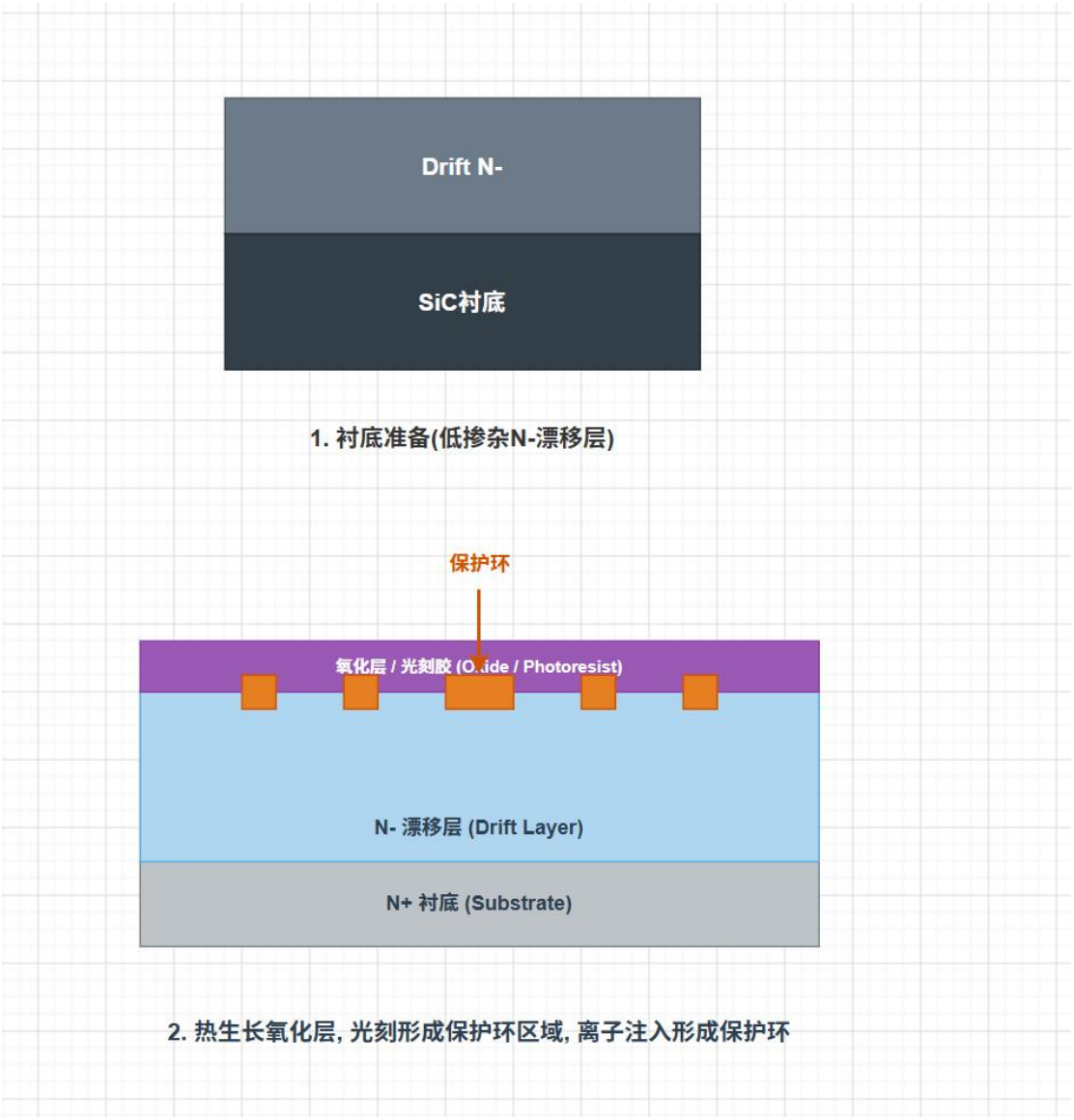
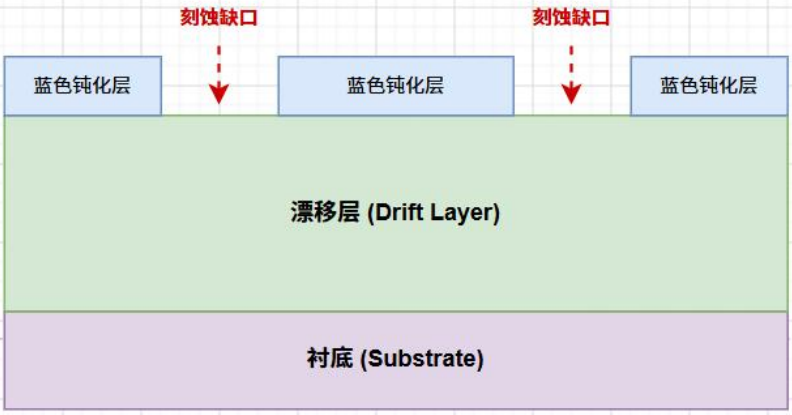
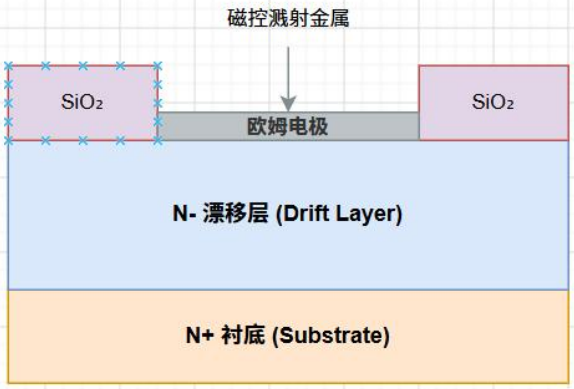
1. 画出保护环终端的器件流程图；



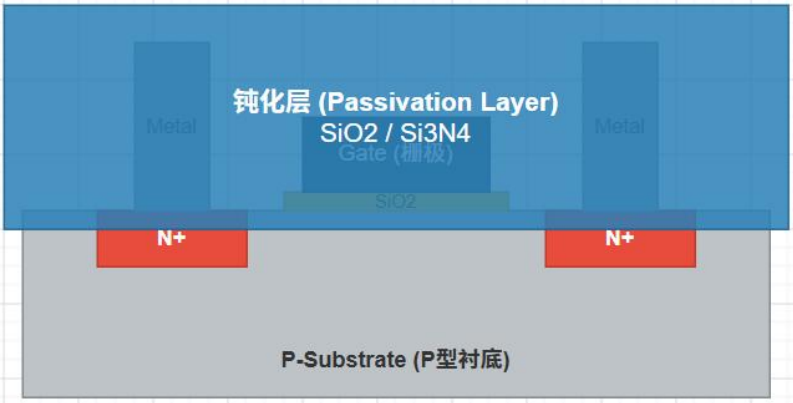
3. 去掉氧化层,重新钝化



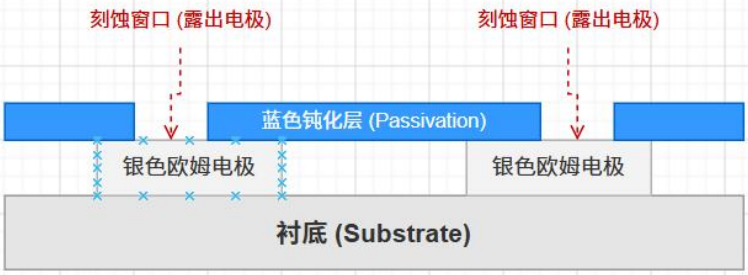
4. 刻蚀
使用任意多边形工具，对蓝色钝化层做裁切，开出多处缺口，保留两侧、中间区域钝化层，露出下方漂移层，形成台面结构。



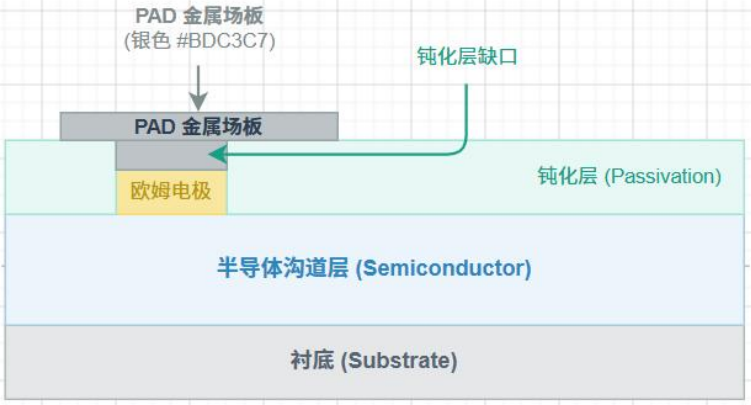
5. 磁控溅射欧姆电极



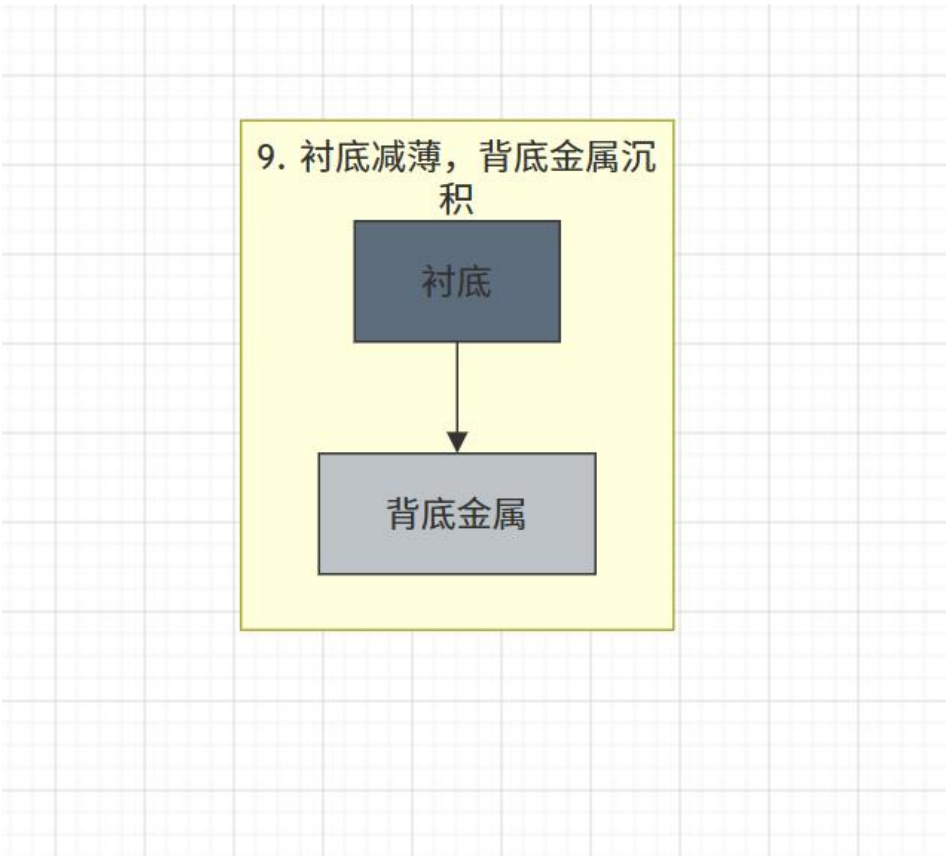
6. 沉积钝化层



7. 刻蚀钝化层

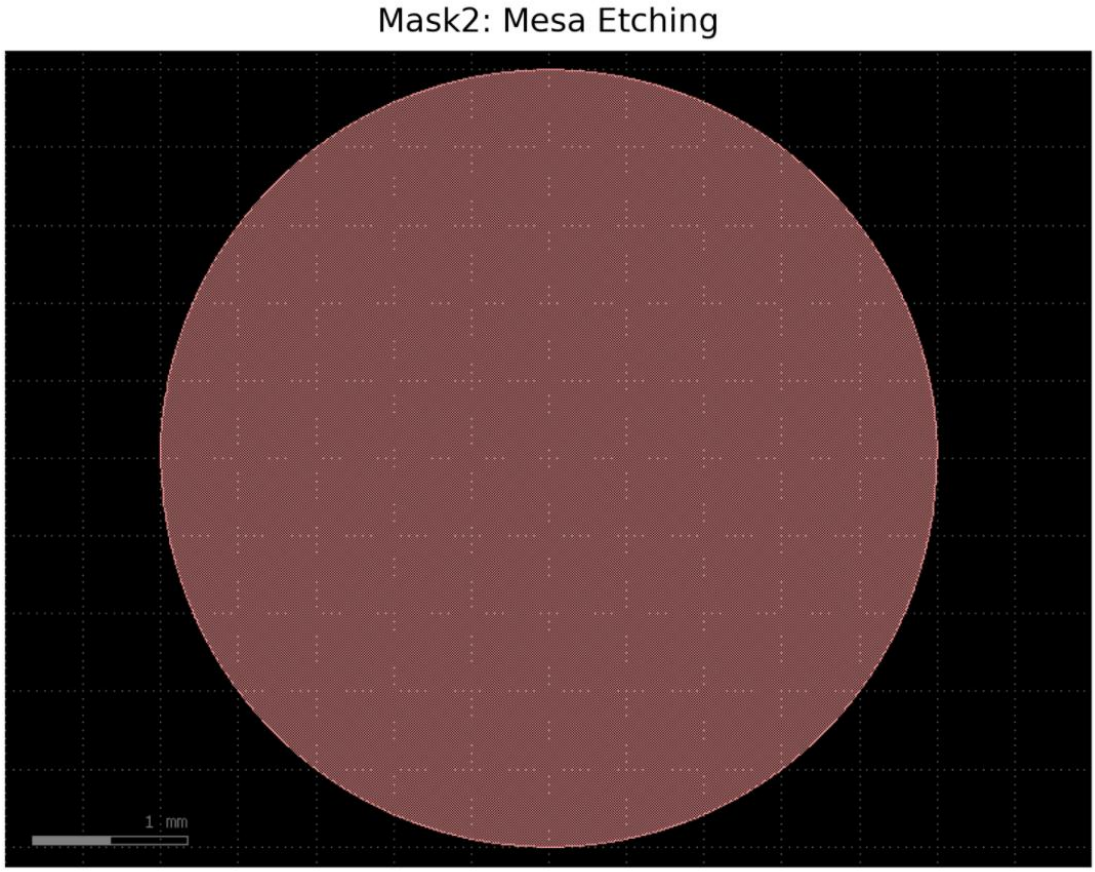


8. 沉积 PAD 金属场板
复制第 7 步图形。在钝化层缺口处沉积银色（#BDC3C7）金属条，金属条从缺口延伸至钝化层表面，连通内部欧姆电极，构成场板结构。

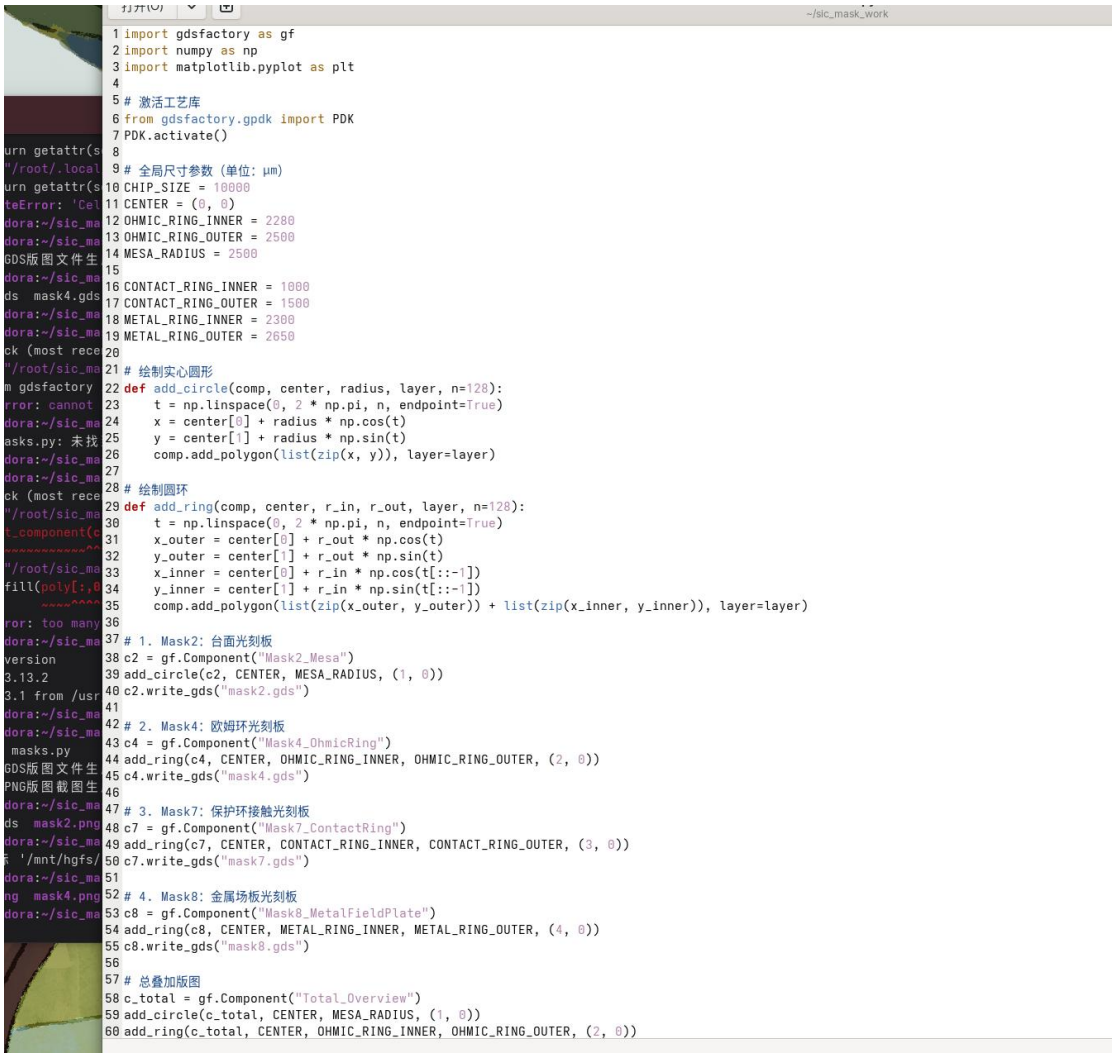


2. 画出刻蚀终端器件的光刻板

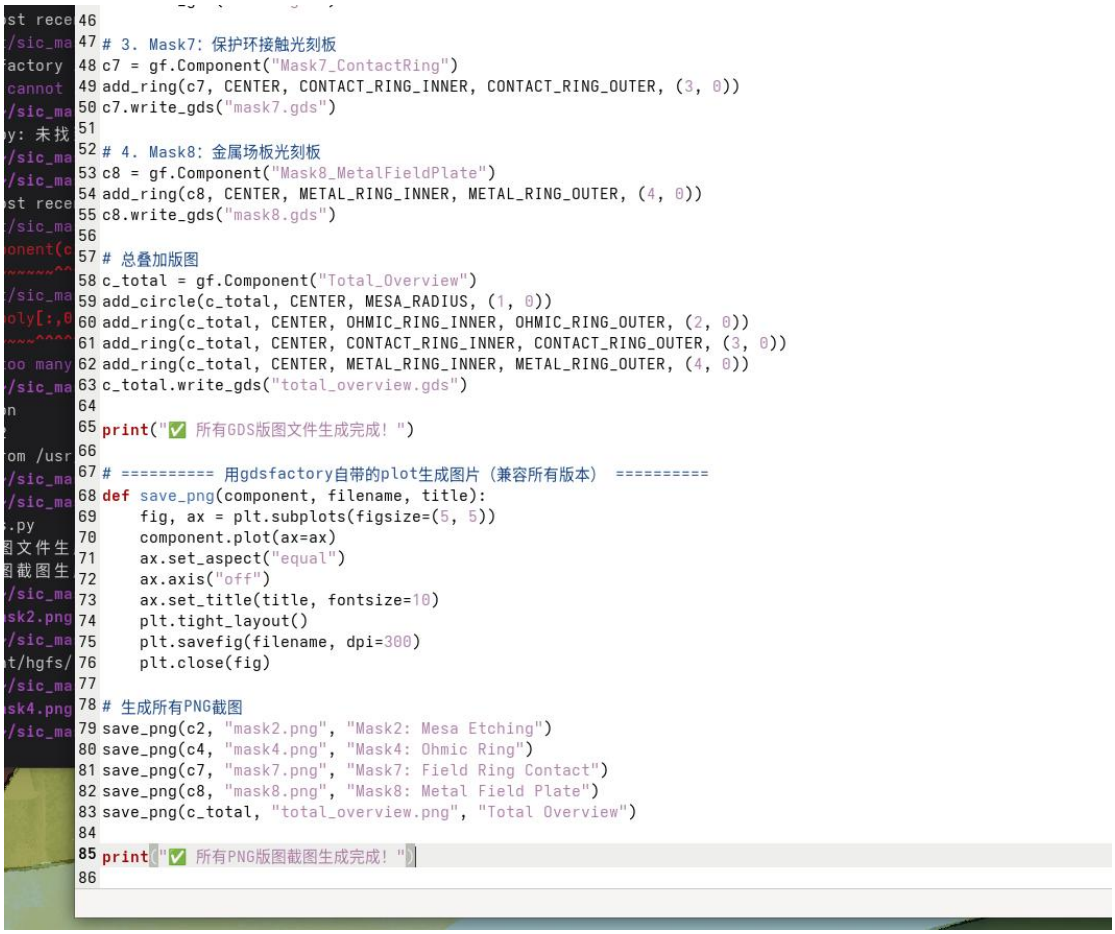
台面光刻板



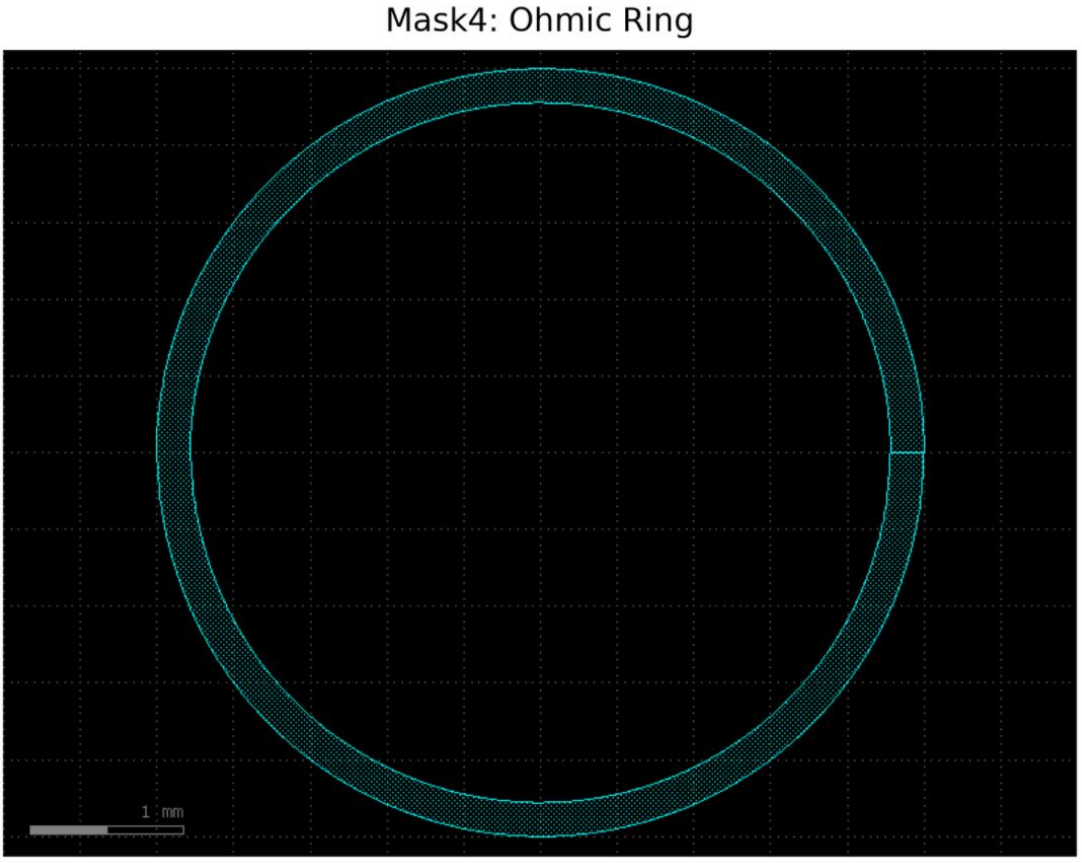
欧姆环光刻板



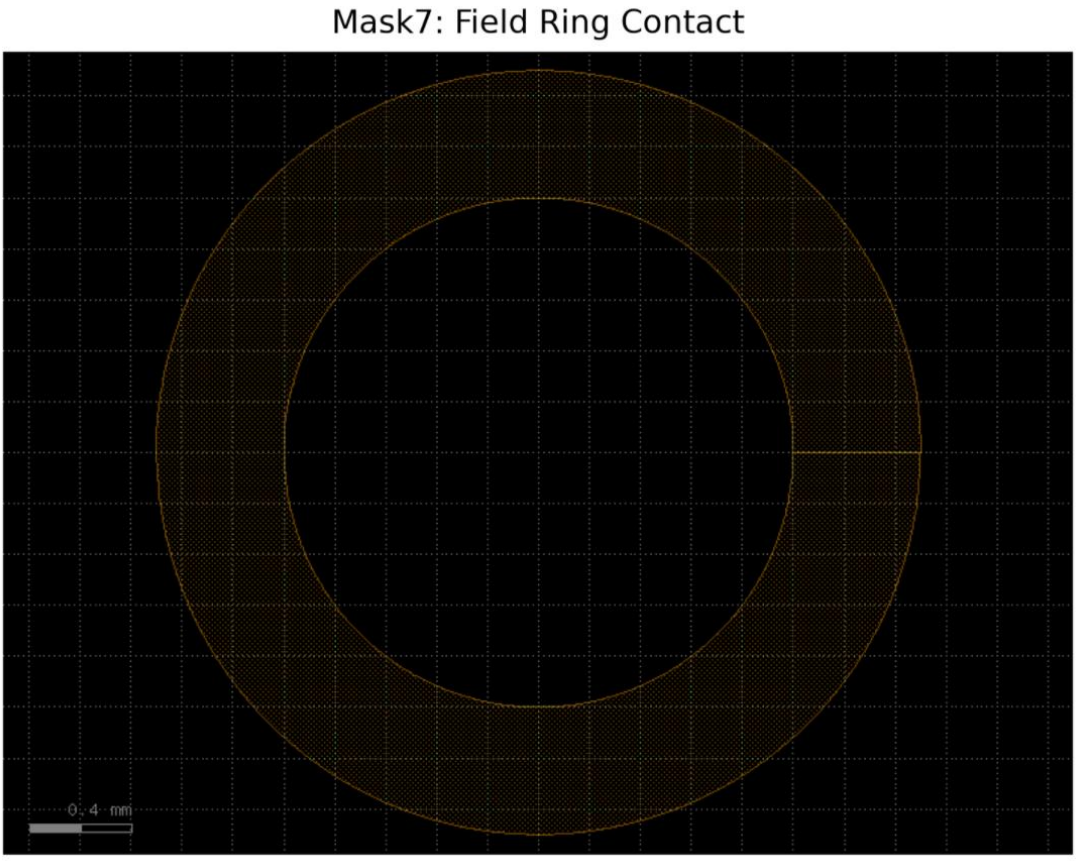
```
import gdsfactory as gf
import numpy as np
import matplotlib.pyplot as plt

# 激活工艺库
from gdsfactory.gpdk import PDK
PDK.activate()

# 全局尺寸参数（单位：μm）
CHIP_SIZE = 10000
CENTER = (0, 0)
OHMIC_RING_INNER = 2280
OHMIC_RING_OUTER = 2500
MESA_RADIUS = 2500

CONTACT_RING_INNER = 1000
CONTACT_RING_OUTER = 1500
METAL_RING_INNER = 2300
METAL_RING_OUTER = 2650

# 绘制实心圆形
def add_circle(comp, center, radius, layer, n=128):
    t = np.linspace(0, 2 * np.pi, n, endpoint=True)
    x = center[0] + radius * np.cos(t)
    y = center[1] + radius * np.sin(t)
    comp.add_polygon(list(zip(x, y)), layer=layer)

# 绘制圆环
def add_ring(comp, center, r_in, r_out, layer, n=128):
    t = np.linspace(0, 2 * np.pi, n, endpoint=True)
    x_outer = center[0] + r_out * np.cos(t)
    y_outer = center[1] + r_out * np.sin(t)
    x_inner = center[0] + r_in * np.cos(t[::-1])
    y_inner = center[1] + r_in * np.sin(t[::-1])
    comp.add_polygon(list(zip(x_outer, y_outer)) + list(zip(x_inner, y_inner)), layer=layer)

# 1. Mask2: 台面光刻板
c2 = gf.Component("Mask2_Mesa")
add_circle(c2, CENTER, MESA_RADIUS, (1, 0))
c2.write_gds("mask2.gds")

# 2. Mask4: 欧姆环光刻板
c4 = gf.Component("Mask4_OhmicRing")
add_ring(c4, CENTER, OHMIC_RING_INNER, OHMIC_RING_OUTER, (2, 0))
c4.write_gds("mask4.gds")

# 3. Mask7: 保护环接触光刻板
c7 = gf.Component("Mask7_ContactRing")
add_ring(c7, CENTER, CONTACT_RING_INNER, CONTACT_RING_OUTER, (3, 0))
c7.write_gds("mask7.gds")

# 4. Mask8: 金属场板光刻板
c8 = gf.Component("Mask8_MetalFieldPlate")
add_ring(c8, CENTER, METAL_RING_INNER, METAL_RING_OUTER, (4, 0))
c8.write_gds("mask8.gds")

# 总叠加版图
c_total = gf.Component("Total_Overview")
add_circle(c_total, CENTER, MESA_RADIUS, (1, 0))
add_ring(c_total, CENTER, OHMIC_RING_INNER, OHMIC_RING_OUTER, (2, 0))
```



```
# 3. Mask7: 保护环接触光刻板
c7 = gf.Component("Mask7_ContactRing")
add_ring(c7, CENTER, CONTACT_RING_INNER, CONTACT_RING_OUTER, (3, 0))
c7.write_gds("mask7.gds")

# 4. Mask8: 金属场板光刻板
c8 = gf.Component("Mask8_MetalFieldPlate")
add_ring(c8, CENTER, METAL_RING_INNER, METAL_RING_OUTER, (4, 0))
c8.write_gds("mask8.gds")

# 总叠加版图
c_total = gf.Component("Total_Overview")
add_circle(c_total, CENTER, MESA_RADIUS, (1, 0))
add_ring(c_total, CENTER, OHMIC_RING_INNER, OHMIC_RING_OUTER, (2, 0))
add_ring(c_total, CENTER, CONTACT_RING_INNER, CONTACT_RING_OUTER, (3, 0))
add_ring(c_total, CENTER, METAL_RING_INNER, METAL_RING_OUTER, (4, 0))
c_total.write_gds("total_overview.gds")

print("✅ 所有GDS版图文件生成完成！")

# ========== 用gdsfactory自带的plot生成图片（兼容所有版本） ==========
def save_png(component, filename, title):
    fig, ax = plt.subplots(figsize=(5, 5))
    component.plot(ax=ax)
    ax.set_aspect("equal")
    ax.axis("off")
    ax.set_title(title, fontsize=10)
    plt.tight_layout()
    plt.savefig(filename, dpi=300)
    plt.close(fig)

# 生成所有PNG截图
save_png(c2, "mask2.png", "Mask2: Mesa Etching")
save_png(c4, "mask4.png", "Mask4: Ohmic Ring")
save_png(c7, "mask7.png", "Mask7: Field Ring Contact")
save_png(c8, "mask8.png", "Mask8: Metal Field Plate")
save_png(c_total, "total_overview.png", "Total Overview")

print("✅ 所有PNG版图截图生成完成！")
```

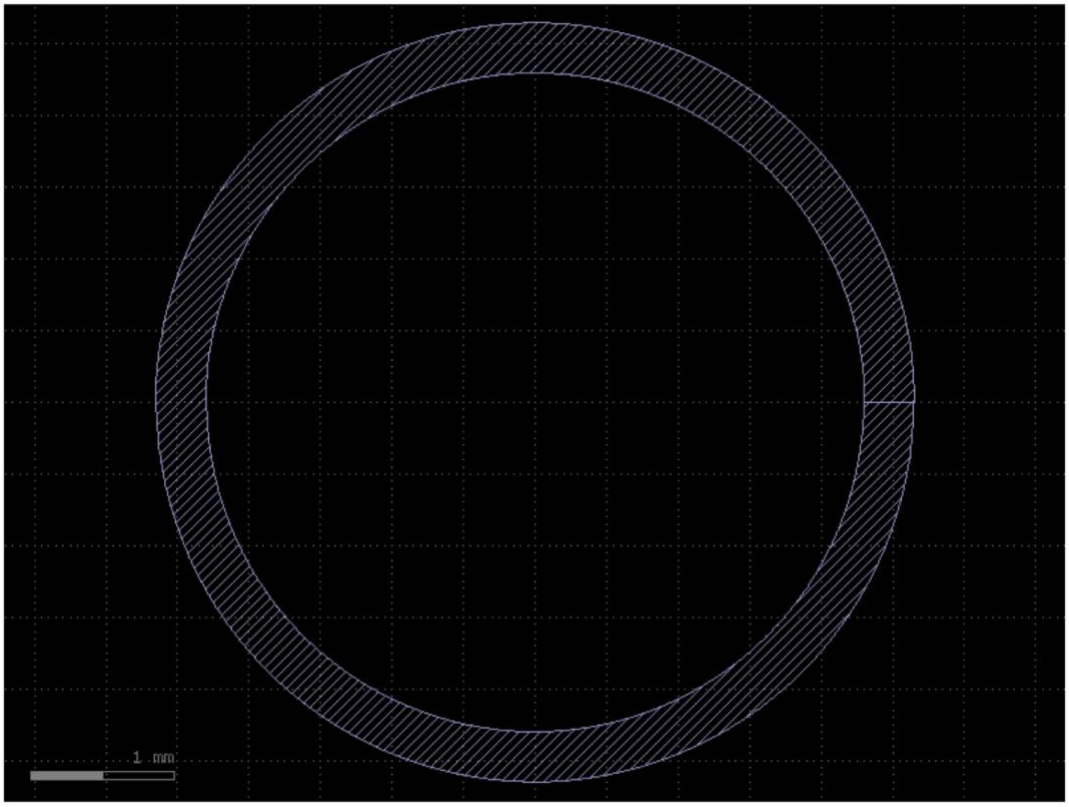


保护环接触光刻板



金属场板光刻板

Mask8: Metal Field Plate



总览版图

Total Overview