UCLA EE 201A -- VLSI Design Automation – Winter 2024
Lab 5: Design Rule Check (DRC)

TA: George Karfakis

****** IMPORTANT ******

- Run your experiments and do your work on SEASnet server eeapps.seas.ucla.edu
- All necessary files can be found in `/w/class.1/ee/ee201o/ee201ot2/2024_labs/lab5/`
- **See last page for important submission instructions**
- **Due Date: Saturday, March 16th, 11:59:59 PM.**

Notes:

- *You should do the problems in order.*
- *Please use Piazza for asking (and answering) questions before coming to office hours. However, you are expected to work on the assignment alone, so do not provide solutions or compare results with your peers.*
- *There are no contest points for this lab.*

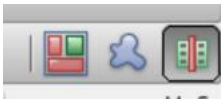
In this lab you will run a design rule check (DRC) on an implemented design (ISCAS89 s15850) with Cadence Innovus and Mentor Graphics Calibre, using the FreePDK 45nm technology library. You will also try to fix any errors you might find. The objective of this laboratory is for you to understand what are the typical DRC errors you can encounter and how to fix them. This is a critical step in any chip fabrication process – the foundry will not accept your GDS if it is not DRC clean!

For this assignment, you will need the following types of files, located in the Lab 5 materials on eeapps:

- Implemented Innovus design (s15850.enc and s15850.dat).
- Library GDS (in the ./gds/ folder) files and a layer map for streaming out the design (gds2_encounter.map).
- DRC rule deck (calibreDRC.rul) – this file describes the technology DRC rules.

Problem 1 (4 points): Run Innovus built-in DRC check.

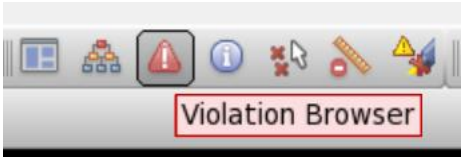
Open Innovus. Instead of implementing a design from scratch, we will open a saved post-implementation session. Once you have the GUI open, click File->Restore Design. Set “Innovus” as the datatype and use s15850.enc as the restore design file. After the design load, make sure you’re using the physical view (top right corner):



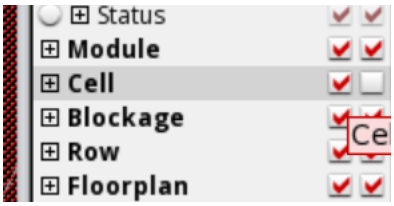
In the Innovus console, type “verify_drc” and run it. This command will run Innovus built in DRC checks. Answer the following question:

- How many errors did Innovus report?

To view the errors, in the innovus window, click on the violation browser:



The violation browsers categorizes errors based on the type and layer on which it occurred. By clicking on an error, the layout window will automatically zoom in on it. Make sure you enable the cell views, it will allow you to see metal 1 polygons within the standard cells themselves. Otherwise some of the errors won't make sense.



- What types of violations has Innovus found? What do they mean? Include example screenshots.

In the violation browser, there is an option to save the DRC report to file. Save it as s15850_initial.inv.drc and attach to your report.

Find a way to fix those violations within Innovus. You can do it manually, Innovus has tools for editing wires:

But there might be a better way of doing that. Consult Innovus documentation. Once you fixed all violations, re-run verify_drc, and write out a new report, s15850_fixed.inv.drc. Attach it to your report. In the report describe what tools and commands you used to fix the errors. Once that's done, stream out the GDS of your design using the following command, and attach it to your report:

```
streamOut s15850.gds -libName DesignLib -structureName s15850 -merge
{ ./gds/*.gds } -stripes 1 -units 10000 -mode ALL -mapFile gds2_encounter.map
```

Finally answer the following question:

- If Innovus is aware of those errors, why couldn't it avoid creating them during PNR in the first place?

Problem 2 (4 points): Run Calibre DRC check.

Innovus has indicated that the design is DRC clean – but is it really? Not necessarily. As you might have noticed, when running "verify_drc" we have not provided Innovus with the DRC deck. So it had to figure out the DRC rules based on some other file used in the implementation process.

- Based on your experience with Lab 4, which file do you think can be used by Innovus to find geometry information?
- Does that file have all the information you would find in the DRC rule deck? You can (and should) look at the DRC rule deck provided with this lab.
- If Innovus DRC checks are not comprehensive, why run them in the first place?

You will need to run the proper DRC checks, and we will use a tool called Calibre Workbench from Mentor Graphics to do that. It is a signoff tool for integrated circuits – it can run multiple other checks besides DRC, and assist with fixing errors. To open it, run "calibrewb". Then click File->Open Layout Files. Open the GDS you generated in Problem

1. Once it loads, first click on View->Change Hierarchy Depth->Depths, and set End Depth to 1. This will allow you to see “inside” the standard cells. Then click Verification->Run nmDRC. If a window asking for a “runset file” opens, close it. In the main DRC window, click on “Rules”, and then load “calibreDRC.rul” as the DRC Rules File. You can leave the other options as is, but you should take a look, so you know where the outputs are generated etc. Then click “Run DRC”. Wait for it to finish. Once it’s done, a new window called “Calibre RVE” should open, giving you the breakdown of the checks and errors.

- How many errors has Calibre found? What do they mean? Include example screenshots.

SUBMISSION INSTRUCTIONS (Read carefully)

- All necessary code, data, and report files must be tarballed and submitted as a single archive file on eeapps.
  - For final submission, create a directory named as follows: `Lastname-Firstname_UID_Username_Lab5/`
  - Inside this submission directory, you must include exactly eight files/subdirectories:
    - `Lastname-Firstname_UID_Username_Lab5_ s15850_initial.inv.drc` (Initial Innovus drc report from Problem 1)
    - `Lastname-Firstname_UID_Username_ s15850_fixed.inv.drc` (clean Innovus drc report from problem 1
    - `Lastname-Firstname_UID_Username_Lab5_s15850.gds` (layout file from Problem 1)
    - `Lastname-Firstname_UID_Username_Lab5_s15850_initial.drc.results` (Calibre DRC results from Problem 2)
    - `Lastname-Firstname_UID_Username_Lab5_s15850_inital.drc.summary` (Calibre DRC summary from Problem 2)
    - `Lastname-Firstname_UID_Username_Lab5_s15850_clean.drc.results` (Calibre DRC results from Problem 2)
    - `Lastname-Firstname_UID_Username_Lab4_s15850_clean.drc.summary` (Calibre DRC summary from Problem 2)
    - `Lastname-Firstname_UID_Username_Lab5_s15850_final.gds` (layout file from Problem 1)
  - Compress and archive this directory using tar/gzip to have a single submission file named `Lastname-Firstname_UID_Username_Lab5_pinXXXX.tar.gz` (Make up a 4-digit numeric PIN of your choice to substitute for `XXXX`)
  - Submit tarball by copying it to `/w/class.1/ee/ee201o/ee201ot2/submission/lab5/`
    - IMPORTANT: Make sure all files you submit, as well as the tarball, have full read and execute permissions to group and others or we will not be able to grade your lab. Do this using `chmod -R go+rx FILE_OR_DIRECTORY`
  - **Late submissions will not be accepted** (or possible – read/write/execute permissions to `submission/lab4/` will be disabled at the deadline). If you cannot finish the entire assignment on time, submit whatever you completed. **No submission = no points.** You are welcome to overwrite your submission as many times as you like before the deadline. However, please only use the filename format that is provided. Duplicates of the form "v1, v2, v3, new, newest," etc. will be ignored. If you accidentally submitted your tarball multiple times using different PINs, only the latest timestamp will be considered.
  - Some students may worry that their work could be visible to other students. We try to reduce this chance by disabling read permissions on `/w/class.1/ee/ee201o/ee201ota/submission/lab4/` directory so that other students cannot list its contents. Thus, they will not know the filename and cannot open your submission unless you give them your full name, student ID number, SEAS username, and arbitrary 4-digit PIN that you substituted for `XXXX` earlier on the tarball. This also means you will not be able to list the directory contents to see if your own submission worked – you can only write to it! In short, please do not give your classmates this information or collaborate on homeworks. The PIN can be whatever 4-digit number you want -- it is just to reduce the likelihood of another student guessing your full file submission path. You can change it for every lab submission if you like.
  - Submission example step-by-step:

```
$ cd /w/classproj/ee201a/mgottscho
$ mkdir Gottscho-Mark_203555232_gottscho_Lab5
$ cp foo.tcl Gottscho-Mark_203555232_gottscho_Lab5/Gottscho-
  Mark_203555232_gottscho_Lab5_1.tcl
# ...
$ cp -r <PREFIX>.invs.dat Gottscho-Mark_203555232_gottscho_Lab5/Gottscho-
  Mark_203555232_gottscho_Lab5_2.invs.dat
# ...
$ chmod -R go+rx Gottscho-Mark_203555232_gottscho_Lab5/
$ tar -czf Gottscho-Mark_203555232_gottscho_Lab4_pin6276.tar.gz Gottscho-
  Mark_203555232_gottscho_Lab5/
```

```
$ chmod go+rx Gottscho-Mark_203555232_gottscho_Lab5_pin6276.tar.gz
$ cp Gottscho-Mark_203555232_gottscho_Lab5_pin6276.tar.gz
  /w/class.1/ee/ee201o/ee201ota/submission/lab4/Gottscho-
  Mark_203555232_gottscho_Lab4_pin6276.tar.gz
```