Cellular Automata Modeling in MEMS Design *†

Ted J. Hubbard‡
Erik K. Antonsson§
Engineering Design Research Laboratory
Division of Engineering and Applied Science
California Institute of Technology

January 25, 1995

Abstract

This paper presents a robust Cellular Automata model which predicts the three dimensional etched shape as a function of time for arbitrary etchants and arbitrary initial mask shapes. This method applies the basic approach of all Cellular Automata to etching: to divide the spatial domain into small cells, provide each cell with simple and primitive (but appropriate) behavior, and the aggregate behavior of many cells will mirror the complex behavior of physical systems. The predictions of the model are compared with experimental results and show excellent agreement.

1 Introduction

Many useful micro electro mechanical systems (MEMS) are now being built using silicon etching technologies. Proposals for MEMS computer aided design (CAD) systems have been made in recent years and considerable work has been done to establish the best architecture for such a system [1, 2]. While much work has been done in other parts of CAD systems, there remains a need for an improved etch simulator. The fundamental problem is how to model the complex transformation from the initial two dimensional input mask to the final three dimensional output shape, particularly when highly anisotropic etchants are used.

A new etch simulation method is presented in this paper based on Cellular Automata. The basic approach is to divide a wafer of silicon into small cubic cells,

*Manuscript prepared for submission to the *IEEE/ASME Journal of Microelectromechanical Systems*

†Word Count: 3100 words.

‡Graduate Research Assistant

§Corresponding author: Associate Professor of Mechanical Engineering, Mail Code 104-44, Caltech, Pasadena, CA 91125, FAX: 818/568-2719, e-mail: erik@design.caltech.edu

and each cell is given a few primitive rules governing its rate of removal when exposed to an etchant. If these few simple rules are properly written, then the aggregate behavior of all the cells will accurately represent the complex geometry of a silicon wafer being etched. Finite element analysis (FEA), computational fluid dynamics (CFD), and other methods are all based upon this approach. This approach permits any etchant to be simulated, by suitable choice of rules for each cell, and can exhibit spatial and temporal non-uniformities, (such as mask erosion and undercutting, etchant aging, etc.) as seen in silicon etching. It also can easily and accurately model complex interactions between etched shapes, such as when one etched regions intersects with another, or when an etched shape intersects itself.

When silicon is etched with anisotropic etchants, the etched shape changes as a function of time. A number of different approaches exist to accurately predict the etched shape given an initial mask [1, 3, 4, 5]. The ASEP program by Buser [6] uses traveling planes and the intersections between them to define the shape as a function of time. The Slowness method of Sequin [7] examines the corners of a shape and predicts the trajectory of those corners. This is done using a vector expression involving the inverse of the rate (the slowness) of the planes which make up the corner. The Eshape method of Hubbard and Antonsson [8] precalculates the traveling planes and then extracts the etched shape section by section. These models deal well with most basic shapes. However, complex shapes are more difficult for theses methods to simulate. For example, when two distinct shapes merge to form a new shape (termed through-cut), substantially more computation must be done to detect and locate the new intersections and corners.

The robust Cellular Automata model presented here predicts the three dimensional etched shape as a function of time for arbitrary etchants and arbitrary initial mask shapes. Some preliminary work has been done on atomistic approachs [7, 9], however this paper will focus on Cellular Automata and address, at greater length, issues such as generality, efficiency, and speed. The model can simulate very complicated geometry and has moderate computational requirements.

2 Modeling Methods

The ASEP, Slowness, and the Eshape methods (mentioned above) are high level; the fundamental modeling unit or primitive is defined in terms of endpoints with an associated interior. The intersection points between two or more units are calculated and new endpoints delimiting new interiors are determined. In two dimensions the units are line segments while in three dimensions they are planar polygons. For example a sharp two dimensional corner is defined by the moving intersection point of the two segments that make up that corner.

The advantage of this approach is that each unit is defined at a relatively high level of abstraction. Thus relatively few units are needed to define shapes. The disadvantage of this approach is the difficulty of calculating interactions between etching primitives. In general as the etched shapes are modeled by fewer, more complex primitives the computational intensity required to calculate the change in primitives increases. When only local calculations are performed, the number of

computations is low since only only a few neighbors may interact. Unfortunately, the most interesting phenomena (through-cut, shape intersections, compensation, etc.) require global calculations. This means that all primitives must be checked against each other for interactions. There are hierarchical testing schemes that narrow the possible interactions using efficient tests, and then examine only those that pass the first tests, but the computational cost is still high. Thus the attributes of high level models are best matched to simple geometry and unchanging topology.

The Cellular Automata model in contrast, uses a low level representation, each shape is represented by a sequence of cells. Each cell interacts *only* with its neighbors, there are no global interactions. Moreover all interactions are computationally efficient. Thus Cellular Automata always perform many calculations, but the individual calculations are small and the net cost (number of calculations times the cost of each calculation) will be less than other methods for typical etched shapes and resolutions. Cellular Automata excel at complex geometries with changing shape and topology.

2.1 Scaling

The Cellular Automata method is based on the principal of scaling presented in [8]. For a given mask shape, the form of the etched shape does not change if the initial mask size is scaled, although the output size and effective etch time do change. Thus the mask can be reduced to smaller and smaller sizes until the crystal nature of silicon becomes important. In effect, the Cellular Automata method replaces many atoms in a shape with one atom (or cell). Within the shape, groups of atoms behave as if they were only one atom obeying crystallographic laws. In this way the number of required primitives is minimized. Linearity (in the sense of scaling [8]) is necessary for this to be true, but the etching itself may be nonlinear in space (meaning that the etched shape changes with time).

2.2 Empirical vs. theoretical

Our analysis of the etching of silicon has been empirical rather than theoretical. The theoretical approach would involve the study of the chemistry of both the silicon and the silicon etchant. However such chemical reactions are very complex and difficult to model. Our emphasis has been on using experimental data and experimental observations to formulate empirical rules that accurately model the etching. The use of such rules allows for the efficient implementation of modelers which would otherwise be very computationally intensive. In formulating these etching rules two guidelines must be followed: (i) the rules must produce accurate results (ii) the model rules must agree with the microscopic chemical behavior.

The empirical rules used in the Cellular Automata model reduce the complexity of the interactions between cells while still producing accurate predictions of the changing shapes as a function of time.

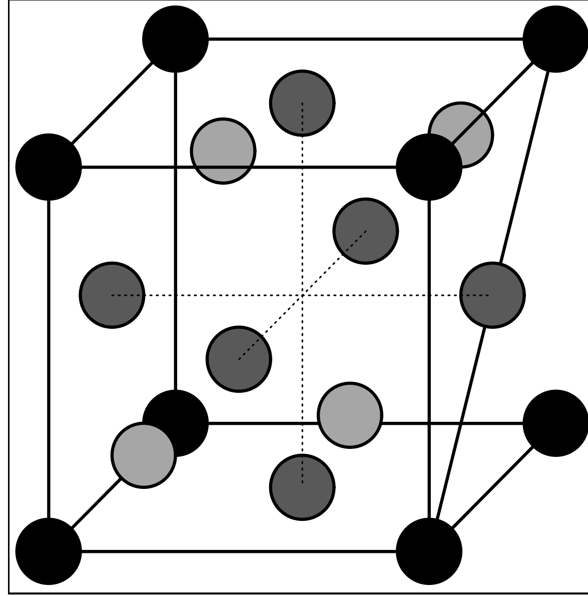

Figure 1: Silicon basis cell.

3 Two Dimensional Algorithm

The Cellular Automata model will first be presented in two dimensions then extended to three dimensions.

To begin developing a two dimensional Cellular Automata etch simulation, consider the diamond lattice crystal structure of silicon as shown in Figure 1. For the purposes of simulation the diamond structure of the lattice is converted into an array of cells. If we examine a plane of atoms parallel to the unetched surface of a (100) wafer, one lattice unit cell deep, from above we see the pattern shown in Figure 2 and Equation 1 (the depth within the page is not shown). An array of cells is constructed such that it contains a 1 when an atom is present and a 0 (or is empty) when no atom is present. Each two dimensional unit cell resembles the number 5 on a throwing die, and has four nearest neighbors, as shown in Equation 1.

$$\textit{cell array} = \begin{bmatrix} 1 & & 1 & & 1 & & 1 & & \cdots \\ & 1 & & 1 & & 1 & & 1 & \\ 1 & & 1 & & 1 & & 1 & & \cdots \\ & 1 & & 1 & & 0 & & 0 & \\ 1 & & 0 & & 0 & & 0 & & \cdots \\ & 1 & & 0 & & 0 & & 0 & \\ \vdots & & \vdots & & \vdots & & \vdots & & \ddots \end{bmatrix} \qquad \textit{unit cell} = \begin{bmatrix} 1 & & 1 \\ & 1 & \\ 1 & & 1 \end{bmatrix} \qquad (1)$$

Once the etching begins, at each cell in sequence, the number of nearest neighbors is counted along with the number of next-nearest neighbors. From this information,

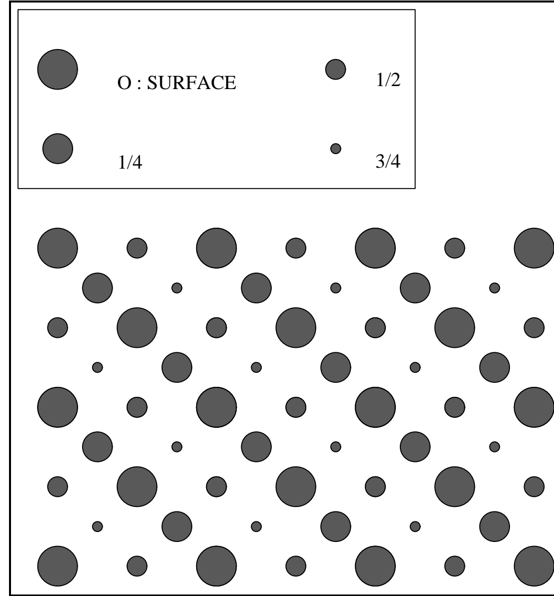

Figure 2: Silicon basis cells viewed from above, size indicates depth. This particular figure shows 3 by 2 by 1 unit cells

the local plane is classified as either (111), (101), or (311). For example, the (111) plane has three nearest neighbors while the (101) plane has only two, as illustrated in Figure 3.

As might be expected, the fastest planes (as measured experimentally) have fewer nearest neighbors. Table 1 shows the neighbor conditions for the different planes. These planes were chosen since they preferentially appear in most etched shapes.

At each time step a fraction of each exposed cell is removed, based on the number of nearest and next-nearest neighbors that the cell has at this time step. Since the number of nearest and next-nearest neighbors that the cell has is determined by the plane that the cell resides on, and the rate of etching perpendicular to these planes is

plane	*nearest neighbors*	*next neighbors*	*speed*
(111)	3	2	*slowest*
(100)	2	3	*intermediate*
(110)	2	3	*intermediate*
(311)	2	2	*fastest*

(2)

Table 1: Number of neighbors for etching planes.

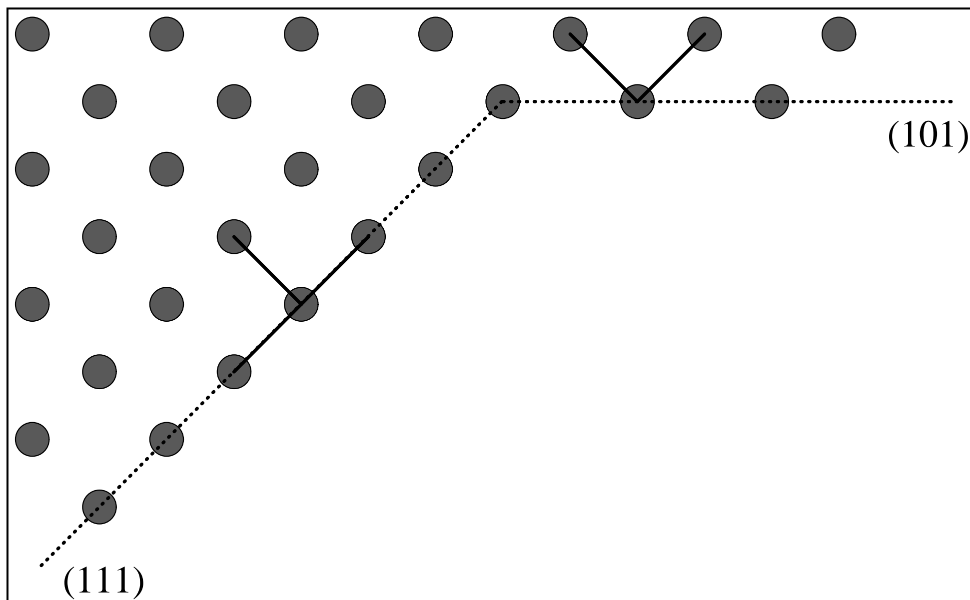

Figure 3: Classification of planes based on neighbors.

known, a calibrated fraction of the cell is removed at each time step. Where removal means that a variable is updated for each cell that represents the remaining fraction. All filled cells start with a 1 (one) for this value. With this scheme, etchants of any rate and selectivity can be modeled, isotropic to anisotropic. When 100 % of a cell is removed, it is removed from further calculations, which exposes cells behind it to etching at subsequent times steps. These steps are then repeated to find the etched shape at each time step. In this way, new planes are introduced automatically.

4 Three Dimensional Extension

This method can easily be extended to three dimensions when a vertical stack of cell arrays is used. Once the topmost array has been calculated for a particular time, it is used to calculate the next lower array. To calculate each succeeding array, at each cell in the lower layer, the five nearby cells above it are examined. If any of these cells are present, the lower cell is protected and kept, otherwise it is removed. This lower array is then used to calculate the one beneath it. This process is repeated for each time step. The appendix lists a pseudocode implementation of the Cellular Automata method.

The three dimensional algorithm described above is valid for etchants modeled by (111), (100), (101), and (311) planes. This is the case for a (100) wafer etched with EDP. Different etchants have different dominate planes; with KOH etching, the (101) planes (45 degree walls) are replaced by the (010) planes (vertical walls). If a different etchant is to be modeled, a slightly different three dimensional extension

algorithm must be used.

A sample of the Cellular Automata simulations (using EDP as the etchant) is shown in Figure 4. Figure 5 shows the result of the Cellular Automata simulation for an initial cross shaped mask. The simulation shows good agreement with the experimental results (see Figure 6). Note that the bounding planes for long times are the (111) planes.

5 Full Three Dimensional Algorithm

In addition to the two dimensional model extended to three dimensions, a full three dimensional algorithm was developed. In this case the actual diamond lattice is modeled with a three dimensional array having elements at each atom location. In three dimensions, each atom has four nearest neighbors. Referring to the five atom unit cell of two dimensions (see Equation 1), the central atom remains the same but the northeast and southwest atoms are one atomic level lower while the southeast and northwest atoms are one atomic level higher. Each cell also has twelve next-nearest neighbors. As with the two dimensional case, a classification scheme is developed based on the the number of nearest and next-nearest neighbors. This algorithm has been used to predict output shapes with results similar to those for the extended two dimensional case. However this algorithm is slower. One reason for the slowness is that this array is not symmetric. Planes at +45 degrees (along the southeast/northwest line) will differ in height from planes at -45 degrees (along the southwest/northeast line) by two atom heights. Thus the results have to be averaged over two atom heights, and the number of calculations is doubled. The two dimensional extension discussed above in effect removes this asymmetry.

5.1 Two dimensions vs. Three dimensions

Both the two and three dimensional algorithms produce similar results, and both classify planes according to the number of neighbors with more neighbors meaning a denser plane. However it is important to note that they do so by using two different classification schemes. The three dimensional algorithm is a general approach, while the extended two dimensional algorithm is a specialized subset of the full three dimensional case which uses certain conditions (e.g. a (100) wafer) to reduce the complexity of the rules.

The choice of classification rules involves a trade off between accuracy of modeling and speed of implementation. A very complex rule system involving many neighbors will provide a more detailed model of the etching, but at the price of increased computation. In practice we choose the extended two dimensional algorithm over the full three dimensional algorithm since it provides a sufficiently accurate result in a faster, more efficient implementation.

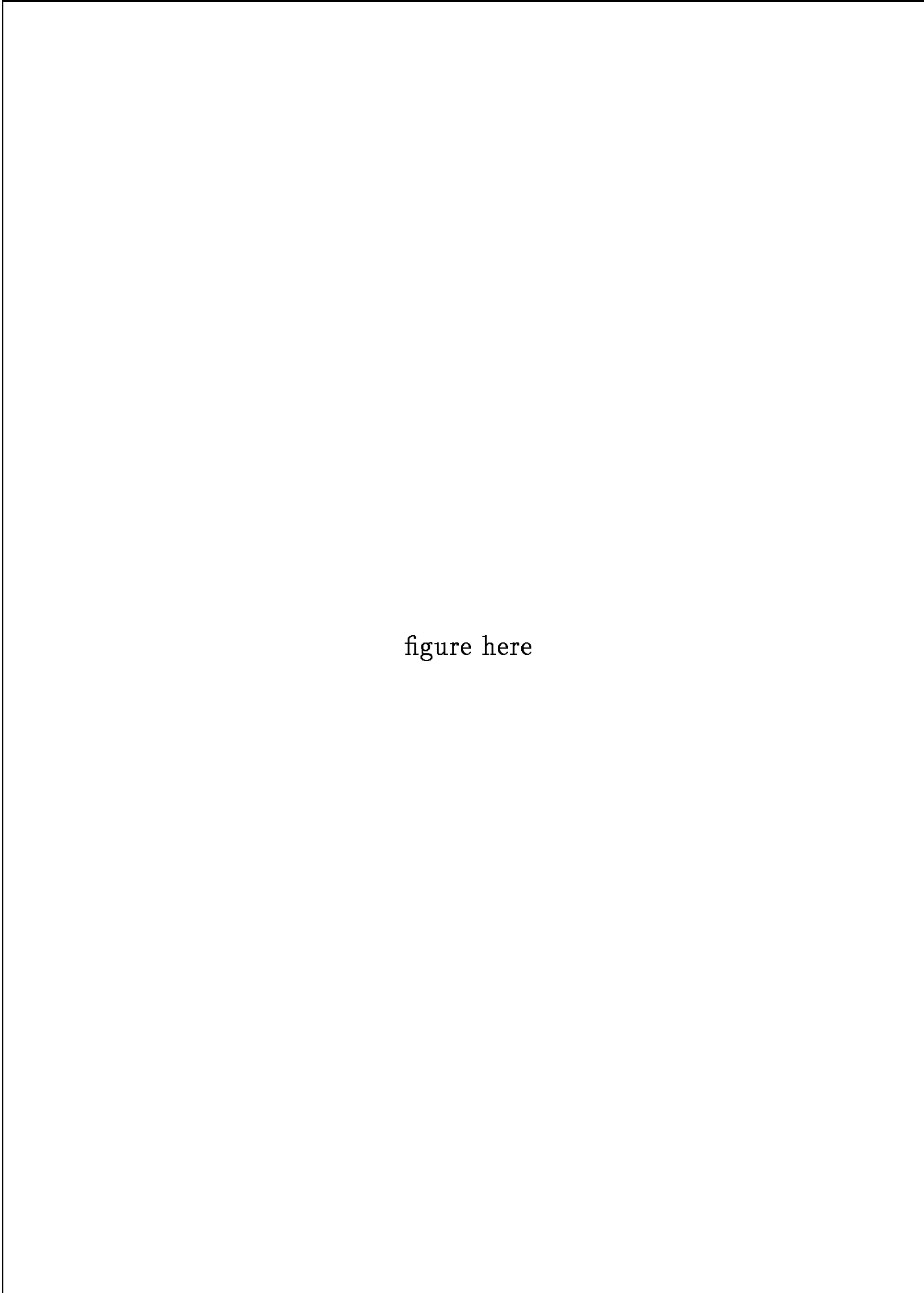

Figure 4: Sample of Cellular Automata algorithm output.

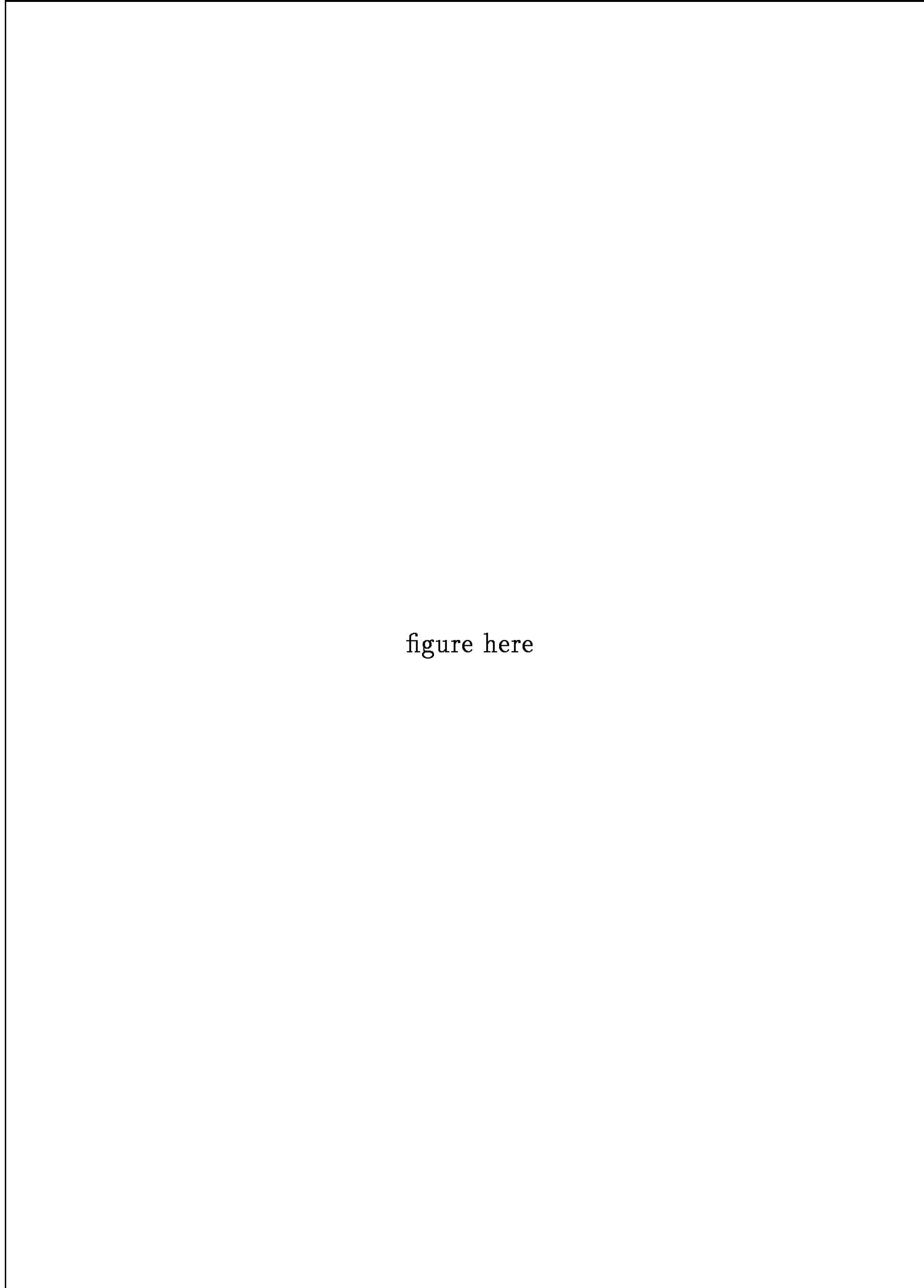

Figure 5: Cellular Automata algorithm output for initially cross shaped pegs.

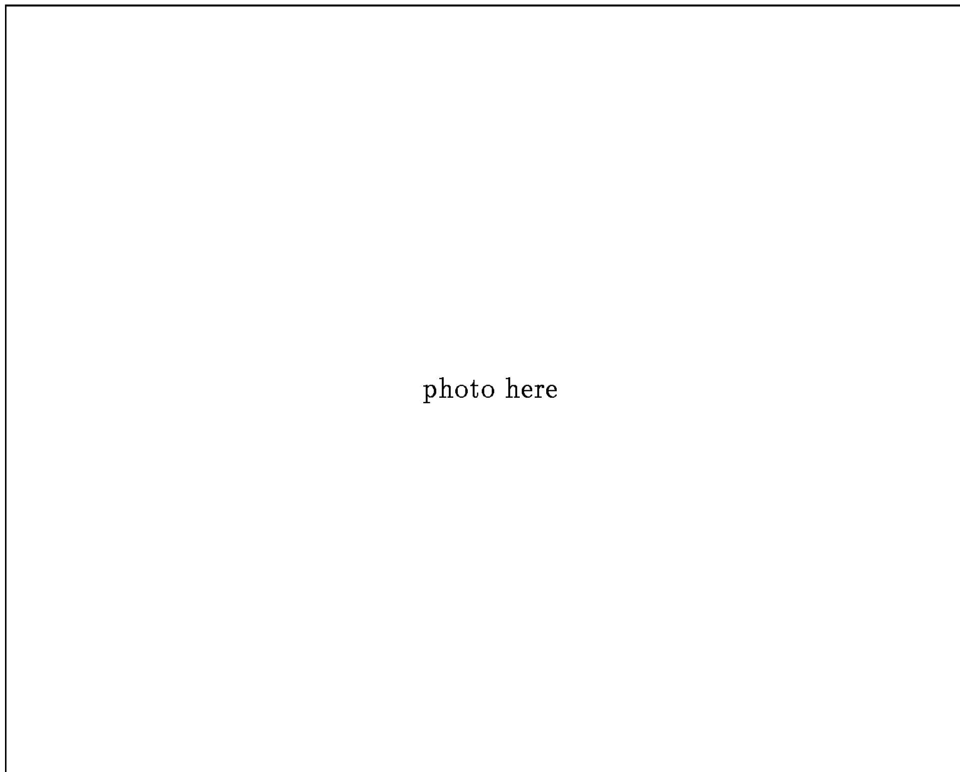

Figure 6: Experimental output shapes for initially cross shaped pegs.

6 Computational Cost

Consider an area divided into N by N cells. Most higher level etch simulations scale on the order of N in two dimensions, but Cellular Automata fair less well. They scale as N^3 (N^2 cells are needed and the effective time scales with N). If the size of the array is doubled, then twice as many time steps are required (since the cell size is halved). In three dimensions, all models multiply an extra factor of N in the scaling.

For three dimensional Cellular Automata, sample calculations for twenty time steps of a 50 by 50 cell input mask required several seconds on a Unix work station. A 100 by 100 array for 40 time steps would require a few minutes, a 200 by 200 for 80 time steps would require on the order of half an hour.

One proposed solution to the rapid rise in processing time is a perimeter based Cellular Automata method. Such a method would examine only the boundary of shapes rather than the entire cell array, thus decreasing the number of cells which must be examined and potentially increasing the modeling speed.

7 Conclusion

A new Cellular Automata etch simulation method has been developed and presented. This method exhibits several advantages over existing geometry based etch simulators in that complex geometry, through-cut, and temporally and spatially varying etch rates can be modeled. Very little computation is performed for each cell, and the rules embodied by each cell are simple. The aggregate behavior of these simple cells accurately reflects the geometry observed in silicon etching.

Despite the computational limitation of the method (which imposes a trade off between the spatial resolution of the CA etch simulation technique and the simulation time) the approach's robustness to arbitrarily complex geometry and an unlimited number of etched shape intersections and interactions, provides a clear advantage for many MEMS design problems.

8 Appendix 1

Listed below is a pseudocode implementation of the extended two dimensional Cellular Automata method. Comments are denoted by (* comments *).

```
1)    Create (N x N x depth) array
      Create (N x N x depth) temporary array
2)    Transfer mask in array level 1
3)    Input percentages to be removed for each plane
4)    (* Two dimensional algorithm *)
      For each (i,j) element in level 1:
            count occupied nearest cells: (i+1,j-1), (i+1,j+1), (i-1,j-1), (i-1,j+1)
            count occupied next-nearest cells: (i,j+2), (i,j-2), (i+2,j), (i-2,j)
            use Table 1 to classify plane
            remove proper percentage of cell (i,j) in temporary array
      Switch level 1 of array and temporary array
5)    (* Three dimensional extension *)
      Set depth level k=2
5.b)  For each (i,j) element in level k:
            count occupied nearest cells in level k-1: (i+1,j-1), (i+1,j+1), (i-1,j-1), (i-1,j+1)
            if count = 0 then remove cell (i,j) in level k of temporary array
      Switch array and temporary arrays
      Set depth level k=k+1, while k < depth loop to 5.b
6)    Write array
7)    Loop to step 4 for next time step
```


Acknowledgments

The authors wish to gratefully acknowledge Professor Y.C. Tai for the use of his facilities and many helpful discussions. This material is based upon work supported, in part, by the National Science Foundation under Grant No. ECS-9023646. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsor.

References

[1] S. D. Senturia, R. M. Harris, B. P. Johnson, S. Kim, M. A. Shulman, and J. K. White. A computer-aided design system for microelectromechanical systems (MEMCAD). *Journal of Microelectomechanical Systems*, 1:3–13, March 1992.

[2] K. D. Wise. Integrated microelectromechanical systems: A perspective on MEMS in the 90s. In *MEMS '91*, pages 33–27, Institute of Electrical Engineers, 1991.

[3] G. DeLapierre. Anisotropic crystal etching: A simulation program. *Sensors and Actuators*, 31:267–274, 1992.

[4] D. W. Shaw. Morphology analysis in localized crystal growth and dissolution. *J. Cryst. Gr.*, 47:509–517, 1979.

[5] T. Thurgate. Segment based etch algorithm and modeling. *IEEE Transactions on computer-aided design*, pages 1101–1109, sep 1991.

[6] R. A. Buser and N. F. de Rooij. ASEP: A CAD program for silicon anisotropic etching. *Sensors and Actuators*, 28:71–78, 1991.

[7] C. H. Sequin. Computer simulation of anisotropic crystal etching. In *Transducers '91*, pages 801–806, San Francisco, CA, USA, 1991.

[8] Ted J. Hubbard and Erik K. Antonsson. Emergent Faces in Crystal Etching. *Journal of Microelectomechanical Systems*, 3(1):19–28, March 1994.

[9] H. Camon and A. Gue. Modelling of an anisotropic etching in silicon-based sensor applications. *Sensors and Actuators*, 33:103–105, 1992.