Homework 5: Automated P&R for Analog Circuits

1. 111062697 吳律穎
2. --HOW to Compile

   In src/, enter the following command:

   $ make

   It will generate the executable file "hw5" in "../bin/".

   If you want to remove it, please enter the following command:

   $ make clean

   --How to run

   In src/ directory, enter the following command:

   Usage: ../bin/[exe] [number of current sources] [def file path]

   e.g.

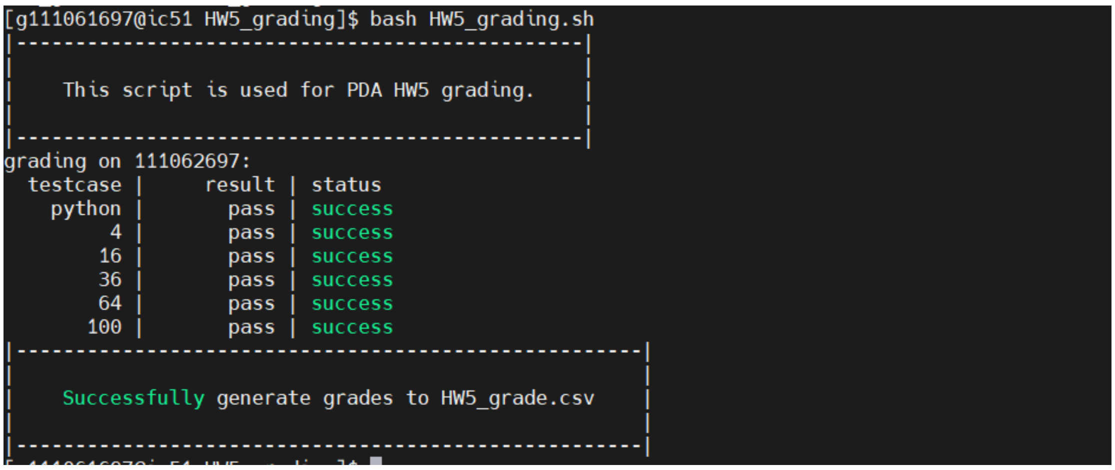
   ../bin/hw5 4 ../output/CS_4.def

   In "HW5/bin/", enter the following command:

   Usage: ./[exe] [number of current sources] [def file path]

   e.g.

   ./hw5 4 ../output/CS_4.def



```
[g111061697@ic51 HW5_grading]$ bash HW5_grading.sh
|-----------------------------------------------|
|                                               |
|    This script is used for PDA HW5 grading.   |
|                                               |
|-----------------------------------------------|
grading on 111062697:
  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    |
|                                                     |
|-----------------------------------------------------|
```

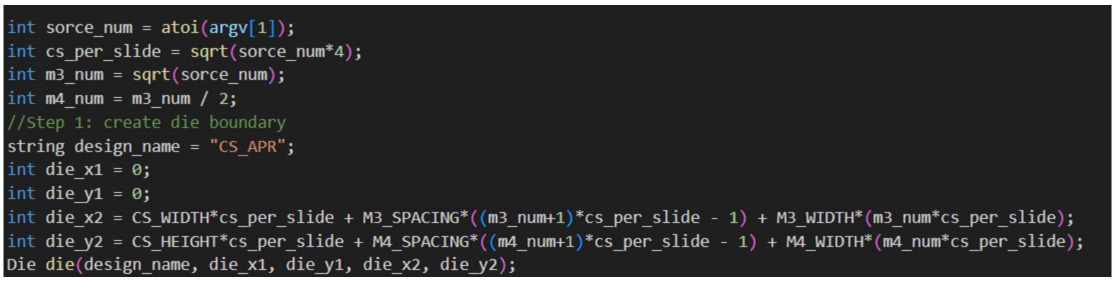
3.

Step1:

每行有 n3_num = source^(1/2)個 M3，每列有 1/2*M3 個 M4

每邊有 cs_per_slide = (source*4)^(1/2)個 CS

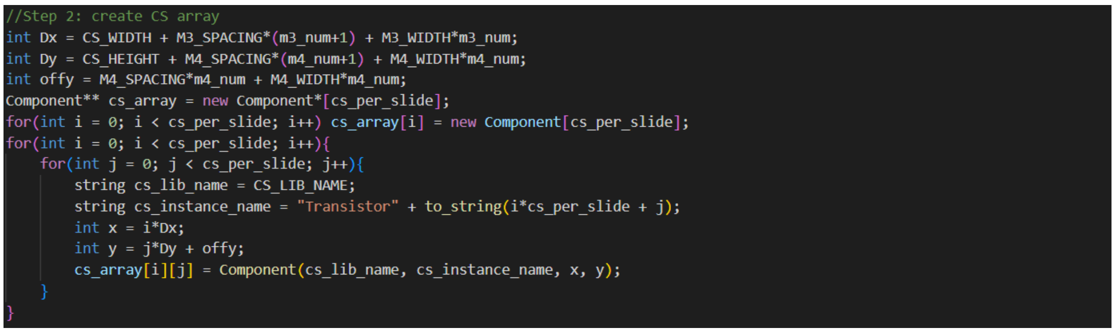
建立 DIE 的範圍, CS_WIDTH + SPCEING + M3/4



```
int sorce_num = atoi(argv[1]);
int cs_per_slide = sqrt(sorce_num*4);
int m3_num = sqrt(sorce_num);
int m4_num = m3_num / 2;
//Step 1: create die boundary
string design_name = "CS_APR";
int die_x1 = 0;
int die_y1 = 0;
int die_x2 = CS_WIDTH*cs_per_slide + M3_SPACING*((m3_num+1)*cs_per_slide - 1) + M3_WIDTH*(m3_num*cs_per_slide);
int die_y2 = CS_HEIGHT*cs_per_slide + M4_SPACING*((m4_num+1)*cs_per_slide - 1) + M4_WIDTH*(m4_num*cs_per_slide);
Die die(design_name, die_x1, die_y1, die_x2, die_y2);
```

Step2:

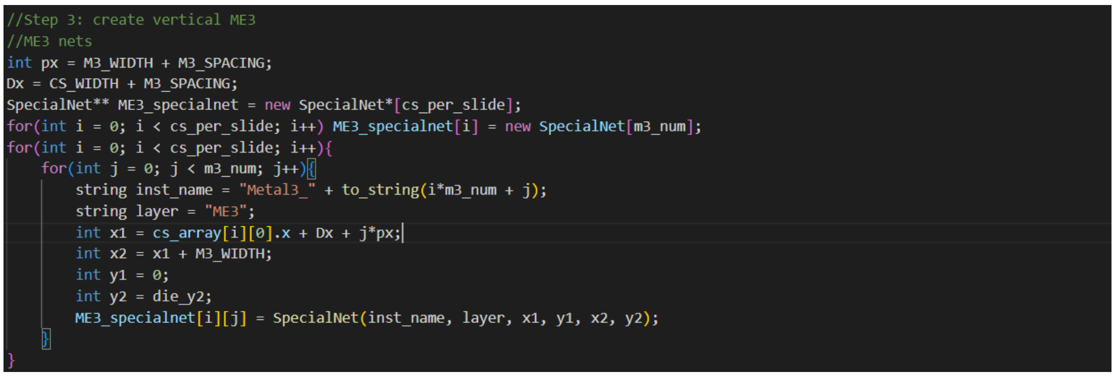
建立 CS 的 array，共((source*4)^(1/2))* ((source*4)^(1/2))個 CS



```
//Step 2: create CS array
int Dx = CS_WIDTH + M3_SPACING*(m3_num+1) + M3_WIDTH*m3_num;
int Dy = CS_HEIGHT + M4_SPACING*(m4_num+1) + M4_WIDTH*m4_num;
int offy = M4_SPACING*m4_num + M4_WIDTH*m4_num;
Component** cs_array = new Component*[cs_per_slide];
for(int i = 0; i < cs_per_slide; i++) cs_array[i] = new Component[cs_per_slide];
for(int i = 0; i < cs_per_slide; i++){
    for(int j = 0; j < cs_per_slide; j++){
        string cs_lib_name = CS_LIB_NAME;
        string cs_instance_name = "Transistor" + to_string(i*cs_per_slide + j);
        int x = i*Dx;
        int y = j*Dy + offy;
        cs_array[i][j] = Component(cs_lib_name, cs_instance_name, x, y);
    }
}
```

Step3:

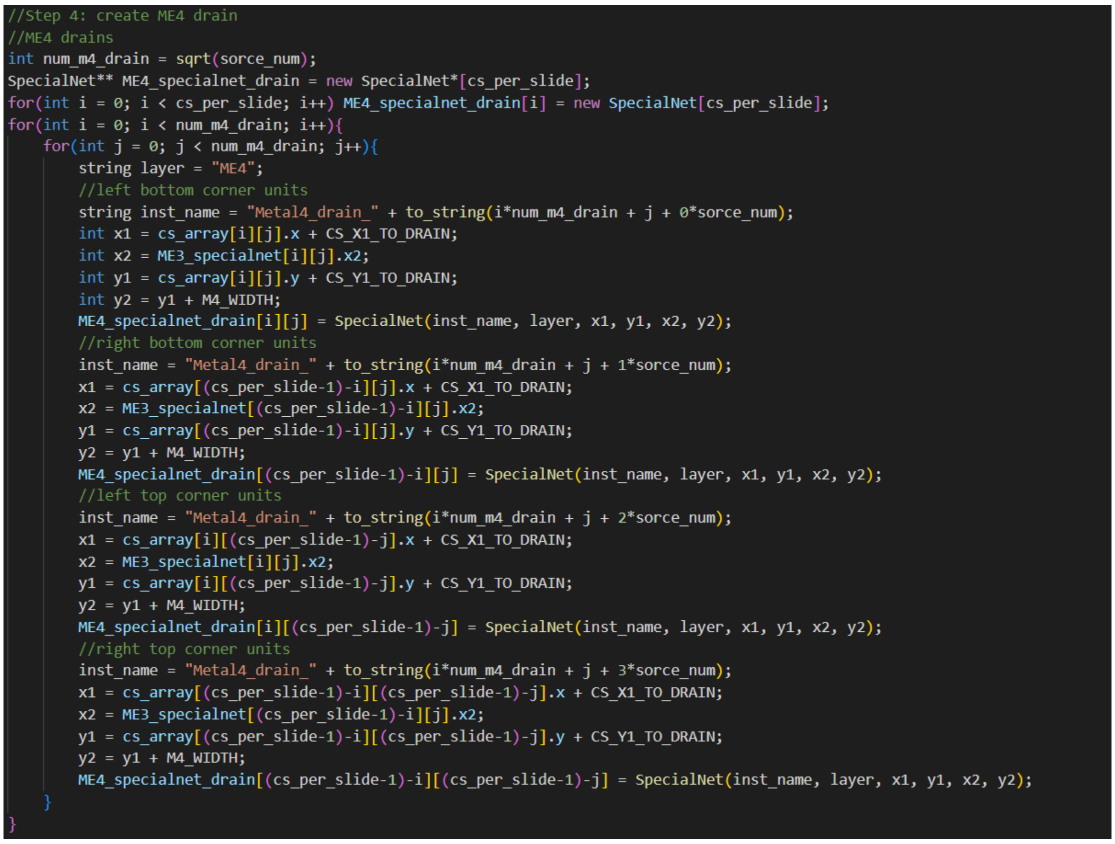
建立 ME3，每行 n3_num 個 M3，共 cs_per_slide 個行



```
//Step 3: create vertical ME3
//ME3 nets
int px = M3_WIDTH + M3_SPACING;
Dx = CS_WIDTH + M3_SPACING;
SpecialNet** ME3_specialnet = new SpecialNet*[cs_per_slide];
for(int i = 0; i < cs_per_slide; i++) ME3_specialnet[i] = new SpecialNet[m3_num];
for(int i = 0; i < cs_per_slide; i++){
    for(int j = 0; j < m3_num; j++){
        string inst_name = "Metal3_" + to_string(i*m3_num + j);
        string layer = "ME3";
        int x1 = cs_array[i][0].x + Dx + j*px;
        int x2 = x1 + M3_WIDTH;
        int y1 = 0;
        int y2 = die_y2;
        ME3_specialnet[i][j] = SpecialNet(inst_name, layer, x1, y1, x2, y2);
    }
}
```

Step4:

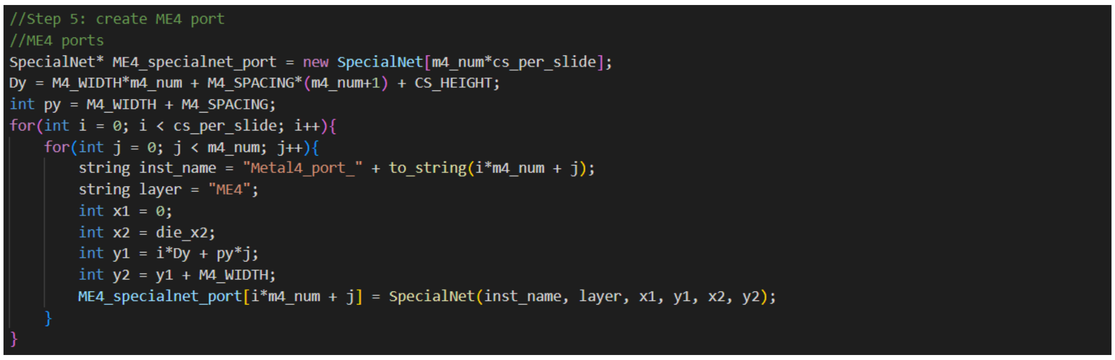
建立 ME4 drain，分成左上左下右上右下四個區塊計算其座標，每塊有 CS_per_slide*CS_per_slide 個要計算



```
//Step 4: create ME4 drain
//ME4 drains
int num_m4_drain = sqrt(sorce_num);
SpecialNet** ME4_specialnet_drain = new SpecialNet*[cs_per_slide];
for(int i = 0; i < cs_per_slide; i++) ME4_specialnet_drain[i] = new SpecialNet[cs_per_slide];
for(int i = 0; i < num_m4_drain; i++){
    for(int j = 0; j < num_m4_drain; j++){
        string layer = "ME4";
        //left bottom corner units
        string inst_name = "Metal4_drain_" + to_string(i*num_m4_drain + j + 0*sorce_num);
        int x1 = cs_array[i][j].x + CS_X1_TO_DRAIN;
        int x2 = ME3_specialnet[i][j].x2;
        int y1 = cs_array[i][j].y + CS_Y1_TO_DRAIN;
        int y2 = y1 + M4_WIDTH;
        ME4_specialnet_drain[i][j] = SpecialNet(inst_name, layer, x1, y1, x2, y2);
        //right bottom corner units
        inst_name = "Metal4_drain_" + to_string(i*num_m4_drain + j + 1*sorce_num);
        x1 = cs_array[(cs_per_slide-1)-i][j].x + CS_X1_TO_DRAIN;
        x2 = ME3_specialnet[(cs_per_slide-1)-i][j].x2;
        y1 = cs_array[(cs_per_slide-1)-i][j].y + CS_Y1_TO_DRAIN;
        y2 = y1 + M4_WIDTH;
        ME4_specialnet_drain[(cs_per_slide-1)-i][j] = SpecialNet(inst_name, layer, x1, y1, x2, y2);
        //left top corner units
        inst_name = "Metal4_drain_" + to_string(i*num_m4_drain + j + 2*sorce_num);
        x1 = cs_array[i][(cs_per_slide-1)-j].x + CS_X1_TO_DRAIN;
        x2 = ME3_specialnet[i][j].x2;
        y1 = cs_array[i][(cs_per_slide-1)-j].y + CS_Y1_TO_DRAIN;
        y2 = y1 + M4_WIDTH;
        ME4_specialnet_drain[i][(cs_per_slide-1)-j] = SpecialNet(inst_name, layer, x1, y1, x2, y2);
        //right top corner units
        inst_name = "Metal4_drain_" + to_string(i*num_m4_drain + j + 3*sorce_num);
        x1 = cs_array[(cs_per_slide-1)-i][(cs_per_slide-1)-j].x + CS_X1_TO_DRAIN;
        x2 = ME3_specialnet[(cs_per_slide-1)-i][j].x2;
        y1 = cs_array[(cs_per_slide-1)-i][(cs_per_slide-1)-j].y + CS_Y1_TO_DRAIN;
        y2 = y1 + M4_WIDTH;
        ME4_specialnet_drain[(cs_per_slide-1)-i][(cs_per_slide-1)-j] = SpecialNet(inst_name, layer, x1, y1, x2, y2);
    }
}
```

Step5:

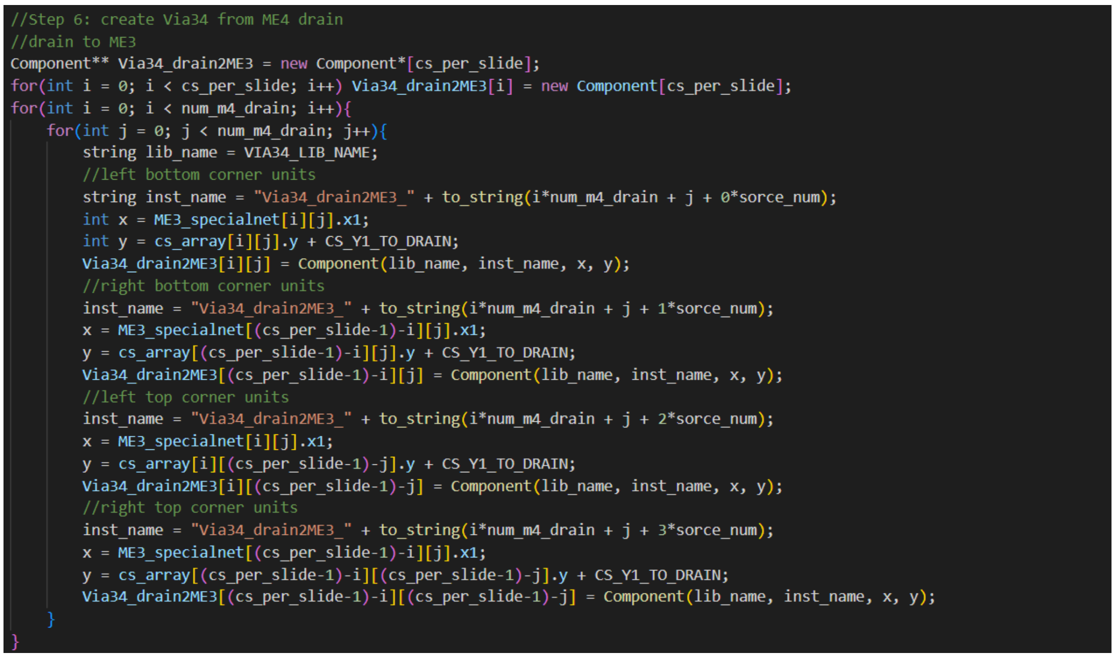
建立 ME4 port，每列 m4_num 個 ME4，共 CS_per_slide 列



```
//Step 5: create ME4 port
//ME4 ports
SpecialNet* ME4_specialnet_port = new SpecialNet[m4_num*cs_per_slide];
Dy = M4_WIDTH*m4_num + M4_SPACING*(m4_num+1) + CS_HEIGHT;
int py = M4_WIDTH + M4_SPACING;
for(int i = 0; i < cs_per_slide; i++){
    for(int j = 0; j < m4_num; j++){
        string inst_name = "Metal4_port_" + to_string(i*m4_num + j);
        string layer = "ME4";
        int x1 = 0;
        int x2 = die_x2;
        int y1 = i*Dy + py*j;
        int y2 = y1 + M4_WIDTH;
        ME4_specialnet_port[i*m4_num + j] = SpecialNet(inst_name, layer, x1, y1, x2, y2);
    }
}
```

Step6:

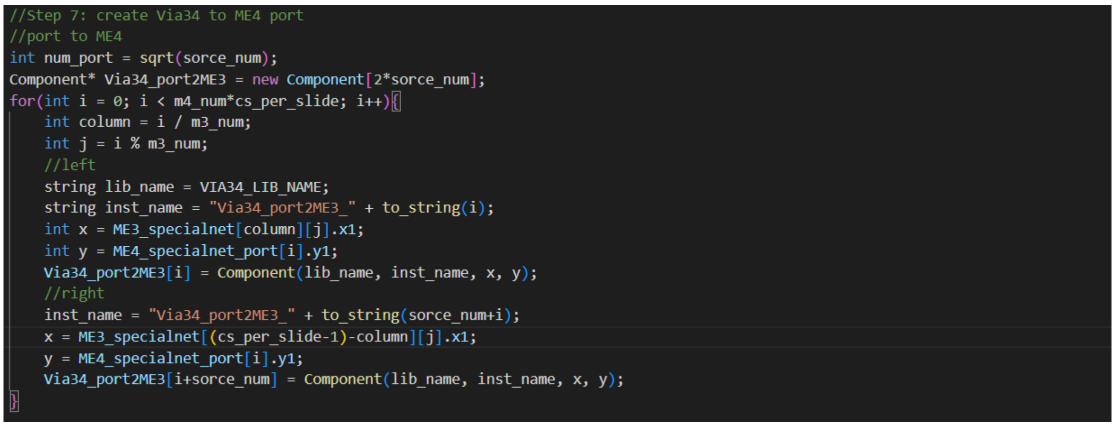
建立 ME4 drain 到 ME3 的 via，左下由下而上由左而右，其他跟他 mirror



```
//Step 6: create Via34 from ME4 drain
//drain to ME3
Component** Via34_drain2ME3 = new Component*[cs_per_slide];
for(int i = 0; i < cs_per_slide; i++) Via34_drain2ME3[i] = new Component[cs_per_slide];
for(int i = 0; i < num_m4_drain; i++){
    for(int j = 0; j < num_m4_drain; j++){
        string lib_name = VIA34_LIB_NAME;
        //left bottom corner units
        string inst_name = "Via34_drain2ME3_" + to_string(i*num_m4_drain + j + 0*sorce_num);
        int x = ME3_specialnet[i][j].x1;
        int y = cs_array[i][j].y + CS_Y1_TO_DRAIN;
        Via34_drain2ME3[i][j] = Component(lib_name, inst_name, x, y);
        //right bottom corner units
        inst_name = "Via34_drain2ME3_" + to_string(i*num_m4_drain + j + 1*sorce_num);
        x = ME3_specialnet[(cs_per_slide-1)-i][j].x1;
        y = cs_array[(cs_per_slide-1)-i][j].y + CS_Y1_TO_DRAIN;
        Via34_drain2ME3[(cs_per_slide-1)-i][j] = Component(lib_name, inst_name, x, y);
        //left top corner units
        inst_name = "Via34_drain2ME3_" + to_string(i*num_m4_drain + j + 2*sorce_num);
        x = ME3_specialnet[i][j].x1;
        y = cs_array[i][(cs_per_slide-1)-j].y + CS_Y1_TO_DRAIN;
        Via34_drain2ME3[i][(cs_per_slide-1)-j] = Component(lib_name, inst_name, x, y);
        //right top corner units
        inst_name = "Via34_drain2ME3_" + to_string(i*num_m4_drain + j + 3*sorce_num);
        x = ME3_specialnet[(cs_per_slide-1)-i][j].x1;
        y = cs_array[(cs_per_slide-1)-i][(cs_per_slide-1)-j].y + CS_Y1_TO_DRAIN;
        Via34_drain2ME3[(cs_per_slide-1)-i][(cs_per_slide-1)-j] = Component(lib_name, inst_name, x, y);
    }
}
```

Strp7:

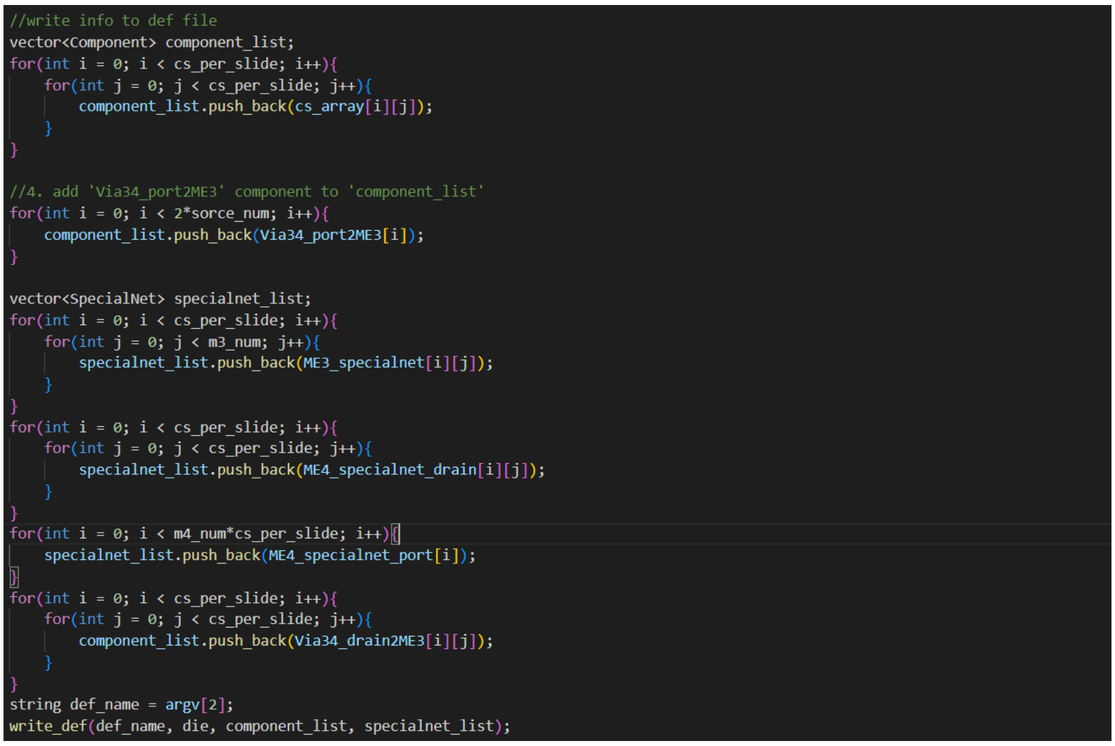
建立 ME3 到 ME4 port 的 via，分為左右兩塊，由下而上由外而內去計算



```
//Step 7: create Via34 to ME4 port
//port to ME4
int num_port = sqrt(sorce_num);
Component* Via34_port2ME3 = new Component[2*sorce_num];
for(int i = 0; i < m4_num*cs_per_slide; i++){
    int column = i / m3_num;
    int j = i % m3_num;
    //left
    string lib_name = VIA34_LIB_NAME;
    string inst_name = "Via34_port2ME3_" + to_string(i);
    int x = ME3_specialnet[column][j].x1;
    int y = ME4_specialnet_port[i].y1;
    Via34_port2ME3[i] = Component(lib_name, inst_name, x, y);
    //right
    inst_name = "Via34_port2ME3_" + to_string(sorce_num+i);
    x = ME3_specialnet[(cs_per_slide-1)-column][j].x1;
    y = ME4_specialnet_port[i].y1;
    Via34_port2ME3[i+sorce_num] = Component(lib_name, inst_name, x, y);
}
```

Step8:

寫入 DEF 檔



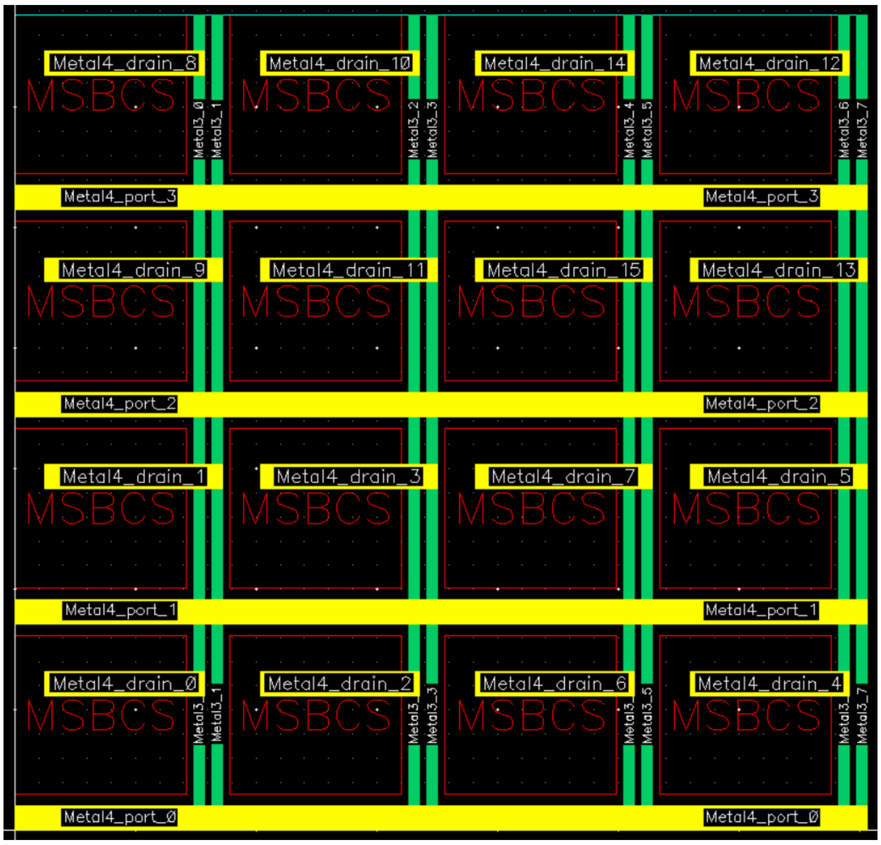
```
//write info to def file
vector<Component> component_list;
for(int i = 0; i < cs_per_slide; i++){
    for(int j = 0; j < cs_per_slide; j++){
        component_list.push_back(cs_array[i][j]);
    }
}

//4. add 'Via34_port2ME3' component to 'component_list'
for(int i = 0; i < 2*sorce_num; i++){
    component_list.push_back(Via34_port2ME3[i]);
}

vector<SpecialNet> specialnet_list;
for(int i = 0; i < cs_per_slide; i++){
    for(int j = 0; j < m3_num; j++){
        specialnet_list.push_back(ME3_specialnet[i][j]);
    }
}
for(int i = 0; i < cs_per_slide; i++){
    for(int j = 0; j < cs_per_slide; j++){
        specialnet_list.push_back(ME4_specialnet_drain[i][j]);
    }
}
for(int i = 0; i < m4_num*cs_per_slide; i++){
    specialnet_list.push_back(ME4_specialnet_port[i]);
}
for(int i = 0; i < cs_per_slide; i++){
    for(int j = 0; j < cs_per_slide; j++){
        component_list.push_back(Via34_drain2ME3[i][j]);
    }
}
string def_name = argv[2];
write_def(def_name, die, component_list, specialnet_list);
```

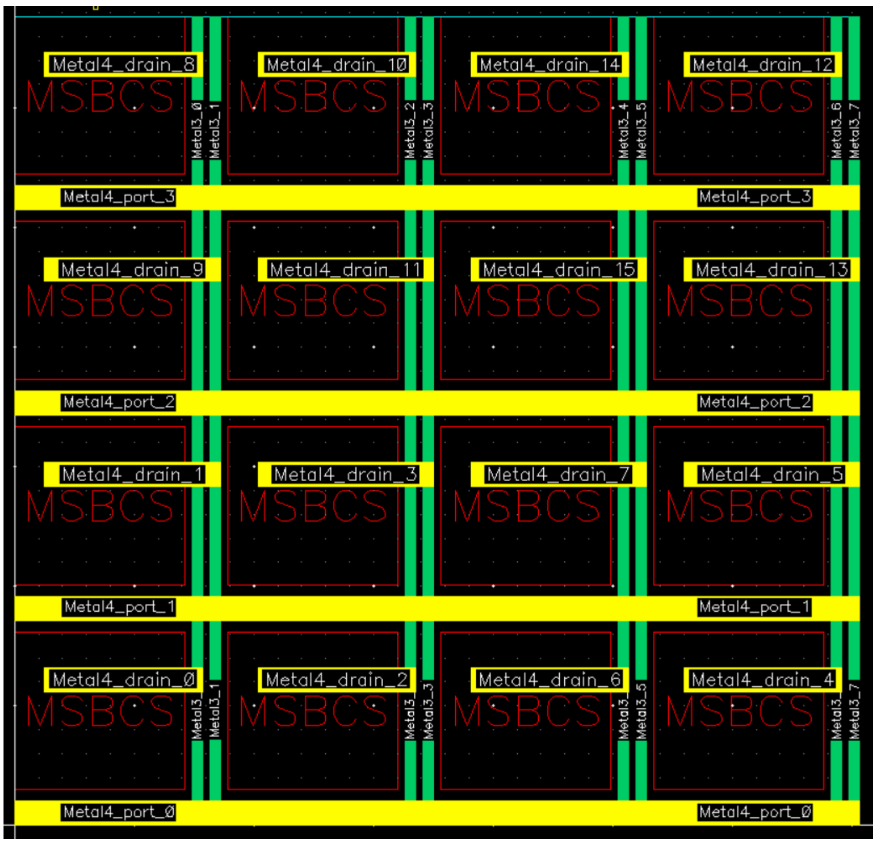
4.

Python:



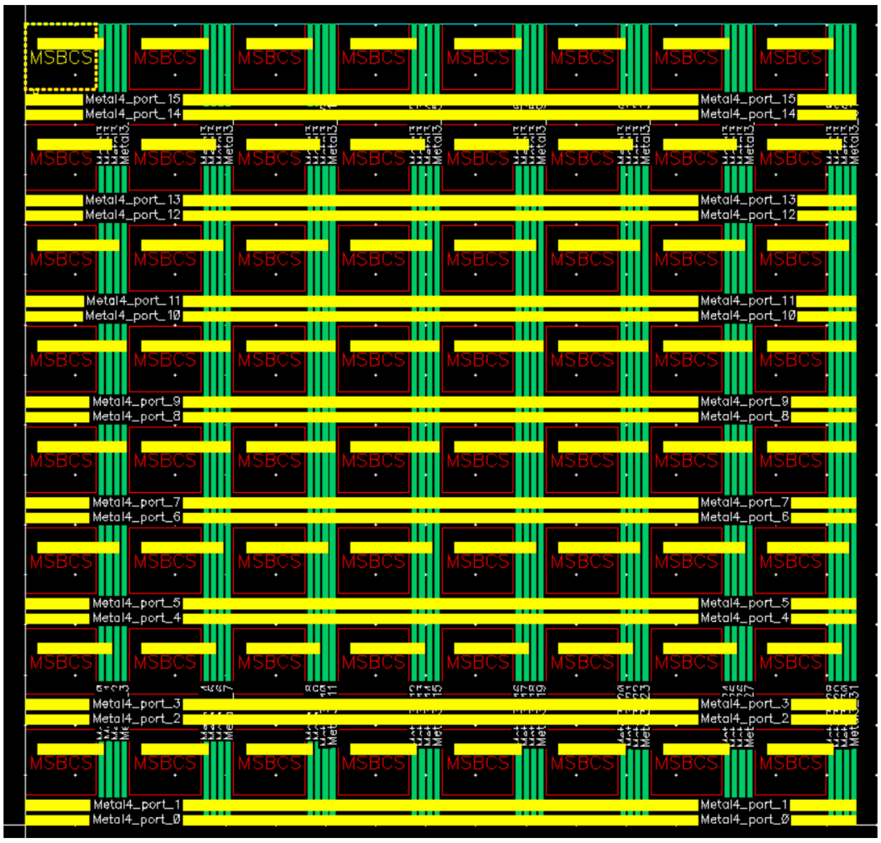
C++ :

4 :



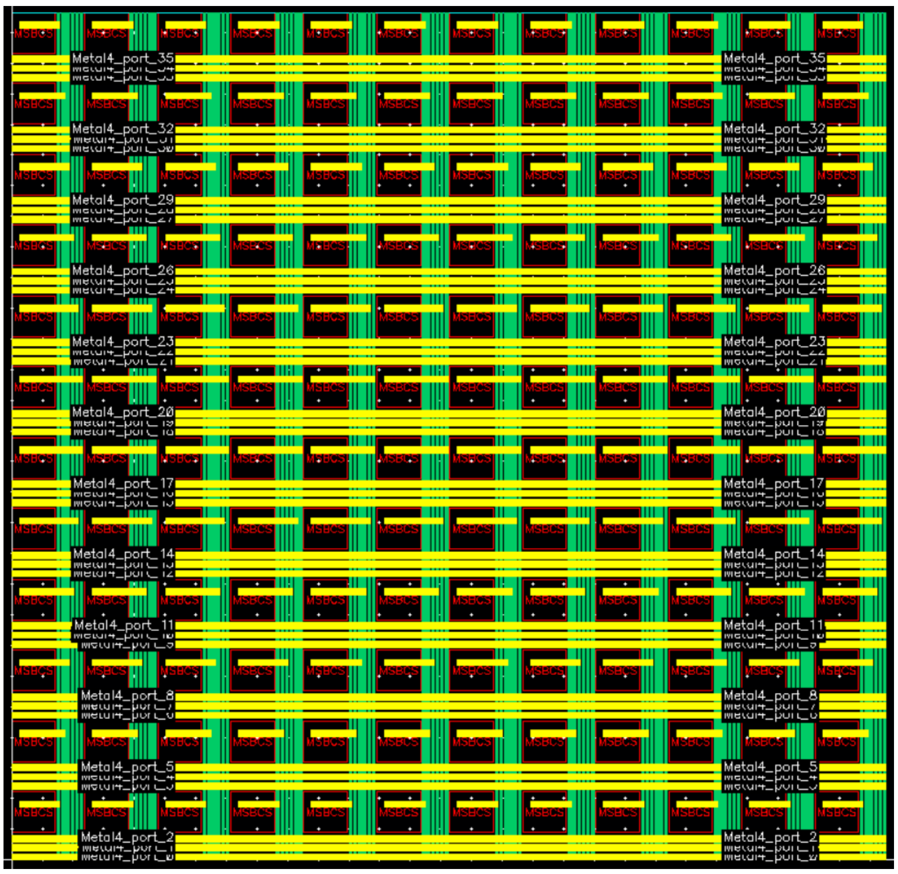
Metal4_drain_8
Metal4_drain_10
Metal4_drain_14
Metal4_drain_12
MSBCS
Metal3_0
Metal3_1
Metal3_2
Metal3_3
Metal3_4
Metal3_5
Metal3_6
Metal3_7
Metal4_port_3
Metal4_drain_9
Metal4_drain_11
Metal4_drain_15
Metal4_drain_13
Metal4_port_2
Metal4_drain_1
Metal4_drain_3
Metal4_drain_7
Metal4_drain_5
Metal4_port_1
Metal4_drain_0
Metal4_drain_2
Metal4_drain_6
Metal4_drain_4
Metal4_port_0

16 :

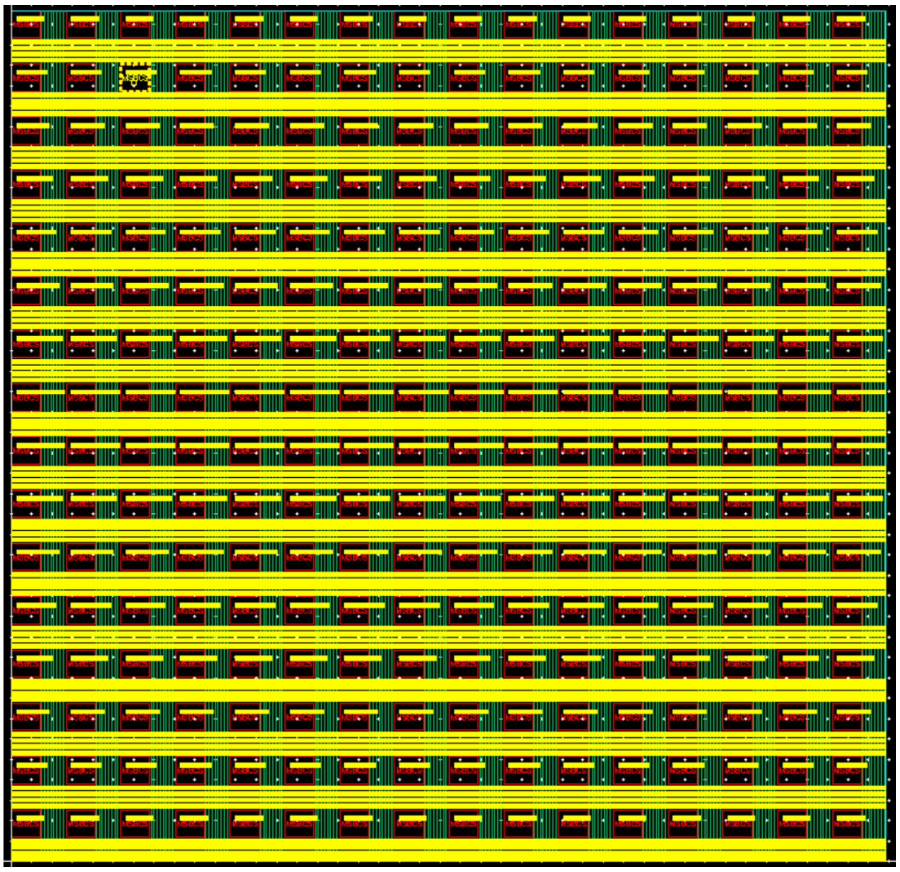


MSBCS
Metal4_port_15
Metal4_port_14
Metal4_port_13
Metal4_port_12
Metal4_port_11
Metal4_port_10
Metal4_port_9
Metal4_port_8
Metal4_port_7
Metal4_port_6
Metal4_port_5
Metal4_port_4
Metal4_port_3
Metal4_port_2
Metal4_port_1
Metal4_port_0

36 :


MSBCS
Metal4_port_35
Metal4_port_32
Metal4_port_29
Metal4_port_26
Metal4_port_23
Metal4_port_20
Metal4_port_17
Metal4_port_14
Metal4_port_11
Metal4_port_8
Metal4_port_5
Metal4_port_2

64 :



100 :