Mesh Generation for Three Dimensional Process Simulation

Norbert Strecker (Mountain View)

The three-dimensional simulation of fabrication processes for the semiconductor industry requires a very robust mesh generation algorithm together with a reliable update of the geometry.

While the process description of a single process step remains simple: dry or wet etch with a mask, isotropic, planarizing or selective deposition, oxidation or silicide growth, the complexity of the resulting structure becomes very high, usually already after a few steps. The situation becomes more complicated due to the presence of non-planar thin layers, e.g. in non-volatile memory cells.

In our 1D, 2D and 3D simulator we use an implicit geometry representation given by region-wise level set functions. Each level set function is constructed when defining a new region. If the initial region geometry is defined by one or several geometry objects, then the signed distance function from these objects is used as level set function.

The union and subtraction of objects is done using a combination of several level set functions.

If a new region is defined using a deposit command, the level set function is calculated based on the exposed regions of the existing mesh according to a formula that depends on the process step. If the region geometry is changed, e.g. by etching, a similar analysis of exposed regions and similar mathematical expressions are applied to define the level set functions for the modified regions.

For the simulation of oxidation and silicide growth the solution of level set equations is required to account for the motion of the interfaces. Diffusion-reaction equations are solved for the transport and surface reactions of the oxidant. A viscoelastic stress problem is solved to account for the mechanical deformation of the structure resulting from both initial or old stresses and from the surface reactions. Given the local mechanical displacements and the local material consumption rates, the velocity of the surface of each region is determined and extended into the entire volume. This extension velocity is used to solve the level set equation

$$\frac{\partial \varphi}{\partial t} + v \cdot \nabla \varphi = 0\,.$$

The level set functions for all regions are utilized to construct a boundary fitted mesh: at the end of the entire process step for an etching or deposition process and at the end of each time step for an oxidation or silicide growth process.

Our meshing algorithm combines the tasks of robust mesh generation with the filtering and correction of small scale noise and topology errors that may be present in the input description or the solution of the level set equations. This makes the algorithm well suited for 3D process simulation.

The mesh construction always starts by generating an initial grid. We allow the user to specify grid lines, by default we construct the grid lines by using all coordinates, provided by the user: all corners of geometry objects, mask corners and contact end points.

During the mesh refinement, a binary tree of mesh elements is constructed. If an element satisfies one of the refinement criteria it is split in an edge midpoint along one of the coordinate axes. The values of the level set functions in the corners of an element defines the regions to which the corner is assigned. If several level set functions provide the same maximum value in a corner, the corner is assigned to several regions. The assignment of corners to regions is used to control refinement at interfaces, to restrict refinements to certain regions and to pick appropriate field values during the data interpolation. Extrapolation of level set functions along edges is used to account for thin regions, since otherwise they might be lost after the meshing is finished.

Once all refinement is done, the boundary fitted mesh is constructed from the leaf elements of the tree using several stages.

In a first stage refinement is propagated to separate dangling points from unterminated refinements and interfaces: subdivided edges must not be cut by an interface.

In the second stage the edges are cut by interfaces. The sets of regions, identified for the end points of an edge are checked. If the two sets are disjoint, an edge cut is computed, based on the maximum intersection of two level set functions inside the edge. If the intersection point is too close to one of the end points, the level set function values and the regions stored for the end point are adjusted. Otherwise a new mesh point is inserted at the cut. After this step every edge can be assigned to at least one region.

In the third stage faces of the elements are triangulated using a generic triangulation that accounts for all points on the perimeter and inside of the face. The region information, stored for the corners is used to detect if the face is intersected by a triple line (intersection of three regions). The location of the triple line intersection is calculated as the position of the intersection of the three level set functions and is inserted to the face. Additionally the user may assist the mesher by supplying the exact location of ridges and triple lines (very important lines, VIL).

If a point is inserted to a face, it is connected to all points on the perimeter of the face. After this step all faces of the mesh elements can be assigned to at least one region.

The fourth stage consists of testing the volume tessellation of the elements. Based on the regions of the corners, the elements are tested for the presence of intersections of triple lines. If an intersection of triple lines is detected, it will be inserted in the interior of the element. Additionally the user may assist the mesher by supplying the exact location of the triple line intersection or corner (very important point, VIP).

After constructing all elements, a region is assigned to each of the sub-elements. The assignment utilizes available unambiguous region information for corners, edges, faces or the element. If no unambiguous information is available (e.g. on a Null patch), the element is assigned to the region with the maximum level set function in the centroid of the elements.

Once the construction of a boundary fitted mesh is finished, the mesh element quality is enhanced by a Delaunizer. The mesh points on interfaces and the corners of the bricks are transferred to the Delaunizer, the faces of the brick elements are flagged as protected and the faces between elements of different regions are flagged as interface.

The Delaunay algorithm establishes first a basic Delaunay mesh for the points defined by the mesh. Then it builds a conforming Delaunay triangulation by inserting mesh points into the protected faces and interfaces. Special tests are applied that guarantee that the control volumes are properly bounded by the interface faces.