CS6135 VLSI Physical Design Automation

Homework 5: Automated P&R for Analog Circuits

Due: 23:59, January 17 2023

---

1. Introduction

In this homework, you are asked to write programs to place and route some analog circuits, and you need to output the results in the DEF (Design Exchange File) format.

2. Problem Description

We will provide you with a Python program for placement and routing (P&R) of 4 current sources each having 4 units, which is not yet completed. You must follow the descriptions in the lecture notes to complete the Python program. After completing the Python program, you will need to convert it into a version written in C/C++.

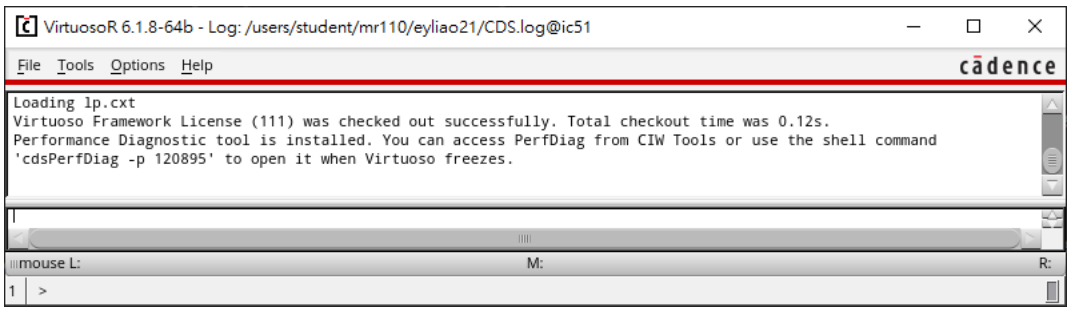
Furthermore, the original Python program is designed to perform placement and routing on a circuit that converts 4 current sources into a 16-unit array. In your C/C++ program, you will also need to generalize it for handling circuits with 16, 36, 64, and 100 current sources (as shown in the lecture notes), where each current source has 4 units.

3. How to Screen Shot Your Result

You can visualize the placement and routing results of circuits produced by your program in Virtuoso and take a screenshot for each of them. Please follow the instructions below to visualize your placement and routing results using Virtuoso:

Step 1. Invoke Virtuoso

```
$ cd HW5_Visual/
$ setenv OA_HOME /usr/cad/cadence/IC/cur/oa_v22.60.074
$ virtuoso &
```

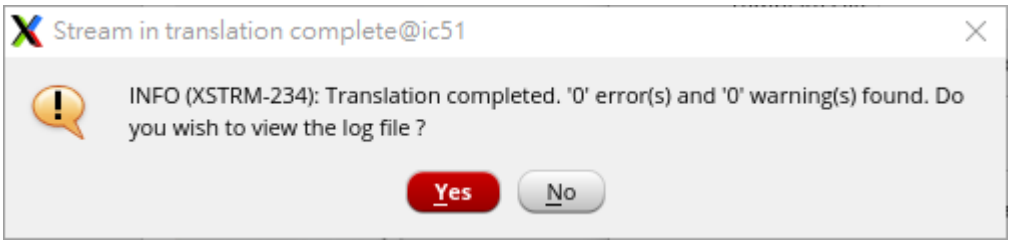


Step 2. Import “cells.gds”

Window > File > Import > Stream…

Stream File	./GDSII/cells.gds
Library	CSarray

Then click **Translate**



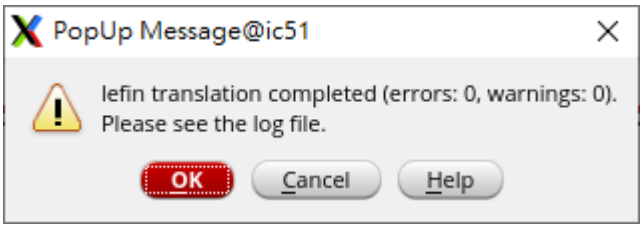
Then click **Yes**

Step 3. Import “CS.lef”

Window > File > Import > LEF…

LEF File Name	./LEF/CS.lef
Target Library Name	CSarray

Then click **OK**



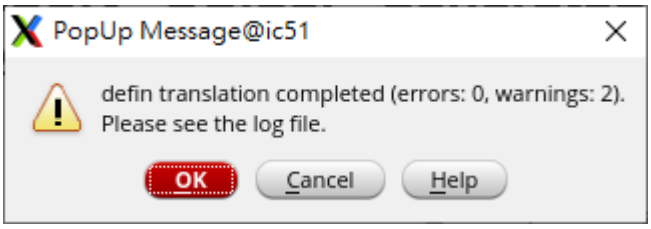
Then click **OK**

Step 4. Import Your DEF File

Window > File > Import > DEF…

DEFIn File Name	./{your DEF filename}.def
Target Library Name	CSarray

Then click **OK**



Then click **OK**

Step 5. Open the Layout

Windows > File > Open…

File

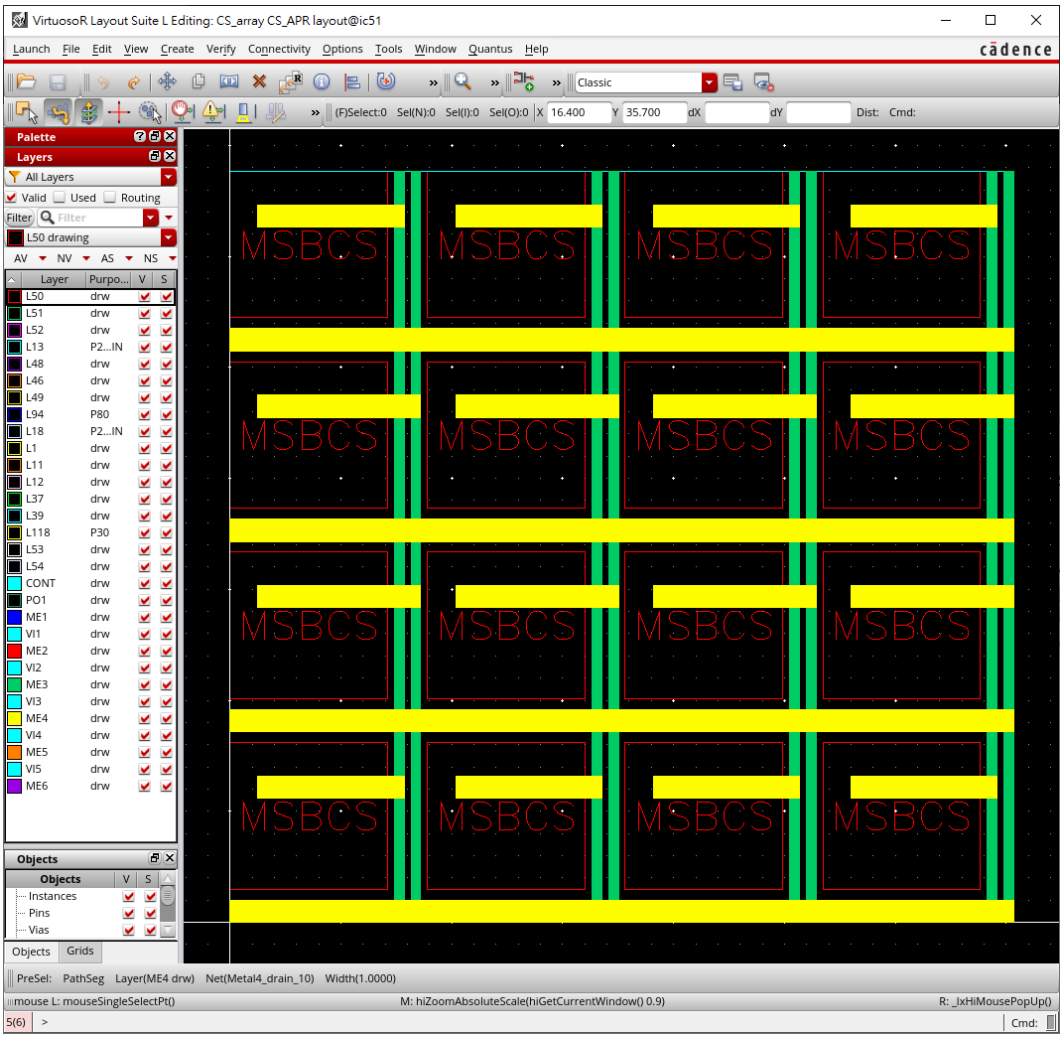
Library	CSarray
Cell	CS_APR

Application

Open with	Layout GXL

Then click **OK**

Then it will open a layout window like the following figure. (You can click on the **F** key to fit the design to the window size. Please remember to fit the design to the window size before you take a screenshot.)



4. Language/Platform

(1) Language: Python/C/C++

Platform: Unix/Linux

5. Report

Your report must contain the following contents, and you can add more as you wish.

(1) Your name and student ID

(2) How to compile and execute your C/C++ program and give an execution example.

(3) The details of your C/C++ program. How do you generalize the original C/C++ program to handle 16 or more current sources? You have to describe what you do step by step in detail.

(4) The screenshots of your placement and routing results for the circuit produced by your Python program for the case of 4 current sources as well as by your C/C++ program for the cases of 4, 16, 36, 64, and 100 current sources.

6. Required Items

Please compress `HW5/` (using tar) into one with the name `CS6135_HW5_${StudentID}.tar.gz` before uploading it to eeclass.

(1) `bin/` contains your executable file.

(2) `output/` contains all the DEF files generated by your C/C++ program with different amounts of current sources for TAs to verify.

(3) `python/` contains the completed `main.py`, and a directory `DEF/` that has the DEF file `CS_4.def` generated by your Python program.

(4) `src/` contains all your C/C++ source code, `Makefile` and `README`.

- `README` must describe how to compile and execute your C/C++ program. An example is like the one shown in HW2.

(5) `CS6135_HW5_${STUDENT_ID}_report.pdf` contains your report.

You can use the following command to compress your directory on a workstation:

```
$ tar -zcvf CS6135_HW5_{StudentID}.tar.gz <directory>
```

For example:

```
$ tar -zcvf CS6135_HW5_11162500.tar.gz HW5/
```

7. Grading

- ✓ **20%:** The completeness of your python code.
- ✓ **40%:** The generalizability of your C/C++ program. We will test your code with different numbers of current sources, i.e., 4, 16, 36, 64, and 100.
- ✓ **40%:** The completeness of your report.

Notes:

- You must ensure that the prefixes (`Transistor, Metal3_, Metal4_drain_, Metal4_port_, Via34_drain2ME3_, Via34_port2ME3_,…`) of the names of components and nets in a DEF file produced by your Python program or C/C++ program are not changed.
- Make sure the following commands can be executed.
  - Go to directory "`src/`", enter "`make`" to compile your C/C++ program and generate the executable file, called "`hw5`", which will be in directory "`bin/`".
  - Go to directory "`src/`", enter "`make clean`" to delete your executable file.
- Please use the following command format to run your C/C++ program.

  ```
  $ ./hw5 [number of current sources] [def file path]
  ```

  E.g.:

  ```
  $ ./hw5 4 ../output/CS_4.def
  ```
- We will test your program by a shell script with GCC 9.3.0 on ic51. Please make sure your C/C++ program can be executed by **HW5_grading.sh** (Your programming score will be 0 if your program cannot be executed by the shell script).
- Note that any form of plagiarism is strictly prohibited. If you have any question about the homework, please contact TAs. (If you have any questions about Python /C/C++ programming, you should try to google it by yourself first.)