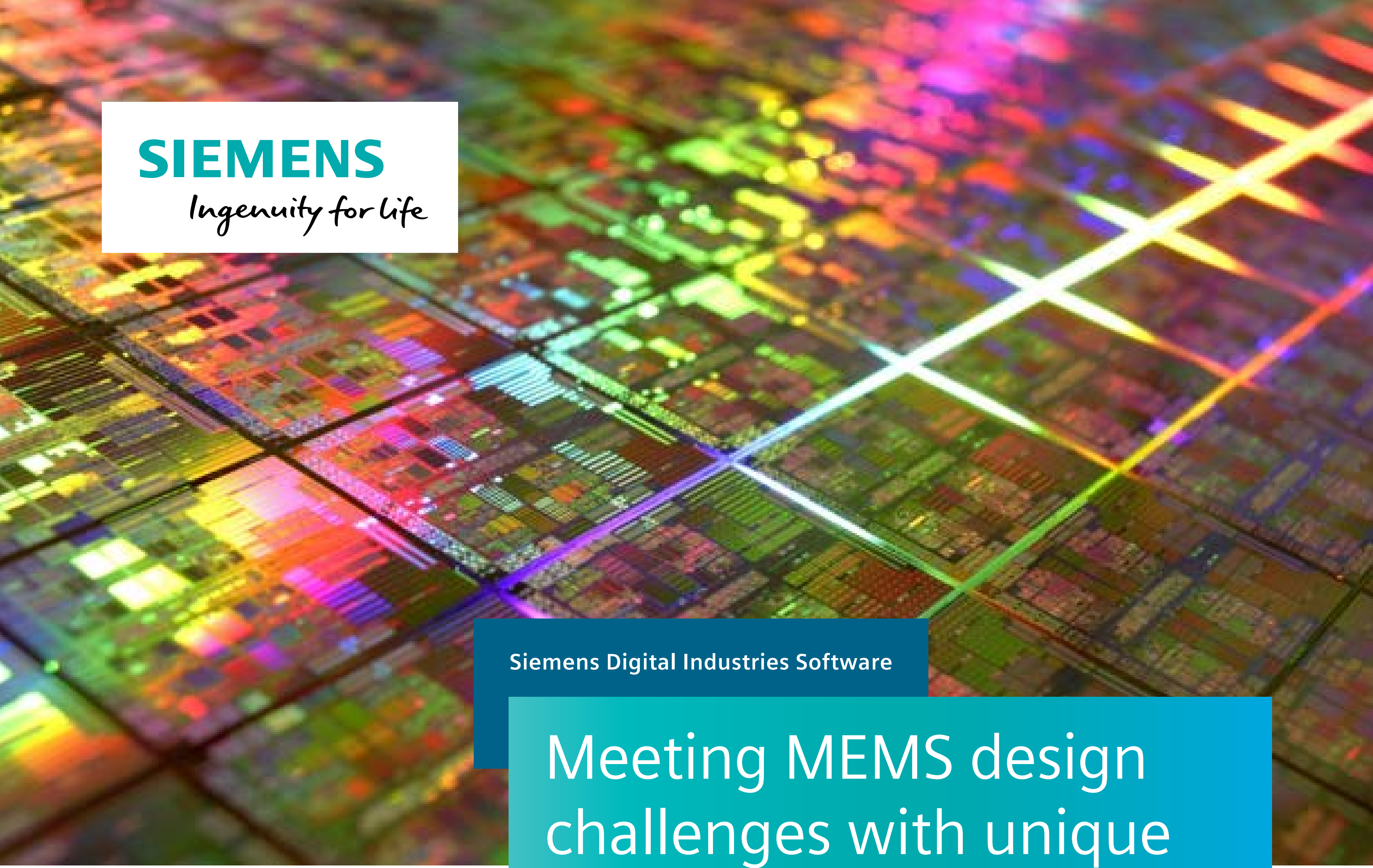

Siemens Digital Industries Software

Meeting MEMS design challenges with unique layout editing and verification features – Part 2

MEMS design

Executive summary

This two-part paper describes how and why support and ease of use for implementation of irregular shapes, including curves and all-angle polygons, is a critical criterion differentiating MEMS-oriented CAD tools from conventional IC-oriented tools. (Part 1 focuses on layout editing; Part 2 on verification.)

Nicolas Williams
product marketing manager
Siemens Digital Industries Software

Introduction

Microelectro-mechanical systems (MEMS) has been growing rapidly ever since it became possible to fabricate MEMS devices using modified semiconductor device fabrication technologies. Layout tools that are widely used in IC design naturally become the tool of choice for MEMS layout design. Although IC layout tools are quite mature and feature-rich for IC applications, many of them lack the capabilities to efficiently handle the challenges MEMS layout imposes. That is why unique MEMS-oriented features are needed in IC tools to address the specific requirements of MEMS layout design and to make the design process fast, easy and accurate.

A big difference between MEMS layout and IC layout is the use of unique, irregular shapes. Unlike conventional CMOS IC design, where layout shapes are usually Manhattan style (such as rectangles and rectilinear polygon) or polygon with 45-degree edges for routing, MEMS design utilizes a much broader variety of geometries, due to its wide application in mechanical, optical, magnetic, fluidic and biological fields. The support and ease of use for implementation of irregular shapes, including curves and all-angle polygons, becomes a critical criterion differentiating MEMS-oriented CAD tools from conventional IC-oriented tools. Most layout and verification tools are focused and optimized on IC design and are not suitable for irregular shapes needed for MEMS designs. MEMS designers need layout and verification tools that can handle challenges that curved or all-angle objects presents. Also, designers need tips and tricks on how to handle false errors that result from rules that are optimized for orthogonal geometry.

Overcoming limitations of mechanical CAD tool

Unlike traditional mechanical CAD tools, where 'zero-width' lines are common, MEMS layout requires all geometry to be represented as "closed" or "filled" polygons. This is needed to define the light and dark regions of the mask. One limitation with some mechanical CAD tools is that they cannot represent a filled polygon easily and any drawing done in these tools usually results in the polygon being represented as several 'zero-width' line segments. When importing DXF files (a common format used to transfer geometry from mechanical CAD tools), Tanner L-Edit MEMS can search for segments having endpoints within the user-specified tolerance and try to reconstruct closed polygons as seen in figure 1. If a closed polygon is found, the individual segments will be replaced by a single polygon. The endpoints of the line segments do not have to match exactly. Tanner L Edit MEMS allows the user to specify the largest gap between segment endpoints when joining them into a polygon.

When sending the MEMS design to the foundry for fabrication, the user typically will export the design in GDSII format. Since the GDSII format does not support curves, a conversion is needed for circles, pie wedges, curved-sided polygons and tori to all-angle polygons that approximate the curve when GDSII mask data is exported. Tanner L-Edit MEMS automatically performs this conversion, and additionally issues warnings if the design contains a wire or polygon with more than 200 vertices, since GDSII has a limit of 8,192 vertices and 200 vertices is a traditional best practice limit. The user can then specify a maximum number of vertices that

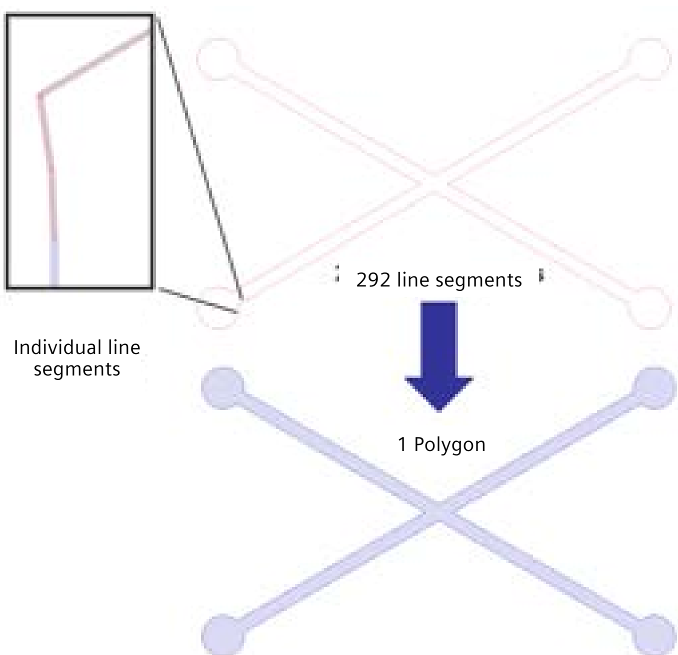

Figure 1: Polygon reconstruction from individual line segments in Tanner L-Edit MEMS.

each polygon should have and if it exceeds this maximum, the polygon will be automatically fractured into smaller polygons with fewer vertices by Tanner L-Edit MEMS.

All-angle polygons can also be converted back to curved polygons in Tanner L-Edit MEMS. Sometimes, a GDSII file, where curves are not preserved, needs to be read back in for design revision; or curves need to be recovered from the result of an advanced editing operation such as Boolean operations, making it easier to edit. To achieve good curve recovery, Tanner L-Edit MEMS searches all-angle polygons for arcs with eight or more vertices and replaces the multiple adjacent segments with curved edges, provided that those vertices lie on an arc with no more than one manufacturing unit radius error (figure 2). Such conversion capabilities make it much more convenient and accurate for users to re-edit curved objects.

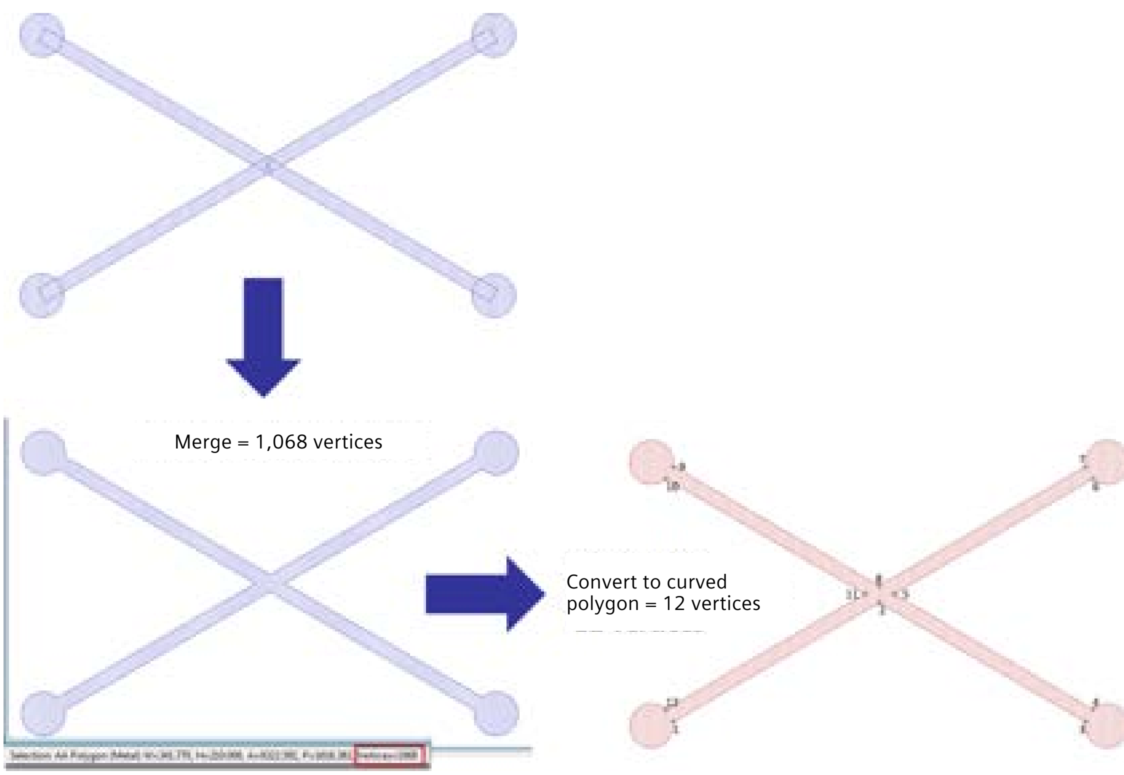

Figure 2: Curve reconstruction in Tanner L-Edit MEMS.

Curve conversion to all-angle edges

Curved polygons need to be converted to all-angle polygons when doing some advanced editing operations, when running design rule checking (DRC) and when exporting to GDSII. The all-angle approximation must represent the actual curve as accurately as possible. In some CAD tools, curves are converted based on a specific number of vertices, which doesn't guarantee the precision between curves of different sizes. Tanner L Edit MEMS converts curves based on the manufacturing grid, which adjusts the number of vertices to use during conversion based on the size of the curves to have maximum precision.

To show the difference between the approach of Tanner L-Edit MEMS and other CAD tools, three circles with a 5 µm, 50 µm, and 250 µm radius were converted in figure 3 to all angle polygons using a fixed number of vertices which is common in other CAD tools. They were also converted using the approach of Tanner L-Edit MEMS. Notice that for small curves such as the 5um radius circle, both approaches do a good job approximating the curve compared to the original curve and have about the same error. For larger curves, however, the error rate increases for the fixed number of vertices method to be as much as 0.3 µm for the 250 µm circle. Since Tanner L-Edit MEMS uses the manufacturing grid to calculate the number of vertices, the error is on average, less than the manufacturing grid of 0.01 µm. Even though edges are smoothed when fabricated, this error can affect how the resulting MEMS structure performs if the error is too high. Also, this conversion error can cause problems when doing Boolean operations on curved geometry and can cause many false DRC errors.

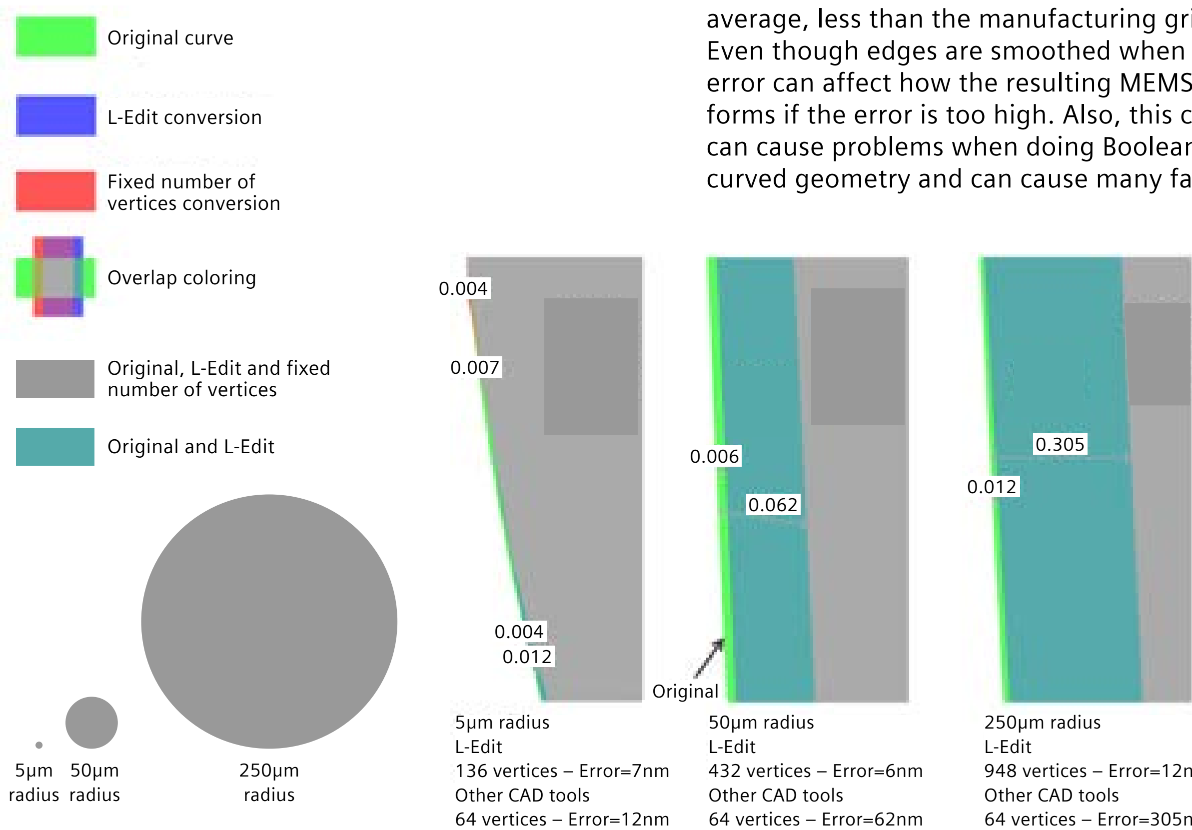

Figure 3: Curve reconstruction in Tanner L-Edit MEMS.

DRC Challenges of curved and all-angle geometry

Similar to IC layout, MEMS layout rLayer 1 ss DRC to minimize manufacturing risks and improve yield. In addition to regular DRC concerns, curved objects that are common in MEMS layout impose some unique challenges to the DRC process. The biggest challenge is the generation of false DRC errors because the DRC rule does not handle MEMS layout well. Most DRC engines and most DRC command files are designed for IC geometry, which is mainly orthogonal, and are not intended to handle all-angle or curved geometry. Standard rules that work fine for checking orthogonal geometry may generate many false errors when run on all-angle geometry. A few false errors can be quickly reviewed, but if there are hundreds or thousands of false errors, it is easy to miss true DRC errors especially across multiple DRC runs.

Tanner L-Edit MEMS is integrated with a hierarchical DRC engine called Tanner Verify that can help filter out some of these false DRC errors. To show the challenge of false DRC errors, a simple extension rule will be used as an example. An extension rule states that an object on one layer has to extend out of an object on another layer by at least a specific distance. The extension rule is measuring the distance from the external side (Outer) of an edge of the layer 1 object to the internal side (Inner) of an edge of the layer 2 object. Typically, this type of check ignores edges that intersect or edges that intersect at a 90-degree angle, as seen in figure 4.

The false DRC errors are generated when curved edges are approximated as multiple small all-angle edges, by

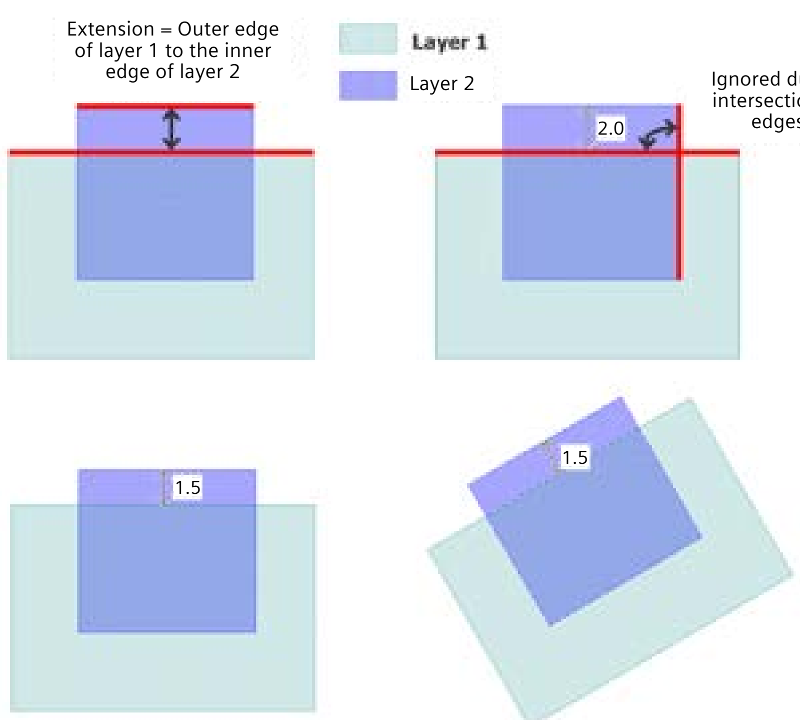

Figure 4: Extension DRC rule.

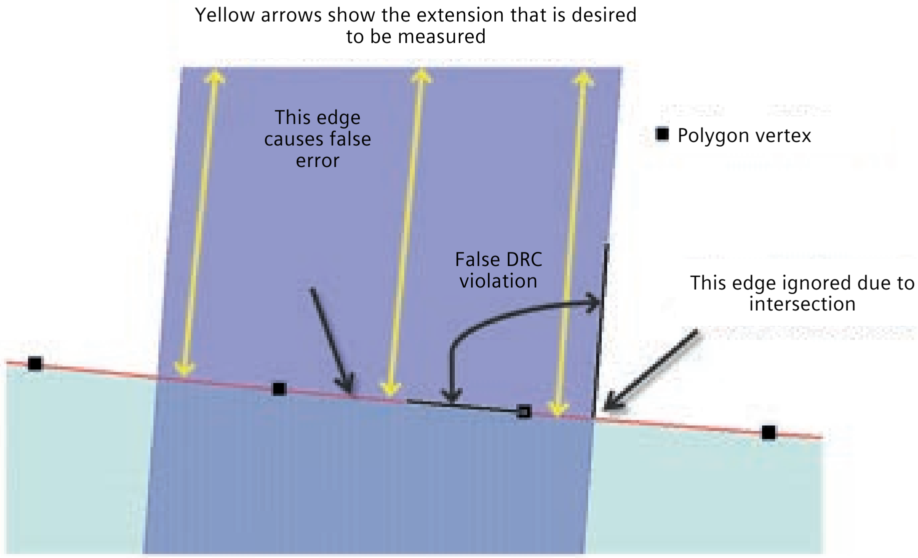

Figure 5: False DRC errors caused by curve approximation.

breaking a long single edge into multiple small edges as seen in figure 5. This makes it impossible to ignore intersections because the single curved edge on Layer 1 that intersected Layer 2 is converted to 3 edges, with one of the edges not intersecting a Layer 2 edge. This is not because the DRC engine cannot how handle this issue but because the rule was written and optimized for orthogonal geometry and needs to be modified to handle the curved edges.

One trick to help filter out many small edges due to curve approximation is to only check edges if they project onto each other by at least a specific amount. The projection of an edge onto another edge is a perpendicular projection from the reference edge to the edge being projected onto. Figure 6 shows an example of the layer 1 edge projecting onto the layer 2 edge and vice versa. A filter length of either ten times the manufacturing grid or a tenth of the measurement distance is a good rule of thumb to use.

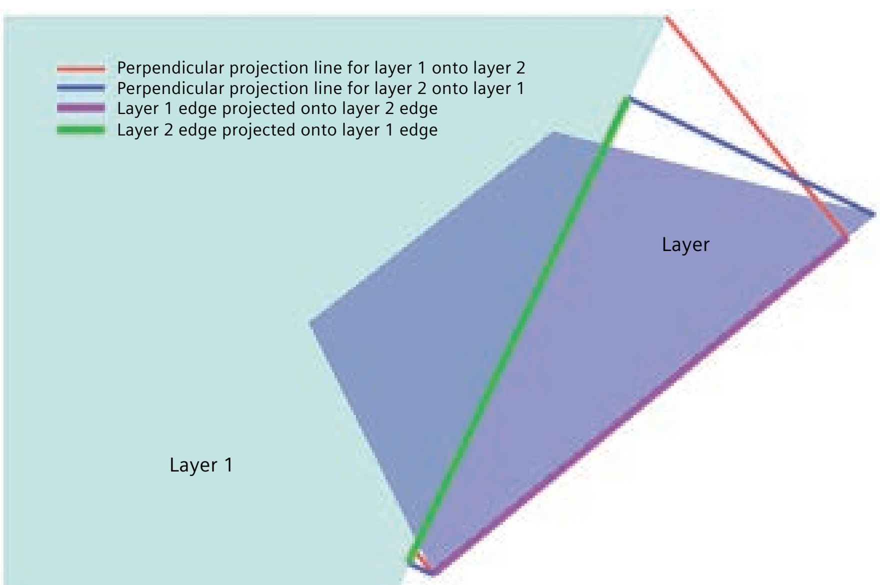

Figure 6: Edge projections.

Unfortunately, filtering by projection length will not remove the false errors that are due to the curved edges getting approximated as multiple small all-angle edges during the extension rule. The small edges that get created and don't intersect with the other layer can have reasonable projection lengths and as a result, will not be filtered out. The challenge is filtering out the edges that would have intersected if there was no approximation. The best approach is to rewrite the DRC run to only check the opposite edges and not the intersection. This can be accomplished by doing an extension check and saving the result as a polygon or region, and then performing an internal or width check that checks the spacing between the internal side (inner) of 2 edges of an object.

Figure 7 shows that when a true error exists, a polygon of four or more vertices is generated during the extension rule check. The extension rule check results in triangles at locations of the false DRC errors that result from a long single edge being broken into multiple small edges. Since intersections are not checked for the width check, these triangles are ignored and are effectively filtered from the results.

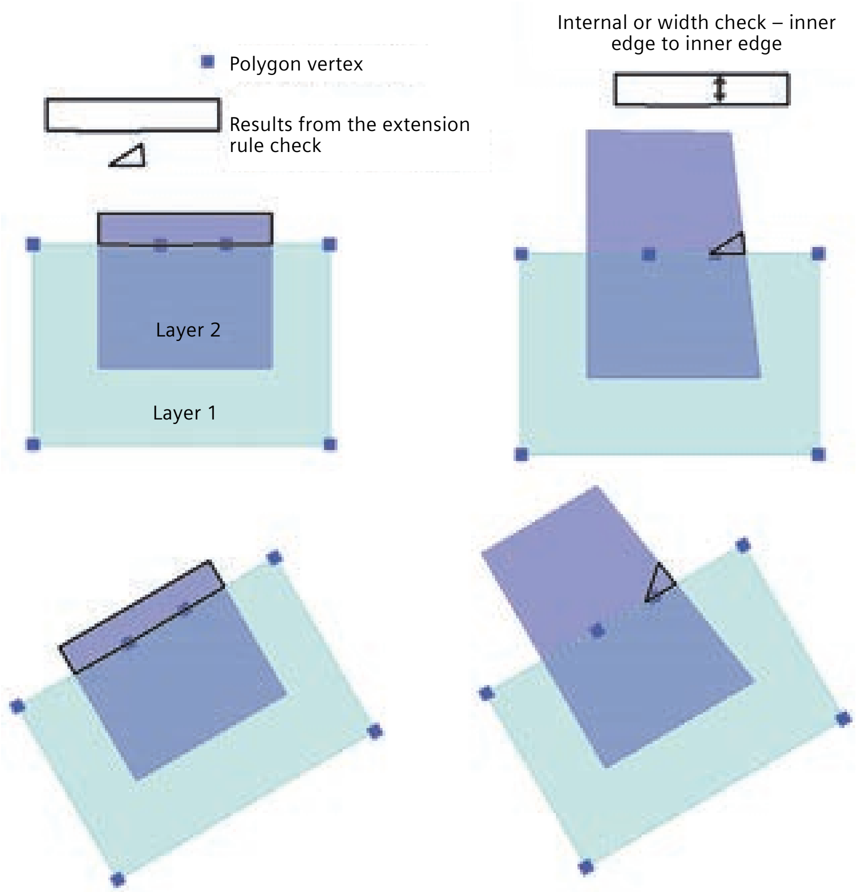

Figure 7: Two-step DRC check to filter out false errors due to curve approximation.

Round-off issues affecting all-angle rotation and node highlighting

Unlike IC layout, where DRC and LVS rely heavily on CAD tools to find and report errors, MEMS layout designers prefer a more intuitive way to check on the connectivity visually, before running verification tools, due to the intrinsic characteristics of MEMS components and the lack of a schematic netlist with which to compare. Tanner L-Edit MEMS supports node highlighting which allows the user to highlight all geometry connected to a node/net or a piece of geometry. To run Node Highlighting, the user first defines the connectivity by specifying names of layers that connect to each other. Layers connect if both layers overlap with each other and overlap with a connection layer such as contact or via or they can be setup to connect if they touch. Objects are defined as connected, if the AND of objects on Layer1, Layer2 and the connection layer results in non-zero area geometry. If a connection layer is not specified, then Layer1 and Layer2 must either overlap or touch to be considered connected. After defining the setup, the user then runs the connectivity extraction engine based on these definitions.

To pick the node to be highlighted, user can either highlight the geometry connected to the selected object, or open a dialog window to specify the targeted node name. Node highlighting works on merged objects

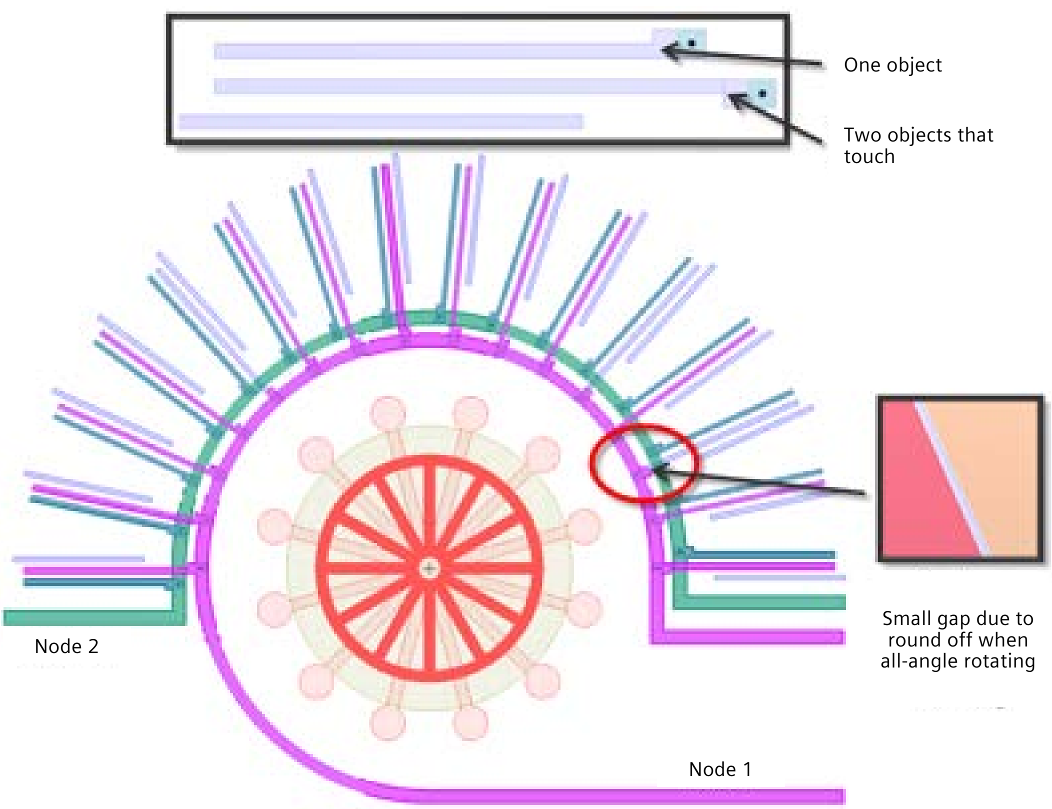

Figure 8: Node Highlighting of MEMS geometry and issues with all-angle rotation of touching objects.

on drawn or derived layers allowing the user to create complex connectivity rules. All the merged geometry of a node is highlighted in layout and the node name will be displayed in the status bar.

Multiple nodes may be sequentially selected and highlighted in different colors. When a node is highlighted, all related connectivity throughout the hierarchy will be displayed. When highlighting, if more than one node is located at the cursor location, a list of all potential nodes are displayed and the user will be prompted to pick one.

Figure 8 shows an example of using node highlighting to check the connectivity visually. Two nodes are highlighted in different colors displaying their connectivity so that user can quickly find erroneous connections. When doing connectivity extraction and node highlighting with an all-angle rotated objects, one issue that can arise is unconnected geometry due to round-off which may cause a small gap between objects in a rotated instance. When rotating an instance by an all-angle amount (non-90 degrees), each object is rotated individually and gets snapped to the resolution of the design database which is typically 1 nm. This snapping can cause round-off issues which may results in two objects that touch when they are not rotated becoming separated or not touching (small gap between them) when they are rotated at non-90 degree angles.

Notice in figure 8 that for node 1, a few of the spokes are not highlighted because small gaps arise during the rotation and snapping to the resolution of the design database. This small gap is hard to see but can be easily detected with node highlighting. The best way to handle this issue is to either merge touching objects in the cell being rotated or to make sure there is some overlap between the touching objects. Notice that the spokes connected to node 2 are all highlighted because that spoke is a single object and not 2 objects that are touching.

Conclusion

When doing MEMS design, design tools need features that can handle the challenges of arbitrary shapes and structures. The effects of curved geometry and how it gets approximated affects all aspects of layout from editing to DRC and Node highlighting. The key is having the right tools to efficiently operate on curved geometry and filter out false DRC errors that result from MEMS structures. With the above unique features specifically developed for the purpose of MEMS design, a MEMS layout can be accurately verified and sent to fabrication. This makes MEMS-oriented layout tools such as Tanner L-Edit MEMS a truly indispensable assistant to MEMS designers.

Siemens Digital Industries Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Our solutions help companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

siemens.com/software

81310-C4 05/21 TD