CS6135 VLSI PDA HW5

111062684　林鼎勳

1) How to compile and execute your program, and give an execution example:

- Compile: Go into the directory “HW5/src/” and enter the command:

  `$ make`

  This will generate the executable file “hw5” in the directory “HW5/bin/”.

- Execute: In the same directory (“HW5/src/”) and enter the command:

  `$ make test`

  Then, the terminal will let you enter the number of current sources. We take "4" for example. This will execute "hw5" and read the number "4" as input and write the result into "CS_4.def" in the directory "../output". After execution, it will automatically run the verification program to test the result.

2) The details of your C/C++ program. How do you generalize the original C/C++ program to handle 16 or more current sources?

In the first step, we need to know how the number of current sources affects the placement and routing of the analog circuit. Here we use CS to represent the number of current sources.

- Number of cells(units): `CS*4`

  This is because each current source has 4 units.

- Number of horizontal / vertical cells: `RC = 2*(CS^0.5)`

  After we get the number of cells, we can immediately find out the number of horizontal / vertical cells. Here we use RC to represent this number.

- Number of vertical metals ME3: `RC*(CS^0.5)`

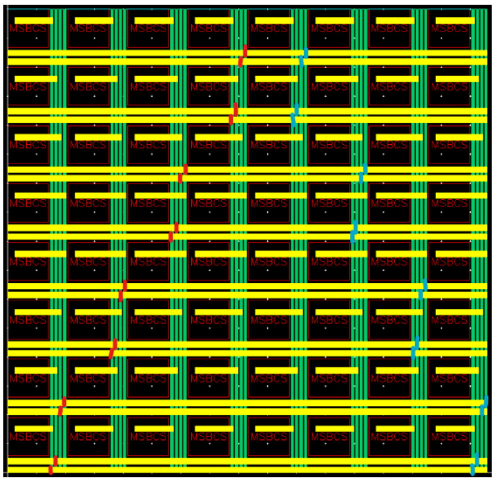
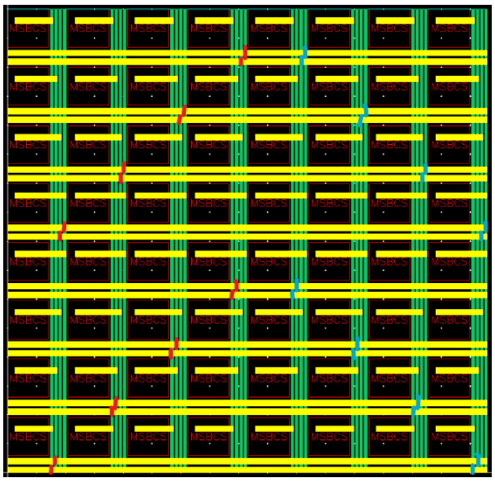
  In each vertical column, we need `(CS^0.5)` vertical metals ME3, and there are RC columns.

- Number of horizontal metals ME4 port: `RC*(CS/RC)`

  In each horizontal row, we need `(CS/RC)` horizontal metals ME4 port, and there are RC rows.

In the second step, after we put all cells and long metals, we now put in small horizontal metals to establish the connection between cells and long vertical metals. The main idea is the same as the lecture: first put in the left-bottom corner, then symmetrically fill in the remaining corner. At the same time, after we decide the coordinate of small horizontal metals, we can also decide the coordinate of the via which connects these two metals.

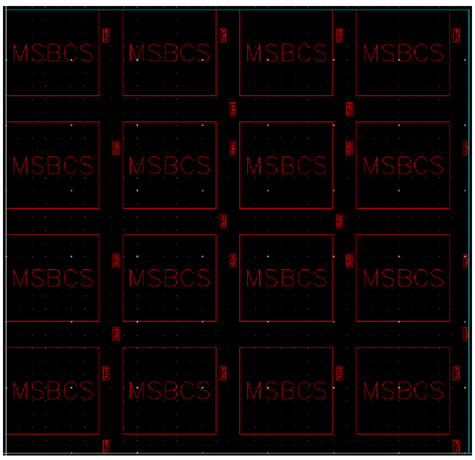
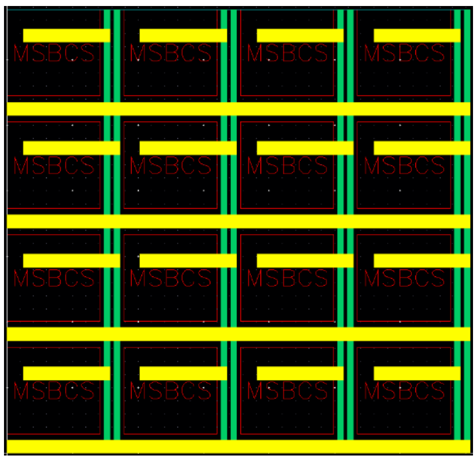
The third step, and also the last step, is to decide the via to connect the long vertical metals and long horizontal metals. From the lecture in class, there seem to be two ways to connect, and both are common centroid. The graphs are shown below:

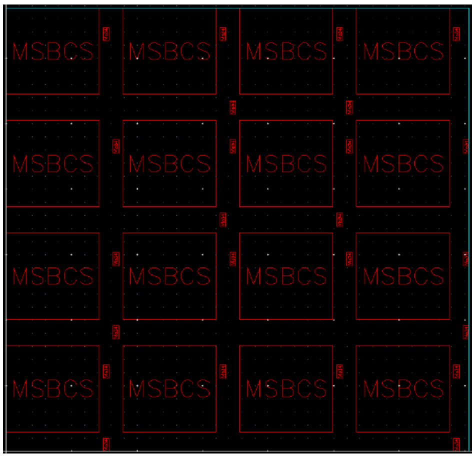
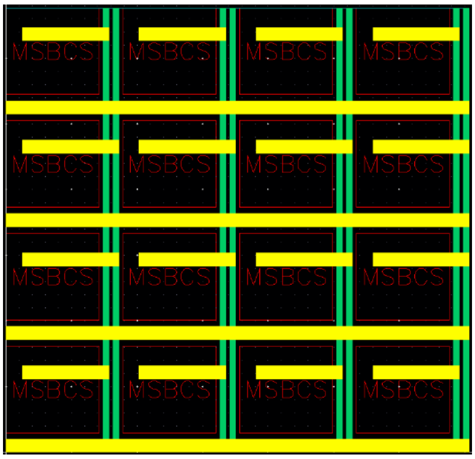
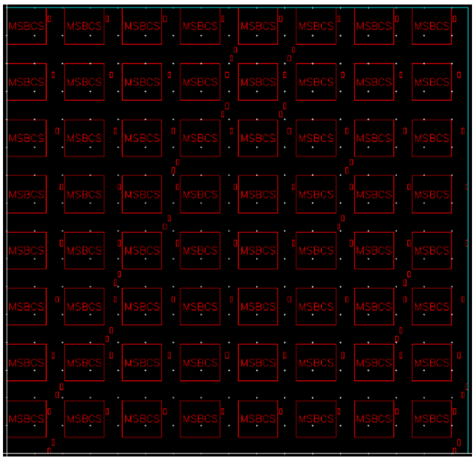
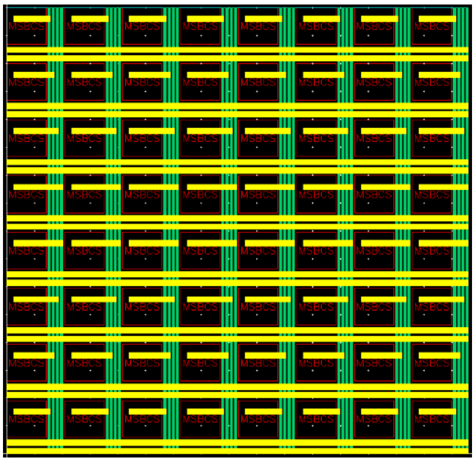
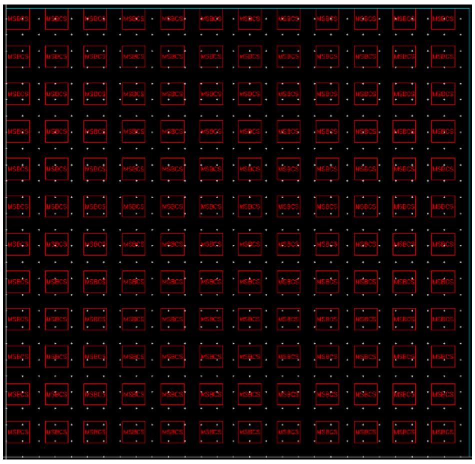
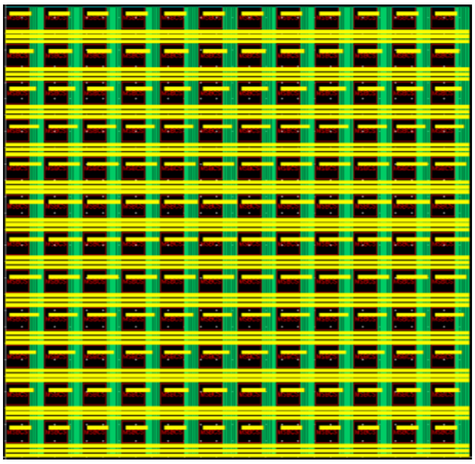
The left one is sequentially going up, and the right one is alternatively going up. Noticed that both of them are symmetric on the y-axis. Here I chose the left one because it was easy for me to implement. While implementing it, I also use the property of symmetric to decide the coordinate of the vias.

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

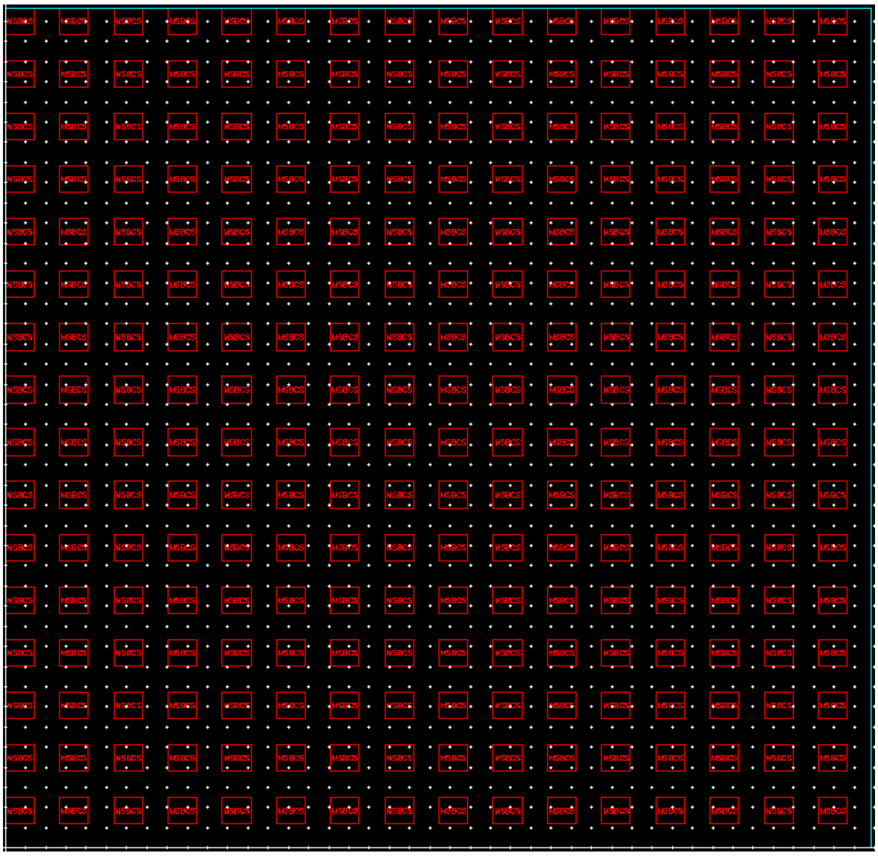
- By Python program:

4 Current Sources	
without metal layer	with metal layer
	

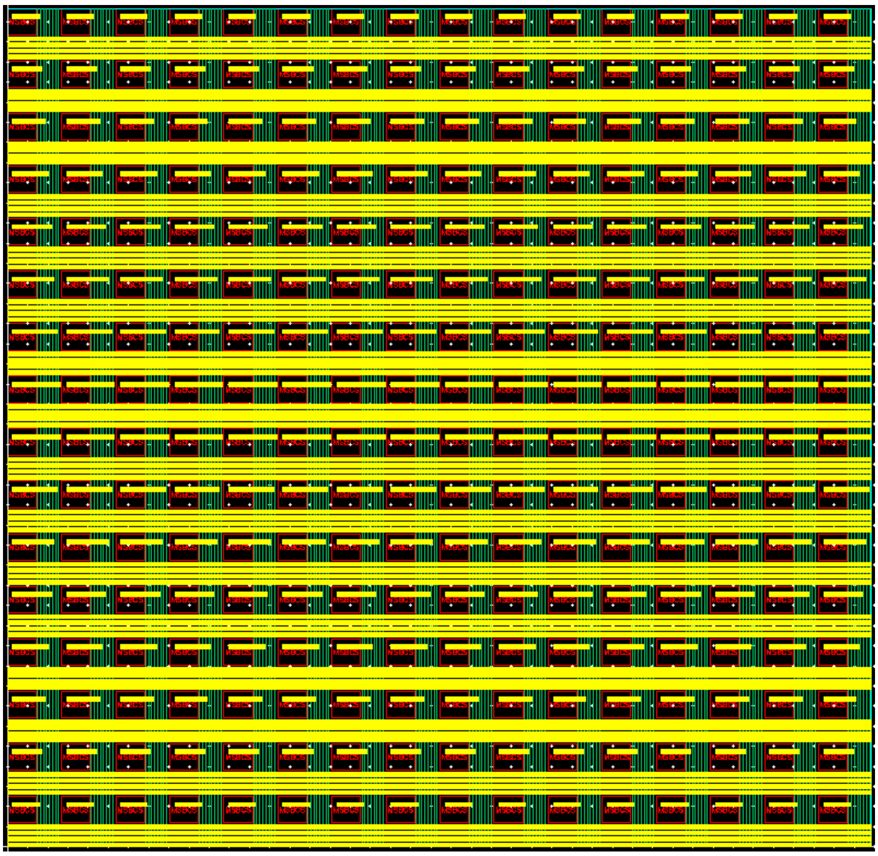
- By C/C++ program:

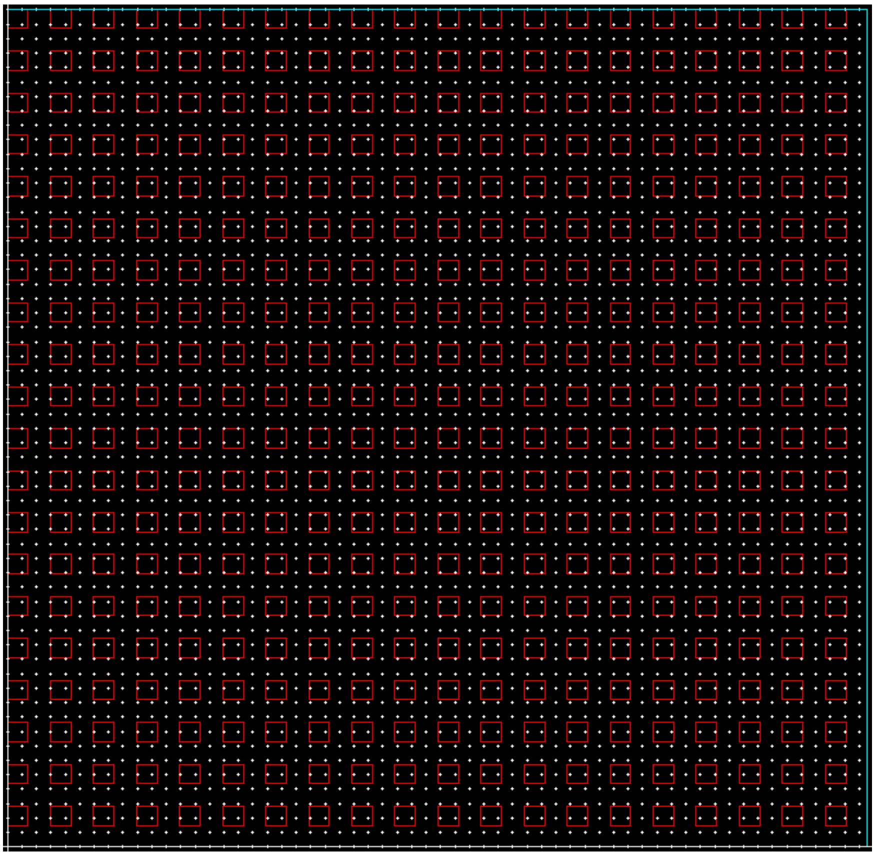
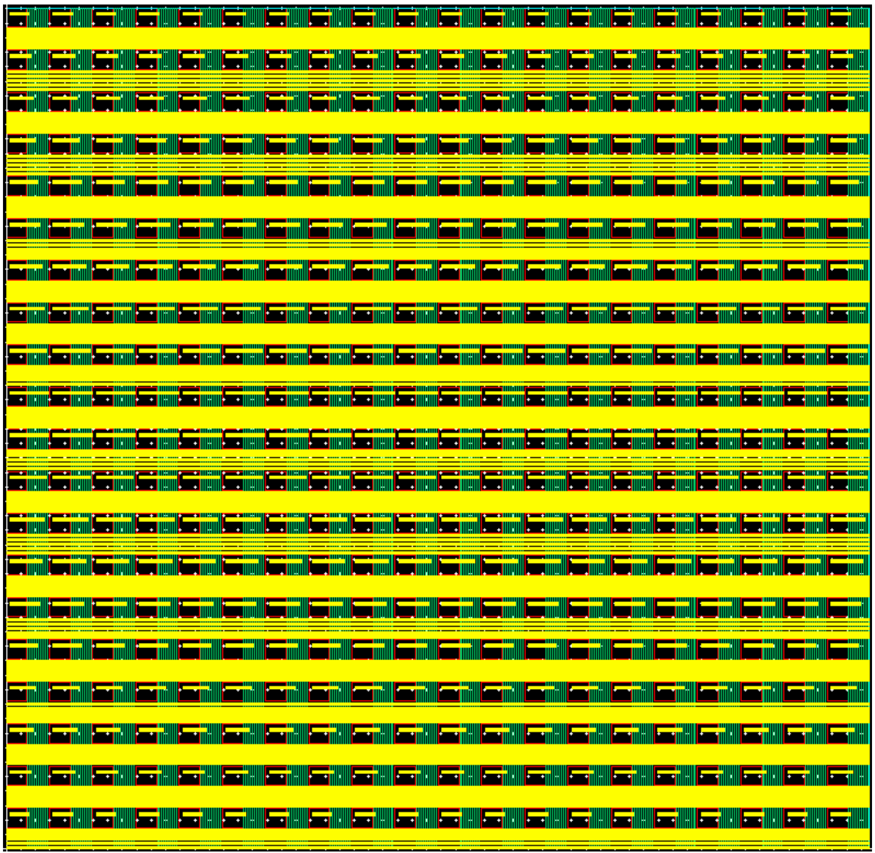
<table>
<tr><th colspan="2">4 Current Sources</th></tr>
<tr><td>without metal layer</td><td>with metal layer</td></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">16 Current Sources</th></tr>
<tr><td>without metal layer</td><td>with metal layer</td></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">36 Current Sources</th></tr>
<tr><td>without metal layer</td><td>with metal layer</td></tr>
<tr><td></td><td></td></tr>
</table>

64 Current Sources without metal layer



64 Current Sources with metal layer



100 Current Sources without metal layer

100 Current Sources with metal layer


4) The screenshots for the result of HW5_grading.sh:

```
[dhlin22@ic51 HW5_grading]$ bash HW5_grading.sh
|-----------------------------------------------|
|                                               |
|    This script is used for PDA HW5 grading.   |
|                                               |
|-----------------------------------------------|
grading on 111062684:
  testcase |      result | status
    python |        pass | success
         4 |        pass | success
        16 |        pass | success
        36 |        pass | success
        64 |        pass | success
       100 |        pass | success
|-----------------------------------------------------|
|                                                     |
|    Successfully generate grades to HW5_grade.csv    |
|                                                     |
|-----------------------------------------------------|
```