Hints, Tips and Solutions

Derek Kimpton, Applications Engineer

Q. Is there a way to automatically specify mesh spacing in *ATHENA*?

A. Yes, there are three ways to automatically specify grid spacing. The most preferred method is described below and could best be described as "semi-automatic". The other methods not described below utilize the *MaskViews/ATHENA* interface and the adaptive meshing feature respectively.

The method described below is much preferred as it leaves the user in complete control of the mesh. If this simple procedure is followed correctly, mesh related problems are much less likely to occur.

I. Introduction

(i) Mesh in the X-Direction

When deciding on the placement of mesh points in the X direction, it is important to ensure that a defined mesh point is placed at every etched edge or deposited edge in the final structure.

If there is a vertical edge in the structure in a location without a corresponding mesh point, the meshing algorithum then has to re-arrange the mesh to ensure that one exists. During the re-arrangement process, X-mesh points near the effected surface may have to be removed to create the new mesh. Usually this is fine if it happens only once or twice, but if it happens repeatedly, the user is no longer in control of the mesh locations in the X-direction.

The meshing algorithum does not know in advance, what the next process step is going to be. Thus the automated re-meshing algorithum may remove X-mesh points that it will require for a later process step, necessatating further automated mesh re-generation. Eventually, if this situation happens repeatedly, the mesh becomes dis-ordered and structural problems become more likely.

Examples of process events that are likely to cause mesh problems due to the effects described above are repeated oxidizing anneals, followed by HF dips or etches which generally occurs in silicon processing. Each time a vertical edge is oxidized, the silicon vertical edge location is shifted, necessatating an automated re-meshing. Repeated oxidation and etching can therefore require numerous automated mesh re-arrangements. In fact, a
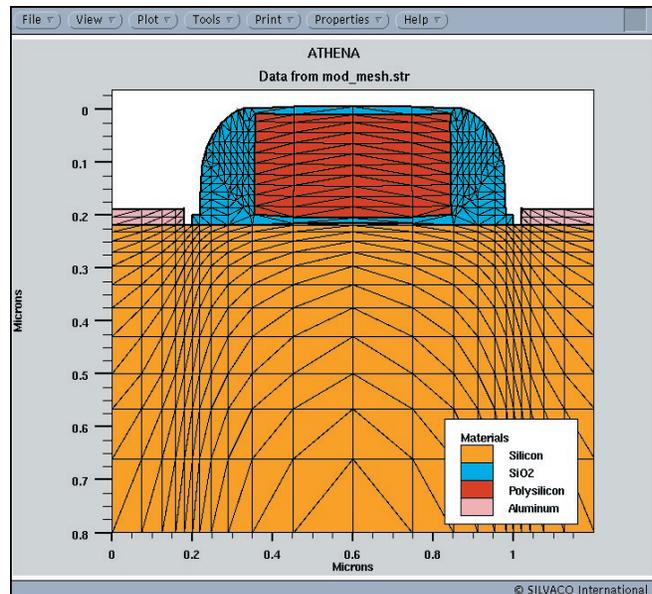

Figure 1. Mos1ex01.str created using the preferred method for automated mesh spacing in the X-direction

single oxidation can require numerous internal re-meshing events as the oxide front advances. Needless to say, if process steps such as repeated HF dips make little difference to the final structure (even though they are necessary in reality), they are best left out.

(ii) Mesh in the Y-direction

In common with meshing in the X-direction, Y-direction meshing should also ensure a Y-mesh point at each etch depth to reduce the occurance of automated re-meshing events. Y meshing definition should also ensure that sufficient meshing is provided to accurately reflect each implanted profile.

II. The Preferred Method for Automated Mesh Spacing

The preferred method for automated mesh spacing requires the user to look through all the processing steps in the input file where etching events occur. Then ensure an X and Y point is defined for these points. X points are easy to define, but Y points may need to be calculated if non absolute coordinates are used in the etch statement. For example, using the statement:

etch silicon dry thick=0.1

requires the user to know the original height of the relevant part of the structure being etched.

In order to illustrate how to create a minimum structure mesh in the X-direction, let us use mos1ex01 as an example. Looking through the input file, it will be noticed that there are etch events at X=0.18, X=0.2 and X=0.35. Using just these locations and adding the locations of the structure edges, we could create an automated grid spacing in the X-direction using the following syntax:-

```
go athena

line x loc=0
line x loc=0.18
line x loc=0.2
line x loc=0.35
line x loc=0.6
```

Notice how the usual "spac=<value>" parameter after all the "line x" statements is absent, instructing Athena to choose a suitable value automatically.

Since there are no critical Y-direction etching events in this example, such as trench etches, the Y mesh points can be left as is for this illustration, but the same techniques can be applied if required.

If all the "line x" statements of mos1ex01.in are replaced with this new automated mesh spacing syntax, the mesh created is shown in Figure 1.

This will provide a good basic mesh which will run fast due to it's minimilist nature and show any errors in process statements. Let's say that the process part is now de-bugged and the user now wishes to increase the number of mesh point around the edge of the poly gate to get an accurate result for the final run. The X-mesh statements could be modified by simply adding one spacing value at the poly gate edge (X=0.35um) as follows:-

```
go athena

line x loc=0
line x loc=0.18
line x loc=0.2
line x loc=0.35 spac=0.005
line x loc=0.6
```

In this instance, only the X-location corresponding to the edge of the poly gate was manually modified. The final mesh is now shown in Figure 2. It is important to note here that in mos2ex01.in, the "space.mult=2" parameter in the "init" statement multplies all "spac=<number>"
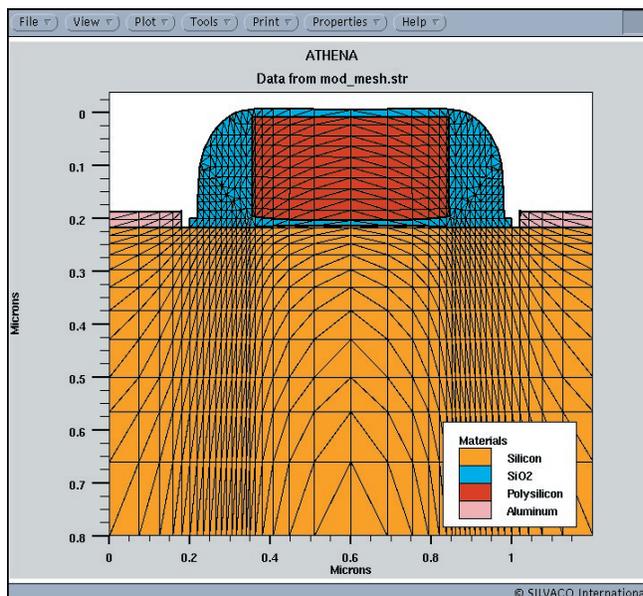

Figure 2. Mos1ex01.str created by manually modifying only the X-spacing at the poly gate edge and leaving all other automatic

values by 2, so the real spacing specified by "line x loc=0.35 spac=0.005" is actually 0.01 microns and not 0.005 microns. The automated spacing feature accessed by removing the "spac" parameter is uneffected by space.mult in the init statement.

A further advantage of the meshing method described above is that if the user is simulating a set of similar structures, such as simulating identical MOS devices with differnt gate lengths, the "set" statement can be used to define critical dimensions such as the X-location of the poly gate edge. If the same variable name is used to define this location in the mesh statement the meshing can remain automatic for each input file, even when the gate length changes.