL RESULTS GENERATED = 1 (18)  
--- ERC RESULTS DATABASE FILE = erc.db (ASCII)  
--- ERC SUMMARY REPORT FILE = erc.report  
--- LVS SUMMARY REPORT FILE = lvs.summary
```

This shows the name of important output files and directories where results are stored. In this case, it also shows the number of ERC rule check results.

7. Close the *extract.log* file.
8. Open the *Module2/lvs.report.ext* file in a text editor.

This file lists any connectivity extraction or device recognition discrepancies. If there were discrepancies at this point, you would investigate and fix them.

9. Close the *lvs.report.ext* file.
10. Open the *Module2/erc.report* file in an ASCII editor and review the results.

Calibre outputs an ERC Summary Report that includes the following information:

- Header information — The first part of the ERC Summary Report lists general information about the run.
- Runtime Warnings — This section lists any warnings that were generated during the DRC run. See “Runtime Messages” in the *SVRF Manual* for a listing.
- Original Layer Statistics — This section lists the original layers and the number of original shapes processed for that layer. The first number is a hierarchical count and the number in parentheses is an estimated flat count.
- Rule Check Results Statistics — This section lists the rule checks and the number of results generated. The ERC Summary Report also lists the rule checks that were not performed. Rule check results are shown per rule check (hierarchical count followed by estimated flat count).
- ERC Rule Check Results Statistics (By Cell) — This section lists ERC rule checks by cell, when there are results. In this case, you will see this:

```
--- ERC RULECHECK RESULTS STATISTICS (BY CELL)
---
CELL SI ..... TOTAL Result Count = 1 (18)
      ERC CHECK psd_no_supply_path ... TOTAL Result Count = 1 (18)
```

11. Close the *erc.report* file.
12. Invoke Calibre nmLVS-H comparison by entering the following command at a shell prompt:

```
$CALIBRE_HOME/bin/calibre -lvs -hier -hcell cells -layout layout.sp
lvs.control |tee compare.log
```

The *-hcell* argument specifies the name of a file that defines corresponding cells in the layout and source files.

13. Review the end of the transcript. You will see these lines:

```
LVS completed. INCORRECT. See report file: lvs.report
...
--- LVS REPORT FILE = lvs.report
...
--- LVS SUMMARY REPORT FILE = lvs.summary
```

This shows the overall result of the LVS comparison and the names of important files generated during the run.

14. Review the *Module2/lvs.summary* file in a text editor.

This file shows some useful statistics and message summaries.

15. Close the *lvs.summary* file.

16. Open the *Module2/lvs.report* file and review the results in a text editor.

The LVS Report includes the following information:

- LVS header section showing information about the run.
- Overall Comparison Results section.  
Shows an INCORRECT status.
- Cell Summary section.  
Shows comp5\_sc and top cells are INCORRECT.
- LVS Parameters section showing LVS option settings.
- Cell-by-cell Comparison Results section.

17. Find the comp5\_sc cell results section. Notice this discrepancy:

```
*****
                        SOURCE ERRORS
*****

DISC#

*****

Properties Missing on Instances:
    1      property c              not found on      CC0 (C)
```

This means device instance CC0 in the source is missing a capacitance value, so it cannot be matched to the layout.

18. Find the TOP LEVEL comparison results section. Notice this discrepancy (adjust the width of your text editor to see the report without line wrapping):

```
*****
                        PROPERTY ERRORS
*****

DISC#  LAYOUT                                SOURCE                                ERROR

*****

1  X19/X1/X74/R0 (145.222,361.572)  R  Xdacinst/XI4/XI41/RR0  R (RP)
   r: 8000 ohm                    r: 1000 ohm              700%

2  X19/X1/X75/R0 (146.222,366.512)  R  Xdacinst/XI4/XI41/RR1  R (RP)
   r: 8000 ohm                    r: 1000 ohm              700%

3  X19/X2/R0 (160.105,380.882)  R (RP)  Xdacinst/XI1/XI0/RR0  R (RP)
   r: 3086.22 ohm                 r: 1000 ohm              209%
```

This means there are resistance value discrepancies between layout and source.

19. Close the *lvs.report* when you have finished.

## Results

You have now run Calibre nmLVS-H and compared the source layout and source schematic files. The outputs from the Calibre nmLVS-H run include:

- Run transcripts (*extract.log* and *compare.log*)
- Connectivity extraction report (*lvs.report.ext*)
- ERC Summary Report (*erc.summary*)
- LVS Summary Report (*lvs.summary*)
- LVS Report (*lvs.report*)
- ERC Results Database (*erc.db*)
- Mask SVDB Directory (*svdb*)

Depending on the tasks you are performing, Calibre nmLVS-H can optionally output the following files:

- Short isolation results database
- Soft connection results database

At this point, you would normally fix any errors and run DRC, netlist extraction, and LVS again.

## Reviewing LVS Results

You can review the Mask SVDB Directory using Calibre RVE.

Ordinarily, Calibre RVE is invoked either from a layout editor or from Calibre Interactive. If your tool environment is configured with Calibre RVE integrated into such tools, then call Calibre RVE from the appropriate menu items. In layout viewers like Calibre DESIGNrev, you call Calibre RVE from the Verification menu. In third-party layout editors, you frequently call Calibre RVE from the Calibre menu. In this procedure, no layout editor is assumed to be configured, and Calibre Interactive is not assumed to be in your environment.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- Licenses for the following products: Calibre RVE.

- A Mask SVDB Directory such as svdb from the procedure “[Performing Layout Versus Schematic Comparison with Calibre nmLVS-H](#)”.

## Procedure

1. Verify your working directory is *Calbr\_qs\_ekit/Module2*.
2. Use one of the following methods to invoke Calibre RVE:
  - Invoke Calibre RVE using the *svdb* file as input.

```
$CALIBRE_HOME/bin/calibre -rve svdb
```

- If your layout viewer is configured to open Calibre RVE, use **File > Open Database** to open Calibre RVE and load the SVDB database located in the working directory.

The navigator (left-hand) pane of the interface includes links to the results, along with links to various report files and debugging tools. The **Comparison Results** tab displays and sorts the LVS comparison errors from the Calibre job. This tab is organized into a tree view (upper pane) and detailed view (lower pane) that displays results on a per-cell basis.

3. In the **Navigator** tab, click **ERC Results**.

An **ERC Results** tab opens that is similar to the results presentation shown in Calibre RVE for DRC. The method for viewing results is similar to the method described in “[Reviewing DRC Results](#)” on page 24.

4. Select the **Comparison Results** tab and expand the results tree by clicking the “top” cell name. Expand the “Discrepancies” and “Property Errors” levels in the same way.

With Property Errors selected, you will see three discrepancies.



**Figure 2-6. Viewing LVS Discrepancies in Calibre RVE**

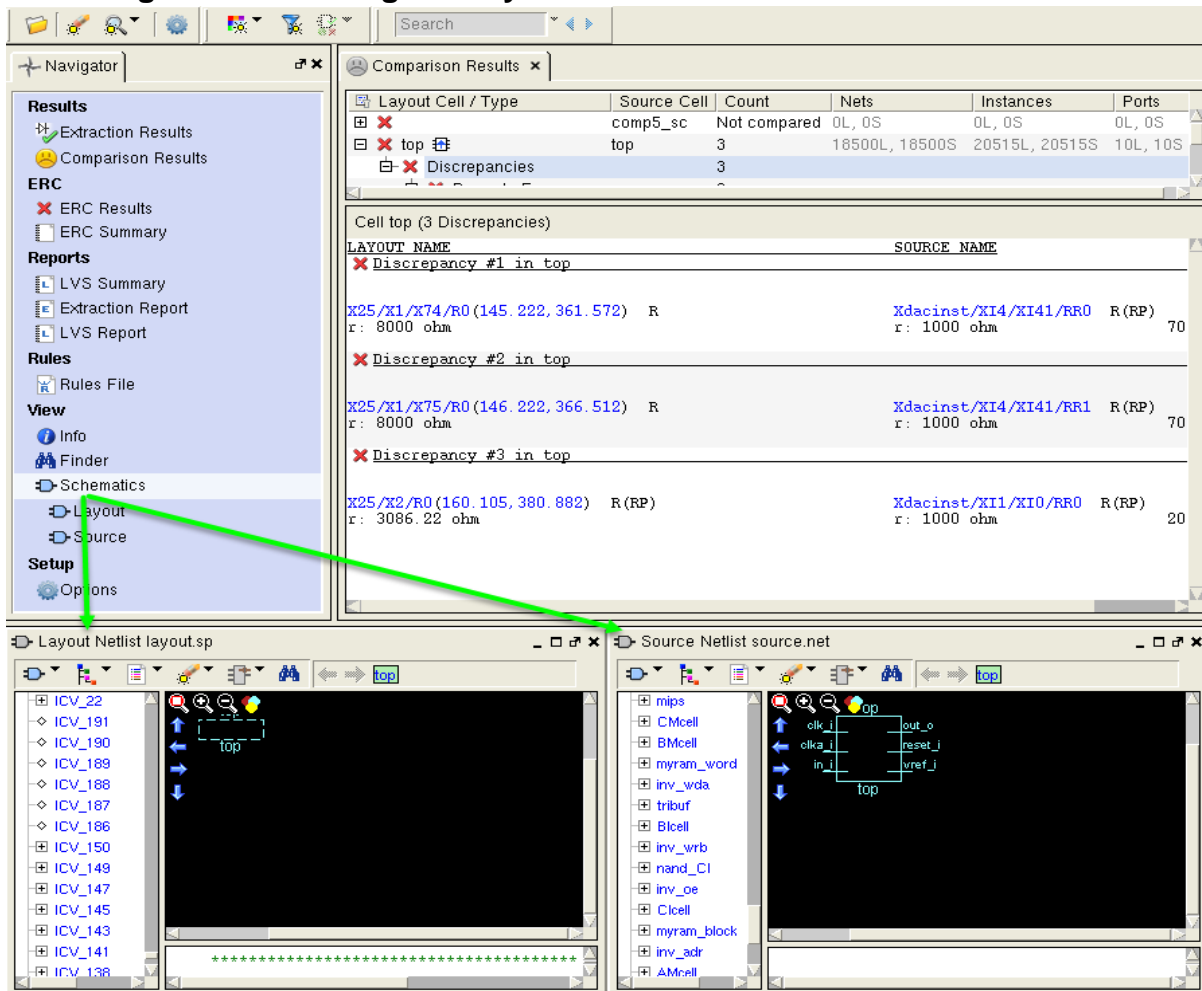
Layout Cell / Type	Source Cell	Count	Nets
✖	comp5_sc	Not compared	0L, 0S
✖ top	top	3	18500L, 18500S
✖ Discrepancies		3	
✖ Property Errors		3	
✖ Discrepancy #1			
✖ Discrepancy #2			
✖ Discrepancy #3			


Cell top (3 Property Errors)			
LAYOUT NAME	SOURCE NAME		
✖ Discrepancy #1 in top			
X25/X1/X74/R0(145.222, 361.572) R r: 8000 ohm	Xdacinst/XI4/XI41/RR0 R(RP) r: 1000 ohm	700%	
✖ Discrepancy #2 in top			
X25/X1/X75/R0(146.222, 366.512) R r: 8000 ohm	Xdacinst/XI4/XI41/RR1 R(RP) r: 1000 ohm	700%	
✖ Discrepancy #3 in top			
X25/X2/R0(160.105, 380.882) R(RP) r: 3086.22 ohm	Xdacinst/XI1/XI10/RR0 R(RP) r: 1000 ohm	209%	

- (Optional) If you started your session from a layout viewer and Calibre RVE has a socket connection with that viewer, click the first link under SOURCE NAME. The schematic browsers open in Calibre RVE and the R devices associated with the discrepancy are highlighted in the layout viewer. In this case, you do not need to perform the subsequent steps to find the R devices in the source netlist. But, knowing how to use the finder tool to locate design objects is useful.
- In the **Navigator** tab, click **Schematics**.

The schematic browsers open displaying the layout and source netlists as shown in [Figure 2-7](#).


**Figure 2-7. Viewing the Layout and Source Netlists in Calibre RVE**



7. In the Source Netlist browser, click the **Find** icon  to display the Find bar.
8. In the Find bar, select **Named Inst/Device** from the menu to change the object type.
9. In the Find bar text entry field, enter the following (note the difference between the letter I and the numeral 1):

**Xdacinst/XI4/XI41/RR0**

Press Enter to highlight the RR0 device instance in the Source Netlist browser. Eight series devices highlight in the Layout Netlist browser. The problem is, the eight devices in the layout were reduced in series to correspond with the source, but their combined resistances are too big.

10. Undock the netlist browsers by choosing the undock icon  in the top-right corner of each browser panel.

Adjust the browser windows to the desired viewing dimensions.

11. You can repeat Step 9 to find the R devices associated with the remaining discrepancies.

12. Close Calibre RVE when you are finished by selecting **File > Exit**.

## Results

You have now run Calibre RVE and reviewed the results in the Mask SVDB Directory. Typically, you would fix the errors in the layout, then run DRC and LVS again to verify the fixes.



# Chapter 3

## Designing for Manufacturability

---

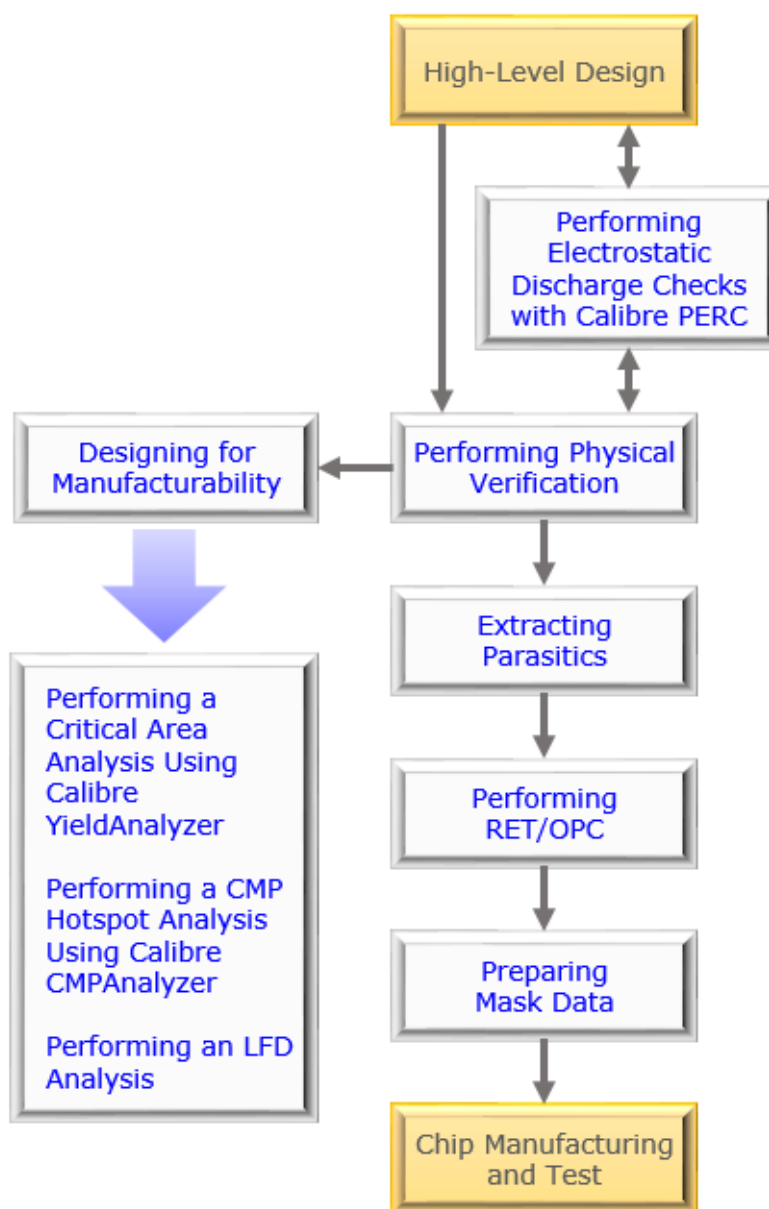
Design for Manufacturability (DFM) is a methodology that includes techniques for improving the yield. The Calibre DFM tools help designers determine which problems have the greatest impact on yield and are most easily corrected. Additionally, the Calibre DFM tools can provide recommendations on how to modify a design to address well-characterized weaknesses.

<b>Design for Manufacturability Design Flow .....</b>	<b>37</b>
<b>Calibre Design for Manufacturability Tool Reference .....</b>	<b>38</b>
<b>Performing a Critical Area Analysis Using Calibre YieldAnalyzer .....</b>	<b>41</b>
<b>Performing a CMP Hotspot Analysis Using Calibre CMPAnalyzer .....</b>	<b>46</b>
<b>Performing an LFD Analysis .....</b>	<b>52</b>

### Design for Manufacturability Design Flow

Figure 3-1 shows some of the DFM checks as described in the procedures in this module.

**Figure 3-1. Design for Manufacturability Flow**



## Calibre Design for Manufacturability Tool Reference

Supporting documentation is available for each licensed Calibre product and, in some cases, training is also available. There may be other products that are closely related to the use of that tool.

**Table 3-1** provides a comprehensive list and short description of the licensed Calibre products associated with design for manufacturability, in addition to information on related documentation, training, and products.

**Table 3-1. Calibre Design for Manufacturability Tool Reference**

Tool	Description	Related Documentation	Related Training	Related Products
Calibre® CMPAnalyzer	A tool used to analyze designs with respect to thickness requirements, and to identify planarization issues that can have a negative impact on yield.	<i>Calibre CMPAnalyzer User's Manual</i>		Calibre RVE ----- Calibre Interactive
Calibre® CMP Modeling	A tool used to model the CMP process, correlate to measured data, and finalize a model for Calibre CMPAnalyzer.	<i>Calibre CMP Model Builder User's and Reference Manual</i>		Calibre WORKbench ----- Calibre CMPAnalyzer
Calibre® DESIGNrev™	A layout viewer used for full-chip design viewing and basic editing.	<i>Calibre DESIGNrev Layout Viewer User's Manual</i> ----- <i>Calibre DESIGNrev Reference Manual</i>	<a href="#">Calibre Fundamentals: DESIGNrev</a>	Calibre WORKbench ----- Calibre LITHOview ----- Calibre MDPview
Calibre® Interactive™	A GUI interface for nmDRC, nmLVS, PERC, xRC, and DFM. Also calls Calibre RVE.	<i>Calibre Interactive User's Manual</i>	<a href="#">Calibre Fundamentals: Performing DRC/LVS</a>	Calibre RVE

**Table 3-1. Calibre Design for Manufacturability Tool Reference (cont.)**

<b>Tool</b>	<b>Description</b>	<b>Related Documentation</b>	<b>Related Training</b>	<b>Related Products</b>
Calibre® LFD™ (Litho-Friendly Design)	A pre-tapeout verification engine designed to help characterize sensitivity of designs to process variability.	<i>Calibre Litho-Friendly Design User's Manual</i>		Calibre RVE
Calibre® RVE™	A GUI used to view and debug output from Calibre tools.	<i>Calibre RVE User's Manual</i>	<a href="#">Calibre Fundamentals: Performing DRC/LVS</a> ----- <a href="#">Calibre Advanced Topics: nmLVS Debug Case Studies</a>	Calibre Interactive
Calibre® YieldAnalyzer	A tool used to calculate quality assessment metrics or yield assessment metrics for a specific layout. This makes it possible to compare layouts, identify potential failure spots, and prioritize design modifications.	<i>Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual</i> ----- <i>SVRF Manual</i>		Calibre RVE ----- Calibre Interactive ----- Calibre YieldServer



**Table 3-1. Calibre Design for Manufacturability Tool Reference (cont.)**

<b>Tool</b>	<b>Description</b>	<b>Related Documentation</b>	<b>Related Training</b>	<b>Related Products</b>
Calibre® YieldEnhancer	A tool used to improve yield through automated design modifications targeted at improving certain well-characterized design weaknesses. Methods of improving yield include adding redundant vias, expanding via enclosures, and growing sensitive features.	<i>Calibre YieldAnalyzer and YieldEnhancer User's and Reference Manual</i> ----- <i>SVRF Manual</i>		Calibre RVE ----- Calibre YieldServer
Calibre® YieldServer	A tool used to manage the layout design data required when designing for improved manufacturability and yield. Provides functionality not available through the standard Calibre nmDRC hierarchical engine.	<i>Calibre YieldServer Reference Manual</i> ----- <i>SVRF Manual</i>		Calibre RVE ----- Calibre YieldAnalyzer ----- Calibre YieldEnhancer

## Performing a Critical Area Analysis Using Calibre YieldAnalyzer

You can run a critical area analysis using Calibre Interactive and review results from the generated DFM database using Calibre RVE.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- Licenses for the following products: Calibre DESIGNrev, Calibre Interactive, Calibre RVE, and Calibre YieldAnalyzer.
- The following inputs are required for this example:
  - Layout, *Shared/Design/fullchip.oas*
  - Layer properties file, *Shared/Design/fullchip.oas.layerprops*

## Procedure

1. In a terminal window, change directory to *Calbr\_qs\_ekit/Module3*:  

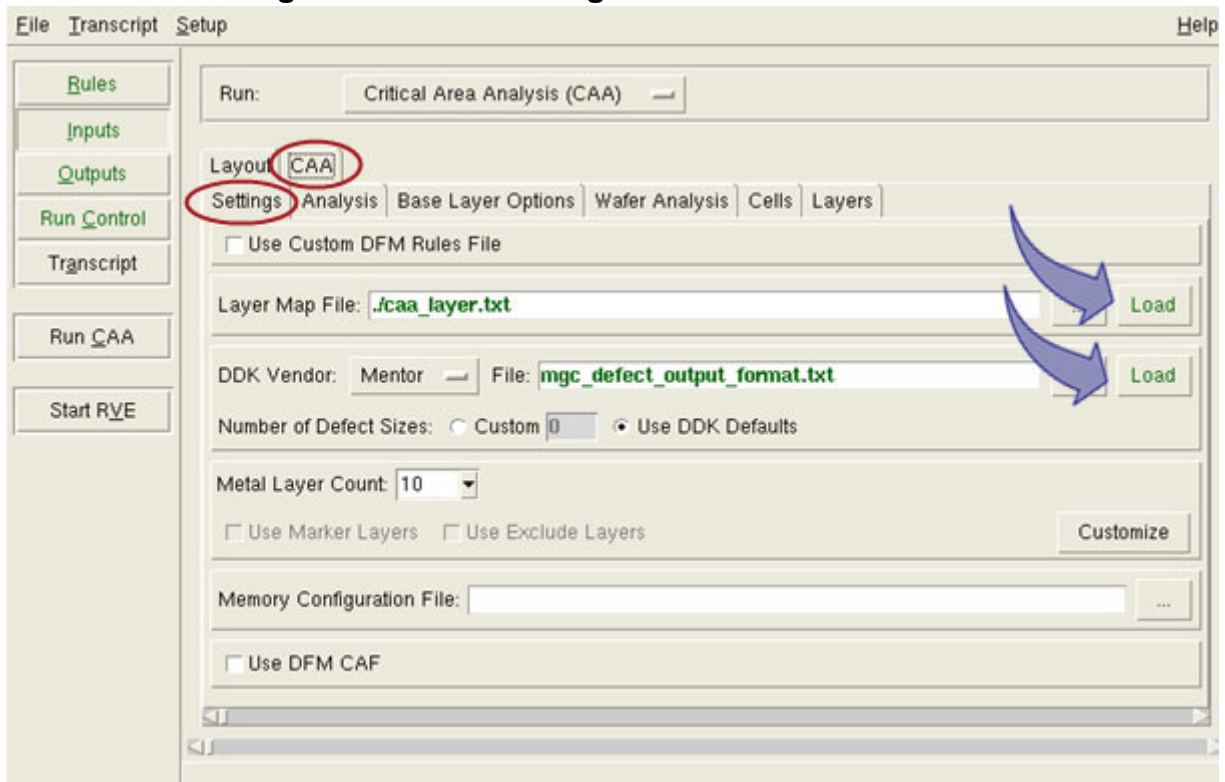
```
cd <path>/Calbr_qs_ekit/Module3
```
2. Examine the *caa\_layer.txt* file in a text editor.  

This file is used to map layers in your design to layers recognized by the CAA tool.
3. Examine the *mgc\_defect\_output\_format.txt* file in a text editor.  

This file follows the CAA Defect Data Format and defines the range of defect size parameters for each layer in the design. The tool uses these parameters to generate critical area.
4. From the *Module3* directory, launch Calibre DESIGNrev:  

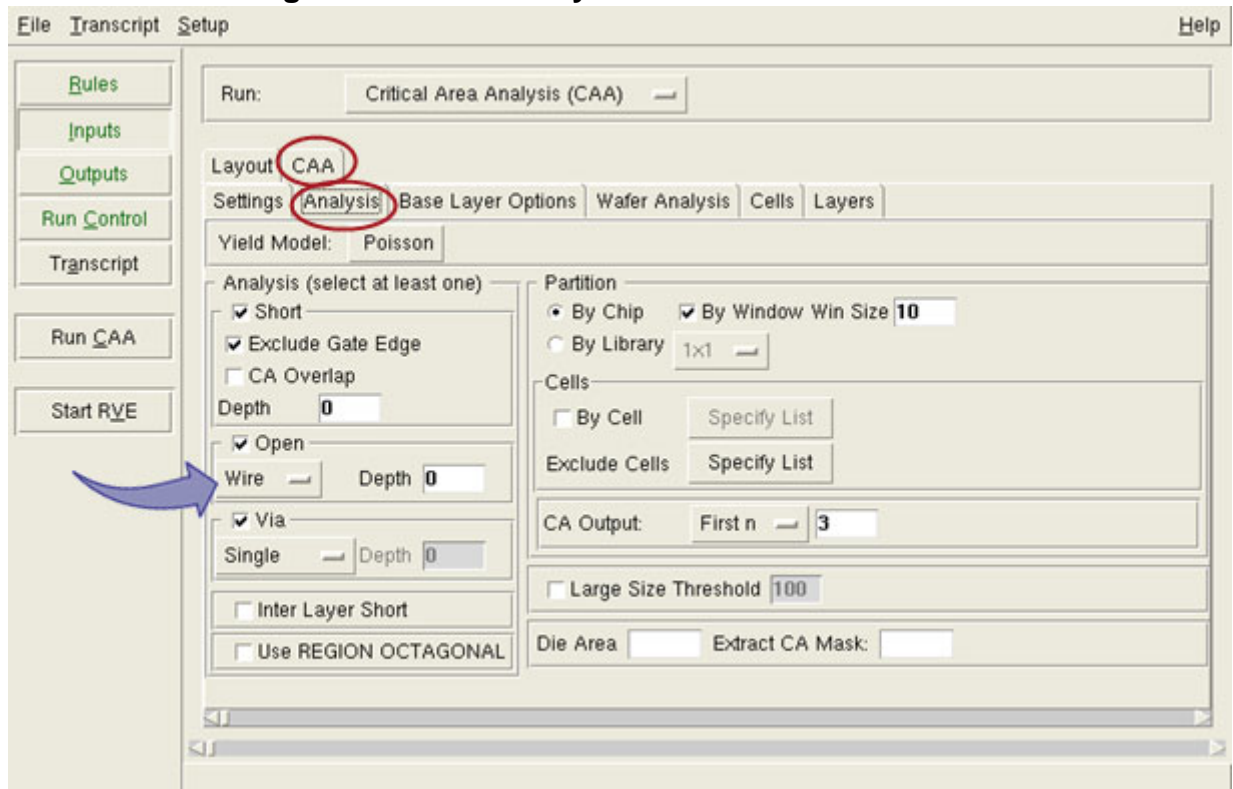
```
$CALIBRE_HOME/bin/calibredrv ../Shared/Design/fullchip.oas
```
5. Choose **Verification > Run DFM** to launch Calibre Interactive.
6. When the Load Runset File dialog box appears, enter *./runset\_caa* in the Runset File Path field and click **OK**.
7. Click the **CAA** tab and then click the **Settings** tab if it is not already selected.
8. Ensure *./caa\_layer.txt* appears in the Layer Map File text box and click **Load**.
9. Ensure *mgc\_defect\_output\_format.txt* appears in the Mentor DDK Vendor text box and click **Load** as shown in [Figure 3-2](#).

Figure 3-2. CAA Settings Tab Calibre Interactive



10. Click the **Analysis** tab.
11. Verify that the **Open** setting is selected and choose Wire, as shown in [Figure 3-3](#).

Figure 3-3. CAA Analysis Tab Calibre Interactive

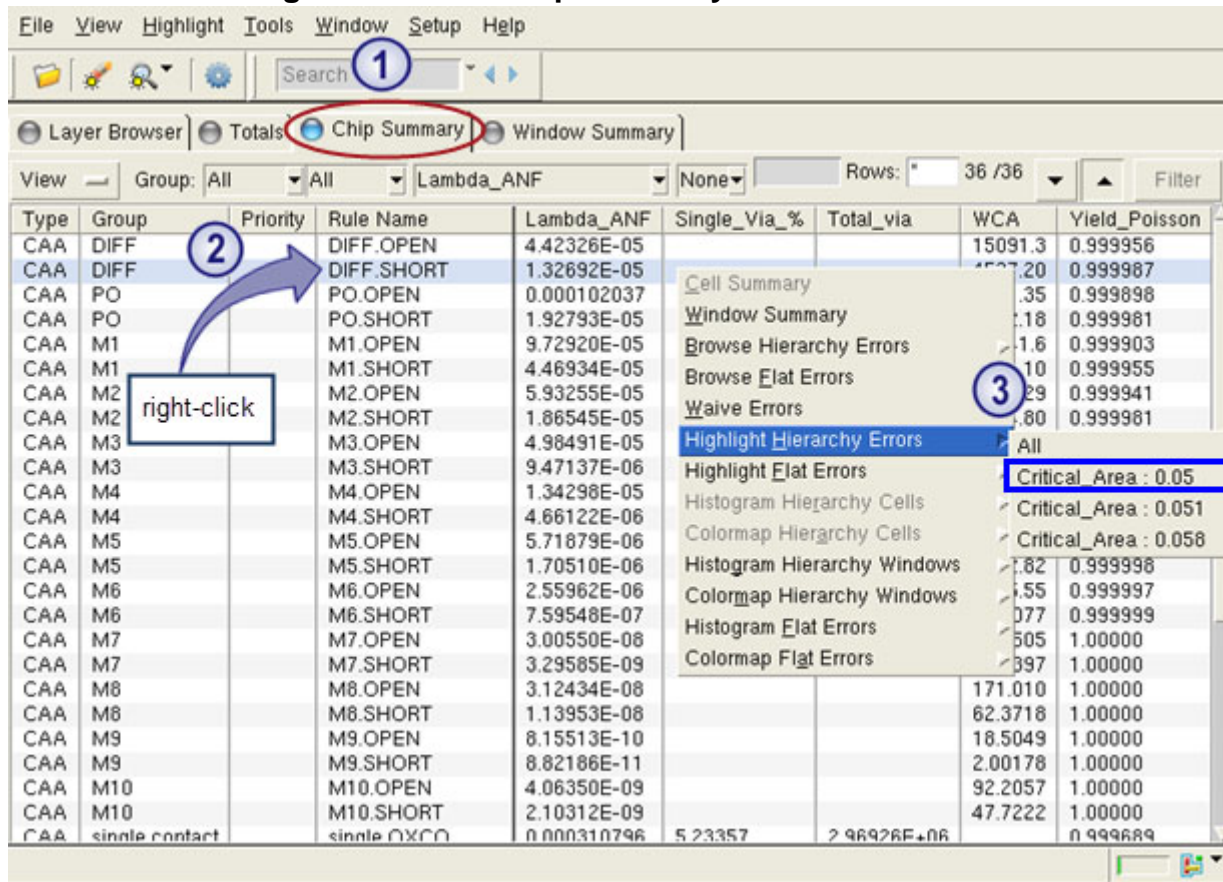


12. Click **Run CAA**. If the Overwrite file? dialog box appears, click **Overwrite**.

This opens the Calibre Interactive Transcript pane and starts the CAA run. The progress meter at the bottom of the window displays the progress of the run. Calibre RVE automatically launches when the CAA run completes.

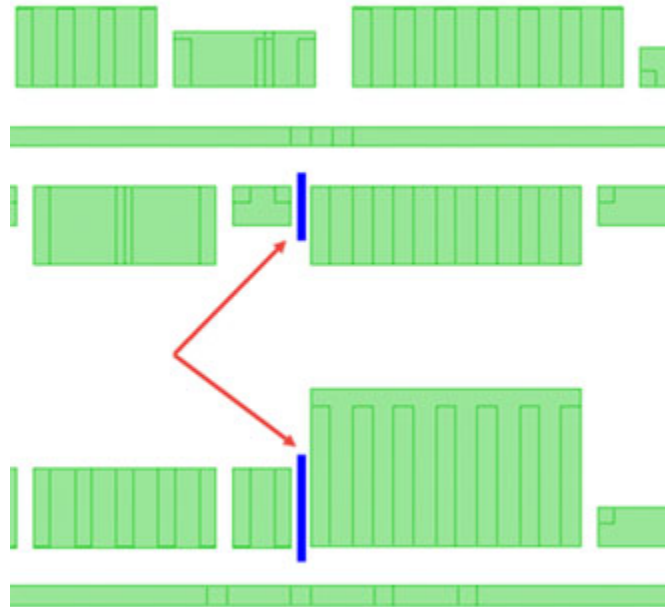
13. From Calibre RVE, you can highlight error data on the layout, and generate histograms and colormaps by chip, by cell, and by window. For example, to highlight critical area on the layout for the DIFF layer:
  - a. In Calibre RVE, click the **Chip Summary** tab.
  - b. Right-click on the row containing the DIFF.SHORT rule name, and choose **Highlight Hierarchy Errors > Critical\_Area : 0.05** as shown in [Figure 3-4](#).

Figure 3-4. CAA Chip Summary Tab Calibre RVE



- c. In Calibre DESIGNrev, use the Ctrl key to select the “active” and “rve” layers in the Layers Browser.
- d. Right-click and choose **Show Selected Only** from the Layers popup menu.
- e. Press 9 to show all levels of hierarchy.
- f. Zoom in to view the critical area (layer rve) between the shapes on the active layer by right-clicking and dragging the cursor.

This area represents the region where a short will occur if the center point of a defect particle lands there. In this case, the critical area shown is based on a defect particle radius of 0.05 units.



For other types of analysis you can perform, refer to the [Calibre CAA Example Kit \(eKit\)](#) available for download on Support Center.

14. Close Calibre RVE, Calibre DESIGNrev, and any open DFM files in your text editor.

## Results

You have now run Calibre YieldAnalyzer and performed a critical area analysis on the design. The key outputs from Calibre YieldAnalyzer include:

- *dfmdb* — The directory containing the DFM database. You load the DFM database into Calibre RVE to view the analysis results.
- *\_caa\_gui.tvf\_* and *\_\_caa\_gui.tvf\_\_* — The generated rule files.

In the next procedure, you will use Calibre CMPAnalyzer to perform a CMP hotspot analysis.

## Performing a CMP Hotspot Analysis Using Calibre CMPAnalyzer

You can run a chemical mechanical polishing (CMP) hotspot analysis using Calibre Interactive and review the results from the generated DFM database using Calibre RVE.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- Licenses for the following products: Calibre DESIGNrev, Calibre Interactive, and Calibre RVE.
- The following inputs are required to perform this example:
  - Layout, *Shared/Design/fullchip.oas*
  - Layer properties file, *Shared/Design/fullchip.oas.layerprops*

## Procedure

1. In a terminal window, change directory to *Calbr\_qs\_ekit/Module3*:

```
cd <path>/Calbr_qs_ekit/Module3
```

2. Examine the *recipe.in* file, which contains the following text:

```
#
=====
# M1 model
# =====
initialize material=silicon position=0A
deposit thickness=200A type=isotropic material=nitride
deposit thickness=3900A type=isotropic material=oxide
deposit thickness=300A material=HM type=isotropic
etch type=anisotropic mask=31 maskFile=mask thickness=2400A
deposit thickness=100A type=isotropic anisotropyFactor=0.95 \
    material=TaN
deposit thickness=350A type=isotropic anisotropyFactor=0.65 \
    material=copper
deposit type=ecd model=ecd1 rate=20.0nm/s time=66.3 dt=0.1s
cmp dz=25A model=cmp1 pressure=2.1psi targetThicknessNT=1500A \
    targetMaterial=copper
cmp dz=10A model=cmp2 pressure=1.8psi targetMaterial=TaN \
    OpenAreaFraction=0.9
cmp time=80s dz=5A model=cmp3 pressure=0.8psi
```



```

# =====
# Mx model. The same as M1.
# =====
deposit thickness=5000A type=isotropic material=ILD
deposit thickness=200A type=isotropic material=nitride
deposit thickness=3900A type=isotropic material=oxide
deposit thickness=300A material=HM type=isotropic
etch type=anisotropic mask=31 maskFile=mask thickness=2400A
deposit thickness=100A type=isotropic anisotropyFactor=0.95 \
    material=TaN
deposit thickness=350A type=isotropic anisotropyFactor=0.65 \
    material=copper
deposit type=ecd model=ecd1 rate=20.0nm/s time=66.3 dt=0.1s
cmp dz=25A model=cmp1 pressure=2.1psi targetThicknessNT=1500A \
    targetMaterial=copper
cmp dz=10A model=cmp2 pressure=1.8psi targetMaterial=TaN \
    OpenAreaFraction=0.9
cmp time=80s dz=5A model=cmp3 pressure=0.8psi

#
# =====
# Model definition
# =====
#
define_model name=ecd1 type=ecd k1=0.15 k2=1.0 k1s=0.62 k2s=1.2 \
    phistar=0.5 sfactor=0.55 wcrit=0.12 keq=8.5 rmod=0 smod=0 \
    LECD=1350 CuDepletionFactor=0.015
define_model name=cmp1 type=cmp spad=45000 material=copper \
    rate=110.0 dmax=55.3 alpha2=0.35 beta2=0.72 dmin=55.3 \
    alpha3=0.72 beta3=0.35
define_model name=cmp2 type=cmp spad=28100 material=copper \
    rate=75 dmax=279 alpha2=0.15 beta2=0.03 dmin=279 alpha3=0.03 \
    beta3=0.15 material=TaN ratefactor=0.0116 dmax=279 \
    alpha2=0.15 beta2=0.03 dmin=279 alpha3=0.03 beta3=0.15
define_model name=cmp3 type=cmp spad=19345 material=copper \
    rate=6.5 dmax=59 alpha2=0.37 beta2=0.26 dmin=59 alpha3=0.26 \
    beta3=0.37 material=TaN ratefactor=0.8 dmax=59 alpha2=0.37 \
    beta2=0.26 dmin=59 alpha3=0.26 beta3=0.37 material=HM \
    ratefactor=0.76 dmax=59 alpha2=0.37 beta2=0.26 dmin=59 \
    alpha3=0.26 beta3=0.37 material=oxide ratefactor=0.67 \
    dmax=59 alpha2=0.37 beta2=0.26 dmin=59 alpha3=0.26 \
    beta3=0.37

```

This recipe file defines the various process-specific parameters that are needed for the CMP analysis. The backslash “\” characters shown in the text indicate line continuation.

### 3. Examine the *runset\_cmp* file.

This runset file stores settings that are read by Calibre Interactive to automatically configure the GUI. There is an extra line break at the end of the file because the last line in the file must be empty. Runset files are normally generated by manually configuring Calibre Interactive, then choosing **File > Save Runset As**.

### 4. Invoke Calibre DESIGNrev:

```
$CALIBRE_HOME/bin/calibredrv ../Shared/Design/fullchip.oas
```

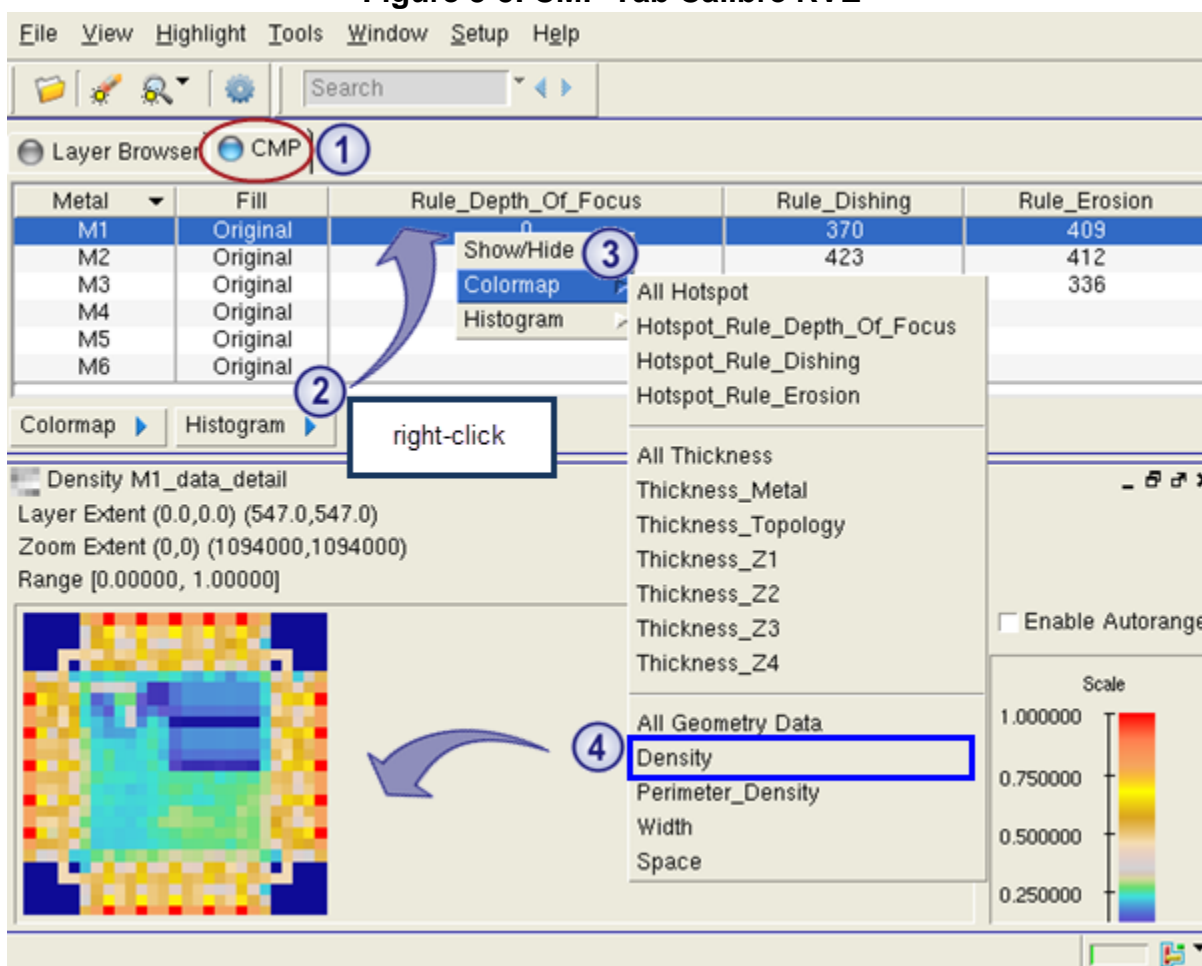


5. In Calibre DESIGNrev, choose **Verification > Run DFM** to launch Calibre Interactive.
6. When the Load Runset File dialog box appears, enter `./runset_cmp` in the Runset File Path text box and click **OK**.
7. In Calibre Interactive Inputs pane, review the **CMP** tab and the **Simulator Options** tab fields and settings.
8. Click **Run CMP**. If the Overwrite file? dialog box appears, click **Overwrite**.

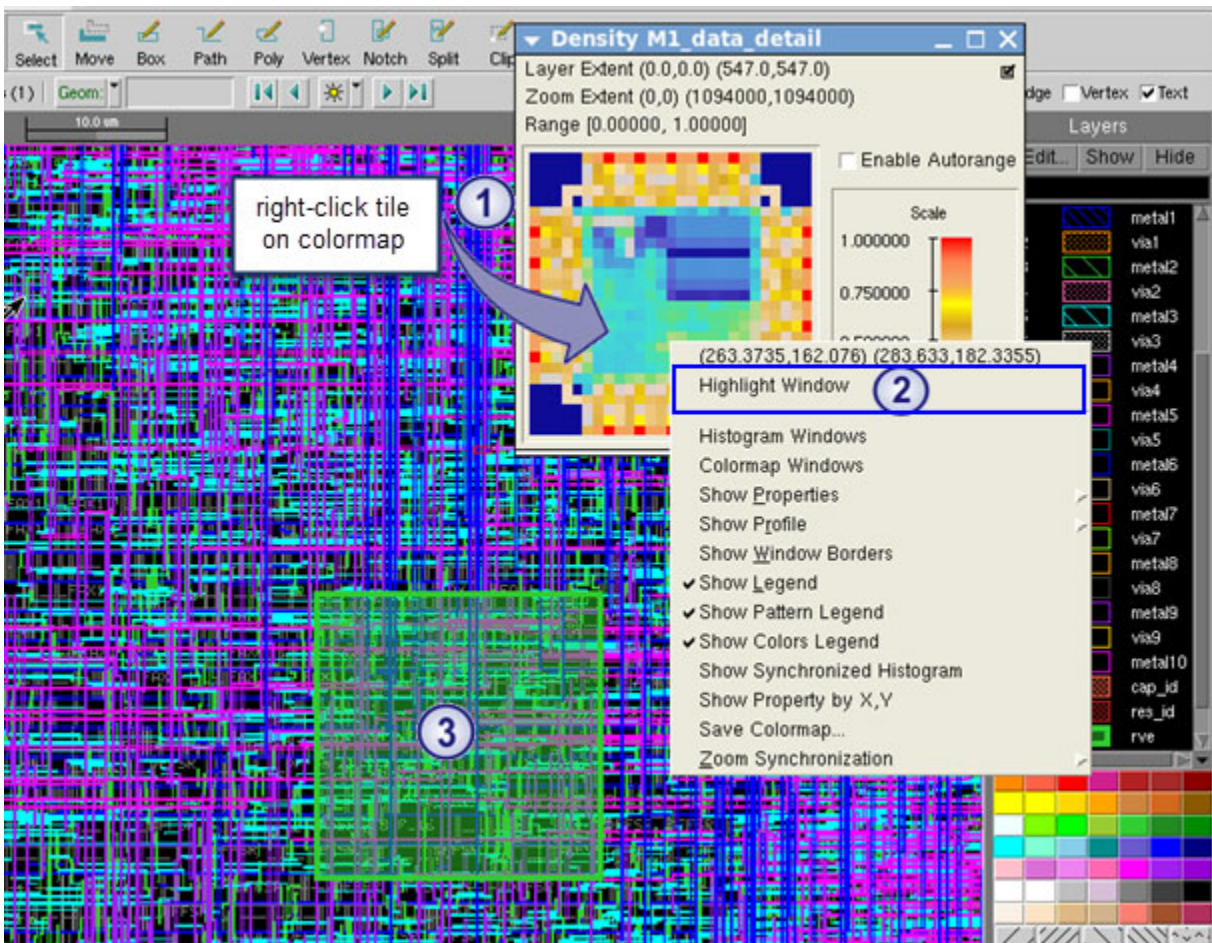
This opens the Calibre Interactive Transcript pane and starts the CMP run. The progress meter at the bottom of the window displays the progress of the run. Calibre RVE automatically launches when the Calibre CMPAnalyzer run completes.

9. From Calibre RVE, you can analyze hotspot, thickness, and density data on the design through the use of colormaps and histograms. For example, to display a colormap of density values for M1:
  - a. Verify the **CMP** tab is selected in Calibre RVE.
  - b. Right-click on the row containing M1, and choose **Colormap > Density**. A colormap appears that represents the M1 density on the layout per rectangular region (also called a window). Warmer colors indicate higher density values, and cooler colors indicate lower density values as shown in [Figure 3-5](#).

**Figure 3-5. CMP Tab Calibre RVE**



- c. From the generated colormap, you can highlight windows on the layout in Calibre DESIGNrev. To do this, right-click on one of the colormap squares and choose **Highlight Window**. The corresponding window is highlighted on the layout.



For further details on the types of analysis you can perform, refer to the “CMP Analysis Results” chapter in the *Calibre CMPAnalyzer User’s Manual* and the [CMPAnalyzer Example Kit \(eKit\)](#) available for download on Support Center.

10. Close the Calibre RVE and Calibre DESIGNrev sessions.

## Results

You have now run Calibre CMPAnalyzer and performed a CMP hotspot analysis on the design. The key outputs from Calibre CMPAnalyzer include:

- *dfmddb* — The directory containing the DFM database. Calibre RVE loads the DFM database when you view the analysis results.
- *\_rules\_* — The generated rule file.

In the next procedure, you will use Calibre LFD to analyze LFD results.

## Performing an LFD Analysis

You can perform a Calibre Litho-Friendly Design (LFD) results analysis and use Calibre DESIGNrev and Calibre RVE to view LFD results. The data used in this example introduces principles involved in using Calibre LFD to help find lithography hotspots.

### Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- Licenses for the following products: Calibre DESIGNrev, Calibre Interactive, and Calibre RVE.
- The following inputs are required for this example:
  - Layout, *Module3/lfd\_example/lfd\_demo.gds*
  - Layer properties file, *Module3/lfd\_example/lfd\_demo.gds.layerprops*
  - LFD results, *Module3/lfd\_results/*

### Procedure

1. In a terminal window, change directory to *Calbr\_qs\_ekit/Module3*:


```
cd <path>/Calbr_qs_ekit/Module3
```
2. From the command line, open Calibre DESIGNrev with the layout:

```
$CALIBRE_HOME/bin/calibredrv lfd_example/lfd_demo.gds
```
3. In Calibre DESIGNrev, set the viewing hierarchy depth to 9 by typing “9” anywhere in the layout window.
4. In Calibre DESIGNrev, hide all layers except M1 by right-clicking on M1 in the Layers browser and choosing **Show Selected Only** from the Layers popup menu.
5. Start Calibre RVE for DFM from within the Calibre DESIGNrev application:
  - a. In Calibre DESIGNrev, choose **Verification > Start RVE**.
  - b. In the Calibre RVE dialog box, choose the DRC/ERC Database Type.
  - c. Click the **Browse** button [...] and navigate to the *lfd\_results* directory.
  - d. Choose the *TOP.dfm.results* file and click **OK**.
  - e. Click **Open** in the Calibre RVE dialog box.

The Calibre RVE window should contain no results, as all results were written to RDB files and not the DRC results database. The rule file used for this particular LFD job writes all results to RDB files which can be accessed from the DRC results database file.

6. In Calibre RVE, open the Process Variability Bands (PV-bands) results.

An important step in the Calibre LFD job flow is the creation of polygons that represent the predicted printed image edges based on process window data. Using sophisticated modeling techniques, Calibre LFD computes the minimum and maximum printed shape or contour edges and outputs the area found between the minimum and maximum edge locations as PV-bands for every shape on a given layer. This data is then used to perform internal, external, and enclosure checks to verify that the printed images lead to acceptable yields.

- a. Click the **Open side RDBs**  icon to display the list of RDB files.
- b. Select the file `./lfd_results/lfd_pvbands.rdb`.


This opens the **lfd\_pvbands.rdb** tab in Calibre RVE.

From the list of checks in Calibre RVE, note that PV-bands were generated for the OX, PO, CO, and M1 layers. The rules have been named such that the BAND results represent the min or max edge difference region. The CNTR results show the entire contour with edges between min and max.

The p1 BAND results represent “priority one” results which should definitely be examined and, if possible, fixed. The p2 BAND results represent “priority two” results which were generated using stricter rules and are therefore more conservative. The p2 BAND results should be reviewed for designs that may be more sensitive to in-die variation.

---

#### Note

 While the rule file used for this Calibre LFD job wrote all PV-bands shapes to an RDB file, a good practice for larger designs is to write the PV-bands shapes to an OASIS or GDS file. PV-bands are polygons with many vertices, and storing this shape information in an ASCII RDB file can lead to excessive file sizes.

---


7. Prepare Calibre RVE for viewing the PV-bands data.
  - a. Select **Setup > Options**.
  - b. Choose the **Highlighting** category to display the Setup Highlighting Options.
  - c. Enable the “Zoom to highlights by option” and set the ratio to 0.6.
  - d. Disable the “Clear existing highlights before showing new highlights” option.
  - e. Click **Apply** and close the Options tab.
8. In the **lfd\_pvbands.rdb** tab, select the M1\_BAND\_p1 check and double-click the fourth result.

The M1 PV-band associated with the above contour is highlighted in the layout.



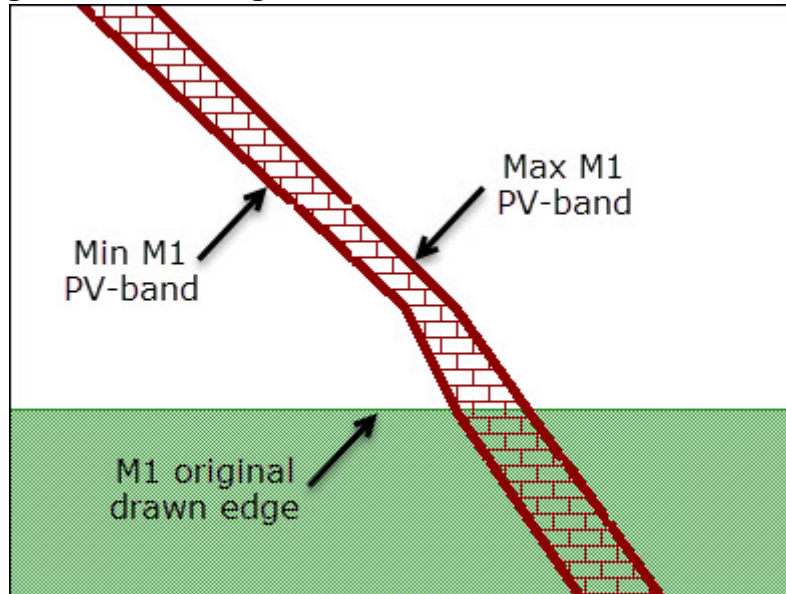
9. In Calibre DESIGNrev, zoom into an edge of the highlighted contour to see the relationship of the contour edges and the drawn edge as shown in [Figure 3-6](#).



**Note**

 The fill pattern for the Calibre RVE layer used for highlighting the PV-bands data fill pattern has been set to better view the contour edges. This is accomplished by selecting the “rve” layer on the Calibre DESIGNrev Layers palette, and selecting a fill pattern that is not opaque.

---

**Figure 3-6. Viewing LFD PV Bands in Calibre DESIGNrev**



10. In Calibre RVE, click the **Clear All Highlights**  icon to clear all highlights.
11. Right-click on the M1\_BAND\_p1 check name and select **Highlight**.  
This highlights all of the “priority one” M1 PV-bands in the layout.
12. Select the **TOP.dfm.results** tab in Calibre RVE.
13. Click the **Open side RDBs**  icon in Calibre RVE and choose the file *.lfd\_results/lfd\_checks.rdb* from the menu.  
  
This opens the **lfd\_checks.rdb** tab in Calibre RVE. Examine the contents of the **lfd\_checks.rdb** tab, and note that from the list of checks that generated results, you can probably use the check names as a guide to determine the purpose of most checks. For example, CO\_MSC\_p1 checks “priority one” minimum CO spacing, M1\_MWC\_p2 checks “priority two” M1 minimum width, and PO\_CO\_MOC\_p1 checks PO enclosure of CO.
14. Select check M1\_MSC\_p1.

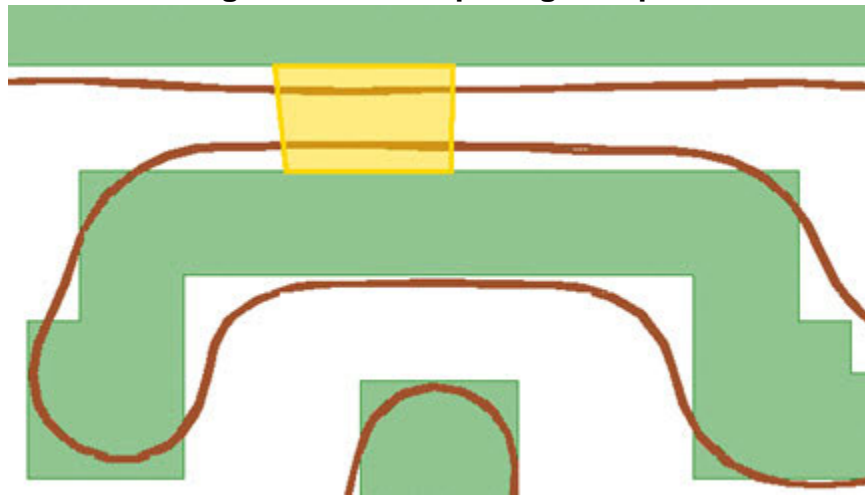
A description of the check along with some fix suggestions appears in the lower Calibre RVE frame:


```
RVE Priority: 910
Minimum Space Violation : Improve symmetry. If feature is dense,
increase spacing. Reduce surrounding feature width.
IL: lfd_M1_MSC_property_p1
```

15. Change the Calibre RVE highlight zoom factor.
  - a. Select **Setup > Options** to display the Options tab.
  - b. Verify the **Highlighting** category is selected.
  - c. Specify 0.2 in the “Zoom to highlights by” field.
  - d. Click **Apply** and then close the Options tab.
16. On the **lfd\_checks.rdb** tab, double-click the first result for the M1\_MSC\_p1 minimum spacing check.

The first M1 spacing hotspot is highlighted in Calibre DESIGNrev as shown in [Figure 3-7](#) (if the M1 PV-bands are not still highlighted, highlight the PV-bands as described earlier in this example).

**Figure 3-7. LFD Spacing Hotspot**

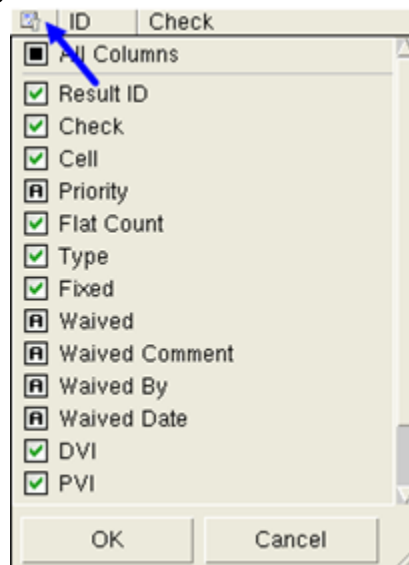


17. In the Calibre RVE Results view, if the results are not displayed in tabular format, click the **Select result view modes**  icon and choose **Details**.

Result details are now displayed in tabular format.

18. Click the icon in the column header to display the dropdown list from which you can select the columns to display in the results view as shown in [Figure 3-8](#).

**Figure 3-8. Selecting Columns in Calibre RVE to Display LFD Results**



19. Expand the Calibre RVE window so that the PVI column is visible.

The PVI, or process variability index, is a measure of how stable the process is across all results. The PVI value can range from zero (theoretical optimum value) to one (worst case value). Lower PVI values are more desirable. You can use the PVI values to help prioritize which hotspots to address first.

20. Click on the PVI column heading until the arrow is pointing up, indicating the PVI column is sorted in descending order. Note that the maximum PVI value is 0.553435, and the minimum PVI value is 0.492218 as shown in [Figure 3-9](#).

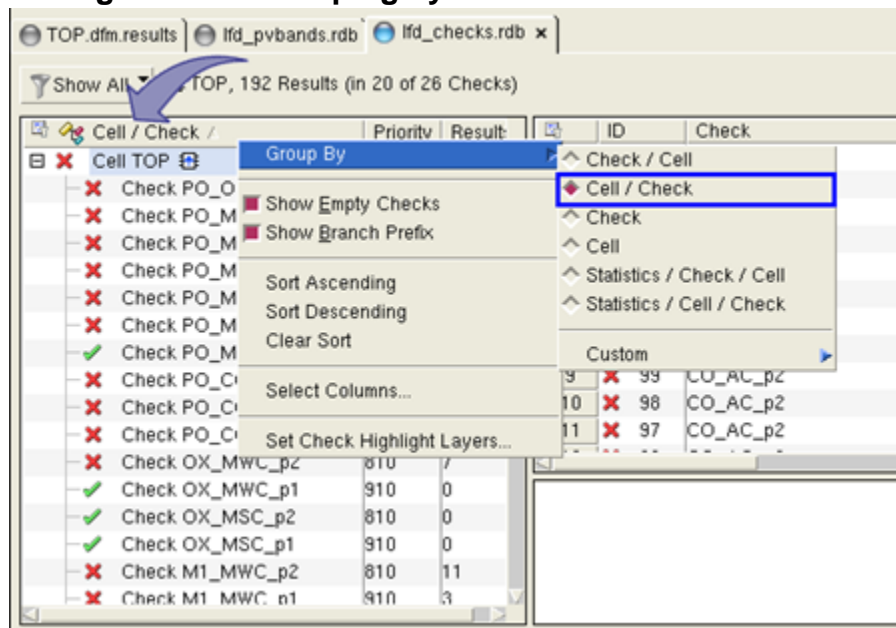
**Figure 3-9. Displaying LFD PVI Results Column in Calibre RVE**

ID	Check	Cell	Priority	Flat	T	F	PVI	Vertices	Coordi
1	64 M1_MSC_p1	TOP	910	1	P	0	0.553435	4	(5.7010
2	66 M1_MSC_p1	TOP	910	1	P	0	0.539929	4	(8.2990
3	65 M1_MSC_p1	TOP	910	1	P	0	0.534300	4	(5.7010
4	70 M1_MSC_p1	TOP	910	1	P	0	0.533650	4	(8.7155
5	63 M1_MSC_p1	TOP	910	1	P	0	0.515253	4	(4.9444
6	69 M1_MSC_p1	TOP	910	1	P	0	0.506178	4	(8.5803
7	68 M1_MSC_p1	TOP	910	1	P	0	0.498130	4	(8.1294
8	67 M1_MSC_p1	TOP	910	1	P	0	0.497377	4	(6.3398
9	71 M1_MSC_p1	TOP	910	1	P	0	0.495535	4	(9.2522
10	72 M1_MSC_p1	TOP	910	1	P	0	0.492218	4	(9.3919

21. Click the **Clear All Highlights** icon to clear all highlights.
22. Right-click the column header for the list of checks or cells to display the popup menu and select **Group By > Cell / Check** as shown in [Figure 3-10](#).

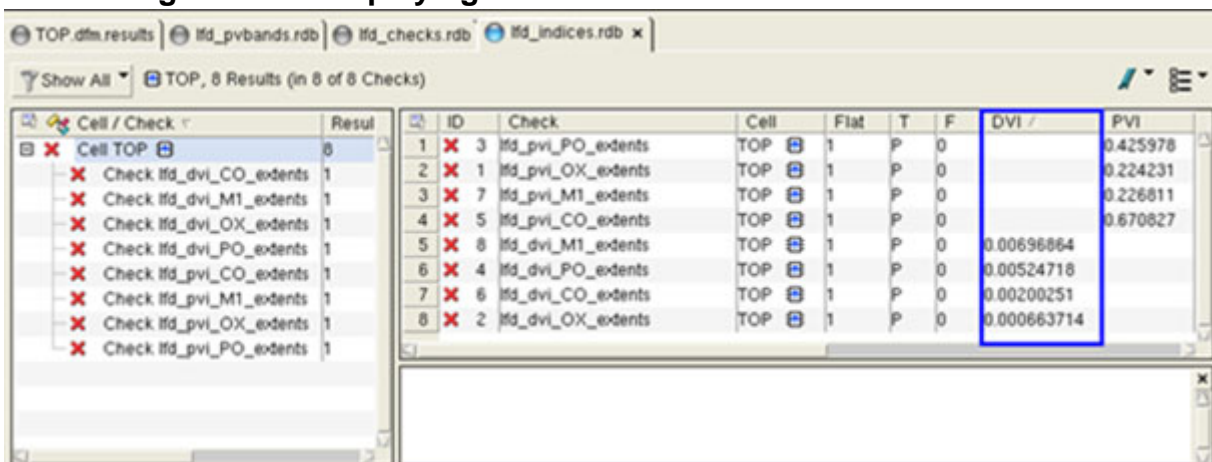


**Figure 3-10. Grouping by Cell/Check in Calibre RVE**



23. Click the **TOP.dfm.results** tab.
24. Click the **Open side RDBs** icon and select `./lfd_results/lfd_indices.rdb` from the menu.  
This opens the **lfd\_indices.rdb** tab.
25. Select the TOP entry in the **Cell/Check** frame, labeled “Cell TOP”.
26. Click the icon in the column header to display the dropdown list for selecting the columns to display and select **DVI**.
27. Click on the DVI column heading until the arrow is pointing up, indicating the DVI column is sorted in descending order as shown in [Figure 3-11](#).

**Figure 3-11. Displaying LFD DVI Results Column in Calibre RVE**



The DVI, or Design Variability Index, is the ratio of failure area to design area for a given layer and area. DVI values can range from zero to one. Smaller DVI values are preferable.

When trying to eliminate LFD hotspots in the layout, focus first on those areas or layers where the DVI and PVI values are larger.

28. Close the Calibre RVE and Calibre DESIGNrev sessions.

## Results

You have now used Calibre DESIGNrev and Calibre RVE to work with the results created by a Calibre LFD job. The key outputs from Calibre LFD include the results files you have studied in this example:

- *lfd\_pvbands.rdb* — Polygons that represent the predicted printed image edges based on process window data.
- *lfd\_checks.rdb* — The list of checks that generated results.
- *lfd\_indices.rdb* — Indices reporting on the design robustness across the lithographic process window.

# Chapter 4

## Extracting Parasitics

---

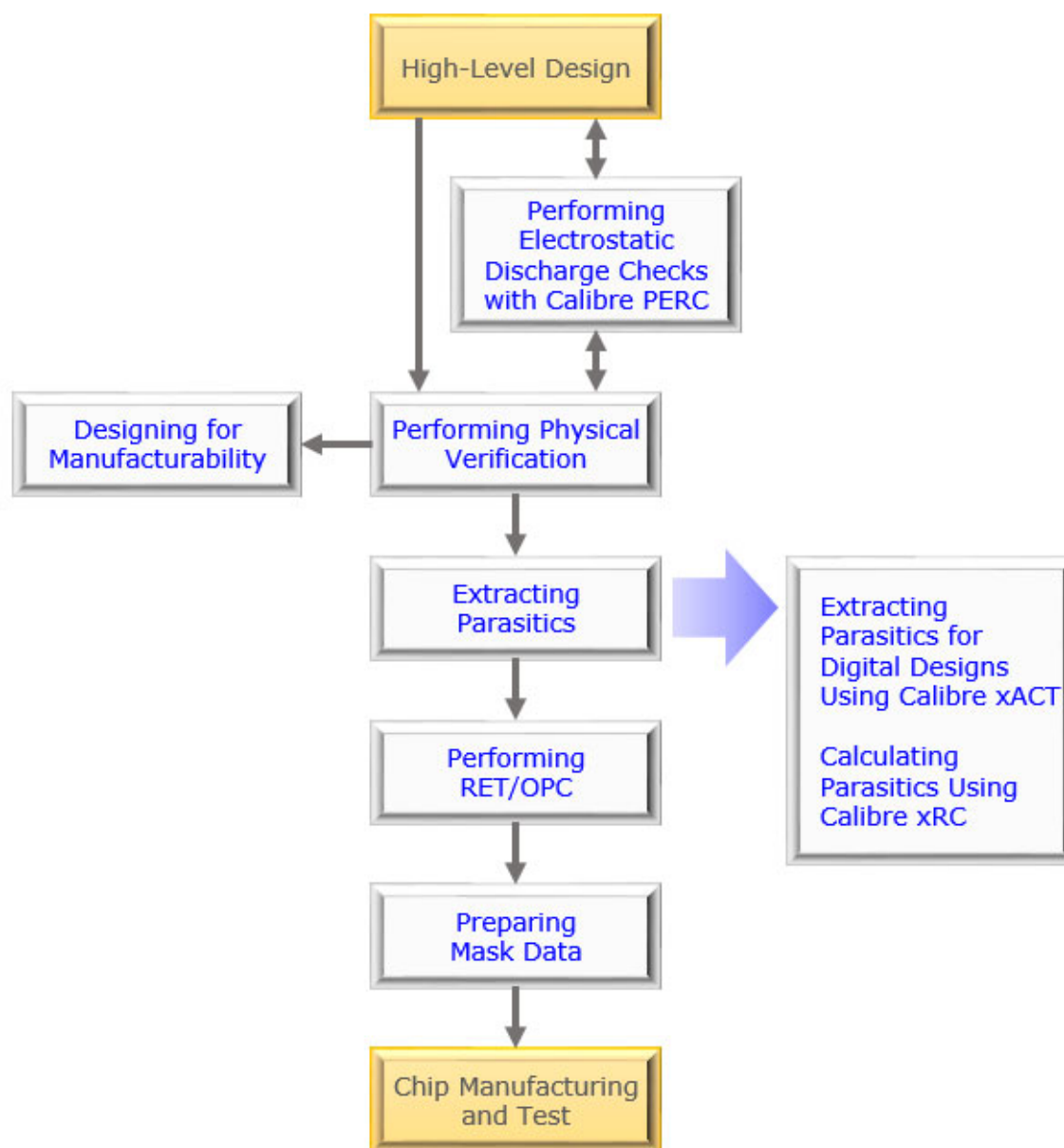
Use Calibre extraction tools to calculate the parasitics on your design. Parasitic effects can slow down signals, add noise, or cause hot spots in your design, among other problems. You can run parasitic extraction after your layout passes LVS. Calibre extraction tools make use of connectivity data. If the layout is not electrically correct, the parasitic results will not apply to the final design either.

<b>Parasitic Extraction Design Flow .....</b>	<b>59</b>
<b>Calibre Parasitic Extraction (PEX) Tool Reference.....</b>	<b>60</b>
<b>Extracting Parasitics for Digital Designs Using Calibre xACT.....</b>	<b>62</b>
<b>Calculating Parasitics Using Calibre xRC.....</b>	<b>67</b>

### Parasitic Extraction Design Flow

Figure 4-1 shows some of the possible parasitic checks as described in the procedures in this module.

Figure 4-1. Parasitic Extraction Design Flow



## Calibre Parasitic Extraction (PEX) Tool Reference

Supporting documentation is available for each licensed Calibre product and, in some cases, training is also available. There may be other products that are closely related to the use of that tool.

Table 4-1 provides a comprehensive list and short description of the licensed Calibre products associated with parasitic extraction, in addition to information on related documentation, training, and products.

**Table 4-1. Calibre Parasitic Extraction Tool Reference**

Tool	Description	Related Documentation	Related Training	Related Products
Calibre® Interactive™	A GUI interface for nmDRC, nmLVS, PERC, xRC, and DFM. Also calls Calibre RVE.	<i>Calibre Interactive User's Manual</i>		Calibre RVE
Calibre® xACT	A tool used to generate parasitic resistance and capacitance netlists for cell-based, digital design flows and transistor-level design flows.	<i>Calibre xACT User's Manual</i> ----- <i>SVRF Manual</i>		
Calibre® xACT 3D	A tool used to generate parasitic netlists and reports using fast fieldsolver technology for capacitance extraction.	<i>Calibre xACT User's Manual</i> ----- <i>SVRF Manual</i>		
Calibre® xACT 3D Reference	A reference level 3-D fieldsolver extraction solution.	<i>Calibre xACT User's Manual</i>		
Calibre® xRC™	A tool used to extract parasitic resistance and capacitance for interconnect.	<i>Calibre xRC User's Manual</i> ----- <i>SVRF Manual</i>	<a href="#">Calibre xRC Parasitic Extraction</a>	
Calibre® xL	A tool used to extract mutual and self inductance for interconnect.	<i>Calibre xL User's Manual</i> ----- <i>SVRF Manual</i>		

**Table 4-1. Calibre Parasitic Extraction Tool Reference (cont.)**

Tool	Description	Related Documentation	Related Training	Related Products
Calibre® RVE™	A GUI used to view and debug output from Calibre tools.	<i>Calibre RVE User's Manual</i>		
xCalibrate	A tool used to describe process technology information and generate SVRF rule files containing capacitance and resistance statements.	<i>xCalibrate Batch User's Manual</i>		

## Extracting Parasitics for Digital Designs Using Calibre xACT

You can extract a full-chip digital design netlist for a digital design. The result of this extraction is a SPEF netlist that can be used with static timing analysis tools.

### Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- A Calibre xACT license available.
- The Calibre software must be installed to run Calibre xACT.
- An SVRF rule file containing information used to control the extraction, such as the name of the SPEF file (using PEX Netlist SPEF), or other optional settings such as temperature.
- Calibre xACT automatically excludes power and ground nets when using LEF and DEF as input. It identifies them with the keywords USE POWER and USE GROUND in the DEF file.
- The following inputs are required to perform this example:
  - Placed and routed DEF design data, *Shared/Design/DEF/top.def*
  - LEF technology file, *Shared/Design/LEF/tech.lef*

- LEF cell library, *Shared/Design/LEF/gscl45nm.lef*
- LEF macro files:
  - Shared/Design/LEF/ADC\_5bit\_sc\_5.lef*
  - Shared/Design/LEF/myram.lef*
  - Shared/Design/LEF/PADS.lef*
- LVS rule file, *Shared/Design/lvs.rules*
- Layer definition file, *Shared/Design/layer.inc*
- Calibre parasitic extraction rule files, *pex\_rules.R*, *pex\_rules.C*, and *pex\_rules.xact*.

## Procedure

1. In a UNIX or Linux terminal window, change directory to *Calbr\_qs\_ekit/Shared/Design*.  

```
cd <path>/Calbr_qs_ekit/Shared/Design
```
2. Examine the contents of the *Design* directory. You should see the following directories:
  - *DEF* directory — Contains the design DEF file.
  - *LEF* directory — Contains the design LEF files.
3. Change directory to *Calbr\_qs\_ekit/Shared/Design/DEF*.  

```
cd <path>/Calbr_qs_ekit/Shared/Design/DEF
```
4. Examine the contents of the *DEF* directory. You should see the *top.def* file. This file contains the design layout information.
5. Change directory to *Calbr\_qs\_ekit/Shared/Design/LEF*.  

```
cd <path>/Calbr_qs_ekit/Shared/Design/LEF
```
6. Examine the contents of the *LEF* directory. You should see the following files:
  - *tech.lef* — The technology LEF file that contains design technology information. This file should appear first in the LAYOUT PATH statement specification.
  - *gscl45nm.lef* — LEF cell library file.
  - *dac6\_op2.lef* — LEF macro file.
  - *ADC\_5bit\_sc\_5.lef* — LEF macro file.
  - *myram.lef* — LEF macro file.
  - *PADS.lef* — LEF macro file.
7. Change directory to *Calbr\_qs\_ekit/Module4/Rules*.  

```
cd <path>/Calbr_qs_ekit/Module4/Rules
```

8. Examine the contents of the *Rules* directory. You should see the following files:
  - *pex\_rules.C* — Contains the capacitance parasitic extraction rules.
  - *pex\_rules.R* — Contains the resistance parasitic extraction rules.
  - *pex\_rules.xact* — Typically, the *pex\_rules.xact* file is required only if provided in your foundry's PDK.

The *pex\_rules.C*, *pex\_rules.R*, and *pex\_rules.xact* files are encrypted and generated from a process technology profile. These files are typically generated and provided by a foundry.

9. Change directory to *Calbr\_qs\_ekit/Module4*.

```
cd <path>/Calbr_qs_ekit/Module4
```

10. Create a symbolic link to the design data directory containing the LEF and DEF files:

```
ln -s ../Shared/Design
```

11. Create and save a rule file called *xact\_layers.svrf* in the *Module4* directory. Using a text editor include the following statements in the file:

```
PEX MAP metal1 M1
PEX MAP metal2 M2
PEX MAP metal3 M3
PEX MAP metal4 M4
PEX MAP metal5 M5
PEX MAP metal6 M6
PEX MAP metal7 M7
PEX MAP metal8 M8
PEX MAP metal9 M9
PEX MAP metal10 M10

PEX MAP via1 V1
PEX MAP via2 V2
PEX MAP via3 V3
PEX MAP via4 V4
PEX MAP via5 V5
PEX MAP via6 V6
PEX MAP via7 V7
PEX MAP via8 V8
PEX MAP via9 V9

PEX MAP contact CO
```



```

PEX DEF MAP metal1 M1
PEX DEF MAP via1 V1
PEX DEF MAP metal2 M2
PEX DEF MAP via2 V2
PEX DEF MAP metal3 M3
PEX DEF MAP via3 V3
PEX DEF MAP metal4 M4
PEX DEF MAP via4 V4
PEX DEF MAP metal5 M5
PEX DEF MAP via5 V5
PEX DEF MAP metal6 M6
PEX DEF MAP via6 V6
PEX DEF MAP metal7 M7
PEX DEF MAP via7 V7
PEX DEF MAP metal8 M8
PEX DEF MAP via8 V8
PEX DEF MAP metal9 M9
PEX DEF MAP via9 V9
PEX DEF MAP metal10 M10

```

The *xact\_layers.svrf* rule file specifies two different types of mapping statements.

- PEX Map maps the calibrated rule file layer names to the LVS layer names so that the layers can inherit the parasitic properties.
  - PEX DEF Map maps the layer names in the LEF/DEF database to the LVS layer names in order to establish connectivity. The layer mappings for these statements should use the metal layer names found in the LEF technology file.
12. Save and close the *xact\_layers.svrf* file. You will use the Include statement to include this file in your top-level rule file created in the next step.
  13. Create and save a top-level rule file called *xact.svrf* in the *Module4* directory. Using a text editor include the following statements in the file:

```

// Design layout information
LAYOUT PRIMARY "top"
LAYOUT PATH    "./Design/LEF/tech.lef"
               "./Design/LEF/gscl45nm.lef"
               "./Design/LEF/PADS.lef"
               "./Design/LEF/myram.lef"
               "./Design/LEF/dac6_op2.lef"
               "./Design/LEF/ADC_5bit_sc_5.lef"
               "./Design/DEF/top.def"

// Design netlist information
PEX NETLIST "netlist.spef.gz" SPEF LAYOUTNAMES MAPNAMES NOINSTANCEX
PEX NETLIST CAPACITANCE UNIT fF
PEX NETLIST ESCAPE CHARACTERS OFF

// Setup verification database location
MASK SVDB DIRECTORY "svdb" XACT

```

```
// Layout uses LEF and DEF files
LAYOUT SYSTEM LEFDEF
LAYOUT CASE YES
SOURCE CASE YES

// Extraction options
PEX LEF EXTRACT CELL OBSTRUCTIONS YES
PEX DEF EXTRACT BLOCKAGES NO
PEX XACT FILL FLOATING
// temperature control
PEX EXTRACT TEMPERATURE 27

// Reduction options
PEX REDUCE CC ABSOLUTE 3 RATIO 0.03
PEX REDUCE DIGITAL YES

// location of DFM database
DFM DATABASE "dfmdb" OVERWRITE

// Comment out the following line
// if Calibre complains that UNIT CAPACITANCE is already defined
// or change the units if
// the results are incorrect by a factor of 1000
UNIT CAPACITANCE fF

// LVS and parasitic extraction rule files
INCLUDE ./Design/lvs.rules
INCLUDE ./Design/layer.inc

INCLUDE ./xact_layers.svrf
INCLUDE ./Rules/pex_rules.C
INCLUDE ./Rules/pex_rules.R
INCLUDE ./Rules/pex_rules.xact
```

In the *xact.svrf* rule file, the following SVRF statements are needed when running the Calibre xACT digital extraction flow:

- Layout System LEFDEF specifies that the layout is a set of LEF and DEF files.
- PEX LEF Extract Cell Obstructions YES specifies to include obstruction geometries as layout objects and treat them as ground when calculating parasitic capacitance.
- PEX DEF Extract Blockages NO specifies to exclude DEF placement and routing blockage areas from parasitic extraction.
- PEX XACT Fill FLOATING specifies to model metal fill as floating. This is the default behavior in the Calibre xACT digital flow if this statement is not specified.
- PEX Extract Temperature modifies the temperature dependence, which can affect resistance.
- PEX Netlist generates a netlist with parasitic elements and places it in the specified file.

For more information about these statements, refer to the *Standard Verification Rule Format (SVRF) Manual*.

14. Save and close the *xact.svrf* file.
15. To perform extraction for resistance with distributed capacitance and coupled capacitance between nets, and generate the netlist, enter the following command:

```
calibre -xact -rcc -turbo xact.svrf |& tee log.xact
```

The string “|& tee log.xact” directs the output from standard output and standard error to be written to a *log.xact* file in the current directory.

## Results

A successful Calibre xACT run issues messages similar to the following in the *log.xact* file:

```
=====
CALIBRE xACT WARNING/ERROR Summary
-----
xACT Warnings = 3
xACT Errors = 0
=====

---EXTRACTION TOTAL CPU TIME=356 REAL TIME=359 PEAK MEMORY=899 MB
ELAPSED TIME=380
```

At the end of the run, you should have the following files in the *Module4* directory:

- *log.xact* — The log file from the run containing warnings, errors, and other messages such as runtime and memory usage.
- *svdb* — The directory containing intermediate files used during extraction.

Several subdirectories and files are also generated and placed in the *svdb* directory. You do not directly access or use this information for Calibre xACT.

- *netlist.spef.gz* — The netlist generated by Calibre xACT.
- *dfmddb* — The directory containing the physical design data derived from the LEF/DEF information.

After running parasitic extraction, you may choose to run your design through your static timing analyzer. You can use the *netlist.spef.gz* netlist to perform timing analysis of your design. Since this file is in SPEF format, you will need a compatible timing analysis product.

## Calculating Parasitics Using Calibre xRC


You can extract a transistor-level netlist for an ADC circuit. The result of this extraction is an HSPICE netlist which can be used in a SPICE-compatible simulator to re-simulate the design.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- Licenses for the following products: Calibre xRC.
- The following inputs are required to perform this example:
  - OASIS layout, *Shared/Design/fullchip.oas*
  - SPICE netlist, *Shared/Design/source.net*
  - LVS rule file, *Shared/Design/lvs.rules*
  - Layer definition file, *Shared/Design/layer.inc*
  - Calibre parasitic extraction rule files, *Module4/Rules/pex\_rules.R* and *Module4/Rules/pex\_rules.C*
- If your *Module4* directory contains the *svdb* directory and *dfmdb* directory from a Calibre xACT digital flow run you must remove them:

```
rm -rf svdb
rm -rf dfmdb
```

### Caution

 The *svdb* and *dfmdb* directories contain data for specific use by the tool that generated it and is not interchangeable. Do not use the *svdb* or *dfmdb* directory generated by Calibre xACT tool when running Calibre xRC.

---

## Procedure

1. In a UNIX or Linux terminal window, change directory to *Calbr\_qs\_ekit/Shared/Design*.

```
cd <path>/Calbr_qs_ekit/Shared/Design
```
2. Examine the contents of the *Design* directory. You should see the following files:
  - *layer.inc* — Specifies the layers used in the layout and rule files.
  - *lvs.rules* — Contains the LVS verification rules, used in the connectivity extraction step (`calibre -lvs`).
3. Change directory to *Calbr\_qs\_ekit/Module4/Rules*.

```
cd <path>/Calbr_qs_ekit/Module4/Rules
```
4. Examine the contents of the *Rules* directory. You should see the following files:
  - *pex\_rules.C* — Contains the capacitance parasitic extraction rules.
  - *pex\_rules.R* — Contains the resistance parasitic extraction rules.

The *pex\_rules.C* and *pex\_rules.R* files are encrypted and generated from a process technology profile. These files are typically generated by a foundry.

---

**Note**



The *pex\_rules.xact* file found in the *Rules* directory is not used by Calibre xRC.

---

5. Change directory to *Calbr\_qs\_ekit/Module4*.

```
cd <path>/Calbr_qs_ekit/Module4
```

6. Create and save a Calibre xRC rule file called *xrc.rules* in the *Module4* directory. Using a text editor, include the following statements in the file:

```
// design layout information
LAYOUT PATH "../Shared/Design/fullchip.oas"
LAYOUT PRIMARY "ADC_5bit_sc_5"
LAYOUT SYSTEM OASIS

// design netlist information
SOURCE PATH "../Shared/Design/source.net"
SOURCE PRIMARY "ADC_5bit_sc_5"
SOURCE SYSTEM SPICE

// setup verification database location
MASK SVDB DIRECTORY "../svdb" QUERY XRC

// specify output reports
LVS REPORT "lvs.report"
PEX REPORT "pex.report" SOURCENAMES

// specify netlisting for extraction
PEX NETLIST "ADC_5bit_sc_5_RC.hspc" HSPICE 1 SOURCEBASED SHORTPINNAMES
    PRUNE

// nets to exclude from extraction
PEX EXTRACT EXCLUDE SOURCENAMES TOPLEVEL "vdd" "gnd!"

// specify LVS and parasitic extraction rule files
#DEFINE PEX
INCLUDE "../Shared/Design/layer.inc"
INCLUDE "../Shared/Design/lvs.rules"
INCLUDE "../Rules/pex_rules.C"
INCLUDE "../Rules/pex_rules.R"

// parasitic extraction statements
CAPACITANCE ORDER nsrhdrn psrhdrn ipoly M1 M2 M3 M4 M5 M6 M7 M8 M9 M10
PEX IGNORE RESISTANCE presbody nwresbody respin
PEX IGNORE CAPACITANCE ALL cappos capneg
PEX MAP active ncont pcont
PEX MAP metal1 M1
PEX MAP metal2 M2
PEX MAP metal3 M3
PEX MAP metal4 M4
PEX MAP metal5 M5
PEX MAP metal6 M6
PEX MAP metal7 M7
PEX MAP metal8 M8
PEX MAP metal9 M9
PEX MAP metal10 M10
```

In the rule file, the following SVRF statements are specifically for running the Calibre xRC tool:

- PEX Report generates a report for parasitic results. The report includes coupling capacitors between nets in the design.

- PEX Netlist generates a netlist with parasitic elements and places it in the specified file.
- PEX Extract Exclude specifies to exclude the named nets from parasitic extraction.
- Capacitance Order defines the vertical order of layers from bottom to top. This is used in parasitic capacitance extraction.
- PEX Ignore Resistance specifies layers to ignore when calculating parasitic resistance.
- PEX Ignore Capacitance specifies capacitance effects to ignore when calculating parasitic capacitance.
- PEX Map specifies the mapping between a calibrated layer and one or more physical layers.

For more information about these statements, refer to the *Standard Verification Rule Format (SVRF) Manual*.

7. Save and close the file.
8. Enter the following command to run Calibre nmLVS-H and build the database for intentional devices:

```
calibre -lvs -hier -spice ./svdb/ADC_5bit_sc_5.sp xrc.rules \
    |& tee lvs.log
```

Adding “|tee lvs.log” directs the output to the shell command window as well as the *lvs.log* file located in the current directory.

At the end of the run, you should have the following files in the *Module4* directory:

- *lvs.log* — Log file from the Calibre nmLVS-H run.
  - *lvs.report* — LVS Report file, detailing the comparison results.
  - *lvs.report.ext* — Report file from the device and net extraction for LVS. The report details problems with shorts, opens, and bad devices. If LVS is clean, this file only contains a header block.
  - *svdb* — Directory containing files and other subdirectories used during intentional device and net extraction. The information in this directory is also used during parasitic extraction.
  - *svdb/ADC\_5bit\_sc\_5.sp* — Netlist generated from the layout.
9. Extract the parasitic effects from interconnects in the design by executing the Calibre xRC tool as follows:

```
calibre -xrc -pdb -rc xrc.rules |& tee pdb.log
```

Output from this step includes the *pdb.log* file, which contains a transcript of messages from running this step. Several subdirectories and files are also generated and placed in the *svdb* directory. You do not directly access or use this information.

10. Generate the parasitic netlist that includes resistance and capacitance as follows:


```
calibre -xrc -fmt -all xrc.rules |& tee fmt.log
```

The extracted parasitics are saved in the following files:

- *ADC\_5bit\_sc\_5\_RC.hspc* — Transistor-level HSPICE file, specified in the PEX Netlist SVRF statement found in the rule file.
- *ADC\_5bit\_sc\_5\_RC.hspc.pex* — Parasitic net models for each extracted net (subcircuit definitions). The parasitic net models contain parasitic capacitors and resistors.
- *ADC\_5bit\_sc\_5\_RC.hspc.ADC\_5BIT\_SC\_5.pxi* — Instances of the parasitic models found in the .pex file. It also contains parasitic resistors and capacitors.

---

**Tip**

 You can combine the shell command statements in Steps 8 through 10 by placing them in a shell script file. This file should be placed in the *Module4* directory.

---

## Results

A successful Calibre xRC transcript concludes with a count of errors and warnings as shown:

```
CALIBRE xRC WARNING / ERROR Summary
-----
xRC Warnings   = 1
xRC Errors     = 0
=====
```

After running parasitic extraction, you may choose to simulate your design. You can use the *ADC\_5bit\_sc\_5\_RC.hspc* netlist to perform parasitic re-simulation of your design with a compatible simulator.

You can also use the report file, *pex.report*, created by the formatter to identify the nets most affected by coupling capacitance.



# Chapter 5

## Performing RET/OPC

---

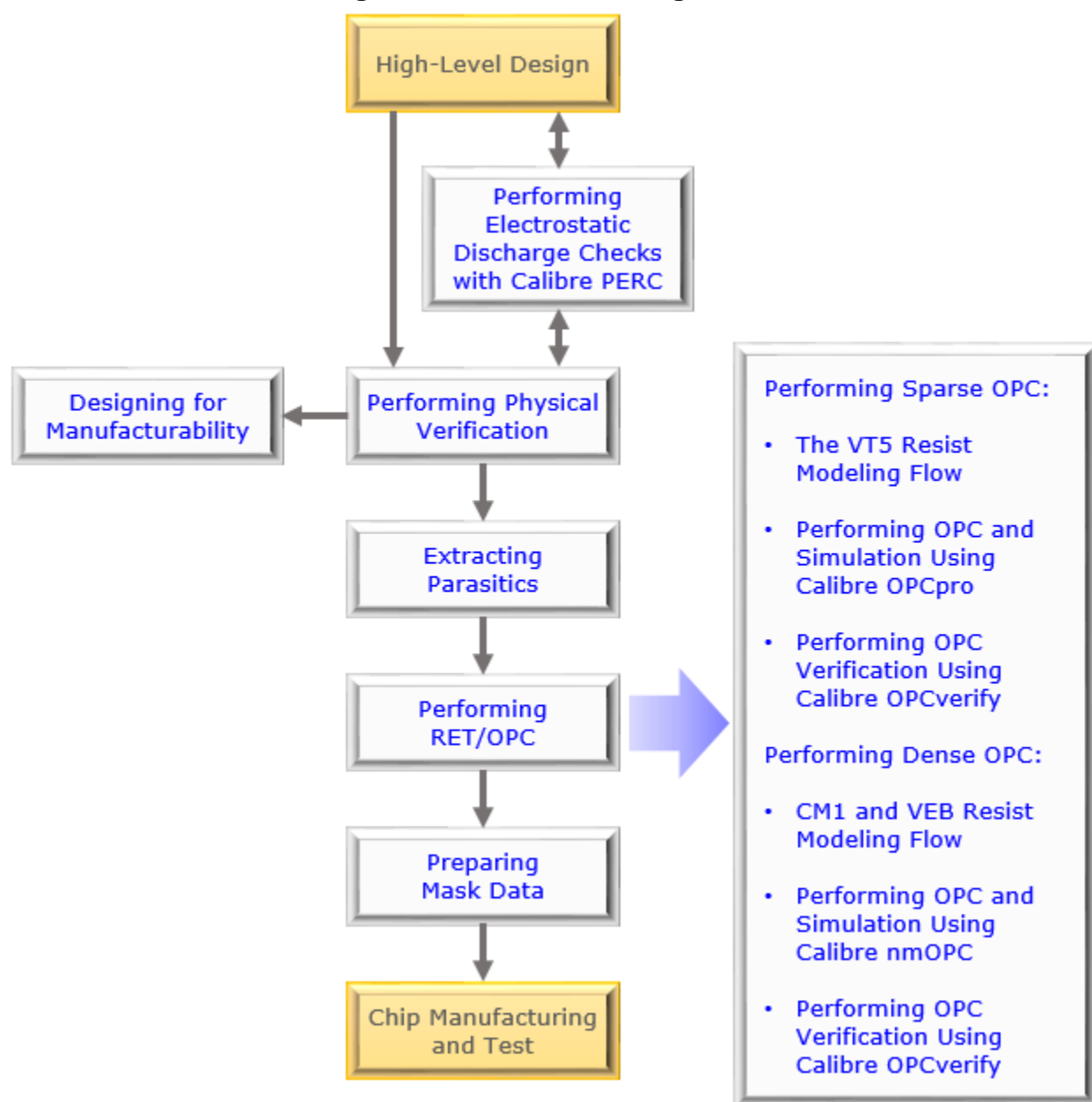
The resolution enhancement technology (RET) and optical process correction (OPC) flows are used to correct potential distortions on a photomask input layer introduced by the optical and resist processes. Calibre provides a suite of tools that can be applied to all stages of the RET process.

<b>RET/OPC Design Flow</b> .....	<b>73</b>
<b>Calibre Resolution Enhancement Technology (RET) Tool Reference</b> .....	<b>74</b>
<b>Sparse OPC and Dense OPC Process Flows</b> .....	<b>77</b>
<b>Performing Sparse OPC</b> .....	<b>79</b>
The VT5 Resist Modeling Flow .....	79
Performing OPC and Simulation Using Calibre OPCpro. ....	81
Performing Sparse OPC Verification Using Calibre OPCverify .....	86
<b>Performing Dense OPC</b> .....	<b>87</b>
CM1 and VEB Resist Modeling Flow .....	87
Performing OPC and Simulation Using Calibre nmOPC. ....	89
Performing OPC Verification Using Calibre OPCverify .....	95

## RET/OPC Design Flow

Figure 5-1 shows some of the OPC checks described in the procedures in this module.

Figure 5-1. RET/OPC Design Flow



## Calibre Resolution Enhancement Technology (RET) Tool Reference

Supporting documentation is available for each licensed Calibre product and, in some cases, training is also available. There may be other products that are closely related to the use of that tool.

Table 5-1 provides a comprehensive list and short description of the licensed Calibre products associated with resolution enhancement technology, in addition to information on related documentation, training, and products.

**Table 5-1. Calibre Resolution Enhancement Technology Tool Reference**

Tool	Description	Related Documentation	Related Training	Related Products
Calibre® EUV	Supports Extreme Ultraviolet Lithography (EUV) correction.	<i>Calibre nmOPC User's and Reference Manual</i>		Calibre OPCverify ----- Calibre pxOPC
Calibre® LITHOview™	Simulates based on model and design.	<i>Calibre WORKbench User's and Reference Manual</i>		
Calibre® Local Printability Enhancement (LPE)	Corrects OPC hotspots using pixel-based optimization.	<i>Calibre LPE User's and Reference Manual</i>		Calibre OPCverify ----- Calibre pxOPC ----- Calibre nmOPC
Calibre® Multi-Patterning™	Supports double- and triple-patterning, and self-aligned double patterning lithography operations to improve image resolution and contrast.	<i>Calibre Multi-Patterning User's and Reference Manual</i>		
Calibre® nmModelflow™	Creates optical and process models used for simulations.	<i>Calibre nmModelflow User's and Reference Manual</i>		Calibre WORKbench (Calibre nmModelflow is invoked from Calibre WORKbench)
Calibre® nmOPC™	Corrects optical and process distortions on the mask. This is used for dense OPC.	<i>Calibre nmOPC User's and Reference Manual</i>	<a href="#">Calibre RET/OPC Basics</a>	

Table 5-1. Calibre Resolution Enhancement Technology Tool Reference (cont.)

Tool	Description	Related Documentation	Related Training	Related Products
Calibre® nmSRAF™	Inserts sub-resolution assist features into a design using a template.	<i>Calibre nmSRAF User's and Reference Manual</i>	<a href="#">Calibre RET/OPC Basics</a>	Calibre OPCsbar ----- Calibre pxOPC
Calibre® OPCpro™	Corrects optical and process distortions on the mask. This is used for sparse OPC.	<i>Calibre OPCpro User's and Reference Manual</i>		
Calibre® OPCsbar™	Places scattering bars.	<i>Calibre OPCsbar User's and Reference Manual</i>		Calibre OPCpro
Calibre® OPCverify™	Verifies OPC corrections and generates simulation images. This can be used for both dense and sparse OPC.	<i>Calibre OPCverify User's and Reference Manual</i>	<a href="#">Calibre RET/OPC Basics</a>	Calibre WORKbench ----- Calibre RVE
Calibre® ORC™	Verifies that corrections do not violate original design rules. This is used for sparse OPC.	<i>Calibre OPCpro User's and Reference Manual</i>		Calibre OPCpro
Calibre® PRINTimage™	Generates simulation images. This is used for sparse OPC.	<i>Calibre OPCpro User's and Reference Manual</i>		Calibre OPCpro ----- Calibre WORKbench
Calibre® PSMgate™	Performs phase assignment for all types of phase shifting masks.	<i>Calibre PSMgate User's Manual</i>		

**Table 5-1. Calibre Resolution Enhancement Technology Tool Reference (cont.)**

<b>Tool</b>	<b>Description</b>	<b>Related Documentation</b>	<b>Related Training</b>	<b>Related Products</b>
Calibre® pxOPC™	Performs full-chip mask optimization of small and medium-sized layouts.	<i>Calibre pxOPC User's and Reference Manual</i>		Calibre LPE
Calibre® TDopc™	Creates rules for OPC that apply corrections to individual polygon edges based on properties of those edges.	<i>Calibre Rule-Based OPC User's and Reference Manual</i>		Calibre OPCpro ----- Calibre OPCsbar ----- Calibre ORC
Calibre® WORKbench™	Creates optical and process models used in simulations.	<i>Calibre WORKbench User's and Reference Manual</i>		
RET Flow Tool	Provides an interface in Calibre WORKbench to access RET-related tools.	<i>Calibre WORKbench: RET Flow Tool User's Manual</i>		

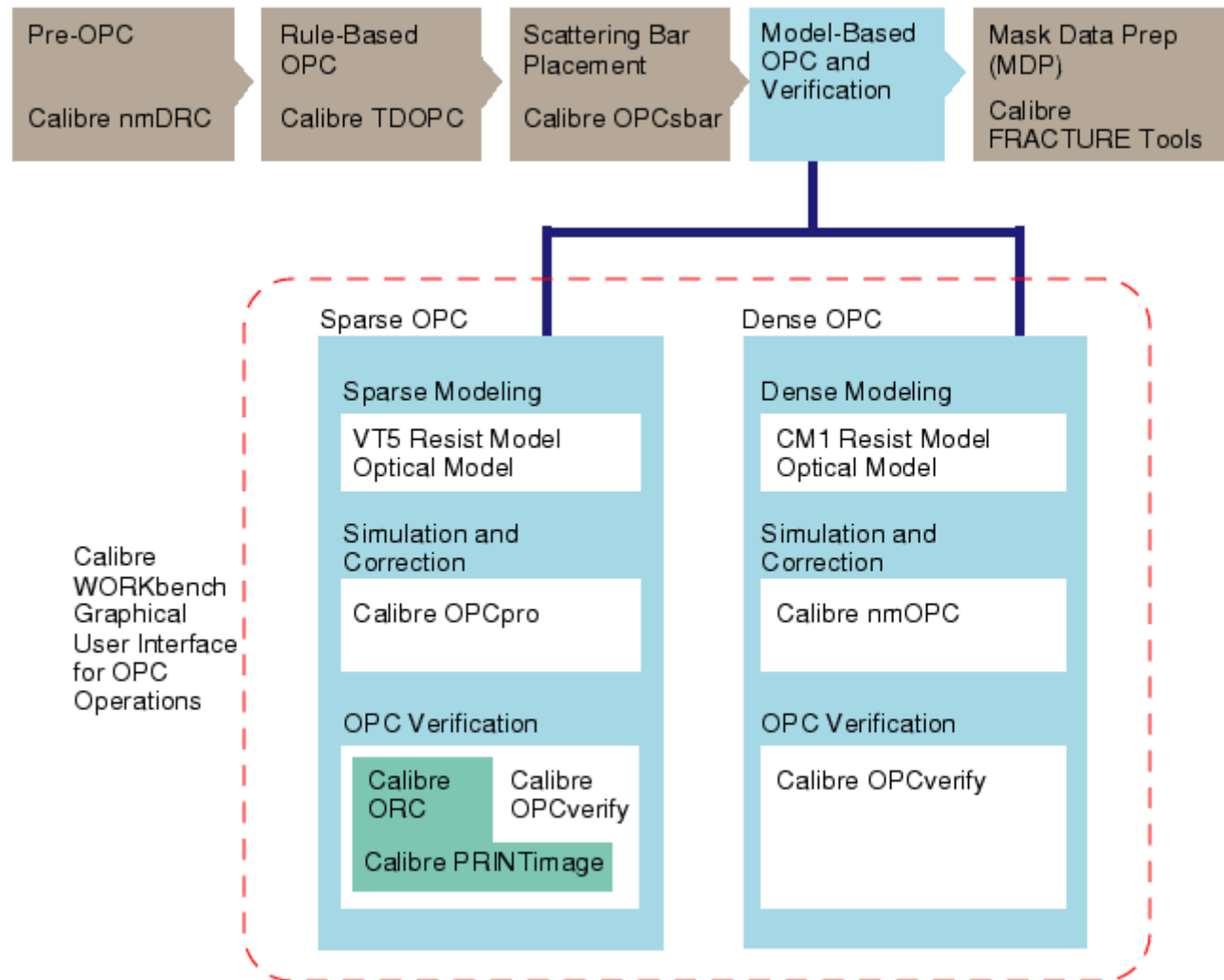
## Sparse OPC and Dense OPC Process Flows

There are two basic flows for OPC. Each uses a different method to simulate and correct the effects of the lithographic process.

- **Sparse OPC** — This flow is performed using a shape-based or *sparse* simulation engine to calculate the wafer image contour.
- **Dense OPC** — This flow is performed using a pixel-based or *dense* simulation engine to calculate the wafer image contour. This flow was developed for the 65 nm technology node and below and can be integrated into the design to mask flow in the same way as sparse OPC.

The sparse and dense OPC process flows are summarized in [Figure 5-2](#).

Figure 5-2. OPC Process Flow



# Performing Sparse OPC

The sparse OPC flow consists of three key processes.

- [The VT5 Resist Modeling Flow](#) — This is typically performed by the foundry or sometimes the manufacturer of the lithographic equipment.
- [Performing OPC and Simulation Using Calibre OPCpro](#) — This is done by the foundry or by the chip design team.
- [Performing OPC Verification Using Calibre OPCverify](#) — This is usually done by the same person who ran OPC.

Use sparse OPC (also known as OPCpro) when working with processes above 65 nm or with simpler layers such as interconnect in a smaller process.


<b>The VT5 Resist Modeling Flow .....</b>	<b>79</b>
<b>Performing OPC and Simulation Using Calibre OPCpro.....</b>	<b>81</b>
<b>Performing Sparse OPC Verification Using Calibre OPCverify .....</b>	<b>86</b>

## The VT5 Resist Modeling Flow

The VT5 resist modeling flow uses the generic optical model and creates a VT5 resist model. The VT5 resist model is specific to sparse simulations. These two models (optical and resist) are required for image simulations for Calibre OPCpro and Calibre OPCverify operations.

Although you can create default models using the Optical Model and VT5 Center tools, using optical and VT5 resist models calibrated to the data measurements provides a far more accurate representation. This process is also known as “fitting” the measurement data to the test chip.

### Tip

 Because modeling is an iterative and hardware-intensive process, creating a model is not described in this Quick Start guide. This section only briefly describes the modeling process. Pre-generated models are provided in the *Modeling* directory.

The flow for VT5 sparse resist modeling is as follows:

1. Collect sample data as described in “[Test Pattern Measurement](#).”
2. Create an acceptable optical model by iteratively performing “[Creation and Optimization of Optical Models for VT5 Models](#).”
3. Use the optimized optical model for [VT5 Resist Model Calibration](#). This is also an iterative process that is performed until the most acceptable VT5 model is found.

Steps 2 and 3 can also be performed using a script, as described in “[Usage of modelflow as a Batch Modeling Command](#).”

## Test Pattern Measurement

Use the Calibre WORKbench Test Pattern tool to create a chip with known geometries. Sending this chip through your lithography process allows collection of critical dimension (CD) measurements on the line widths of structures and the spaces between structure edges. This is the measurement data that Calibre WORKbench uses to calibrate models. Both the optical and resist models use this data, which must be entered in a sample data spreadsheet file.

Test pattern creation tasks and commands are described in the following chapters in the *Calibre WORKbench User's and Reference Manual*:

- Chapter 4, “Test Pattern Creation”
- Chapter 5, “Test Pattern Reference”

## Creation and Optimization of Optical Models for VT5 Models

Optical models are usually created in the Calibre WORKbench Optical Model tool. They simulate the illumination source, ambient medium, and the film stack.

Optical models are optimized using the optical characteristics you select as the most important for matching the test pattern data on an aerial image simulation.

Optical models are optimized separately from resist models. The same optical model can be used in VT5 modeling and CM1 modeling.

Optical model tasks and commands are described in the following chapters in the *Calibre WORKbench User's and Reference Manual*:

- Chapter 6, “Optical Model Creation”
- Chapter 8, “Optical Model Reference”

## VT5 Resist Model Calibration

VT5 resist models are calibrated in the VT5 Center using the simulated properties of the optical model combined with the measurements taken on a test pattern chip to extrapolate resist etching effects. VT5 resist models are calibrated separately from optical models after the optical model calibration is complete.

VT5 resist model tasks and commands are described in the following chapters in the *Calibre WORKbench User's and Reference Manual*:

- Chapter 9, “VT5 Resist Model Creation”
- Chapter 11, “VT5 Resist Model Reference”



## Usage of modelflow as a Batch Modeling Command

While Calibre WORKbench provides the VT5 Center tool to give you a GUI-driven way to optimize and calibrate optical and resist models, the VT5 Center GUI uses your choices to generate input for an underlying batch command (modelflow). Some users may want to skip the GUI altogether and just work with modelflow directly; VT5 Center also allows you to export a set of GUI choices as script input to modelflow.

The modelflow syntax is described in the “modelflow Version 1 Reference” chapter of the *Calibre WORKbench User’s and Reference Manual*.

## Performing OPC and Simulation Using Calibre OPCpro

In the following steps, you run Calibre OPCpro on a 45 nm design. This produces an output OPC layer. Typically, you analyze and fine-tune your OPC output before it is used to create reticle masks (the MDP step). This module, however, covers only how to run OPC and not how to optimize the output.

### Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- Licenses for the following products: Calibre OPCpro. If you have a license for Calibre WORKbench, you can run OPC interactively as described in the section “[Alternate Procedure Using Calibre WORKbench](#).”
- The following inputs are required to perform this example:
  - OASIS layout, *Shared/Design/fullchip.oas*
  - Resist model, *Module5/Modeling/Models/vt5\_output.mod*
  - Optical model, *Module5/Modeling/Models/opticalD0*

### Batch Procedure

1. In a UNIX or Linux terminal window, change directory to *Calbr\_qs\_ekit/Module5*:
 

```
cd <path>/Calbr_qs_ekit/Module5
```

2. In a text editor, create an SVRF file called *opcpro.rul* and add the following lines:

```
LAYOUT PATH "../Shared/Design/fullchip.oas"
LAYOUT SYSTEM OASIS
LAYOUT PRIMARY "*"

PRECISION 2000
RESOLUTION 1

FLAG SKEW NO
FLAG ACUTE NO
FLAG OFFGRID NO
LAYOUT ERROR ON INPUT YES

DRC MAXIMUM RESULTS all
DRC RESULTS DATABASE "./opcpro_error.rdb" ASCII
DRC SUMMARY REPORT "./opcpro_summary.rep"

LAYOUT WINDOW 129 256 133 259
LAYOUT WINDOW CLIP yes
```

These SVRF statements specify the input and output, and restrict the run to a specific section of the layout. In a real run you would perform OPC on complete layers which can take significantly longer.

---

**Tip**



An annotated version of this file is also available in the *Module5* directory as *opcpro\_example.rul*.

---

3. Add the following OPC-specific lines.

```
LAYER M3 15
m3_OPC {
    LITHO OPC FILE opcpro.setup M3
}
DRC CHECK MAP m3_OPC m3_opc.gds
```

These lines identify the layer to correct, run OPCpro as part of a rule check, and write the resulting layer to a GDS file, *m3\_opc.gds*. The OPCpro settings are in a separate file, *opcpro.setup*, so that the same settings can be used to run Calibre OPCpro interactively within the Calibre WORKbench viewer.

4. Save and close the rule file.

5. Create a litho setup file called *opcpro.setup* and add the following lines:

```
# ----- simulation models -----
modelpath      Modeling/Models
opticalmodel    opticalD0
resistpolyfile  vt5_output.mod

# ----- OPC algorithm -----
iterations 4
tilemicrons 160.000
stepsize 0.001
gridsize 0.0005
siteinfo AERIAL
cornerSiteStyle SITES_ON_ARC

# ----- fragmentation -----
minfeature      0.045
minedgelenlength 0.015
maxedgelenlength 150.00
cornedge 0.015 0.015
concavecorn 0.015 0.015
minjog 0.001
lineEndLength 0.045


# ----- Layer info -----
background clear

layer 15 M3 0 0 opc dark
```

These are the contents of a basic litho setup file for Calibre OPCpro. Litho setup files are layer specific; you need separate setup files for each layer on which you run OPC. The statements are described in detail in the *Calibre OPCpro User's and Reference Manual*.

Unlike SVRF statements, the lines in the litho setup file are case-sensitive.

#### **Note**

 For convenience, a copy of the *opcpro.setup* file above has been provided for you in the example directory.

6. Save and close the setup file.
7. Invoke Calibre by entering the following command at a shell prompt:

```
$CALIBRE_HOME/bin/calibre -drc opcpro.rul
```

The settings in *opcpro.rul* instruct Calibre to perform OPC on M3 using the settings in *opcpro.setup*. These settings use the optical and resist models to simulate the photolithographic effects and attempt to improve the predicted image by fragmenting the layout polygons and moving edges to compensate.

Calibre generates a transcript while it is running, ending with lines similar to the following:

```
Cumulative ONE-LAYER BOOLEAN Time: CPU = 0   REAL = 0
Cumulative LITHO OPC Time: CPU = 0   REAL = 0
Cumulative RDB Time: CPU = 0   REAL = 0

--- CALIBRE::DRC-F EXECUTIVE MODULE COMPLETED.   CPU TIME = 1
      REAL TIME = 1
--- TOTAL RULECHECKS EXECUTED = 1
--- TOTAL RESULTS GENERATED = 13
--- DRC RESULTS DATABASE FILE = ./opcpro_error.rdb (ASCII)

--- CALIBRE::DRC-F COMPLETED - Tue May 15 09:49:49 20<xx>
--- TOTAL CPU TIME = 1   REAL TIME = 1
--- SUMMARY REPORT FILE = ./opcpro_summary.rep
```

---

### Note



If you get an invocation error, verify that *opcpro.rul* is in the current working directory, or its relative path is given in the invocation.

If you get an input error, verify that layer and file names are correct, including case. Use only alphanumeric characters in layer and file names.

---

8. View the OPC output layer (*m3\_opc.gds*) in any GDS viewer.

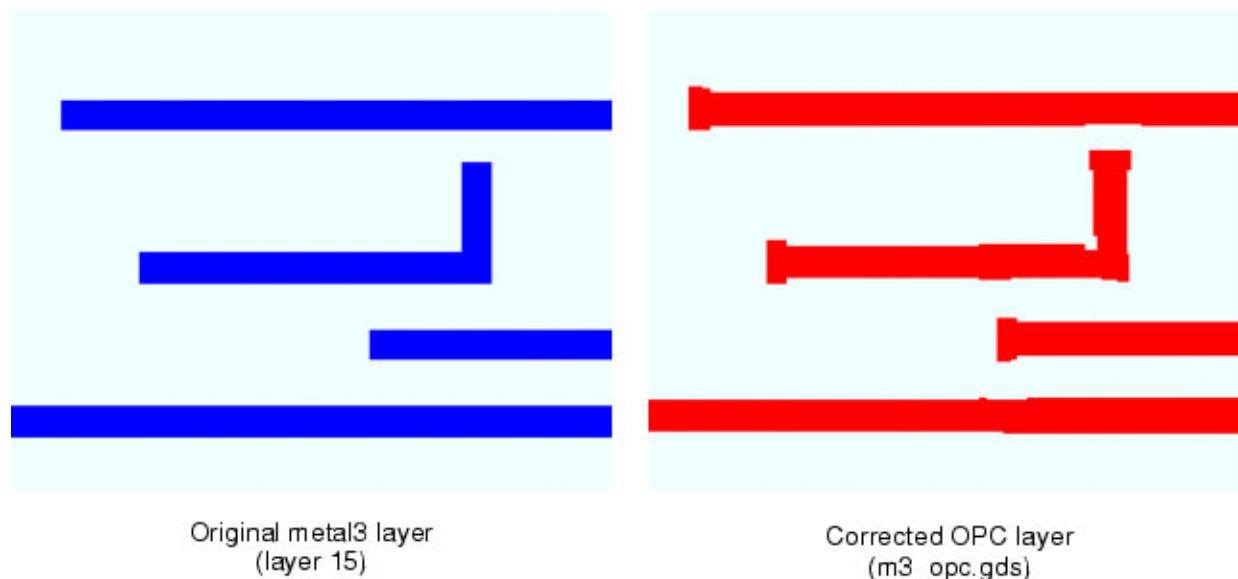
## Results

Upon completion, the following new files are in your directory:

- *m3\_opc.gds*
- *opcpro\_error.rdb*
- *opcpro\_summary.rep*

The OPC results are written as a single layer in the file *m3\_opc.gds*. You can open the file in any layout editor to see the edge-adjusted polygons, the red shapes in [Figure 5-3](#).

Figure 5-3. OPCpro Input and Output



The OPC and fragmentation algorithm used in this example is very basic, meant only to give a sense of what OPC can do. A more robust typical algorithm would include additional settings to create better OPC output.

After performing OPC, you detect potential errors in the output using Calibre OPCverify as described in the section “[Performing OPC Verification Using Calibre OPCverify](#).”

### Alternate Procedure Using Calibre WORKbench

This procedure requires you to have a license for Calibre WORKbench. You may find this method useful when developing OPC algorithms because it allows you to focus on typical layout geometries for testing.

1. Create a litho setup file using the lines from Step 5 of “[Batch Procedure](#)”.
2. Start Calibre WORKbench with the layout loaded:

```
$CALIBRE_HOME/bin/calibrewb ../Shared/Design/fullchip.oas
```

Calibre WORKbench uses the terminal window for input and output; do not run as a background process or by redirecting the output.

3. In the Layers list, right-click on a metal layer choose **Show Selected Only**. (The examples in this procedure use metal3 on layer 15 in *fullchip.oas*.)
4. Flatten the layer view. Click the [0 0] in “Depth: [0 0]” in the status bar at the bottom of the window. In the View Depth dialog box, enter 100 in the End depth text box and click **OK**.

5. Zoom in until you can see polygons clearly. (To go to the area used in the batch procedure and the results shown in [Figure 5-3](#), select **View > Go To** and enter the following coordinates:

**131 257.6 4.2**

If you see a blank area, check that the GoTo Location units are um and not dbu.

6. In the Calibre WORKbench toolbar, click the **Litho** button.

The RET Flow Tool: Tools (Simulation) window appears.

7. In the RET Flow Tool, select **File > Add Existing Setup**.

8. In the file browser, verify “File type” is set to All Files (\*.\*)).

9. Select the Calibre OPCpro setup file named *opcpro.setup* and click **Open**.

The RET Flow Tool window displays an OPC session. If you selected layer 15 in *fullchip.oas*, the **Setup Layers Mapping** tab in the lower left shows layer M3 mapped to layer 15 in the layout.

10. In the RET Flow Tool window, click the **OPC** button. If an OPC Run Options dialog box appears, click **OK** to use the default options to run the OPC tool.

In Calibre WORKbench, the visible region is overlaid with an OPC layer. If you zoom out, you can see that only the region that was displayed in the viewing area was corrected.

11. In the RET Flow Tool window, click the **PI** button to simulate what the printed image may look in Calibre WORKbench. (If the run options dialog box appears, click **OK**.)

The accuracy of the results, of course, are strongly influenced by the quality of the resist and optical models.

12. In Calibre WORKbench, select **File > Exit**.

13. If prompted to save changes, click **Discard** and then click **OK** to close the Calibre WORKbench window.

## Performing Sparse OPC Verification Using Calibre OPCverify

The OPC verification process for sparse OPC is identical to the OPC verification process for dense OPC. For more information, refer to the section “[Performing OPC Verification Using Calibre OPCverify](#).”

# Performing Dense OPC

The dense OPC process consists of three key processes: CM1 and VEB resist modeling flow, performing OPC and simulation using Calibre nmOPC, and performing OPC verification using Calibre OPCverify.


<b>CM1 and VEB Resist Modeling Flow .....</b>	<b>87</b>
<b>Performing OPC and Simulation Using Calibre nmOPC .....</b>	<b>89</b>
<b>Performing OPC Verification Using Calibre OPCverify.....</b>	<b>95</b>

## CM1 and VEB Resist Modeling Flow

The CM1 resist modeling flow uses the generic optical model and creates a dense modeling-specific CM1 resist model, plus an optional VEB etch model if etch data is available separately from resist data. These two models (optical and resist) are required for Calibre nmOPC and Calibre OPCverify operations, because the models are used in image simulations.

Although you can create and use default models using the Optical Model and CM1 Center tools, using optical and CM1 resist models calibrated to the data measurements provides a far more accurate representation of the FAB. This process is also known as “fitting” the measurement data to the test chip.

### Tip

 Because modeling is an iterative and hardware-intensive process, creating a model is not described in this document. This section only briefly describes the modeling process. Pre-generated models are provided for the OPC and OPCverify sections in the *Modeling* directory.

Calibre WORKbench is the tool used for model creation. You invoke Calibre WORKbench in dense mode with the `calibrewb` command.

The CM1 and VEB resist modeling flow consist of the following phases:

1. Collect sample data in a gauge file as described in “[Measurement of Test Patterns for Gauge Objects](#)”.
2. Create an acceptable optical model by iteratively performing “[Creation and Optimization of Optical Models for CM1 Modeling](#)”.
3. Use the optimized optical model for [CM1 Resist Model Calibration](#). This is also an iterative process until the most acceptable resist model is found.
4. If separate etch data is available, use the optimized optical and CM1 resist models for [VEB Etch Model Calibration](#).

## Measurement of Test Patterns for Gauge Objects

CM1 modeling uses the Test Pattern tool (**Litho > Test Patterns** menu item) to create a chip with known geometries. Sending this tool through your lithography process allows taking critical dimension (CD) measurements around special gauge lines placed on those test structures. This is the measurement data Calibre WORKbench uses to calibrate models. All the models use this data, which must be entered in a gauge data file.

Test Pattern and Gauge Data creation tasks and commands are described in the following chapters in the *Calibre WORKbench User's and Reference Manual*:

- Chapter 4, “Test Pattern Creation”
- Chapter 5, “Test Pattern Reference”
- Chapter 13, “CM1 and VEB Model Development Tasks” contains information on how to convert a sample spreadsheet file into a gauge data object.
- Chapter 15, “CM1 and VEB Model Reference” contains information on the gauge data object format.

## Creation and Optimization of Optical Models for CM1 Modeling

Optical models are usually created in the Calibre WORKbench Optical Model tool (**Optics > Optical Model** menu item). They simulate the illumination source, ambient medium, and the film stack.

Optical models are optimized using the characteristics you select as the most important for matching the test pattern data on an aerial image simulation. Optical models are independently optimized separately from resist models. The same optical model can be used with both CM1 models and VT5 models.

Optical model tasks and commands are described in the following chapters in the *Calibre WORKbench User's and Reference Manual*:

- Chapter 6, “Optical Model Creation”
- Chapter 8, “Optical Model Reference”

## CM1 Resist Model Calibration

CM1 resist models are calibrated in CM1 Center (**Litho > CM1 Center** menu item) using the simulated properties of the optical model combined with the measurements taken on a test pattern chip to extrapolate resist etching effects.

CM1 models are calibrated separately from optical models after the optical model calibration is complete. A CM1 model can account for both resist and etch measurements, or a separate VEB etch model can be created for etch if etch-specific data is available.



CM1 resist model tasks and commands are described in the following chapters in the *Calibre WORKbench User's and Reference Manual*:

- Chapter 13, “CM1 and VEB Model Development Tasks”
- Chapter 15, “CM1 and VEB Model Reference”

## VEB Etch Model Calibration

VEB etch models can be created and calibrated in CM1 Center when etch data is available separately from the resist measurement data. VEB etch models are optional.

VEB resist model tasks and commands are described in the following chapters in the *Calibre WORKbench User's and Reference Manual*:

- Chapter 13, “CM1 and VEB Model Development Tasks”
- Chapter 15, “CM1 and VEB Model Reference”

## modelflow\_v2 as a Batch Modeling Command

While Calibre WORKbench provides the CM1 Center tool to give you a GUI-driven way to optimize and calibrate optical and resist models, the CM1 Center GUI uses your choices to generate input for an underlying batch command (`modelflow_v2`). Some users may want to skip the GUI altogether and just work with `modelflow_v2` directly. CM1 Center can also export a script file for use with `modelflow_v2`.

The `modelflow_v2` syntax is described in Chapter 14, “`modelflow_v2` Reference” of the *Calibre WORKbench User's and Reference Manual*.

# Performing OPC and Simulation Using Calibre nmOPC

Calibre nmOPC is part of the dense OPC flow, targeted for 65 nm nodes and below. In this example, you run Calibre nmOPC to correct photolithographic effects on a layout. This produces an output OPC layer. Typically, you would analyze and fine-tune your OPC output before it is used to create reticle masks (the MDP step). This module, however, covers only how to run OPC and not how to optimize the output.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”

- Licenses for the following products: Calibre nmOPC. If you have a license for Calibre WORKbench, you can run OPC interactively as described in the section [“Optional Dense OPC Procedure Using Calibre WORKbench.”](#)
- The following inputs are required to run this example:
  - OASIS layout, *Shared/Design/fullchip.gds*
  - The litho model directory, *Models/dense\_lm*, which contains the following files:
    - *Lithomodel*, a file that identifies the models and defines the mask and background characteristics
    - CM1 model, *resist.mod*
    - Optical model, *opticalD0\_cm1\_calibrated*

## Batch Procedure

In this procedure, you use a batch process to perform dense OPC. If you have a license for Calibre WORKbench, you can optionally skip this procedure and perform the procedure [“Optional Dense OPC Procedure Using Calibre WORKbench.”](#)

1. In a UNIX or Linux terminal window, change directory to *Calbr\_qs\_ekit/Module5*:

```
cd <path>/Calbr_qs_ekit/Module5
```

2. In a text editor, create an SVRF file called *nmopc.svrf*.

---

### Tip



An annotated version of this file showing two styles of syntax is available in the *Module5* directory as *nmopc\_example.svrf*.

---

3. In the *nmopc.svrf* file, add the following SVRF commands:

```
LAYOUT PATH "../Shared/Design/fullchip.oas"
LAYOUT SYSTEM OASIS
LAYOUT PRIMARY "*"
PRECISION 2000
RESOLUTION 1

FLAG SKEW NO
FLAG ACUTE NO
FLAG OFFGRID NO
LAYOUT ERROR ON INPUT YES

DRC MAXIMUM RESULTS ALL
DRC RESULTS DATABASE "./nmopc_error.rdb" ASCII
DRC SUMMARY REPORT "./nmopc_summary.rep"

LAYOUT WINDOW 129 256 133 259
LAYOUT WINDOW CLIP yes
```

These SVRF statements specify the input and output, and restrict the run to a specific section of the layout. (In a real run, you would perform OPC on complete layers; this can take significantly longer, however.)

4. In the *nmopc.svrf* file, add the following OPC-specific lines:

```
LAYER M1 11
opc_m1 = LITHO DENSEOPC FILE run_M1 M1 MAP M1_opc
opc_m1 {COPY opc_m1} DRC CHECK MAP opc_m1 oasis 111 m1_out.oas
```

This specifies that layer 11 of the *fullchip.oas* design will be assigned to “M1” and that the OPC output layer (layer 111) will be placed in an OASIS-format file called “*m1\_out.oas*.”

5. In the *nmopc.svrf* file, add the following:

```
LITHO FILE run_M1 [
  modelpath Modeling/Models
  layer M1 hidden atten 0.06

  denseopc_options dopc {
    version 1
    layer M1 opc
    image dense_lm
  }
  setlayer M1_opc = denseopc M1 MAP M1 OPTIONS dopc
]
```


This portion of the SVRF file, also known as the Calibre nmOPC “setup file,” contains the basic OPC recipes to correct the 45 nm design to account for the errors that may be introduced by the photolithographic process. Each Calibre nmOPC block contains all of the information necessary for running corrections, including:

- Model directory location. Litho models contain the optical and resist model and mask definition. For non-litho model, the setup file loads each one individually.
- Input layer definitions. The definitions map the layers from SVRF to the setup file. The mapping is strictly by order, and at least one is required.
- A setlayer command to generate dense OPC output.
- OPC options. The layer names must match the names in the setlayer command.

Complete information on the Calibre nmOPC setup file is documented in the *Calibre nmOPC User’s and Reference Manual*.

This example uses a minimal Calibre nmOPC setup file. Unlike SVRF statements, the lines in the setup file are case sensitive.

### Note

 The example file provided (*nmopc\_example.svrf*) also contains legacy code for a non-litho model version of the setup file, which individually loads models and explicitly declares layer and background information. It is retained for comparison.

---

6. Save and close the *nmopc.svrf* file.
7. Invoke Calibre nmDRC by entering the following command at a shell prompt:

```
$CALIBRE_HOME/bin/calibre -drc nmopc.svrf
```

Calibre uses the optical and resist models to simulate the photolithographic effects and then attempts to correct the original layout by fragmenting the layout polygons and moving the edges to compensate.

Calibre generates a transcript while it is running, ending with lines similar to the following:

```
Cumulative ONE-LAYER BOOLEAN Time: CPU = 0   REAL = 0
Cumulative LITHO DENSEOPC Time: CPU = 3   REAL = 4
Cumulative RDB Time: CPU = 0   REAL = 0

--- CALIBRE::DRC-F EXECUTIVE MODULE COMPLETED.   CPU TIME = 4
      REAL TIME = 5
--- TOTAL RULECHECKS EXECUTED = 1
--- TOTAL RESULTS GENERATED = 43
--- DRC RESULTS DATABASE FILE = ./nmopc_error.rdb (ASCII)

--- CALIBRE::DRC-F COMPLETED - Thu Apr 19 14:41:06 2019
--- TOTAL CPU TIME = 4   REAL TIME = 13
--- SUMMARY REPORT FILE = ./nmopc_summary.rep
```

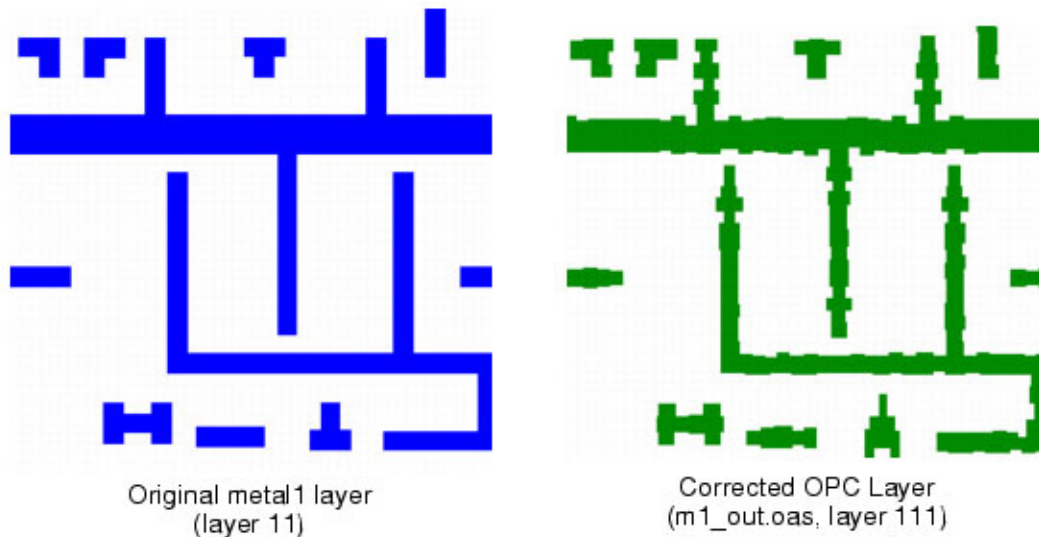
## Results

Upon completion, the following new files are in your directory:

- *m1\_out.oas*
- *nmopc\_error.rdb*
- *nmopc\_summary.rep*

The OPC results are in *m1\_out.oas*. You can open it in any layout editor to see the edge-adjusted polygons, as shown in [Figure 5-4](#).

Figure 5-4. Original Design Versus OPC Output



The OPC and fragmentation algorithm used in this example is very basic, meant only to give a general sense of what OPC can do. A more robust Calibre nmOPC recipe includes additional settings to create better output, as described in the *Calibre nmOPC User's and Reference Manual*.

After performing OPC, you detect potential errors (such as bridging and pinching) in the output using Calibre OPCverify.

### Optional Dense OPC Procedure Using Calibre WORKbench

This procedure is an alternative to the previous [Batch Procedure](#). Some people find this method useful when developing OPC algorithms because they can focus on typical layout geometries for testing. This procedure requires you to have a license for Calibre WORKbench.

1. Start Calibre WORKbench with the layout loaded:

```
$CALIBRE_HOME/bin/calibrewb ../Shared/Design/fullchip.oas
```

Calibre WORKbench uses the terminal window for input and output; do not run as a background process or by redirecting the output.

2. In the layer list, right-click on a metal layer and choose **Show Selected Only**.  
In *fullchip.oas*, M1 is metal1 on layer 11.
3. Flatten the layer view by clicking the [0 0] in “Depth: [0 0]” in the status bar at the bottom of the window.
4. In the View Depth dialog box, enter 10 in the End depth text box and click **OK**.
5. Zoom in until you can see polygons clearly.

To go to the area used in the batch procedure and the results shown in [Figure 5-4](#), select **View > Go To** and enter the following coordinates:

**131 257.6 4.2**

6. Click the **Extract** button in the Calibre WORKbench toolbar. A file browser dialog box appears.
7. Select the *nmopc\_example.svrf* file located in your *Module5* directory.

This file contains SVRF code with a Calibre nmOPC setup file block. This setup file block contains code nearly identical to the one created in “[Batch Procedure](#)” on page 90.

The RET Flow Tool window appears with the setup file, called *opc\_m1.lsf*, displayed for a DENSEOPC (Calibre nmOPC) session.

8. In the Setup Layers Mapping area in the lower left of the RET Flow Tool, verify the **Layers Mapping** tab is selected.
9. Click the WB Layer # - Name dropdown list and choose layer 11 - metal1 from the list.
10. In the Calibre RET Flow Tool window, click the **OPC** button.

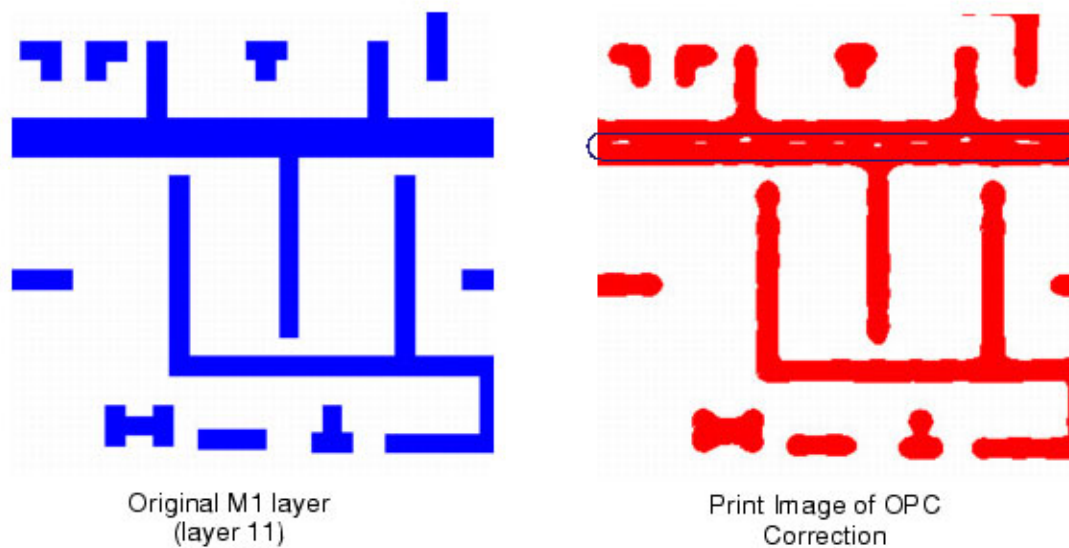
In Calibre WORKbench, the visible region is overlaid with an OPC layer (M1\_opc). If you zoom out, you can see that only the region that was displayed in the viewing area was corrected.

11. You can simulate what the printed image may look like using the **PI** button in the Calibre RET Flow Tool window. The accuracy of the results, of course, are strongly influenced by the quality of the resist and optical models.
12. In the Calibre WORKbench window, select **File > Exit**.
13. If prompted to save changes, click **Discard** and then click **OK** to close the Calibre WORKbench window.

## Results

[Figure 5-5](#) shows a portion of the printed image from the resulting OPC run using the minimal setup file.

Figure 5-5. Original Design Versus Printed Image



Notice that in the example printed image, a number of holes appear in a shape from the original layout. Issues such as holes, as well as pinching and bridging, can be detected by Calibre OPCverify to allow you to make further corrections to either the design or to the OPC recipe.

## Performing OPC Verification Using Calibre OPCverify

The OPC verification process for dense OPC is identical to the OPC verification process for sparse OPC.

Calibre OPCverify is a Tcl-based batch programming language that works with both VT5 and CM1 models. You can write short applications with the custom language (described in the *Calibre OPCverify User's and Reference Manual*) to verify your most important OPC layer conditions.

In the following steps, you use Calibre OPCverify to find potential errors in the test pattern design. Calibre OPCverify simulates fabrication conditions using optical and resist models you supply, and checks for possible problems based on operational tests you program. The OPCverify rule files, optical, and resist models are provided for you. The example OPCverify file detects possible pinching and bridging conditions on one of the layers.

### Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”

- Licenses for the following products: Calibre OPCverify.
- The following inputs are required to perform this example:
  - Optical model for sparse OPC, *Module5/Modeling/Models/opticalD0\_calibrated*
  - VT5 resist model file for sparse OPC, *Module5/Modeling/Models/vt5\_output.mod*
  - Litho model for dense OPC, *Module5/Modeling/Models/dense\_lm*, which contains the following files:
    - *Lithomodel*, an information file containing mask and background information, and a list of model files in the litho model
    - Optical model, *opticalD0\_cm1\_calibrated*
    - CM1 resist model file, *resist.mod*, used in the CM1 version of the Calibre OPCverify rule file
  - SVRF rule file for VT5 simulation example, *Module5/Opcverify/opcv\_vt5.svrf*
  - SVRF rule file for CM1 simulation example, *Module5/Opcverify/opcv\_cm1.svrf*
  - OASIS layout, based on the eKit chip pre-OPCpro run, *Module5/Opcverify/top\_ic\_pre\_opc.oas*
  - OASIS layout, based on the eKit chip post-OPCpro run, *Module5/Opcverify/top\_ic\_opc.oas*.

## Procedure

1. Open a new terminal window and change directory to *Calbr\_qs\_ekit/Module5/Opcverify*:
2. Enter one of the following commands at a shell prompt, corresponding to either the VT5 or CM1 model:

```
$CALIBRE_HOME/bin/calibre -drc -hier -turbo opcv_vt5.svrf
$CALIBRE_HOME/bin/calibre -drc -hier -turbo opcv_cm1.svrf
```

This invokes Calibre nmDRC-H, which runs the SVRF rule file. The SVRF rule file calls the Calibre OPCverify command file that is inlined (contained) inside the rule file. Calibre OPCverify only returns error results when all of the following conditions are true:


- An operation is coded to check for a particular error condition.
- The error condition is met.
- The error lies in a region and layer that is output to a design file.



3. Change directory to *output* and open the output results, *opcv\_out\_vt5.oas* or *opcv\_out\_cm1.oas*, in a layout viewer such as Calibre DESIGNrev.
4. Look for possible problems in the short, open, pinch and bridge layers.
  - *short* — Indicates where the contour has touched another contour. Also known as “hard bridging.”
  - *open* — Indicates where the contour has separated. Also known as “hard pinching.”
  - *pinch* — Indicates where the contour is narrow enough to be at risk of hard pinching.
  - *bridge* — Indicates where two contour edges are close enough together to be at risk of hard bridging.

---

**Tip**

 Depending on the quality of your models, you may not see all of the error layers. A set of good models should have no opens or shorts, and few pinch or bridge errors.

This module features a VT5 result that has only bridge errors, and a CM1 result that has only pinch and bridge errors.

---

5. Close the output results.
6. Edit the appropriate SVRF rule file, *opcv\_vt5.svrf* or *opcv\_cm1.svrf*, and make the following changes:
  - Comment out the input file (*top\_ic\_pre\_opc.oas*) from the pre-OPC input file.
  - Uncomment the input file (*top\_ic\_opc.oas*) from the post-OPC input file.
7. Save and close the rule file.
8. Re-run Calibre nmDRC.
9. Examine the improved results (because post-OPC image contours ideally contain less errors) by reloading the resulting design output.
10. Exit the layout viewer tool.

## Results

When design errors are detected by a coded operation, they are output on the specified layer in the output results file.



# Chapter 6

## Preparing Mask Data

---

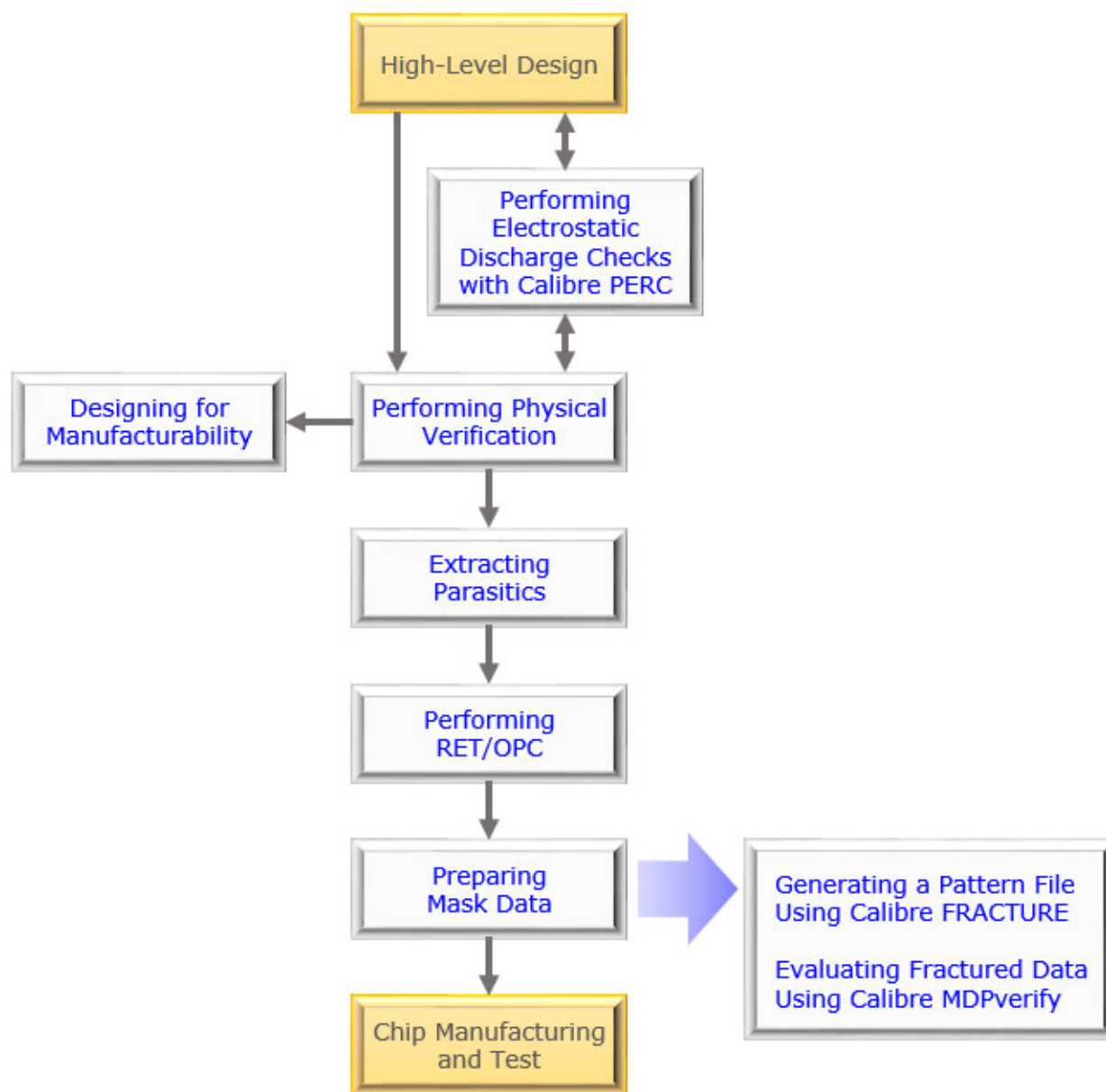
Calibre Mask Data Preparation (MDP) is a series of tools used to generate, correct, verify, and analyze photomask layouts. The tools are used to fracture complex polygons into simpler shapes that can be written to a proprietary format, in addition to evaluating the compliance of a data set to the mask making limits prior to manufacturing.

<b>Mask Data Preparation Design Flow</b> . . . . .	<b>99</b>
<b>Calibre Mask Data Preparation (MDP) Tool Reference</b> . . . . .	<b>100</b>
<b>Generating a Pattern File Using Calibre FRACTURE</b> . . . . .	<b>105</b>
<b>Evaluating Fractured Data Using Calibre MDPverify</b> . . . . .	<b>107</b>

### Mask Data Preparation Design Flow

Figure 6-1 shows some of the MDP tasks as described in the procedures in this module.

Figure 6-1. Mask Data Preparation Design Flow



## Calibre Mask Data Preparation (MDP) Tool Reference

Supporting documentation is available for each licensed Calibre product and, in some cases, training is also available. There may be other products that are closely related to the use of that tool.

Table 6-1 provides a comprehensive list and short description of the licensed Calibre products associated with the mask data preparation process, in addition to information on related documentation, training, and products.

**Table 6-1. Calibre Mask Data Preparation Tool Reference**

Tool	Description	Related Documentation	Related Training	Related Products
Calibre® FRACTURE	A tool used to convert layer data into MEBES, JEOL, Hitachi, Micronic, Toshiba/Nuflare, and OASIS.MASK formatted data.	<i>Calibre Mask Data Preparation User's and Reference Manual</i>		Calibre® MDPview™
Calibre® Job Deck Editor	A feature of Calibre WORKbench and MDPview used to generate an optimized chip placement on the wafer.	<i>Calibre Job Deck Editor User's Manual</i>		Calibre MDPview
Calibre® MAPI (Metrology API)	A set of iTcl (incr Tcl) classes for creating custom extensions for supporting of manufacturing machines (metrology machines) with Calibre WORKbench or Calibre MDPview.	<i>Calibre Metrology API (MAPI) User's and Reference Manual</i>		Calibre MDPview
Calibre® MASKOPT™	A feature that modifies the input geometry to improve fracture quality.	<i>Calibre Mask Data Preparation User's and Reference Manual</i>		Calibre MDPview

**Table 6-1. Calibre Mask Data Preparation Tool Reference (cont.)**

<b>Tool</b>	<b>Description</b>	<b>Related Documentation</b>	<b>Related Training</b>	<b>Related Products</b>
Calibre® MDP Embedded SVRF™	Allows a block of SVRF commands to be inserted in a FRACTURE invocation.	<i>Calibre Mask Data Preparation User's and Reference Manual</i>		Calibre MDPview
Calibre® MDPAutoClassify™	A tool used for automatic classification of defects observed on a blank substrate before any patterning is performed on the substrate.	<i>Calibre MDPAutoClassify User's Manual</i>		Calibre MDPDefectReview
Calibre® MDPDefectAvoidance™	A tool invoked from Calibre MDPview that is used to find shifts in a layout to prevent extreme ultraviolet lithography (EUVL) blank mask defects from appearing on patterns.	<i>Calibre MDPDefectAvoidance User's Manual</i>		Calibre MDPview
Calibre® MDPDefectReview™	Provides efficient analysis, classification, and trend analysis of defects identified by mask inspection systems.	<i>Calibre MDPDefectReview User's Manual</i>		Calibre MDPAutoClassify

**Table 6-1. Calibre Mask Data Preparation Tool Reference (cont.)**

Tool	Description	Related Documentation	Related Training	Related Products
Calibre® MDPmerge™	A tool used to read a job deck and merge individual chips and chip placements into a single chip and placement based on certain criteria.	<i>Calibre Mask Data Preparation User's and Reference Manual</i>		Calibre MDPview
Calibre® MDPPatternClassify™	Enables automatic classification of defects observed on a patterned mask.	<i>Calibre MDPPatternClassify User's Manual</i>		Calibre MDPAutoClassify
Calibre® MDPstat™	A tool used to gather statistics on Hitachi, JEOL, Micronic, VSB11/ VSB12, and OASIS.VSB formatted data, allowing you to assess the quality of the fracture output.	<i>Calibre Mask Data Preparation User's and Reference Manual</i>		Calibre MDPview
Calibre® MDPverify™	A tool used to evaluate fractured MDP data by comparing either: - Fractured data to the original layout data - Any Calibre database layer or fractured data in one pattern file to fractured data in a different pattern file.	<i>Calibre Mask Data Preparation User's and Reference Manual</i>		Calibre MDPview

**Table 6-1. Calibre Mask Data Preparation Tool Reference (cont.)**

<b>Tool</b>	<b>Description</b>	<b>Related Documentation</b>	<b>Related Training</b>	<b>Related Products</b>
Calibre® MDPview™	A GUI viewer that enables you to visually inspect the results of your MDP operations.	<i>Calibre MDPview User's and Reference Manual</i>		
Calibre® Metrology Interface (CMi)	A tool created as a design-based off-line marking system for SEM measurements.	<i>Calibre Metrology Interface (CMi) User's Manual</i>		Calibre® WORKbench™®
Calibre® MPCpro™	A series of SVRF commands that apply pattern-based mask process corrections to your design data prior to fracturing.	<i>Calibre Mask Data Preparation User's and Reference Manual</i>		Calibre MDPview
Calibre® MPCverify™	A grid-based mask process simulator and MPC results verification tool that is designed to predict a mask manufacturing process.	<i>Calibre MPCverify User's and Reference Manual</i>		Calibre nmMPC
Calibre® nmMPC™	Provides a suite of functions for modeling, simulation, and correcting distortions of the mask manufacturing process.	<i>Calibre nmMPC User's and Reference Manual</i>		Calibre WORKbench ----- Calibre MPCverify



# Generating a Pattern File Using Calibre FRACTURE

A FRACTURE operation converts layer data into a formatted data for a photomask machine (such as MEBES, JEOL, Hitachi, Micronic, Toshiba/Nuflare, or VB:OASIS). This formatted data file is also known as a “pattern file.”

In this example, you will convert a provided design layout into a MEBES-format pattern file.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- Licenses for the following products: Calibre FRACTUREm.
- The following inputs are required to perform this example:
  - OASIS layout, *Shared/Design/fullchip.oas*

## Procedure

1. In a UNIX or Linux terminal window, change directory to *Calbr\_qs\_ekit/Module6*:

```
cd <path>/Calbr_qs_ekit/Module6
```

2. Create a text MDP FRACTURE rule deck, *mdp.svrf*, in an ASCII editor using the following SVRF statements:

```
LAYOUT PATH "../Shared/Design/fullchip.oas"
LAYOUT SYSTEM OASIS
LAYOUT PRIMARY "*"
LAYOUT MAGNIFY 1
PRECISION 2000.0
RESOLUTION 1
FLAG ACUTE NO
FLAG SKEW NO
FLAG OFFGRID NO
LAYOUT ERROR ON INPUT Yes
LAYOUT INPUT EXCEPTION SEVERITY PRECISION_RULE_FILE 1
DRC MAXIMUM RESULTS ALL
DRC RESULTS DATABASE "../drc_result.oas" OASIS
DRC SUMMARY REPORT "../drc_summary.rep"
Layer M1 11
```

### Note



This code example is also available in the *mdp\_example.svrf* file provided with your example kit.

This block instructs Calibre to convert the layer data using the layout file *fullchip.oas* as input and generate an OASIS results database, *drc\_result.oas*, and a summary report, *drc\_summary.rep*.

3. To the same rule deck, add the following FRACTURE block:

```
MEBES_Fracture {  
  Fracture  MEBES M1 INSIDE OF 24.7 310.8 67 350 FILE [  
    file_name "TEST1XXXX.PF"  
    mode 5  
    log_file_name "TEST1XXXX_PF.log"  
    address_size 0.02  
    compaction_stripes 32  
  ] }  
}
```

---

#### Note



This code example is also available in the *mdp\_example.svrf* file provided with your example kit.

---

This block instructs Calibre to convert a region of the layer data (the coordinates “24.7 310.8 67 350”) to MEBES format (as well as produce a log file). Specific FRACTURE MEBES command descriptions can be found in the *Calibre Mask Data Preparation User’s and Reference Manual*.

4. Save and close the *mdp.svrf* file.
5. Enter the following command at a shell prompt:

```
$CALIBRE_HOME/bin/calibre -drc -hier mdp.svrf
```

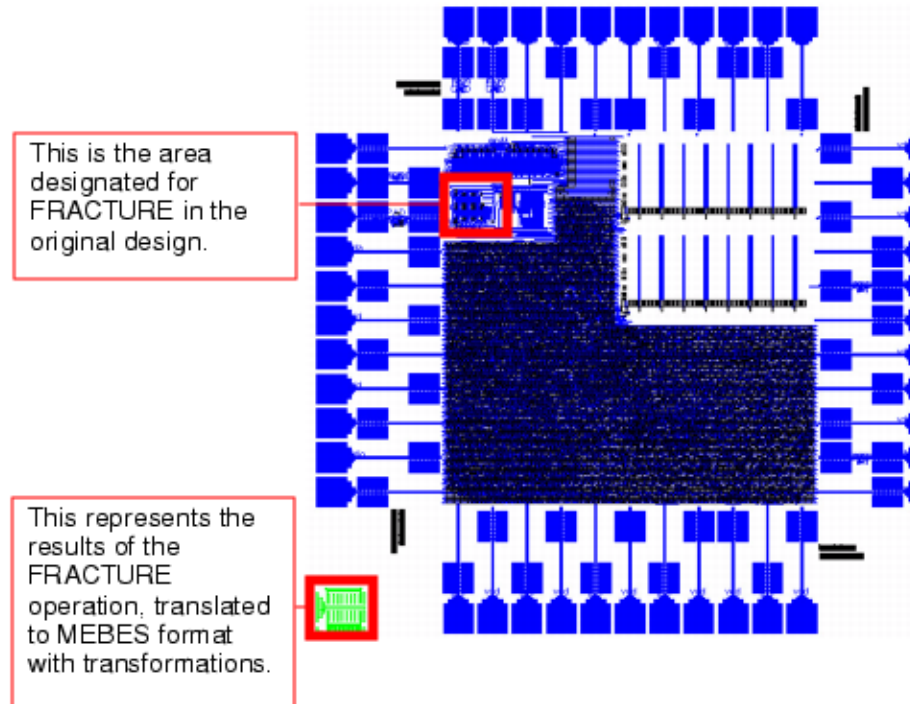
This executes Calibre FRACTURE and produces the following files:

- *TEST1XXXX.PF* — The MEBES-format pattern file.
- *TEST1XXXX\_PF.log* — The log file from the Calibre run.

## Results

You should now have a pattern file that simulates how a MEBES-format photomask machine will translate the original layer data. Viewing the pattern file and the original layout in a layout viewer (such as Calibre WORKbench or Calibre MDPview) shows that the original layer has been “transformed” to suit the photomask machine. [Figure 6-2](#) shows the results of the MEBES fracture operation.

**Figure 6-2. Fracture Results**



However, it is typical in the MDP process to also verify the results by comparing the pattern file to another database file (either the original database or layer, or another pattern file) as described in the section “[Evaluating Fractured Data Using Calibre MDPverify](#).”

## Evaluating Fractured Data Using Calibre MDPverify

Calibre MDPverify allows you to evaluate fractured MDP data by performing key comparisons. These are:

- Fractured data to the original layout data or any Calibre database layer.
- Fractured data in one pattern file to fractured data in a different pattern file.

In this example, you will use the rule file and pattern file generated in the section “[Generating a Pattern File Using Calibre FRACTURE](#)” for a comparison against the original Calibre layer database.

### Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”

- Licenses for the following products: Calibre FRACTUREm and Calibre MDPverify.
- The following inputs are required to perform this example:
  - OASIS layout, *Shared/Design/fullchip.oas*
  - The rule file, *mdp.svrf*, created in “[Generating a Pattern File Using Calibre FRACTURE](#).”
  - The MEBES-based pattern file, *TEST1XXXX.PF*, created in “[Generating a Pattern File Using Calibre FRACTURE](#).”

## Procedure

1. Open the *mdp.svrf* file created in “[Generating a Pattern File Using Calibre FRACTURE](#)” in an ASCII text editor.
2. Note the area (**INSIDE OF 24.7 310.8 67 350**) in the original FRACTURE block.  
This will need to be matched in the Calibre MDPverify command block.
3. Add the following Calibre MDPverify command block to the end of the *mdp.svrf* file.

```
MEBES2DB_verify {  
    MDPverify M1 INSIDE OF 24.7 310.8 67 350 FILE [  
        verify_type MEBES2DB  
        input_file TEST1XXXX.PF  
        reverse_tone NO  
        maximum_output_count 2000000  
    ]  
}  
DRC CHECK MAP MEBES2DB_verify OASIS 600
```

This code example is also available in the *mdp\_example.svrf* file provided with your example kit.

This Calibre MDPverify command block checks a specified area of the input pattern file (*TEST1XXXX.PF*) and compares it to the original database (MEBES2DB). The DRC Check Map statement then outputs the results plus the original data to the Results Database.

Specific MDPverify command descriptions can be found in the *Calibre Mask Data Preparation User's and Reference Manual*.

4. Save and close the *mdp.svrf* file.
5. Enter the following command at a shell prompt:

```
$CALIBRE_HOME/bin/calibre -drc -hier mdp.svrf
```

The Calibre FRACTURE software is executed in the Calibre nmDRC hierarchical engine.

This produces a MEBES-format pattern file (*TEST1XXXX.PF*) and compares the resulting file to the original layer for verification. The results of Calibre MDPverify are

all geometry that is in either the layout database or the fractured data but not in both; these results are written to the DRC results database file, *drc\_result.oas*.

## Results

You have now seen the basic MDP flow. At this point, you now have verified that your layout design will translate correctly to the specific photomask machine format. A number of more advanced MDP tools are also available to help produce, verify, and correct mask data, as described in the section “[Calibre Mask Data Preparation \(MDP\) Tool Reference](#).”



# Chapter 7

## Performing Electrostatic Discharge Checks with Calibre PERC

---

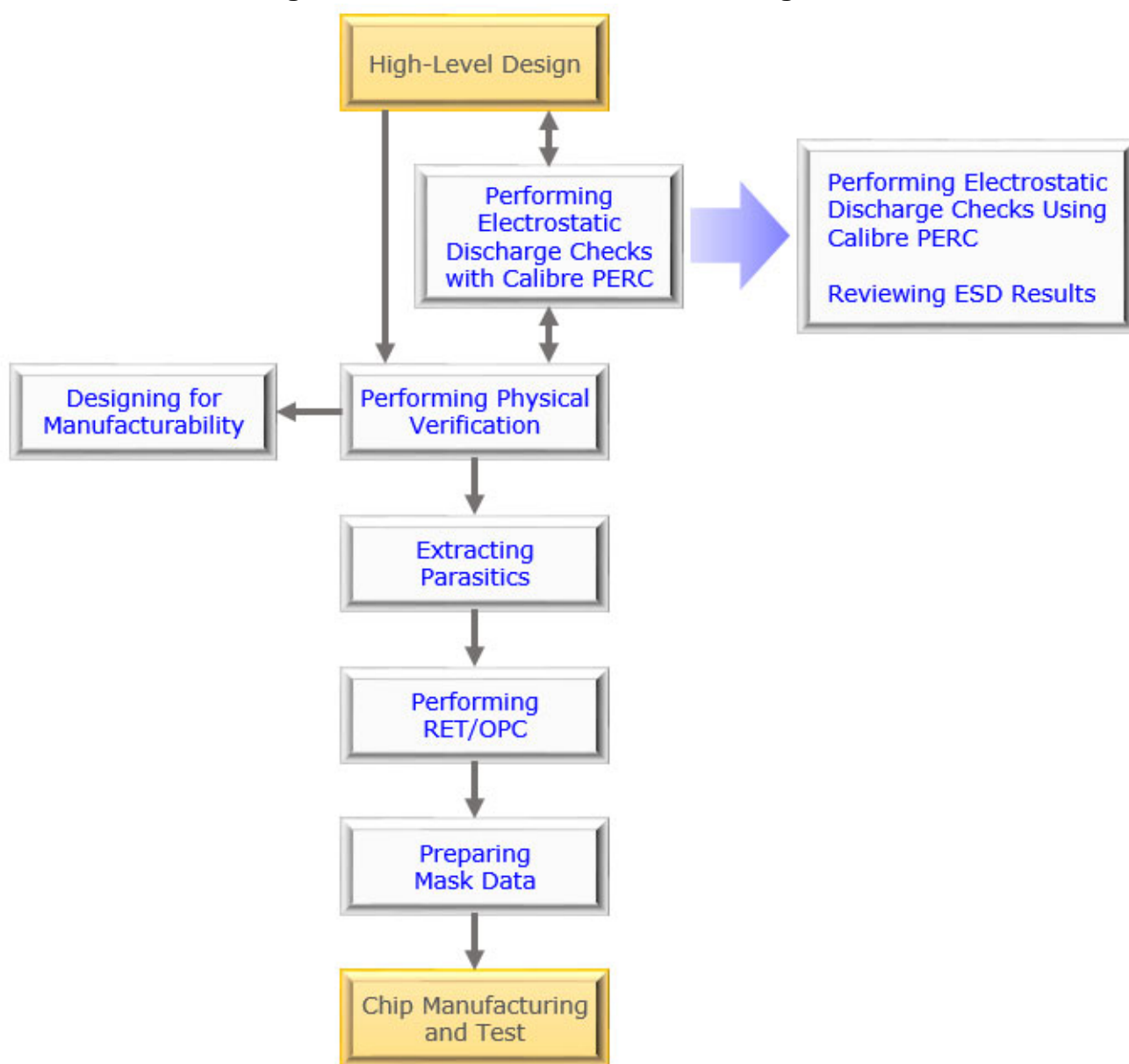
Electrostatic Discharge (ESD) checks are best performed on the source design, prior to the initiation of the layout phase. ESD checks can also be performed on the physical layout.

<b>ESD Checks in the Design Flow .....</b>	<b>111</b>
<b>Performing Electrostatic Discharge Checks Using Calibre PERC.....</b>	<b>113</b>
<b>Reviewing ESD Results .....</b>	<b>117</b>

### ESD Checks in the Design Flow

Figure 7-1 shows where ESD is typically done in the design flow. Additionally, this process may involve performing advanced electrical rule checks (ERCs).

**Figure 7-1. ESD Verification in the Design Flow**



Refer to [Table 2-1](#) for information on related documentation, training, and applications.

See the `$CALIBRE_HOME/shared/examples/calibre_perc` directory in your Calibre software tree for a library of working examples for ESD and advanced ERC checks.



# Performing Electrostatic Discharge Checks Using Calibre PERC

A source design has been completed and a SPICE netlist generated. In the following example, you run Calibre PERC to detect electrostatic discharge (ESD) errors in the source design.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in “[Installing the Quick Start Example Data](#)” and set up your environment according to the information in “[Global Prerequisites](#).”
- Licenses for the following products: Calibre PERC.
- The following inputs are required to perform this example:
  - SVRF rule file, *Module7/perc.rules*
  - SPICE netlist, *Module7/src.net*

## Procedure

1. In a UNIX or Linux terminal window, change directory to *Calbr\_qs\_ekit/Module7*:

```
cd <path>/Calbr_qs_ekit/Module7
```

2. Open the PERC rule file, *perc.rules*, in a text editor and notice the following SVRF rule file statements:

```
SOURCE PATH      src.net
SOURCE PRIMARY   TOPCELL
SOURCE SYSTEM    SPICE

PERC NETLIST SOURCE
PERC REPORT "perc.report"
MASK SVDB DIRECTORY "svdb" QUERY

PERC PROPERTY R r
PERC PROPERTY MN w l

PERC LOAD esd INIT init SELECT check_HBM_diodes
check_gate_protection check_driver_protection
check_RC_triggered_rail_clamp
```

All of these statements except Mask SVDB Directory and PERC Property are required to run Calibre PERC.

3. Briefly review the remainder of the rule file. Notice the rules are coded in Tcl procedures.
4. Close the *perc.rules* file.
5. In the *Module7* working directory, enter the following command in a shell prompt:

```
$CALIBRE_HOME/bin/calibre -perc -hier perc.rules | tee perc.log
```

The run transcript will appear in the shell window.

6. Open the *perc.log* file.

The header in the transcript includes the following information:

- The Calibre version used for the run:

```
// Calibre v20xx.x_xx.xx <time stamp>
```

- The operating system, hostname, OS version, and build:

```
// Mentor Graphics software executing under <architecture>
...
// Running on <OS> <hostname> <version>.
```

- The path of the executable, the command line options, and the PID are shown on these lines:

```
// Running <version>/pkgs/icv/pvt/calibre -perc -hier perc.rules
// Process ID: <pid>
```

- The start time of the run and the number of CPUs used for making runtime calculations (those for which licenses are required) are shown on these lines:

```
// Starting time: <timestamp>
//
// Running on 1 CPU (pending licensing)
```

- Calibre PERC automatically compiles the rule file at the beginning of the run. The transcript includes the pathname of the rule file, the contents of the rule file, and the amount of CPU and real time required for the compilation.

```
-----
-----
----- STANDARD VERIFICATION RULE FILE COMPILATION MODULE -----
-----
-----

--- RULE FILE = perc.rules
```

- The “CALIBRE::PERC - INITIALIZATION MODULE” section in the transcript shows reading of the Tcl procedures that comprise the ESD rule checks.
- The “CALIBRE::PERC - EXECUTIVE MODULE” section in the transcript shows the reading of the netlist and the execution of the rule checks.

```
Executing RuleCheck "check_HBM_diodes" ...
  Checking RULE: ESD double diodes required for primary
  protection of IO pads
  RuleCheck "check_HBM_diodes" executed. CPU TIME = 0  REAL TIME
  = 0  LVHEAP = 1
  /3/3  MALLOC = 7/7/7  ELAPSED TIME = 3
```

The Tcl code is read and interpreted at runtime, not at compile time. Errors in the Tcl code are returned to STDERR as they are encountered.

- The end of the transcript contains this summary information:

```
--- CALIBRE::PERC EXECUTIVE MODULE COMPLETED. CPU TIME = 0 REAL  
TIME = 1 LVHEAP = 3/5/5 MALLOC = 47/47/47 ELAPSED TIME = 2  
...  
--- CALIBRE::PERC CHECK(s) FAILED - Fri Aug 24 10:50:58 2018  
  
--- TOTAL CPU TIME = 0 REAL TIME = 1 LVHEAP = 3/5/5 MALLOC =  
47/47/47 ELAPSED TIME = 2  
--- PROCESSOR COUNT = 1  
--- PERC REPORT FILE = perc.report  
--- PERC SUMMARY REPORT FILE = NONE
```

7. Close the *perc.log* file.
8. Open the *perc.report* file in an ASCII editor and review the results.

The PERC Report contains these major sections:

- **OVERALL VERIFICATION RESULTS** — shows whether checks passed or failed.
- **CELL SUMMARY** — shows which cells were completed, the hierarchical and flat result counts, and the cell names.
- **RULECHECK SUMMARY** — shows which checks were run, their completion status, the hierarchical and flat result counts, and rule check comments, if any.

- CELL VERIFICATION RESULTS — shows cell-by-cell results:

```

CELL VERIFICATION RESULTS

#####
#                                     #
#          CHECK(S) FAILED          #
#                                     #
#####

Results:  Total RuleCheck result count = 1 (1) <-- errors
CELL NAME:      cellA (1 placement)      <-- in cell
-----

NUMBERS OF OBJECTS
-----

Count      Component Type
-----
Ports:      4
Nets:       8
Instances:   3      D (2 pins)
              2      R (2 pins)
              2      MP (4 pins)
              2      MN (4 pins)
Total Inst:  9
-----

*****
RULECHECK RESULTS
*****
RESULT#
*****

o PERC LOAD esd INIT init  <-- Associated PERC LOAD statement
o RuleCheck:  check_gate_protection (CDM protection for NMOS
gates connected to I/O pads)
-----

1      Net   9 [ Gate ] [ Pad Gate ] (1 placement, LIST# = L1)
      Missing down CDM diode

```

9. Read through the remainder of the report.
10. Close the PERC Report file.

## Results

You have now run Calibre PERC, which performed electrostatic discharge checking of the source netlist. The outputs from Calibre PERC include:

- Run transcript (*perc.log*)
- Summary report (*perc.report*)
- Results database (*svdb*)

If errors were found during the run, you would normally fix any errors and run Calibre PERC again. In the next procedure, you will use Calibre RVE to review the results.

## Reviewing ESD Results

You can review ESD check results in the Mask SVDB Directory using Calibre RVE.

Ordinarily, Calibre RVE is invoked either from a layout editor or from Calibre Interactive. If your tool environment is configured with Calibre RVE integrated into your tools, then call Calibre RVE from the appropriate menu items. In Calibre layout viewers like Calibre DESIGNrev, you call Calibre RVE from the Verification menu. In third-party layout editors, you frequently call Calibre RVE from the Calibre menu. In this procedure, no layout editor is assumed to be configured, and Calibre Interactive is not assumed to be in your environment.

## Prerequisites

- To perform this procedure, you must have installed the example data as described in [“Installing the Quick Start Example Data”](#) and set up your environment according to the information in [“Global Prerequisites.”](#)
- Licenses for the following products: Calibre RVE.
- A Mask SVDB Directory generated in the [“Performing Electrostatic Discharge Checks Using Calibre PERC”](#) procedure.

## Procedure

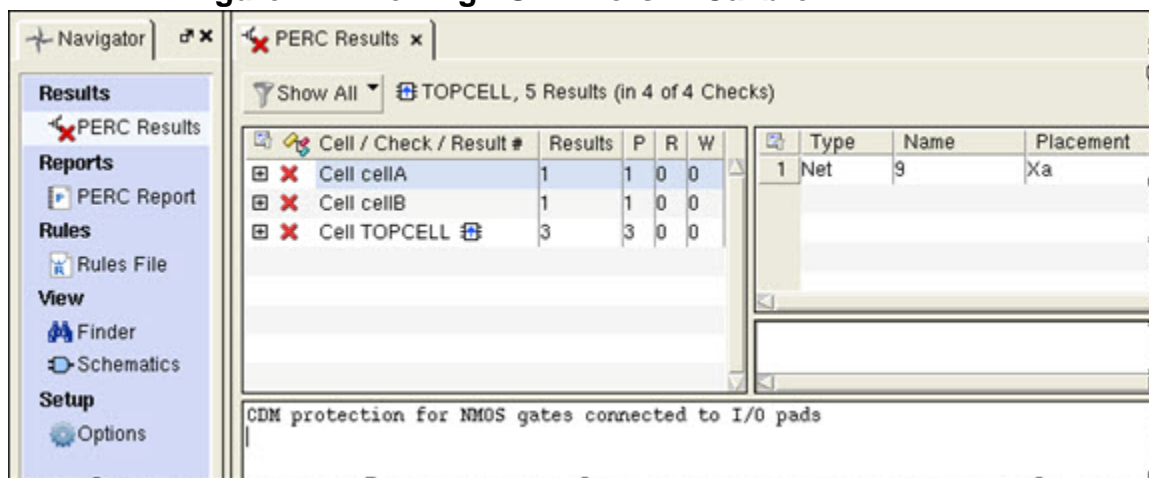
1. Verify your working directory is set to *Calbr\_qs\_ekit/Module7*.
2. Invoke Calibre RVE using the *svdb* file as input.

```
$CALIBRE_HOME/bin/calibre -rve -perc svdb
```

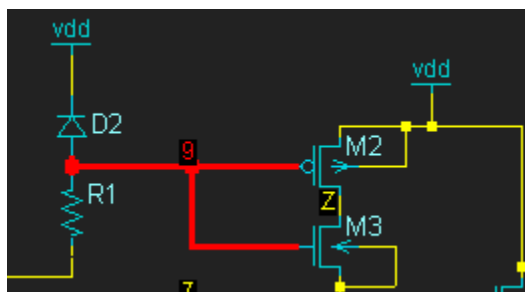
The Navigator pane, as shown in [Figure 7-2](#), contains icons that access the results, the rule file, the PERC Report, and other debugging features.

The PERC Results pane shows an expandable tree of results. By default, they are grouped by cell, then by rule. The results details are shown in the pane below the results tree.

Figure 7-2. Viewing ESD Errors in Calibre RVE



3. In the Navigator pane, click **Schematics**.  
The Source Netlist browser opens.
4. Click **Cell cellA** in the **PERC Results** tab.
5. Type H or click the **Highlight Selected Result** icon.  
The schematic viewer highlights the bad net in TOPCELL context.
6. Click and drag a box around the highlighted net to zoom in.



There is a missing “down” diode for this net.

7. Review other results using similar methods discussed previously.
8. Close Calibre RVE when you are finished by selecting **File > Exit**.

## Results

You have now run Calibre RVE and reviewed the Mask SVDB directory.

# Third-Party Information

For third-party information, refer to [\*Third-Party Software for Calibre Products\*](#).





# End-User License Agreement

The latest version of the End-User License Agreement is available on-line at:  
[www.mentor.com/eula](http://www.mentor.com/eula)

## IMPORTANT INFORMATION

**USE OF ALL SOFTWARE IS SUBJECT TO LICENSE RESTRICTIONS. CAREFULLY READ THIS LICENSE AGREEMENT BEFORE USING THE PRODUCTS. USE OF SOFTWARE INDICATES CUSTOMER'S COMPLETE AND UNCONDITIONAL ACCEPTANCE OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT. ANY ADDITIONAL OR DIFFERENT PURCHASE ORDER TERMS AND CONDITIONS SHALL NOT APPLY.**

## END-USER LICENSE AGREEMENT ("Agreement")

This is a legal agreement concerning the use of Software (as defined in Section 2) and hardware (collectively "Products") between the company acquiring the Products ("Customer"), and the Mentor Graphics entity that issued the corresponding quotation or, if no quotation was issued, the applicable local Mentor Graphics entity ("Mentor Graphics"). Except for license agreements related to the subject matter of this license agreement which are physically signed by Customer and an authorized representative of Mentor Graphics, this Agreement and the applicable quotation contain the parties' entire understanding relating to the subject matter and supersede all prior or contemporaneous agreements. If Customer does not agree to these terms and conditions, promptly return or, in the case of Software received electronically, certify destruction of Software and all accompanying items within five days after receipt of Software and receive a full refund of any license fee paid.

### 1. ORDERS, FEES AND PAYMENT.

- 1.1. To the extent Customer (or if agreed by Mentor Graphics, Customer's appointed third party buying agent) places and Mentor Graphics accepts purchase orders pursuant to this Agreement (each an "Order"), each Order will constitute a contract between Customer and Mentor Graphics, which shall be governed solely and exclusively by the terms and conditions of this Agreement, any applicable addenda and the applicable quotation, whether or not those documents are referenced on the Order. Any additional or conflicting terms and conditions appearing on an Order or presented in any electronic portal or automated order management system, whether or not required to be electronically accepted, will not be effective unless agreed in writing and physically signed by an authorized representative of Customer and Mentor Graphics.
- 1.2. Amounts invoiced will be paid, in the currency specified on the applicable invoice, within 30 days from the date of such invoice. Any past due invoices will be subject to the imposition of interest charges in the amount of one and one-half percent per month or the applicable legal rate currently in effect, whichever is lower. Prices do not include freight, insurance, customs duties, taxes or other similar charges, which Mentor Graphics will state separately in the applicable invoice. Unless timely provided with a valid certificate of exemption or other evidence that items are not taxable, Mentor Graphics will invoice Customer for all applicable taxes including, but not limited to, VAT, GST, sales tax, consumption tax and service tax. Customer will make all payments free and clear of, and without reduction for, any withholding or other taxes; any such taxes imposed on payments by Customer hereunder will be Customer's sole responsibility. If Customer appoints a third party to place purchase orders and/or make payments on Customer's behalf, Customer shall be liable for payment under Orders placed by such third party in the event of default.
- 1.3. All Products are delivered FCA factory (Incoterms 2010), freight prepaid and invoiced to Customer, except Software delivered electronically, which shall be deemed delivered when made available to Customer for download. Mentor Graphics retains a security interest in all Products delivered under this Agreement, to secure payment of the purchase price of such Products, and Customer agrees to sign any documents that Mentor Graphics determines to be necessary or convenient for use in filing or perfecting such security interest. Mentor Graphics' delivery of Software by electronic means is subject to Customer's provision of both a primary and an alternate e-mail address.

2. **GRANT OF LICENSE.** The software installed, downloaded, or otherwise acquired by Customer under this Agreement, including any updates, modifications, revisions, copies, documentation, setup files and design data ("Software") are copyrighted, trade secret and confidential information of Mentor Graphics or its licensors, who maintain exclusive title to all Software and retain all rights not expressly granted by this Agreement. Except for Software that is embeddable ("Embedded Software"), which is licensed pursuant to separate embedded software terms or an embedded software supplement, Mentor Graphics grants to Customer, subject to payment of applicable license fees, a nontransferable, nonexclusive license to use Software solely: (a) in machine-readable, object-code form (except as provided in Subsection 4.2); (b) for Customer's internal business purposes; (c) for the term of the license; and (d) on the computer hardware and at the site authorized by Mentor Graphics. A site is restricted to a one-half mile (800 meter) radius. Customer may have Software temporarily used by an employee for telecommuting purposes from locations other than a Customer office, such as the employee's residence, an airport or hotel, provided that such employee's primary place of employment is the site where the Software is authorized for use. Mentor Graphics' standard policies and programs, which vary depending on Software, license fees paid or services purchased, apply to the following: (a) relocation of Software; (b) use of Software, which may be limited, for example, to execution of a single session by a single user on the authorized hardware or for a restricted period of time (such limitations may be technically implemented through the use of authorization codes or similar devices); and (c) support services provided, including eligibility to receive telephone support, updates, modifications, and revisions. For the avoidance of doubt, if Customer provides any feedback or requests any change or enhancement to Products, whether in the course of receiving support or consulting services, evaluating Products, performing beta testing or otherwise, any inventions, product improvements, modifications or developments made by Mentor Graphics (at Mentor Graphics' sole discretion) will be the exclusive property of Mentor Graphics.

### 3. BETA CODE.

- 3.1. Portions or all of certain Software may contain code for experimental testing and evaluation (which may be either alpha or beta, collectively "Beta Code"), which may not be used without Mentor Graphics' explicit authorization. Upon Mentor Graphics' authorization, Mentor Graphics grants to Customer a temporary, nontransferable, nonexclusive license for experimental use to test and evaluate the Beta Code without charge for a limited period of time specified by Mentor Graphics. Mentor Graphics may choose, at its sole discretion, not to release Beta Code commercially in any form.
- 3.2. If Mentor Graphics authorizes Customer to use the Beta Code, Customer agrees to evaluate and test the Beta Code under normal conditions as directed by Mentor Graphics. Customer will contact Mentor Graphics periodically during Customer's use of the Beta Code to discuss any malfunctions or suggested improvements. Upon completion of Customer's evaluation and testing, Customer will send to Mentor Graphics a written evaluation of the Beta Code, including its strengths, weaknesses and recommended improvements.
- 3.3. Customer agrees to maintain Beta Code in confidence and shall restrict access to the Beta Code, including the methods and concepts utilized therein, solely to those employees and Customer location(s) authorized by Mentor Graphics to perform beta testing. Customer agrees that any written evaluations and all inventions, product improvements, modifications or developments that Mentor Graphics conceived or made during or subsequent to this Agreement, including those based partly or wholly on Customer's feedback, will be the exclusive property of Mentor Graphics. Mentor Graphics will have exclusive rights, title and interest in all such property. The provisions of this Subsection 3.3 shall survive termination of this Agreement.

### 4. RESTRICTIONS ON USE.

- 4.1. Customer may copy Software only as reasonably necessary to support the authorized use. Each copy must include all notices and legends embedded in Software and affixed to its medium and container as received from Mentor Graphics. All copies shall remain the property of Mentor Graphics or its licensors. Except for Embedded Software that has been embedded in executable code form in Customer's product(s), Customer shall maintain a record of the number and primary location of all copies of Software, including copies merged with other software, and shall make those records available to Mentor Graphics upon request. Customer shall not make Products available in any form to any person other than Customer's employees and on-site contractors, excluding Mentor Graphics competitors, whose job performance requires access and who are under obligations of confidentiality. Customer shall take appropriate action to protect the confidentiality of Products and ensure that any person permitted access does not disclose or use Products except as permitted by this Agreement. Customer shall give Mentor Graphics written notice of any unauthorized disclosure or use of the Products as soon as Customer becomes aware of such unauthorized disclosure or use. Customer acknowledges that Software provided hereunder may contain source code which is proprietary and its confidentiality is of the highest importance and value to Mentor Graphics. Customer acknowledges that Mentor Graphics may be seriously harmed if such source code is disclosed in violation of this Agreement. Except as otherwise permitted for purposes of interoperability as specified by applicable and mandatory local law, Customer shall not reverse-assemble, disassemble, reverse-compile, or reverse-engineer any Product, or in any way derive any source code from Software that is not provided to Customer in source code form. Log files, data files, rule files and script files generated by or for the Software (collectively "Files"), including without limitation files containing Standard Verification Rule Format ("SVRF") and Tcl Verification Format ("TVF") which are Mentor Graphics' trade secret and proprietary syntaxes for expressing process rules, constitute or include confidential information of Mentor Graphics. Customer may share Files with third parties, excluding Mentor Graphics competitors, provided that the confidentiality of such Files is protected by written agreement at least as well as Customer protects other information of a similar nature or importance, but in any case with at least reasonable care. Customer may use Files containing SVRF or TVF only with Mentor Graphics products. Under no circumstances shall Customer use Products or Files or allow their use for the purpose of developing, enhancing or marketing any product that is in any way competitive with Products, or disclose to any third party the results of, or information pertaining to, any benchmark.
  - 4.2. If any Software or portions thereof are provided in source code form, Customer will use the source code only to correct software errors and enhance or modify the Software for the authorized use, or as permitted for Embedded Software under separate embedded software terms or an embedded software supplement. Customer shall not disclose or permit disclosure of source code, in whole or in part, including any of its methods or concepts, to anyone except Customer's employees or on-site contractors, excluding Mentor Graphics competitors, with a need to know. Customer shall not copy or compile source code in any manner except to support this authorized use.
  - 4.3. Customer agrees that it will not subject any Product to any open source software ("OSS") license that conflicts with this Agreement or that does not otherwise apply to such Product.
  - 4.4. Customer may not assign this Agreement or the rights and duties under it, or relocate, sublicense, or otherwise transfer the Products, whether by operation of law or otherwise ("Attempted Transfer"), without Mentor Graphics' prior written consent and payment of Mentor Graphics' then-current applicable relocation and/or transfer fees. Any Attempted Transfer without Mentor Graphics' prior written consent shall be a material breach of this Agreement and may, at Mentor Graphics' option, result in the immediate termination of the Agreement and/or the licenses granted under this Agreement. The terms of this Agreement, including without limitation the licensing and assignment provisions, shall be binding upon Customer's permitted successors in interest and assigns.
  - 4.5. The provisions of this Section 4 shall survive the termination of this Agreement.
5. **SUPPORT SERVICES.** To the extent Customer purchases support services, Mentor Graphics will provide Customer with updates and technical support for the Products, at the Customer site(s) for which support is purchased, in accordance with Mentor Graphics' then current End-User Support Terms located at <http://supportnet.mentor.com/supportterms>.
6. **OPEN SOURCE SOFTWARE.** Products may contain OSS or code distributed under a proprietary third party license agreement, to which additional rights or obligations ("Third Party Terms") may apply. Please see the applicable Product documentation (including license files, header files, read-me files or source code) for details. In the event of conflict between the terms of this Agreement

(including any addenda) and the Third Party Terms, the Third Party Terms will control solely with respect to the OSS or third party code. The provisions of this Section 6 shall survive the termination of this Agreement.

## **7. LIMITED WARRANTY.**

- 7.1. Mentor Graphics warrants that during the warranty period its standard, generally supported Products, when properly installed, will substantially conform to the functional specifications set forth in the applicable user manual. Mentor Graphics does not warrant that Products will meet Customer's requirements or that operation of Products will be uninterrupted or error free. The warranty period is 90 days starting on the 15th day after delivery or upon installation, whichever first occurs. Customer must notify Mentor Graphics in writing of any nonconformity within the warranty period. For the avoidance of doubt, this warranty applies only to the initial shipment of Software under an Order and does not renew or reset, for example, with the delivery of (a) Software updates or (b) authorization codes or alternate Software under a transaction involving Software re-mix. This warranty shall not be valid if Products have been subject to misuse, unauthorized modification, improper installation or Customer is not in compliance with this Agreement. MENTOR GRAPHICS' ENTIRE LIABILITY AND CUSTOMER'S EXCLUSIVE REMEDY SHALL BE, AT MENTOR GRAPHICS' OPTION, EITHER (A) REFUND OF THE PRICE PAID UPON RETURN OF THE PRODUCTS TO MENTOR GRAPHICS OR (B) MODIFICATION OR REPLACEMENT OF THE PRODUCTS THAT DO NOT MEET THIS LIMITED WARRANTY. MENTOR GRAPHICS MAKES NO WARRANTIES WITH RESPECT TO: (A) SERVICES; (B) PRODUCTS PROVIDED AT NO CHARGE; OR (C) BETA CODE; ALL OF WHICH ARE PROVIDED "AS IS."
- 7.2. THE WARRANTIES SET FORTH IN THIS SECTION 7 ARE EXCLUSIVE. NEITHER MENTOR GRAPHICS NOR ITS LICENSORS MAKE ANY OTHER WARRANTIES EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO PRODUCTS PROVIDED UNDER THIS AGREEMENT. MENTOR GRAPHICS AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY.

8. **LIMITATION OF LIABILITY.** TO THE EXTENT PERMITTED UNDER APPLICABLE LAW, IN NO EVENT SHALL MENTOR GRAPHICS OR ITS LICENSORS BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY, EVEN IF MENTOR GRAPHICS OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL MENTOR GRAPHICS' OR ITS LICENSORS' LIABILITY UNDER THIS AGREEMENT EXCEED THE AMOUNT RECEIVED FROM CUSTOMER FOR THE HARDWARE, SOFTWARE LICENSE OR SERVICE GIVING RISE TO THE CLAIM. IN THE CASE WHERE NO AMOUNT WAS PAID, MENTOR GRAPHICS AND ITS LICENSORS SHALL HAVE NO LIABILITY FOR ANY DAMAGES WHATSOEVER. THE PROVISIONS OF THIS SECTION 8 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

## **9. THIRD PARTY CLAIMS.**

- 9.1. Customer acknowledges that Mentor Graphics has no control over the testing of Customer's products, or the specific applications and use of Products. Mentor Graphics and its licensors shall not be liable for any claim or demand made against Customer by any third party, except to the extent such claim is covered under Section 10.
- 9.2. In the event that a third party makes a claim against Mentor Graphics arising out of the use of Customer's products, Mentor Graphics will give Customer prompt notice of such claim. At Customer's option and expense, Customer may take sole control of the defense and any settlement of such claim. Customer WILL reimburse and hold harmless Mentor Graphics for any LIABILITY, damages, settlement amounts, costs and expenses, including reasonable attorney's fees, incurred by or awarded against Mentor Graphics or its licensors in connection with such claims.
- 9.3. The provisions of this Section 9 shall survive any expiration or termination of this Agreement.

## **10. INFRINGEMENT.**

- 10.1. Mentor Graphics will defend or settle, at its option and expense, any action brought against Customer in the United States, Canada, Japan, or member state of the European Union which alleges that any standard, generally supported Product acquired by Customer hereunder infringes a patent or copyright or misappropriates a trade secret in such jurisdiction. Mentor Graphics will pay costs and damages finally awarded against Customer that are attributable to such action. Customer understands and agrees that as conditions to Mentor Graphics' obligations under this section Customer must: (a) notify Mentor Graphics promptly in writing of the action; (b) provide Mentor Graphics all reasonable information and assistance to settle or defend the action; and (c) grant Mentor Graphics sole authority and control of the defense or settlement of the action.
- 10.2. If a claim is made under Subsection 10.1 Mentor Graphics may, at its option and expense: (a) replace or modify the Product so that it becomes noninfringing; (b) procure for Customer the right to continue using the Product; or (c) require the return of the Product and refund to Customer any purchase price or license fee paid, less a reasonable allowance for use.
- 10.3. Mentor Graphics has no liability to Customer if the action is based upon: (a) the combination of Software or hardware with any product not furnished by Mentor Graphics; (b) the modification of the Product other than by Mentor Graphics; (c) the use of other than a current unaltered release of Software; (d) the use of the Product as part of an infringing process; (e) a product that Customer makes, uses, or sells; (f) any Beta Code or Product provided at no charge; (g) any software provided by Mentor Graphics' licensors who do not provide such indemnification to Mentor Graphics' customers; (h) OSS, except to the extent that the infringement is directly caused by Mentor Graphics' modifications to such OSS; or (i) infringement by Customer that is deemed willful. In the case of (i), Customer shall reimburse Mentor Graphics for its reasonable attorney fees and other costs related to the action.
- 10.4. THIS SECTION 10 IS SUBJECT TO SECTION 8 ABOVE AND STATES THE ENTIRE LIABILITY OF MENTOR GRAPHICS AND ITS LICENSORS, AND CUSTOMER'S SOLE AND EXCLUSIVE REMEDY, FOR DEFENSE,

SETTLEMENT AND DAMAGES, WITH RESPECT TO ANY ALLEGED PATENT OR COPYRIGHT INFRINGEMENT OR TRADE SECRET MISAPPROPRIATION BY ANY PRODUCT PROVIDED UNDER THIS AGREEMENT.

**11. TERMINATION AND EFFECT OF TERMINATION.**

- 11.1. If a Software license was provided for limited term use, such license will automatically terminate at the end of the authorized term. Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement immediately upon written notice if Customer: (a) exceeds the scope of the license or otherwise fails to comply with the licensing or confidentiality provisions of this Agreement, or (b) becomes insolvent, files a bankruptcy petition, institutes proceedings for liquidation or winding up or enters into an agreement to assign its assets for the benefit of creditors. For any other material breach of any provision of this Agreement, Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement upon 30 days written notice if Customer fails to cure the breach within the 30 day notice period. Termination of this Agreement or any license granted hereunder will not affect Customer's obligation to pay for Products shipped or licenses granted prior to the termination, which amounts shall be payable immediately upon the date of termination.
- 11.2. Upon termination of this Agreement, the rights and obligations of the parties shall cease except as expressly set forth in this Agreement. Upon termination of this Agreement and/or any license granted under this Agreement, Customer shall ensure that all use of the affected Products ceases, and shall return hardware and either return to Mentor Graphics or destroy Software in Customer's possession, including all copies and documentation, and certify in writing to Mentor Graphics within ten business days of the termination date that Customer no longer possesses any of the affected Products or copies of Software in any form.
12. **EXPORT.** The Products provided hereunder are subject to regulation by local laws and European Union ("E.U.") and United States ("U.S.") government agencies, which prohibit export, re-export or diversion of certain products, information about the products, and direct or indirect products thereof, to certain countries and certain persons. Customer agrees that it will not export or re-export Products in any manner without first obtaining all necessary approval from appropriate local, E.U. and U.S. government agencies. If Customer wishes to disclose any information to Mentor Graphics that is subject to any E.U., U.S. or other applicable export restrictions, including without limitation the U.S. International Traffic in Arms Regulations (ITAR) or special controls under the Export Administration Regulations (EAR), Customer will notify Mentor Graphics personnel, in advance of each instance of disclosure, that such information is subject to such export restrictions.
13. **U.S. GOVERNMENT LICENSE RIGHTS.** Software was developed entirely at private expense. The parties agree that all Software is commercial computer software within the meaning of the applicable acquisition regulations. Accordingly, pursuant to U.S. FAR 48 CFR 12.212 and DFAR 48 CFR 227.7202, use, duplication and disclosure of the Software by or for the U.S. government or a U.S. government subcontractor is subject solely to the terms and conditions set forth in this Agreement, which shall supersede any conflicting terms or conditions in any government order document, except for provisions which are contrary to applicable mandatory federal laws.
14. **THIRD PARTY BENEFICIARY.** Mentor Graphics Corporation, Mentor Graphics (Ireland) Limited, Microsoft Corporation and other licensors may be third party beneficiaries of this Agreement with the right to enforce the obligations set forth herein.
15. **REVIEW OF LICENSE USAGE.** Customer will monitor the access to and use of Software. With prior written notice and during Customer's normal business hours, Mentor Graphics may engage an internationally recognized accounting firm to review Customer's software monitoring system and records deemed relevant by the internationally recognized accounting firm to confirm Customer's compliance with the terms of this Agreement or U.S. or other local export laws. Such review may include FlexNet (or successor product) report log files that Customer shall capture and provide at Mentor Graphics' request. Customer shall make records available in electronic format and shall fully cooperate with data gathering to support the license review. Mentor Graphics shall bear the expense of any such review unless a material non-compliance is revealed. Mentor Graphics shall treat as confidential information all information gained as a result of any request or review and shall only use or disclose such information as required by law or to enforce its rights under this Agreement. The provisions of this Section 15 shall survive the termination of this Agreement.
16. **CONTROLLING LAW, JURISDICTION AND DISPUTE RESOLUTION.** The owners of certain Mentor Graphics intellectual property licensed under this Agreement are located in Ireland and the U.S. To promote consistency around the world, disputes shall be resolved as follows: excluding conflict of laws rules, this Agreement shall be governed by and construed under the laws of the State of Oregon, U.S., if Customer is located in North or South America, and the laws of Ireland if Customer is located outside of North or South America or Japan, and the laws of Japan if Customer is located in Japan. All disputes arising out of or in relation to this Agreement shall be submitted to the exclusive jurisdiction of the courts of Portland, Oregon when the laws of Oregon apply, or Dublin, Ireland when the laws of Ireland apply, or the Tokyo District Court when the laws of Japan apply. Notwithstanding the foregoing, all disputes in Asia (excluding Japan) arising out of or in relation to this Agreement shall be resolved by arbitration in Singapore before a single arbitrator to be appointed by the chairman of the Singapore International Arbitration Centre ("SIAC") to be conducted in the English language, in accordance with the Arbitration Rules of the SIAC in effect at the time of the dispute, which rules are deemed to be incorporated by reference in this section. Nothing in this section shall restrict Mentor Graphics' right to bring an action (including for example a motion for injunctive relief) against Customer in the jurisdiction where Customer's place of business is located. The United Nations Convention on Contracts for the International Sale of Goods does not apply to this Agreement.
17. **SEVERABILITY.** If any provision of this Agreement is held by a court of competent jurisdiction to be void, invalid, unenforceable or illegal, such provision shall be severed from this Agreement and the remaining provisions will remain in full force and effect.
18. **MISCELLANEOUS.** This Agreement contains the parties' entire understanding relating to its subject matter and supersedes all prior or contemporaneous agreements. Any translation of this Agreement is provided to comply with local legal requirements only. In the event of a dispute between the English and any non-English versions, the English version of this Agreement shall govern to the extent not prohibited by local law in the applicable jurisdiction. This Agreement may only be modified in writing, signed by an authorized representative of each party. Waiver of terms or excuse of breach must be in writing and shall not constitute subsequent consent, waiver or excuse.