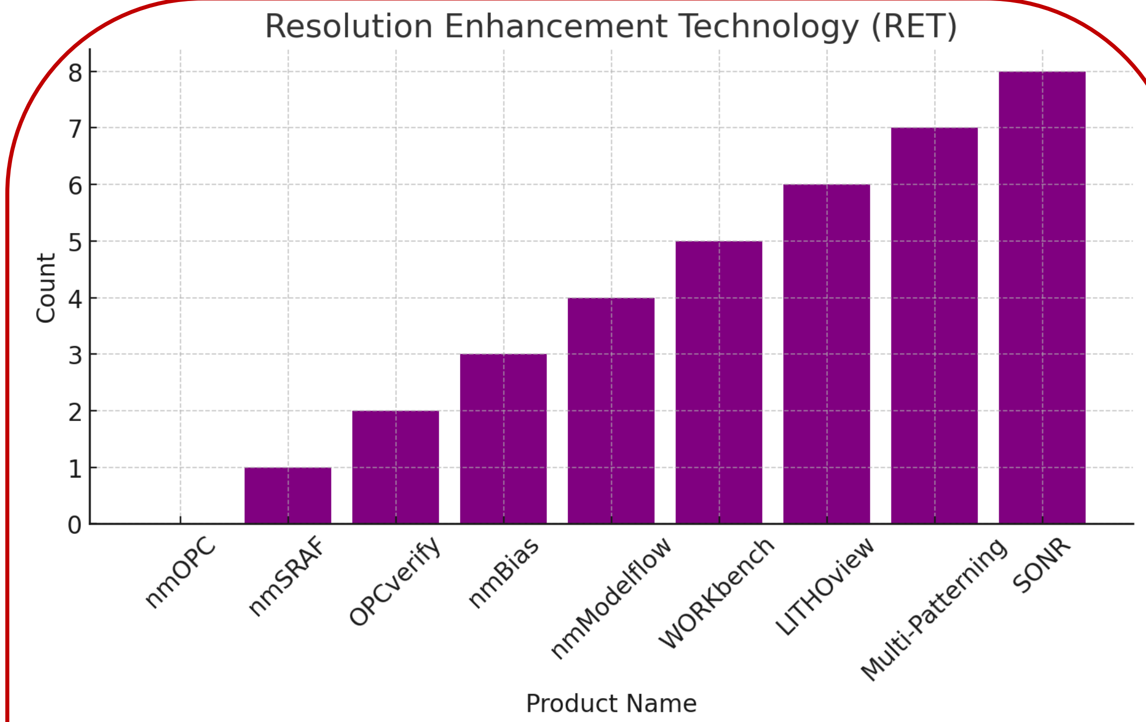
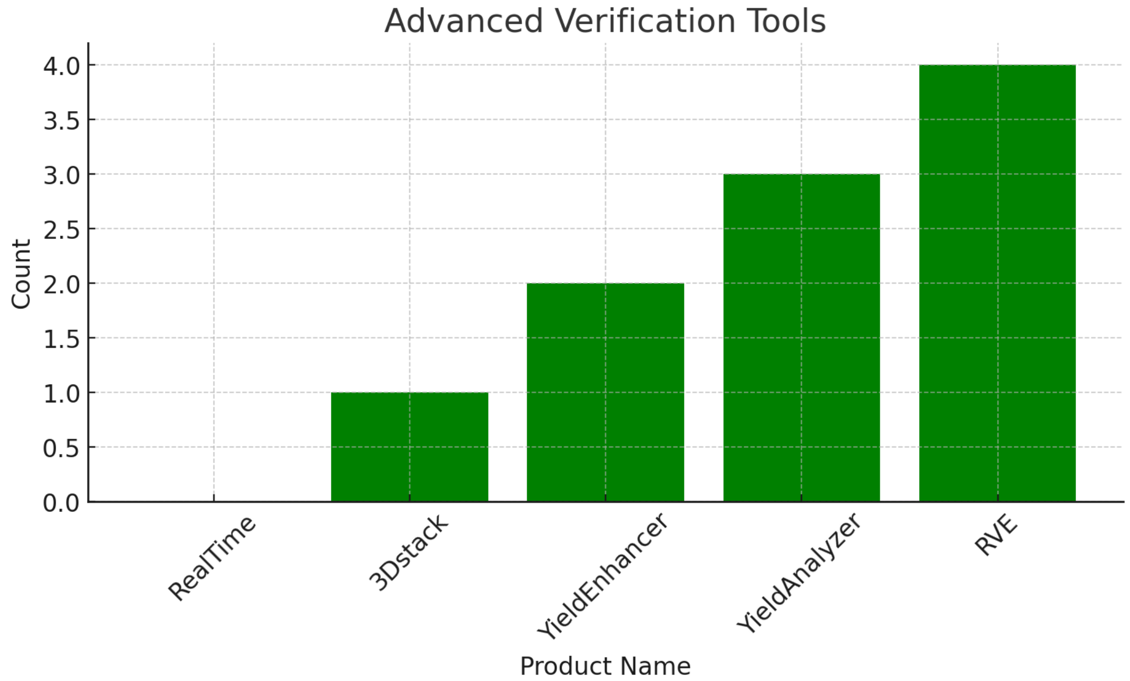
Selected **- Semiconductor Manufacturing-oriented Engineering**

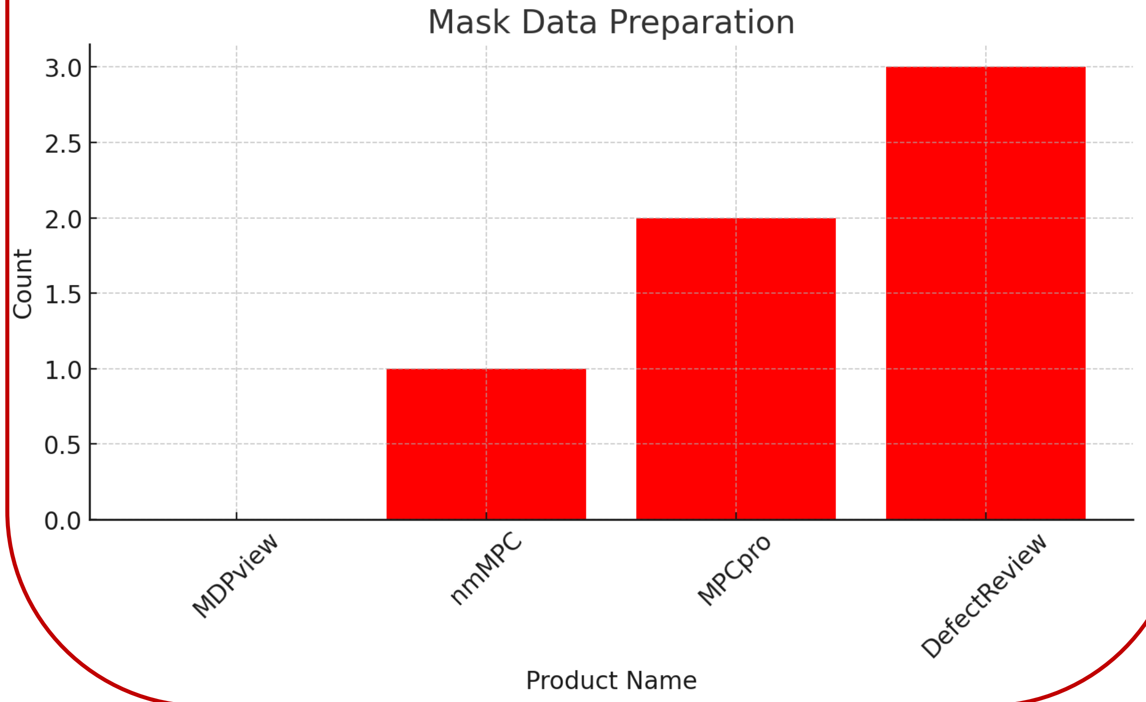
- Focused on manufacturing-oriented engineering using Calibre RET/OPC and lithography workflows for top European foundries.
- Executed ILT/OPC simulation, OPC-Verify checks, multi-patterning analysis, and SRAF optimization for next-generation nodes.
- Built manufacturable layout strategies and improved wafer pattern fidelity through lithography modeling and process-aware tuning.
- Developed and validated MDP/JobDeck structures, including MPC, ModelFlow, LithoView, and Workbench integrations.
- Supported GlobalFoundries, STMicroelectronics, Infineon, and Intel via on-site guidance, remote troubleshooting, and lithography engineering collaboration.

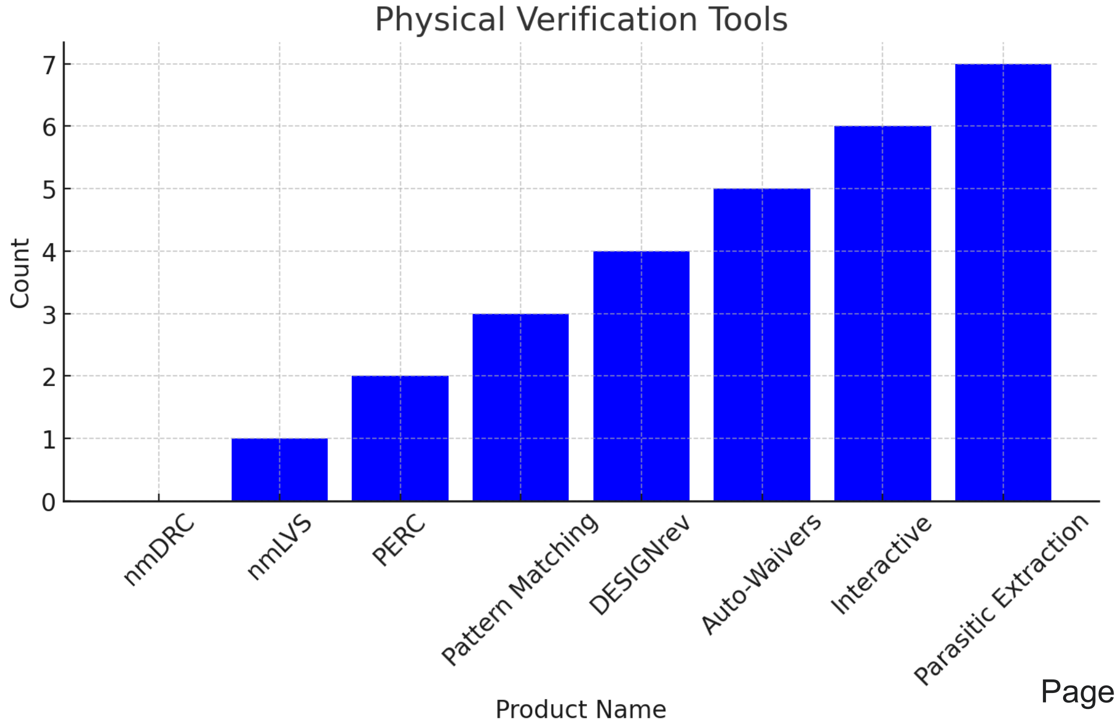
Calibre Products Portfolio

Backend manufacturing

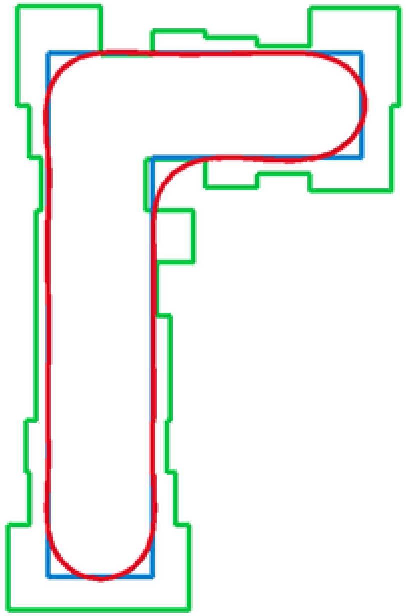








Calibre OPC - Inverse Lithography Technology



A conventional optical proximity correction. The blue Γ-like shape is what chip designers would like print on the wafer, in green is the shape after applying optical proximity correction(OPC), and the red contour is how the shape is actually printed.

In semiconductor device fabrication,
The **inverse lithography technology** (**ILT**) is an approach to photomask design.

An approach to solve an inverse imaging problem: to calculate the shapes of the openings in a **photomask ("source")** so that the passing light produces a good approximation of the **desired pattern ("target")** on the illuminated material, typically a **photoresist.**

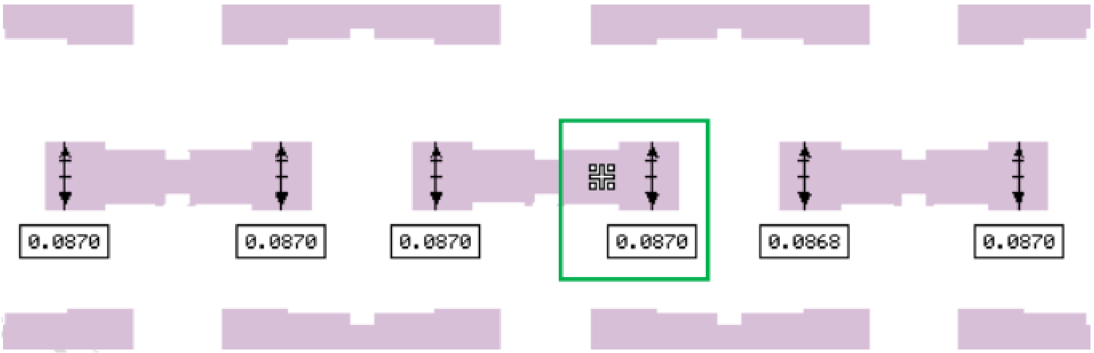
As such, it is treated as a mathematical optimization problem of a special kind, because usually an analytical solution does not exist.

In conventional approaches known as the optical proximity correction (OPC) a "target" shape is augmented with carefully tuned rectangles to produce a "Manhattan shape" for the "source", as shown in the illustration.

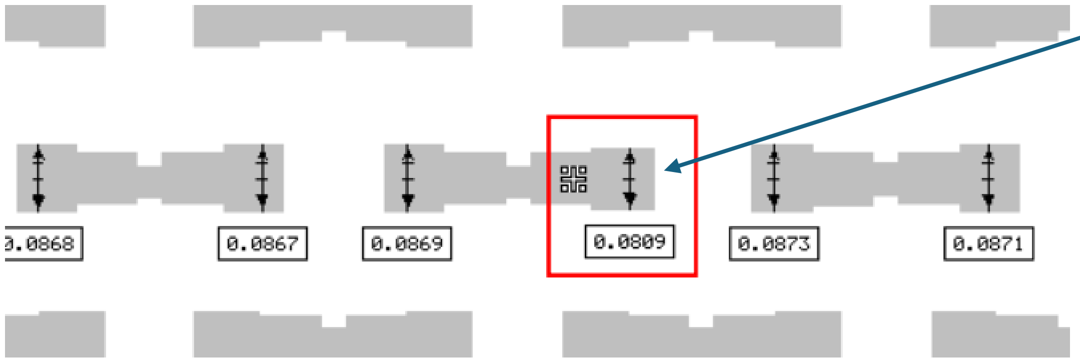
The ILT approach generates curvilinear shapes for the "source", which deliver better approximations for the "target".

Calibre OPC outcome – Inconsistency

OPC -1x1mm² widow clip

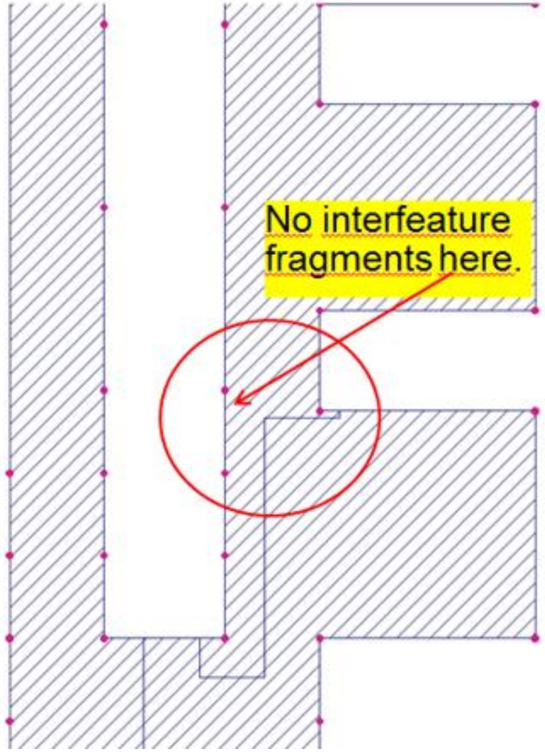


2x2mm² window clip error

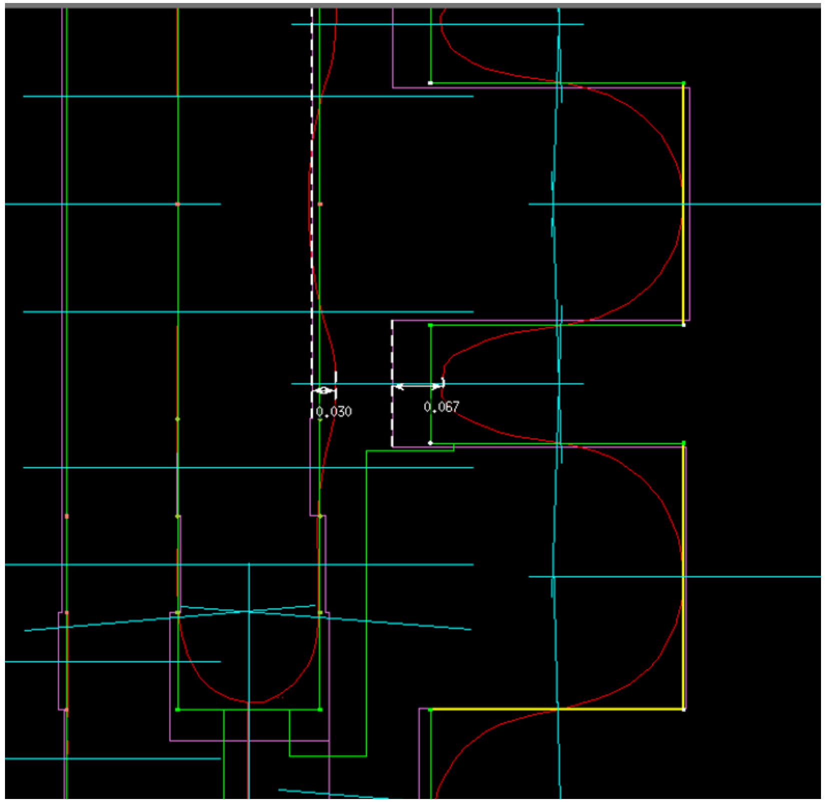


"The edge **post-OPC** feature appears to be undersized. The CUT layer analysis indicates post-OPC mask errors, as expected. The observed CD (~81 nm) deviates significantly from the target CD (~87 nm), with no apparent justification for the observed asymmetric shape."

Calibre OPC – No ripple generation observed due to the absence of inter-feature fragments



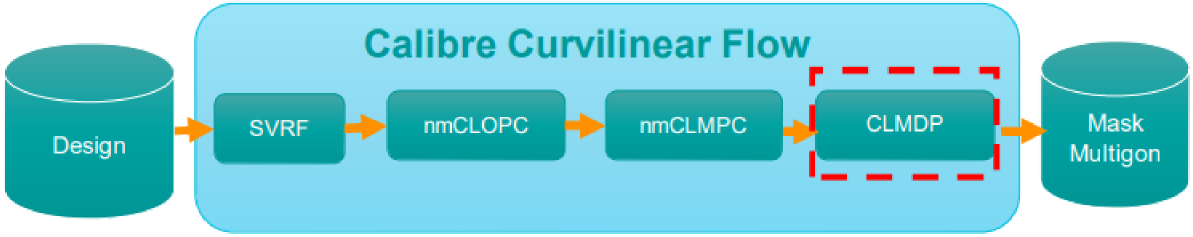
OPC simulation



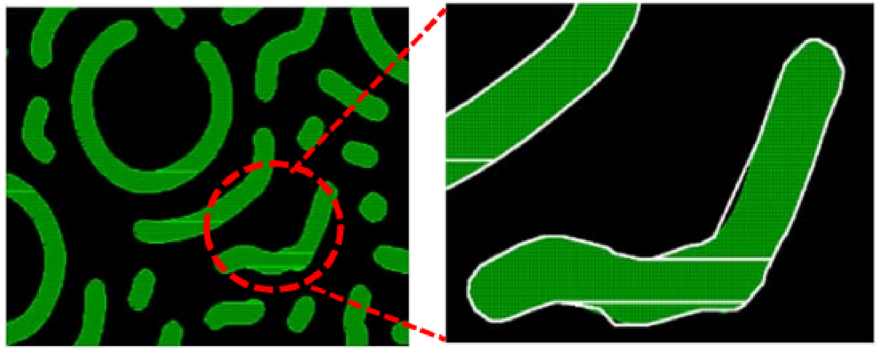
A conventional optical proximity correction. The green Γ-like shape is what chip designers would like printed on the wafer, in purple is the shape after applying optical proximity correction(OPC), and the red contour is how the shape is actually printed.

Calibre Curvilinear MDP Support New Fracture Formats for Multigons

- The Calibre Curvilinear MDP solution enables the support of new Multibeam formats in the fracture and verification flows.



- Calibre Fracture commands are updated to support the new Multibeam fracture formats.
  - `FRACTURE NUFLARE_MBF` now supports version 2.1.
  - `FRACUTER OASIS_MBW` now supports version 3.0.

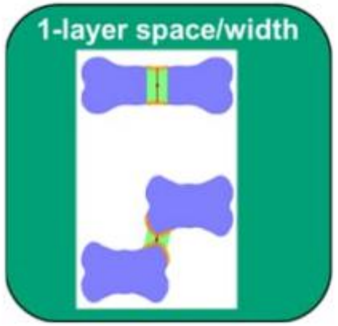


MBW 3.0 fracture output **Zoomed-in**

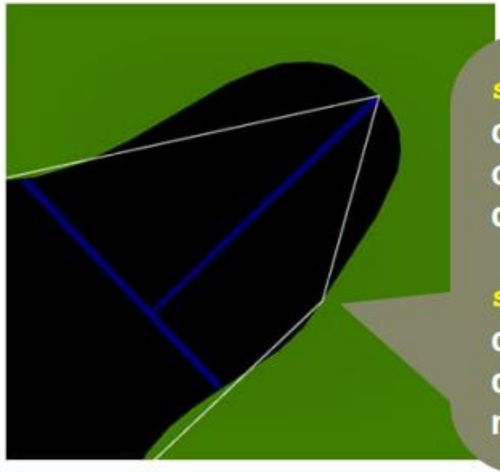
Calibre OPCverify Curvilinear Verification for Multigons

- Calibre OPCverify is updated to address the challenges of curvilinear verification for multigons.
- In 2024.4 release, four spline-based checks are production released to support post-OPC verification on multigon layers.

Distance checks on a multigon layer using `spline_distance` command.



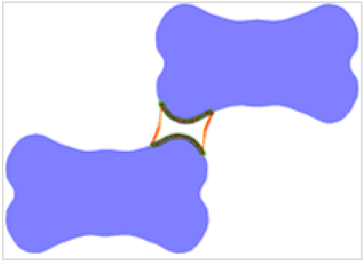
Angle check on a multigon layer using `spline_angle` command.



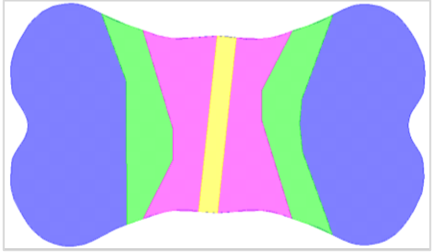
`spline_depth_width` checks minimum width of a multigon using its depth as a constraint.

`spline_dw_ratio` checks the maximum depth-to-width ratio for multigons on a layer.

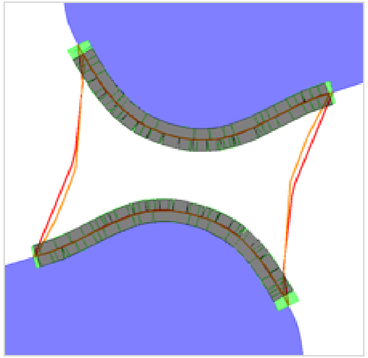
Calibre OPCverify Multigon Support - spline_distance check



Red—SD, non-Manhattan region
Green—SD, non-Manhattan edge
Orange—MD, non-Manhattan region
Black—MD, non-Manhattan edge



Green — separation=90nm
Pink — separation_factor=3.0
Yellow — separation_factor=3.4



Spline distance

Properties

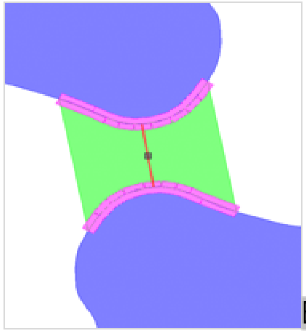
Type	Attribute	Value
User	min	0.00653758

Measure distance

Properties

Type	Attribute	Value
User	min	0.0065102

- An example of Spline distance(SD) check performed on MULTIGON.
  - In comparison, Measure distance(MD) is performed on PWL(converted from Multigon with dev=1dbu).

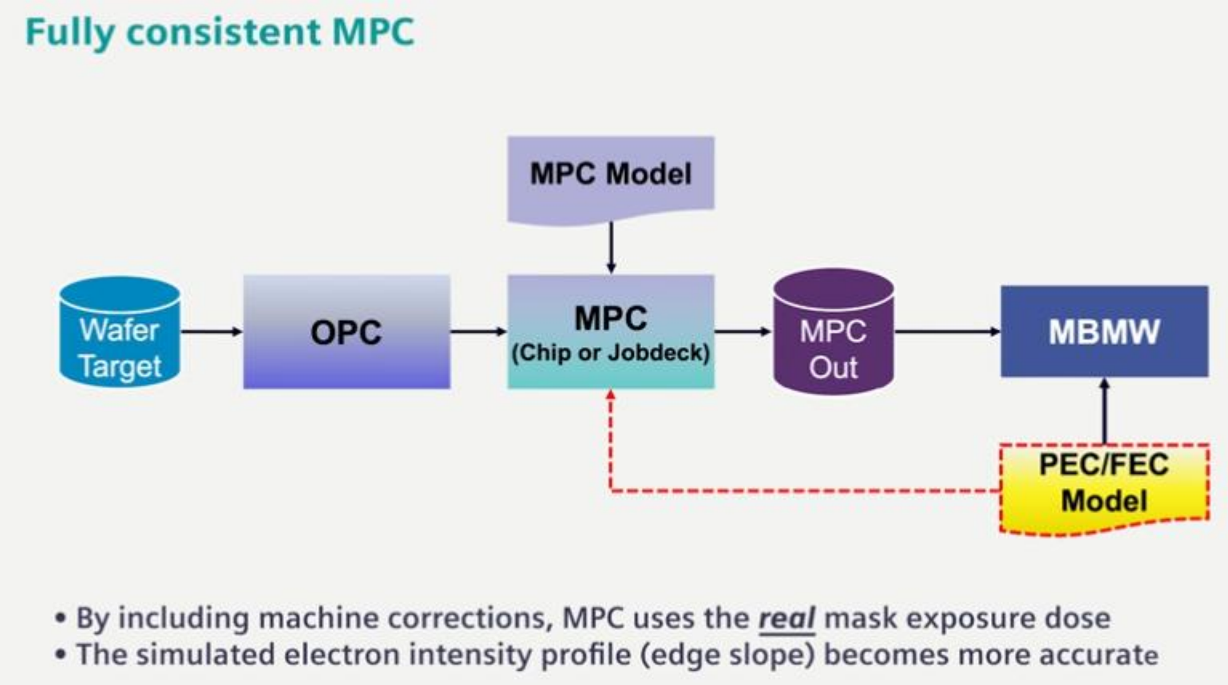


Pink—non-Manhattan edge
Green—non-Manhattan region
Red—pinpoint gauges
Black—pinpoint markers

Error marker

Calibre Mask Process Correction(MPC)

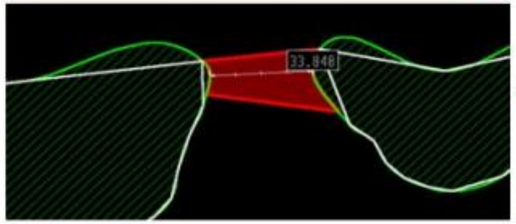
The Calibre Mask Process Correction family of **rule and model-based products** is used in advanced photomask manufacturing to correct for **systematic mask lithography** and **process error sources** to ensure that the mask critical dimension signature is within specification



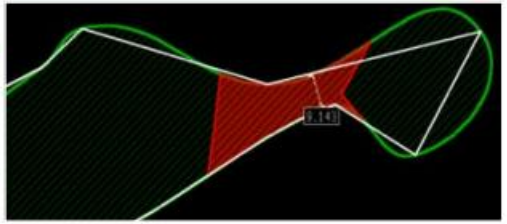
Calibre Standalone Curvilinear Mask Rule Checks (MRC) for Multigons

- Calibre provides new curvilinear MRC methodologies to support the requirements of leading-edge mask manufacturing through key innovations in native curvilinear data handling and representation techniques.
- New standalone SVRF commands are available for curvilinear MRC checks on multigons.
  - `RET SPLINE_ANGLE` checks the angle of splines on a multigon layer.
  - `RET SPLINE_DEPTH_WIDTH` checks the minimum width of a multigon shape.
  - `RET SPLINE_DISTANCE` measures the distance between spline sections on multigon layers.
  - `RET SPLINE_DW_RATIO` computes the depth-to-width ratio for multigon shapes.

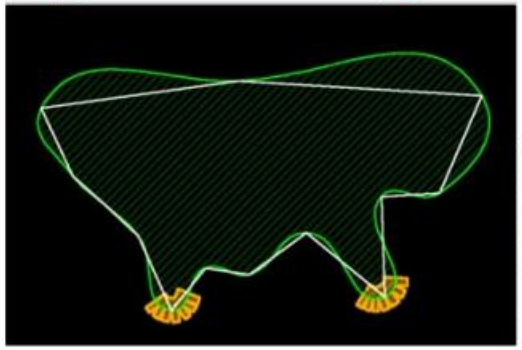
External Distance Check Example



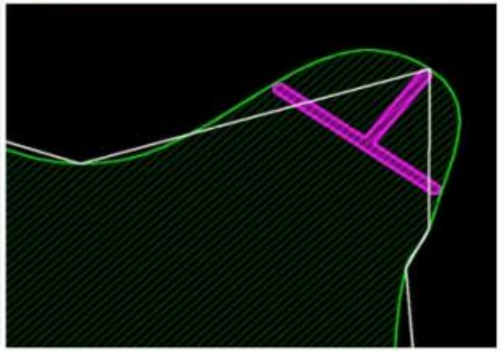
Internal Distance Check Example



Spike Detection Example

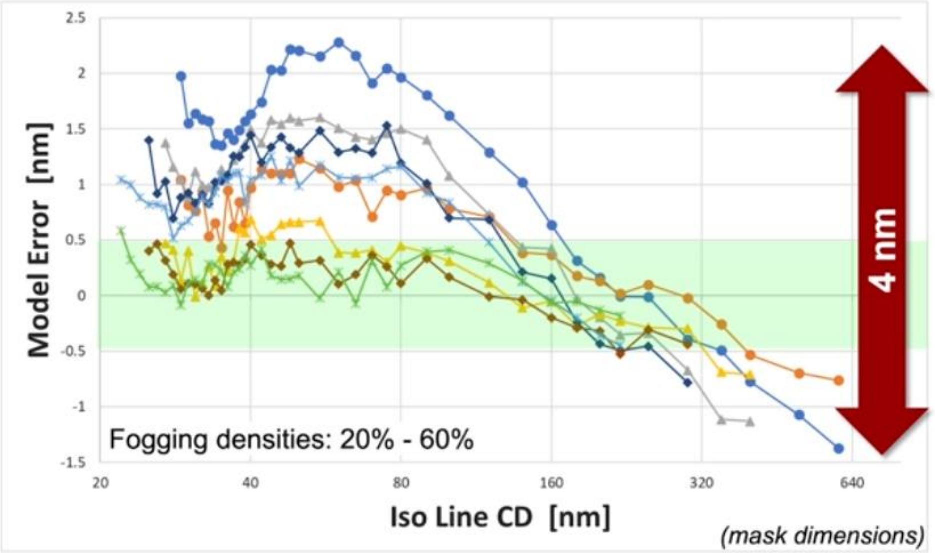


Depth-Width Check Example

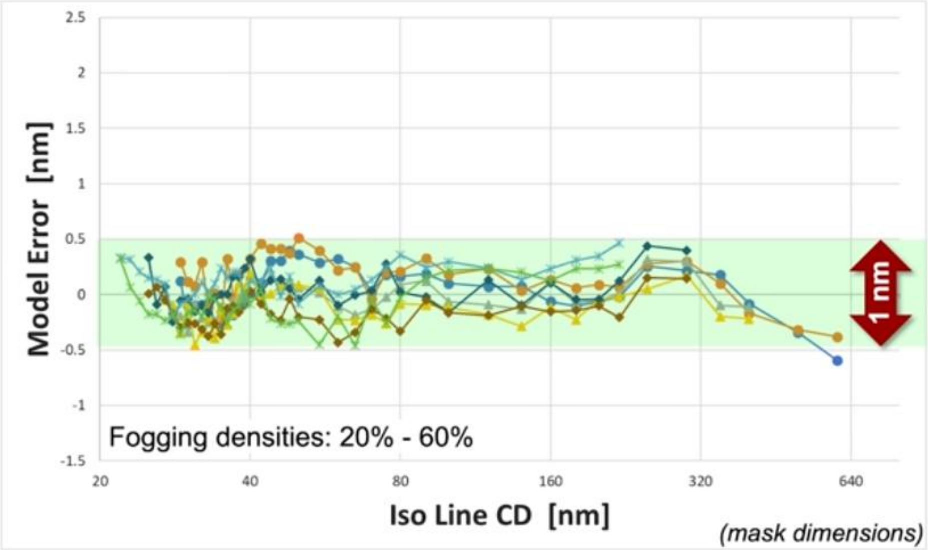


Calibre Mask Process Correction(MPC) – Reduces the CD model error

IsoLn CD model error when ignoring e-beam writer corrections for Fogging

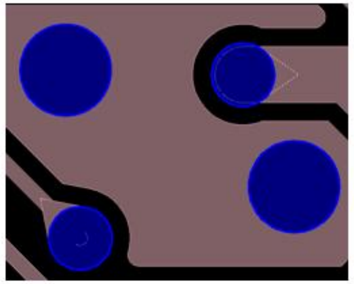
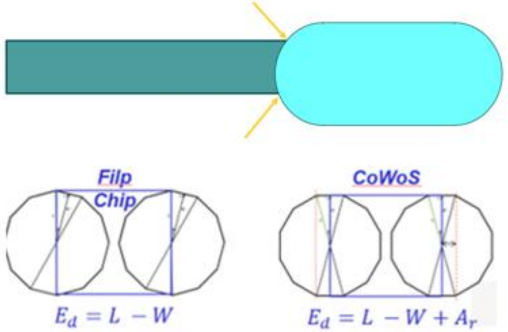
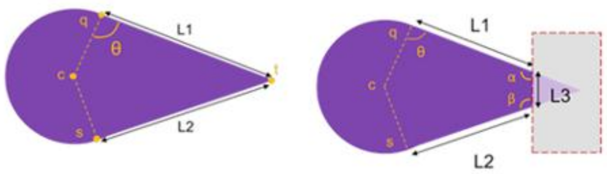
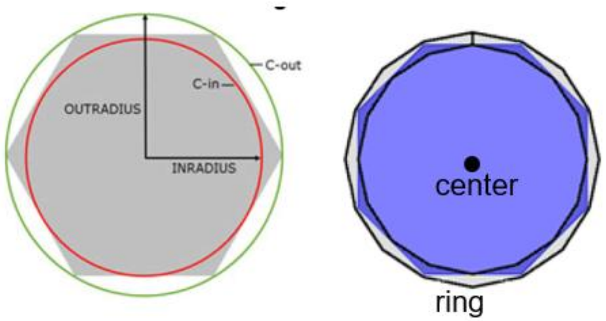
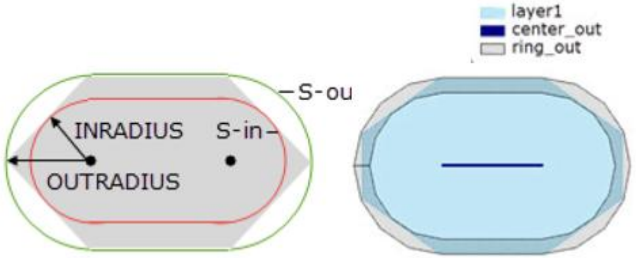
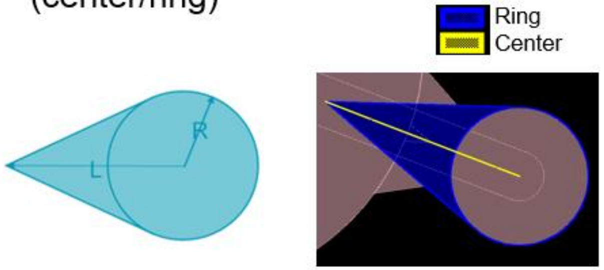


IsoLn CD model error reduced from over ± 2 nm to ± 0.5 nm when including e-beam writer corrections for Fogging



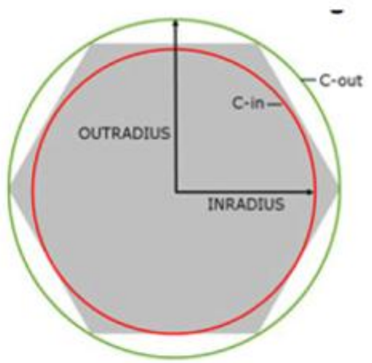
Calibre Capabilities to Simplify Irregular Shape Checking

Photonics, through-silicon via, & packaging checks

	Circle	Stadium	Teardrop
Recognize & separate	• INSIDE CELL [NOT] CIRCLE 	• INSIDE CELL [NOT] STADIUM  	• INSIDE CELL [NOT] TEARDROP 
Check Dimension & Ref Output	• DFM CIRCLE ANALYZE • DFM CIRCLE ANALYZE (center/ring)  	• DFM STADIUM ANALYZE • DFM STADIUM ANALYZE (center/ring)  	• DFM TEARDROP ANALYZE • DFM TEARDROP ANALYZE (center/ring) 

Calibre Capabilities to Simplify Irregular Shape Design Rule Check
DFM Circle Analyze

```
VARIABLE minimum_radius 1.0
VARIABLE maximum_radius 2.0
VARIABLE minimum_curvature_width 0.1
VARIABLE maximum_curvature_width 0.2

LAYER M1  54 350

// To obtain a Circle with a width of 0.14um (>=0.14)
circle_props = DFM CIRCLE ANALYZE M1
circle_ringL14 = DFM PROPERTY circle_props
                [inrad   = PROPERTY(circle_props, INRADIUS)]
                [outrad  = PROPERTY(circle_props, OUTRADIUS)]
                [annulus_in = PROPERTY(circle_props, ANNULUS_INRADIUS)]
                [ratio   = PROPERTY_REF(inrad)/PROPERTY_REF(outrad)] > 0.9
                [enclosing_width = 2*(PROPERTY(circle_props, OUTRADIUS))]
                [width   = (PROPERTY_REF(annulus_in)!=0) ? (PROPERTY_REF(inrad)+PROPERTY_REF(outrad))/2 –
PROPERTY_REF(annulus_in): PROPERTY_REF(inrad)+PROPERTY_REF(outrad)] >= minimum_curvature_width <=
maximum_curvature_width
                [radius  = (PROPERTY_REF(annulus_in)!=0) ? (PROPERTY_REF(width))+PROPERTY_REF(annulus_in):
(PROPERTY_REF(inrad)+PROPERTY_REF(outrad))/2] >= minimum_radius <= maximum_radius

DFM_out { DFM RDB circle_ringL14 "dfm.rdb"  ALL CELLS CHECKNAME "%_l_“ }
```



Python Code and its output: S-bend for SiN waveguide (Cubic Bezier segments)

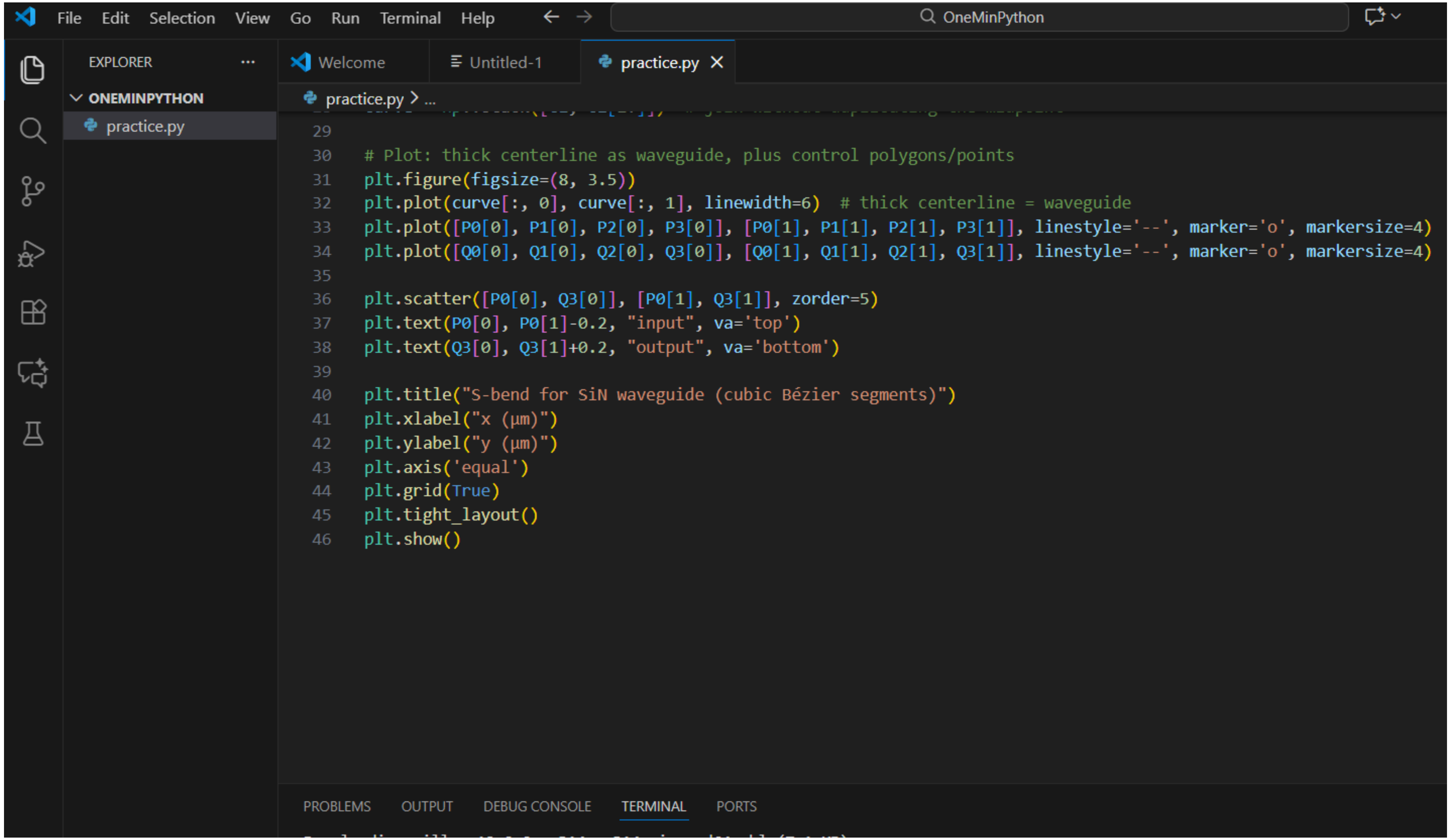
```python
# S-bend (SiN waveguide) — cubic Bézier 2-segment example
import numpy as np
import matplotlib.pyplot as plt

# Segment 1 control points (P0..P3)
P0 = np.array([0.0, 0.0])
P1 = np.array([2.0, 0.0])
P2 = np.array([3.0, 1.5])
P3 = np.array([5.0, 1.0])  # mid point

# Segment 2 control points (Q0..Q3)
Q0 = P3
Q1 = np.array([7.0, 0.5])
Q2 = np.array([8.0, 2.5])
Q3 = np.array([10.0, 2.0])

def cubic_bezier(A, B, C, D, n=300):
    t = np.linspace(0, 1, n)
    return (
        ((1 - t)**3)[:, None] * A +
        (3 * ((1 - t)**2 * t))[:, None] * B +
        (3 * ((1 - t) * t**2))[:, None] * C +
        (t**3)[:, None] * D
    )

c1 = cubic_bezier(P0, P1, P2, P3, n=200)
c2 = cubic_bezier(Q0, Q1, Q2, Q3, n=200)
curve = np.vstack([c1, c2[1:]])  # join without duplicating the midpoint
```

(Code continued)

Cubic Bezier curve in mathematics

$B(t) = (1-t)3P0 + 3t(1-t)2P1 + 3t2(1-t)P2 + t3P3, 0 \le t \le 1$

Python Code and its output: S-bend for SiN waveguide (Cubic Bezier segments)



(Code output)

Python Code and its output: S-bend for SiN waveguide (Cubic Bezier segments)