JM3 Journal of Micro/Nanopatterning, Materials, and Metrology

RESEARCH PAPER

# You do not need 1 nm contours for curvilinear shapes: pixel-based computing is the answer

**Abhishek Shendre,* Aki Fujimura, Mariusz Niewczas, and Tom Kronmiller**
D2S Inc., San Jose, California, United States

**ABSTRACT.** Enabled by multibeam mask writing, curvilinear free-form inverse lithography technology (ILT), and graphics processing unit acceleration, curvilinear masks are quickly becoming the norm in leading edge masks, whether for 193i or for extreme ultra violet (EUV), particularly for contact and via layers. An industry standard for compactly representing curvilinear shapes is being developed for Semiconductor Equipment and Materials International through an industry working group. Bezier and B-spline "Multigon" formats are proposed to augment the piecewise linear polygons that are supported today. Whether these infinite-resolution curvilinear formats are used or piecewise linear polygons are used, there is a question of what constitutes a high enough vertex density to be of some predefined accuracy requirement. With these infinite-resolution curvilinear formats, the vertex density would be lower than with piecewise linear polygons for a particular accuracy requirement. But it is still useful to know what density is theoretically sufficient. We explore the concept of rasterization and the mathematical dual between contours and pixel dose arrays given a particular known resolution limit. We further argue that curvilinear ILT, practically speaking, is all computed in the pixel domain. And all curvilinear masks, with the notable exception of MWCO masks for 193i, are written with multibeam machines using pixel dose arrays. We further argue that all images taken of the resulting masks, whether for inspection, disposition, or for metrology are pictures taken as pixel dose arrays of some resolution with some image processing afterward. Information theory is a branch of computer science that, among other things, gives insight on how much data are sufficient to represent any particular information content. Generally, the field covers the idea of digitizing the analog world to some known limit of resolution. Rasterization is digitalization of images that converts from contours, be it piecewise linear polygons, or some infinite resolution curves, to pixel doses of some pixel size and dose range. Contouring is the converse, going from pixel doses to geometric space. By understanding information theory, how curvilinear mask shapes are computed, and how curvilinear mask shapes are generated on the mask, we compute the theoretical limit of how much data are required to represent 193i and EUV curvilinear masks.

 [DOI: 10.1117/1.JMM.23.1.011202]

**Keywords:** curvilinear; curvilinear masks; curvilinear data format; information theory; pixel-based computing

Paper 23048SS received Jun. 27, 2023; revised Oct. 7, 2023; accepted Oct. 11, 2023; published Nov. 24, 2023.

## 1 Introduction: Curvilinear Masks Representation Overview

There are various ways in which semiconductor masks can be represented.[1–8] Having an efficient file format is essential to make sure that the mask processing data path is not overloaded. This has

*Address all correspondence to Abhishek Shendre, ashendre@design2silicon.com

2708-8340/2023/$28.00 

encouraged the creation of different file formats, such as GDSII[9] and Oasis[10] to represent the mask data. Mask data preparation (MDP) is a very crucial step in the mask making industry. The multibeam mask (MBM) writers have enabled curvilinear mask writing.[11,12] With the wafer process windows known to improve substantially with free-form curvilinear inverse lithography technology (ILT),[2] representing and computing with curvilinear shapes in MDP has gained in importance.[13] File sizes and computational complexity have become an increasing concern for the mask industry in the curvilinear era, prompting the industry to develop a Semiconductor Equipment and Materials International (SEMI) standard.[4]

The various file formats used for representing curvilinear masks can be broadly classified into three categories as depicted in Fig. 1: 1. piecewise-linear format, 2. curvilinear format, and 3. pixel-based format. Each of these categories has its own characteristics and can provide some benefits to represent the underlying mask data. Any of the representations can be made accurate within some prespecified specification as required by the consumers of the data.

### 1.1 Piecewise-Linear Formats

The legacy piecewise-linear formats, such as GDSII or OASIS,[9,10,13,14] have a widespread adoption in the electronic design automation industry. Early file formats were mostly representing rectangular and Manhattan shapes. Since the mask data were mainly arrays or repetitions of simple shapes, hierarchical representation avoids unnecessary repetition of the same data in formats like GDSII. As the technology nodes advanced, the need for optical proximity correction (OPC) made each instance of the mask shapes different based on the lithographic context around that instance, reducing substantially the ability to deploy hierarchy to compress data. Repeated chips on a reticle may be represented by hierarchy for 193i chips, but complex effects of extreme ultra violet (EUV) threatens to require different OPC/ILT for each chip along the $Y$ axis, too. OPC also gave rise to an increase in the number of vertices required to represent the data as shown in Fig. 2. This in turn created the need for data compression. Formats like the OASIS format took information-based data compression a little further using less bits to represent only differences in between adjacent coordinates and storing $X$ or $Y$ coordinates in alternating Manhattan coordinate sequences as first introduced for the DEF format.[15]

With MBM writers, curvilinear masks can be written in the same time with the same precision as Manhattan masks. Since wafer quality is better with curvilinear masks output by curvilinear ILT, curvilinear masks are projected to become common among leading edge masks for

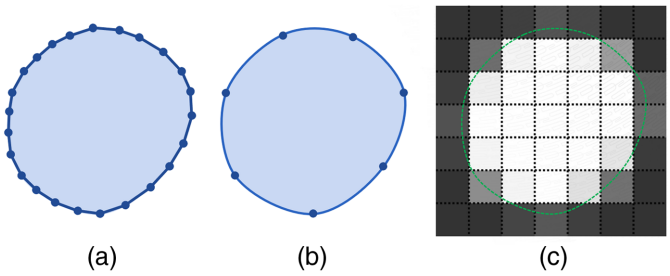


**Fig. 1** Classification of data representation formats for curvilinear shapes: (a) piecewise linear formats like GDS/OASIS; (b) curvilinear formats that may be based on Bezier/splines; and (c) pixel-based formats such as Tiff or Bitmap file formats.

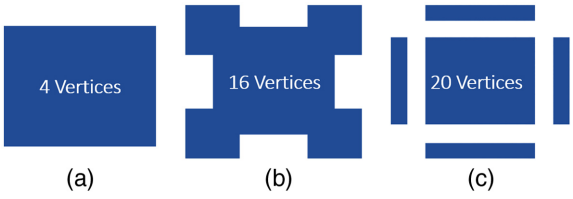


**Fig. 2** Increase in vertex count with OPC: (a) simple mask shape, which gets converted to (b) or (c) with OPC-based enhancements and subresolution assist features.

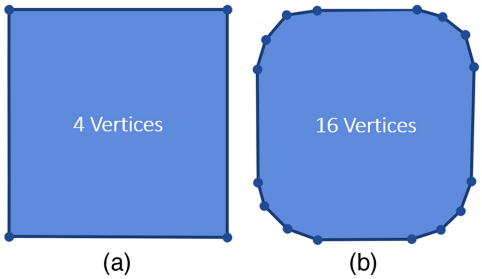


**Fig. 3** Example showing increase in vertex count for curvilinear designs: (a) simple Manhattan data and (b) piecewise linear representation of curvilinear data.

both 193i and for EUV.[16] Curvilinear shapes[17] represented as piecewise-linear polygons cause a data explosion as shown in Fig. 3.

In order to understand the implication of piecewise linear formats, we can just try to figure out the file size for different circle array patterns as described in Fig. 4. A circle array, arranged in this way, represents a good upper bound on the packing density of vertices required to express any manufacturable set of curved shapes on any mask. With this case, we can analytically estimate the number of vertices per circle and number of circles per $\mu m^2$ of mask area. Since we cannot guarantee that every shape in an ILT mask for EUV process will be identical, we must assume a representation of flat (no hierarchy) data. We can still consider representation with deltas as defined in OASIS file format. So say each vertex takes 2 bytes of memory to save on disk. The estimated file size for some representative test cases is shown in Table 1 for a full reticle (104 mm × 132 mm) design.

The circle array in Fig. 4 is placed such that three nearby circles form an equilateral triangle with side equal to the pitch of the design. Such a triangle covers $1/6^{\text{th}}$ of the area of each circle that it goes through. We can also say that each such triangle covers $3 \times 1/6^{\text{th}} = 1/2$ the perimeter of the entire circle and thus half the vertices.

The number of vertices in a circle can be calculated using the following equation:

$$\text{number of vertices per circle} = \frac{\text{perimeter}}{\text{segment length}} = \pi \times \frac{\text{diameter}}{\text{segment length}}. \tag{1}$$

Since each equilateral triangles cover half a circle, the number of circles per $\mu m^2$ can be computed using the following equation:

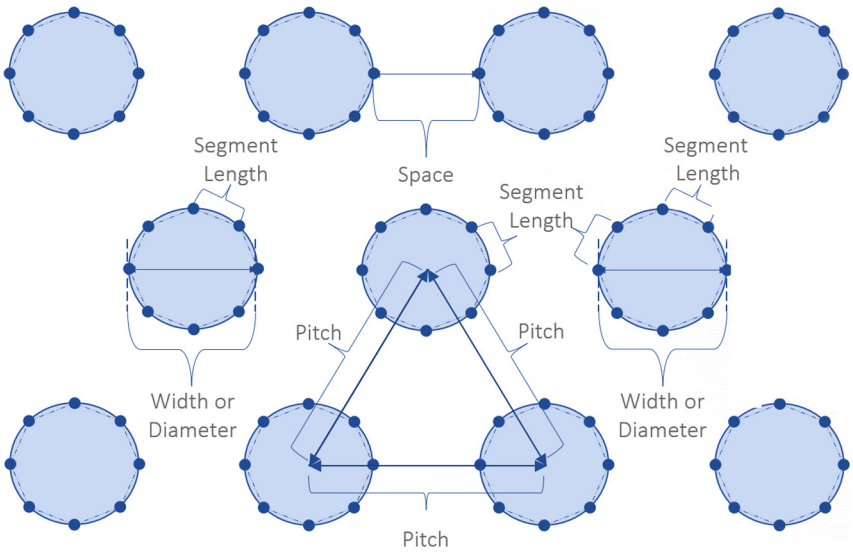


**Fig. 4** Circle array pattern representing curvilinear design with parameter definition for diameter, space, pitch, and segment length.

**Table 1** File size estimations for circle array shown in Fig. 4 with different parameters.

| Circle diameter (nm) | Pitch (nm) | Segment length (nm) | Vertex density (Count per $\mu m^2$) | Full reticle file size (TB) |
|---|---|---|---|---|
| 100 | 150 | 1 | ~16,000 | ~440 |
| 100 | 150 | 5 | ~3200 | ~88 |
| 100 | 150 | 10 | ~1600 | ~44 |
| 100 | 150 | 20 | ~800 | ~22 |
| 100 | 200 | 1 | ~9000 | ~250 |
| 100 | 200 | 5 | ~1800 | ~50 |
| 100 | 200 | 10 | ~900 | ~25 |
| 100 | 200 | 20 | ~450 | ~12.5 |
| 150 | 225 | 1 | ~11,000 | ~300 |
| 150 | 225 | 5 | ~2200 | ~60 |
| 150 | 225 | 10 | ~1100 | ~30 |
| 150 | 225 | 20 | ~550 | ~15 |
| 150 | 300 | 1 | ~6000 | ~160 |
| 150 | 300 | 5 | ~1200 | ~32 |
| 150 | 300 | 10 | ~600 | ~16 |
| 150 | 300 | 20 | ~300 | ~8 |

$$\text{number of circles per}\,\mu m^2 = \frac{1}{2} \times \frac{1000\ \text{nm} \times 1000\ \text{nm}}{\text{area of equilateral triangle}} = \frac{1}{2} \times \frac{1000\ \text{nm} \times 1000\ \text{nm}}{\frac{\sqrt{3}}{4} \times \text{pitch\_nm}^2}. \tag{2}$$

Using Eqs. (1) and (2) and considering 2 bytes per vertex for some file size estimations, we computed Table 1.

Based on the information from Table 1, a reasonable upper-bound on an EUV mask with curvilinear geometries can take >100 Terabytes of data to represent a full reticle design with no hierarchy. This is certainly too expensive for the data path in MDP.

### 1.2 Curvilinear Formats

In order to assuage the effects of curvilinear design on file size, the mask industry is currently working on a Bezier-based curve format to be adopted as a SEMI standard[4] that is suitable to represent curvilinear shapes. An example of such formats is shown in Fig. 5(b).

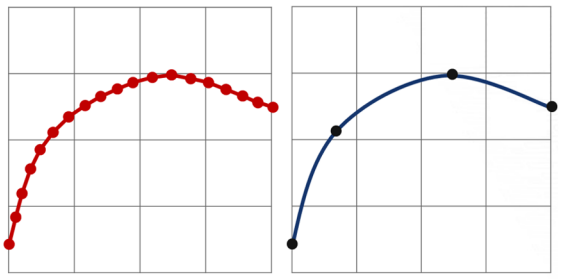


**Fig. 5** Section of a curvilinear shape: (a) piecewise-linear format and (b) Bezier/spline-based curve format.

Bezier curve is a parametric curve that utilizes control points to define a smooth curve. The curve format will have some control points on the curve and some additional control points that may be either implicit (i.e., based on curve control points and some predefined parameters) or explicit. The implicit Bezier-based curvilinear format has huge potential to reduce the effective vertex count required to represent the shapes. In curvilinear format's case, the vertex counts just represent the number of explicit control points required to faithfully represent a curvilinear shape. Interpreting the data from Table 1 and assuming that we can use implicit Bezier-based curvilinear format, it can be estimated that such a format will have effective vertex density (or control point density) in the order of the data corresponding to 10 or 20 nm segment length. Thus file sizes may be limited to average case of 10 to 30 TB, which is about 10× reduction compared to piecewise linear formats.

### 1.3 Pixel Format

Semiconductor masks can also be represented using pixels. Each pixel value in such a case would just represent the percentage area of the pixel covered by the shape that the pixel needs to represent. Figure 1(c) represents such an example of pixel-based representation. There are many formats like TIFF or BITMAP format that can be used to represent pixels. In general, pixel-based representation of mask data is used for eBeam or ILT simulations. Although it represents the natural language of MBM writers, it is too expensive in terms of file size.

There are approximately $(104\ \text{mm} \times 132\ \text{mm}/(16\ \text{nm})^2) \approx 50 \times 10^{12}$ pixels in an entire reticle if we consider pixel size of 16 nm. If we use 2 bytes to represent each pixel, then a pixel-based format without any compression may take 100 TB of file size. The good thing about pixel format is that this is a constant number irrespective of design complexity. This suggests that if the eventual target is to write masks using MBM writers, it may be more useful to use a pixel-based format instead of piecewise linear formats. An alternative to pixel format would be a format that is pixel-based-computing-aware and uses it to represent the data using piecewise curvilinear format thus getting a greater reduction in file size to a value <10 TB.

## 2 Fundamentals of Pixel-Based Computing

Pixel-based computing is the basis of many simulation softwares and graphical tools. It has been used in computer aided design tools for various physical simulations. The most advanced semiconductor masks written today already use pixel-based computing, accelerated by graphics processing units (Fig. 6).

### 2.1 Rasterization

The MBM writers get the input design in some geometry format. In order the write the mask, the writer needs to know the amount of dose it needs to project at a certain location on mask. To compute the dose amount, the machine internally runs a process called rasterization. Rasterization is a process that converts the geometrically expressed shapes into a pixel map based on the area of the pixel covered by the shape as shown in Fig. 7.

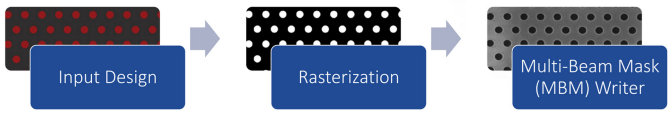


**Fig. 6** Processing steps in MBM writers.

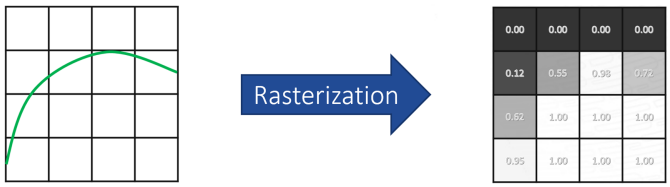


**Fig. 7** Rasterization of an input contour into pixel grid based on area coverage.

### 2.2 Pixel Domain Sampling

Rasterization can also be defined as digital sampling of vector shapes. Pixel domain sampling has roots in the digital signal processing. Just like audio signals are sampling data in 1D, rasterization is sampling data into a 2D map of pixels. So it must follow the Nyquist rate to capture all relevant information. It also adheres to the information theory that, among other things, says that the minimum volume of data required to represent something, is indeed, limited by the amount of information, contained in that data.

We can say contour geometry and pixels are duals, and whatever you can do in one domain that can also be done in the other. The duality holds true only if we follow the resolution limit that is determined by the Nyquist–Shannon sampling theorem. It seems that the curvy geometries require a lot of data to be represented faithfully. However, the information content is limited. If, at some point in the entire MDP flow, we are going to represent the same data in pixel domain, then we have an implied limit on the information content. This limit is going to be a function of the number of pixels or the pixel size.

We can get some intuition of how Nyquist rate affects the pixel sampling using some examples with different grid alignments. In the pixel domain, the Nyquist rate limits the pitch, which in line:space (L:S) patterns is the sum of linewidth and spacewidth. Figure 8 assumes an equal L:S pattern. An important intuition about rasterization is that a given pattern must be discernable irrespective of the pixel-grid alignment. The example in Fig. 8(a) is a case of perfect grid alignment. It is easy to determine the original design from pixel values as a clear boundary can be seen in the pixel data between pixels with value 0.0 and those with value 1.0. So this is not an ambiguous example. The example in Fig. 8(b) shows 50% misalignment or perfect misalignment, where all the pixel values are 0.5. From the pixel data, we cannot restore the L:S pattern because that information has been lost in the translation to the pixel domain. The last example from Fig. 8(c) shows an 80%/20% split in alignment. This case is again ambiguous as there are two possibilities of line-space patterns rasterizing to the corresponding pixel values. Note that the Nyquist rate says sampling rate for the pitch needs to be "greater than 2×" and not "greater than or equal to 2×" for such reasons. So the Nyquist criteria also govern the mask rules in pixel domain.

### 2.3 Pixel Dose Equivalence

The exact vertex location is never used for wafer quality evaluation and it not relevant for mask writing or inspection as you never see sharp corners. So there is no so-called accurate polygon in pixel domain. In fact, the both red as well as blue curve from Fig. 9 rasterize to the same pixel values. Hence, rasterizations inherently act like a low-pass filter. As far as we are in pixel domain, red and blue contours have the same information content due to pixel dose equivalence. Here red contour uses much more data (in terms of vertices) in conveying the same information (in terms of pixels). Therefore, the upper bound on the maximum number of vertices required is governed by pixel size using the following equation:

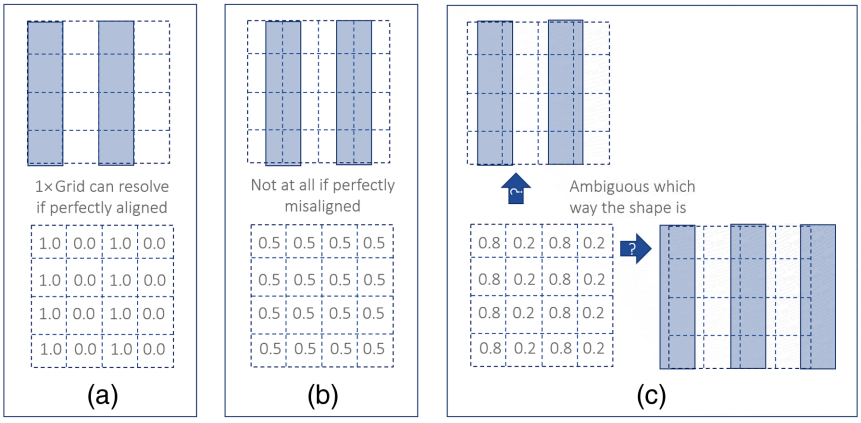


**Fig. 8** Rasterization of simple line-space pattern with linewidth and space equal to pixel size across different pixel alignments: (a) perfectly aligned with the input line; (b) 50% aligned with the input line; and (c) 80% or 20% aligned with the input line.

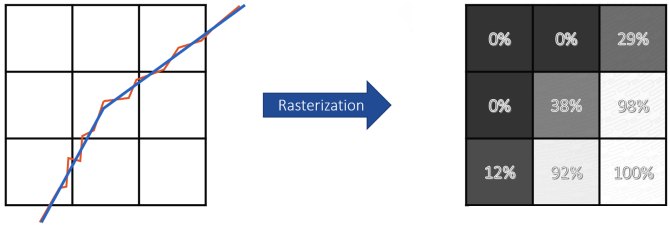


**Fig. 9** Rasterization of two different pixel dose equivalent input contours (red and blue) into pixel grid based on the pixel area coverage by the input contour.

$$\text{number of effective vertices} \approx \left(\frac{\text{contour perimeter}}{\text{pixel size}}\right). \tag{3}$$

A shape like the red contour is not repeatably and reliably manufacturable. So real shapes on production mask are likely to require few vertices.

## 3 Pixel-Based Computing for Curvilinear Masks

### 3.1 Pixel-Based Mask Writing

Most advanced masks are written by MBM writers and are therefore written using pixels. We are limited by what multibeam can write. Mask writer's pixel size governs the smallest printable feature size. Figure 10 shows three examples of how MBM writers would interpret the input data after the rasterization process. Figure 10(a) shows a large enough circle that, ignoring other mask process effects, can be represented on mask unambiguously. However, the examples from Figs. 10(b) and 10(c) show that different circles rasterized to the same data in pixels and thus they are ambiguous. Here the Nyquist rate and pixel sampling theory suggest that the pixel sizes are not sufficient to sample the data from Figs. 10(b) and 10(c). This issue would exist even if the masks undergo mask process corrections. The pixel size is generally fixed for a mask writer. Therefore, the data defining the mask shapes are bounded by the information theory.

### 3.2 ILT Computes the Mask by Evaluating Wafer Quality in Pixel Domain

The information content is also limited when considering how ILT simulates iteratively to generate the desired mask shapes that optimizes for wafer performance. ILT performs lithography simulation of the mask using fast Fourier transforms, which is done in the pixel domain. Input curve defining the ILT mask is passed through a wafer simulation engine to generate the output curve as shown in Fig. 11. For the user or the application engineer, the external behavior of ILT looks like the edges of polygons (either piecewise linear or curvilinear format) are being manipulated to generate the mask shapes. But the backend calculations involve rasterizing the mask contour, and using Fourier domain calculations for wafer simulation, which then gets contoured, to create the simulated contour. The simulated contour is then checked against the target, for validation, and sent to the mask shop after validation.

The ILT software needs to run as fast as possible and as accurately as possible. Smaller grid sizes take longer to compute, so all ILT tools use sufficiently small, but large enough pixel sizes for computation, as dictated by the resolution of the wafer writing process, principally limited by

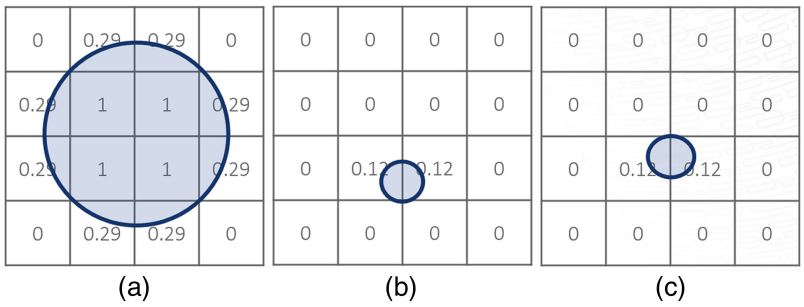


**Fig. 10** Rasterization of circles with different diameters and pixel alignments.

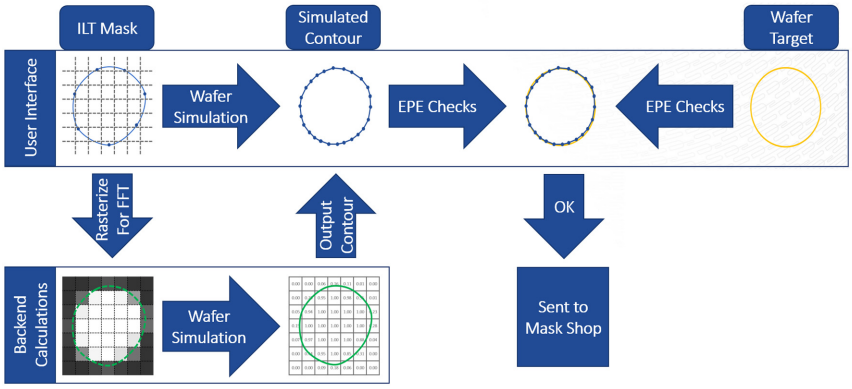


**Fig. 11** ILT mask validation process.

the lithographic step, EUV or 193i, and the resist used for wafer processing. In either case, the smallest circle that can be represented reliably across all pixel-grid alignments is also a function of this pixel size. Thus pixel-based computing limits the information content in ILT computations too.

### 3.3 Information Theory in Mask Processing Data Flow

An ILT/OPC shop needs to follow mask rules prescribed by their mask shop. The mask shop, in turn, is expected to accurately print the mask as determined by ILT/OPC, so long as the mask rules are followed. ILT/OPC shop is essentially saying "please produce this mask as described, because we verified that wafer performance will be best if you could manufacture this mask." And the mask shop's challenge becomes manufacturing the specified mask shapes on the physical mask as close as possible to the described shapes (Fig. 12).

In the VSB era, a further approximation was implied in this collaboration agreement between the ILT/OPC shop and the mask shop. Because 90-deg corners (or any sharp corners for that matter) are known not to be manufacturable, ILT/OPC shops made assumptions about "corner rounding" that the mask shapes would suffer. Because of limitations in computational time, corner rounding is done in simple rule-based approximations. An exact mask process may not be known at the time of ILT/OPC also, making any simulation-based methods inaccurate anyway. Furthermore, the corners have significantly worse dose margin on the mask as compared to straight edges in VSB writing. This makes the corner rounding different for different instances of corners, even if the corners belong to the same overall shapes in different locations across the mask. This practice is fine for lightly OPC'ed shapes because the precision of those corners is not that important to overall circuit performance, even if the errors are transmitted to the wafer. But it is increasingly a problem for heavily decorated OPC or Manhattanized ILT, as the corners are close to each other with small jogs of 20 nm occurring often in the mask shape. Since 20 nm is well below the three sigmas of the blur radius of the mask process, the error induced by "corner rounding" can be very significant. And dose margin is effectively bad everywhere on these shapes.

Curvilinear shapes, specifically, manufacturable curvilinear shapes fix this problem. Wafer simulation done in the ILT/OPC shop to certify the mask to produce good wafer performance is now simulating a mask that can actually be manufactured. Furthermore, and perhaps much more importantly, these masks have a much more uniform dose margin across all shape edges. Manufacturable shapes are more reliably manufacturable. 90-deg corners are not manufacturable.

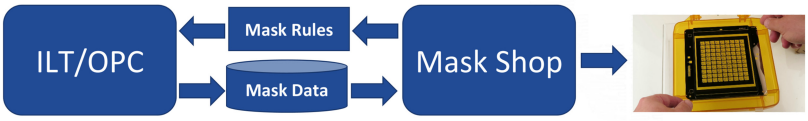


**Fig. 12** Collaboration between the ILT/OPC shop and mask shop.

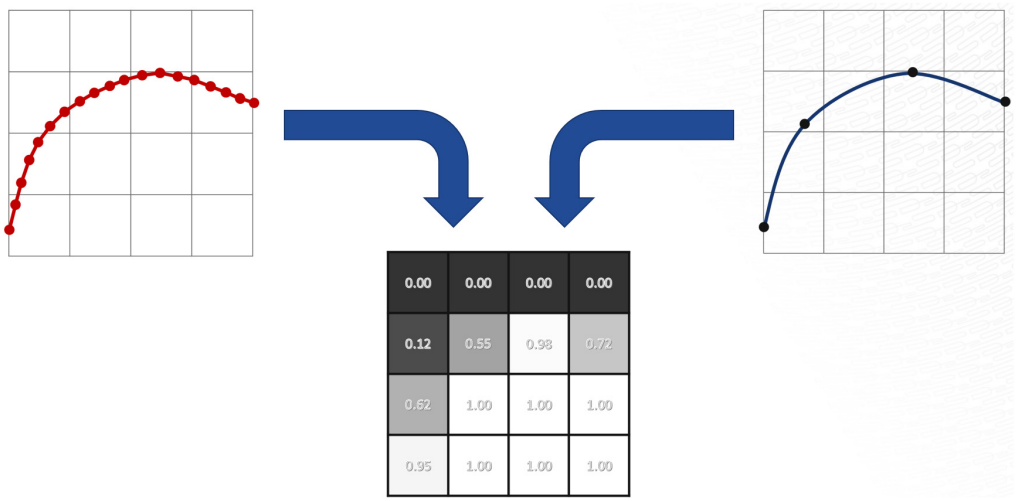


**Fig. 13** Example showing pixel dose equivalence is sufficient.

Not by accident, the MBM writers all have pixel resolution of the mask writers smaller than the resolution of 193i or EUV (at 4× dimensions), and therefore the resolution used by curvilinear ILT in computing wafer images during ILT iterations. This does not by itself guarantee that all curvilinear ILT outputs are inherently manufacturable on MWM writers. But it does make it much more likely and therefore reduce significantly the number of adjustments needed to obey MRC rules as prescribed by the mask shop.

Nevertheless, it is critically important for the mask shop to be provided a sufficiently exact specification of the mask shapes that must be produced to preserve the wafer quality achieved by the ILT/OPC shop. The mask writer must be provided the shapes to write, and inspection, metrology and repair tools need to know the shapes to compare with. Of debate is how much information is sufficient to exactly represent the information content assumed during ILT/OPC.

Demanding precise contour edge placement error (EPE) at infinite resolution is enticing but it is not necessary. What is more important is pixel dose equivalence at wafer simulation. The data are sampled into pixels in two ends: first in the wafer simulation software with ILT grid and then in the mask writer using multibeam pixel size. The data sampling during these two steps would act like low-pass filters that make the notion of exact contour irrelevant and thus making contour-based EPE checks somewhat insufficient and misleading in 2D curvy region.

So reliably transferring the dose information to the mask is important. Mask writer only cares about dose that it needs to project at every pixel. Any more data will be lost in the low-pass filtering process as depicted in Fig. 9. Transferring anything less is inaccurate and anything more is wasteful.

## 4 Conclusion

OPC/ILT that generates the mask shapes as well as the mask writer both use pixels in some part of the data path. Pixel-based computing adds a low-pass filter to data processing. So in summary, we can say information theory bounds all mask computations. The information beyond what OPC/ILT uses is irrelevant as the information used by ILT is sufficient for the wafer quality. The information beyond what MBM writers can write is wasted as it cannot be interpreted by the writer.

As shown in Fig. 13, it does not matter whether the red piecewise linear contour was used to define the mask or the dark blue piecewise curvilinear contour was used. The multibeam writer will rasterize both these contours in the same way and they get translated to the same pixel values. So the amount of dose projected on mask is the same. Thus by construction, it will produce the same mask. That is why pixel dose equivalence is sufficient and oversampling is just an unnecessary burden to the data path:

$$\text{maximum number of vertices required} \approx \left(\frac{\text{contour perimeter}}{\text{pixel size}}\right). \tag{4}$$

We need enough information to accurately represent the area coverage of the pixel. So having a maximum vertex count roughly in the order of contour perimeter over pixel size is sufficient to have pixel dose equivalence.

It is certainly possible to represent some curves using fewer vertices. For example, a circle can just be represented using its center and radius irrespective of it size or perimeter. Therefore, Eq. (4) represents the maximum number of effective vertices required for tighter curves that still satisfy all the constraints of information theory and Nyquist rate. For a mask writer with 16 nm pixel, this is roughly 16 nm on mask or 4 nm on wafer. So we do not need 1 nm segments polygon to represent mask. Here 1 nm is just a metaphor for oversampling. Yes, it is necessary to know the exact contour desired by ILT for the mask writer as well as for inspection and metrology tools. Representing the curve with segments that accurately interprets the desired curvy target, using a sampling interval in the order of multibeam writer pixel size should be more than sufficient. Having a curvy representation that is pixel-based computing aware and using pixel-based computing in MDP will help keep the file size small while maintaining information integrity and thus the quality of the masks written by MBM writers.

## Code, Data, and Materials Availability

All data in support of the findings of this paper are available within this article.

## Acknowledgment

There are no potential conflicts of interest as this paper is trying to formulate a theoretical limitation on curvilinear mask data based purely on the ideas of Information Theory.

## References

1. H. Matsumoto et al., "Multi-beam mask writer MBM-1000 and its application field," *Proc. SPIE* **9984**, 998405 (2016).
2. L. Pang, "Inverse lithography technology: 30 years from concept to practical, full-chip reality," *J. Micro/Nanopattern. Mats. Metrol.* **20**(3), 030901 (2021).
3. R. Pearman et al., "Full-chip GPU-accelerated curvilinear EUV dose and shape correction," *Proc. SPIE* **10451**, 1045108 (2017).
4. J. Choi et al., "Status of curvilinear data format working group," *Proc. SPIE* **12325**, 1232508 (2022).
5. L. Pang et al., "Enabling faster VSB writing of 193i curvilinear ILT masks that improve wafer process windows for advanced memory applications," *Proc. SPIE* **11518**, 115180W (2020).
6. C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.* **27**(3), 379–423 (1948).
7. C. Shannon, "Communication in the presence of noise," *Proc. IRE* **37**(1), 10–21 (1949).
8. H. Nyquist, "Certain topics in telegraph transmission theory," *Trans. Am. Inst. Electr. Eng.* **47**(2), 617–644 (1928).
9. Calma Corporation, "GDS II stream format," (1984).
10. OASIS: Open Artwork System Interchange Standard, P39-0304E2, "SEMI: Semiconductor Equipment and Materials International," (2002).
11. H. Zable et al., "GPU-accelerated inline linearity correction: pixel-level dose correction (PLDC) for the MBM-1000," *Proc. SPIE* **10454**, 104540D (2017).
12. H. Matsumoto et al., "Multi-beam mask writer, MBM-2000," *Proc. SPIE* **11908**, 119080L (2021).
13. F. Abboud et al., "Mask data processing in the era of multibeam writers," *Proc. SPIE* **9235**, 92350W (2014).
14. A. Reich, K. Nakagawa, and R. Boone, "OASIS vs. GDSII stream format efficiency," *Proc. SPIE* **5256**, 163–173 (2003).
15. Cadence Design Systems, "LEF/DEF language reference," Product Version 5.8, (May 2017). coriolis.lip6.fr/doc/lefdef/lefdefref/lefdefref.pdf
16. "Confidence in curvilinear mask making remains high," p. 15, "2022 eBeam Luminaries Survey," www.ebeam.org/docs/2022-ebeam-luminaries-survey-final-1.pdf (2022).
17. R. Pearman et al., "Adopting curvilinear shapes for production ILT: challenges and opportunities," *Proc. SPIE* **11148**, 111480T (2019).

**Abhishek Shendre** is a principal software engineer at D2S, Inc. He received his master's degree in computer engineering from the University of Illinois at Chicago with majors in very large scale integration (VLSI) and computer aided design. He is an expert in geometry and pixel-based software algorithms. He has concentrated his career in GPU-based software system design and has worked on many high-performance computing projects using compute unified device architecture (CUDA), message passing interface (MPI), and OpenMP. During the last 7 years, he has been involved in the research and development of algorithms for model-based mask data preparation and mask process correction software for multibeam mask writing and extreme ultraviolet lithography. He also has experience in designing EDA software for geometric manipulation of curvilinear shapes. He is a member of the SEMI working group responsible for developing format for curvilinear mask geometries.

**Aki Fujimura** is the founder and CEO of D2S, Inc. Previously, he served as CTO at Cadence Design Systems. He returned to Cadence for the second time through the acquisition of Simplex Solutions where he was a president/COO and an inside board member. He was also an inside board member and VP at Pure Software. Simplex and Pure both IPO'd during his tenure. He was a founding member of Tangent Systems in 1984, which was subsequently acquired by Cadence Design Systems in 1989. He was on the boards of HLDS, RTime, Bristol, S7, and Coverity, Inc., all of which were successfully acquired. He received his BS/MS degrees in electrical engineering from Massachusetts Institute of Technology (MIT).

**Mariusz Niewczas** is currently a senior member of consulting staff at D2S Inc. Previously, he was a principal engineer at AboundLogic and Staff Engineer at PDF Solutions Inc. He holds his PhD EE and MSc degree from Warsaw University of Technology, Poland, and has spent more than 10 years as an academic teacher and a researcher, first at Warsaw University of Technology and then at Carnegie Mellon University. He has published more than 35 papers and holds several patents about physical design and verification algorithms, design for manufacturability, statistical simulation, yield modeling, and analog circuit design and testing.

**Tom Kronmiller** is currently a distinguished engineer at D2S, Inc. Previously, he was a chief software engineer at Simplex Solutions and a fellow at Cadence. He received his BSEE in electrical engineering and computer science from Princeton University. He has 10 patents related to VLSI technologies. He is a member of the SEMI working group responsible for developing format for curvilinear mask geometries.