

# Hints, Tips and Solutions

SILVACO PDK Development Group

## Q. How can I view a Foundry DRC run in *Expert*?

A. Before a Foundry will start creating the masks to fabricate a design, they normally will do some sort of drc verification with legacy tools. For the Analog Designer, it is always important to be able to understand the feedback from the Foundry. Silvaco makes that easier than ever since *Expert* can import two other commonly used error database formats: Calibre and DRACULA. Just request the error databases from the foundry, and you can save countless hours locating the errors that you didn't know you made!

First Open up the TOP level cell, in which the Foundry performed the DRC run. Make sure that the error database has the proper file extension, then select the import tab on the load DRC errors Dialog. You can now load and navigate through the errors, with your familiar and helpful DRC navigator from Silvaco International.

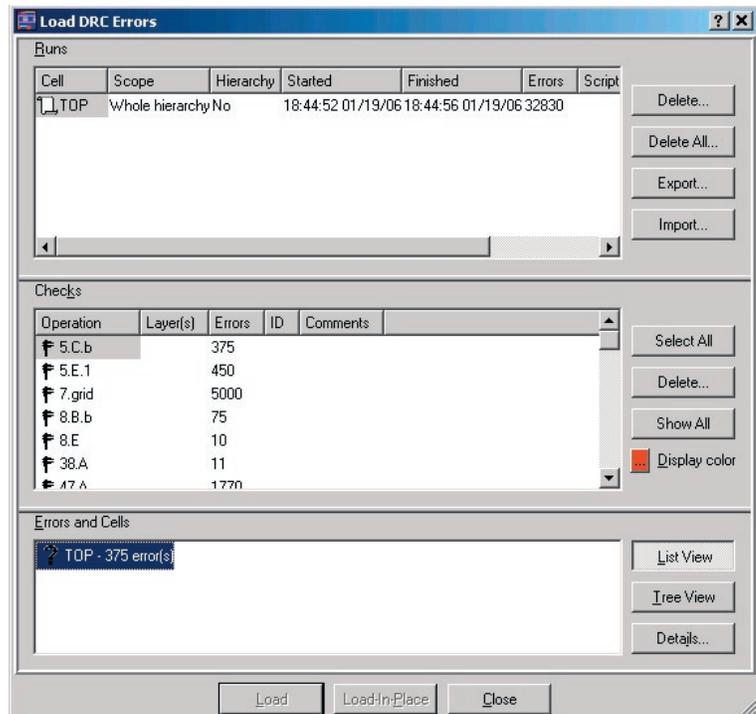


Figure 1.

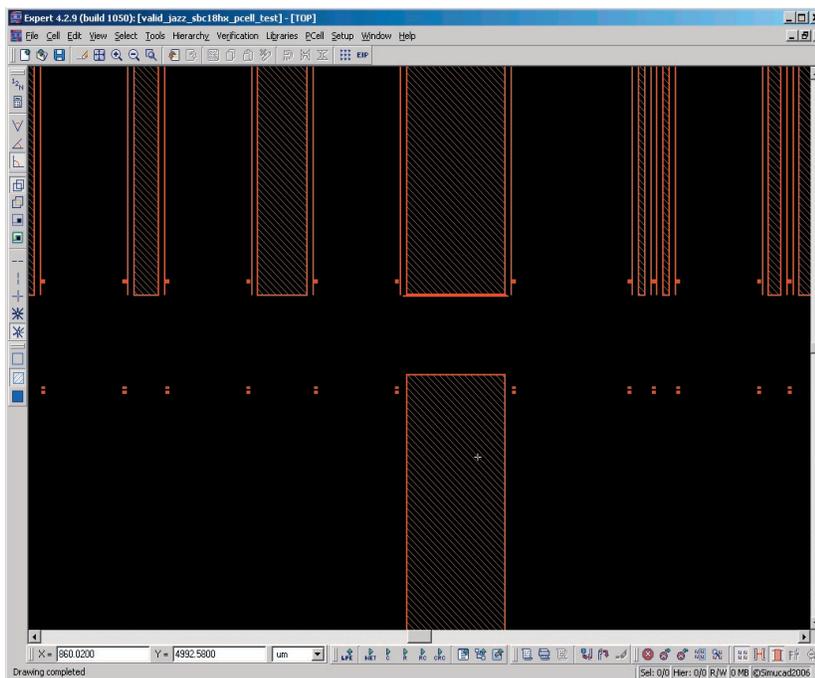


Figure 2.

## Q. How can I optimize DRC runtime?

- **Log file slows run time:** Writing the DRC log file takes a lot of time. The speed of the run time can be drastically improved if the “Write log file” option is disabled. One can also limit the total number of errors to be reported. In the figure below the total number of errors are 10,000. These options can be modified at Setup → Current DRC script run preference.
- **Running only sections of DRC that you need:** Executing only sections of DRC code or only on parts of layout also help to speed up run time. The scope of the DRC run in the above figure is set as “Whole Hierarchy”.
- **DRC guard:** DRC guard is used for checking the rules real time. Small DRC decks can be setup while doing routing so that an errors are detected immediately. For example, simple rules like width and spacing of metals can be coded in DRC guard.



Figure 1.

- **#DEFINE /#IFDEF adding:** Implementing the DRC with Conditional Branching Directives like #define, #ifdef #ifndef, #else, #endif, and Block grouping Directives like #EXEC, #SKIP, #stop makes it possible to execute specific parts of the script. Conditional Branching directives enables grouped statements to be executed based on values specified provided they are nested properly.

### Example:

```
#define M5 1
#define M6 0
#ifdef M6
... statements ignored here
#else
#ifdef M5
...statements ignored here
#else
...statements executed here
#endif
... statements executed here
#endif
... statements executed here
#ifdef M4
... statements ignored here, because M4 is undefined
#endif
```

- **Layout styles to improve productivity (partitioning the job):** Make sure that individual cells (building blocks) are DRC clean, before reusing them. This way the total number of errors can be cut down. Implementing the layout using a proper hierarchical structure enables faster DRC cleaning and editing if needed.
- **Reducing redundant operations:** Having repetitive code in the DRC deck slows down the run time. Temporary layers that will be used frequently can be defined as global variables, so that redundancy is eliminated.

### Call for Questions

If you have hints, tips, solutions or questions to contribute, please contact our Applications and Support Department  
Phone: (408) 567-1000 Fax: (408) 496-6080  
e-mail: support@silvaco.com

### Hints, Tips and Solutions Archive

Check our our Web Page to see more details of this example plus an archive of previous Hints, Tips, and Solutions  
[www.silvaco.com](http://www.silvaco.com)