SILVACO

Application Note

Layout Verification in Batch Mode

1. Introduction

Verification of a layout design can be executed in GUI (Graphical User Interface) mode or batch mode in Silvaco's Custom IC CAD tools.

In this application note, we will introduce the method of executing the following layout verifications in the batch mode when using Expert version 4.2107.0.R or higher.

- DRC (Design Rule Check)
- LVS (Layout Versus Schematic)
- NVN (Netlist Versus Netlist)
- XOR
- Fill layers generation

The details of each run, regardless of the task, are specified in the run configuration file (RCF), the name of which is passed to the executable (smartdrc or smartnvn) as the first and mandatory command line argument.

2. Execution of DRC Rule File (Execution of SmartDRC)

Use the smartdrc command to execute the DRC rules in batch mode.

The parameters specifying the details of the DRC execution are specified as follows:

```
>smartdrc <run configuration file name>
<options>
```

The input layout file, DRC rule file, top cell name, and DRC execution options are specified as parameters in the run configuration file. The run results are saved in the working directory; DRC run summary in plain text, and the output DRC database with the name and format specified in the RCF. The SmartDRC/LVS run log is written to console and to <RunName>.log file.

Refer to "2.1 Invocation" of the SmartDRC/LVS User's Manual for the list of available command line options and to "2.3 DRC Run Configuration File" for details of each parameter.

Execution example 2-1:

```
>smartdrc sample_drc.rcf
```

Execution example 2-2:

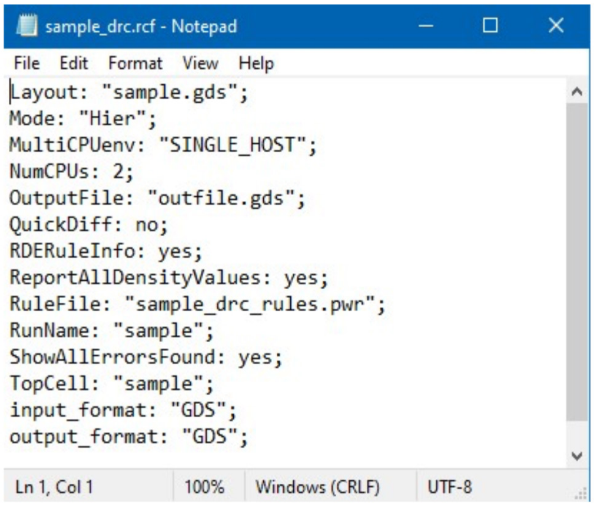
```
>smartdrc sample_drc.rcf > my.log
```

Example 2-2 will save the console output to a log file

An error message is displayed in the Windows command prompt or the Linux terminal when there is an error in the specified parameters of the RCF after executing SmartDRC.

```
>smartdrc sample_drc.rcf
>ERROR: Top cell is not defined
```

Any detected DRC errors are output to the DRC error database (a GDSII file, outfile.gds in the example) and are displayed in the layout after the DRC error database is loaded by Expert.



```
Layout: "sample.gds";
Mode: "Hier";
MultiCPUenv: "SINGLE_HOST";
NumCPUs: 2;
OutputFile: "outfile.gds";
QuickDiff: no;
RDERuleInfo: yes;
ReportAllDensityValues: yes;
RuleFile: "sample_drc_rules.pwr";
RunName: "sample";
ShowAllErrorsFound: yes;
TopCell: "sample";
input_format: "GDS";
output_format: "GDS";
```

Figure 1. Content of sample_drc.rcf

Note the NumCPUs parameter, which instructs Smart DRC to use two CPU cores. You can set its value to 1 or remove the line if you want to run the app in single-CPU mode. Setting any number up to 999 will make SmartDRC use the maximum number of licensed CPUs available with your hardware configuration. You can also specify the number of cores with the "–cpu" command line option (overrides the NumCPUs value in the RCF).

Copyright © 2022 Silvaco Inc.

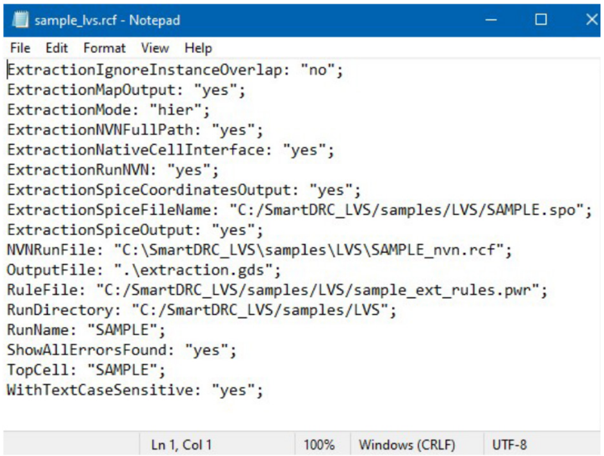
To view violations interactively in a layout Editor tool such as Expert, one must also enable the RDE RuleInfo parameter.

3. Extraction of Devices and Nets (Execution of SmartLVS)

The same smartdrc command is used to execute the LVS in batch mode.

Execution of SmartLVS is:

```
>smartdrc <LVS run configuration file name>
<options>
```



```
ExtractionIgnoreInstanceOverlap: "no";
ExtractionMapOutput: "yes";
ExtractionMode: "hier";
ExtractionNVNFullPath: "yes";
ExtractionNativeCellInterface: "yes";
ExtractionRunNVN: "yes";
ExtractionSpiceCoordinatesOutput: "yes";
ExtractionSpiceFileName: "C:/SmartDRC_LVS/samples/LVS/SAMPLE.spo";
ExtractionSpiceOutput: "yes";
NVNRunFile: "C:\SmartDRC_LVS\samples\LVS\SAMPLE_nvn.rcf";
OutputFile: ".\extraction.gds";
RuleFile: "C:/SmartDRC_LVS/samples/LVS/sample_ext_rules.pwr";
RunDirectory: "C:/SmartDRC_LVS/samples/LVS";
RunName: "SAMPLE";
ShowAllErrorsFound: "yes";
TopCell: "SAMPLE";
WithTextCaseSensitive: "yes";
```

Figure 2. Content of sample_lvs.rcf

The following input files are necessary to run LVS (Extraction and NVN):

- LVS run configuration file (RCF)
- NVN run configuration file
- Extraction rule deck
- Input layout database to be verified in binary format (GDSII, OASIS, Open Access)
- Schematic netlist (CDL, SPICE, VERILOG)

Hierarchical extraction requires schematic netlist and therefore an NVN RCF (see section 4) must always be present and contain the Schematic parameter.

Output files created by extraction:

- Extracted layout netlist, in SPICE format by default: <TopCell>.spo
- LVS summary file in text format: <TopCell>_summary.log
- LVS output database in binary format (GDSII, OASIS, Open Access)
- Various text reports generated on request

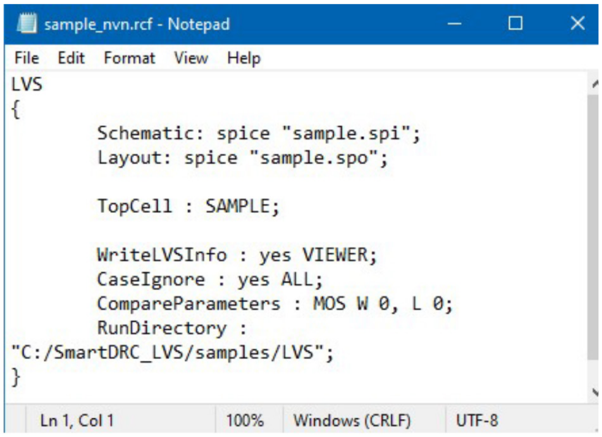
Refer to “9.1 Flow Description” of SmartDRC/LVS User’s Manual for details.

4. Netlists Comparison (Execution of SmartNVN)

The “smartnvn” command is used for comparison of netlists in batch mode.

Execution of SmartNVN is:

```
>smartnvn <NVN run configuration file name>
<options>
```



```
LVS
{
        Schematic: spice "sample.spi";
        Layout: spice "sample.spo";

        TopCell : SAMPLE;

        WriteLVSInfo : yes VIEWER;
        CaseIgnore : yes ALL;
        CompareParameters : MOS W 0, L 0;
        RunDirectory :
"C:/SmartDRC_LVS/samples/LVS";
}
```

Figure 3. Content of sample_nvn.rcf

Note the mandatory usage of LVS { } block in NVN RCF.

The following input files are necessary to run NVN:

- NVN run configuration file (RCF)
- Schematic netlist (CDL, SPICE, VERILOG format)
- Extracted layout netlist (CDL, SPICE, VERILOG, binary format)

The following output files are created by NVN:

- LVS summary file containing the list of pairs of compared cells in text format: <TopCell>.lvs.sum
- NVN error report in text format: <TopCell>.lvs.err
- On request, files containing description of schematic and layout netlists and the results of comparison: *.nav
- On request, NVN Results Viewer files: <TopCell>.nr file and <TopCell>_NVN_DB folder

Refer to “2.4 NVN Run Configuration File” of SmartDRC/LVS User’s Manual for details.

5. Execution of XOR

The command smartdrc is also used to execute the XOR in batch mode.

Execution of XOR is:

```
>smartdrc <XOR run configuration file name>
<options>
```

Copyright © 2022 Silvaco Inc.

You need to additionally define a second input file in ***LayoutFile2*** RCF parameter and a second top cell in ***Topcell2*** RCF parameter. You will also need to add “XOR: yes;” line to your RCF to switch the XOR mode on.

Refer to “7.1 Flow Description” of SmartDRC/LVS User’s Manual for details of XOR execution options.

The result is a geometric application of the widely used “exclusive-OR” Boolean logic. Similar to a regular DRC run, the results of XOR operation are a layout file (GDS,

OA, etc.) containing the differing shapes, one or more (in multi-CPU mode) logs, and a summary file.

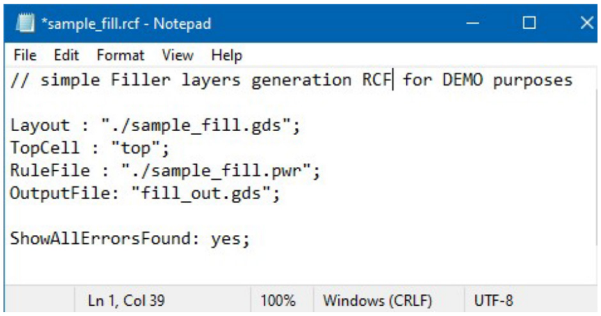
XOR operation needs a special rule file, but if it is not specified in XOR RCF, SmartDRC will create the file automatically.

6. Generation of Dummy Fill Layers

Fill operation is invoked by starting SmartDRC either with a special rule deck or with a mixed rule deck that combines DRC checks with fill operations.

Use the smartdrc command to execute the DRC rules in batch mode:

```
>smartdrc <run configuration file name>
<options>
```



```
// simple Filler layers generation RCF for DEMO purposes

Layout : "./sample_fill.gds";
TopCell : "top";
RuleFile : "./sample_fill.pwr";
OutputFile: "fill_out.gds";

ShowAllErrorsFound: yes;
```

Figure 4. Content of sample_fill.rcf

Running SmartDRC on a fill rule deck results in generating output.gds file with fill layers. The developer may attach these layers later as a separate group to some cell/top cell (hierarchically) using a layout editor.

You may merge your layout file with up to 10 previously saved fill layers files. To do that, you need to make sure that AllowDuplicateCells RCF key is set to “yes” (the default value) and list the fill layers files right after the layout file name in the value of **Layout** RCF parameter, e.g.:

Layout : “./sample1.gds” “./sample2.oas”;

Refer to “8.1 Flow Description” of SmartDRC/LVS User’s Manual for details of the $fill command and available options.

7. Conclusion

The layout verification can be executed from the specified layout file (GDSII, OASIS, and OA) using batch mode.

After the layout verification is executed in batch mode, it releases the license when the process is completed, allowing licenses to be used efficiently.

Copyright © 2022 Silvaco Inc.