ETH zürich

Institut für Integrierte Systeme
Integrated Systems Laboratory

Department of Information Technology and Electrical Engineering

EFCL Winter School 2026

Sample Solution Exercise 06

Finishing and DRC/LVS

F. K. Gürkaynak
Prof. L. Benini

Last Changed: 2026.02.09

1 Introduction

We are now in the final stages of the back-end design process. The journey began with floorplanning, where we defined the chip's layout, including block placement and area constraints. Then we moved on to placement, ensuring that standard cells and macros were optimally positioned while considering power distribution and timing constraints. Clock Tree Synthesis (CTS) was the next step followed by global routing and detailed routing, where the wires between the cells are first planned and then implemented on the metal layers.

At the end of routing, we would usually perform some final checks in OpenROAD, insert the fillers and then exported a DEF (Design Exchange Format) file, this was omitted but can be seen in the **`scripts/05_finishing.tcl`**. DEF contains abstract physical information (placements, orientations, routing, pins), but it does not contain the final geometric polygons needed for manufacturing. To obtain a manufacturable layout database we convert DEF to GDSII (Graphic Data System II) using KLayout. GDSII is the de-facto industry standard for representing IC layout data (geometries, layers, hierarchy, labels) and is the typical final output of physical design.

Before shipping a GDSII file to the foundry, we must perform final signoff verification:

- **DRC** (Design Rule Checking): checks the layout against foundry manufacturing rules.
- **LVS** (Layout Versus Schematic): checks that the layout electrically matches the intended circuit netlist.

1.1 Goals of this Exercise

By the end of this exercise, you should be able to:

- Understand the signoff process and why final verification is required.
- Run DRC and interpret violation reports; locate violations and fix them.
- Run LVS and interpret reports; identify and resolve mismatches.

1.2 About the Style

We will use a number of different styles to identify different types of actions as shown below:

> **Student Task:** Parts of the text that have a gray background, like the current paragraph, indicate steps required to complete the Exercise.

Actions that require you to select a specific menu will be shown as follows:
`menu→ sub-menu→ sub-sub-menu`

Whenever there is an option or a tab that can be found in the current view/menu we will use a BUTTON to indicate such an option.

Throughout the exercise, you will be asked to enter certain commands using the command-line[1].

```
sh> command to be entered on the Linux command line
```

1.3 Getting Started

> **Student Task 1:**
>
> - Start by navigating to your home and then the exercise directory
>
> ```
> sh> cd ~/ex06
> ```
>
> - Enter the `oseda` container's shell

[1] There are many reasons for using a command-line, some functionality can not not be accessed through GUI commands, and in some cases, using the command-line will be much faster. Most importantly, things you enter on the command line can be converted into a script and executed repeatedly.

```
sh> oseda bash
```

Note 1: If you ever open a new terminal window or tab, don't forget to re-enter the oseda containers shell, otherwise you won't be able to start the open-source tools.

2 Design Flow Recap

The DEF file exported by *OpenROAD* contains only abstract placement and routing information. For physical verification we need real polygons. Therefore the first step is *DEF* → *GDSII* conversion, which merges

- the DEF netlist / placement,
- the LEF abstracts,
- and the GDSII slices (polygons) of every standard-cell, I/O and macro.

The resulting GDSII is the single file that enters the Calibre / KLayout verification loop.

Student Task 2 (DEF → GDSII):

- Open a console and move to the **`klayout`** directory:

```
sh>  cd klayout
```

- Execute the conversion script:

```
sh>  oseda bash
sh>  ./run_finishing.sh --gds
```

- This takes the def from **`openroad/out/croc.def`** and creates the file **`klayout/out/croc.gds.gz`**.

3 Accessing the Layout

We inspect and edit the layout with *Siemens Calibre DesignRev* (DRV).

Student Task 3 (Load GDSII):

- Go into the **`calibre`** directory and use the `start_calibre` script to open Calibre. Note that you have to exit the `oseda` shell to run Calibre.
- `File→ Open Layout Files...` Navigate to **`klayout/out/croc.exercise.gds.gz`** and open it.
  (this is a pre-prepared GDS with known DRC and LVS properties, making it easier to manage the exercise)
- Optional: disable reference outlines via `Options→ Layout View...` and un-tick DRAW REFERENCE OUTLINES.

3.1 Hierarchy Levels

The design is hierarchical (9 levels). By default only the top level is shown; inner cells appear as empty boxes with instance names. Use the DEPTH indicator (bottom-right) to control which levels are displayed. Remember: independent of the view depth you may *edit* only the currently opened cell, usually the top level.

Student Task 4: Experiment with hierarchy depth, zooming, layer visibility and rulers.

4 Design Rule Checking (DRC)

DRC ensures every polygon respects the foundry rules (width, space, density, antenna, etc.). Violations lead to yield loss or outright manufacturing failure, therefore *zero DRC errors* is a hard requirement.

Student Task 5 (Start DRC):

- In DRV choose `Verification→ Run nmDRC...`
- Load the prepared run-set when asked (*Main DRC option*).
- In the *Customization Settings* window enable OFFGRID, ANGLE, FILLER, ANTENNA, SANITY and SWITCH OFF RECOMMENDED RULES CHECK.
- Press RUN DRC.

While DRC is running the *RVE* (Results Viewing Environment) window pops up. After completion a summary similar to the following appears:

```
TOTAL DRC Results Generated: 37 (37)
RULECHECK M2.b ........... 1
RULECHECK M2.d ........... 2
RULECHECK M3.a ........... 5
...
RULECHECK Ant.b_Metal4 ... 2
```

4.1 Fixing DRC Errors

We will walk through one example; the remaining 36 are left as an exercise.

Student Task 6 (Analyse M2.b):

- In RVE click the first M2.b violation and press H to highlight it in DRV.
- Raise visible hierarchy levels: press Shift+> three times.
- Show only *Metal-2* and the error marker layer.
- Use the ruler (R) to measure the notch: you will read $0.16\,\mu\mathrm{m}$, less than the required $0.21\,\mu\mathrm{m}$.
- Discuss with a TA how to close the notch (move edge, add polygon, etc.) and apply the fix while staying on the $5\,\mathrm{nm}$ manufacturing grid.

General editing advice:

- Always keep relevant via/contact layers visible when editing metals.
- Remain in *Top-Level Selection Mode* unless you *intend* to modify a cell reference.
- Use the grid (`View→ Grid→ 5 nm`) to guarantee on-grid coordinates.
- Undo is available via Ctrl+Z .

Student Task 7: Continue fixing until the summary shows

```
TOTAL DRC Results Generated: 0 (0)
```

and save the cleaned layout via `File→ Save`.

5 Layout Versus Schematic (LVS)

LVS reconstructs a netlist from the polygons (devices + wiring) and compares it to the golden gate-level netlist. A match proves that the layout electrically realises the intended circuit.

Student Task 8 (Run LVS):

- In DRV choose `Verification→ Run nmLVS...`
- Select the prepared *Main LVS option* run-set.
- Keep default inputs/outputs and press RUN LVS.

When LVS finishes the RVE window lists three categories:

Property Errors Device parameters differ (width, length, multiplier...).

Net Errors A net is missing or contains wrong devices.

Port Errors Top-level ports mismatch.

Expand any category, double-click an entry and DRV zooms to the offending structure. Typical fixes: correct device geometry, add missing via, remove unintended short, etc.

Student Task 9: Iterate until the LVS summary states

```
COMPARISON RESULTS:  CORRECT
```

Save the final GDSII, your chip is now sign-off clean and ready for tape-out.

Acknowledgement

Siemens *Calibre* is provided through the EDA Higher Education Software Program[2].

You are done with entire design flow from RTL to GDS.
Discuss your experience or continue to explore any part of the exercises.
We hope you enjoyed the EFCL 2026 Winter School and your stay with us!

[2] https://www.sw.siemens.com/en-US/academic/educators/eda-higher-education-software/