UCLA EE 201A -- VLSI Design Automation – Winter 2024
Lab 1: Using OpenAccess

TA: George Karfakis

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

- Run your experiments and do your work on SEASnet server `eeapps.seas.ucla.edu`
- You are responsible for backing up your code & data
- All necessary files can be found in `/w/class.1/ee/ee201o/ee201ot2/2024_labs/lab1`
- **See last page for important submission instructions**
- **Due Date: Tuesday, January 23, 2024 at 11:59:59 P.M. Pacific Daylight Time (PDT)**

Notes:

- *You will find it useful to refer to the OA tutorial from class and the official OA documentation.*
- *Do problem 1 first as you will need the OA database for problems 2 and 3. Please use the "`DesignLib`" as the OA database library's name, as shown in the appendix.*
- *Problems 2 and 3 are based on the provided `lab1.cpp` template. Please implement both in that C++ file where indicated.*
- *Please use Piazza for asking (and answering) questions before office hours. However, you are expected to work on the assignment alone, so do not provide solutions or compare results with your peers.*

Problem 1 (1 point): Import design into OpenAccess database.

Import design information given in Verilog and DEF as well as library information given in LEF into **one** OA database (DesignLib). In your report, please include a list of the commands you used.

Hint: see Appendix

Problem 2 (3 points): Compute/plot the fan-out distribution of the design.

Write a small routine in C++, using the OA API and OA database, to compute the overall average fan-out in the design. Include the average fan-out value and plot the fan-out distribution in your lab report. You may wish to use MATLAB or a similar tool for generating the plot. In the report, please explain any assumptions you made and your methodology.

Note: fan-out = number of inputs driven by a gate's output.
Hint: oaBlock->(oaIter)->oaNet->(oaInstTerm, oaTerm).

Problem 3 (6 points): Compute/plot the wire length distribution of the design.

Write a small routine in C++, using the OA API and OA database, to compute the half-perimeter wire length (HPWL) of all nets with 2 ends in the design, across all metal layers. Include the average HPWL value and plot the HPWL distribution in your lab report. You may wish to use MATLAB or a similar tool for generating the plot. In the report, please explain any assumptions you made and your methodology.

Hint: oaBlock->(oaIter)->oaNet->oaInstTerm->oaInst->(getOrigin)->oaPoint

APPENDIX

How to import a design into OA database:

You should follow a specific procedure for translating LEF/DEF/Verilog into OA. Here's a rough example from some OA documentation (one LEF for in our case).

Note: The translator examples below include many extra options that might be unneeded for our case. You should use the simplest version of the commands that solves the purpose.

The following list shows the steps for translating legacy data into OpenAccess.

Use `lef2oa` to create the reference libraries and the technology data.
Use `verilogAnnotate` to establish terminal order and to group the multi-bit interfaces of the reference library designs.
Use `verilog2oa` to import the logical description of the design.
Use `def2oa` to annotate the logical description of the design with the physical implementation.
Use `spef2oa` to annotate parasitics onto the design. (Optional)

For example, consider a design that consists of:

Two LEF files: `foundry18u7lm.lef` and `stdLib.lef`
A Verilog file `myDesign.v` (which contains Verilog module definitions of every cell in the design, including the reference library cells)
A DEF file `myDesign.def`
A SPEF file `myDesign.spef`

The basic process to import this design into OpenAccess uses the following flow:

```
$ lef2oa -lib foundryLib -lef foundry18u7lm.lef
$ lef2oa -lib stdLib -lef stdLib.lef -techLib foundryLib
$ verilogAnnotate -refLibs "foundryLib stdLib" -verilog myDesign.v
$ verilog2oa -lib DesignLib -refLibs "foundryLib stdLib"  -view layout -
viewType maskLayout -verilog myDesign.v
def2oa -lib DesignLib -cell top -view layout -def myDesign.def -refLibs
"foundryLib stdLib" -techLib foundryLib
$ spef2oa -lib DesignLib -cell top -view layout -spef myDesign.spef -ap
typ
```

This process works for simple designs that are not partitioned for implementation. If the design is partitioned, you must modify the process to accommodate the partitions.

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_Lab1/`
  - Inside this submission directory, you must include exactly three files:
    - `Lastname-Firstname_UID_Username_Lab1.cpp` (source code for lab)
    - `Lastname-Firstname_UID_Username_Lab1` (compiled binary file for lab)
    - `Lastname-Firstname_UID_Username_Lab1.pdf` (lab report – answer any questions here)
  - Compress and archive this directory using tar/gzip to have a single submission file named `Lastname-Firstname_UID_Username_Lab1_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/lab1/`
    - 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/lab1/` 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/lab1/` 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 <YOUR_LAB1_WORK_DIRECTORY>
$ mkdir Gottscho-Mark_203555232_gottscho_Lab1
$ cp foo.cpp Gottscho-Mark_203555232_gottscho_Lab1/Gottscho-
  Mark_203555232_gottscho_Lab1.cpp
$ cp bar Gottscho-Mark_203555232_gottscho_Lab1/Gottscho-
  Mark_203555232_gottscho_Lab1
$ cp report.pdf Gottscho-Mark_203555232_gottscho_Lab1/Gottscho-
  Mark_203555232_gottscho_Lab1.pdf
$ chmod -R go+rx Gottscho-Mark_203555232_gottscho_Lab1/
$ tar -czf Gottscho-Mark_203555232_gottscho_Lab1_pin1543.tar.gz Gottscho-
  Mark_203555232_gottscho_Lab1/
$ chmod go+rx Gottscho-Mark_203555232_gottscho_Lab1_pin1453.tar.gz
$ cp Gottscho-Mark_203555232_gottscho_Lab1_pin1543.tar.gz
  /w/class.1/ee/ee201o/ee201ota/submission/lab1/Gottscho-
  Mark_203555232_gottscho_Lab1_pin1543.tar.gz
```