SPLayout 快速开始

赵振宇

2021-07-02

1 基本介绍

SPLayout是为了加快硅基光电子版图绘制而设计的，相较于基于的gdspy有更加简化的接口和操作方式。这个工具来源于加快绘制阵列光栅版图时将器件和指定位置的光栅进行连接的操作，基本思想为只要已知器件端口的中心点，即可快速完成与之相连的器件的绘制。同时新绘制的器件又可以反馈其各端口中心点，进而加速整个版图的绘制流程。

2 第一个 GDSII

可以通过以下代码创建第一个包含一个基本波导的 gdsii 文件。

```python
from splayout import *

# define cell
cell = Cell("waveguide")
# define layer
wg_layer = Layer(1,0)

# start point and end point for the waveguide
wg_start_point = Point(0,0)
wg_end_point = wg_start_point + (10,0)
# make a waveguide
wg = Waveguide(wg_start_point,wg_end_point,width=0.5)
# draw the waveguide on the layout
wg.draw(cell,wg_layer)

# create file and save
make_gdsii_file("waveguide.gds")
```

首先创建一个名为“waveguide”的 cell，然后定义了一个从 Point(0,0) 到 Point(10,0) 的波导，并将它画在“waveguide” cell 上，位于“1/0”层。最后生成这个“waveguide.gds”文件。这个文件可以使用KLayout查看。

3 基本结构

一个演示基本结构的可运行 python 脚本“basic.py”可以在SPLayout/basic.py at main ·Hideous-mon/SPLayout (github.com)中获取。具体代码如下：

```python
""" https://github.com/Hideousmon/SPLayout """
from splayout import *
import math


############### cell and layer define ###############

# define cell and layer
cell = Cell("basic")
wg_layer = Layer(1,0)
grating_layer = Layer(2,0)
heater_layer = Layer(3,0)
contact_layer = Layer(4,0)
inv_layer = Layer(50,0)
cover_layer = Layer(51,0)


############### components layout ###############


#### waveguide example ####
# start point and end point for the waveguide
wg_start_point = Point(0,0)
wg_end_point = wg_start_point + (10,0)
# make a waveguide
wg = Waveguide(wg_start_point,wg_end_point,width=0.5)
# draw the waveguide on the layout
wg.draw(cell,wg_layer)

#### taper example ####
# start point and end point for the taper
tp_start_point = Point(30,0)
tp_end_point = tp_start_point + (0,5)
# make a taper
tp = Taper(tp_start_point,tp_end_point,start_width=0.5,end_width
    =1)
# draw the taper on the layout
tp.draw(cell,wg_layer)

#### bend example ####
# center point and angle for the bend
center_point = Point(60,0)
start_angle = math.pi*0
end_angle = math.pi*3/5
```

```
width = 0.5
radius = 5
# make a bend
first_bend = Bend(center_point, start_angle, end_angle, width , radius)
# draw the bend on the layout
first_bend.draw(cell,wg_layer)


#### clockwise quarbend example ####
# start point and end point for the quarbend
start_point = Point(90,0)
end_point = start_point + (-7,20)
# make a quarbend
first_QuarBend = QuarBend(start_point,end_point,width=0.5)
# draw the quarbend on the layout
first_QuarBend.draw(cell,wg_layer)

#### anti-clockwise quarbend example 2 ####
# start point and end point for the anti-clockwise quarbend
start_point = Point(0,-30)
end_point = start_point + (-7,20)
# make the anti-clockwise quarbend
first_AQuarBend = AQuarBend(start_point,end_point,width=0.5)
# draw the anti-clockwise quarbend on the layout
first_AQuarBend.draw(cell,wg_layer)

#### polygon example ####
# points for the polygon
pointlist = [Point(30,-30) ,Point(30,-25),Point(37,-20),Point(33,-27),Point(32,-28),Point(31,-29)] ## or [(30,-30),(30,-25),(37,-20),(33,-27),(32,-28),(31,-29)]
# make the polygon
polygon = Polygon(pointlist)
# draw the polygon on the layout
polygon.draw(cell,wg_layer)

#### add drop microring example ####
# start point(input point) for the microring, and radius, gap, waveguide width, coupling length
start_point = Point(50,40)
radius = 5.1973
gap = 0.18
wg_width = 0.45
coupling_length = 5.5
```

```
85  # make the add drop microring
86  first_ring = AddDropMicroring(start_point,radius,gap,wg_width,
        coupling_length)
87  # drawe the microring on the layout
88  first_ring.draw(cell,wg_layer)
89  # add heater for the microring
90  first_ring.add_heater(cell, heater_layer, contact=1,
        contact_layer=contact_layer)
91
92
93  #### flat coupling microring example ####
94  # start point(input point) for the microring, and radius, gap,
        waveguide width, coupling length
95  start_point = Point(50,-300)
96  radius = 5.1973
97  gap = 0.18
98  wg_width = 0.45
99  coupling_length = 5.5
100 # make the add drop microring
101 second_ring = AddDropMicroringFlat(start_point,radius,gap,
        wg_width,coupling_length)
102 # draw the microring on the layout
103 second_ring.draw(cell,wg_layer)
104 # add heater for the microring
105 second_ring.add_heater(cell, heater_layer, contact=1,
        contact_layer=contact_layer)
106
107
108 #### doubleconnector example ####
109 # start point and end point for the doubleconnector
110 double_connect_start_point = Point(60,-30)
111 double_connect_end_point = double_connect_start_point + (20,10)
112 # make the doubleconnector
113 connector = DoubleBendConnector(double_connect_start_point,
        double_connect_end_point, width=0.5)
114 # draw the doubleconnector on the layout
115 connector.draw(cell,wg_layer)
116
117 #### text example ####
118 # start point for the text
119 text_start_point = Point(0,-60)
120 # make the text
121 text = Text(text_start_point,"OTIP2021")
122 # draw the text on the layout
123 text.draw(cell,wg_layer)
```

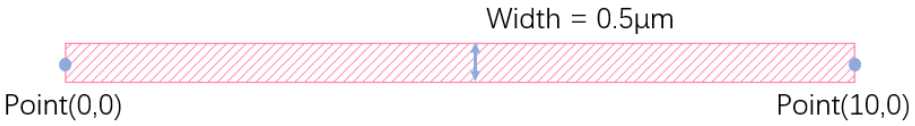
```
124
125 #### AEMD grating example ####
126 # get a AEMD grating definition
127 AEMDgrating = MAKE_AEMD_GRATING(port_width=0.5)
128 # start point for the AEMD grating
129 grating_point = Point(90,-30)
130 # make the AEMD grating
131 right_grating = AEMDgrating(grating_point,RIGHT)
132 # draw the AEMD grating on the layout
133 right_grating.draw(cell)
134
135 #### self define component example ####
136 # take the "selfdefine.gds" as an example
137 SelfDefineComponent = MAKE_COMPONENT("selfdefine.gds")
138 # start point for the component
139 start_point = Point(0,-90)
140 # make the component
141 component = SelfDefineComponent(start_point,RIGHT)
142 # draw the component on the layout
143 component.draw(cell)
144
145 ############### make gdsii file ###############
146 # create file and save
147 make_gdsii_file("basic.gds")
148 # create file and save the layout with an inverse layer
149 make_gdsii_file("basic_inverse.gds",inv_source_layer=wg_layer,
      inv_target_layer=inv_layer)
150 # create file and save the layout with inverse layer and cover
      layer
151 make_gdsii_file("basic_inverse_and_cover.gds",inv_source_layer=
      wg_layer,inv_target_layer=inv_layer,cover_source_layer=
      wg_layer,cover_target_layer=cover_layer)
```

下面内容为对这些基本结构对应的版图属性。

Waveguide

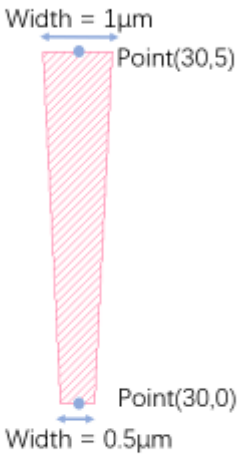
```
1 # start point and end point for the waveguide
2 wg_start_point = Point(0,0)
3 wg_end_point = wg_start_point + (10,0)
4 # make a waveguide
5 wg = Waveguide(wg_start_point,wg_end_point,width=0.5)
6 # draw the waveguide on the layout
```

```
7 wg.draw(cell,wg_layer)
```



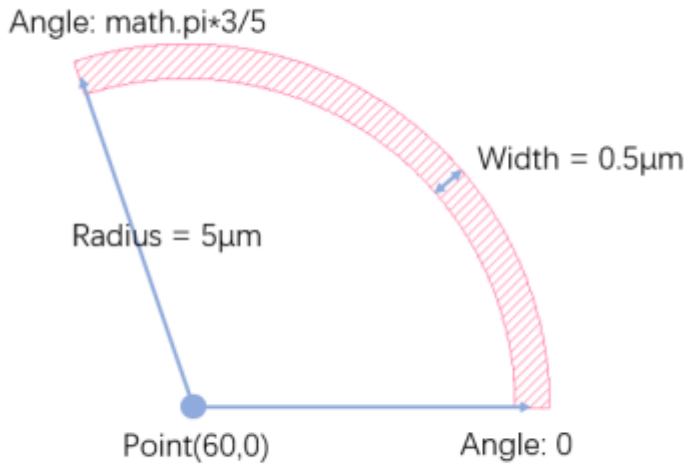
Taper

```
1 # start point and end point for the taper
2 tp_start_point = Point(30,0)
3 tp_end_point = tp_start_point + (0,5)
4 # make a taper
5 tp = Taper(tp_start_point,tp_end_point,start_width=0.5,end_width
     =1)
6 # draw the taper on the layout
7 tp.draw(cell,wg_layer)
```



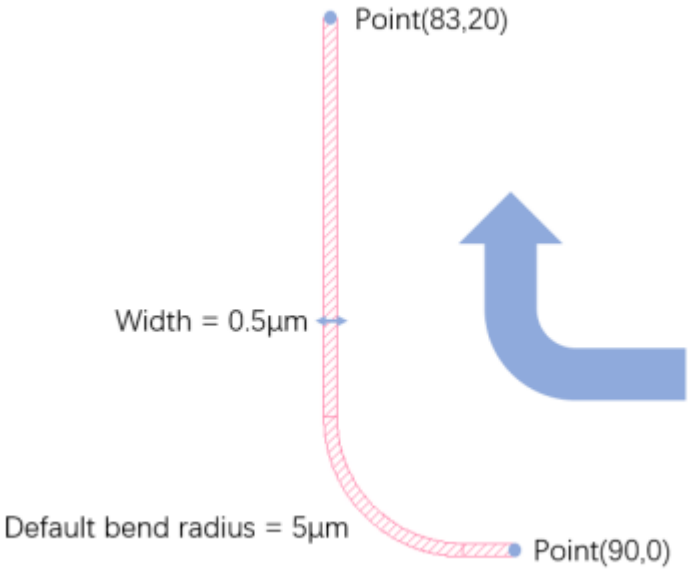
Bend

```
# center point and angle for the bend
center_point = Point(60,0)
start_angle = math.pi*0
end_angle = math.pi*3/5
width = 0.5
radius = 5
# make a bend
first_bend = Bend(center_point, start_angle, end_angle, width ,
    radius)
# draw the bend on the layout
first_bend.draw(cell,wg_layer)
```



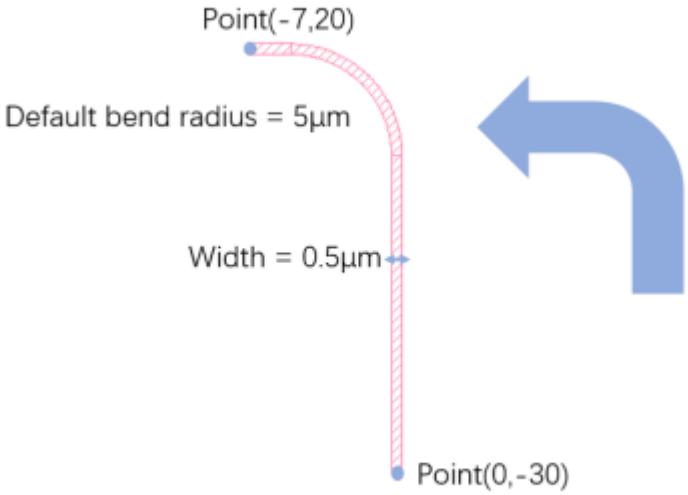
QuarBend

```
# start point and end point for the quarbend
start_point = Point(90,0)
end_point = start_point + (-7,20)
# make a quarbend
first_QuarBend = QuarBend(start_point,end_point,width=0.5)
# draw the quarbend on the layout
first_QuarBend.draw(cell,wg_layer)
```



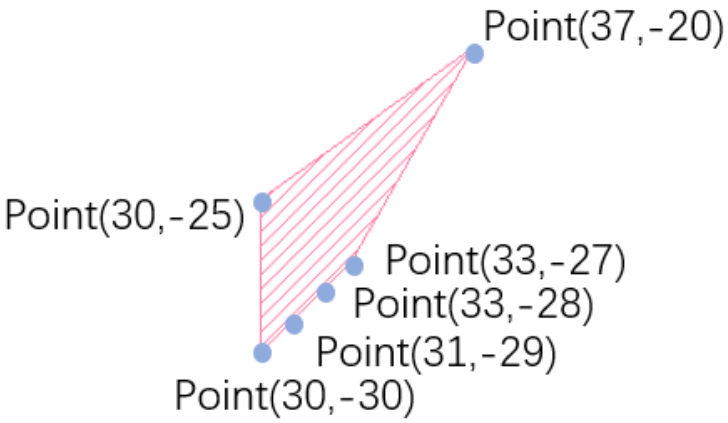
AQuarBend

```
# start point and end point for the anti-clockwise quarbend
start_point = Point(0,-30)
end_point = start_point + (-7,20)
# make the anti-clockwise quarbend
first_AQuarBend = AQuarBend(start_point,end_point,width=0.5)
# draw the anti-clockwise quarbend on the layout
first_AQuarBend.draw(cell,wg_layer)
```



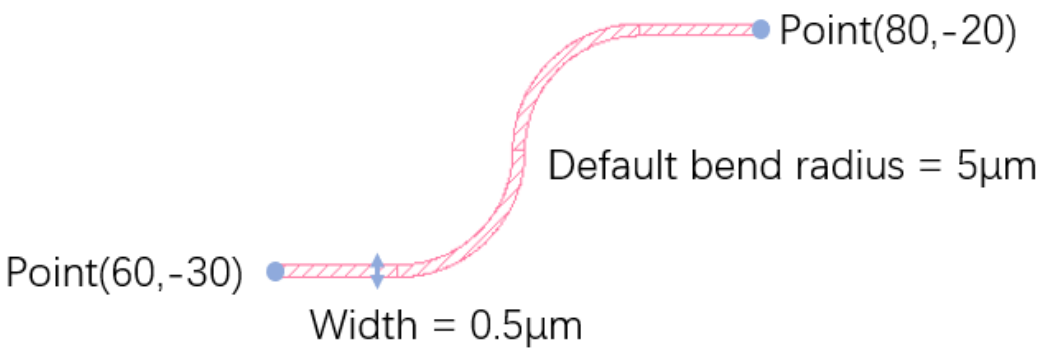
Polygon

```
# points for the polygon
pointlist = [Point(30,-30) ,Point(30,-25),Point(37,-20),Point
    (33,-27),Point(32,-28),Point(31,-29)] ## or [(30,-30)
    ,(30,-25),(37,-20),(33,-27),(32,-28),(31,-29)]
# make the polygon
polygon = Polygon(pointlist)
# draw the polygon on the layout
polygon.draw(cell,wg_layer)
```



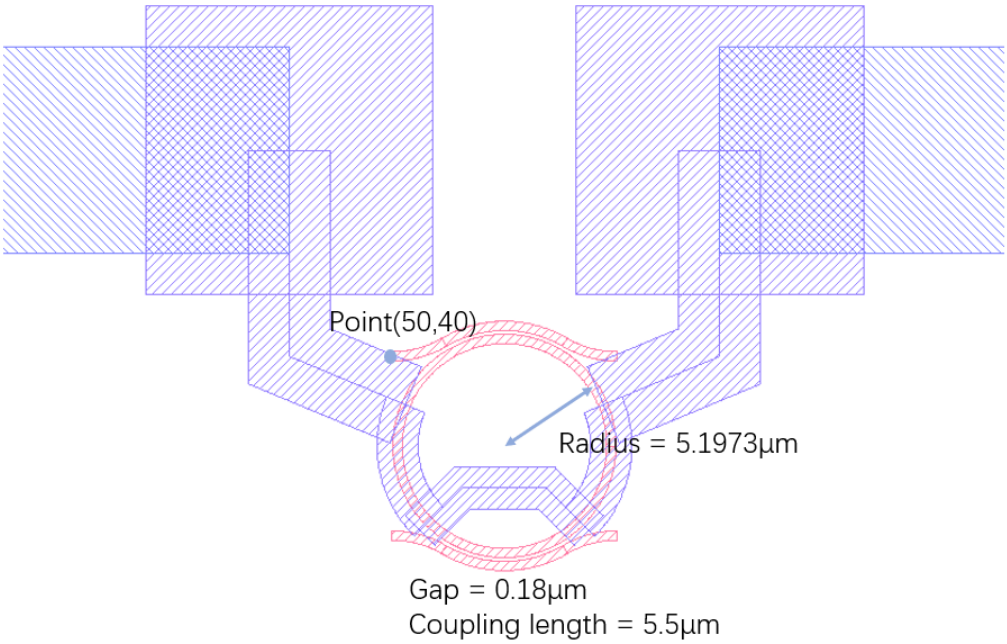
DoubleBendConnector

```
# start point and end point for the doubleconnector
double_connect_start_point = Point(60,-30)
double_connect_end_point = double_connect_start_point + (20,10)
# make the doubleconnector
connector = DoubleBendConnector(double_connect_start_point,
   double_connect_end_point, width=0.5)
# draw the doubleconnector on the layout
connector.draw(cell,wg_layer)
```



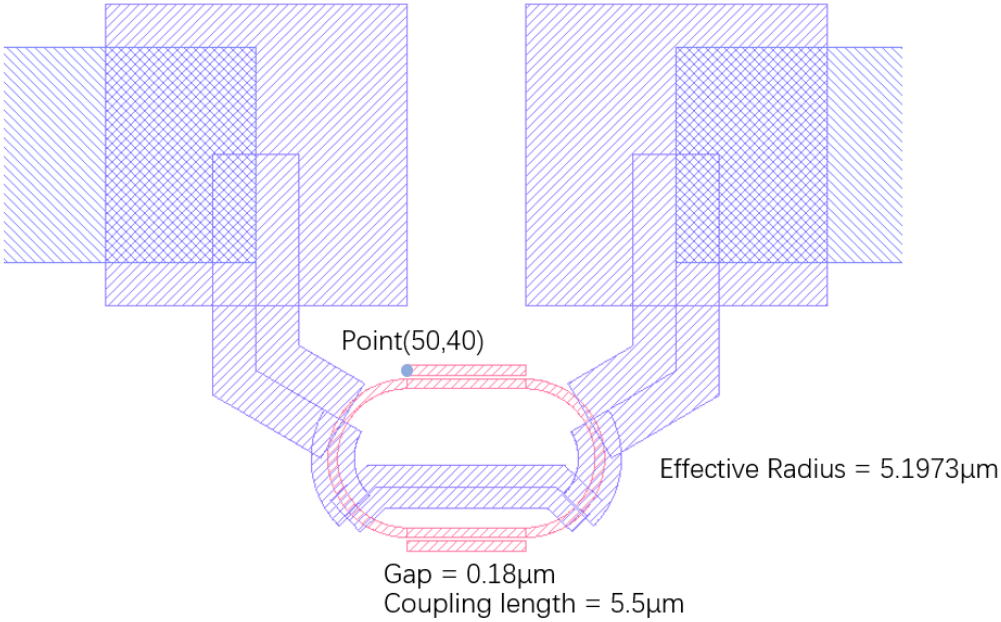
AddDropMicroring

```
# start point(input point) for the microring, and radius, gap,
    waveguide width, coupling length
start_point = Point(50,40)
radius = 5.1973
gap = 0.18
wg_width = 0.45
coupling_length = 5.5
# make the add drop microring
first_ring = AddDropMicroring(start_point,radius,gap,wg_width,
    coupling_length)
# drawe the microring on the layout
first_ring.draw(cell,wg_layer)
# add heater for the microring
first_ring.add_heater(cell, heater_layer, contact=1,
    contact_layer=contact_layer)
```



AddDropMicroringFlat

```
start_point = Point(50,-300)
radius = 5.1973
gap = 0.18
wg_width = 0.45
coupling_length = 5.5
# make the add drop microring
second_ring = AddDropMicroringFlat(start_point,radius,gap,
   wg_width,coupling_length)
# draw the microring on the layout
second_ring.draw(cell,wg_layer)
# add heater for the microring
second_ring.add_heater(cell, heater_layer, contact=1,
   contact_layer=contact_layer)
```



Text

```
# start point for the text
text_start_point = Point(0,-60)
# make the text
text = Text(text_start_point,"OTIP2021")
# draw the text on the layout
text.draw(cell,wg_layer)
```

4 半自定义/自定义器件

AEMD Grating

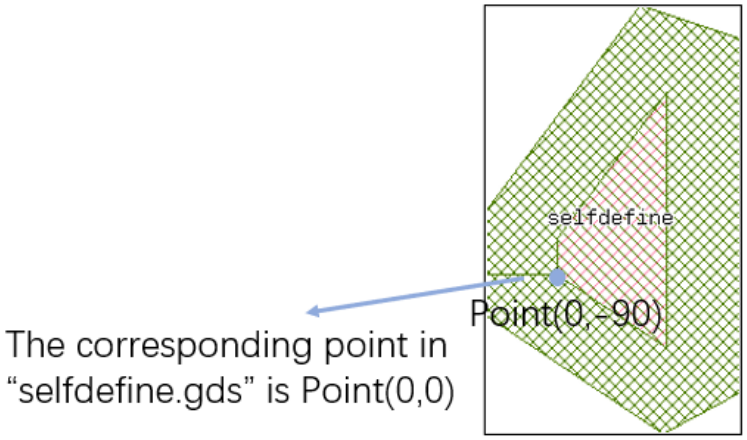
可以使用“MAKE_AEMD_GRATING”函数返回一个“类”，返回的类的用法与之前的基本结构中器件的使用方法类似. 具体细节可以参考SPLayout — SPLayout 0.0.3 documentation中的“API Reference”中相应的部分.

```
# get a AEMD grating definition
AEMDgrating = MAKE_AEMD_GRATING(port_width=0.5)
# start point for the AEMD grating
grating_point = Point(90,-30)
# make the AEMD grating
right_grating = AEMDgrating(grating_point,RIGHT)
# draw the AEMD grating on the layout
right_grating.draw(cell)
```

Self-define Components

可以使用“MAKE_COMPONENT”函数根据已有的 gdsii 文件返回一个“类”，返回的类的用法与之前的基本结构中器件的使用方法类似. 具体细节可以参考SPLayout — SPLayout 0.0.3 documentation中的“API Reference”中相应的部分.

```
# take the "selfdefine.gds" as an example
SelfDefineComponent = MAKE_COMPONENT("selfdefine.gds")
# start point for the component
start_point = Point(0,-90)
# make the component
component = SelfDefineComponent(start_point,RIGHT)
# draw the component on the layout
component.draw(cell)
```

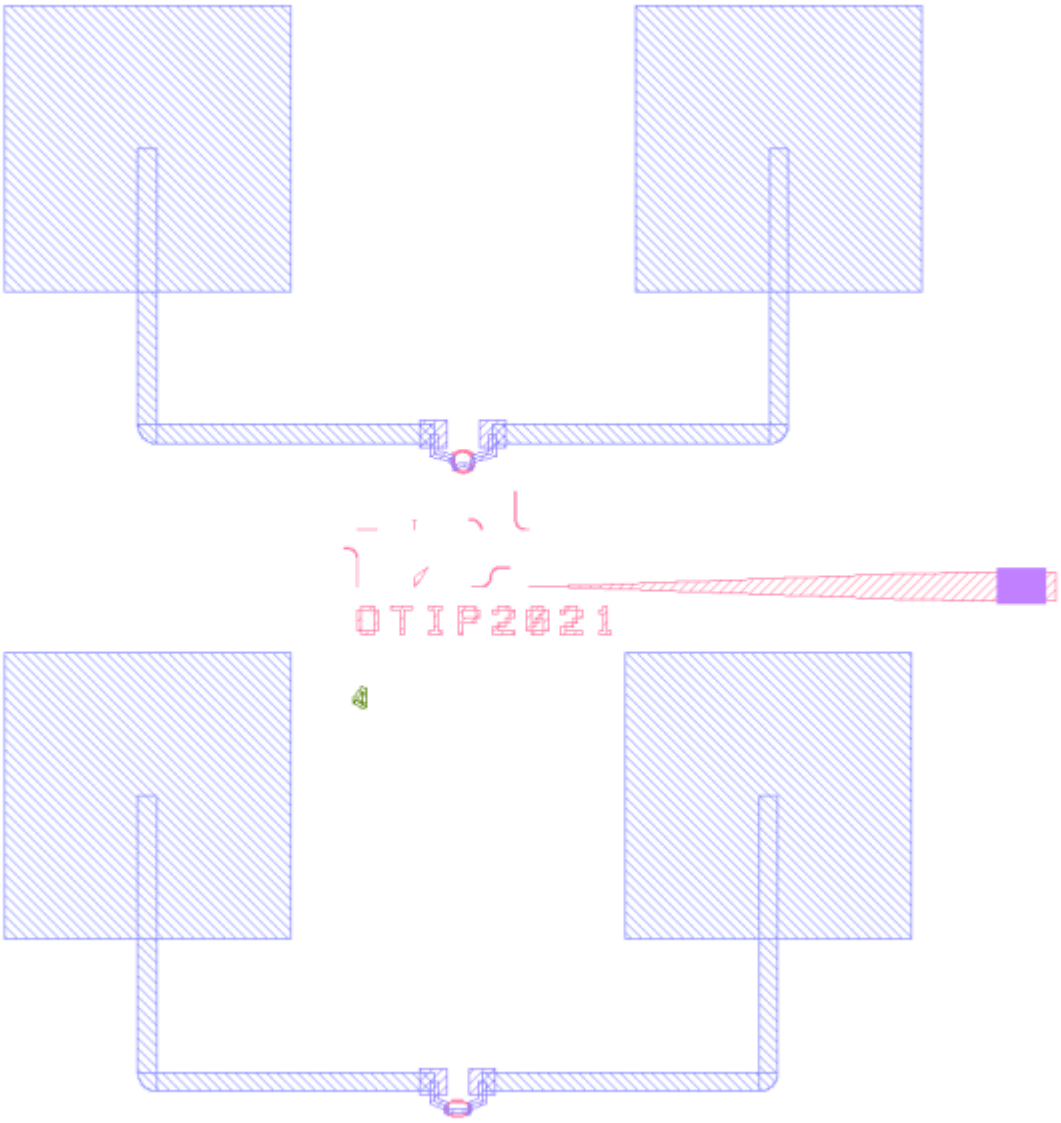


生成文件，并可从负胶版图转向正胶版图

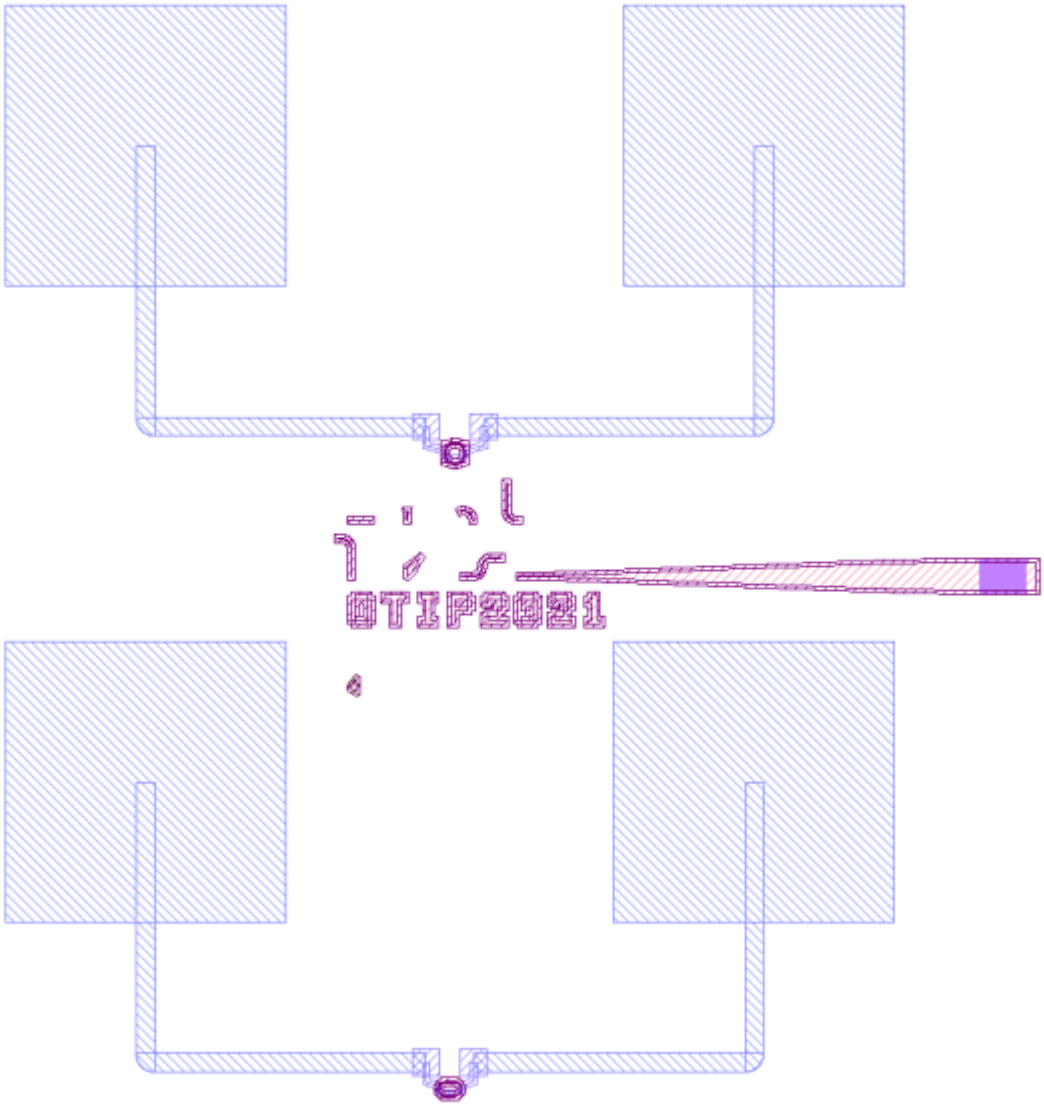
make_gdsii_file

生成 gdsii 文件，并可以指定产生相应的覆盖层 (cover layer) 或反向层 (inverse layer, 即从负胶层生成的正交层).

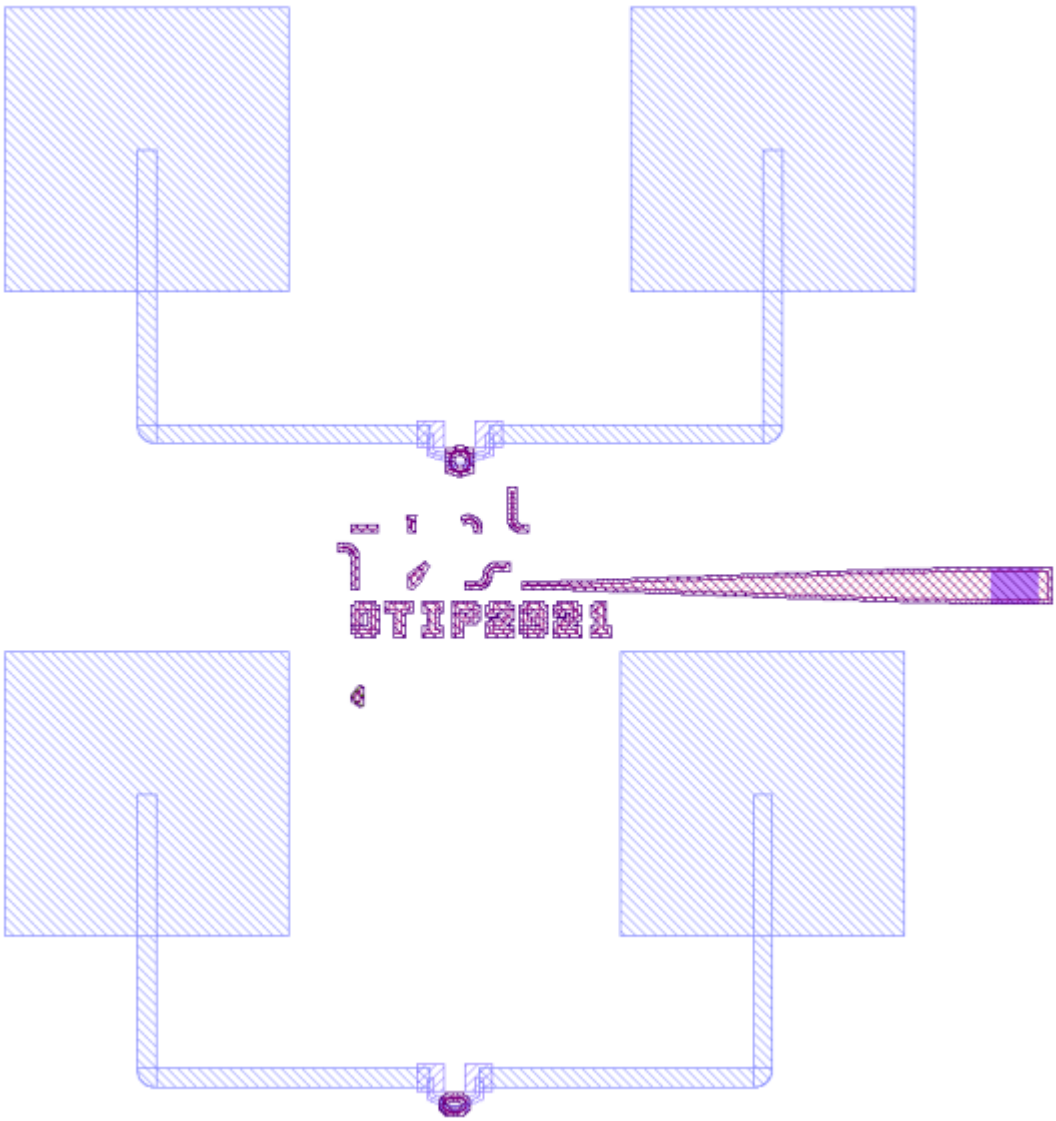
```
# create file and save
make_gdsii_file("basic.gds")
```




```
# create file and save the layout with an inverse layer
make_gdsii_file("basic_inverse.gds",inv_source_layer=wg_layer,
    inv_target_layer=inv_layer)
```



```
# create file and save the layout with inverse layer and cover
    layer
make_gdsii_file("basic_inverse_and_cover.gds",inv_source_layer=
    wg_layer,inv_target_layer=inv_layer,cover_source_layer=
    wg_layer,cover_target_layer=cover_layer)
```



5 具体设计例子

以下是两根根据 AEMD 设计规则和 CUMEC 设计规则实际绘制版图的脚本。

AEMD design

CUMEC design （其中的 CUMEC 光栅需要自行取得 sdk 的“GC_TE_1550.gds”放入同一文件夹）