PIC Palette（矽绘） —— 硅光芯片自动化设计和绘版

PIC Palette —— Design and Layout Automation of Silicon Photonics

吴 昊
旭创研究院—苏州旭创科技有限公司

1. 硅光芯片设计软件

半导体物理

求解有源器件中的光-电-物质相互作用问题

Sentaurus TCAD

SILVACO

SYNOPSYS

COMSOL MULTIPHYSICS

光学

求解光在硅光器件/芯片中的传输问题

Ansys lumerical

Max-Optics 曼光

SYNOPSYS Silicon to Software

RSOFT SYNOPSYS

Photon Design

紧凑模型

Advanced Design System
Premier High-Frequency and High-Speed Design Platform

VPIphotonics DESIGN AUTOMATION

硅光

PIC Palette 矽绘

LUCEDA PHOTONICS

逍遥科技 LATITUDE DESIGN AUTOMATION

Ansys ELECTRONICS DESKTOP

CST STUDIO SUITE
ELECTROMAGNETIC FIELD SIMULATION SOFTWARE

求解有源器件中的微波传输问题

微波传输

cadence

PhoeniX Software
Solutions for Micro and Nano Technologies

Tanner EDA

Mentor Graphics Calibre

Empyrean 华大九天

KLayout

硅光芯片版图的检查与输出

微电子

GitHub

2. PIC Palette介绍——Roadmap

PIC Palette是一款针对硅基光子芯片及其它光电芯片特点打造的集仿真、绘版、检查和测试的一体化设计软件。

已就绪　研发中　敬请期待

lumerical

Device Simu
Schematic Simu
pysimu1

Inline Check
Offline Check
pydrc

数据可视化和数据库管理
测试-仿真数据关联
pymeas

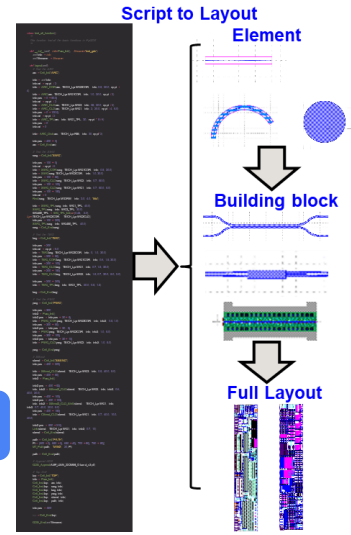
PIC Palette 矽 绘

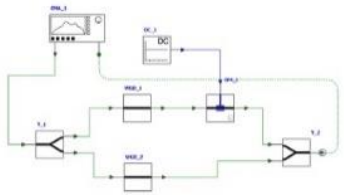
Powered by python

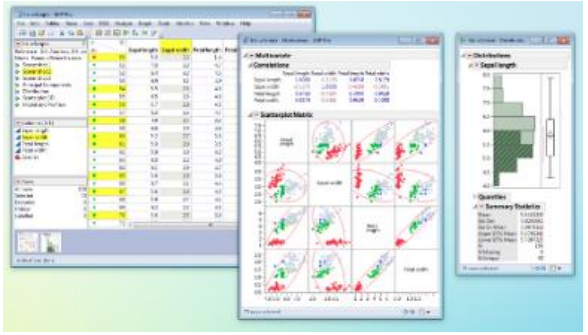
pydgs2
基本图案绘制
图形操作运算
BB Library
Auto Routing

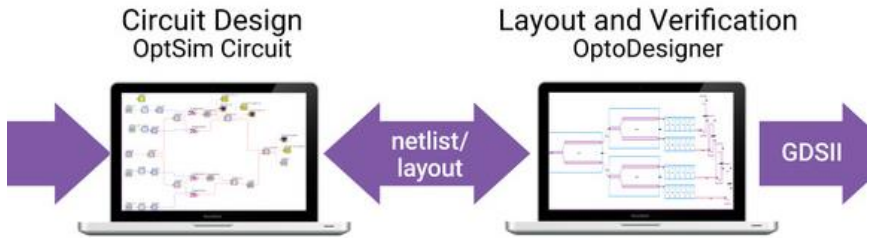
pypdk
主流Foundry PDK 支持
主客户自建PDK

pyschm
LVS：Layout Vers us Schematic Check
SDL：Schematic Driven Layout









3. PIC Palette介绍——软件特点

PIC Palette 矽 绘

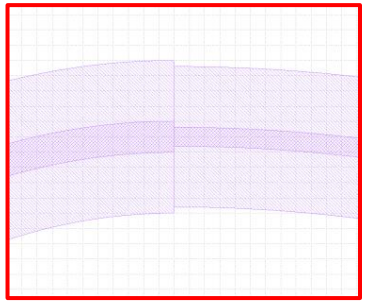
1. 绘图逻辑特点	A. **集成专为光子芯片作图**函数，方便快捷； B. 绘图语句简单，层级明确。
2. 自动化加成	A. 端口自动**对齐**，端口自动**生成**，端口自动删除； B. 基础图形**Warning功能**； C. **自动化参数计算**和显示，端口提取； D. FDTD**互联仿真**；
3. PDK 支持	A. 兼容**主流Foundry** PDK （GDSII/端口）； B. 支持**自建PDK**（GDSII、端口、FDTD模型）； C. 不同PDK之间**快速切换**；
4. PIC Palette独占	A. 方便计算变换的坐标系统； B. 生成Layout**效率高**； C. 快速上手，**学习成本低**；

目标：用最简单的语句画出你想要的图形

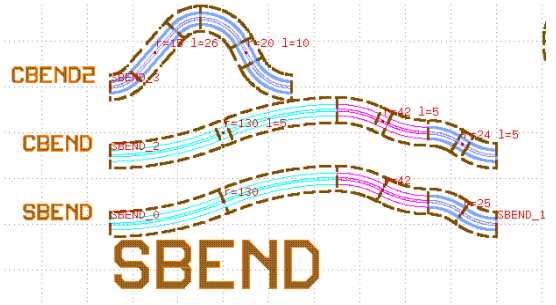
针对硅光芯片Layout特点开发的三十多个函数

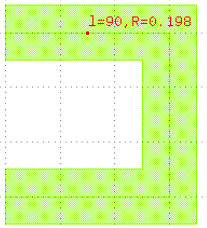
功能	函数
基础绘图	polygon/path/tag
定制化绘图	swg(swg_cld)/arc(arc_cld)/twg(twg_cld)/pwg(pwg_cld)/sbend/cbend/bezierbend/mt_path...
操作	aref/insert/boolean/flatten/port operation...

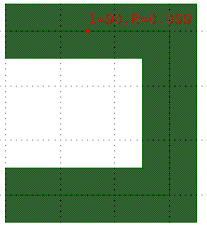
针对硅光芯片Layout特点开发的十几种错误警告



一些便捷的计算小工具（半径计算、电阻计算）







4. PIC Palette介绍——发展历程

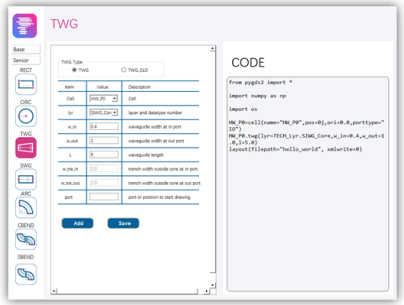
2021.03 alpha版本发布

2021.06 beta版本发布

2022.01 V1版本发布

2022.03 Device Simulation

2022.06 GUI 代码部署



2022.Q3 正式发布

2022.Q4 pysimu1/pypdk 功能完善

2023~2024 更多功能开发。。。

经历5轮TO验证

3. PIC Palette介绍

PIC Palette 矽绘

PIC Palette设计软件包含六大模块：pygds2、pysimu、pypdk、pydrc、pyschm、pymeas。

1.pygds2

基于python和自定义程序的GDSII文件生成软件，包含硅光器件中常用的波导、弯曲、锥形等结构，包含自动端口连接功能；仅需掌握简单python编程语句即可掌握；包含GUI指导快速上手。

2. pysimu

基于pygds2与主流硅光子器件仿真软件互联，实现基于Layout物理结构快速建模和获取仿真结果。

3. pypdk

基于pygds2建立客户的PDK库，并实现多foundry兼容，Layout快速切换；开发算法协助客户导入主流Foundry PDK。

4. pydrc

基于pygds2的inline check，结合硅光芯片特点在生成layout的过程中进行实时检测；对已经生成或其他方法产生GDSII的数据，根据一定规则进行offline检测。

5. pyschm

对Layout进行Schematic对应检查（LVS）；根据Schematic生成Layout (SDL)

6. pymeas

测试结果、仿真结果和Layout关联，PDK数据管理；研发和量产数据可视化和数据库管理。

3. PIC Palette介绍

PIC Palette现在推出免费使用版pygds和专业订阅版pygds2，免费版和订阅版功能区分如下表。免费版和专业版在license有效期内都可以或得软件持续更新服务。

免费版 pygds1

有限
技术支持

- 基础Layout功能
- Label和Tag标签生成功能
- 基础硅光子器件图形(SWG/ARC/TWG等)
- 金属走线功能
- 自动端口对准
- 外部GDSII格式导入功能
- 基础图形运算功能(XOR/UNION等)

专业订阅版 pygds2

- 高级波导Routing
- 高级金属Routing
- 高级Building block Library
- 高级图形和Layout操作工具

1. 工作日24小时技术支持
2. TO节点在线快速响应
3. 定制化功能开发
4. 授权包含pysimu/pydrc/pypdk模块相关功能使用权限

永久免费

提供3~6个月试用申请，可申请延期一次

优秀国产开源/商业硅光设计软件介绍

- OSIM——香港科技大学
  （https://eexu.home.ece.ust.hk/BOSIM.html ）

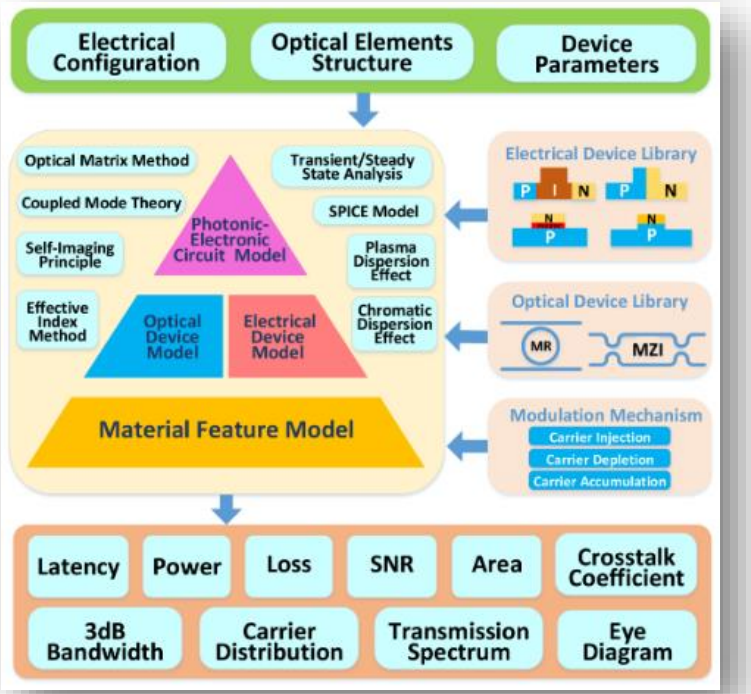
- 智能光传输仿真平台——上海交通大学
  （https://ifibertrans.sjtu.edu.cn/ ）

- 曼光信息科技
  （https://max-optics.com/）

- 华大九天
  （https://www.empyrean.com.cn/）

- 逍遥科技
  （https://www.latitudeda.com/）

希望大家多多支持国产软件，欢迎多提建议！



PIC Palette 软件功能介绍

用最简单的语句画出你想要的图形

目 录

CONTENTS

1. **GDSII文件的组成；**

2. **第一次使用PIC Palette画图；**
   - 开始画图：新建一个Cell
   - 画图过程：你可以做哪些操作
   - 画图过程：基础绘图
   - 画图过程：硅光绘图1——波导类函数
   - 画图过程：硅光绘图2——其他图形
   - 画图过程：图形操作——引用
   - 画图过程：图形操作——其他
   - 图形重要属性——PORT
   - 结束绘图

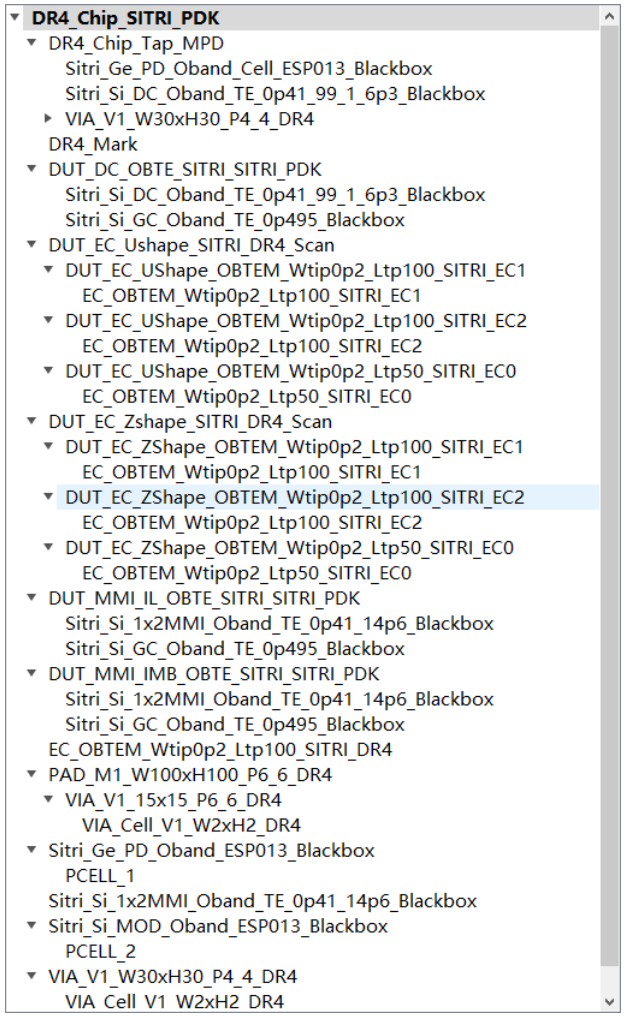
3. **PDK，联合仿真；**

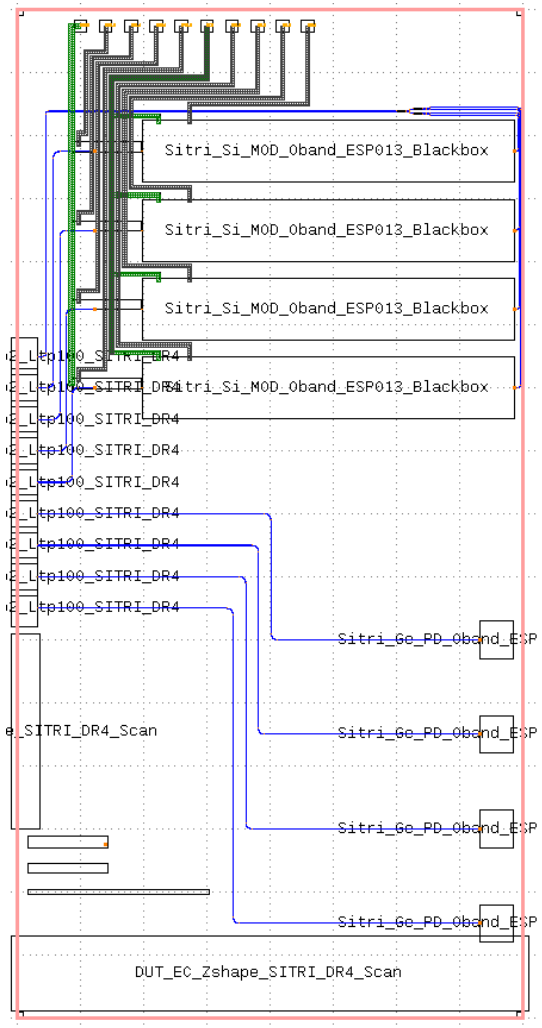
4. **案例展示。**

GDSII文件的组成

All About Calma's GDSII Stream File Format (artwork.com) https://www.artwork.com/gdsii/gdsii/

1. GDSII 由一个或者数个Cell（Instance）组成；
2. 每个Cell由如下基础元素组成：
   A. Polyon
   B. Path
   C. Tag
   D. Reference (ARef/SRef)
3. Cell之间有相互引用关系（减少重复单元造成体积，方便修改）；





开始画图：新建一个Cell

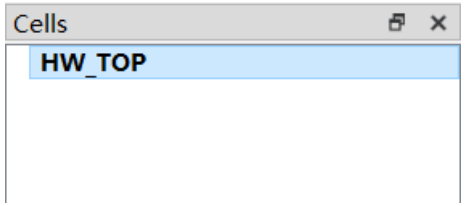
1. **AA = cell(cellname)，定义一个cell对象；**
2. **以cell为对象完成所有绘图操作；**
   - 格式: AA.xxx (xxx是cell对象内置的操作函数)
3. **不同cell之间的操作独立切互不干涉；**
4. **防止Cell重名，可以自动加入前缀**

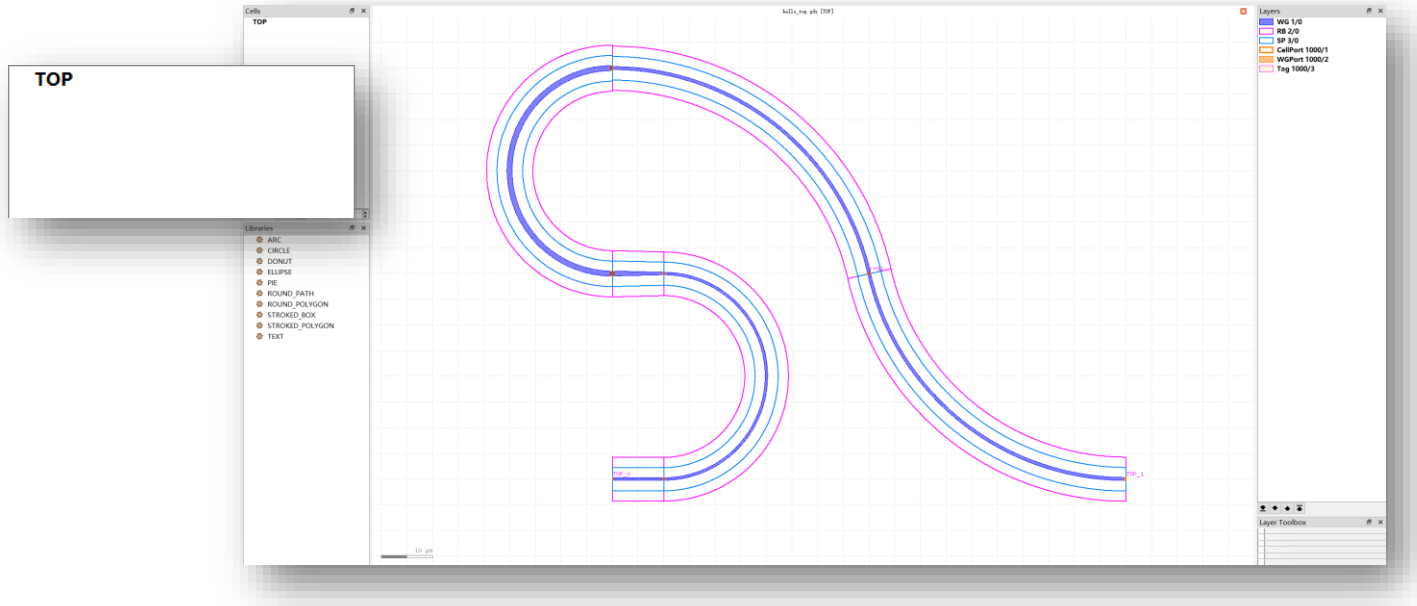
示例：ex_pygds2/top.py

```
top = cell("TOP")

top.swg_cld(lyr=TECH_Lyr.WG2, w=0.5, l=10)
top.arc_cld(lyr=TECH_Lyr.WG2, w=0.5, r=20, ang=np.pi)
top.twg_cld(lyr=TECH_Lyr.WG2, w_in=0.5, w_out=1.0, l=10)
top.arc_cld(lyr=TECH_Lyr.WG2, w=1.0, r=20, ang=-np.pi)
top.sbend_cld(lyr=TECH_Lyr.WG2, w=0.6, dx=100, dy=-80)

layout('hello_top', xmlwrite=0)
```

tech/technology.py

```
"""
Created on 2022/7/23 13:08
@author: HW
"""

Owner = ''  # Owner shown before cell name automatically. '': add nothing
```





画图过程：你可以做哪些操作

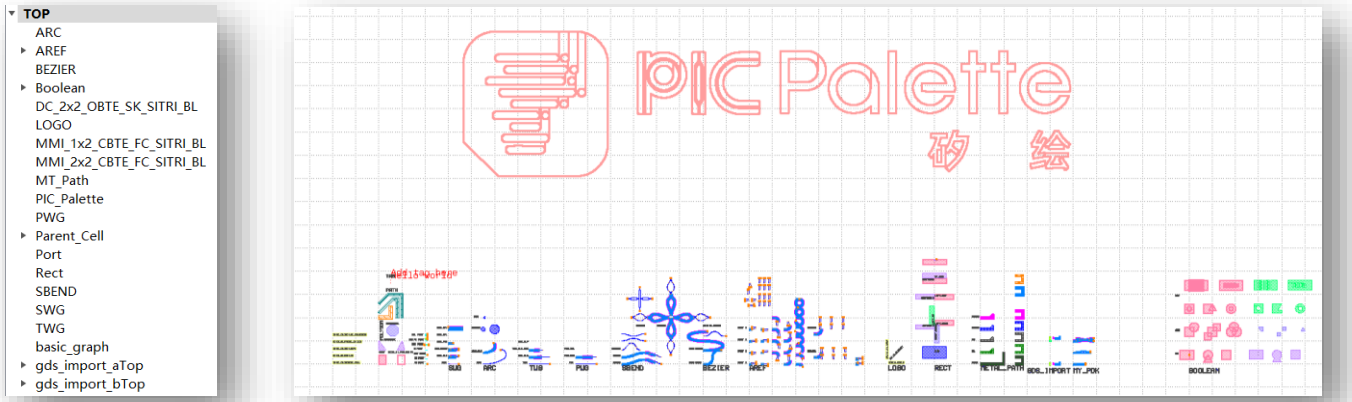
PIC Palette 矽 绘

1. 通过AA = cell(cellname)定义一个cell对象后，您可以做如下的操作...（格式: AA.xxx() ）

功能	函数
基础绘图	polygon/path/tag
硅光绘图	swg/arc/twg/pwg/sbend/cbend/ubend/zbend/bezierbend/mt_path/rect/logo/...
图形操作	aref/insert/boolean/flatten/port operation/...

2. 同时cell有如下对象可以调用:（格式: AA.yyy ）

name	名字
portS/portA	可用端口/激活端口
child	cell对象的子Cell



示例：ex_pygds2/cell_func_test.py

```
from pygds2 import*
from func_test import*
from pypdk import*

iffaltten = 0

top = cell('\\TOP', porttype='N')

top.aref(cell_test(), pos=0, portAref=0)  # This function test adding a cell
top.aref(graph_test(), pos=200, portAref=0)  # This function test the basic graphs including polygon, path and tag

if iffaltten:
    top.flatten()
top.aref(port_test(), pos=400, portAref=0)
top.aref(swg_test(), pos=500, portAref=0)
top.aref(arc_test(), pos=650, portAref=0)
top.aref(twg_test(), pos=850, portAref=0)
top.aref(pwg_test(), pos=1050, portAref=0)
top.aref(sbend_test(), pos=1250, portAref=0)
top.aref(bezier_test(), pos=1600, portAref=0)
top.aref(aref_test(), pos=1800, portAref=0)
top.aref(logo_test(), pos=2400, portAref=0)
top.aref(rect_test(), pos=2600, portAref=0)
top.aref(mt_path_test(), pos=2800, portAref=0)

gds = gds_import('func_test/gds_import_test.GDS')
top.aref(gds, pos=3100)
top.aref('gds_import_aTop', pos=3100 + 100j)

inno = gds_import('func_test/logo.GDS')
top.aref(inno, pos=-1000 + 300j, portAref=0)

# Test Own cell
try:
    from mypdk import MMI1x2, MMI2x2, DC2x2

    mmi_cell1 = MMI1x2(label='BL', band='C', mode='TE', type='FC',
                       l_IO=10, w_taper=1.5, w_MMI=5, w_MMI_tre=2, l_MMI=20, s_MMI=2.5,
                       w_IO=0.6, lyr=TECH_Lyr.WG1)
    top.addport(pos=3200, type='IO', npp=0)
    top.swg_cld(lyr=TECH_Lyr.WG1, w=0.6, l=10)
    top.aref(cell_child=mmi_cell1)

    mmi_cell2 = MMI2x2(label='BL', band='C', mode='TE', type='FC',
                       l_IO=20, w_taper=1.5, w_MMI=5, w_MMI_tre=2, l_MMI=20, s_MMI=2.5,
                       w_IO=0.6, lyr=TECH_Lyr.WG1)
    top.addport(pos=3200 + 50j, type='IO', npp=0)
    top.swg_cld(lyr=TECH_Lyr.WG1, w=0.6, l=10)
    top.aref(cell_child=mmi_cell2)

    dc_cell = DC2x2(label='BL', band='O', mode='TE', type='SK',
                    w_dc=0.5, g_dc=0.2, l_dc=10, s_IO=10,
                    w_IO=0.6, lyr=TECH_Lyr.WG2)
    top.addport(pos=3200 + 100j, type='IO', npp=0)
    top.swg_cld(lyr=TECH_Lyr.WG2, w=0.6, l=10)
    top.aref(cell_child=dc_cell)
except Exception as err:
    pass

top.logo(text='My_PDK', pos=3200 - 40j, w=15, ifnofill=0)
top.logo(text='GDS_IMPORT', pos=3000 - 40j, w=15, ifnofill=0)

try:
    top.aref(cell_child=gds_import_test(), pos=3400, portAref=1)
except Exception as e:
    print(e)
    pass

top.aref(cell_child=xor_test(), pos=3700 + 40j, portAref=0)

# Write GDS
filepath = 'hello_world'
layout(filepath, xmlwrite=0, timelogo=0)
```

画图过程：基础绘图

AA = cell(cellname)

AA.xxx()

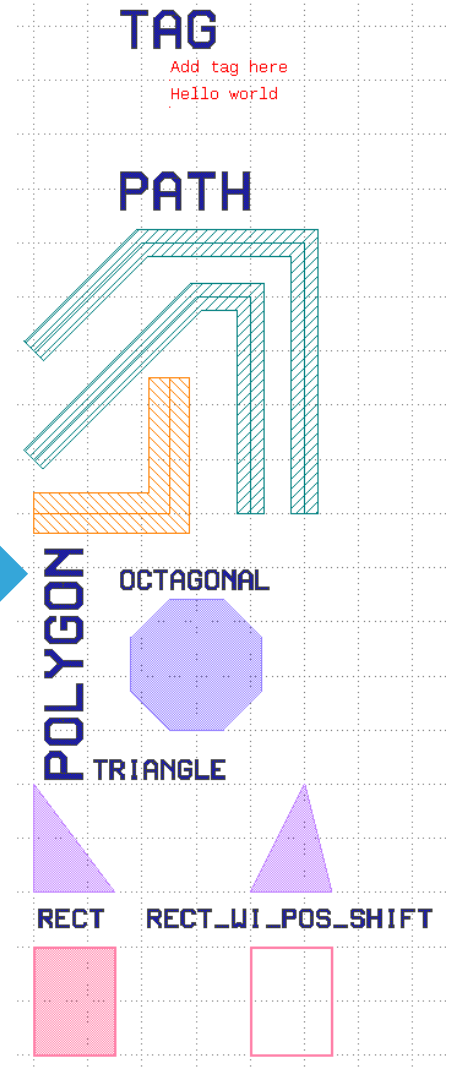
xxx()可选项：
1. polygon 绘制封闭的图案
示例: AA.polygon(lyr=[1, 0], pi=[0, 40, 40 + 30j, 30j], pos=0)
2. path 绘制一条有宽度的线段
示例: AA.path(lyr=[1, 0], pi=[50j, 100 + 50j], w=10, pathtype=0)
3. tag 增加一个标签
示例: AA.tag(self, lyr=[0, 0], pos=100 + 100j, context='Hello_World')

Tips:

1. **lyr=[1,0] list表示图层对应的layer和datatype；**
2. **lyr=[[1,0],[2,0]]包含多层时，会在所有图层都绘制；**
3. **X+Yj表示（X, Y）坐标点，方便平移、旋转操作。**

示例：ex_pygds2/func_test/test_graph.py

```python
from pygds2 import *
import numpy as np


def graph_test():
    """
    This function test the basic graphs including polygon, path and tag
    :return: a parent cell
    """

    graph = cell('\\basic_graph')

    # polygon
    graph.polygon(lyr=[1, 1], pi=[0, 30, 30 + 40j, 40j])
    graph.logo(text='Rect', pos=45j, ifnofill=0)
    graph.polygon(lyr=[1, 2], pi=[0, 30, 30 + 40j, 40j], pos=80)
    graph.logo(text='Rect_wi_pos_shift', pos=40 + 45j, ifnofill=0)

    graph.polygon(lyr=[2, 1], pi=[[0, 30, 40j], [80, 110, 100 + 40j]], pos=60j)
    graph.logo(text='Triangle', pos=100j + 20, ifnofill=0)
    graph.polygon(lyr=[3, 1], pi=[0, 20, 20 + 10 * (2 ** 0.5) * (1 + 1j), 20 + 10 * (2 ** 0.5) * (1 + 1j) + 20j,
                                  20 + 10 * (2 ** 0.5) * (1j + 1j) + 20j, 0 + 10 * (2 ** 0.5) * (1j + 1j) + 20j,
                                  0 + 10 * (2 ** 0.5) * (-1 + 1j) + 20j, 0 + 10 * (2 ** 0.5) * (-1 + 1j)],
                  pos=50 + 120j)
    graph.logo(text='Octagonal', pos=30 + 170j, ifnofill=0)

    graph.logo(text='polygon', pos=20 + 100j, ori=np.pi / 2, w=10, ifnofill=0)
    graph.path(lyr=[641, 1], pi=[200j, 50 + 200j, 50 + 250j], w=15)
    graph.path(lyr=[642, 1], pi=[[80 + 200j, 80 + 280j, 60 + 280j, 220j], [100 + 200j, 100 + 300j, 40 + 300j, 260j]],
               w=10)
    graph.logo(text='Path', pos=30 + 310j, w=10, ifnofill=0)

    graph.tag(lyr=[15, 0], pos=50 + 350j, context='Hello world')
    graph.tag(lyr=[15, 0], pos=50 + 360j, context='Add tag here')
    graph.logo(text='Tag', pos=30 + 370j, w=10, ifnofill=0)

    return graph


if __name__ == '__main__':
    tmp = graph_test()
    layout('gdsii\\' + tmp.name)
```



画图过程：硅光绘图1——波导类函数

AA = cell(cellname)

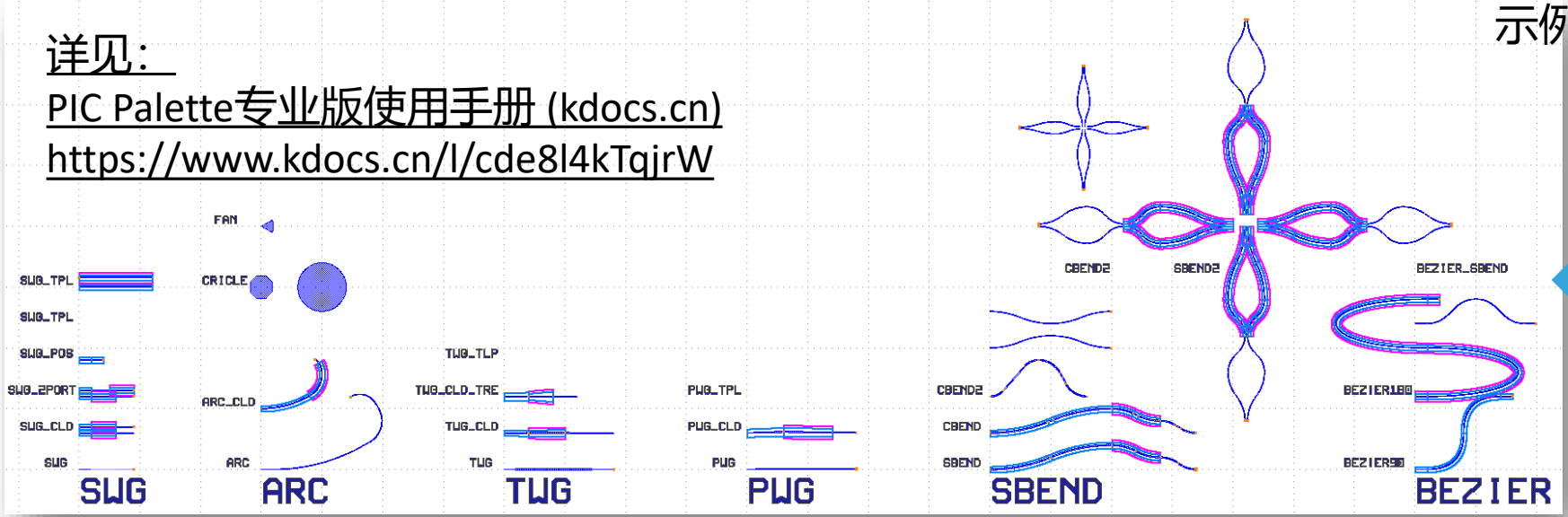
AA.xxx()

xxx()可选项：
1. swg/swg_cld绘制一段直波导；示例: AA.swg(lyr=[1, 0], w=0.5, l=10, port='0') / AA.swg_cld(lyr=[[1, 0], [1, 1]], w=0.45, l=10, w_tre=-1, port='0')
2. arc/arc_cld绘制一段弯曲波导；示例: AA.arc(lyr=[1, 0], w=0.45, r=15, ang=np.pi / 4, port='0')/AA.arc_cld(lyr=[[1, 0], [1, 1]], w=0.45, r=15, ang=np.pi / 4, w_tre=-1, port='0')
3. twg/twg_cld绘制一段锥形波导；示例: AA.twg(lyr=[1, 0], w_in=0.4, w_out=1.0, l=10.0, port='0')/AA. twg_cld(lyr=[[1, 0], [1, 1]], w_in=0.4, w_out=1.0, w_tre_in=-1, w_tre_out=-1, l=10.0, port='0')
4. pwg/pwg_cld绘制一段楔形波导；示例: AA.pwg(lyr=[1, 0], w_in=0.4, w_out=1.0, biasw=0.2, l=10.0, port='0')/AA.pwg_cld(lyr=[[1, 0], [1, 1]], w_in=0.4, w_out=1.0, biasw=0.2, w_tre_in=-1, w_tre_out=-1, l=10.0, port='0')
5. sbend/sbend_cld绘制一段S-波导；示例: AA.sbend(lyr=[1, 0], w=0.45, dx=30, dy=5, port='0')/AA.sbend_cld(lyr=[[1, 0], [1, 1]], w=0.45, w_tre=-1, dx=30, dy=5, port='0')
6. sbend2/sbend2_cld绘制2段对称的s-波导；示例： AA.sbend2(lyr=[[1, 0], [1, 1]], w=0.45, s_out=30, dx=100, port=[0, 1])/sbend2_cld(lyr=[[1, 0], [1, 1]], w=0.45, w_tre=-1, s_out=30, dx=100, port=[0, 1])
cbend/cbend_cld/ubend_cld/zbend_cld/beziersbend/beziersbend_cld/bezier180/bezier180_cld/bezier90_cld/bezier90/bezier180_dw/bezier180_cld_dw

详见：

PIC Palette专业版使用手册 (kdocs.cn)

https://www.kdocs.cn/l/cde8l4kTqjrW



示例：ex_pygds2/cell_func_test.py

```
top = cell('\\TOP', porttype='N')

top.aref(cell_test(), pos=0, portAref=0)
top.aref(graph_test(), pos=200, portAref=0)

if iffaltten:
    top.flatten()
top.aref(port_test(), pos=400, portAref=0)
top.aref(swg_test(), pos=500, portAref=0)
top.aref(arc_test(), pos=650, portAref=0)
top.aref(twg_test(), pos=850, portAref=0)
top.aref(pwg_test(), pos=1050, portAref=0)
top.aref(sbend_test(), pos=1250, portAref=0)
top.aref(bezier_test(), pos=1600, portAref=0)
```

画图过程：硅光绘图2——其他图形

AA = cell(cellname)

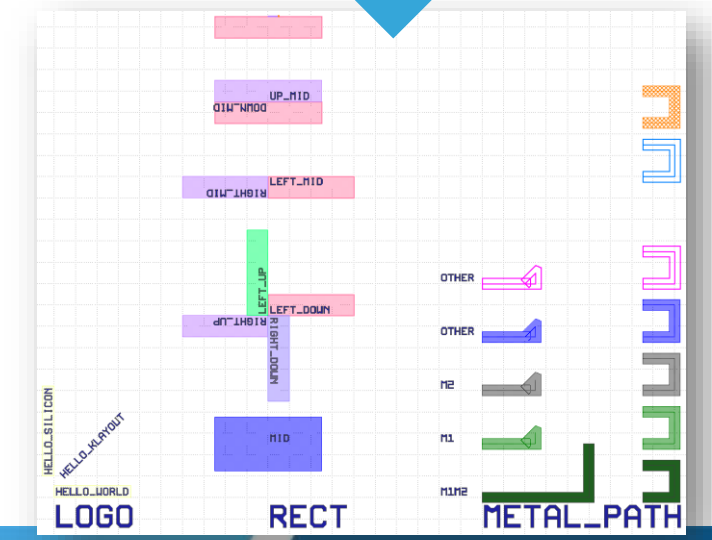
AA.xxx()

xxx()可选项：
1. logo 将输入字符转换为logo画到mask上
示例: AA.logo(lyr=TECH_Lyr.LOGO, text='Hello_World', w=5, pos='NA', ori=0, port='0', rpos='NA', portIn='NA', portOut='NA', ifnofill=1)
2. rect 绘制一个矩形
示例: AA.rect(lyr=[1, 0], w=100, h=50, port='0', pos='mid', ori=0, portAref=1)
3. mt_path/mt_path90 绘制一组金属走线（任意角度/90°）
示例: AA.mt_path(lyr='M1M2', pi=[0 + 0j, 100 + 0j, 100 + 50j], w=10, pathtype=0, pos=0) AA.mt_path90(lyr='M1M2', pi=[0 + 0j, 50, 90j], w=10, pathtype=0, pos=0, label='NA')

PIC Palette 矽 绘

示例：ex_pygds2/cell_func_test.py

```
top = cell('\\TOP', porttype='N')

top.aref(cell_test(), pos=0, portAref=0)  # This function test adding a cell
top.aref(graph_test(), pos=200, portAref=0)  # This function test the basic graphs including polygon, path and tag

if iffalltten:
    top.flatten()
top.aref(port_test(), pos=400, portAref=0)
top.aref(swg_test(), pos=500, portAref=0)
top.aref(arc_test(), pos=650, portAref=0)
top.aref(twg_test(), pos=850, portAref=0)
top.aref(pwg_test(), pos=1050, portAref=0)
top.aref(sbend_test(), pos=1250, portAref=0)
top.aref(bezier_test(), pos=1600, portAref=0)
top.aref(aref_test(), pos=1800, portAref=0)
top.aref(logo_test(), pos=2400, portAref=0)
top.aref(rect_test(), pos=2600, portAref=0)
top.aref(mt_path_test(), pos=2800, portAref=0)
```



画图过程：图形操作——引用

PIC Palette 矽 绘

示例：ex_pygds2/func_test/test_aref.py

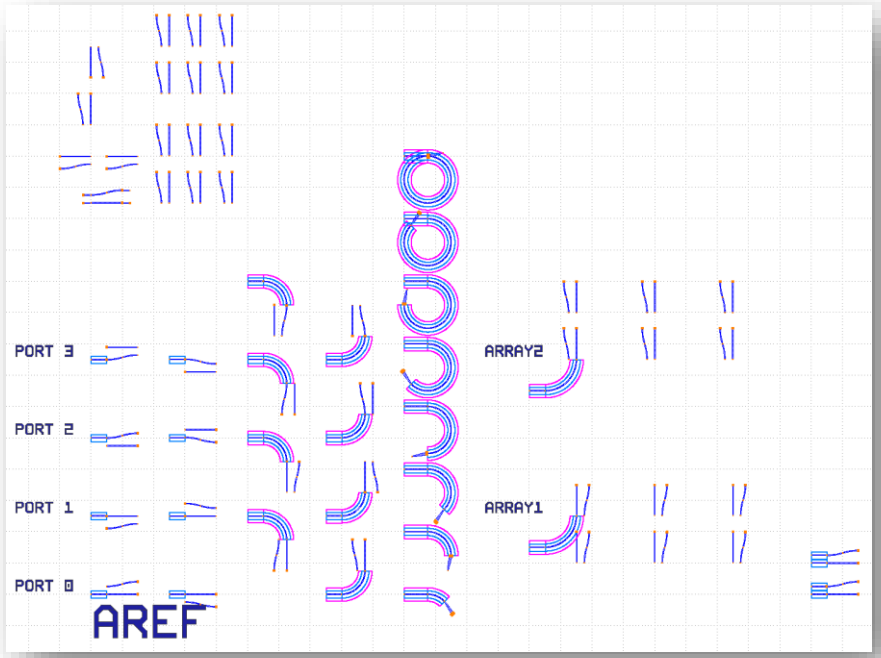
AA = cell(cellname)

BB = cell(childname)

引用用法
AA.aref(cell_child=BB, port_parent='0', port_child=0, portAref=1, portArr=0, portDel=0, portlabel='-1', pos='NA', mirror='NA', ori='NA', Nxy=[1, 1], dxy=[0, 0], port_in='NA', mag=1)
BB.insert(cell_parent=AA, port_parent='0', port_child=0, portAref=1, portArr=0, portDel=0, portlabel='-1', pos='NA', mirror='NA', ang='NA', Nxy=[1, 1], dxy=[0, 0], port_in='NA')

一些特点：

1. **可以自动将AA和BB的端口对齐；**
2. **由于引用BB，在AA中会自动产生新的端口。**




```
def aref_test():
    """
    This function test the aref and insert
    :return:
    """
    pitch = 50

    aref_cell = cell('\\AREF', porttype='IO')
    child1 = cell('child1')
    child1.portA[0].npp = 0
    child1.swg_cld(lyr=TECH_Lyr.WG3, w=0.5, r=20)
    child1.addport(pos=5j, npp=0, type='IO')
    child1.sbend_cld(lyr=TECH_Lyr.WG3, w=0.6, dx=20, dy=3)

    # 4 port test
    for i in range(4):
        aref_cell.addport(pos=pitch * i, type='IO', npp=0)
        aref_cell.swg_cld(lyr=TECH_Lyr.WG1, w=0.6, r=10)
        # aref_cell.arc_cld(lyr=TECH_Lyr.WG2, w=0.6, r=20, ori=np.pi / 6)
        aref_cell.portA[0].mirror = 0
        aref_cell.aref(cell_child=child1, port_child=i, mirror=0)
        aref_cell.logo(text='port {}'.format(i), pos=-50 + pitch * i, ifnofill=0)

    for i in range(4):
        aref_cell.addport(pos=50 + pitch * i, type='IO', npp=0)
        aref_cell.swg_cld(lyr=TECH_Lyr.WG1, w=0.5, r=10)
        # aref_cell.arc_cld(lyr=TECH_Lyr.WG2, w=0.6, r=20, ori=-np.pi / 6)
        aref_cell.portA[0].mirror = 1
        aref_cell.aref(cell_child=child1, port_child=i, mirror=1)

    for i in range(4):
        aref_cell.addport(pos=100 + pitch * (i + 1), type='IO', npp=0)
        aref_cell.swg_cld(lyr=TECH_Lyr.WG2, w=0.5, r=10)
        aref_cell.arc_cld(lyr=TECH_Lyr.WG2, w=0.5, r=15, ang=-np.pi / 2)
        aref_cell.portA[0].mirror = 0
        aref_cell.aref(cell_child=child1, port_child=i, mirror=1)

    for i in range(4):
        aref_cell.addport(pos=150 + pitch * i, type='IO', npp=0)
        aref_cell.swg_cld(lyr=TECH_Lyr.WG2, w=0.6, r=10)
        aref_cell.arc_cld(lyr=TECH_Lyr.WG2, w=0.6, r=15, ang=np.pi / 2)
        aref_cell.portA[0].mirror = 1
        aref_cell.aref(cell_child=child1, port_child=i, mirror=0)

    child2 = cell('child2', owner='FS', porttype='IO')
    child2.pwg_cld(lyr=TECH_Lyr.WG3, w_in=0.6, w_out=2.0, biasw=-2.0, r=10)

    # 2 structure port test
    for i in range(8):
        aref_cell.addport(pos=200 + 40j * i, type='IO', npp=0)
        aref_cell.swg_cld(lyr=TECH_Lyr.WG2, w=0.6, r=15)
        aref_cell.arc_cld(lyr=TECH_Lyr.WG2, w=0.6, r=15, ang=-2 * (i + 1) / 2 * np.pi / 4)
        aref_cell.portA[0].mirror = int(i / 4)
        aref_cell.aref(cell_child=child2, port_child=int(i % 2), mirror=int((i / 2) % 2))

    # port arr test
    for i in range(2):
        aref_cell.addport(pos=280 + 50j * (2 * i - 1) + 80j, type='IO', npp=0)
        aref_cell.swg_cld(lyr=TECH_Lyr.WG2, w=0.6, r=10)
        aref_cell.arc_cld(lyr=TECH_Lyr.WG2, w=0.6, r=20, ang=np.pi / 2)
        aref_cell.portA[0].Nxy = [3, 2]
        aref_cell.portA[0].dxy = [60, 30 * (2 * i - 1)]
        aref_cell.aref(cell_child=child1, port_child=0, mirror=1 - i, portArr=(i == 0))
    aref_cell.logo(text='Array1'.format(i), pos=250 + 50j, ifnofill=0)
    aref_cell.logo(text='Array2'.format(i), pos=250 + 150j, ifnofill=0)

    N = len(aref_cell.portS)
    # port aref del test
    aref_cell.addport(pos=460, type='IO', npp=0)
    aref_cell.addport(pos=460 + 5j, type='IO', npp=0)
    aref_cell.swg_cld(lyr=TECH_Lyr.WG1, w=0.6, r=10, port=[N + 1, N + 3])
    aref_cell.aref(cell_child=child1, port_child=0, port_parent=N + 1, mirror=0, portDel=0)

    N = len(aref_cell.portS)
    aref_cell.addport(pos=460 + 20j, type='IO', npp=0)
    aref_cell.addport(pos=460 + 25j, type='IO', npp=0)
    aref_cell.swg_cld(lyr=TECH_Lyr.WG1, w=0.6, r=10, port=[N + 1, N + 3])
    aref_cell.aref(cell_child=child1, port_child=0, port_parent=N + 1, mirror=0, portDel=1)

    # Reference with positions
    aref_cell.addport(pos=250j, ori=0, type='IO')
    aref_cell.aref(cell_child=child1, port_child=0, pos=250j)
    aref_cell.actport([120, 121])
    aref_cell.swg_cld(lyr=TECH_Lyr.WG3, w=0.5, r=5)
    aref_cell.actport([122, 123])
    aref_cell.swg_cld(lyr=TECH_Lyr.WG3, w=0.6, r=5)

    aref_cell.actport([123])
    aref_cell.aref(cell_child=child1, port_child=0, pos=250j + 30j, ori=np.pi)
    aref_cell.aref(cell_child=child1, port_child=0, pos=250j + 50j, ori=np.pi / 2)
    aref_cell.aref(cell_child=child1, port_child=0, pos=250j + 100j, ori=np.pi / 2 * 3, portAref=0)

    aref_cell.portA[0].ori = 0
    aref_cell.aref(cell_child=child1, port_child=1, pos=250j + 30j + 10)

    aref_cell.aref(cell_child=child1, port_child=0, pos=50 + 250j, Nxy=[3, 2], dxy=[20, 30], ori=np.pi / 2, portAref=0)
    aref_cell.aref(cell_child=child1, port_child=0, pos=50 + 250j + 70j, Nxy=[3, 2], dxy=[20, 30], ori=np.pi / 2,
                   portArr=1)

    aref_cell.logo(text='Aref', w=15, pos=-30j, ifnofill=0)

    return aref_cell
```

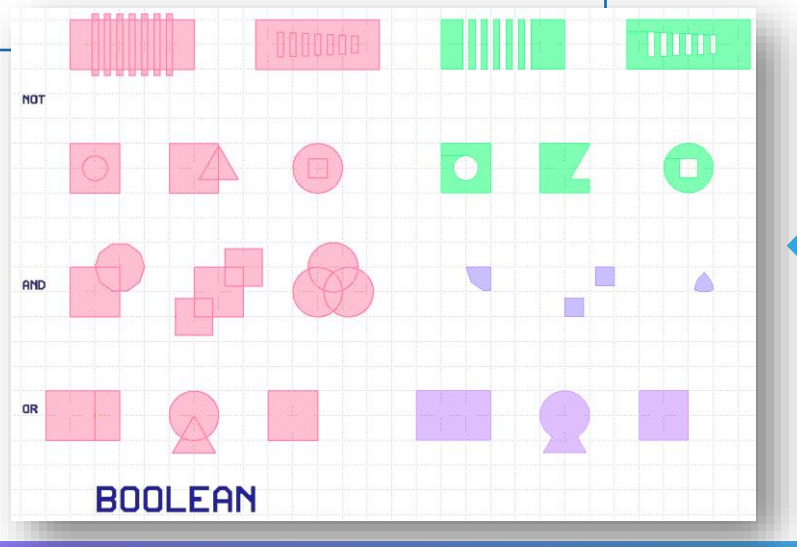
画图过程：图形操作——其他

AA = cell(cellname1)

BB = cell(cellname2)

xxx()可选项：
1. boolean 对输入的两组点阵列进行OR/AND/NOT布尔运算（pi点整列可以在cell的属性中获取）
示例: AA.boolean(lyr=[1, 1], mode='OR', pi1=[BB.pi[xx]], pi2=[BB.pi[yy]])
2. flatten
示例: AA.flatten()
3. portprint
示例: AA.portprint()

示例：ex_pygds2/func_test/test_xor.py




```
def xor_test():
    """
    This function test the XOR function
    :return: bool test cell
    """
    bool_test = cell('\\Boolean', port='N')

    # OR example
    bool_test.rect(lyr=[1, 1], w=20, h=40, pos='left_mid')
    bool_test.rect(lyr=[1, 1], w=40, h=40, pos='right_mid')
    bool_test.shiftport(pos=80)
    bool_test.circ(lyr=[1, 1], r=20, vex=40)
    bool_test.shiftport(pos=-20)
    bool_test.circ(lyr=[1, 1], r=20, vex=3)
    bool_test.shiftport(pos=80 + 20)
    bool_test.rect(lyr=[1, 1], w=40, h=40, pos='mid')
    OR = cell('\\OR', port='N')
    OR.boolean(lyr=[2, 1], mode='OR', pi1=bool_test.pi)
    bool_test.aref(cell_child=OR, pos=300, portAref=0)

    # AND example
    bool_test.addport(pos=100j, type='N')
    bool_test.rect(lyr=[1, 1], w=40, h=40, pos='mid')
    bool_test.shiftport(pos=20 + 20j)
    bool_test.circ(lyr=[1, 1], r=20, vex=10)
    N = len(bool_test.pi) - 1
    AND = cell('\\AND', port='N')
    AND.boolean(lyr=[3, 1], mode='AND', pi1=bool_test.pi[N], pi2=[bool_test.pi[N - 1]])

    bool_test.shiftport(pos=80 - 20j)
    bool_test.rect(lyr=[1, 1], w=40, h=40, pos='mid')
    bool_test.shiftport(pos=20 + 20j)
    bool_test.rect(lyr=[1, 1], w=30, h=30, pos='mid')
    bool_test.shiftport(pos=-40 - 40j)
    bool_test.rect(lyr=[1, 1], w=30, h=30, pos='mid')
    N = len(bool_test.pi) - 1
    AND.boolean(lyr=[3, 1], mode='AND', pi1=bool_test.pi[N - 2], pi2=[bool_test.pi[N - 1], bool_test.pi[N]])

    bool_test.shiftport(pos=80 + 20 + 20j)
    bool_test.circ(lyr=[1, 1], r=20, vex=40)
    bool_test.shiftport(pos=25)
    bool_test.circ(lyr=[1, 1], r=20, vex=40)
    bool_test.shiftport(pos=-25 / 2 + 20j)
    bool_test.circ(lyr=[1, 1], r=20, vex=40)
    N = len(bool_test.pi) - 1
    AND.boolean(lyr=[3, 1], mode='AND', pi1=[bool_test.pi[N], bool_test.pi[N - 1], bool_test.pi[N - 2]], pi2=[])

    bool_test.aref(cell_child=AND, pos=300, portAref=0)

    # NOT example
    bool_test.addport(pos=200j, type='N')
    bool_test.rect(lyr=[1, 1], w=40, h=40, pos='mid')
    bool_test.shiftport(pos=0)
    bool_test.circ(lyr=[1, 1], r=10, vex=40)

    NOT = cell('\\NOT', port='N')
    N = len(bool_test.pi) - 1
    NOT.boolean(lyr=[4, 1], mode='NOT', pi1=bool_test.pi[N - 1], pi2=[bool_test.pi[N]])

    bool_test.shiftport(pos=80)
    bool_test.rect(lyr=[1, 1], w=40, h=40, pos='mid')
    bool_test.shiftport(pos=20)
    bool_test.circ(lyr=[1, 1], r=18, vex=3)
    N = len(bool_test.pi) - 1
    NOT.boolean(lyr=[4, 1], mode='NOT', pi1=bool_test.pi[N - 1], pi2=[bool_test.pi[N]])

    bool_test.shiftport(pos=80)
    bool_test.circ(lyr=[1, 1], r=20, vex=40)
    bool_test.rect(lyr=[1, 1], w=15, h=15, pos='mid')
    N = len(bool_test.pi) - 1
    NOT.boolean(lyr=[4, 1], mode='NOT', pi1=bool_test.pi[N - 1], pi2=[bool_test.pi[N]])

    bool_test.shiftport(pos=200j + 30 + 100j, type=1)
    bool_test.rect(lyr=[1, 1], w=100, h=40, pos='mid')
    bool_test.shiftport(pos=-40)
    for i in range(7):
        bool_test.shiftport(pos=10)
        bool_test.rect(lyr=[1, 1], w=5, h=50, pos='mid')
    N = len(bool_test.pi) - 1
    NOT.boolean(lyr=[4, 1], mode='NOT', pi1=bool_test.pi[N - 7], pi2=[bool_test.pi[i] for i in range(N - 6, N)])

    bool_test.shiftport(pos=200j + 30 + 100j + 150, type=1)
    bool_test.rect(lyr=[1, 1], w=100, h=40, pos='mid')
    bool_test.shiftport(pos=-40)
    for i in range(7):
        bool_test.shiftport(pos=10)
        bool_test.rect(lyr=[1, 1], w=5, h=20 - 1 * i, pos='mid')
    N = len(bool_test.pi) - 1
    NOT.boolean(lyr=[4, 1], mode='NOT', pi1=bool_test.pi[N - 7], pi2=[bool_test.pi[i] for i in range(N - 6, N)])
    bool_test.aref(cell_child=NOT, pos=300, portAref=0)

    bool_test.logo(text='OR', pos=-60, ifnofill=0)
    bool_test.logo(text='AND', pos=-60 + 100j, ifnofill=0)
    bool_test.logo(text='NOT', pos=-60 + 250j, ifnofill=0)
    bool_test.logo(text='Boolean', pos=-80j, w=15, ifnofill=0)

    return bool_test
```

图形重要属性——PORT

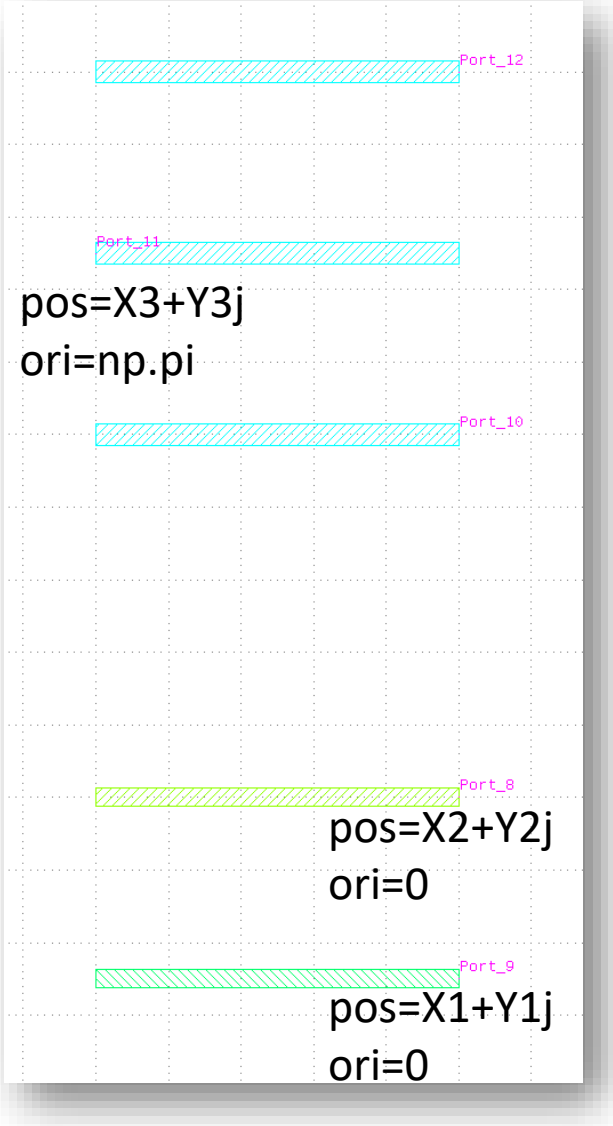
name	名字
portS/portA	可用端口/激活端口
child	cell对象的子Cell

AA = cell(cellname) AA中包含两个Port属性

portS: list；会以tag形式显示在GDSII上的Port，严格按照0,1,2...的顺序排列
示例：AA.portS[i]
portA: list；当前Active的Port，接下来所有的绘图和操作都会在这个port上进行
示例：AA.portA[i]

Port的属性（Port也是一个对象）

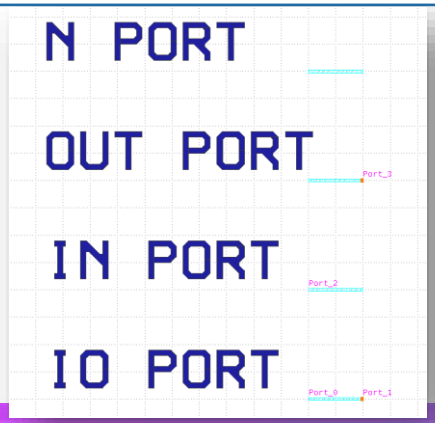
重要属性	**pos: 位置**	**影响绘制的图形** 示例： AA.portA[i].pos AA.portA[i].ori AA.portA[i].bias
	ori: 角度	
	bias: 波导宽度补偿Flag	
纪录属性	l: 累积长度	不影响绘制图形，仅做记录，但可以用于检查、自动化等用途
	w: 宽度	
	wgtype: 波导类型	
	mode/band/type/logo	
Aref用port属性	mirror/Nxy/dxy/MAG	在Aref时影响图形，但会被Aref函数中的变量覆盖

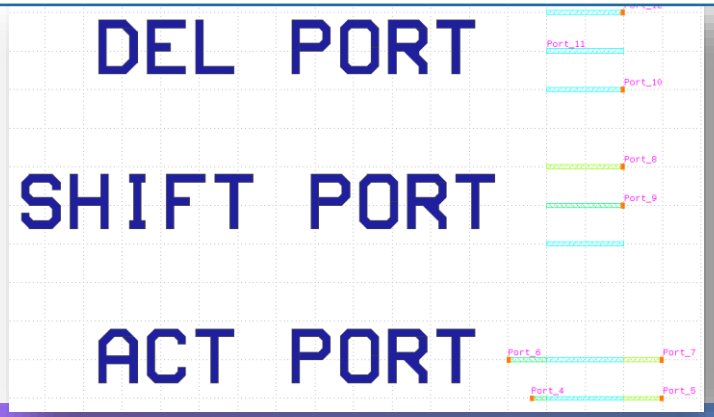


图形重要属性——PORT

AA = cell(cellname)

Port 相关操作：
1. 新建cell时候可以定义新增port类型
示例: AA = cell(name='test_cell', pos=0 + 0j, ori=0, porttype='IO', portlabel=0, owner='')
2. addport 新增一个/一对port（IO/I/O/N）
示例: AA.addport(port='0', pos=0 + 0j, ori=0, l=0, mirror=0, MAG=1, rev=0, Nxy=[1, 1], dxy=[0, 0], w=0, bias=wg_bias, mode='TE', band='C', wgtype='FC', type='I', logo='-1', label='-1')
3. portprint 将当前cell的portS和portA显示在脚本运行界面
示例: AA.portprint()
4. actport将某一个/一组port设置为激活状态
示例: AA.actport(0)/AA.actport([0,1])
5. shiftport移动一个port
示例: AA.shiftport(port='0', type=0, pos=0 + 0j, ori=0, l='0', mirror='0', w='0', Nxy='0', dxy='0', MAG='0', rev='0', bias='0', npp='0', mode='0', band='0', wgtype='0')
6. delport删除一个/一组port，注意删除后的portS将重新按照0,1,2...的排列。
示例: AA.delport(0)/AA.delport([1,2])





示例：ex_pygds2/func_test/test_port.py

```python
def port_test():
    """
    This function test the port function
    :return:
    """
    pitch = 20j
    port = cell('\\Port', porttype='N')
    print('\033[1;32mPort Information at initial \033[0m')
    port.portprint()

    # Add a port
    port.addport(pos=0, type='IO')
    port.swg(lyr=[600, 1])
    port.logo(text='IO port', pos=-50, ifnofill=0)

    port.addport(pos=pitch, type='I')
    port.swg(lyr=[600, 1])
    port.logo(text='In port', pos=-50 + pitch, ifnofill=0)

    port.addport(pos=pitch * 2, type='O')
    port.swg(lyr=[600, 1])
    port.logo(text='Out port', pos=-50 + pitch * 2, ifnofill=0)

    port.addport(pos=pitch * 3, type='N')
    port.swg(lyr=[600, 1])
    port.logo(text='N port', pos=-50 + pitch * 3, ifnofill=0)

    print('\033[1;32mPort Information after add several port \033[0m')
    port.portprint()

    # Active a port
    port.addport(pos=pitch * 4, type='IO')
    port.swg(lyr=[600, 1], w=0.5, l=10)
    port.addport(pos=pitch * 4 + 5j, type='IO')
    port.swg(lyr=[600, 1], w=0.5, l=10)
    port.actport([5, 7])
    port.swg(lyr=[602, 1], w=0.5, l=5)
    port.actport(6)
    port.swg(lyr=[601, 1], w=0.5, l=5)
    port.actport(4)
    port.swg(lyr=[601, 1], w=0.5, l=2)
    port.logo(text='ACT port', pos=-60 + pitch * 4, ifnofill=0)

    # Shift a port
    port.addport(pos=pitch * 5, type='IO')
    port.swg(lyr=[600, 1], w=0.5)

    port.shiftport(pos=-10 + 5j, type=0)  # shift the port with relatively coordinate
    port.swg(lyr=[601, 1], w=0.5)

    port.actport([8])
    port.shiftport(pos=pitch * 5 + 10j, type=1)  # shift the port with absolute coordinate
    port.swg(lyr=[602, 1], w=0.5)
    port.logo(text='shift port', pos=-70 + pitch * 5, ifnofill=0)

    # delete a port
    port.addport(pos=pitch * 6, type='IO')
    port.swg(lyr=[600, 1], w=0.6)
    port.delport(10)

    port.addport(pos=pitch * 6 + 5j, type='IO')
    port.swg(lyr=[600, 1], w=0.6)
    port.addport(pos=pitch * 6 + 10j, type='IO')
    port.swg(lyr=[600, 1], w=0.6)
    port.delport([12, 13])
    port.logo(text='del port', pos=-60 + pitch * 6, ifnofill=0)

    return port
```

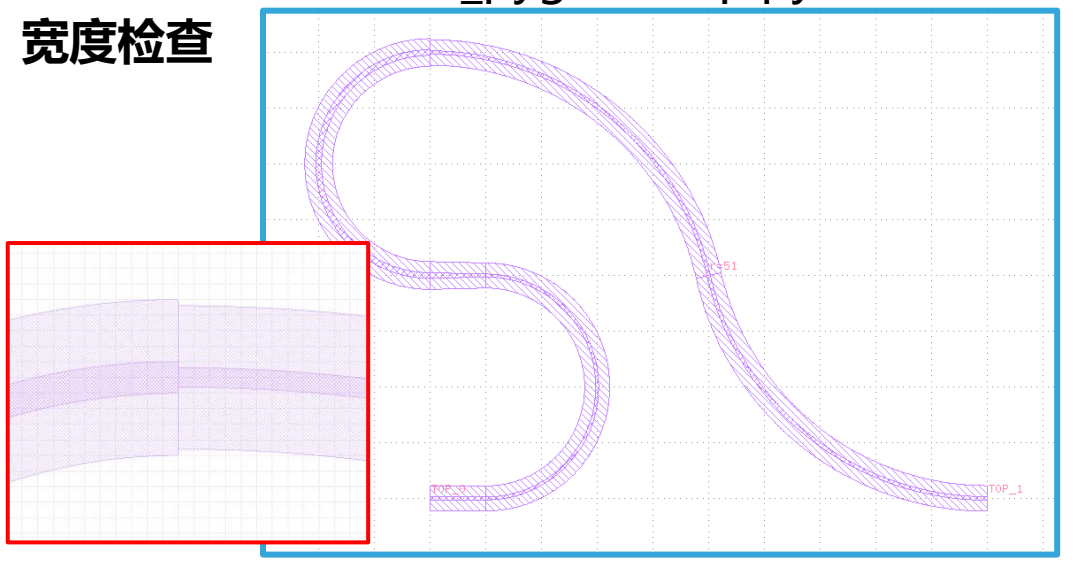
图形重要属性——PORT带来的便利

1. 不移动port，画出的图形都是首尾相接（坐标、角度），省去计算端口的繁琐；
2. Port包含的宽度等信息，可以用于波导连续性检查；
3. 特殊port端口，如EC、GC端口，可以提取处位置信息，生成对应测试信息；

记录属性	l: 累积长度	不影响绘制图形，仅做记录，但可以用于检查、自动化等用途
	w: 宽度	
	wgtype: 波导类型	
	mode/band/type/logo	

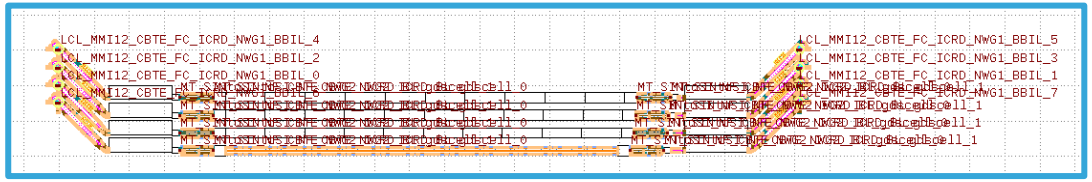
宽度检查

示例：ex_pygds2/top.py



```
G:\develop\picpalette\venv\Scripts\python.exe G:/develop/picpalette/ex_pygds2/test.py
Warning!(w0020) Width discontinuity = -0.4 at cell "TOP" pos:(0.0, 80.0) lyr:(2, 1)
Layout saved at G:\develop\picpalette\ex_pygds2\hello_top.gds
```

测试端口输出

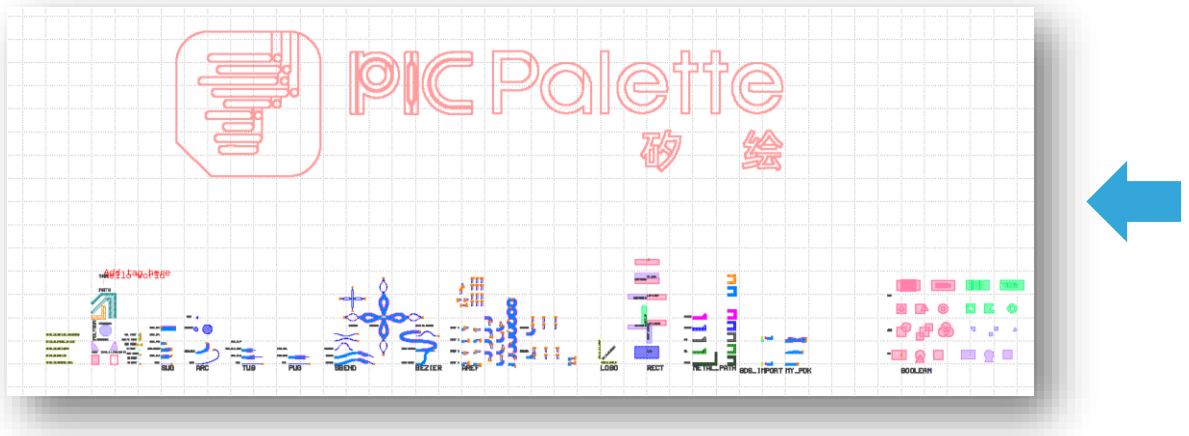


label_In	In/Out	x	y	Coupler	Mode	Band	MFD	Fiber Angle	label_Out	In/Out	x	y	Coupler	Mode	Band	MFD	Fiber Angle
H01	In	4300	850	GC	TE	C	9	10	H01	Out	9800	850	GC	TE	C	9	10
H02	In	4300	750	GC	TE	C	9	10	H02	Out	9800	750	GC	TE	C	9	10
H03	In	4300	950	GC	TE	C	9	10	H03	Out	9800	982	GC	TE	C	9	10
H04	In	5490	5400	GC	TE	C	9	10	H04	Out	9790	5400	GC	TE	C	9	10
H05	In	5490	5300	GC	TE	C	9	10	H05	Out	9790	5300	GC	TE	C	9	10
H06	In	5490	5500	GC	TE	C	9	10	H06	Out	9790	5532	GC	TE	C	9	10
H07	In	3840	8330	GC	TE	C	9	10	H07	Out	8140	8330	GC	TE	C	9	10
H08	In	3840	8230	GC	TE	C	9	10	H08	Out	8140	8230	GC	TE	C	9	10
H09	In	3840	8430	GC	TE	C	9	10	H09	Out	8140	8462	GC	TE	C	9	10

结束绘图

layout(filepath, xmlwrite=1, timelogo=0, ifclear=1)

(相对/绝对)存储路径　是否存储port　是否记录时间　是否清除绘图内容




```python
from pygds2 import*
from func_test import*
from pypdk import*

iffalten = 0

top = cell('\\TOP', porttype='N')

top.aref(cell_test(), pos=0, portAref=0)  # This function test adding a cell
top.aref(graph_test(), pos=200, portAref=0)  # This function test the basic graphs including polygon, path and tag

if iffalten:
    top.flatten()
top.aref(port_test(), pos=400, portAref=0)
top.aref(swg_test(), pos=500, portAref=0)
top.aref(arc_test(), pos=650, portAref=0)
top.aref(twg_test(), pos=850, portAref=0)
top.aref(pwg_test(), pos=1050, portAref=0)
top.aref(sbend_test(), pos=1250, portAref=0)
top.aref(bezier_test(), pos=1600, portAref=0)
top.aref(aref_test(), pos=1800, portAref=0)
top.aref(logo_test(), pos=2400, portAref=0)
top.aref(rect_test(), pos=2600, portAref=0)
top.aref(mt_path_test(), pos=2800, portAref=0)

gds = gds_import('func_test/gds_import_test.GDS')
top.aref(gds, pos=3100)
top.aref('gds_import_aTop', pos=3100 + 100j)

inno = gds_import('func_test/logo.GDS')
top.aref(inno, pos=-1000 + 300j, portAref=0)

# Test Own cell
try:
    from mypdk import MMI1x2, MMI2x2, DC2x2

    mmi_cell1 = MMI1x2(label='BL', band='C', mode='TE', type='FC',
                       l_IO=10, w_taper=1.5, w_MMI=5, w_MMI_tre=2, l_MMI=20, s_MMI=2.5,
                       w_IO=0.6, lyr=TECH_Lyr.WG1)
    top.addport(pos=3200, type='IO', npp=0)
    top.swg_cld(lyr=TECH_Lyr.WG1, w=0.6, l=10)
    top.aref(cell_child=mmi_cell1)

    mmi_cell2 = MMI2x2(label='BL', band='C', mode='TE', type='FC',
                       l_IO=20, w_taper=1.5, w_MMI=5, w_MMI_tre=2, l_MMI=20, s_MMI=2.5,
                       w_IO=0.6, lyr=TECH_Lyr.WG1)
    top.addport(pos=3200 + 50j, type='IO', npp=0)
    top.swg_cld(lyr=TECH_Lyr.WG1, w=0.6, l=10)
    top.aref(cell_child=mmi_cell2)

    dc_cell = DC2x2(label='BL', band='O', mode='TE', type='SK',
                    w_dc=0.5, g_dc=0.2, l_dc=10, s_IO=10,
                    w_IO=0.6, lyr=TECH_Lyr.WG2)
    top.addport(pos=3200 + 100j, type='IO', npp=0)
    top.swg_cld(lyr=TECH_Lyr.WG2, w=0.6, l=10)
    top.aref(cell_child=dc_cell)
except Exception as err:
    pass

top.logo(text='My_PDK', pos=3200 - 40j, w=15, ifnofill=0)
top.logo(text='GDS_IMPORT', pos=3000 - 40j, w=15, ifnofill=0)

try:
    top.aref(cell_child=gds_import_test(), pos=3400, portAref=1)
except Exception as e:
    print(e)
    pass

top.aref(cell_child=xor_test(), pos=3700 + 40j, portAref=0)

# Write GDS
filepath = 'hello_world'
layout(filepath, xmlwrite=0, timelogo=0)
```

预习Python

```
from pygds2 import *

AA = cell('name')

BB = gds_import('path')

AA.xxx(a=1,b=2,c=3…)

layout(path)
```

PDK——导入和自建PDK

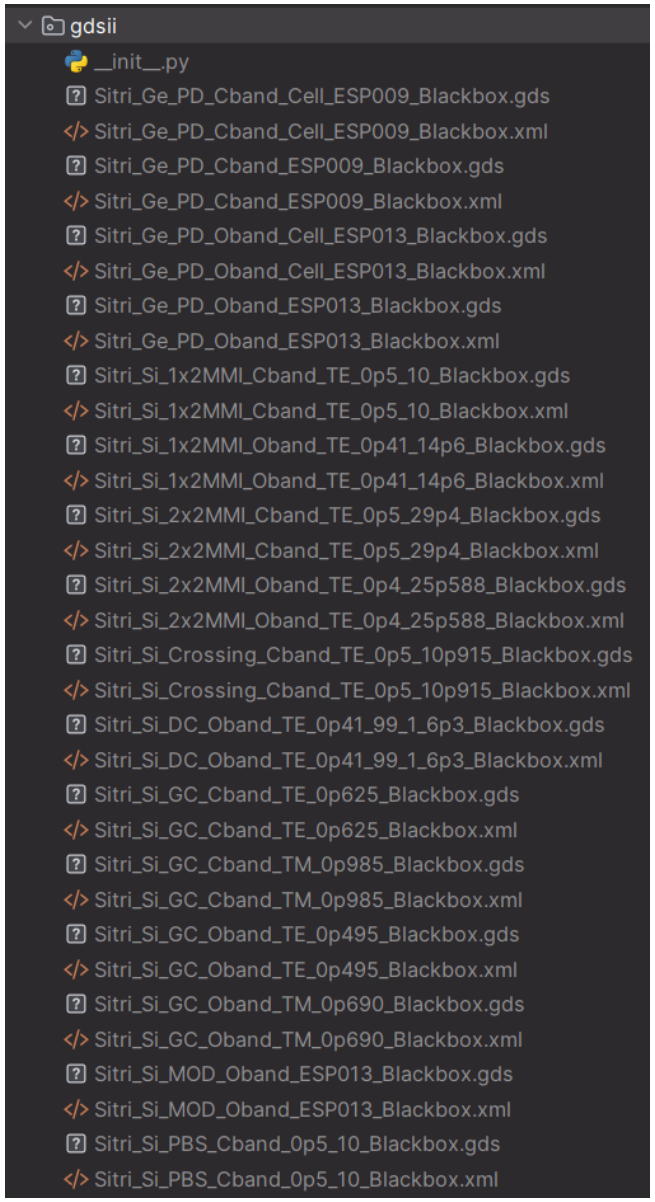
AA = cell(cellname)

BB = gds_import(filepath)

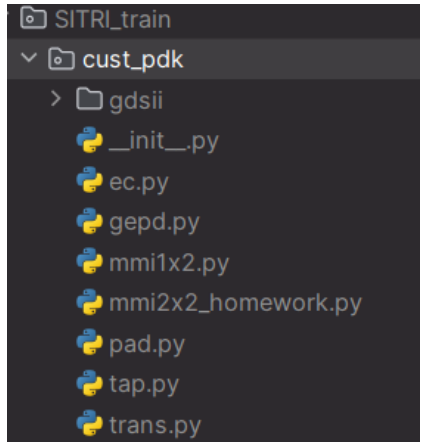
AA.aref(child_cell=BB, port_child=i)

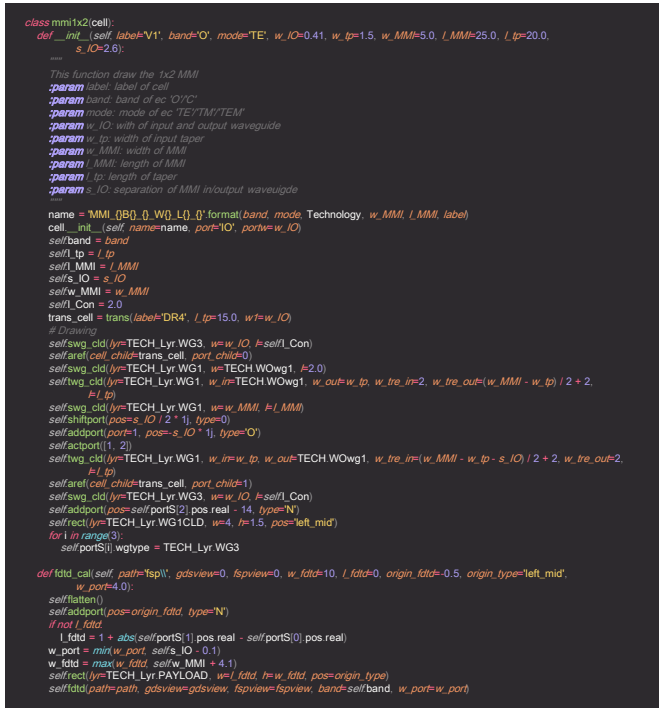
1. Fab提供的GDSII，可以直接导入；配合.xml文件可以自动添加port，并在引用时使用；
2. 自建PDK
   A. Cell对象，以参数化的形式引用；
   B. 生成GDSII+XML后，以GDSII的形式引用。

Fab PDK



用户自建 PDK





PDK——绘图属性

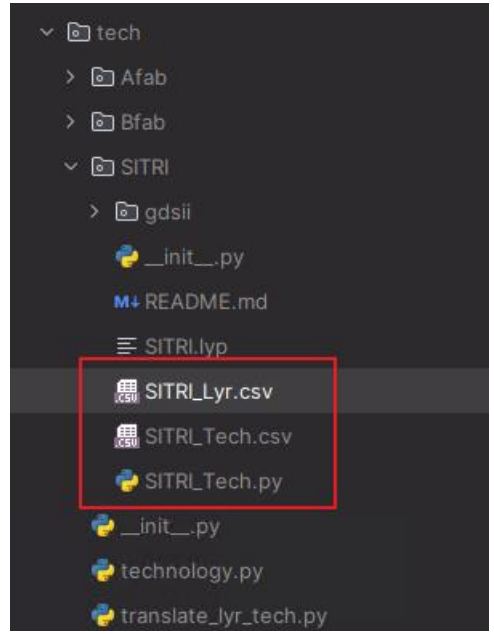
PIC Palette 矽 绘

SITRI_Lyr.py

	A	B	C	D	E	F	G
1	General Definition (Definition used in all PDK)	PDK definition (Definition from PDK)	gdsii layer	gdsii datatype	Waveguide type	Owner Name (Other definitions)	description
2	WG	HARDMASK	1	0		WG3COR	Silicon waveguide core: Hard mask (Dark)
3	RB	RIBWG	2	0			60nm Si dry etch on patterned area only (Clear)
4	SP	STRIPWG	3	0			150nm Si dry etch until BOX (Clear)
5	PB	P	13	0			P-type modulator PN junction implant
6	NB	N	14	0			N-type modulator PN junction implant
7		PM	15	0			P-type intermediate implant
8		NM	16	0			N-type intermediate implant
9	PPLUS	PH	11	0			P-type contact implant
10	NPLUS	NH	12	0			N-type contact implant
11	PP	P_PD	17	0			P-type Si implant for GePD
12	NP	N_PD	18	0			N-type Si implant for GePD
13	GE	GeW	20	0			Window for Ge epitaxy
14	GN	GeNP	21	0			N-type Ge implant for GePD
15	GP	GePP	22	0			P-type Ge implant for GePD
16	CS	CTSI	35	0			Contact Plugs from M1 to Si
17	CG	CTGE	36	0			Contact Plugs from M1 to Ge
18	M1	M1	40	0		M1D;M1DRW	Metal 1 drawing
19	HT	TiN	45	0		MHD;MHDRW	Metal Heater
20	V1	VIA1	50	0		VIA12	Via1 contact M1 and M2
21	M2	M2	55	0		M2D;M2DRW	Metal 2 drawing
22	PAD	AIPAD	66	0		PADOPEN	Pad open to Al pad
23	LP	DIOPEN	70	0		LPASS	OPEN for GC
24	DT	EDGE	71	0		TRENCH	Deep Trench
25	PAYLOAD	FRAME	0	0			Pygds2: Outline of design block
26	NOPASS	NOWG	150	0			No WG dummy
27		NOMET	151	0			No M1 dummy
28		NOFILL	152	0			No WG and M1 dummy
29	LOGO	Union_Lyr				WG;M2	LOGO print on the chip; (WG1; M2)
30	WG1COR	Union_Lyr				WG	Rib waveguide 150
31	WG1CLD	Union_Lyr		2		SP	Rib waveguide 150
32	WG2COR	Union_Lyr				WG	Rib waveguide 90
33	WG2CLD	Union_Lyr	2,4			SP;RB	Rib waveguide 90
34	WG3	Union_Lyr				WG	Strip Waveguide
35	WG_LYR	Union_Lyr				WG1;WG2;WG3	List the waveguide layers (leave empty if default)
36	WG_Con_Check	Union_Lyr				WG;WG1COR;WG2COR	Waveguide layers for connection check
37	FDTD_Lyr	Union_Lyr				PAYLOAD;WG,[SP;WG],[RB;WG],GeW	option

SITRI_Tech.py

	A	B	C
1	Item	Value (um)	description
2	Wtre	2	width for trech beside waveguide core
3	Biaswg1	0	width bias for WG1
4	Biaswg2	0	width bias for WG2
5	Biaswg3	0	width bias for WG3
6	RCwg1	80	Radius threshold value for Cband WG1 warning
7	RCwg2	10	Radius threshold value for Cband WG2 warning
8	RCwg3	5	Radius threshold value for Cband WG3 warning
9	ROwg1	50	Radius threshold value for Oband WG1 warning
10	ROwg2	10	Radius threshold value for Oband WG2 warning
11	ROwg3	5	Radius threshold value for Oband WG3 warning
12	WCwg1	0.7	Default width of Cband WG1
13	WCwg2	0.5	Default width of Cband WG2
14	WCwg3	0.5	Default width of Cband WG3
15	WOwg1	0.6	Default width of Oband WG1
16	WOwg2	0.41	Default width of Oband WG2
17	WOwg3	0.41	Default width of Oband WG3
18	HSitop	0.22	Height of silicon top (Used in FDTD Simulation)
19	Hwg1etch	0.07	Thickness of wg1 etch (Used in FDTD Simulation)
20	Hwg2etch	0.13	Thickness of wg2 etch (Used in FDTD Simulation)
21	Hwg3etch	0.22	Thickness of wg3 etch (Used in FDTD Simulation)
22	HGe	0.515	Thickness of Germanium
23	HGeRes	0.115	Thickness of Germanium1
24	FDTD_material	['SiO2 (Glass) - Palik','Si (Silicon) - Palik',['Si (Silicon) - Palik',	(FDTD Simulation) Material List
25	FDTD_mesh	[5,1,2,2,1]	(FDTD Simulation) mesh list
26	FDTD_zmin	[0,0,0,($HSitop-$Hwg2etch),$HSitop]	(FDTD Simulation) min z list
27	FDTD_zmax	[0,$HSitop,($HSitop-$Hwg1etch),$HSitop,($HSitop+$HGe)]	(FDTD Simulation) max z list



translate_lyr_tech.py

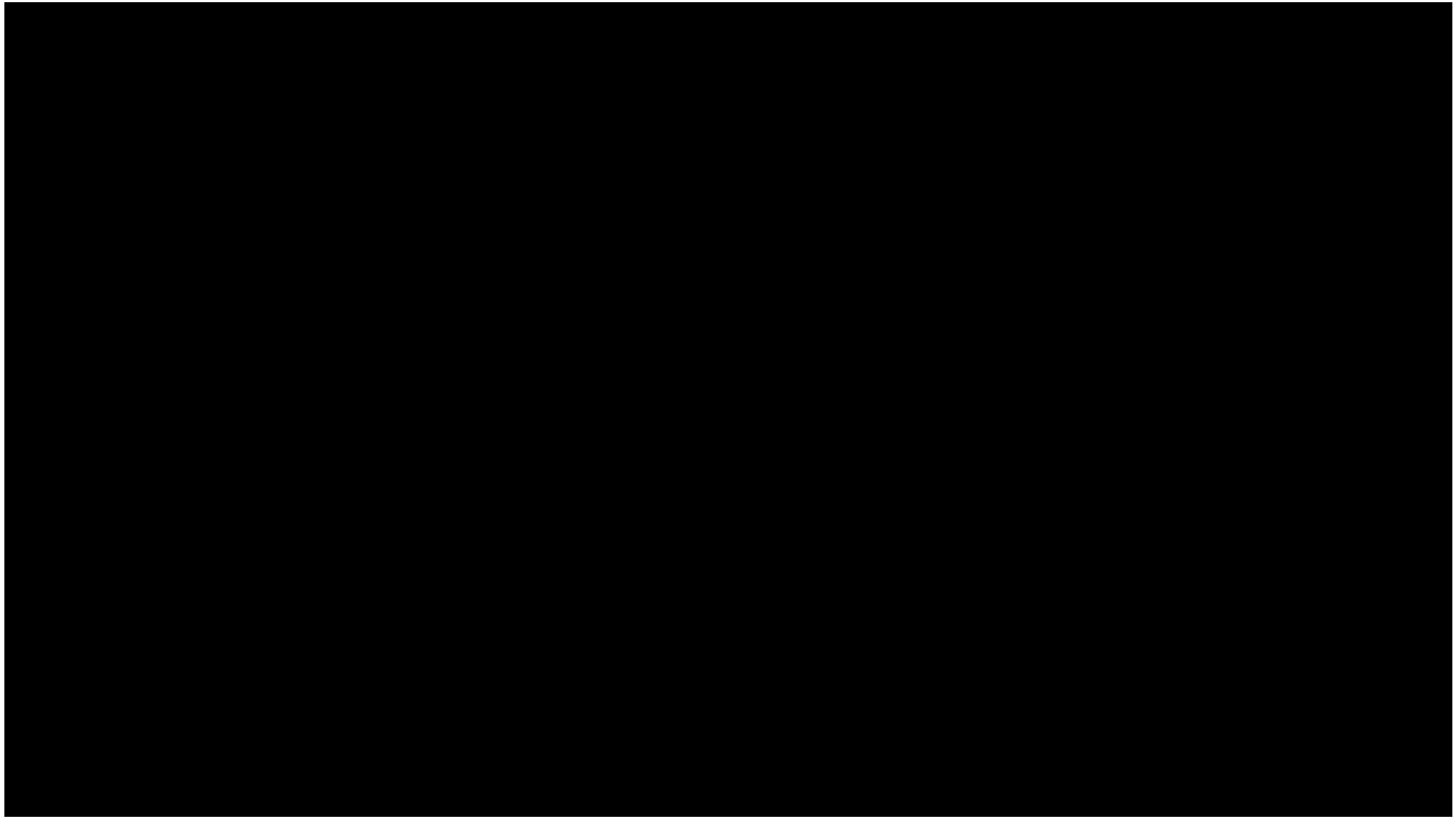
PIC Palette 使用的PDK文件（SITRI_Tech.py）

1. **层定义（单层波导/多层波导）**
2. **Waveguide bias、Radius、Trench width等常数；**
3. **FDTD导入模型定义**

互联仿真功能演示

PIC Palette 矽 绘

示例：90 degree bend绘图→器件仿真演示

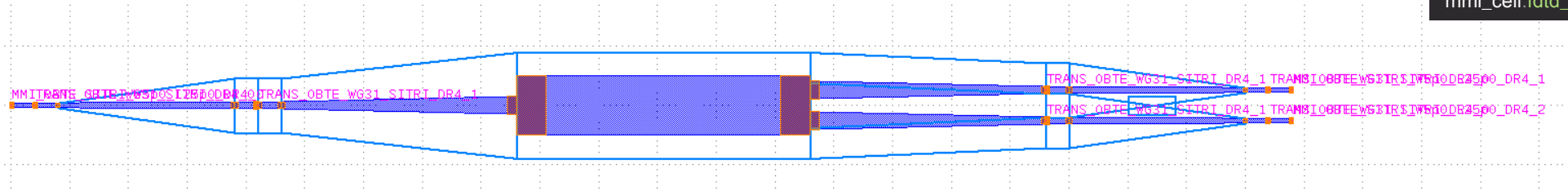


FDTD 流程解析

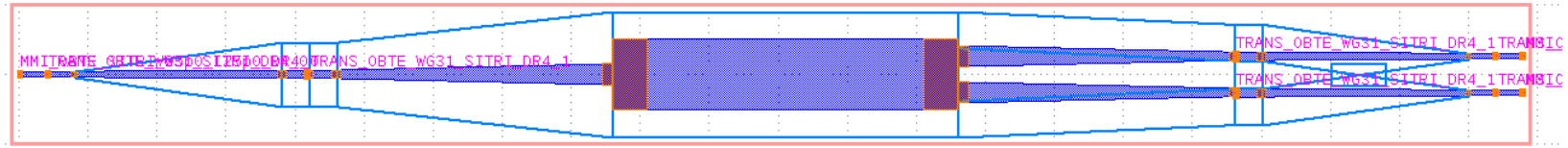
1. 绘制一个GDSII，设置好端口参数（wgtype, w）。

SITRI_train/SIMU_mmi.py

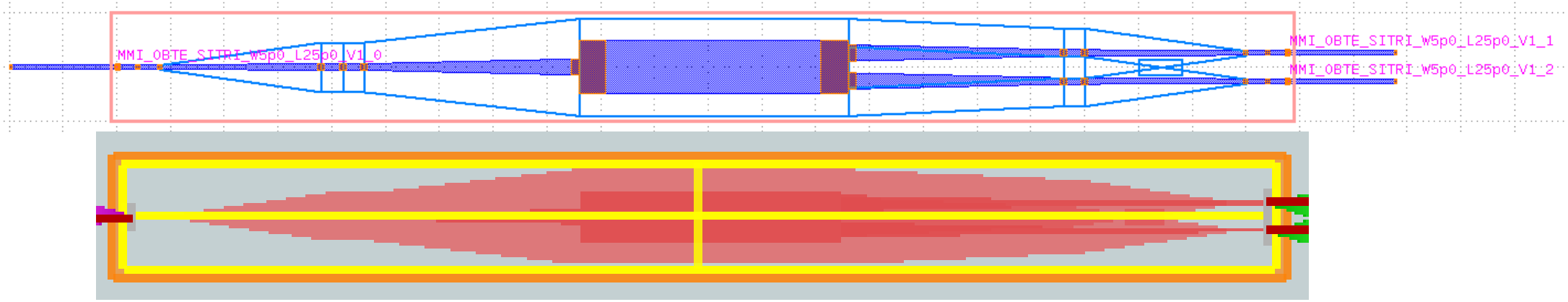
```
mmi_cell = mmi1x2()
mmi_cell.fdtd_cal(path='fsp\\', gdsview=0, fspview=1)
```



2. 用PAYLOAD层绘制出FDTD区域



3. 导入fdtd计算method→自动延伸端口→自动导入FDTD软件建模



查错功能

1. **针对硅光器件特点设置一些差错功能：**

A. Width discontinuity 波导不连续；

B. Bend Radius过小；

C. Cell旋转角度；

D. Cell Name重名；

E. 更多定制功能……

```
Warning!(w0020) Width discontinuity = 0.1 at cell "SWG" pos:(30.0, 0.0) lyr:(3, 1)
Warning!(w0020) Width discontinuity = -0.1 at cell "SWG" pos:(10.0, 30.0) lyr:(2, 1)
Warning!(w0020) Width discontinuity = -0.11 at cell "SWG" pos:(10.0, 35.0) lyr:(2, 1)
Warning!(w0023) Input waveguide layer template in cell SWG not in Technology.
Warning!(w0020) Width discontinuity = -0.6 at cell "SWG" pos:(20.0, 120.0) lyr:([4, 1], [5, 1])
Warning!(w0020) Width discontinuity = -0.1 at cell "ARC" pos:(10.0, 0.0) lyr:(1, 1)
Warning!(w0021) CBand WG1 ARC radius = 80 at cell "ARC" pos:(0.0, 50.0)
Warning!(w0020) Width discontinuity = -0.1 at cell "ARC" pos:(40.0, 60.718) lyr:(2, 1)
Warning!(w0021) OBand WG3 ARC radius = 4 at cell "ARC" pos:(47.321, 88.038)
Warning!(w0020) Width discontinuity = -0.5 at cell "TWG" pos:(50.0, 0.0) lyr:(3, 1)
Warning!(w0020) Width discontinuity = -0.5 at cell "TWG" pos:(50.0, 30.0) lyr:(3, 1)
Warning!(w0020) Width discontinuity = 0.05 at cell "TWG" pos:(50.0, 90.0) lyr:([4, 1], [5, 1])
Warning!(w0020) Width discontinuity = 0.05 at cell "PWG" pos:(70.0, 60.5) lyr:([4, 1], [5, 1])
Warning!(w0011) ARef Rotation angle is not 90/180/270! Cell Name: FS_child2
Warning!(w0011) ARef Rotation angle is not 90/180/270! Cell Name: FS_child2
Warning!(w0011) ARef Rotation angle is not 90/180/270! Cell Name: FS_child2
Warning!(w0011) ARef Rotation angle is not 90/180/270! Cell Name: FS_child2
Warning!(w0014) 2 point annihilate at AREF, (470.0, 25.0)
Warning!(w0003) gds_import_bTop is active. Multiple Top cell in gds_import_test.GDS.
Layout saved at G:\develop\picpalette\ex_pygds2\hello_world.gds
```

Code	functiuon	pkg	flags	Warning text	Comments
w0001	cellinit	pygds2	warning_cell	Warning!(w0001) Cellname Repeat! The existed CellName {}!'	Cell name repeated. The existed object will be returned
w0002	gds_import	pygds2	warning_cell	Warning!(w0002) Import GDSII existed CellName {} in {}!	Cell name repeated. The cell will not be imported
w0003	gds_import	pygds2	warning_cell	Warning!(w0003) {} is active. Multiple Top cell in {}	The imported gds has more than 1 top cell. Only 1 top cell is returned.
w0003	polygon_draw	pygds2	-	Warning!(0003) >{} points in one polygon! Please check the layout carefully!	Too much points in a polygon. Decide by "max_polygon_points" in a_tech._init_
w0010	_aref_draw	pygds2	warning_aref	Warning!!! ARef Magnification is not equal to 1! Cell Name: {}	The magnification of cell is not 1
w0011	_aref_draw	pygds2	warning_aref	Warning!(w0011) ARef Rotation angle is not 90/180/270! Cell Name: {}'	The rotation anlge of cell is not 0/90/180/270
w0012	cell.delrepport	pygds2	warning_aref	Warning!(w0012) Over 2 points coincide at same position {}, ({}, {})	More than 2 ports has the same coordination
w0013	cell.delrepport	pygds2	warning_aref	Warning!(w0013) 2 points coincide at same position {}, ({}, {})	2 ports has the same coordination. The angle difference is not pi
w0020	w_cont	pygds2	warning_waveguide	Warning!(w0020) Width discontinuity = {} at {} pos:({}, {}) lyr:({}, {})	Two connected waveguides has different width
w0021	r_check	pygds2	warning_waveguide	Warning!(w0021) {}Band WG1 ARC radius = {} at {} ({}, {})	Arc radius is smaller than warning threshold
w0021	r_check	pygds2	warning_waveguide	Warning!(w0021) {}Band WG2 ARC radius = {} at {} ({}, {})	Arc radius is smaller than warning threshold
w0021	r_check	pygds2	warning_waveguide	Warning!(w0021) {}Band WG3 ARC radius = {} at {} ({}, {})	Arc radius is smaller than warning threshold
w0022	swg/twg/pwg	pygds2	warning_waveguide	Warning!(w0022) Waveguide length smaller than min_XY {} setup.	Waveguide length smaller than min_XY 0.001 setup
e0000	gds_import	pygds2	-	Error!(e0000) Import GDS header ERROR in {}	Import GDSII has an error in head
e0001	gds_import	pygds2	-	Error!(e0001) Import GDSII data size is too small in {}	Import GDSII has a data size error
e0002	gds_import	pygds2	-	Error!(e0002) data size is odd in {} {} {}	Wrong input layer definition for waveguide with cladding
e0010	swg_cld	pygds2	-	Error!(e0010) Wrong lyr input for swg_cld in {} ({},{})	Wrong input layer definition for waveguide with cladding
e0010	arc_cld	pygds2	-	Error!(e0010) Wrong lyr input for arc_cld in {} ({},{})	Wrong input layer definition for waveguide with cladding
e0010	sbend_cld_draw	pygds2	-	Error!(e0010) Wrong lyr input for SBend_CLD in {} ({},{})	Wrong input layer definition for waveguide with cladding
e0010	cbend_cld_draw	pygds2	-	Error!(e0010) Wrong lyr input for CBend_CLD in {} ({},{})	Wrong input layer definition for waveguide with cladding
e0010	euler_cld	pygds2	-	Error!(e0010) Wrong lyr input for euler_cld in {} ({},{})	Wrong input layer definition for waveguide with cladding
e0010	pwg_cld	pygds2	-	Error!(e0010) Wrong lyr input for pwg_cld in {} ({},{})	Wrong input layer definition for waveguide with cladding
e0010	twg_cld	pygds2	-	Error!(e0010) Wrong lyr input for twg_cld in {} ({},{})	Wrong input layer definition for waveguide with cladding
e0011	cld drawing in	pygds1/pygds2	-	Error!(e0011) Wrong lyr input for cld drawing!	Wrong input layer definition for waveguide with cladding
e0012	ext_wtre	pygds1		Error!(e0012) Wrong w_tre(_in/out) input for cld drawing! (Example: w_tre(_in/out) = 2.0)	Wrong input w_tre(_in/out) for waveguide with cladding
e0011	logo	pygds1/pygds2	-	Error!(e0011) Label {} is not available in logo	Label is not supported in the logo
e0012	cell.addport	pygds1/pygds2	-	Error!(e0012) Wrong port type! No ports added. Please type in 'I'/'O'/'IO'/'N'	Wrong type of port. Only 'I'/'O'/'IO'/'N' supported

SITRI Technology

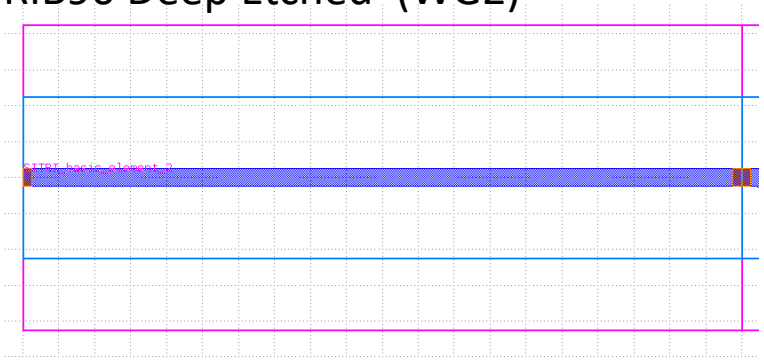
NO	GDS No	Data type	Description	Layer Name	Field	Overlay spec (nm)	Minimum Feature Size (μm)
1	1	0	Hard Mask (WG)	WG	Dark	/	0.2
2	2	0	Rib WG	RB	Clear	80	0.3
3	3	0	Strip WG	SP	Dark	50	0.3
4	13	0	P	PB	Clear	50	0.5
5	14	0	N	NB	Clear	50	0.5
6	15	0	P+	PM	Clear	80	0.5
7	16	0	N+	NM	Clear	80	0.5
8	11	0	P++	PH	Clear	80	0.5
9	12	0	N++	NH	Clear	80	0.5
10	17	0	P-PD	PP	Clear	80	0.5
11	18	0	N-PD	NP	Clear	80	0.5
12	20	0	Ge window	GE	Clear	50	0.5
13	21	0	GeN+	GN	Clear	80	0.5
14	22	0	GeP+	GP	Clear	80	0.5
15	35	0	Contact-Si	CS	Clear	80	1
16	36	0	Contact-Ge	CG	Clear	80	1
17	40	0	M1	M1	Dark	200	1
18	45	0	TiN heater	HT	Dark	200	1
19	50	0	VIA1	V1	Clear	200	2
20	55	0	M2	M2	Dark	200	2
21	66	0	Pad open to Al pad	PAD	Clear	1000	5
22	70	0	Dielectric open	LP	Clear	1000	5
23	71	0	edge facet	DT	Clear	200	2

Layout

截面

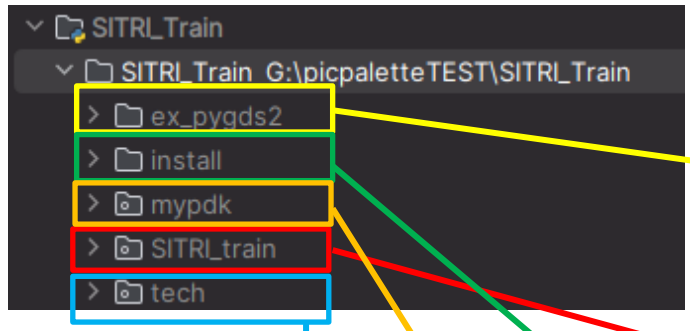
RIB150 Shallow Etched (WG1)

RIB90 Deep Etched (WG2)

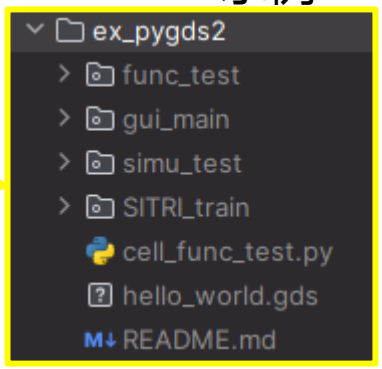


Channel Waveguide (WG3)

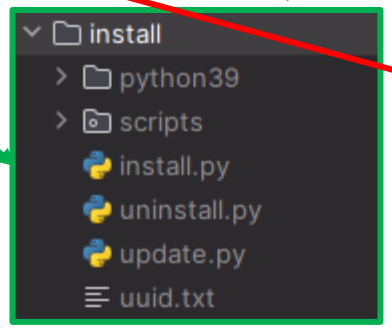
SITRI Train



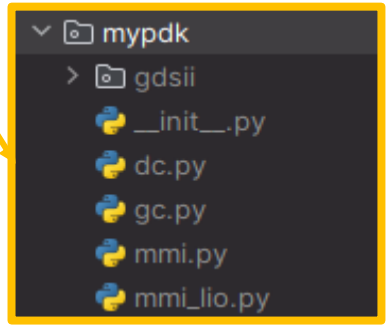
PIC Palette示例



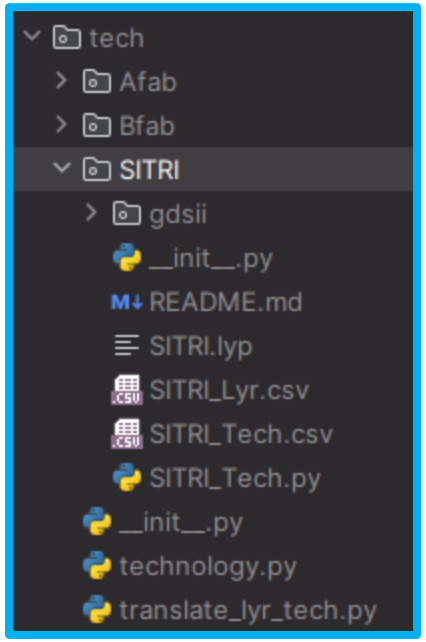
PIC Palette安装



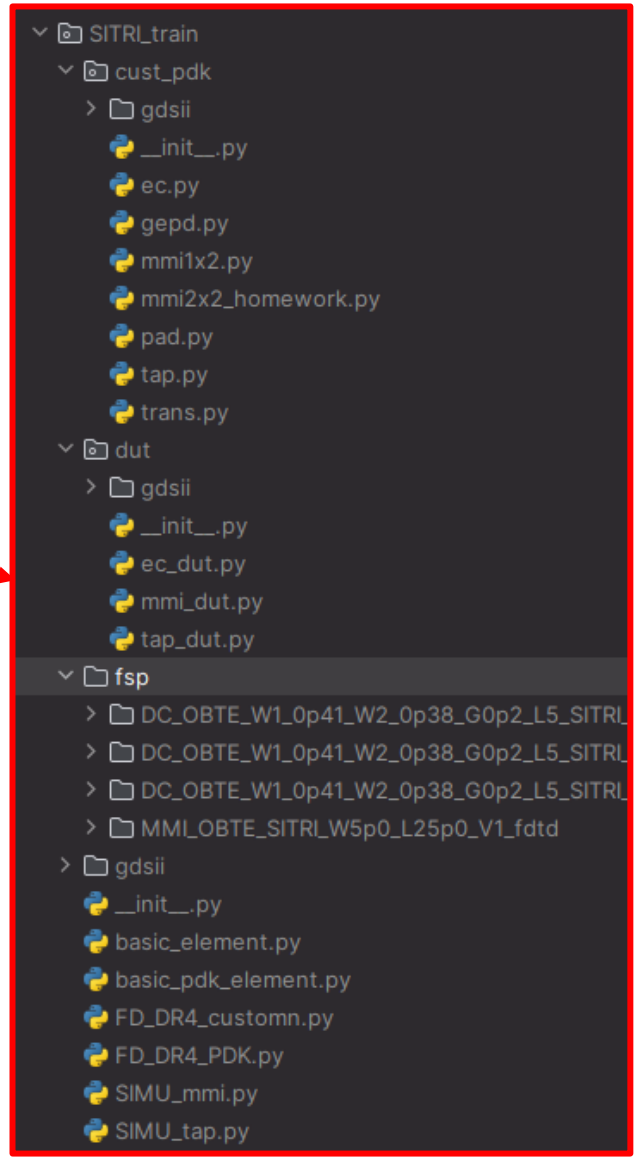
PIC Palette 示例



Technology文件

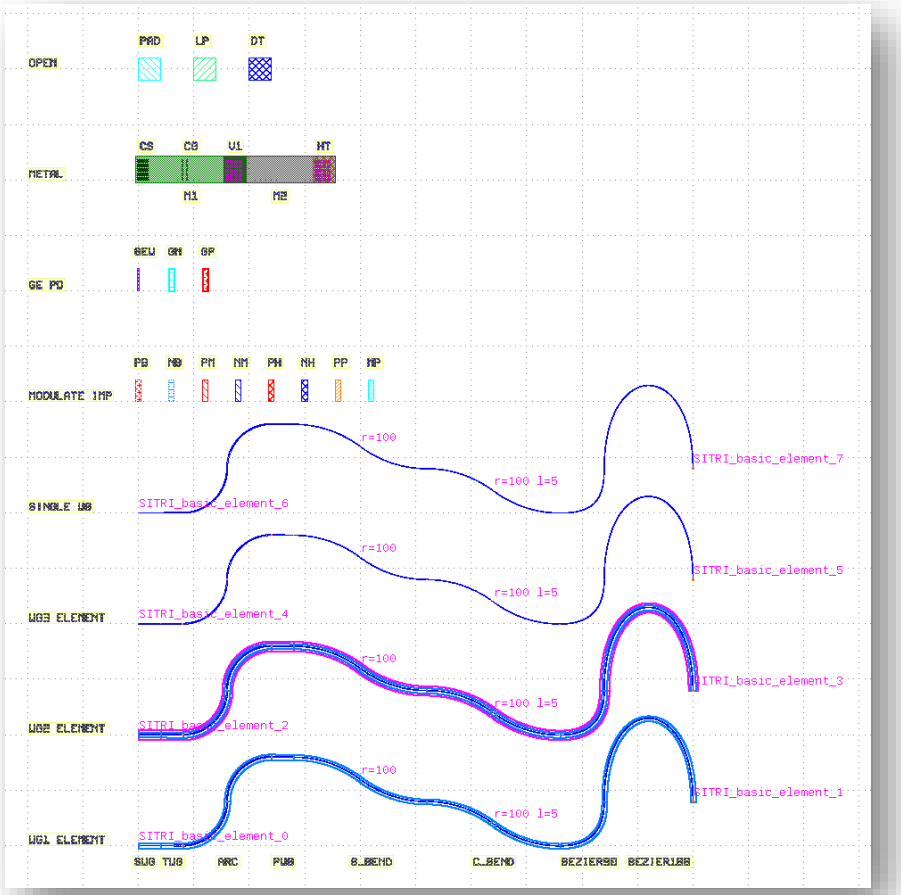


SITRI Train

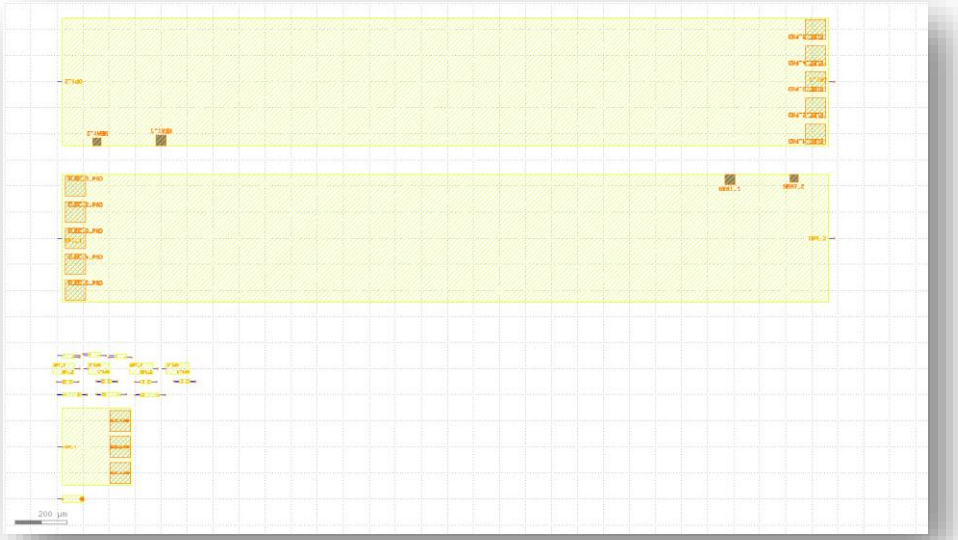


案例展示

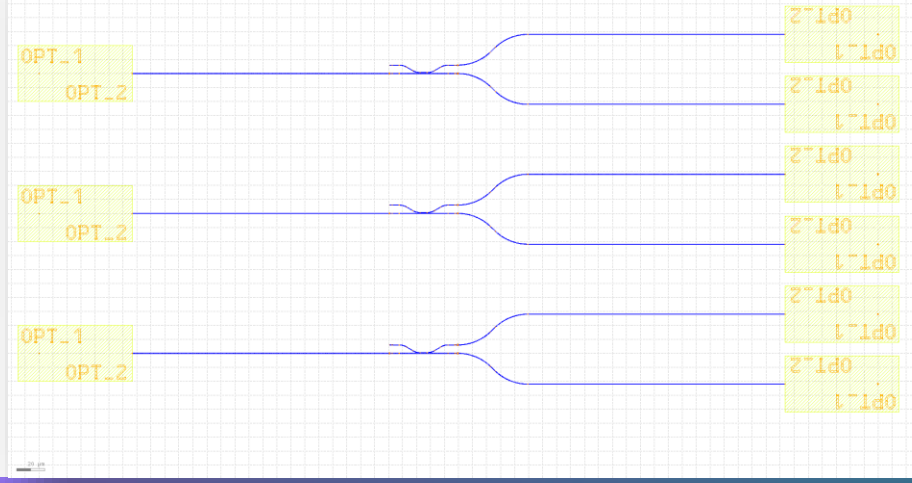
SITRI_train/basic_element.py
SITRI_train/gdsii/SITRI_basic_element.gds



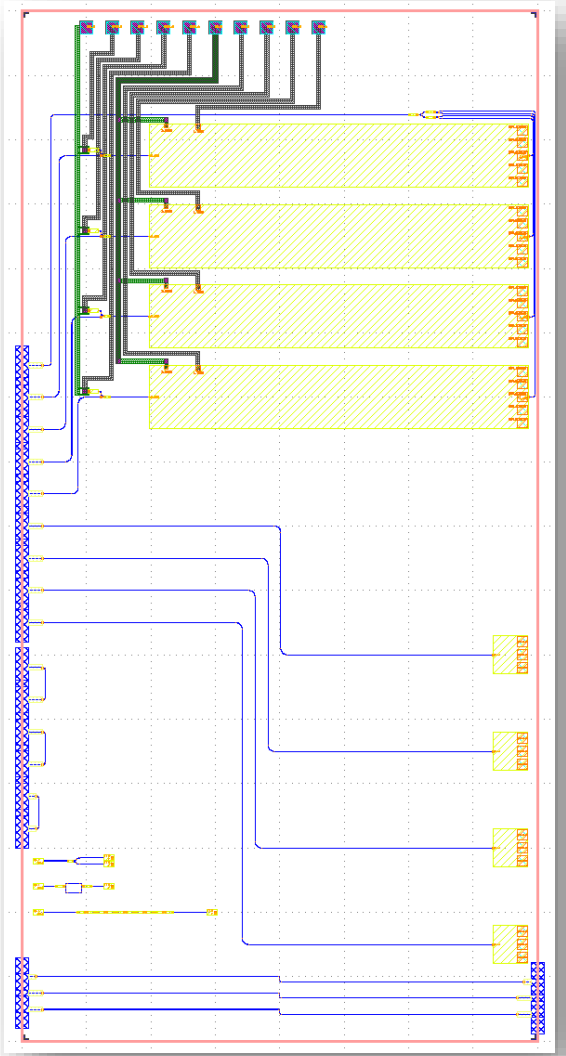
SITRI_train/basic_pdk_element.py
SITRI_train/gdsii/SITRI_pdk_element.gds



SITRI_train/dut/tap_dut.py
SITRI_train/dut/gdsii/DUT_DC_OBTE_SITRI_DR4_Scan.gds

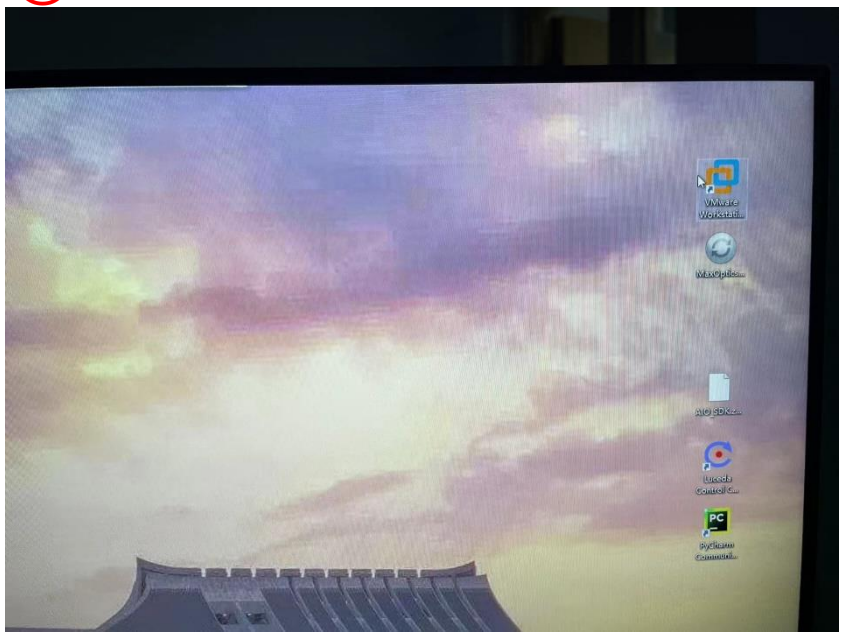


SITRI_train/FD_DR4_PDK.py
SITRI_train/gdsii/DR4_Chip_SITRI_PDK.gds

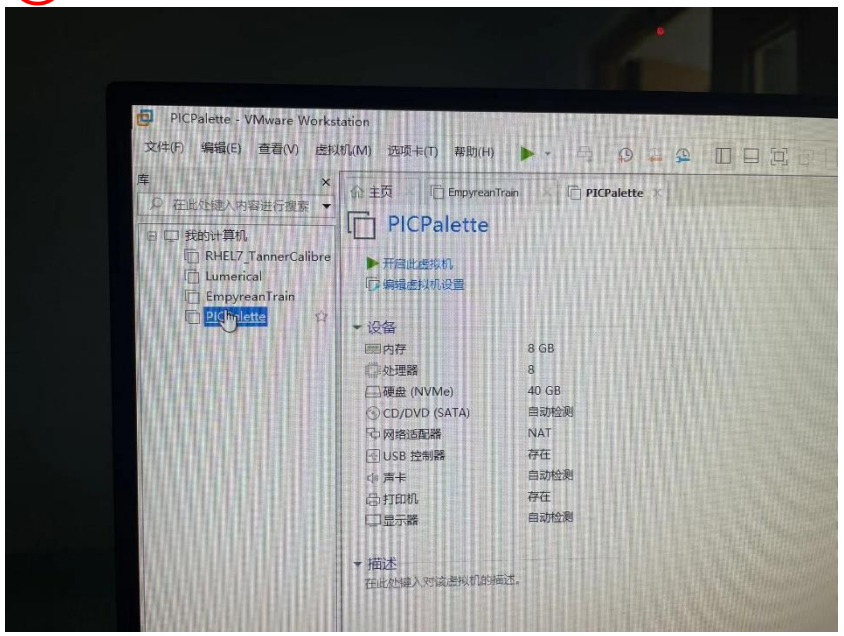


Start

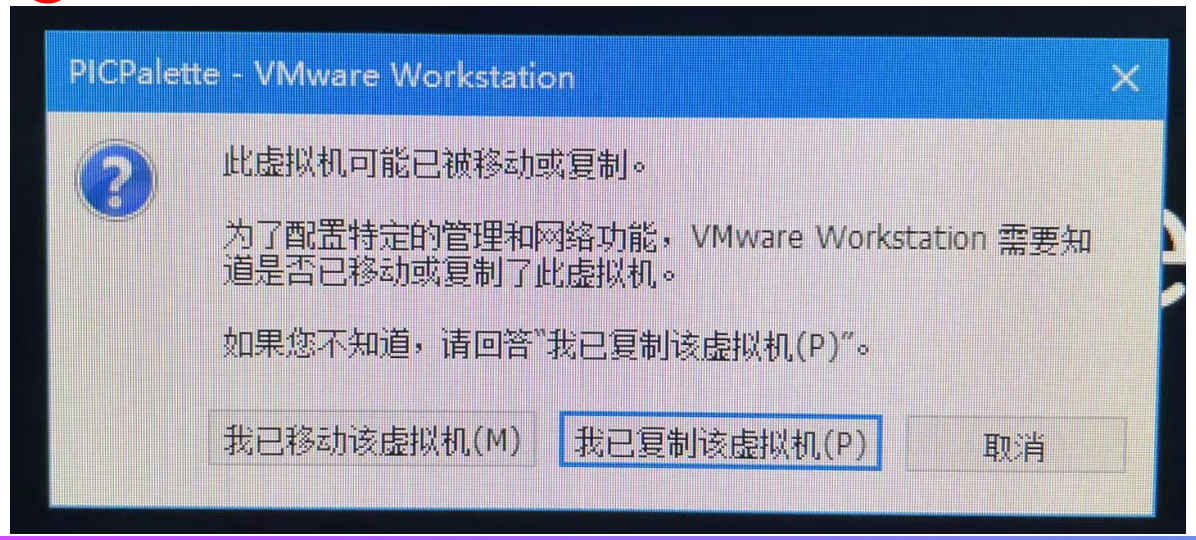
①



②


PICPalette - VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
库
在此处键入内容进行搜索
我的计算机
RHEL7_TannerCalibre
Lumerical
EmpyreanTrain
PICPalette
主页
EmpyreanTrain
PICPalette
PICPalette
开启此虚拟机
编辑虚拟机设置
设备
内存 8 GB
处理器 8
硬盘 (NVMe) 40 GB
CD/DVD (SATA) 自动检测
网络适配器 NAT
USB 控制器 存在
声卡 自动检测
打印机 存在
显示器 自动检测
描述
在此处键入对该虚拟机的描述。

③


PICPalette - VMware Workstation
此虚拟机可能已被移动或复制。
为了配置特定的管理和网络功能，VMware Workstation 需要知道是否已移动或复制了此虚拟机。
如果您不知道，请回答"我已复制该虚拟机(P)"。
我已移动该虚拟机(M)
我已复制该虚拟机(P)
取消

PIC Palette 矽绘

感谢您的关注

Thank you for attention!

致谢：

INNOLIGHT, ITRI

Jiangpeng Chen (INNOLIGHT)

Chenlei Li (INNOLIGHT, ZJU)

Hongbing Lei (Futurewei, Neo Photonics)

Sitao Chen (ZJU, Hisense)

打造属于自己的生产力工具！

下载链接：http://picpalette.innolight.com:60000/

技术支持

技术交流群