ADVANCED ALGORITHMS FOR ETCHING SIMULATION OF 3D MEMS-TUNABLE LASERS

Abderrazzak El Boukili

Al Akhawayn University, Avenue Hassan II, P.O. Box 104, Ifrane 53000, Morocco
a.elboukili@aui.ma

ABSTRACT

This The integrated circuits (ICs) industry uses a number of technology computer aided design (TCAD) software tools to simulate the manufacturing and the operation of many ICs at different levels. At very low level, the simulation tools are used to simulate the device fabrication and design. These simulation tools are based on solving mathematical equations that describe the physics of dopant diffusion, silicon oxidation, etching, deposition, lithography, implantation, and metallization. The simulation of physical etching solves etching equations to calculate the etching rate. And this rate is used to move the geometry of the device. The simulation of non-physical (geometrical) etching is based on geometrical Boolean operations. In this paper, we are proposing new and advanced geometrical etching algorithms for the process simulation of three dimensional (3D) micro electro mechanical systems (MEMS) and MEMS-tunable vertical cavity semiconductor optical amplifiers (VCSOAs). These algorithms are based on advanced domain decomposition methods, Delaunay meshing algorithms, and surface re-meshing and smoothing techniques. These algorithms are simple, robust, and significantly reduce the overall run time of the process simulation of 3D MEMS and MEMS-tunable laser devices. The description of the proposed etching algorithms will be presented. Numerical simulation results showing the performances of these algorithms will be given and analyzed for realistic 3D MEMS and MEMS-tunable laser devices.

KEYWORDS

Advanced etching algorithms, domain decomposition, MEMS-tunable optical amplifiers, process simulation

1. INTRODUCTION

In our real life, MEMS devices are used almost everywhere. They are used in medicine, automotive industry, sensors industry, telecommunication industry and more. For example, MEMS technology leads to greater safety for automobile drivers. With their super-powerful sensors, MEMS accelerometers can sense when a vehicle has been in an impact; they can even judge the speed and severity of the impact in order to deploy airbags at the right speed and volume.

The iPhone actually uses extensive MEMS technology for many of its applications. Sensitive MEMS accelerometers can be scaled down and incorporated into handheld devices like mobile phones. They allow the phone to sense which way it is being turned and shift the screen from a portrait layout to a landscape layout, for example. They are also responsible for much of the hype about iPhone games, which use gimmicks like shaking the phone in order to roll dice.

Cell phone MEMS devices can also be integrated with an electronic compass in order to provide the GPS positioning system that iPhones offer. Because of MEMS' tiny size and versatility this technology can produce everything a consumer could ever want in a phone. Already, it has led to

DOI : 10.5121/ijctcm.2013.3201

cell phone microphones, autofocus actuators, BAW filters and duplexers, projectors, inclinometers, pressure sensors, and pico-projectors.

Vertical cavity semiconductor optical amplifiers (SOAs) represent a low-cost alternative to existing amplifier technologies. They could be used in fiber-optic communication systems such as metro and access networks [2], [3].

The surface-normal operation of vertical-cavity (SOAs) gives rise to many advantages as high coupling efficiency to optical fibers, polarization insensitive gain, the potential to fabricate high fill-factor two-dimensional arrays, and the ability to test devices on wafer [1],[2], [3], [6],[9],[11],[12].

Understanding MEMS and MEMS-tunable device fabrications is essential in optimizing design and enabling a more rapid production of mature devices. The 3D MEMS process simulation is a complex and challenging issue. The structures are geometrically complicated and inherently three-dimensional.

The main goal of this paper is to significantly improve, enhance, optimize and extend from two dimensions (2D) to three dimensions the existing geometrical etching algorithms for process simulation of 3D MEMS and MEMS-tunable vertical cavity semiconductor optical amplifiers (MT-VCSOAs) as shown in Figure 1. Most of commercial process simulators used worldwide are based on the industry standard two dimensional (2D) process simulator SUPREM-IV [5] or 3D process simulator FLOOPS [7]. SUPREM-IV has been developed by Stanford University in California. In this paper, firstly, we are significantly improving the 2D etching algorithms used in SUPREM-IV. And, secondly, we did extend these 2D etching algorithms to 3D which was an other challenging task.

We were, exactly, motivated by solving the following three difficulties :

1. In case of a 2D refined mesh using SUPREM-IV algorithms, a strange ripped nylon mesh is generated after etching as seen in “Fig. 2” and “Fig. 5”.
2. Extending the 2D geometrical etching algorithms of SUPREM-IV to 3D for process simulations of complex geometries of MEMS and MT-VCSOAs, while keeping the trade-off between computational efficiency and the simulation's run time.
3. Creating an accurate and suitable 3D geometry and mesh for the fabrication, mechanical, and electrical simulations of MT-VCSOAs and MEMS.

The traditional geometrical etching algorithms [5],[7],[10] are based on geometrical Boolean operations as region subtractions, region intersections, and region unions. They also incorporate additional supporting algorithms as, geometry validation, mesh quality control after etching, de-looping, removing holes, and making regions convex. In case of moving boundaries, as in physical etching, moving boundaries algorithms [4],[8] are used to calculate the etching rate (or the velocity of the boundary). In our case, we are not dealing with moving boundaries.

The main step in these etching algorithms is the Boolean subtraction operation. Our contribution is firstly to solve the difficulty 1. by applying boundary smoothing and re-meshing idea after subtraction operation as can be seen in “Fig. 3” and “Fig. 6”. Secondly, we used domain decomposition method (DDM) together with geometrical etching algorithms for the etching simulation of the complex 3D MEMS and MT-VCSOAs. The use of DDM is an excellent remedy to the difficulty in 2. The application of DDM significantly increases the efficiency and reduces the run time of etching simulation and the overall run time of the other process simulations including deposition, implantation, oxidation, and diffusion.

The DDM consists in decomposing the whole 3D structure into different 3D blocks. We could then apply etching algorithms to each block and then merge all the blocks to get the whole 3D structure. Advanced merging algorithms have also been developed and used after etching. They are based on region unions and intersections. The DDM is also very convenient for parallelization.

This paper is organized as follows. The second section describes in details the new etching algorithms. The third section presents the 2D and 3D numerical results that validate qualitatively and quantitatively the introduced geometrical etching algorithms. The performance of these etching algorithms are also discussed and analyzed. A comparison of the obtained 3D numerical results with other results found in literature is also presented. Section 4 contains conclusions, future work, and recommendations stemming from the work presented in this paper.

2. DESCRIPTION OF THE NEW ETCHING ALGORITHMS

Every 2D or 3D semiconductor structure could be defined in terms of geometry, surface mesh, and volume mesh. Geometry represents the gross outline of the structure and the different materials in the structure. We could look at the geometry as a set of regions. And each region is defined by a material. Surface or boundary mesh refers to a set of planar elements (points, edges, or triangles) whose union form the boundary of the geometry. The volume mesh refers to a set of volumetric elements (as tetrahedral elements) whose union defines the interior and the exterior of the whole device. The basic idea of this paper is to smooth and re-mesh properly the surface of each region of the geometry after a geometrical etching. By doing so, we were successful to solve the difficulty 1. in 2D and 3D.

In this section, we will describe in details the algorithms developed and used to perform the geometrical etching simulations in 2D. The extension of these algorithms to 3D has also been achieved and used.

The geometrical etching algorithm first defines the geometry of the region to be etched. This region is then subtracted from each region of the whole structure. The pseudo-code of the algorithm is given in the Algorithm 1 given below. The main idea behind this etching algorithm is as follows: the intersection **i** between the etch region **e** and a region **r** in the geometry is first computed (line 3 in Algorithm 1). If the etch region is fully enclosed within the geometry region, then, the etch is completed by defining the geometry region to have two boundaries (lines 4-5). The outer boundary is the existing boundary of the geometry region and the inner boundary is the etch region's boundary. Our new implementation of the etching algorithm consists in adding a new procedure **SmoothBoudary()** that will smooth properly these inner and outer boundaries according to some given data from the user. If the etch region completely contains the geometry region, then, the geometry region is completely deleted (lines 6-7). Otherwise, the intersection between the etch region and the geometry region is considered. The geometry region **r** is to be replaced with the regions that are outside the etch region **e**. Let **I** be the set of all the points of the intersection **i**. Let **R** be the set of all the points of the geometry region **r**. Let **E** be the set of all the points of the etch region **e** and let **P = R - I**. The Algorithm 1 below, then, works as follows.

First a point p0 of the region **r**'s boundary that is not part of the intersection (member of the set **P**) is found (line 14). Staring from this point p0, all the other subsequent points of **R** that are member of **P** are collected to form a new boundary **B** until the intersection with the etch region is found (lines 16-20). The collection continuous along the intersection from the set **I** until the region **r** is found (lines 21-26). The points of the region **r** in the set **R** are then collected until the first point is encountered (lines 27-30). This closed loop then generates a new boundary. This new boundary is then smoothed and re-meshed by calling the new procedure **SmoothBoundary()** and finally added to the geometry data structure. The procedure continues until all the points of the geometry region r's boundary that belong to **P** are part of the new boundary (lines 15, 33-34).

The pseudo-code of the new geometrical etching algorithm (Algorithm 1) is given in the following subsection.

2.1. Pseudo-Code of the New Etching Algorithm

```
Algorithm 1: new geometrical etching algorithm
Procedure Etch (e, r)
Inputs: e: etch region, r: geometry region
Outputs: new smoothed boundaries and new regions
begin
1. e:= Etch Region;
2. r:= Geometry Region;
3. i:= RegionIntersection(e,r);
4. if  ( i=e)  then (etch region inside region)
5. add boundary of e as internal boundary of r
6.  SmoothBoundary(); (significant improvement to etching algorithms)
7. elseif  ( i=r) then (geometry region inside etch region)
8.  delete  r;
    else
9.  E:= Points of e;
10. R:= Points of r;
11. I:= Points of i;
12. P:= R - I;
13. B:= 0;
14. first:=p:= MEMBER(0,P)
15. while ($P \neq 0$) do  (find the points that are outside)
16. INSERT(p,B);
17. np:=FIND(R, point next to p);
18. if  ($np \in P$) then
19.   DELETE(p,P);
20.    p:=np;
     else(find the points that belong to i)
21. np:=FIND(I, point on edge from p to np);
22. INSERT(np,B);
23. repeat
24. np:=FIND(I, point next to np);
25. INSERT(np,B);
26. until np is on r; (loop back to get all the other points of r)
27. p=FIND(R,point next to np);
28. while ( $p \neq$ first) do
29. INSERT(p,B);
30. DELETE(p,P);
31. add new region formed by points in the set B;
32. SmoothBoundary(); (significant improvement to etching algorithms)
33. B:=0;
34. first:=p:=MEMBER(0,P);
35. delete region r;
36. SmoothAllBoundaries(); (significant improvement to etching algorithms)
 this procedure is optional. It could be called to smooth globally all the boundaries.
end
```

The meaning of the other main procedures in Algorithm 1 is as follows:

MEMBER(i,S). Return the ith member of the set S.
INSERT(u,S). Make u an element of the set S.
DELETE(u,S). Remove element u from the set S if u is a member.
FIND(S,L). Find and return an element u of the set S that meets criteria L.

We should note that if the boundary smoothing procedures (**SmoothBoundary()** and **SmoothAllBoundaries()**) are called outside the geometrical etching Algorithm 1, then, the analysis of the complexity in time of the Algorithm 1 shows that the Algorithm 1 is O(**N+K**). The number **N** represents the number of points of the geometry region **r**. And **K** is the number of the points of the etch region **e.** Then, the complexity of the Algorithm 1 has the advantage to be linear.

2.2. Pseudo-Code of the New Smoothing Boundary Procedures

During etching algorithm, we call the smoothing boundary procedure **SmoothBoundary**() to smooth and re-mesh the new boundary that we just get. This procedure will split up some or all the long edges of the new boundary according to some criteria and to the user defined parameter α. The shape of the geometry is not altered if the long edges are split up. However, what constitutes a long edge? In this algorithm, long edges are judged in 2D according to the original perimeter of the boundary under hand. If an edge of length l_s is bigger than the user specified percentage (α) of the perimeter or bigger than 3 times the smallest edge of length l_min, then, this edge will be split up. The algorithm 2, given bellow, first calculates the perimeter of the boundary under hand (line 1). Secondly, it finds the smallest edge (line 2). Thirdly, it loops over all the edges of the boundary and checks to see if it is bigger than $\alpha \times$ perimeter or bigger than $3 \times$ l_min (lines 3-5). If it is, then, the procedure **SplitEdge()** is called to split up the edge (line 6).

Algorithm 2: Boundary Smoothing
Procedure SmoothBoundary ()
Inputs: **B**: region boundary to be smoothed according to user
α : user defined parameter used as a smoothing factor
Outputs: new smoothed and re-meshed boundary **B**

```
begin
1. p:=perimeter of boundary B; or area of a surface in 3D
2. l_min=length of the smallest edge of boundary B;
3. for each edge e of B do
4.     l_s = length of e;
5.     if (l_s > 3 × l_min or l_s > $\alpha$ × p ) then
6.         SplitEdge(e);
end
```

The etching and smoothing algorithms rely on many other geometry utility algorithms such as bulk and boundary mesh generation algorithms, algorithms to remove holes and make all the regions convex, algorithms assuring boundary orientation in counter-clockwise manner and other similar utilities.

3. 3D NUMERICAL RESULTS AND ANALYSIS

Using the proposed new geometrical etching algorithms, we were able to solve the 3 difficulties 1., 2. and 3. described above. On the other hand, we were successful to etch efficiently all the complicated 3D structures under hand. Figure 1 shows a schematic representation of a realistic MEMS tunable VCSOAs.

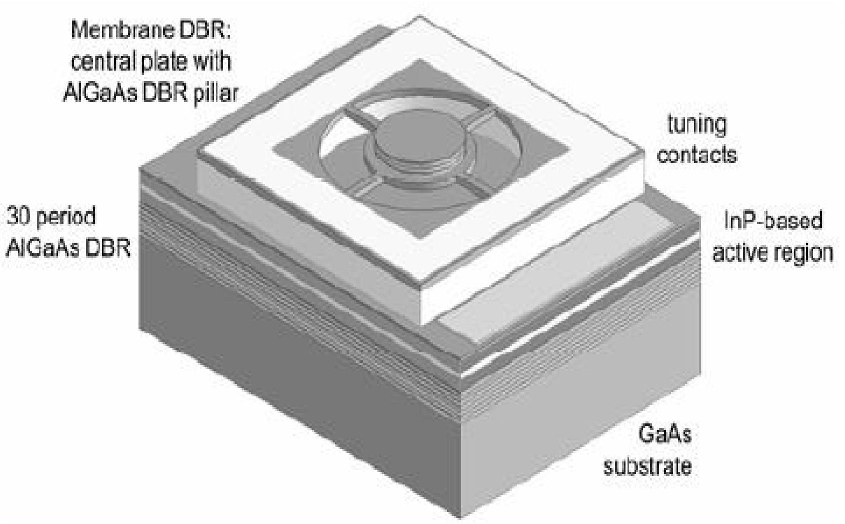

Figure 1, Schematic representation of MT-VCSOAs

Figure 2 and Figure 5 show a ripped nylon mesh after using SUPREM-IV etching algorithms. This type of mesh does cause troubles for the convergence of Newton's algorithm used to solve diffusion equations and for the simulation results. By using Algorithms 1 and 2 described above we were able to generate a better and high quality mesh as shown in Figure 3 and Figure 6. We show in Figure 4 a 2D mesh before etching. The Figure 7 to Figure 15 show the qualitative and quantitative behaviour of the proposed etching algorithms for different realistic and complicated geometries of 3D MEMS and MEMS tunable VCSOAs. The quality of the mesh and the geometry shown in all these figures (Figure 7 to Figure 11) does validate the excellent performance of the proposed etching algorithms. These performances are comparable with those find in [4] for different 3D structures. The size of our obtained meshes are even smaller and better optimized than those in [4]. This is due to the use of the domain decomposition method and the local mesh adaptation.

4. CONCLUSION

In conclusion, by using these new geometrical etching algorithms, it is possible to generate accurately complex geometries and meshes for mechanical and electrical simulations of 3D MEMS and MEMS-tunable laser devices. Parallelization of these algorithms could be investigated in future work.

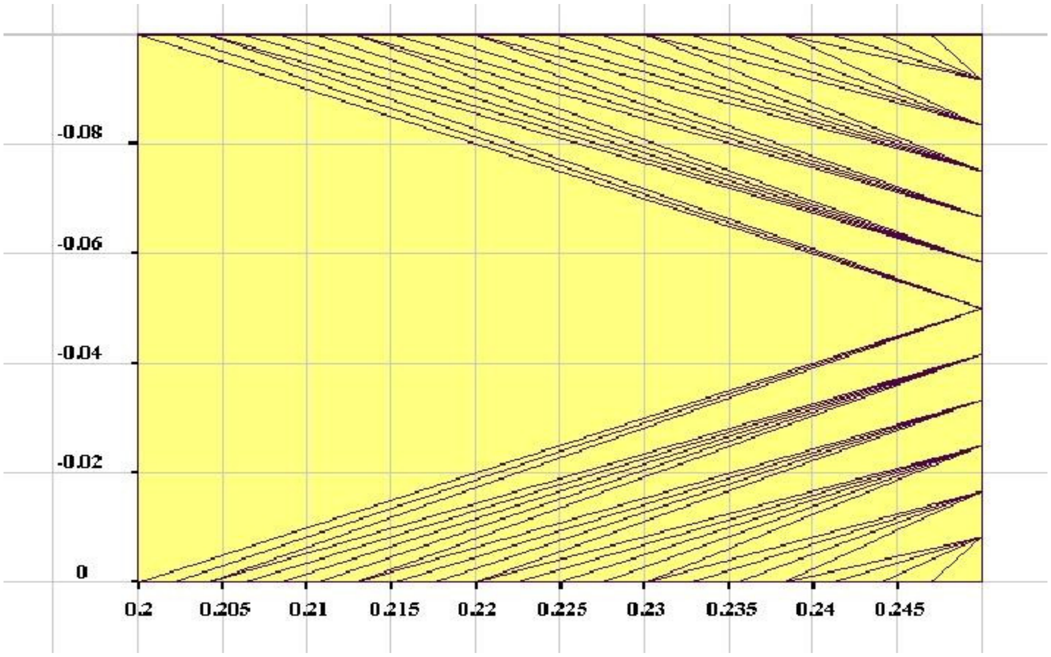

Figure 2, Ripped nylon mesh after SUPREM-IV etching

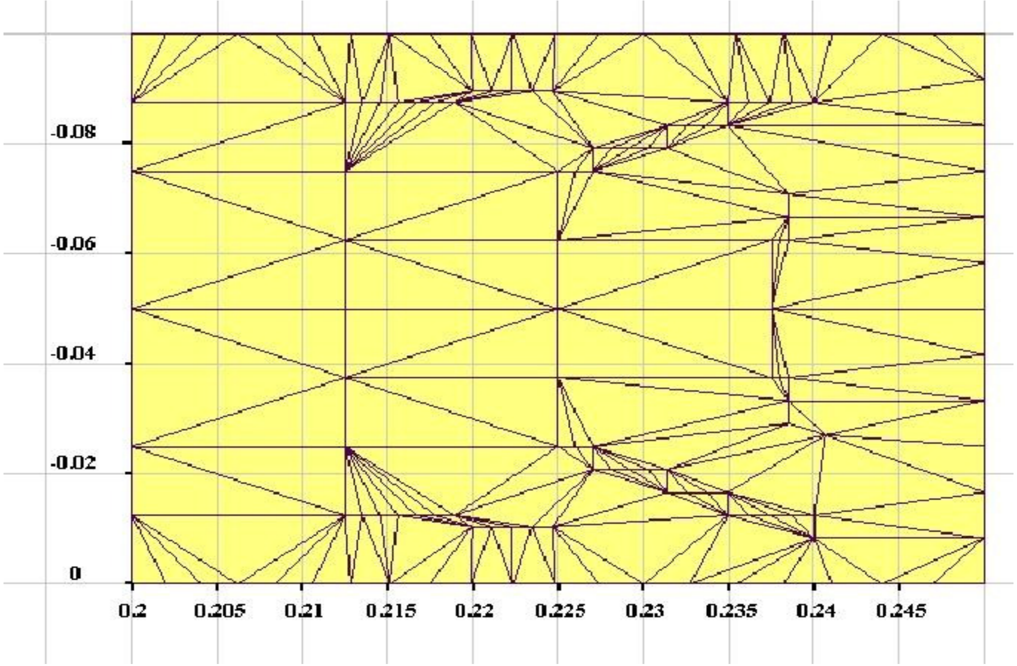

Figure 3, Smoothing boundary helped repair the ripped mesh

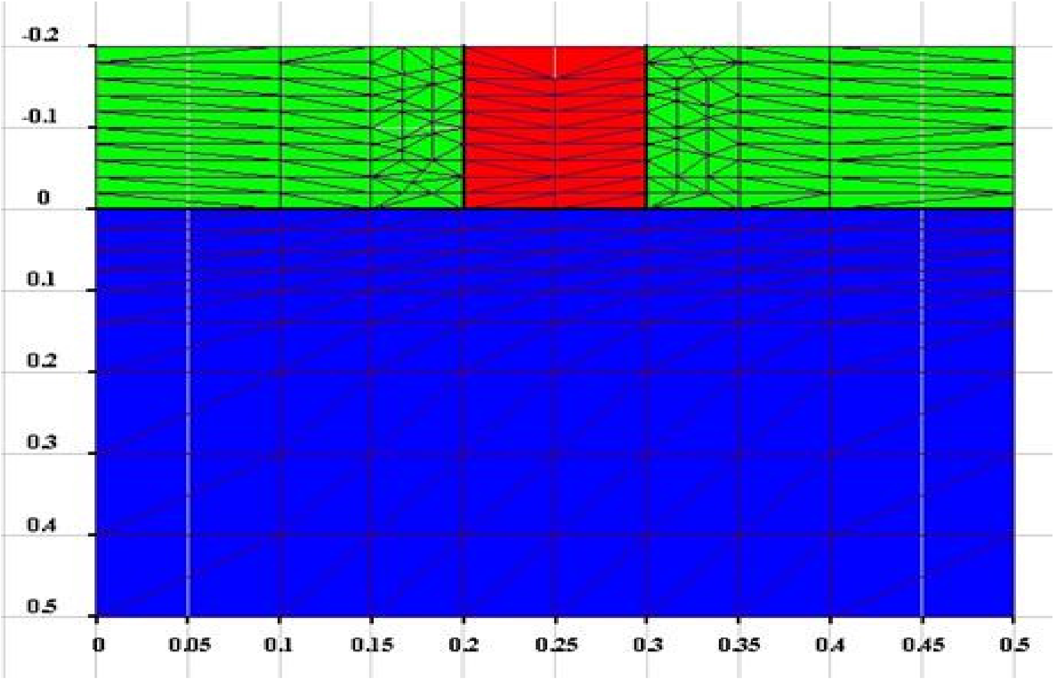

Figure 4, 2D mesh before etching

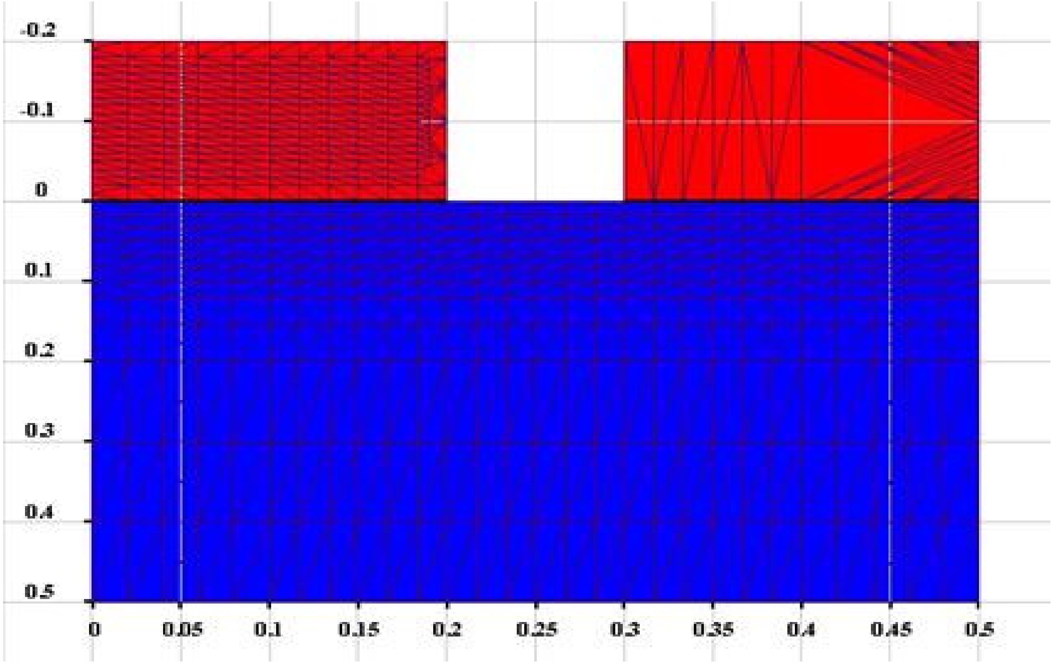

Figure 5, 2D mesh problem after SUPREM-IV etching.

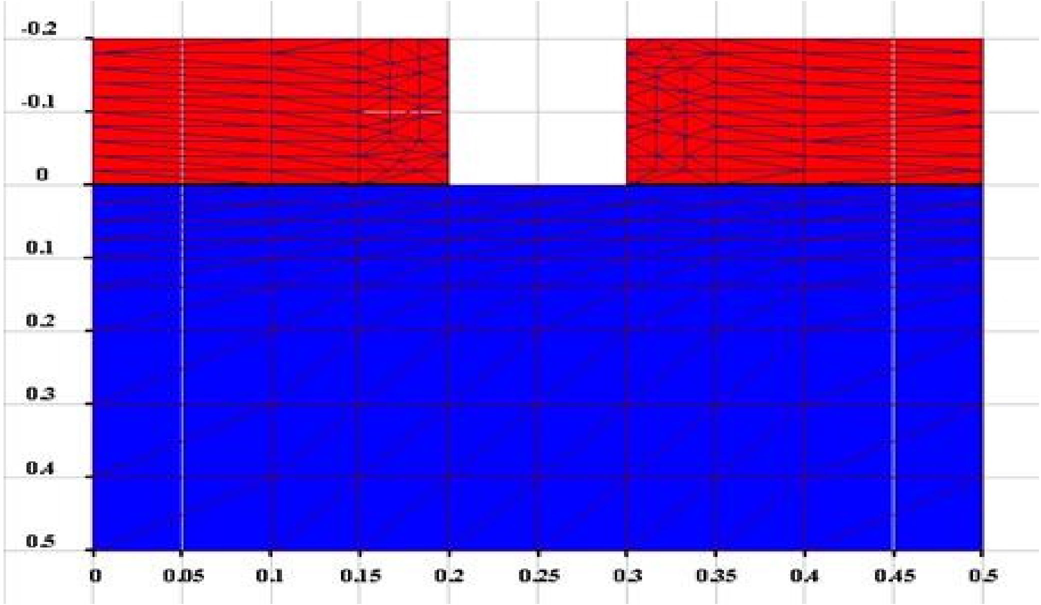

Figure 6, Better quality mesh after our improved etching algorithm

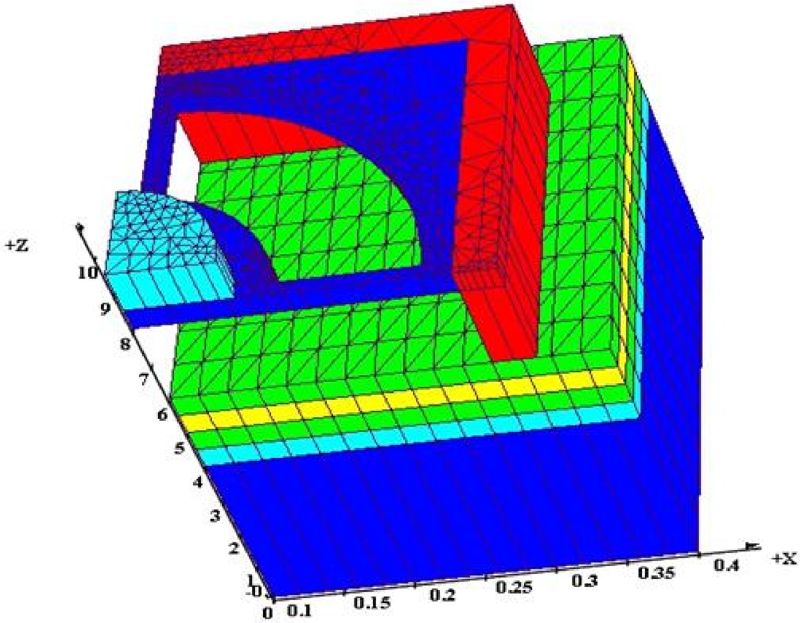

Figure 7, 3D MEMS Tunable VCSOAs after improved etching

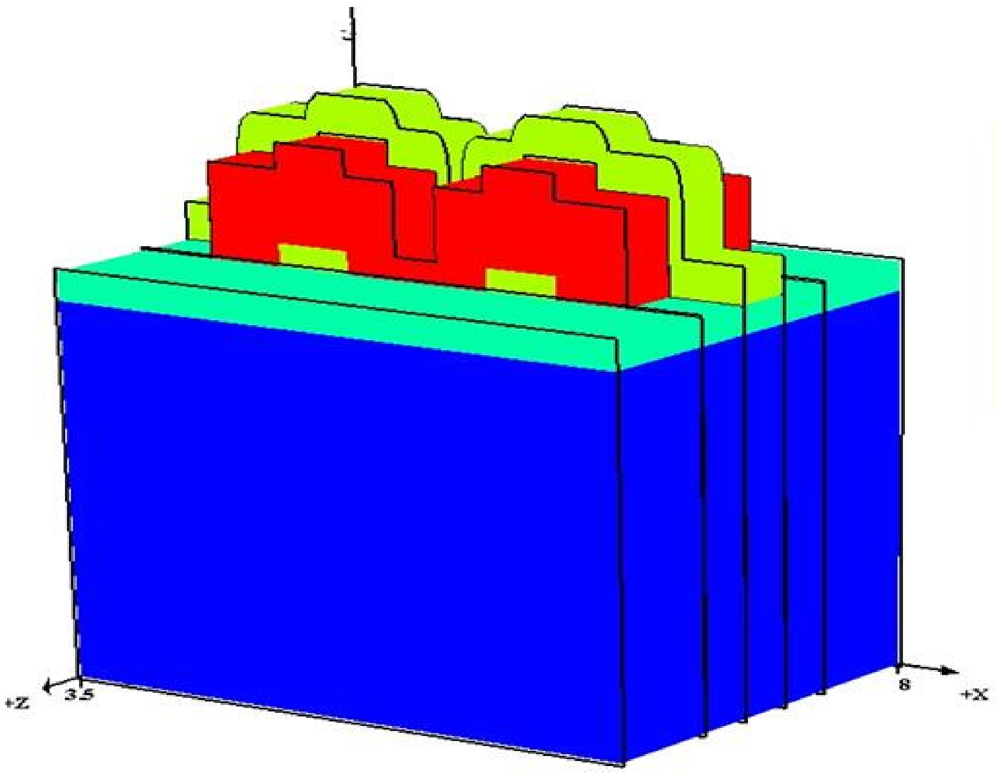

Figure 8, Radio frequency MEMS before 3D etching

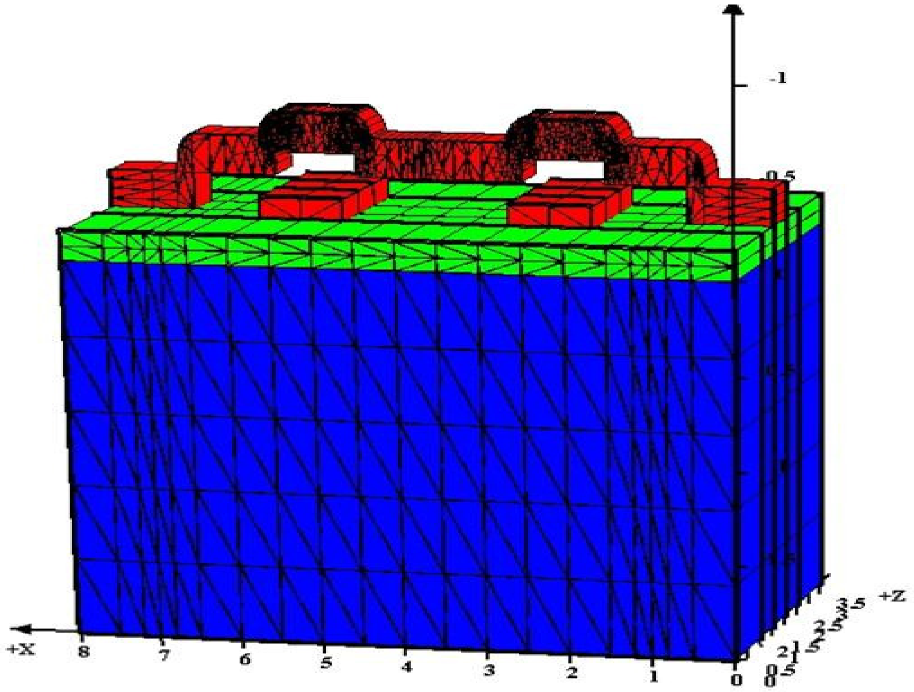

Figure 9, Radio frequency MEMS after 3D improved etching

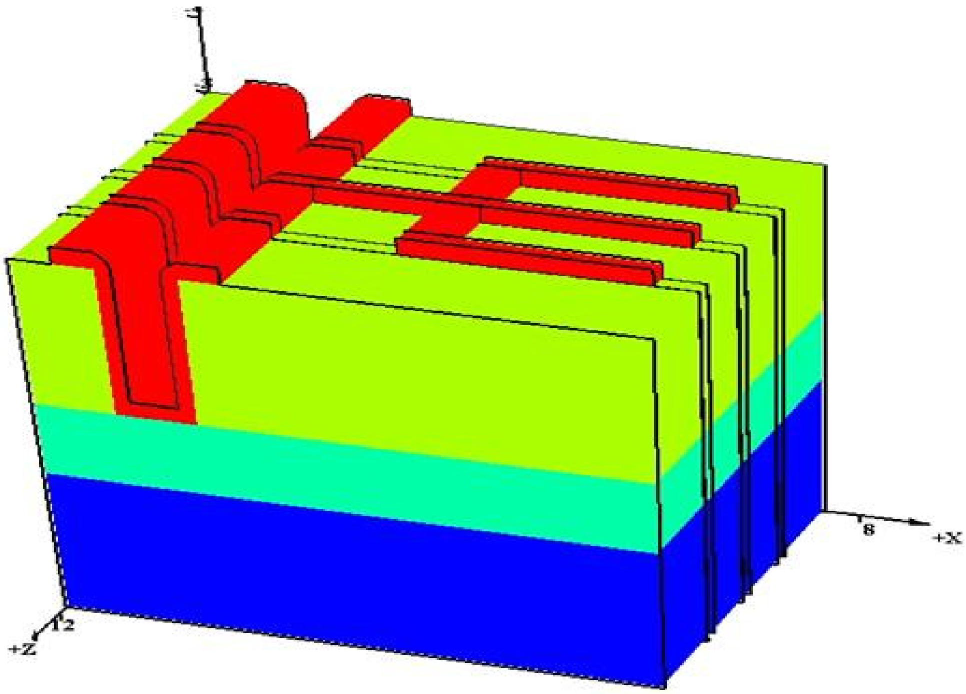

Figure 10, Electrometer MEMS before 3D etching

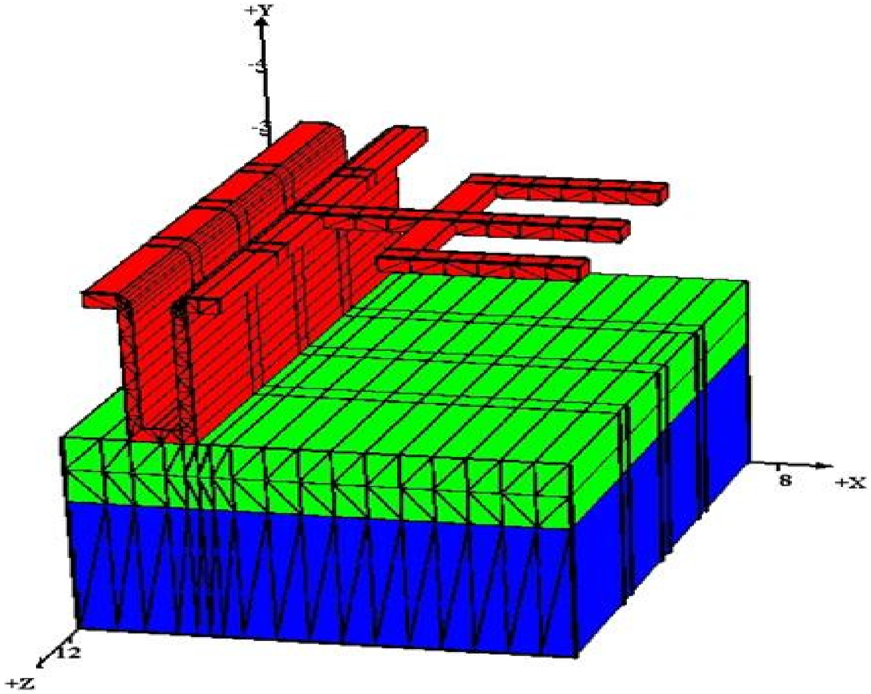

Figure 11, Electrometer MEMS after 3D improved etching

REFERENCES

[1] G. L. Christensen, A.T.T. Tran, Z.H. Zhu, Y.H. Lo, M. Hong, J.P. Mannaerts, R. Bhat, (1997) "Long Wavelength Resonant Vertical-Cavity LED/Photodetector with a 75-nm tuning range", *IEEE Photonics Technology Letters,* vol.9, pp725-727.

[2] D.C. Garett, C. Qi, C. Chaung-Yeng, W. Shaomin, S.W. Chad, C.M. Noel, J.E. Bowers, (2005) "MEMS-Tunable Vertical Cavity SOAs", *IEEE Journal of quantum electronics,* vol. 41, No 3, March.

[3] D.C. Garett, (2005) "MEMS-Tunable Vertical Cavity SOAs", Ph.D. Thesis, University of California, Santa Barbara, CA, USA.

[4] W.V. Glenn, (2000) "*A Three-Dimensional Front tracking Algorithm for Etching and Deposition processes*", Ph.D. Thesis, State University of New York, Stony Brook, USA.

[5] S.E. Hansen, M.D. Deal, (1993) "*User's Manual of SUPREM-IV.GS: Two dimensional Process Simulation for Silicon and Gallium Arsenide*", Integrated Circuit Laboratory, Stanford University, Stanford, California, USA.

[6] M.C. Larson, A.R. Massengale, J.S. Harris, (1996) "Continuously Tunable Micromachined Vertical-Cavity Surface Emitting Laser," *Electron Letter*, vol. 32, pp330-332.

[7] Law, M.E., 1994., (1994) "*User's Manual of FLOOPS: Florida Objected Oriented Process Simulator Manual,*" Dept. of Elect. Eng., University of Florida, Florida, USA.

[8] W.G. Oldham, A.R. Neureuther, C. Sung, J.L. Reynolds, S.N. Nandgaonakar, (1980) "A General Simulator for VLSI Lithograpgy and Eteching Process: PartII-Application to Deposition and Etching," *IEEE Trans Electron Devices*, vol.ED-27,pp1455-1459.

[9] W.S. Rabinovich, T.H. Stievater, N.A. Papanicolaou, D.S. Katzer, P.G. Goetz, (2003) "Demonstration of a Microelectromechanical Tunable Asymmetric Fabry-Perot Quantum Well Modulator ," *Applied Physics Letter* , vol.83, pp1923-1926.

[10] Z.H. Sahul, (1996) "Grid and Geometry Servers for Semiconductor Process and Simulation," Ph.D. Thesis, Stanford University, Stanford, CA, USA.

Authors

Abderrazzak El Boukili received both the PhD degree in Applied Mathematics in 1995, and the MSc degree in Numerical Analysis, Scientific Computing and Nonlinear Analysis in 1991 at **Pierre et Marie Curie University in Paris-France**. He received the BSc degree in Applied Mathematics and Computer Science at **Picardie University in Amiens-France**. In 1996 he had an industrial Post-Doctoral position at Thomson-LCR company in Orsay-France where he worked as software engineer on Drift-Diffusion model to simulate heterojunction bipolar transistors for radar applications. In 1997, he had European Post-Doctoral position at University of Pavia-Italy where he worked as research engineer on software development for simulation and modeling of quantum effects in heterojunction bipolar transistors for mobile phones and high frequency applications. In 2000, he was Assistant Professor and Research Engineer at the University of Ottawa-Canada. Through 2001-2002 he was working at Silvaco Software Inc. in Santa Clara, California-USA as Senior Software Developer on mathematical modeling and simulations of vertical cavity surface emitting lasers. Between 2002-2008, he was working at Crosslight Software Inc. in Vancouver-Canada as Senior Software Developer on 3D Process simulation and Modeling. Since Fall 2008, he is working as Assistant Professor of Applied Mathematics at Al Akhawayn University in Ifrane-Morocco. His main research interests are in industrial TCAD software development for simulations and modeling of opto-electronic devices and processes. http://www.aui.ma/personal/~A.Elboukili