UCLA EE 201A -- VLSI Design Automation – Winter 2024
Lab 4: Physical Design using Cadence Innovus

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/lab4`
- **See last page for important submission instructions**
- **Due Date: Friday, Feb 16th , 2024 at 11:59:59 P.M. Pacific Daylight Time (PDT)**

Notes:

- *You will find it useful to refer to the Innovus tutorial slides uploaded to Piazza as well as the official Cadence Innovus documentation, located at `/w/apps3/Cadence/INNOVUS161/doc/{innovusUG, innovusTCR}/` on the eeapps server. You can also use help documentation from within the tool.*
- *It is strongly suggested you use Innovus in GUI mode to gain intuition for the processing steps you will be doing. Since you are logging in remotely, one way is to use X window forwarding. Please do a Web search if you don't know how to set this up for your personal computer.*
- *Use the skeleton Tcl code provided in the Lab 4 materials directory listed above. The scripting language is the same as that for Lab 2 where you used Cadence Genus. However, the commands are for Cadence Innovus, which is a different tool. Don't get confused by the two tools using similar Tcl scripts.*
- *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 perform physical design of a small circuit (ISCAS89 s1494) on *Cadence Innovus* (which is the new tool version of the more well-known *Cadence Encounter)*, using the Nangate 45nm technology library. Essentially, you will be doing place and route. The objective of this lab is to give the flavor of a chip physical implementation process and help you develop insight into some of the issues in physical design. Chip physical implementation process typically starts from a gate-level netlist (along with timing constraints, timing simulation results) and ends with a finished design (GDSII sent for fabrication, or "taped out"). The physical implementation of a design is a lengthy process, made correct over many iterations, with many detailed steps. This lab serves as a basic introduction to this process.

For this assignment, you basically need the following types of files, located in the Lab 4 materials on eeapps:

- Gate-level netlist (.v file, generated by synthesis tool, e.g., Cadence Genus)
- Timing constraints (.sdc file, generated by synthesis tool)
- LEF for library/technology (.lef file, Nangate/FreePDK)
- Liberty timing specs for library (.lib file, Nangate)
- GDS for Nangate library cells (.gds file, to write final design GDS for tapeout)

You could try doing physical design using Innovus' GUI tools. In fact, we encourage you to play with different features and commands and get a feel for what is going on through the visualization of the layout. However, to answer the problems and submit your solutions, you need to script all of your processing steps. Anything you can do in the GUI can also be done on the command line.

We've provided a `lab4_skeleton.tcl` script to help you get started. All of your solutions to the problems below should be implemented using changes to the Tcl script.

Problem 1 (3 points): Implement your design with minimum physical layout area.

Determine the minimum design area while meeting setup and hold timing constraints and without any design-rule violations. You will need to modify the provided Tcl script. You may not change the number of available routing layers nor the method by which geometry violations are checked.

Report Deliverables: In your lab report, describe the steps you took to achieve your minimum area. Report both the initial and final cell utilization statistics, where initial refers to your target utilization, and final refers to the actual result the tool achieved. Also report the worst-case setup-time and hold-time slack for your finished design.

Hint: Minimum area corresponds to maximum cell utilization.

Problem 2 (4 points): Report timing and power at different steps of physical design.

Using your script that generated your final solution to Problem 1, report timing (both setup and hold) and power before placement, after placement, after global routing, after detailed routing, after each `optDesign` command, after RC (resistor-capacitor, not RTL Compiler) parasitic extraction, and the final result (which should match that of Problem 1). Now disable timing-driven placement in your script and repeat.

Report Deliverables: Include the timing (setup and hold) and power results for each step of the physical design process as listed above. Explain the differences, if any, when you disabled timing-driven placement. What does this feature do? Explain.

Hint: Read Innovus documentation to understand what timing-driven placement does.

Problem 3 (3 points): Modify the power structure.

Add VDD/ground power stripes to your design's power structure using the GUI or command line (Tcl script). Place the stripes in the vertical orientation on the Metal 2 layer (M2) with 210 nm width, stripe-to-stripe spacing of 4.13, and 2 micron offset from the left edge of the design core. Run your physical design flow again.

Report Deliverables: Describe what you did and report initial/final cell utilization and worst-case setup and hold timing before and after adding power stripes. Can you achieve the same cell utilization without timing/geometric violations as you could in Problem 1? What is the new best final utilization with no violations? What is the effect of adding power stripes on delay and power? Explain.

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_Lab4/`
  - Inside this submission directory, you must include exactly eight files/subdirectories:
    - `Lastname-Firstname_UID_Username_Lab4_1.tcl` (physical design script for Problem 1)
    - `Lastname-Firstname_UID_Username_Lab4_1.invs` (physical design output for Problem 1)
    - `Lastname-Firstname_UID_Username_Lab4_1.invs.dat` (physical design output subfolder for Problem 1)
    - `Lastname-Firstname_UID_Username_Lab4_2.tcl` (physical design script for Problem 2)
    - `Lastname-Firstname_UID_Username_Lab4_3.tcl` (physical design script for Problem 3)
    - `Lastname-Firstname_UID_Username_Lab4_3.invs` (physical design output for Problem 3)
    - `Lastname-Firstname_UID_Username_Lab4_3.invs.dat` (physical design output subfolder for Problem 3)
    - `Lastname-Firstname_UID_Username_Lab4.pdf` (lab report – answer any questions here)
    - `Lastname-Firstname_UID_Username_Lab4_results_submission.txt` (fill in the results submission txt file here)
  - Compress and archive this directory using tar/gzip to have a single submission file named `Lastname-Firstname_UID_Username_Lab4_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/lab4/`
    - 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_Lab4
$ cp foo.tcl Gottscho-Mark_203555232_gottscho_Lab4/Gottscho-
  Mark_203555232_gottscho_Lab4_1.tcl
# ...
$ cp -r <PREFIX>.invs.dat Gottscho-Mark_203555232_gottscho_Lab4/Gottscho-
  Mark_203555232_gottscho_Lab4_2.invs.dat
# ...
$ chmod -R go+rx Gottscho-Mark_203555232_gottscho_Lab4/
$ tar -czf Gottscho-Mark_203555232_gottscho_Lab4_pin6276.tar.gz Gottscho-
  Mark_203555232_gottscho_Lab4/
$ chmod go+rx Gottscho-Mark_203555232_gottscho_Lab4_pin6276.tar.gz
$ cp Gottscho-Mark_203555232_gottscho_Lab4_pin6276.tar.gz
  /w/class.1/ee/ee201o/ee201ot2/submission/lab4/Gottscho-
  Mark_203555232_gottscho_Lab4_pin6276.tar.gz
```