# 類比電路佈局合成自動化
# Automatic Layout Synthesis for Analog Circuits

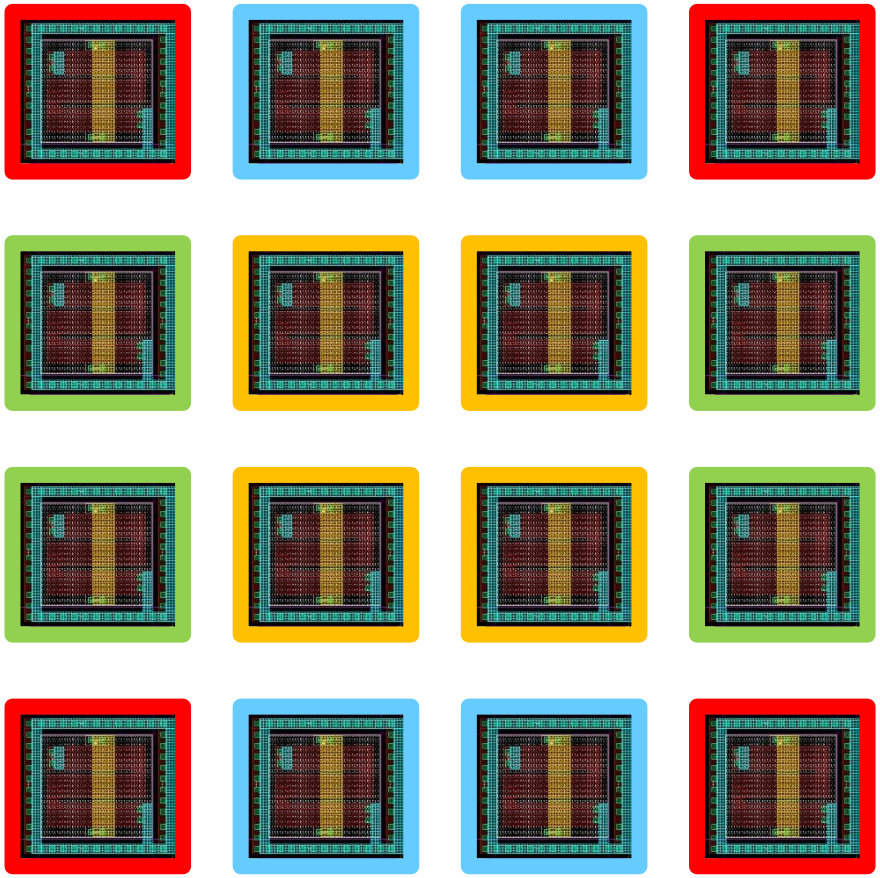
## 單元二
## 考慮一階系統製程變異的矩陣元件佈局自動化

**Lecturer: Shao-Yun Fang**
**The Electronic Design Automation Laboratory**
**Department of Electrical Engineering**
**National Taiwan University of Science and Technology**
**Taipei 106, Taiwan**

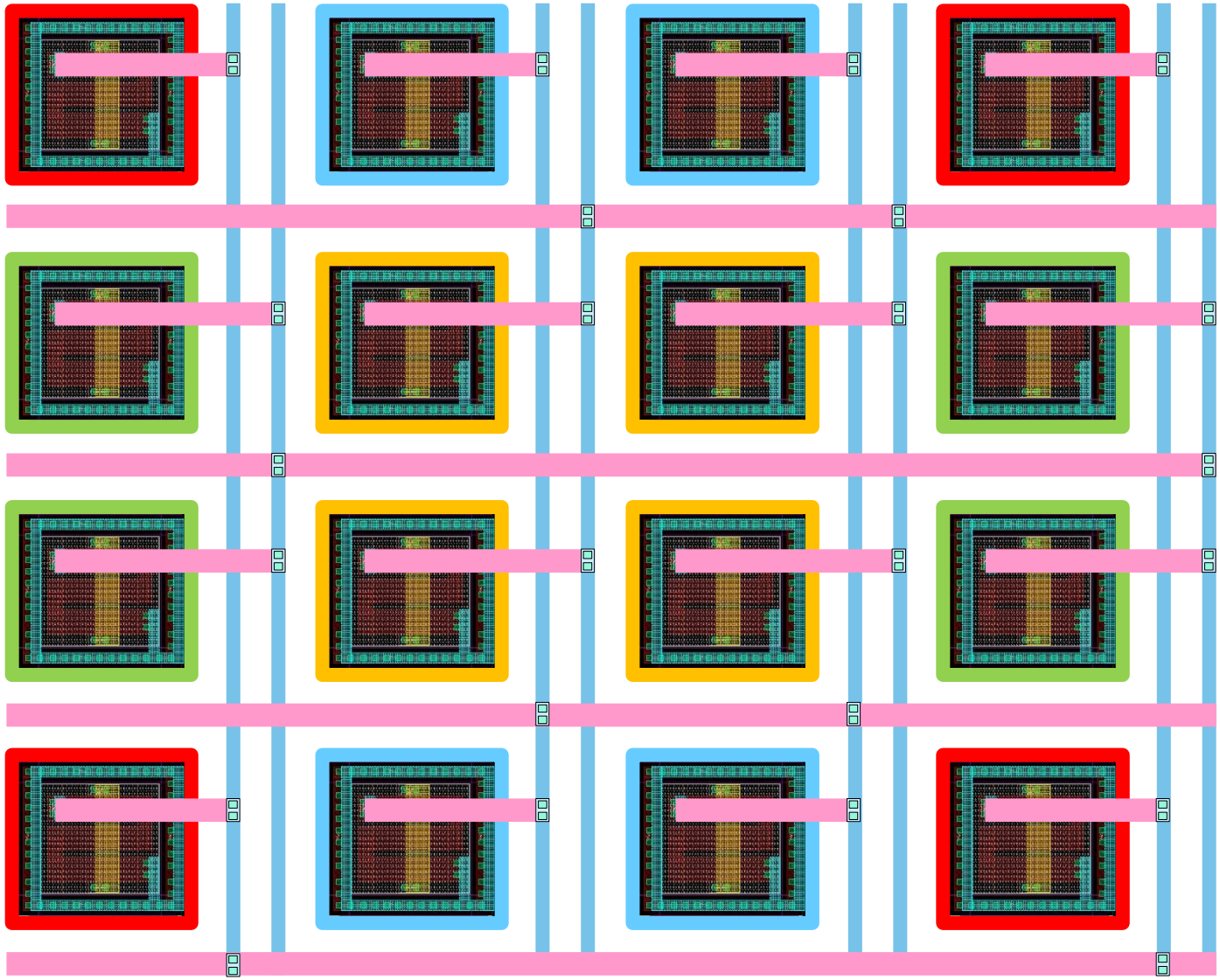
THE EDA LAB TAIWAN TECH

# Placement

- **Current source array placement**
  - 4 current sources
  - 16 units
  - Common centroid

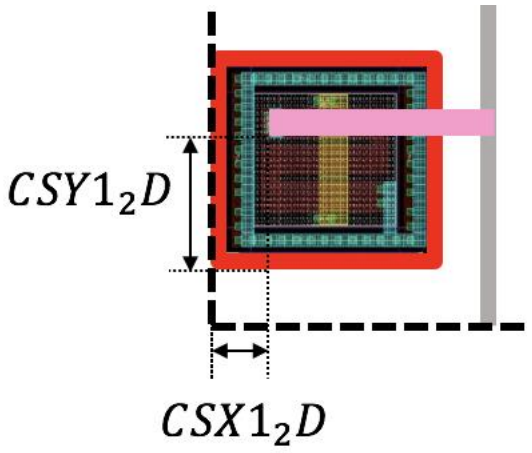


# Routing



# Global Parameters

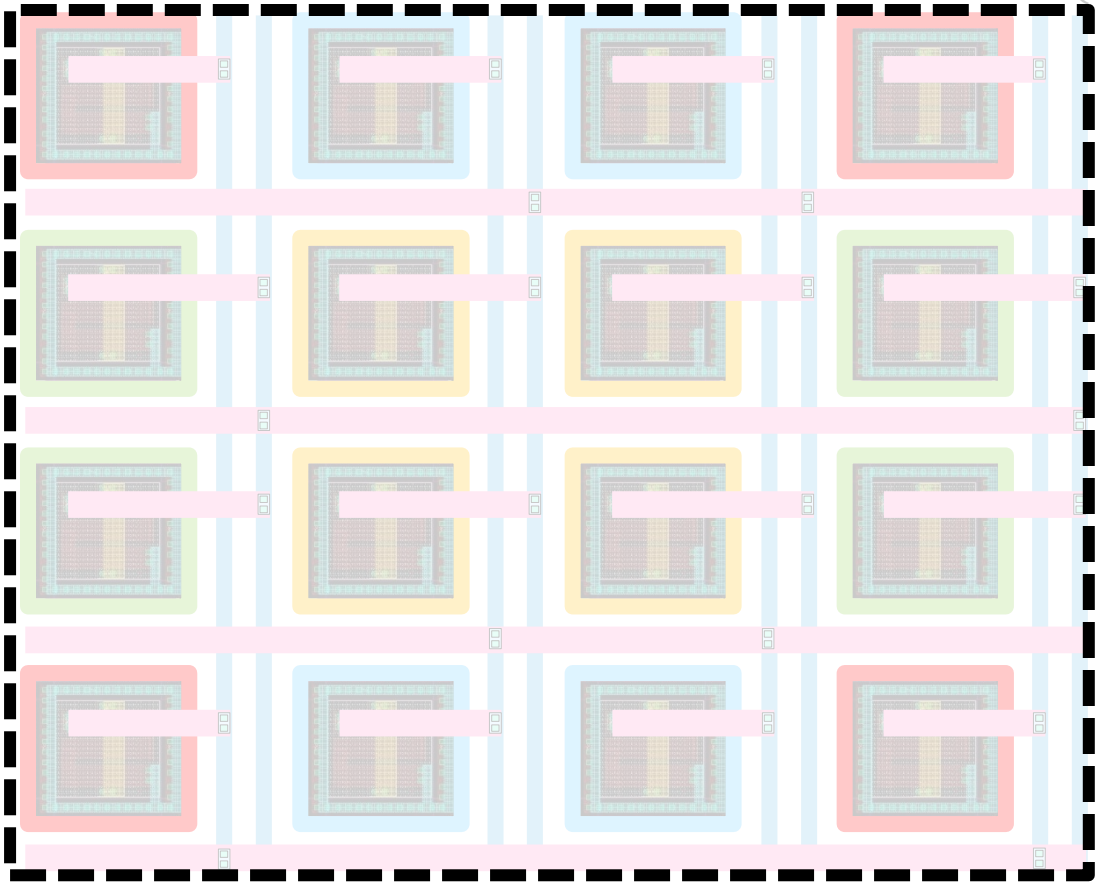
```
#!/usr/bin/python3
# import required classes and functions
from myObject import Die, Component, SpecialNet
from write_def import write_def

# define global parameters
CS_WIDTH = 7100
CS_HEIGHT = 6600
M3_WIDTH = 440
M3_SPACING = 310
M4_WIDTH = 1000
M4_SPACING = 490

CS_X1_TO_DRAIN = 1260
CS_Y1_TO_DRAIN = 4100
CS_LIB_NAME = 'MSBCS'
VIA34_LIB_NAME = 'Via34'
```



# Step 1: Create Die Boundary

- **Preserve required routing resource**

M3 M4 Via34




M3 width 0.44

M3 spacing 0.31

M4 width 1

M4 spacing 0.49

Cell width 7.1

Cell height 6.6

$$bx_2 = 7.1 \times 4 + 0.31 \times (3 \times 4 - 1) + 0.44 \times 2 \times 4 = 35.33$$

$$by_2 = 6.6 \times 4 + 0.49 \times (2 \times 4 - 1) + 1 \times 1 \times 4 = 33.83$$

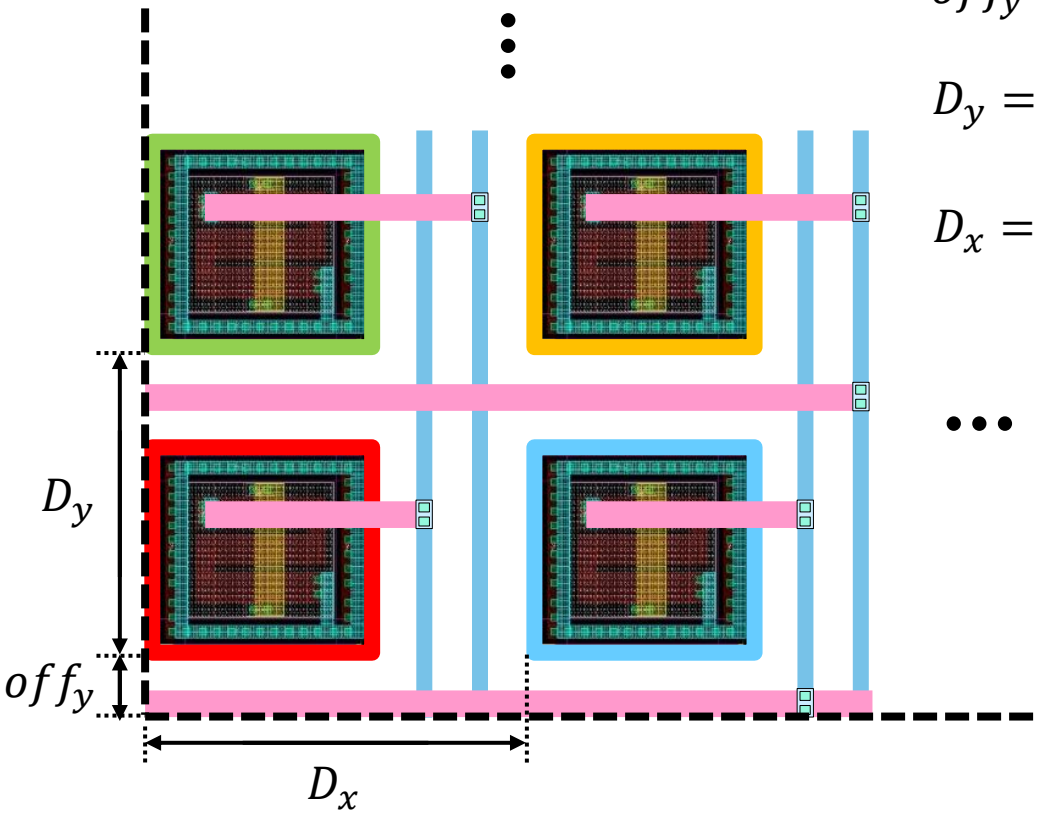
# Step 1: Create Die Boundary (cont'd)

- **Define lower-left and upper-right coordinates of die**

```
##### Step 1: create die boundary #####
design_name = 'CS_APR'
die_x1 = 0
die_y1 = 0
die_x2 =
die_y2 =
die = Die(design_name, die_x1, die_y1, die_x2, die_y2)
```

# Step 2: Create CS Placement

- **Place each current source unit at the corresponding (the left-bottom corner) coordinates**



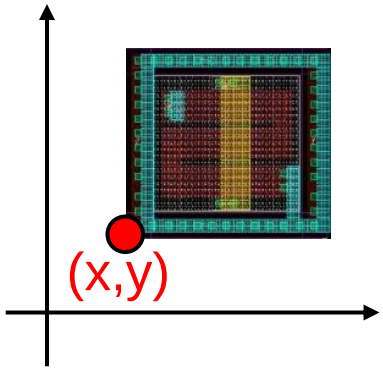

$$off_y = M4_{spacing} + M4_{width}$$

$$D_y = CS_{height} + M4_{spacing} \times 2 + M4_{width}$$

$$D_x = CS_{width} + M3_{spacing} \times 3 + M3_{width} \times 2$$

$$X(cs_{ij}) = i \times D_x$$

$$Y(cs_{ij}) = j \times D_y + off_y$$



# Step 2: Create CS Placement (cont'd)

- **Use two for loops to place all CS units at their corresponding positions**

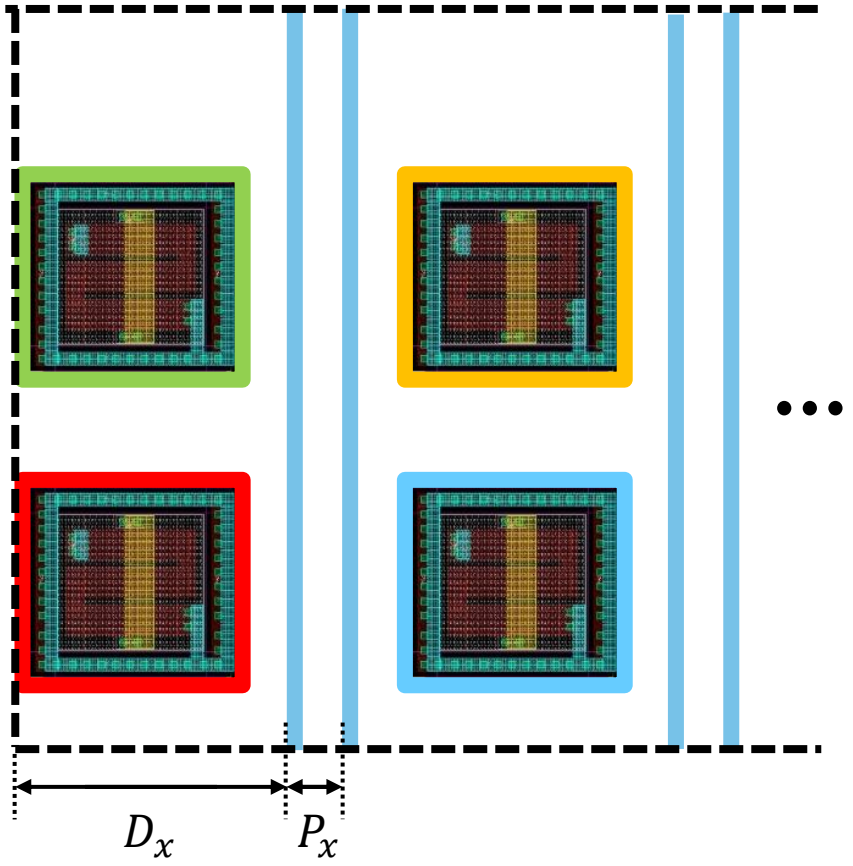
```
##### Step 2: create CS array #####
cs_array = [[Component for j in range(4)] for i in range(4)]
for i in range(4):
    for j in range(4):
        cs_lib_name = CS_LIB_NAME
        cs_instance_name = 'Transistor' + str(i * 4 + j)
        x =
        y =
        cs_array[i][j] = Component(cs_lib_name, cs_instance_name, x, y)
```

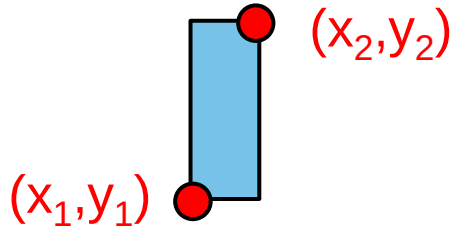
- **Append CS instances to component list**

```
# write info to def file
component_list = []
for i in range(4):
    for j in range(4):
        component_list.append(cs_array[i][j])
```

# Step 3: Create Vertical ME3

- **Create vertical ME3 metals**




$$x_1(M3_{00}) = x_1(CS_{0*}) + D_x + 0 \times P_x$$
$$x_2(M3_{00}) = x_1(M3_{00}) + M3_{width}$$
$$x_1(M3_{01}) = x_1(CS_{0*}) + D_x + 1 \times P_x$$
$$x_2(M3_{01}) = x_1(M3_{01}) + M3_{width}$$

$$x_1(M3_{ij}) = x_1(CS_{i*}) + D_x + j \times P_x$$
$$x_2(M3_{ij}) = x_1(M3_{ij}) + M3_{width}$$
$$y_1(M3_{ij}) = 0$$
$$y_2(M3_{ij}) = by_2$$

# Step 3: Create Vertical ME3 (cont'd)

- **Create vertical ME3 metals**

```
##### Step 3: create vertical ME3 #####
# ME3 nets
ME3_specialnet = [[SpecialNet for j in range(2)] for i in range(4)]
for i in range(4):
    for j in range(2):
        inst_name = 'Metal3_' + str(i * 2 + j)
        layer = 'ME3'
        x1 =
        x2 =
        y1 =
        y2 =
        ME3_specialnet[i][j] = SpecialNet(inst_name, layer, x1, y1, x2, y2)
```

- **Append ME3 instances to specialnet_list**

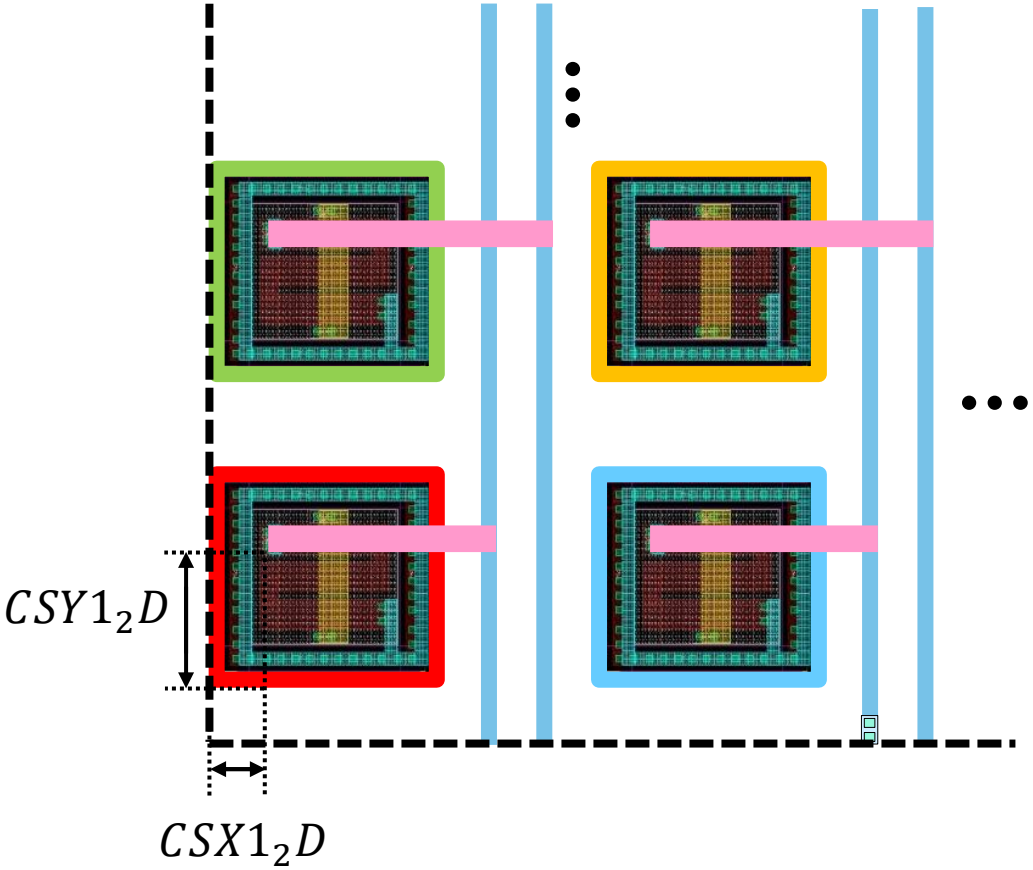
```
specialnet_list = []
for i in range(4):
    for j in range(2):
        specialnet_list.append(ME3_specialnet[i][j])
```

# Step 4: Create ME4 Drain Connection

- **Generate the connections for the four units of a device in a single for loop iteration**
- **Left-bottom portion** $(x_1,y_1)$


CS X1 to drain 1.26

CS Y1 to drain 4.1




$$x_1(M4D_{00}) = x_1(CS_{00}) + CSX1_2D$$

$$y_1(M4D_{00}) = y_1(CS_{00}) + CSY1_2D$$

$$x_2(M4D_{00}) = x_2(M3_{00})$$

$$y_2(M4D_{00}) = y_1(M4D_{00}) + M4_{width}$$

$$x_1(M4D_{ij}) = x_1(CS_{ij}) + CSX1_2D$$

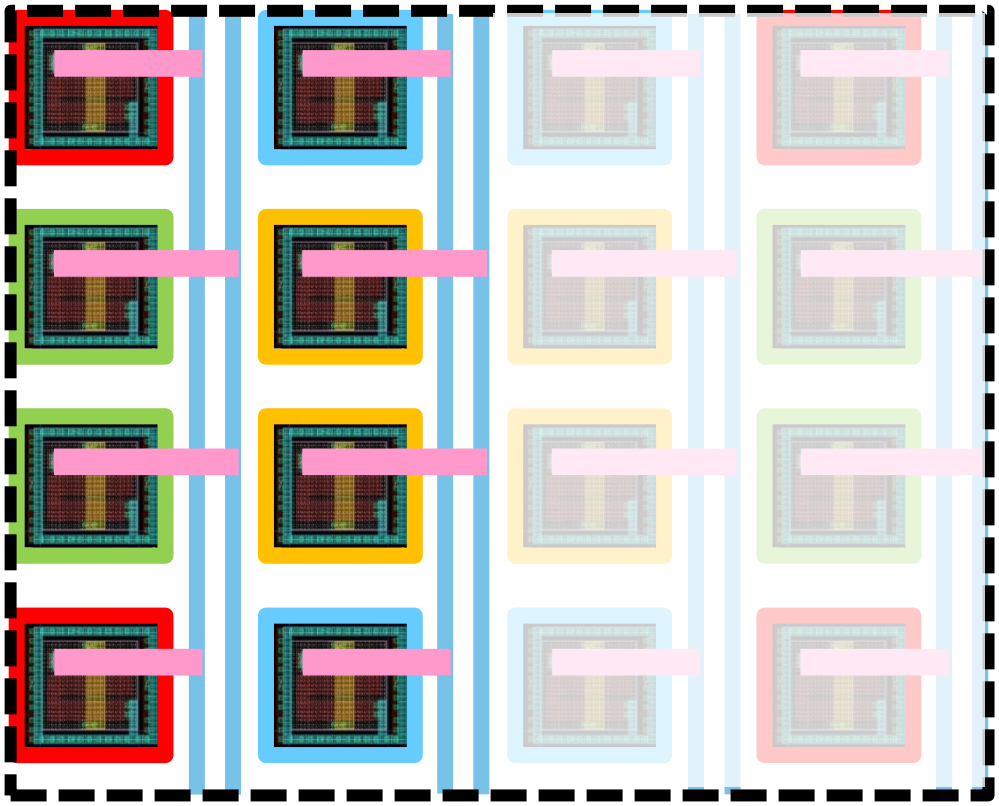
$$y_1(M4D_{ij}) = y_1(CS_{ij}) + CSY1_2D$$

$$x_2(M4D_{ij}) = x_2(M3_{ij})$$

$$y_2(M4D_{ij}) = y_1(M4D_{ij}) + M4_{width}$$

# Step 4: Create ME4 Drain Connection (cont'd)

- **Mirroring to x-axis**



**Left-bottom**

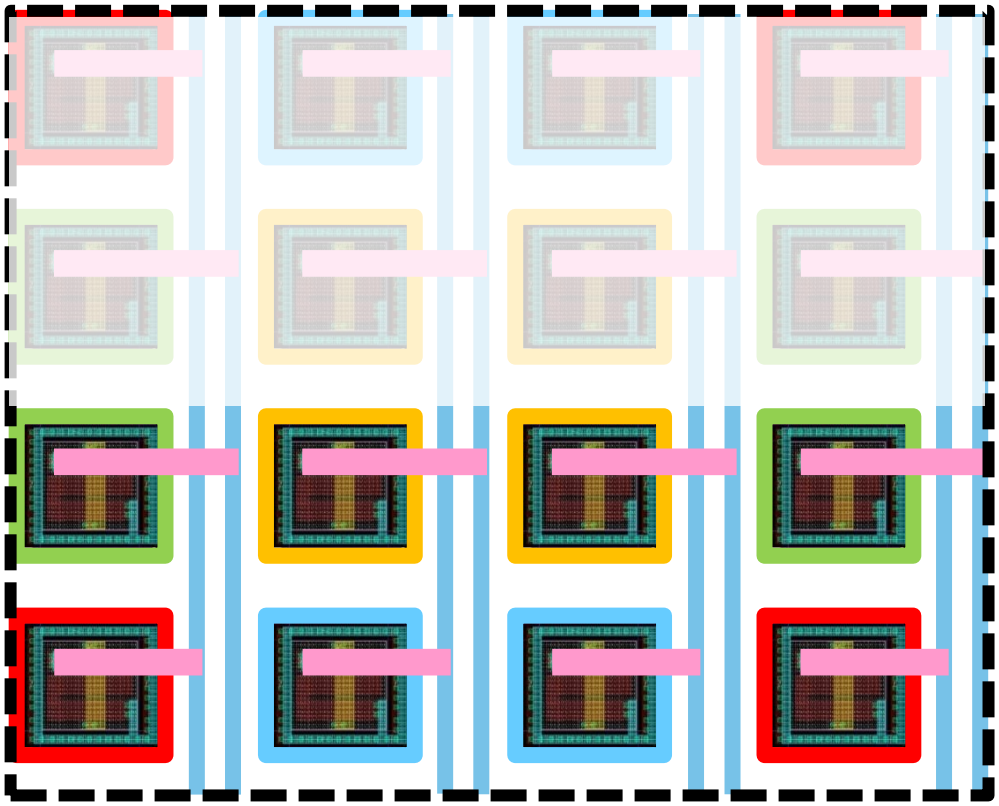
$$x_1(M4) = x_1(CS_{ij}) + CSX1_2D$$
$$y_1(M4) = y_1(CS_{ij}) + CSY1_2D$$
$$x_2(M4) = x_2(M3_{ij})$$
$$y_2(M4) = y_1(M4D_{ij}) + M4_{width}$$

**Left-top**

$$x_1(M4) = x_1(CS_{i,3-j}) + CSX1_2D$$
$$y_1(M4) = y_1(CS_{i,3-j}) + CSY1_2D$$
$$x_2(M4) = x_2(M3_{i,j})$$
$$y_2(M4) = y_1(M4D_{ij}) + M4_{width}$$

# Step 4: Create ME4 Drain Connection (cont'd)

- **Mirroring to y-axis**



**Left-bottom**

$$x_1(M4) = x_1(CS_{ij}) + CSX1_2D$$

$$y_1(M4) = y_1(CS_{ij}) + CSY1_2D$$

$$x_2(M4) = x_2(M3_{ij})$$

$$y_2(M4) = y_1(M4D_{ij}) + M4_{width}$$

**Right-bottom**

$$x_1(M4) = x_1(CS_{3-i,j}) + CSX1_2D$$

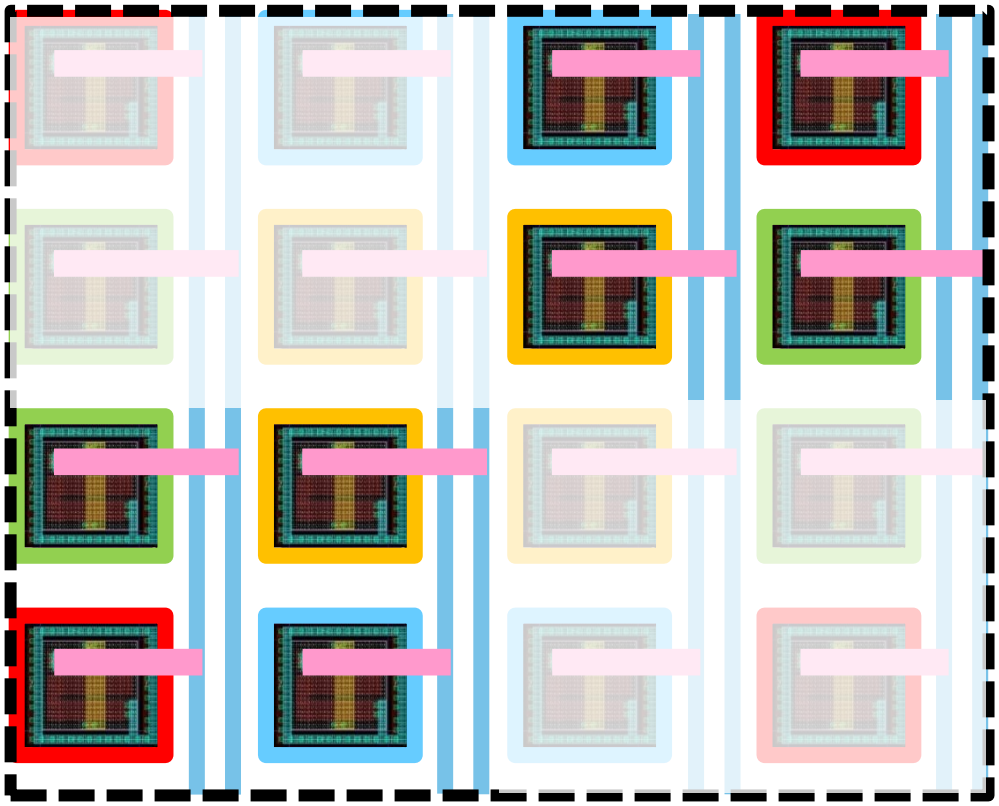
$$y_1(M4) = y_1(CS_{3-i,j}) + CSY1_2D$$

$$x_2(M4) = x_2(M3_{3-i,j})$$

$$y_2(M4) = y_1(M4D_{ij}) + M4_{width}$$

# Step 4: Create ME4 Drain Connection (cont'd)

- Mirroring to origin



Left-bottom

$$x_1(M4) = x_1(CS_{ij}) + CSX1_2D$$

$$y_1(M4) = y_1(CS_{ij}) + CSY1_2D$$

$$x_2(M4) = x_2(M3_{ij})$$

$$y_2(M4) = y_1(M4D_{ij}) + M4_{width}$$

Right-top

$x_1(M4) =$

$y_1(M4) =$

$x_2(M4) =$

$y_2(M4) =$

# Step 4: Create ME4 Drain Connection (cont'd)

- **Create horizontal ME4 metals from CS drain to ME3**

```
##### Step 4: create ME4 drain #####
# ME4 drains
ME4_specialnet_drain = [[SpecialNet for j in range(4)] for i in range(4)]
for i in range(2):
    for j in range(2):
        layer = 'ME4'
        # left bottom corner units
        inst_name = 'Metal4_drain_' + str(i * 2 + j + 0 * 4)
        x1 = cs_array[i][j]._x + CS_X1_TO_DRAIN
        x2 = ME3_specialnet[i][j]._x2
        y1 = cs_array[i][j]._y + CS_Y1_TO_DRAIN
        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_' + str(i * 2 + j + 1 * 4)
        x1 = cs_array[3-i][j]._x + CS_X1_TO_DRAIN
        x2 = ME3_specialnet[3-i][j]._x2
        y1 = cs_array[3-i][j]._y + CS_Y1_TO_DRAIN
        y2 = y1 + M4_WIDTH
        ME4_specialnet_drain[3-i][j] = SpecialNet(inst_name, layer, x1, y1, x2, y2)
        # left top corner units
        inst_name = 'Metal4_drain_' + str(i * 2 + j + 2 * 4)
        x1 = cs_array[i][3-j]._x + CS_X1_TO_DRAIN
        x2 = ME3_specialnet[i][j]._x2
        y1 = cs_array[i][3-j]._y + CS_Y1_TO_DRAIN
        y2 = y1 + M4_WIDTH
        ME4_specialnet_drain[i][3-j] = SpecialNet(inst_name, layer, x1, y1, x2, y2)
        # right top corner units
        inst_name = 'Metal4_drain_' + str(i * 2 + j + 3 * 4)
        x1 =
        x2 =
        y1 =
        y2 =
        ME4_specialnet_drain[3-i][3-j] = SpecialNet(inst_name, layer, x1, y1, x2, y2)
```

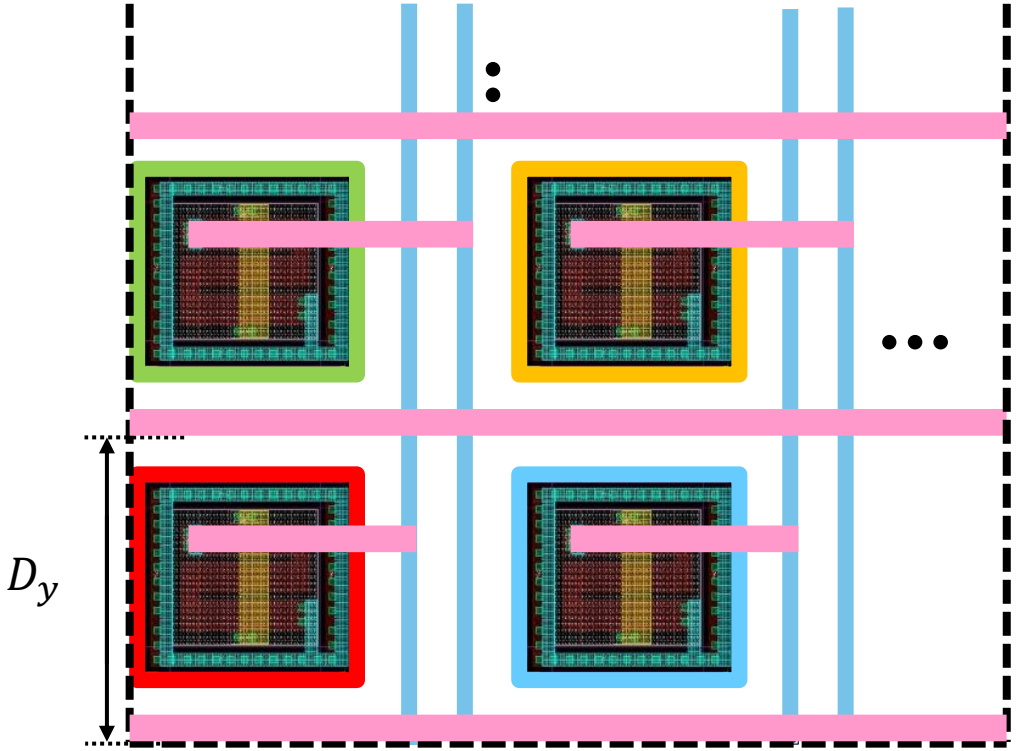
# Step 4: Create ME4 Drain Connection (cont'd)

- **Append ME4 Drain-ME3 instances to specialnet_list**

```
for i in range(4):
    for j in range(4):
        specialnet_list.append(ME4_specialnet_drain[i][j])
```

# Step 5: Create ME4 Port

- **Create ME4 ports for CSs**




$$x_1(M4_i) = 0$$

$$x_2(M4_i) = bx_2$$

$$y_1(M4_i) = i \times D_y$$

$$y_2(M4_i) = y_1(M4_i) + M4_{width}$$

# Step 5: Create ME4 Port (cont'd)

- **Create ME4 ports**

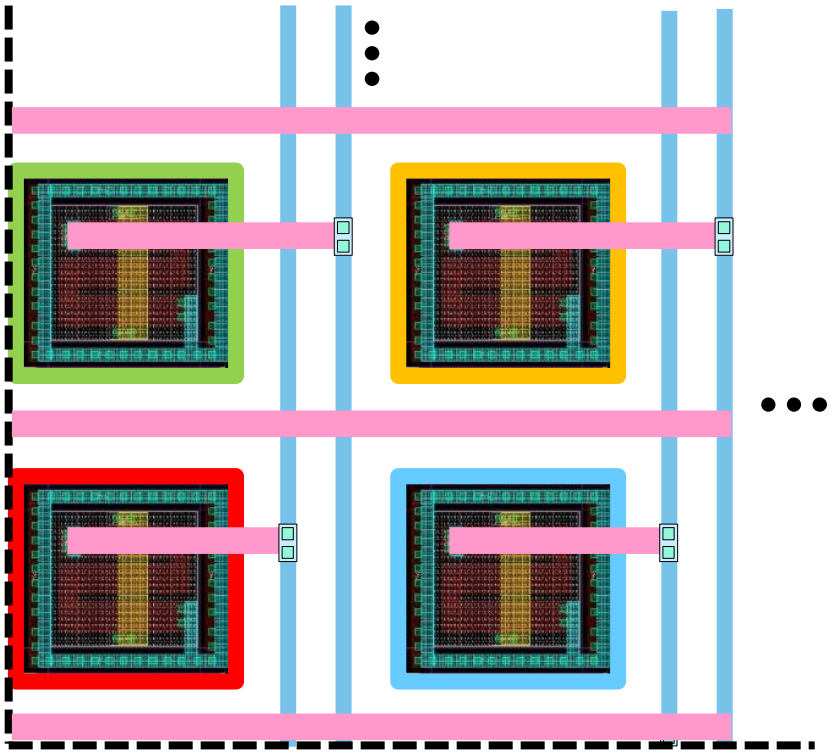
```
##### Step 5: create ME4 port #####
# ME4 ports
ME4_specialnet_port = [SpecialNet for i in range(4)]
for i in range(4):
    inst_name = 'Metal4_port_' + str(i)
    layer = 'ME4'
    x1 =
    x2 =
    y1 =
    y2 =
    ME4_specialnet_port[i] = SpecialNet(inst_name, layer, x1, y1, x2, y2)
```

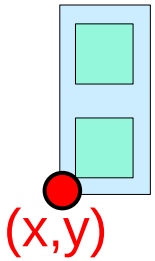
- **Append ME4 ports instances to specialnet_list**

```
for i in range(4):
    specialnet_list.append(ME4_specialnet_port[i])
```

# Step 6: Via34 from ME4 Drain to ME3

- **Left-bottom portion**






$$x_1(VD_{00}) = x_1(M3_{00})$$

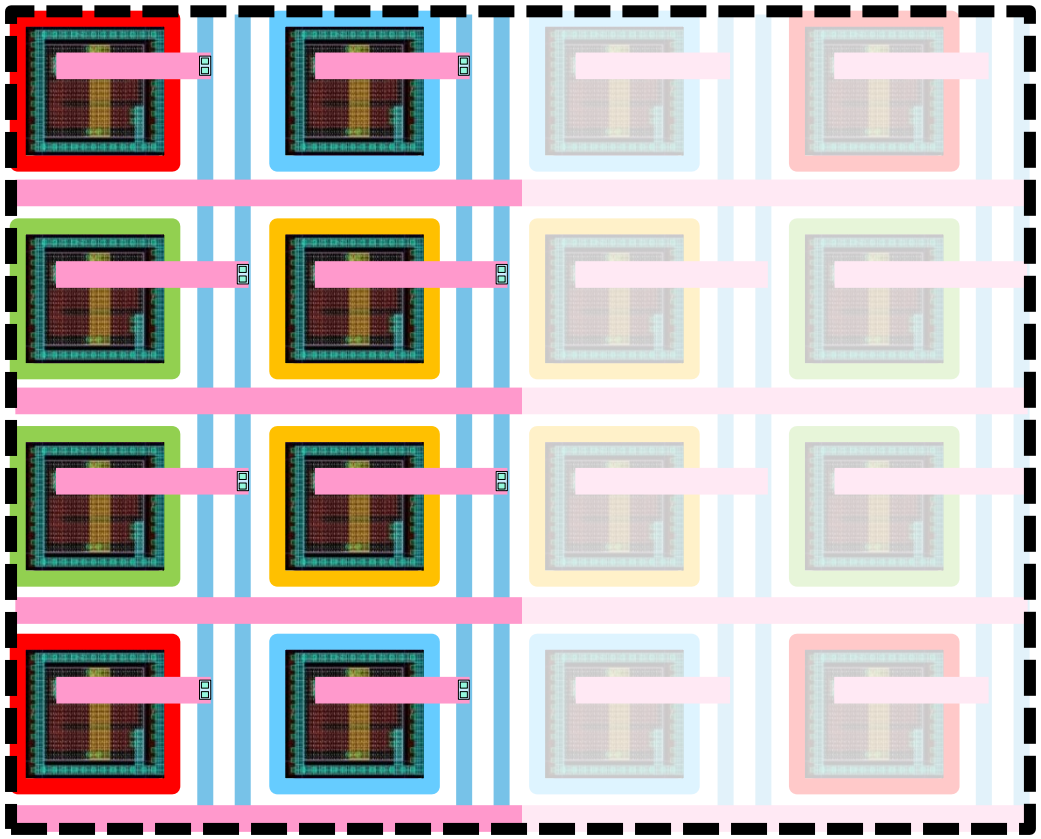
$$y_1(VD_{00}) = y_1(CS_{00}) + CSY1_2D$$

$$x_1(VD_{ij}) = x_1(M3_{ij})$$

$$y_1(VD_{ij}) = y_1(CS_{ij}) + CSY1_2D$$

# Step 6: Via34 from ME4 Drain to ME3 (cont'd)

- **Mirroring to x-axis**



Left-bottom

$$x_1(VD) = x_1(M3_{ij})$$

$$y_1(VD) = y_1(CS_{ij}) + CSY1_2D$$

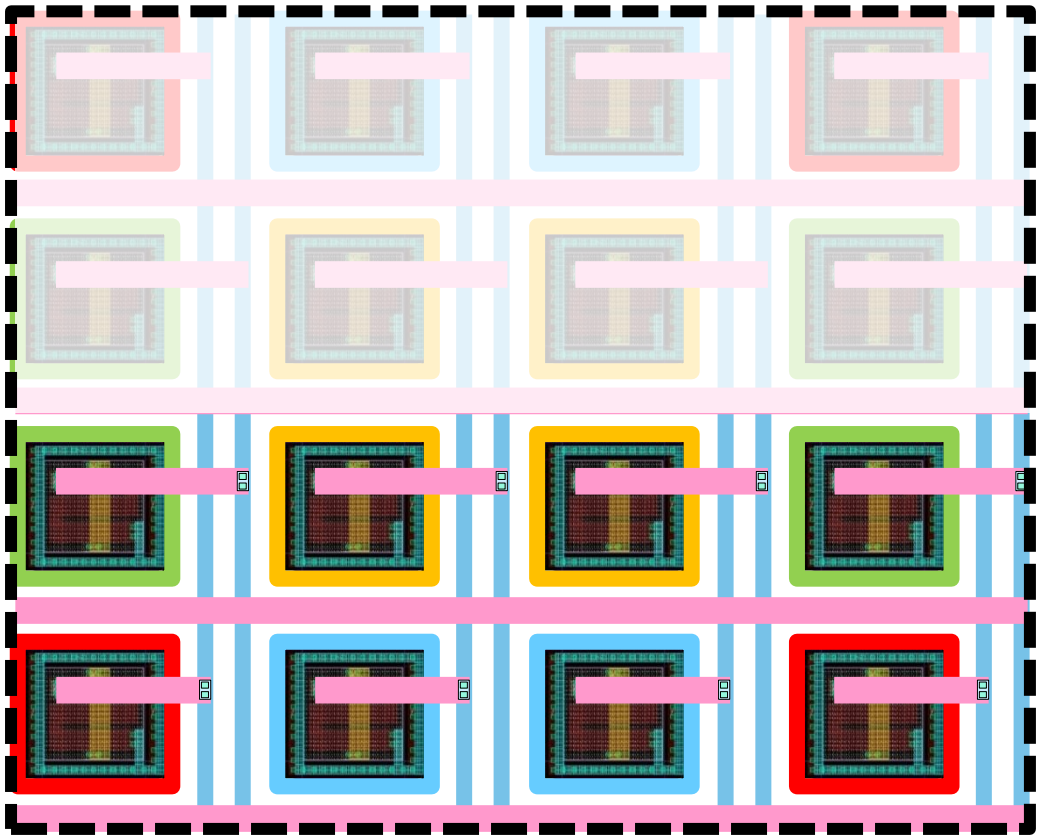
Left-top

$$x_1(VD) = x_1(M3_{ij})$$

$$y_1(VD) = y_1(CS_{i,3-j}) + CSY1_2D$$

# Step 6: Via34 from ME4 Drain to ME3 (cont'd)

- **Mirroring to y-axis**



Left-bottom

$$x_1(VD) = x_1(M3_{ij})$$

$$y_1(VD) = y_1(CS_{ij}) + CSY1_2D$$

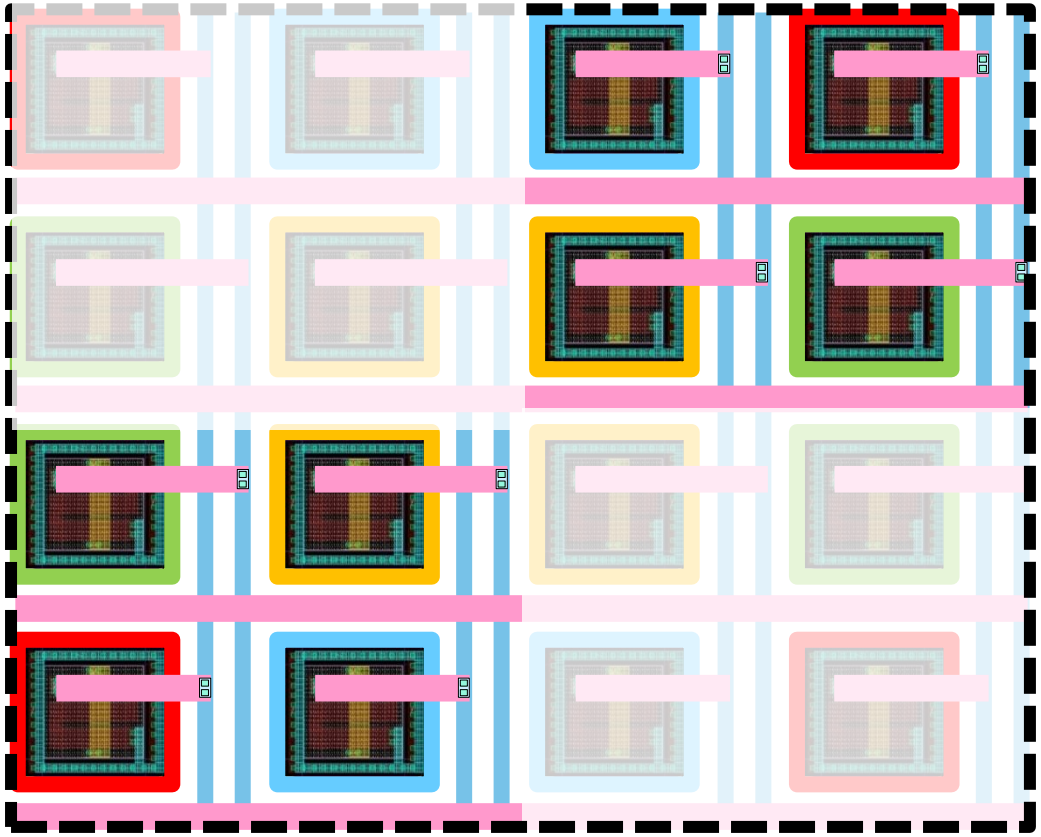
Right-bottom

$$x_1(VD) = x_1(M3_{3-i,j})$$

$$y_1(VD) = y_1(CS_{3-i,j}) + CSY1_2D$$

# Step 6: Via34 from ME4 Drain to ME3 (cont'd)

- Mirroring to origin



Left-bottom

$$x_1(VD) = x_1(M3_{ij})$$

$$y_1(VD) = y_1(CS_{ij}) + CSY1_2D$$

Right-top

$$x_1(VD) =$$

$$y_1(VD) =$$

# Step 6: Via34 from ME4 Drain to ME3 (cont'd)

- **Create ME4 drain to ME3 via34**

```
##### Step 6: create Via34 from ME4 drain #####
# drain to ME3
Via34_drain2ME3 = [[Component for j in range(4)] for i in range(4)]
for i in range(2):
    for j in range(2):
        lib_name = VIA34_LIB_NAME
        # left bottom corner units
        inst_name = 'Via34_drain2ME3_' + str(i * 2 + j + 0 * 4)
        x = ME3_specialnet[i][j]._x1
        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_' + str(i * 2 + j + 1 * 4)
        x = ME3_specialnet[3-i][j]._x1
        y = cs_array[3-i][j]._y + CS_Y1_TO_DRAIN
        Via34_drain2ME3[3-i][j] = Component(lib_name, inst_name, x, y)
        # left top corner units
        inst_name = 'Via34_drain2ME3_' + str(i * 2 + j + 2 * 4)
        x = ME3_specialnet[i][j]._x1
        y = cs_array[i][3-j]._y + CS_Y1_TO_DRAIN
        Via34_drain2ME3[i][3-j] = Component(lib_name, inst_name, x, y)
        # right top corner units
        inst_name = 'Via34_drain2ME3_' + str(i * 2 + j + 3 * 4)
        x =
        y =
        Via34_drain2ME3[3-i][3-j] = Component(lib_name, inst_name, x, y)
```

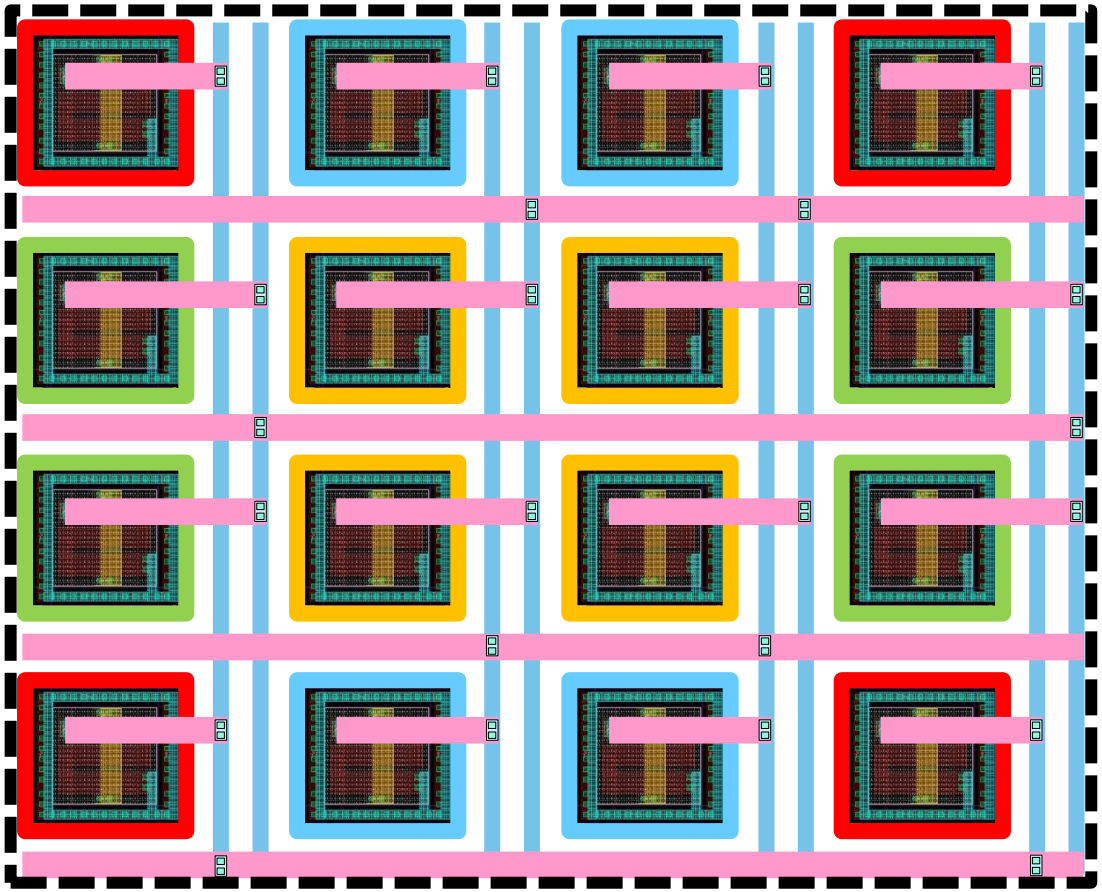
# Step 6: Via34 from ME4 Drain to ME3 (cont'd)

- **Append drain to ME3 Via34 instances to component_list**

```
for i in range(4):
    for j in range(4):
        component_list.append(Via34_drain2ME3[i][j])
```

# Step 7: Via34 from ME3 to ME4 Port

- Red CS connection



# Step 7: Via34 from ME3 to ME4 Port (cont'd)

- **Create port Via34**

```
##### Step 7: create Via34 to ME4 port #####
# port to ME4
# Precaution:
# 1. create a port list 'Via34_port2ME3' to contain port 'Component'
# 2. lib_name = VIA34_LIB_NAME
# 3. inst_name = 'Via34_port2ME3_'
# TODO
```

- **Append port Via34 instances to component_list**

```
# 4. add 'Via34_port2ME3' component to 'component_list'
# TODO
```

# How to Execute Python

- **Please enter the follow command to generate the DEF file.**

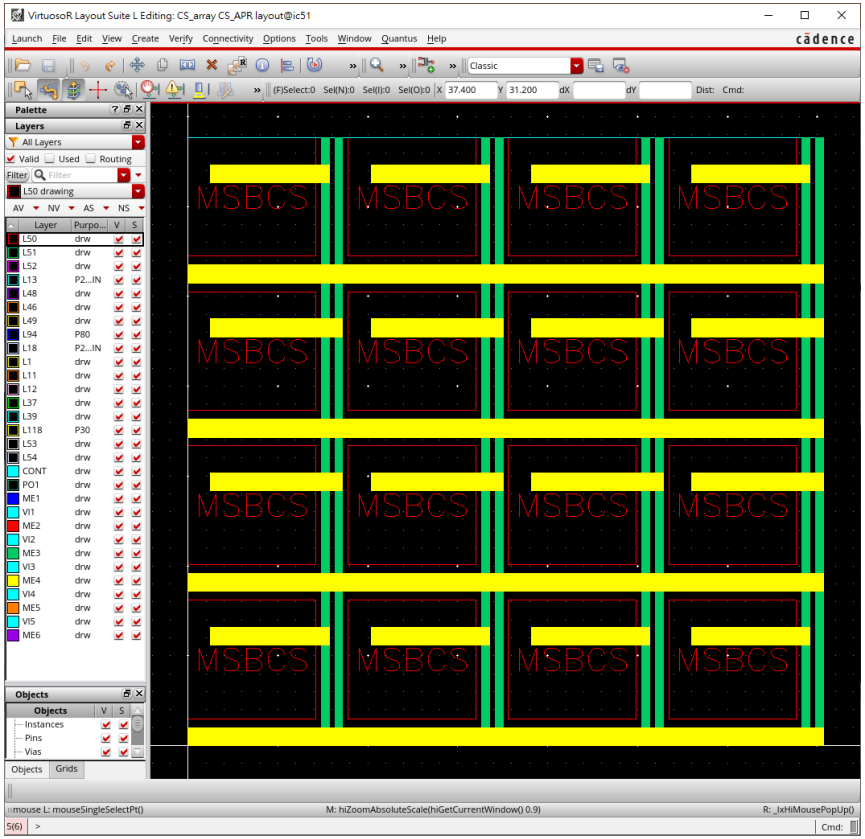
```
$ cd HW5/python
$ python3 main.py
```

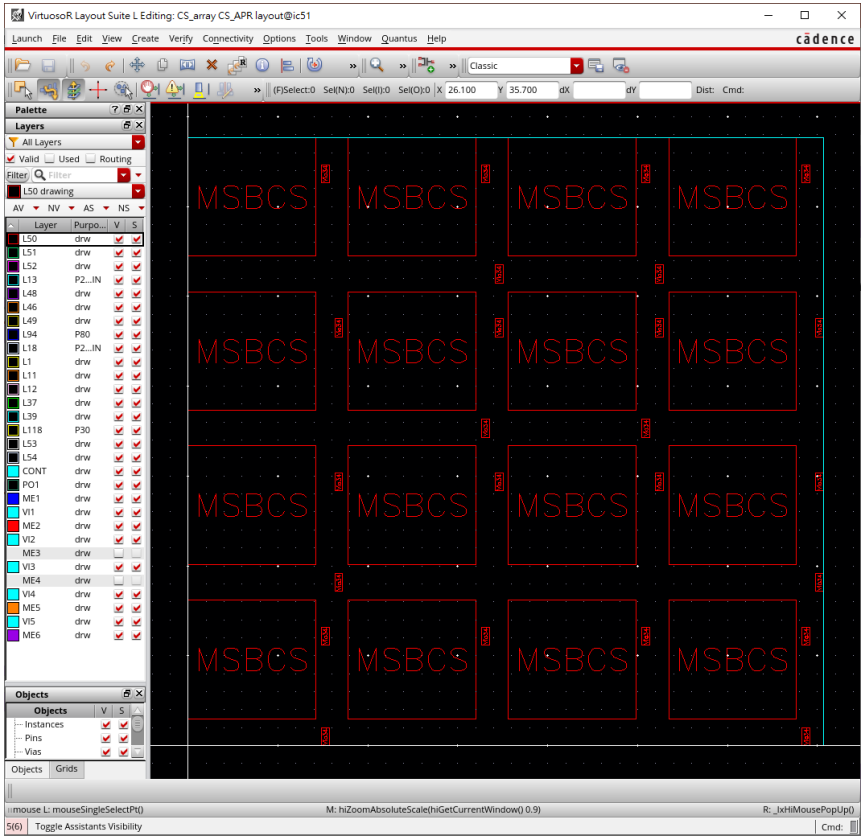
- **The DEF file will be generated in the “DEF” directory and named CS_4.def.**

- **Please make sure the DEF file exists when you hand in HW5.**

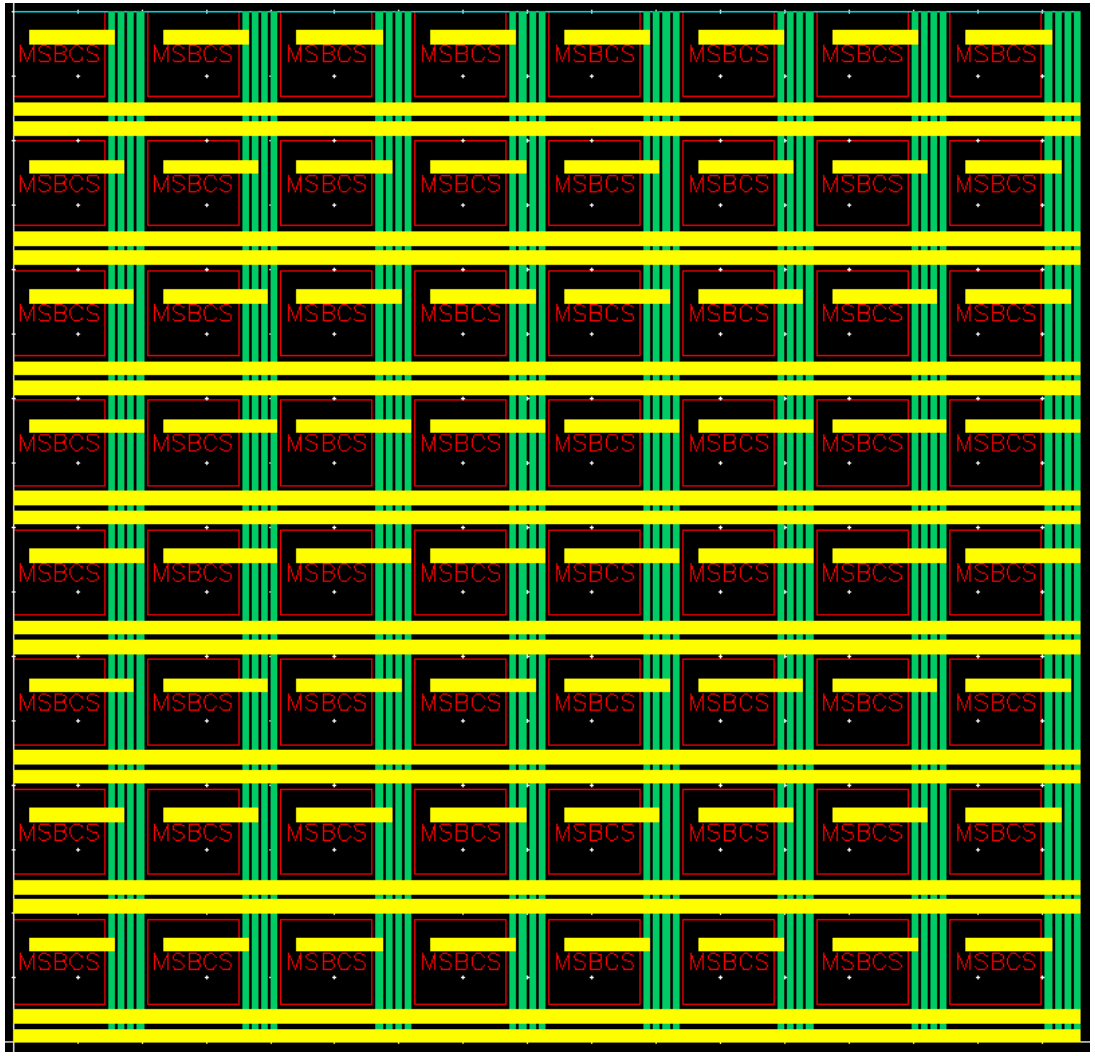
# Visualization

- **Please follow the tutorial written in CS6135_HW5_spec.pdf to visualize your circuit.**
- **It should be the same as the pictures below.**

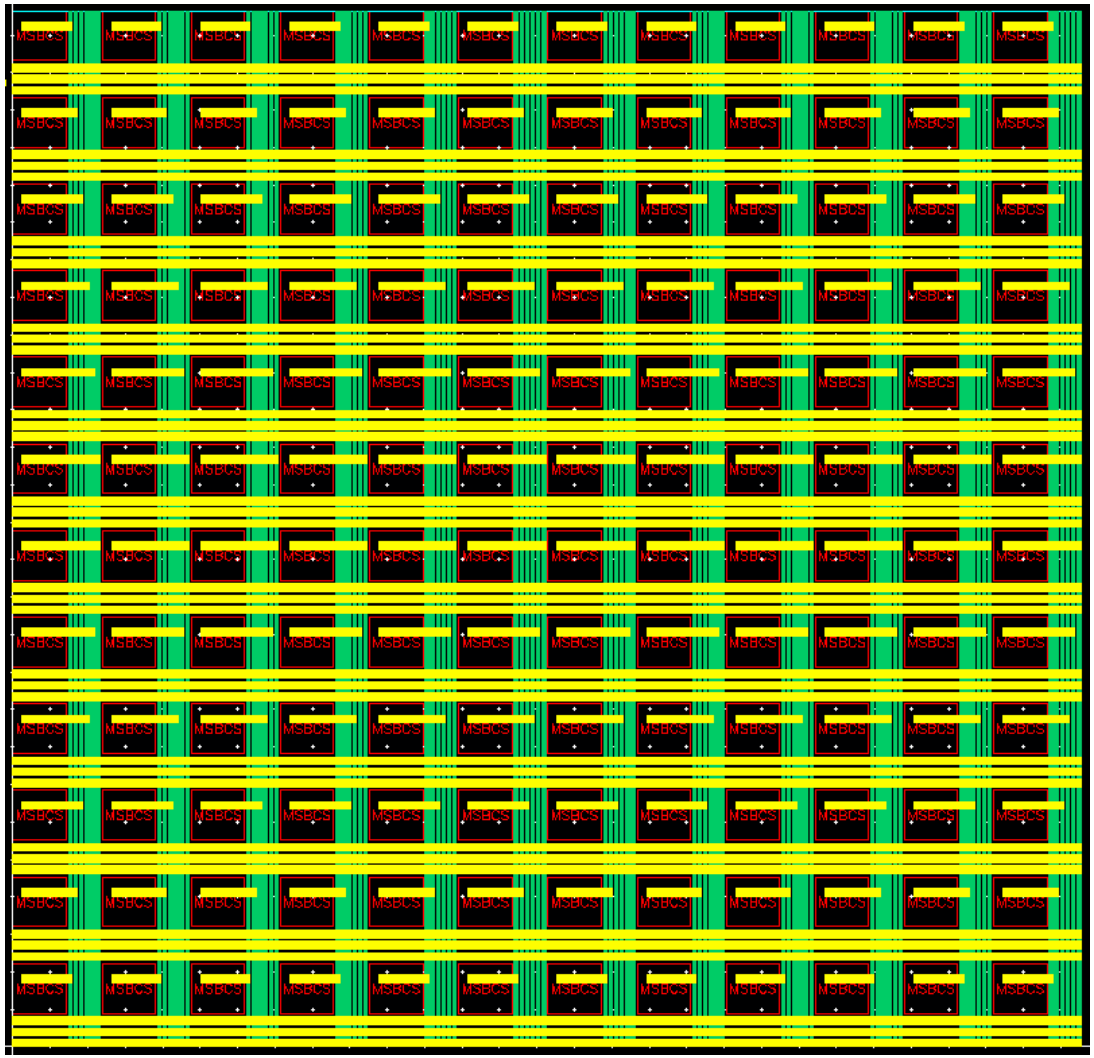


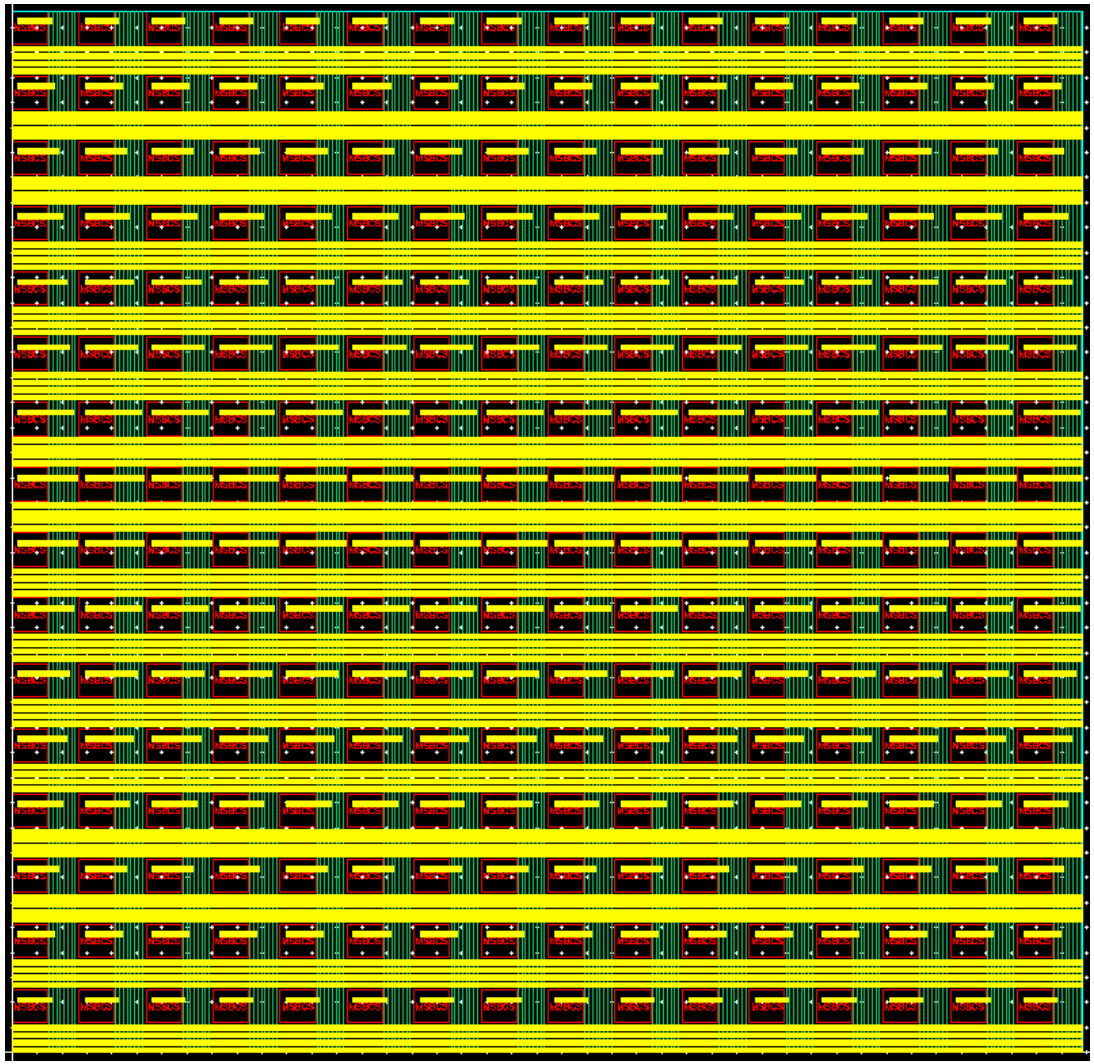
# 16 Current Sources



# 36 Current Sources



# 64 Current Sources



# 100 Current Sources