

# SIMULATION OF ETCHING PROCESSES FOR MEMS USING CELLULAR AUTOMATA

*Maria del C. Sevillano Rivera*  
*Advisor: Dr. José Soltero*  
*University of Puerto Rico at Humacao*

## ABSTRACT

We discuss our implementation of a simulation of an etch, by means of a solvent, of a homogeneous substrate using a mask. This implementation uses a cellular automata. A 2D version of the simulation is currently completed. The simulation showing an etching process although the channel shape is polygonal. An improvement of the simulation will consider the effect of the solvent on the substrate depends on various factors including the angle of incidence. Better graphics, extension to 3D and representation of non-homogeneous media will be added.

## 1. INTRODUCTION

The simulation or modeling of a system is the way to imitate the operation of a real system. It is the execution of a model, represented by a computer program that gives the information on the system that is being investigated; so that a design can be evaluated and be corrected before its manufacture. This representations and evaluations give us important information, especially for Micro electromechanical Systems and devices; which are devices that contain extremely small mechanical elements.

Now, Cellular Automata are an idealization of a physical system whose space and time are discrete. Although Cellular Automata have been reinvented several times under different names, this concept dates from the end of 40's; being used through the years in different fields. The pioneer of the Cellular Automata was Jon Von Neumann (1903-1957), who was involved in the design of the first digital computers. Cellular Automata according to Von Neumann are a discrete universe made up of cells. Each cell is characterized by an internal state that typically consists of a finite number of bits information. This system evolves in discrete time steps, the rule that determines its evolution is the same one for all cells, and it is a function of the states of its neighbor cells.

In order to understand the concept of a Cellular Automata we need to know the group of elements, the neighbors, the group of states and the rule of local transition. The group of elements is the simple objects that constitute the population of the system. In principle the

elements do not have restriction to their number, that is, they can be from few to infinity.

For each element of the system, as much inner as of border, it is necessary to establish its neighborhood. The neighborhood, are those other elements that will be exerting a direct influence on a specific cell; usually the cells that surround it. The neighborhood of an element in the border, that is, those elements that are located in the edges or the corners, must be considered in a specific way, so that their neighbors do not exceed the limits imposed.

The group of states defines how the elements are in each instant; and the rule of local transition, the dynamics of the given system. Given a moment and a certain element, the rule of local transition gives back the following state of the element, using the actual states; as much as for the element in consideration, as for the ones that conform its neighborhood.

## 2. METHODS

As a first step in the process of making a complex simulation we began with a simple simulation in two dimensions that show how the substrate will dissolve with the given initial conditions.

First an array in two dimensions was defined (a matrix), initialized so that it represented a substrate. Then the first rows that will simulate the mask and the solvent were also initialized. The initial state of the matrix can be represented as in figure 1. Ampersands (@) represent the mask, dots (.) represent the solvent, X's represent the substrate.

```
@@@@@@@@@@@@@@ . . . . @@@@@@@@@@@@@@
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Figure 1. Example of an initial state of the simulation.

The concept of Cellular Automata was used to define a series of rules in which the behavior of each element in the array (cell) at each timestep depends on its neighborhood; taking in consideration the edges and the corners from the matrix. For the neighborhood it was used the Von Neumann Neighborhood, which takes in consideration four sides (above, down, right and left) of the cell or element that is going to be verify.

A piece of the code using C, for the cellular automata iterations reads as follows:

```
for(i=0; i<HEIGHT; i++){
  for(j=0; j < WIDTH; j++)
  {
    if(cell[i][j] == SUSTRATO){
      // left border
      if ((j==0 && i!=0 &&
        i!=HEIGHT-1) &&
        (cell[i-1][j]==SOLVENT ||
        cell[i][1]==SOLVENT ||
        cell[i+1][0]==SOLVENT))

        cell[i][j]=DISOLVED;

      // right border
      else if((j==WIDTH-1 &&
        i!=HEIGHT-1 && i!=0) &&
        (cell[i-1][j]==SOLVENT ||
        cell[i][WIDTH-2]==SOLVENT
        || cell[i+1][j]==SOLVENT))

        cell[i][j]=DISOLVED;

      // inferior border
      else if((i==0 && j!= WIDTH-1
        && j!=0)&&
        (cell[i][j+1]==SOLVENT ||
        cell[i][j-1]==SOLVENT ||
        cell[i+1][j]==SOLVENT))

        cell[i][j]=DISOLVED;

      // upper border
      else if((i==HEIGHT-1 && j!=0
        && j!= WIDTH-1) &&
        (cell[HEIGHT-1][j+1]==SOLVENT
        || cell[HEIGHT-2][j]==SOLVENT
        || cell[HEIGHT-1][j-1]==
        SOLVENT))
        cell[i][j]=DISOLVED;

      /* cell[i][j] is an interior
      cell */
      else if((i!=0 && i!=HEIGHT-1
        && j!= WIDTH-1 && j!=0 )&&
        (cell[i-1][j]==SOLVENT ||
```

```
cell[i][j-1]==SOLVENT ||
cell[i][j+1]==SOLVENT ||
cell[i+1][j]==SOLVENT))

        cell[i][j] = DISOLVED;
      }
    }
  }

  /* convert dissolved to solvent */
  for(i=0; i<HEIGHT; i++)
    for(j=0; j < WIDTH; j++)
      if(cell[i][j] == DISOLVED)
        cell[i][j] = SOLVENT;
```

were i and j represent rows and columns of the matrix representing the substrate, respectively.

Using a loop, the elements of the interior of the matrix were checked to determine their behavior. Each element that fulfills the rules was assigned with another value to assure the previous state of the matrix. When the loop finishes checking the entire matrix, the new value is translated to the state that is wanted to represent, and it is printed in screen.

### 3. RESULTS

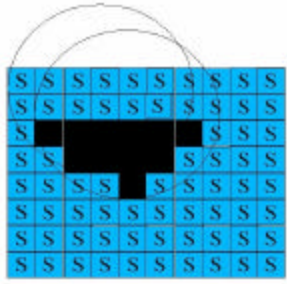
As a result of this investigation its clear that the results of the method used in the 2D version of the simulation is not a realistic one, because the representation of the etching at different times or iterations shows a define inverted trapezium shape- like etch (see figure 2), and in a real system we can't define an specific shape for the etch.

```
@@@@@@@@@@@@@@@@.....@@@@@@@@@@@@@@@@
XXXXXXXXXX.....XXXXXXXXXX
XXXXXXXXXX.....XXXXXXXXXX
XXXXXXXXXX.....XXXXXXXXXX
XXXXXXXXXX.....XXXXXXXXXX
XXXXXXXXXX.....XXXXXXXXXX
XXXXXXXXXX.....XXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Figure 2. Example of the text output of the simulation.

A way to make a more realistic simulation would be possible considering a bigger neighborhood for the cells of substrate in contact with the solvent, as seen on the following figure:

Figure 3: Improvement over Cellular Automata Simulation



The effect of the angle of incidence of the solvent on the substrate surface can be simulated by using an ellipse instead of a circle. With a modification to the previous program, involving the angle of incidence and other various factors, we get a more realistic etch, but then we would not be using the concept of a classic Cellular Automata.

#### 4. CONCLUSION

In conclusion we can say that the method using Cellular Automatas does not produce a realistic simulation. On the other hand modifying the program using the angle of incident will be better, but then the classic definition of a Cellular Automata will not apply.

A much better simulation using graphics, 3D and non-homogeneous materials is the next improvement for the desired simulation.

#### 5. ACKNOWLEDGEMENTS

Part of this work was done at the center for *Material research Science and Engineering Center* and supported by NSF Grant #0079909. We also want to thank the *Penn-UPR Partnership for Research and Education in Materials* program (NSF award number DMR-0353730) for its support.

#### 6. REFERENCES

- [1] Hubbard, Ted J. and Antonsson Erik K. (1995) Cellular Automata modeling in Mems Design. *Preprint*.
- [2] Chopard and Droz, *Cellular Automata Modeling of Physical Systems*, Cambridge University Press, UK, 1998.
- [3] Strasser, Ernst and Selberherr, Siegfried. (1995) Algorithm and Models for Cellular based Topography Simulation. *IEEE Transactions on Computer -Aided Design of Integrated Circuits and Systems*, Vol. 14, No.9, September 1995.
- [4] Woo, Neider, Davis, & Shreiner, *OpenGL Programming Guide*, Addison Wesley,2003.