

Faster, exact implementation of the Continuous Cellular Automaton for anisotropic etching simulations

N Ferrando¹, M A Gosálvez^{2,3}, J Cerdá¹, R Gadea¹ and K Sato²

¹ Instituto de Instrumentación de Imagen Molecular, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain

² Dept. of Micro-Nanosystems Engineering, Nagoya University, 464-8603 Aichi, Japan

³ Dept. of Materials Physics, University of the Basque Country (UPV-EHU), Donostia International Physics Center (DIPC), and Spanish National Research Council (CSIC), 20018 Donostia - San Sebastian, Spain

E-mail: nesferjo@upvnet.upv.es

Abstract. The current success of the Continuous Cellular Automata (CCA) for the simulation of anisotropic wet chemical etching of silicon in microengineering applications is based on a relatively fast, approximate, constant time stepping implementation (CTS), whose accuracy against the exact algorithm –a computationally slow, variable time stepping implementation (VTS)– has not been previously analyzed in detail. In this study we show that the CTS implementation can generate moderately wrong etch rates and overall etching fronts, thus justifying the presentation of a novel, exact reformulation of the VTS implementation based on a new state variable, referred to as the Predicted Removal Time (PRT), and the use of a self-balanced binary search tree (SB-BST) that enables the storage and efficient access to the PRT values in each time step in order to quickly remove the corresponding surface atom/s. The proposed PRT method reduces the simulation cost of the exact implementation from $\mathcal{O}(N^{5/3})$ to $\mathcal{O}(N^{3/2} \log N)$ without introducing any model simplifications. This enables more precise simulations (only limited by numerical precision errors) with affordable computational times that are similar to the less precise CTS implementation and even faster for low reactivity systems.

PACS numbers: 89.20.Ff, 85.85.+j, 81.65.Cf

Submitted to: *J. Micromech. Microeng.*

1. Introduction

In the world of microfabrication, anisotropic wet chemical etching is one of the most popular bulk micromachining methods, with several features that make it technologically attractive, such as the low cost of the process, the ability to obtain smooth and flat surfaces, the capability to release suspended structures, and the possibility to process many wafers simultaneously in the same batch [1]. On the other hand, the nature of the process makes it hard to predict the shape of the resulting structures, since many factors affect the results, including the crystal orientation of the surface, the etchant type, its concentration and temperature or even the addition of small amounts of surfactants or alcohols, such as Triton X-100 [2] or isopropyl alcohol (IPA) [3]. Because of this, an extensive effort has been made through the years in order to accurately model and simulate the process for engineering applications as well as for improving our fundamental understanding of the underlying reaction kinetics, physics and chemistry.

A number of recently presented models are based on the Continuous Cellular Automata (CCA) [4, 5, 6]. These models take into account several physical phenomena, such as the *step flow* nature of the process [7], in order to accurately simulate wet etching for most crystal orientations of silicon. One of the most important advantages over the classical geometrical methods [8, 9] is the possibility of obtaining accurate calibrations of the model parameters by using a reduced dataset of experimental etch rates. In addition, Cellular Automata (CA) simulators, such as AnisE [10] and IntelliEtch [11], have progressively appeared in the market, demonstrating in this way the value and potential of the underlying simulation approach.

In spite of the overall success, the computational cost of the presently existing, exact implementation of the CCA method has remained prohibitively high for typical engineering applications. This has forced the use of approximate implementations to perform the simulations within affordable computational times [12]. The present study aims at correcting this situation by presenting a completely equivalent reformulation of the exact model that significantly reduces the computational cost, thus leading to dramatically faster simulations without affecting the accuracy of the results. This is made possible by the introduction of a new theoretical concept, referred to as the Predicted Removal Time (PRT), which, in combination with the use of a self-balanced binary search tree, enables the efficient determination of the next group of atoms to be removed in each time step.

The rest of the paper is organized as follows. The fundamental structure of the CCA model and the two main exact and approximate implementations are introduced in Section 2. This is followed by an analysis of the computational cost of the two implementations in Section 3 and the presentation of the new reformulation of the exact method in Section 4. In Section 5 we provide a comparison of the relative efficiency and error by the three considered implementations, demonstrating the reduction in computational cost between the new and old exact versions as well as the improved accuracy between the new implementation and the traditionally used approximate

method. Finally, we consider a realistic application of the method in Section 6, showing the benefits of the new implementation, and proceed to present our conclusions in Section 7.

2. The Continuous Cellular Automata Model

2.1. Rate equivalence

The Cellular Automata (CA) are models where space and time are described using discrete values. Typically, a CA is formed by a lattice of individual units, referred to as cells, where each cell interacts with the nearby cells. On each time step the state of each cell is defined as a function of its previous state and the state of the neighbors. The objective of any CA model is to define microscopic relationships between neighboring cells in order to describe an observed macroscopic behavior as the result of coupled cell modifications in successive time steps.

In order to simulate wet etching of silicon, the lattice represents the silicon crystalline structure and the cells represent the crystallographic sites where the silicon atoms are located. In this context, the sites can have two possible states, namely, occupied by an atom (1) or empty (0), and the surface sites can be classified according to their neighborhood in terms of four indices (n^{1s} , n^{1b} , n^{2s} , n^{2b}), which take into account the number of surface nearest neighbors (n^{1s}), bulk nearest neighbors (n^{1b}), surface next-nearest neighbors (n^{2s}) and bulk next-nearest neighbors (n^{2b})[14]. In a particularly successful CA method, known as the Continuous Cellular Automata (CCA), the sites are allowed to have any occupation value in the continuous range [0,1], thus enabling the description of smooth parameter variations across the silicon/etchant interface and an essentially continuous propagation of the simulated surface [5, 4, 13, 15].

The etching process in the CCA method is understood as a continuous decrease of the occupation (Π) of the surface sites. When a bulk atom becomes a surface atom due to the removal of one or several of its nearest neighbors, Π is initialized with value '1'. During the simulation, Π is gradually reduced until it reaches value '0' (completely removed). At each time step (k), the reduction in the occupation of the i -th surface site (Π_i) is equal to the current value of the removal rate of the atom that populates the site (r_i), multiplied by the time step (Δt):

$$\Pi_i^{(k+1)} = \Pi_i^{(k)} - \Delta t \cdot r_i^{(k)}, \quad (1)$$

where the atom removal rate r_i depends on the number of neighbors, as in:

$$r_i^{(k)} = R(n_i^{1s}(k), n_i^{1b}(k), n_i^{2s}(k), n_i^{2b}(k)). \quad (2)$$

Here the particular values of R depend on the actual behavior of the etchant (such as KOH, TMAH, etc...), including its concentration and temperature. Until a surface atom is completely removed, its removal rate suffers a number of changes as the numbers of surface and bulk nearest and next-nearest neighbors are modified due to the removals of nearby atoms. For this reason, r_i and n_i^{1s} , n_i^{1b} , ... are written as functions of the time

step k in Equations 1 and 2. Equation 1 is referred to as the principle of *rate equivalence*, stating that the rate of decrease of the occupation of a site is, at any moment, equal to the rate of removal of the atom populating it. In other words, the occupancy reduction rate is equal to the removal rate of the populating atom.

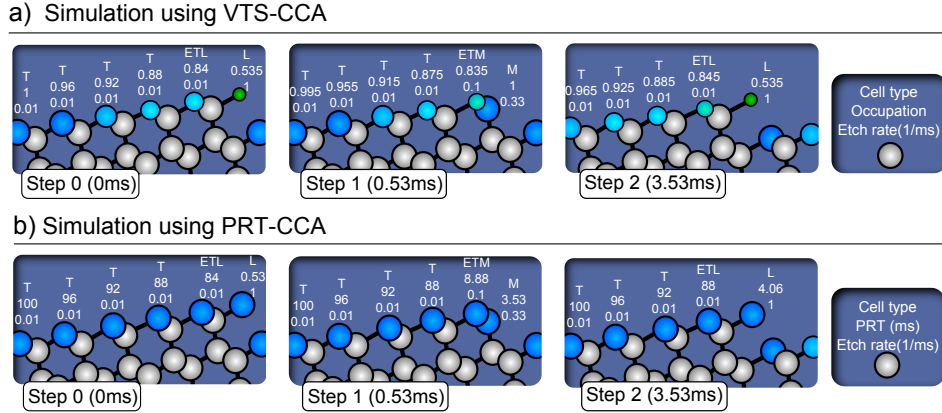


Figure 1. Evolution of selected atom parameters during a CCA simulation of wet etching on a $(h + 2, h + 2, h)$ -oriented silicon surface. Three fields are displayed for each surface atom: (top) the site names [5], (middle) the occupation value (a) / PRT value (b), and (bottom) the occupation reduction rate (= atomistic removal rate).

In practice, the CCA method results in the alternation of removals of specific surface atoms, as shown in Fig. 1(a). On the left-hand-side, the "L" atom is removed and time is stepped correspondingly (in this case to 0.53 ms), whereafter atom "M" is removed in the center snapshot and time is correspondingly incremented (to 3.53 ms). This takes the system to the right-hand-side snapshot, which is identical to the initial state. The removal of L and M atoms in alternation leads to the advancement of the morphological step structure towards the left and the downward propagation of the overall surface. This makes possible to describe the overall etch rate of any surface orientation (*i.e.* the rate of downward propagation) as a function of the removal rates of the atoms appearing on it [5]. Based on this observation, an exhaustive classification of the different surface atoms was made, and the functional dependence of the etch rates of a wide range of crystallographic planes was obtained in terms of the removal rates of the classified atom types [5]. By solving the resulting system of equations, it has been shown that the removal rates of the different atom types can be successfully calibrated in order to describe the overall etch rates of a wide range of crystallographic planes in dramatically different etchants [4]. In order to assign removal rates to atom configurations that do not appear in the equations, the calibration method makes use of the Removal Provability Function (RPF) introduced in Ref. [16].

2.2. Exact and approximate CCA implementations

One of the most important characteristics of a CCA implementation is the way time is evolved during the simulation. By zeroing the left-hand-side of Eq. 1, one can calculate the magnitude of the time increment required for atom i to be completely removed:

$$\Delta t_i^{(k)} = \Pi_i^{(k)} / r_i^{(k)}. \quad (3)$$

For an exact model implementation, the simulated time should be advanced according to the minimum increment found by evaluating Eq. 3 for all the surface atoms. The next atom or group of atoms that share the minimum time increment will be removed and new removal rates will be calculated for the neighboring atoms. Since the magnitude of the minimum time increment will typically change during the simulation, this method is referred to as Variable Time Stepping (VTS). The corresponding CCA implementation is referred to as VTS-CCA. As explained in section 3, in practice the computational cost of the VTS-CCA method is too high to perform realistic simulations for engineering applications. An alternative, faster, exact method is thus desirable.

So far, the accepted solution has been to make the time step constant and to choose a small time increment value in order to avoid the removal of too many atoms whose time increment is larger than the minimum. This is referred to as Constant Time Stepping (CTS) and the corresponding CCA implementation is referred to as CTS-CCA. Although this approach reduces the computational cost of the simulation, it introduces errors due to two main artifacts [12]: (i) the occupation of many surface sites becomes negative just before the occupying atoms are removed (thus, the part of the time increment associated to the negative occupation is effectively lost, while formally it would result in a reduction of the occupation of the neighboring atoms); and (ii) the method enables the simultaneous removal of different groups of atoms that should occur sequentially according to the exact implementation (thus inducing undesired competition between their updated neighbors in the next time step). Although time compensation methods have been introduced in order to improve these errors, none of the methodological modifications completely cures the intrinsic deficiencies of the CTS-CCA method [12].

The previous errors introduce differences in the etch rates of the silicon planes during the simulations. As an example, Figure 2 shows a comparison of the experimental and simulated etch rates for a wide range of crystallographic orientations from $\langle 100 \rangle$ to $\langle 111 \rangle$ and from $\langle 110 \rangle$ to $\langle 111 \rangle$, for four widely different etchants, namely, KOH and KOH+IPA at two different concentrations (11 wt and 37wt) at 70C [17]. For all four instances, the results show that the VTS-CCA etch rates remain closer to the experiment than the CTS-CCA values. The errors are more accused in fast etching planes and the planes of families $\langle h11 \rangle$ and $\langle hh1 \rangle$. Since the errors are dependent of the surface orientation they will lead to changes in the actual simulated etchant anisotropy. Although the CTS-CCA implementation has been traditionally used as a valid alternative to the slower VTS-CCA version, Figure 2 strongly indicates that the use of the exact VTS-CCA method would substantially improve the overall results

obtained in a wide range of wet etching simulations, if only the method implementation would be fast enough. In order to obtain Figure 2, the CCA is calibrated according to the experimental data (continuous line) using the method presented in Ref. [5].

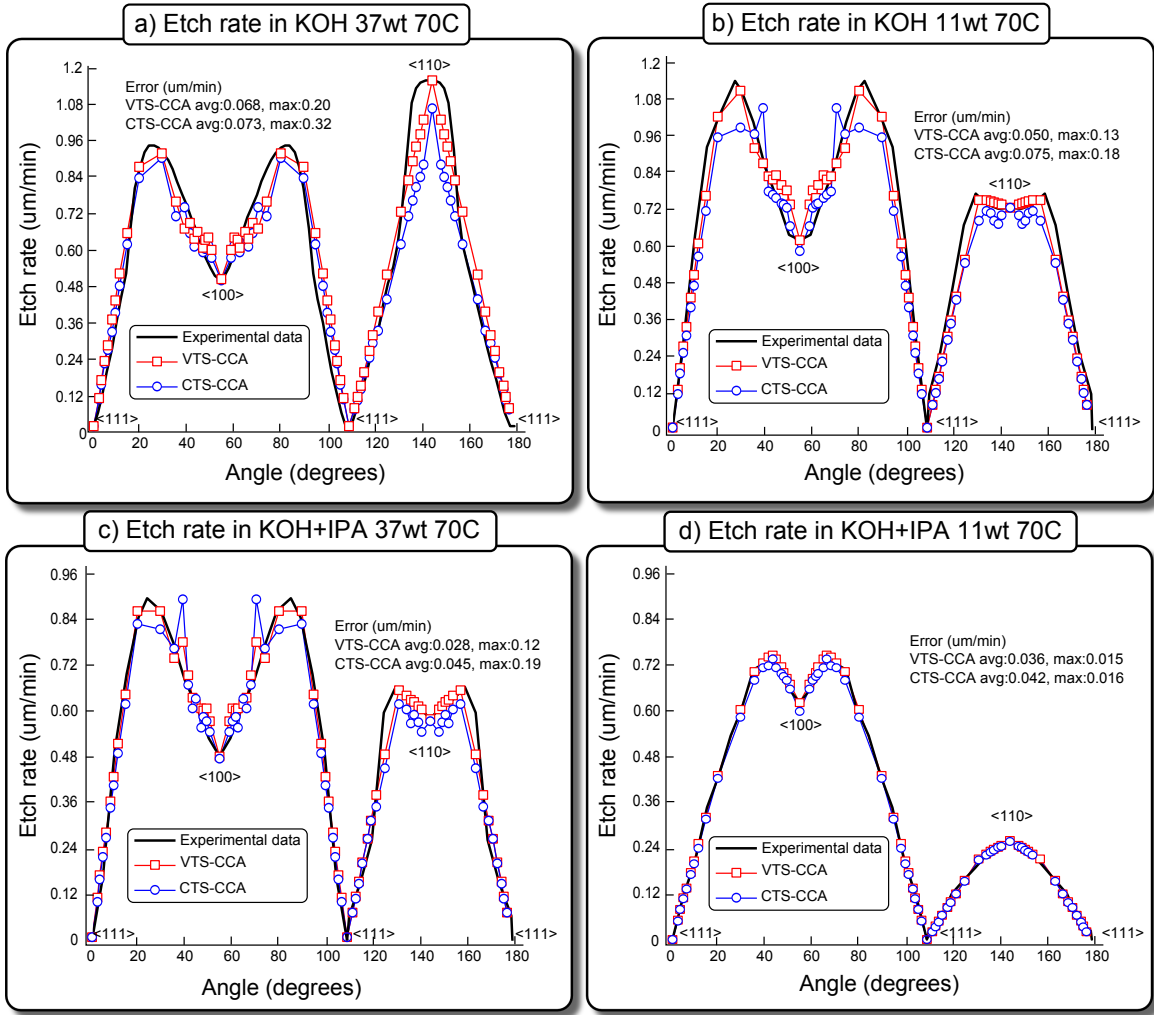


Figure 2. Orientation-dependant etch rates from experiments [17] and, CTS-CCA and VTS-CCA simulations for 11 wt% KOH, 37 wt% KOH, 11 wt% KOH+IPA and 37 wt% KOH+IPA at 70 C.

3. Computational cost of CCA simulations

Classically, a CCA-based simulator keeps in memory a list of surface atoms [15]. As explained in Section 2, the surface atoms are reacting with the etchant and their occupation is being gradually decreased. A direct implementation of this behavior means that the occupation of all surface atoms must be reduced on each time step. In addition, when the occupation of a surface site is completely depleted, the corresponding atom must be removed from the list, the state of the neighbors must be updated and, if new atoms appear in the surface, they have to be added to it.

In this report we use "T" (upper case t) to refer to the computational cost, *i.e.* the time spent by the computer in performing different tasks. We reserve the use of "t" (lower case t) for the simulated time, *i.e.* the time that ticks between the atom removals. The computational cost (T) of the CCA method has two main contributions: the time spent in tasks that decrease the occupation of each surface site (T_{oc}) and the time spent in bookkeeping tasks that update the neighborhood of each removed atom (T_{bk}):

$$T = \sum_{k=0}^{K-1} (T_{oc} \cdot N^{(k)} + T_{bk} \cdot N_{removed}^{(k)}) , \quad (4)$$

$$T \approx T_{oc} \cdot K \cdot N + T_{bk} \cdot K \cdot N_{removed} , \quad (5)$$

where K is the total number of time steps, $N^{(k)}$ is the total number of surface atoms in time step k , $N = \sum_{k=0}^{K-1} N^{(k)}/K$ is the time average of the number of surface atoms, $N_{removed}^{(k)}$ is the number of removed surface atoms in time step k and $N_{removed} = \sum_{k=0}^{K-1} N_{removed}^{(k)}/K$ is the time average of the number of removed atoms per time step. The bookkeeping cost per atom (T_{bk}) depends on the average size of the atom neighborhood, roughly estimated to be ten times larger than the cost of decreasing the occupation per atom (T_{oc}). Eq. 5 assumes that T_{oc} and T_{bk} can be roughly considered to be constant during the course of a simulation.

Let $T^{(RA)} = T/N_{removed}^{total}$ be the average computational cost per removed atom, where $N_{removed}^{total} = K \cdot N_{removed}$ is the total number of removed atoms. In addition, let us define $g = (N_{removed}^{total})^{1/3}/N^{1/2}$ as the aspect ratio of a characteristic depth associated to the removed volume of silicon atoms and a characteristic length associated to the current silicon surface. This definition is based on the fact that two scaled systems $N_2 = e \cdot N_1$ will reach the same propagated state when $N_{removed,2}^{total} = e^{3/2} \cdot N_{removed,1}^{total} = (N_2/N_1)^{3/2} \cdot N_{removed,1}^{total}$, and thus $g_2 = (N_{removed,2}^{total})^{1/3}/N_2^{1/2} = (N_{removed,1}^{total})^{1/3}/N_1^{1/2} = g_1 = constant$. In other words, g is a constant and can be used to express other variables and thus meaningfully compare the total computational cost of equivalent, scaled systems. Using that $K \cdot N_{removed} = N_{removed}^{total} = g^3 N^{3/2}$, Equation 5 gives:

$$T \approx \left(T_{oc} \frac{N}{N_{removed}} + T_{bk} \right) \cdot g^3 \cdot N^{3/2} , \quad (6)$$

$$T^{(RA)} \approx T_{oc} \frac{N}{N_{removed}} + T_{bk} . \quad (7)$$

According to Eq. 7, if the number of removed atoms per time step $N_{removed}$ is similar to the number of visited atoms N , the bookkeeping term $T_{bk} \approx 10T_{oc}$ will dominate the overall cost per removed atom. However, if $N_{removed}$ is significantly smaller than N , the cost will be dominated by the occupation reduction tasks performed for all the surface atoms $T_{oc} \cdot N/N_{removed}$.

As discussed later in Section 5, we make the observation that the computational cost shown in Eqs 6–7 for the CCA method is directly proportional to the simulation *inefficiency*, which we define as $q = N/N_{removed}$, *i.e.* the ratio of the number of visited

atoms to the number of removed atoms:

$$T \approx (T_{ocq} + T_{bk}) \cdot g^3 \cdot N^{3/2}, \quad (8)$$

$$T^{(RA)} \approx T_{ocq} + T_{bk}. \quad (9)$$

3.1. CTS-CCA

Let us define the total rate of the system $R^{(k)}$ as the sum over all the removal rates of all the surface atoms:

$$R^{(k)} = \sum_0^{N^{(k)}-1} r_i^{(k)}, \quad (10)$$

and its spacial average:

$$r^{(k)} = R^{(k)}/N^{(k)}. \quad (11)$$

Typically, R changes from one time step to another since certain surface atoms are removed/added during the previous time step. In the CTS-CCA method, the number of removed atoms $N_{removed}^{(k)}$ appearing in Equations 6 – 7 can be expressed as the total reduction in the occupation of all the surface sites:

$$N_{removed}^{(k)} = \sum_{i=0}^{N^{(k)}-1} \left(\Pi_i^{(k)} - \Pi_i^{(k+1)} \right), \quad (12)$$

$$N_{removed}^{(k)} = \sum_{i=0}^{N^{(k)}-1} r_i^{(k)} \Delta t, \quad (13)$$

$$N_{removed}^{(k)} = N^{(k)} r^{(k)} \Delta t, \quad (14)$$

where Eq. 1 has been used to go from Eq. 12 to Eq. 13. Using Eq. 14, Equations 6 – 7 become:

$$T_{CTS} \approx \left(\frac{T_{oc}}{r \Delta t} + T_{bk} \right) \cdot g^3 \cdot N^{3/2} \approx \mathcal{O}(N^{3/2}), \quad (15)$$

$$T_{CTS}^{(RA)} \approx \frac{T_{oc}}{r \Delta t} + T_{bk}. \quad (16)$$

Equation 15 shows that the total cost is $\mathcal{O}(N^{3/2})$ while Eq. 16 shows that the cost per removed atom is independent of N . Larger removal rates r lead to a lower computational cost per removed atom, due to the induced larger occupation reductions (see Eq. 1). This is related to the fact that larger etch rates r require fewer time steps K in order to remove any predefined number of atoms. Similarly, selecting a bigger time step Δt will also lead to larger occupation reductions (Eq. 1), fewer time steps K and an overall lower computational time per removed atom (Eq. 16). Although the simulations can become computationally fast for large values of r and/or Δt , this will occur at the expense of introducing larger errors.

3.2. VTS-CCA

The VTS-CCA method advances time using the minimum time increment obtained by evaluating Eq. 3 for all the surface atoms. In a typical engineering simulation, thousands of surface atoms cohabit on the etch front, each having a different removal rate and occupation. This heterogeneity is especially pronounced when fast and slow etching planes are simultaneously present in the front, as in the case of systems containing convex corners. In this case, the surface atoms located at the corner regions appear and disappear very rapidly while the atoms situated at the side walls have much longer lifetimes. An exact model implementation requires advancing time from one atom removal at one corner to the next atom removal at another corner with the additional peculiarity that the occupation of *all* the surface sites must be reduced in each time step, even if only one or a few atoms are removed.

As a result, the number of removed atoms $N_{removed}^{(k)}$ appearing in Equations 6 – 7 can be roughly estimated for the VTS-CCA implementation as the number of atoms having the maximum removal rate $n_{r_{max}}$. Thus, Equations 6 – 7 become:

$$T_{VTS} \approx \left(T_{oc} \frac{N}{n_{r_{max}}} + T_{bk} \right) g^3 \cdot N^{3/2} \approx \mathcal{O}(N^{5/2}), \quad (17)$$

$$T_{VTS}^{(RA)} \approx T_{oc} \frac{N}{n_{r_{max}}} + T_{bk}. \quad (18)$$

Eq. 18 shows that N now appears in the computational cost per removed atom. According to Eq. 17, this means that the total computational cost of the VTS-CCA method is between quadratic and cubic $\mathcal{O}(N^{2.5})$: bigger simulated structures require bigger surfaces (N) and many more time steps ($K = g^3 N^{3/2} / n_{r_{max}}$). Although the VTS-CCA implementation is not affected by the global etch rate r of the system, the computational price to get an exact model implementation is a power law dependence on the simulated surface size. As traditionally known, a VTS-CCA simulation can easily take many hours and even a few days.

In summary, we have analyzed the algorithm complexity of the existing CCA methods in this section, obtaining that the total computational cost is $\mathcal{O}(N^{3/2})$ for CTS-CCA and $\mathcal{O}(N^{5/2})$ for VTS-CCA. In addition to the approximate nature described in Section 2.2 for the CTS-CCA method, we have described in this section that its computational cost is directly affected not only by the size of the time step but also by the etchant characteristics through the dependence on the average etch rate r . This means that the simulation of etchants at low temperatures or high concentrations –where the etch rates are low– becomes significantly less efficient than at high temperatures or low concentrations. Although the cost of the VTS-CCA method is independent of the global etch rate r , the power law dependence on the system size makes the method completely unsuitable for realistic engineering simulations. To overcome these deficiencies, a new, exact CCA implementation is presented in Section 4.

4. Efficient Continuous Cellular Automata

An efficient CCA is one whose computational cost is as small as possible. In this section, we present a new methodology for increasing the CCA model efficiency without introducing errors in the time evolution.

4.1. Predicted Removal Time

We introduce a new concept, namely, the Predicted Removal Time (PRT), defined for any surface atom i as the calculated instant (or predicted time) when the occupation of the underlying surface site becomes zero and, thus, the atom will be removed from the surface:

$$PRT_i^{(k)} = t^{(k)} + \Delta t_i^{(k)}, \quad (19)$$

$$PRT_i^{(k)} = t^{(k)} + \frac{\Pi_i^{(k)}}{r_i^{(k)}}. \quad (20)$$

In this definition $t^{(k)}$ is the current value of the simulated time and $\Delta t_i^{(k)}$ the time increment required for atom i to be completely removed (see Eq. 3).

We make the observation that the VTS-CCA simulations can be performed accurately and efficiently by using the PRT as the main state variable instead of the occupation. In fact, only three simple modifications are needed to change the VTS-CCA implementation into a fully functioning PRT-CCA implementation: (i) to completely abandon the procedure that updates the occupation values of all the surface atoms, (ii) to substitute the procedure that decides which atoms need to be removed, now selecting the atoms that have minimum PRT (instead of the previous criterium focused on the atoms that reached zero occupation), and (iii) to update the PRT values of the surface atoms whose removal rates have changed (due to modifications in their neighborhoods) by simply using the following expression:

$$PRT_i^{(k+1)} = t^{(k+1)} + \left(PRT_i^{(k)} - t^{(k+1)} \right) \frac{r_i^{(k)}}{r_i^{(k+1)}}, \quad (21)$$

where $r_i^{(k)}$ and $r_i^{(k+1)}$ are the old and new removal rates, respectively, $PRT_i^{(k)}$ and $PRT_i^{(k+1)}$ are the old and new PRTs, respectively, and $t^{(k+1)}$ is the new value of the simulated time. Equation 21 is easily obtained by writing Eq. 20 for $k + 1$ and using Eq. 1 with $\Delta t = t^{(k+1)} - t^{(k)}$ in order to obtain $\Pi_i^{(k+1)} = \Pi_i^{(k)} - (t^{(k+1)} - t^{(k)}) r_i^{(k)}$. A graphical comparison of the occupation and PRT implementations is given in Figure 1, showing that the two methods are completely equivalent and will lead to exactly the same time evolution. Although the two methods differ only in the actual choice of variable that monitors which surface atoms need to be removed, by construction the new proposed method selects the removal of exactly the same atoms as the VTS-CCA implementation.

4.2. Self-balancing Binary Search Trees

The high computational cost of the traditional CCA implementations is due to the need of visiting all the surface sites in each time step in order to reduce their occupation, even if only a few atoms are typically removed. Although all the surface atoms are labeled with the corresponding PRT value, the philosophy behind the new implementation is the complete suppression of the need to perform parameter updates on a regular basis. In fact, no updates are needed for the predicted removal time of any atom during a sequence of time steps, unless the atom neighborhood has changed. This should reduce significantly the computational effort. However, this objective is only feasible if an ordered surface atom list with ascending PRT values can be maintained so that the next atom to be removed will always be the first one in the list –instead of searching for it among all the surface atoms in each time step.

The problem at this point is to find the most efficient data structure that can handle a dynamically modified ordered list. Traditional arrays have $\mathcal{O}(1)$ access cost but removing or adding one element has $\mathcal{O}(N)$ cost. Linked lists have $\mathcal{O}(1)$ addition/removal costs but $\mathcal{O}(N)$ access times. As a more efficient alternative, we propose the use of the so-called *Self-Balancing Binary Search Tree*, which has a cost of $\mathcal{O}(\log(N))$ for both access and insertion/deletion of elements.

The $\mathcal{O}(N)$ cost to insert/delete data elements to/from a traditional one dimensional array $A(1, 2, 3, \dots, N, \dots)$ can be easily understood as follows. In order to remove element $A(i)$ from the array, all the elements $A(j)$ with $j > i$ have to be "shifted one position down", *i.e.* $\forall j > i, A(j) \rightarrow A(j - 1)$. In the worst case, if $i = 1$, one has to read/write $N - 1$ values, *i.e.* $\mathcal{O}(N)$ operations. Similarly, in order to add a new element at position i all the elements $A(j)$ with $j > i$ have to be "shifted one position up", *i.e.* $\forall j \geq i, A(j) \rightarrow A(j + 1)$. Again, if $i = 1$ one has to write N values, *i.e.* cost $\mathcal{O}(N)$. The larger the number of surface atoms N , the longer it takes to insert/delete elements from the array, where *longer* means "proportional to N ". On the contrary, accessing one element (*e.g.* reading from or writing to $A(i)$) does not depend on N , *i.e.* the operation has a constant cost $\mathcal{O}(1)$.

A linked list is a data structure where each data element is stored in one node. An example is shown in Figure 3(a), where each node stores the surface site occupation value for one surface atom. Although each node is connected to the left and right nodes, there is no notion of the global position of each node with respect to left-most and/or right-most nodes. In other words, there is a notion of locality but there is no notion of the absolute position within the chain of nodes. Thus, inserting and deleting one node is a simple task that only requires breaking the local links to the left and right nodes, and creating new links between the affected new/old nodes. This task is independent of the number of surface atoms N , *i.e.* the operation is $\mathcal{O}(1)$. However, accessing one particular element of the list is a more time consuming task. As an example, this can be done by starting at the left-most node of the chain and checking one by one the content of every node as one moves towards the right-most node. On average one has

to read $(N + 1)/2$ elements before arriving at the desired element. Thus, reading from and writing on specific elements has a cost of $\mathcal{O}(N)$.

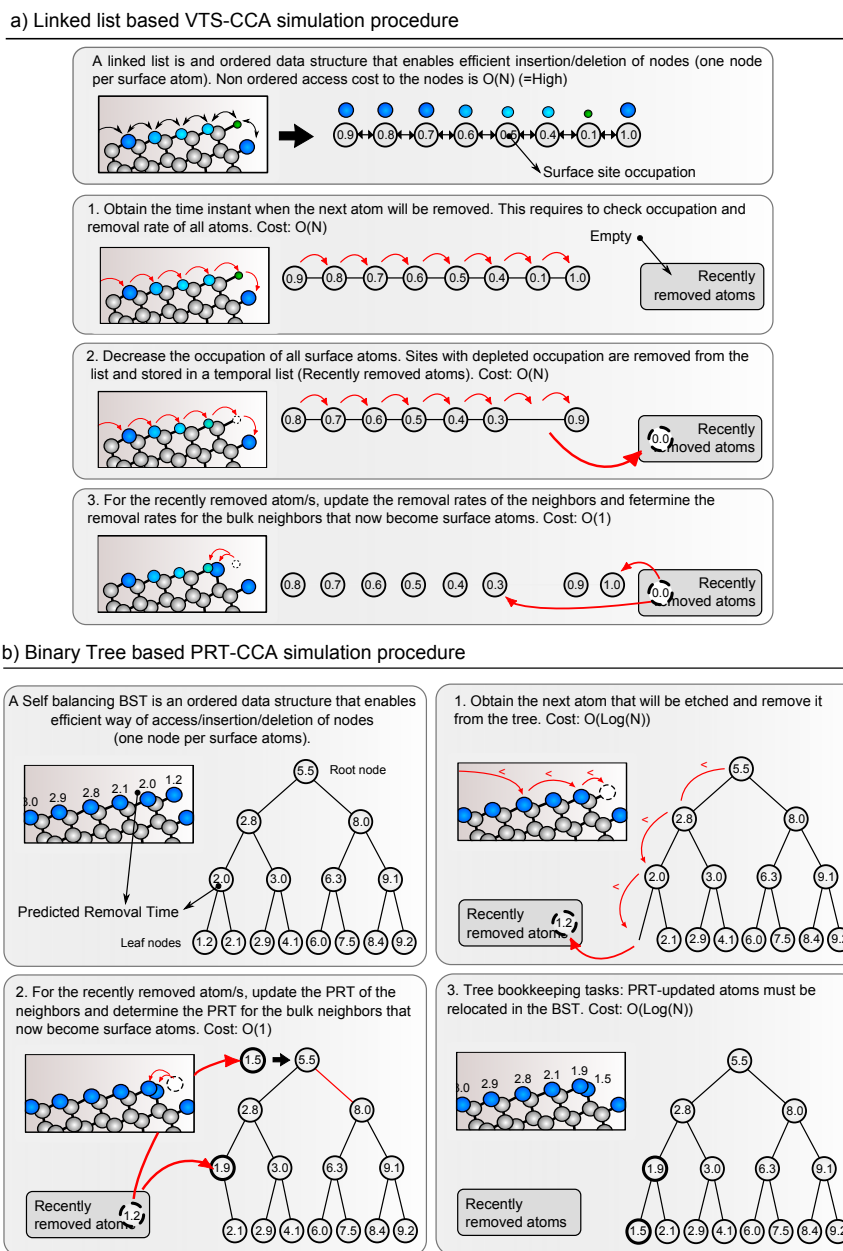


Figure 3. Procedure to perform one time step in a CCA simulation. (a) Linked list based VTS-CCA. (b) Self-Balanced Binary Search Tree based PRT-CCA.

As for linked lists, Binary Trees (BTs) are also data structures where each data element is stored in one node. However, each node of a BT can have up to two children nodes, forming a structure that resembles the shape of a tree, where any node can be accessed by following links from the first node or *root node* (see Figure 3(b)). The last nodes, which do not have any children, are referred to as *leaf nodes*. Binary Search Trees (BSTs) are BTs where the left child has always a smaller stored value than the parent

and the right child has a larger value. As an example, the BT shown in Figure 3(b) is actually a BST, where each node stores the PRT value for one surface atom. The most important advantage of the BST structure is that accessing one element has a cost of $\mathcal{O}(\lceil \text{Log}_2(N) \rceil)$ [18], where $\lceil x \rceil$ is the nearest integer of x towards infinity. The reason for this reduced cost is that traversing the structure in a node-by-node manner (as for linked lists, in order to find a particular stored value) is performed vertically from the root node towards the leaf nodes and, at every level the searched value is either smaller (left subtree) or larger (right subtree) than for the current node. Therefore, the number of visited nodes is equal to the number of levels of the tree (also referred to as the tree depth), which is $\lceil \text{Log}_2(N) \rceil$.

As for the case of a linked list, the cost of inserting/deleting one node in a BST is also independent of N , *i.e.* $\mathcal{O}(1)$. In this case, all that is needed is to simply break the local links to the parent and left/right children, and to create new links between the affected new/old parent/children nodes. While deleting one node of a BST may reduce the tree depth by one unit, inserting one node may increase the depth by the same amount. In fact, the number of levels may become rather large as a result of inserting new nodes, thus leading to time consuming vertical traversals. Furthermore, the BST structure can degenerate to that of a linked list if many unfortunate insertions increase the tree depth along one and the same branch repeatedly, thus deteriorating the access cost to $\mathcal{O}(N)$. Effectively, such BSTs look unbalanced, with one subtree very shallow (few levels) and another subtree very deep (many levels). In order to keep a good access performance, the shape of the tree should be kept as balanced as possible or, in other words, the tree depth should be minimized.

A *Self-Balancing Binary Search Tree* (SB-BST), introduced by Adelson-Velskii et al [19], adds several procedures for use after node insertion and deletion in order to ensure that the tree remains balanced and the tree depth remains as small as possible. The price to minimize the tree depth and thus the access times is the need to perform a number of tree-bookkeeping tasks. The SB-BST data structures ensure logarithmic costs for data access, insertion and deletion, which are very attractive properties for large data collections, for which the extra overhead in the insertion and deletion operations is substantially compensated by the minimized access times.

Figure 3 shows a comparison of the simulation procedure to perform one time step in (i) a traditional VTS-CCA implementation based on the use of a linked list and (ii) the proposed PRT-CCA implementation based on the use of a SB-BST. By showing the associated computational costs for each operation, the figure graphically explains the computational advantage of PRT-CCA over VTS-CCA.

4.3. Computational cost of the PRT-CCA method

The two tasks that consume most processing resources in a PRT-CCA simulation are steps "1." and "3." in Figure 3(b), *i.e.* the tree vertical traversal (TVT) in order to find the next atom to be removed (T_{tv}) and tree bookkeeping (TBK) tasks that

ensure a minimal tree depth (T_{tbk}). Since the computational cost for both tasks is $\mathcal{O}(\log_2 N)$ per removed atom and the number of removed atoms is $N_{removed}^{total} = g^3 N^{3/2}$, the computational time to perform a PRT-CCA simulation can be written as:

$$T_{PRT} \approx [(T_{tvt} + T_{tbk}) \cdot \log_2 N] g^3 \cdot N^{3/2} \approx \mathcal{O}(N^{3/2} \log N), \quad (22)$$

$$T_{PRT}^{(RA)} \approx (T_{tvt} + T_{tbk}) \cdot \log_2 N. \quad (23)$$

Thus, the computational cost of PRT-CCA is $\mathcal{O}(N^{3/2} \log N)$. Without reducing the simulation accuracy, PRT-CCA provides a dramatic cost reduction over VTS-CCA, whose cost is $\mathcal{O}(N^{5/2})$. In comparison to CTS-CCA, the computational time for PRT-CCA is independent of (i) the etchant reactivity (*i.e.* it does not depend on the average removal rate r) and (ii) the time step size (since Δt is not a parameter in the PRT method). Since the cost of CTS-CCA is $\mathcal{O}(\frac{N^{3/2}}{r\Delta t})$, the actual relative efficiency of the two methods depends on the particular details of the simulated system. In other words, either method can perform better than the other, depending on the etching conditions, as shown in Section 5. However, PRT-CCA is always guaranteed to produce the correct results while the accuracy of CTS-CCA will depend on the size of the time step.

5. Results

In order to compare the traditional implementations and the newly proposed variant, a Java based wet etching simulator that includes the CTS-CCA, VTS-CCA and PRT-CCA versions has been implemented. While the two traditional forms store the surface atoms as nodes of a linked list, PRT-CCA is implemented using a Red-Black Self-Balanced Binary Search Tree [20].

5.1. Simulated system

The simulated system consists on a square chip from a Si{100} wafer masked using a centered square pattern aligned with the $\langle 110 \rangle$ direction and covering 45% of the surface, with a thin mask frame that also covers the perimeter of the silicon chip, as shown in Figure 4(a). The simulations are stopped when the etched depth reaches 1/15 of the chip width. The obtained structure is characterized by intense underetching at the four convex corners of the square mask and the formation of $\langle 111 \rangle$ side walls, as shown in Figure 4(b). The simulations are performed for four different etchant temperatures, since this allows the analysis of the relative efficiency of the CTS-CCA method as a function of the average rate of the system r (as higher temperatures mean larger r). The experimental data to calibrate the CCA method at the four temperatures is obtained from Ref. [17]. At each temperature, four different surface sizes are used, in such a manner that each size doubles the previous size ($N_{i+1} = e \cdot N_i$, where $e = 2$). Although the aspect ratio of the resulting three-dimensional shape is kept constant (etched depth = 1/15 of the chip width), the absolute etched depth increases by a factor of $e^{1/2} = \sqrt{2}$ per surface doubling. Similarly, the etching time increases by the same factor ($e^{1/2} = \sqrt{2}$) per surface doubling. This results in a total increase of the

volume of the etched structures by a factor of $e^{3/2} = 2^{3/2} = 2.83$ per surface doubling. The computer used to perform the simulations is equipped with an Intel core i7 processor at 2.66GHz with 3GB of DDR3 PC1066 RAM memory.

5.2. Figures of merit

The table shown in Figure 4 gathers the most relevant data for all the performed simulations. As briefly described in the legend on the upper-right corner, the results of each simulated system are characterized by a total of eight figures of merit: (1) t , the etching time, in ms; (2) K , the total number of time steps; (3) T , the total computational time, in ms; (4) $N_{removed}^{total}$, the total number of removed atoms; (5) $T^{(RA)} = T/N_{removed}^{total}$, the computational time per removed atom, in μs ; and (6) $q = N_{visited}^{total}/N_{removed}^{total}$, the ratio of the total number of visited atoms to the total number of removed atoms (*i.e.* the inefficiency); (7) $T^{(VA)} = T/N_{visited}^{total}$, the computational time per visited atom, in μs ; and (8) $p = T^{(VA)}/T_{92 \times 92}^{(VA)}$, the performance penalty ratio associated to an increase of the system size.

The definition of p is due to the fact that in practice the average access time per node of a linked list (and in a SB-BST) is not exactly constant (as theoretically expected) but slightly increases with the system size N , as shown in Figure 5. This curve behaves in a non-trivial manner and, in fact, the details will change with the actual computer architecture.

The increase in the access time is due to the fact that the memory access performance in a computer processor varies greatly with the memory access pattern. This effect is due to the available hierarchy of cache memories that the processor has in order to accelerate most frequently accessed data [22]. As the linked list size N increases, it is stored in the slower memories, thus giving rise to small jumps in the access time. An effective manner to eliminate the cumulative contribution of these access times in our measurements of T , is to define the *corrected computational time* as $T_c = T/p$ and the corresponding *corrected computational time per removed atom* as $T_c^{(RA)} = T^{(RA)}/p$. Note that p indicates the access time per atom as a multiple of the access time for the smallest system at each temperature (92×92) and for each of the three computational methods.

The way $N_{visited}^{total}$ is determined in the simulations is important as it affects the value of the inefficiency q . For CTS-CCA, $N_{visited}^{total}$ is taken as the sum of all the atom visits required to (i) reduce the occupations and (ii) update the neighbors. For VTS-CCA, $N_{visited}^{total}$ includes the visits required to (i) reduce the occupations, (ii) update the neighbors and (iii) calculate the next time step. For PRT-CCA, $N_{visited}^{total}$ contains the visits required to (i) vertically traverse the BST in order to find PRT_{min} , (ii) update the neighbors, and (iii) insert/delete/relocate nodes in the BST.

From a purely theoretical perspective, the contribution (ii) can be considered to be identical in all three methods. The number of update visits depends only on the physical etching model and the underlying geometry of the silicon crystal, and thus

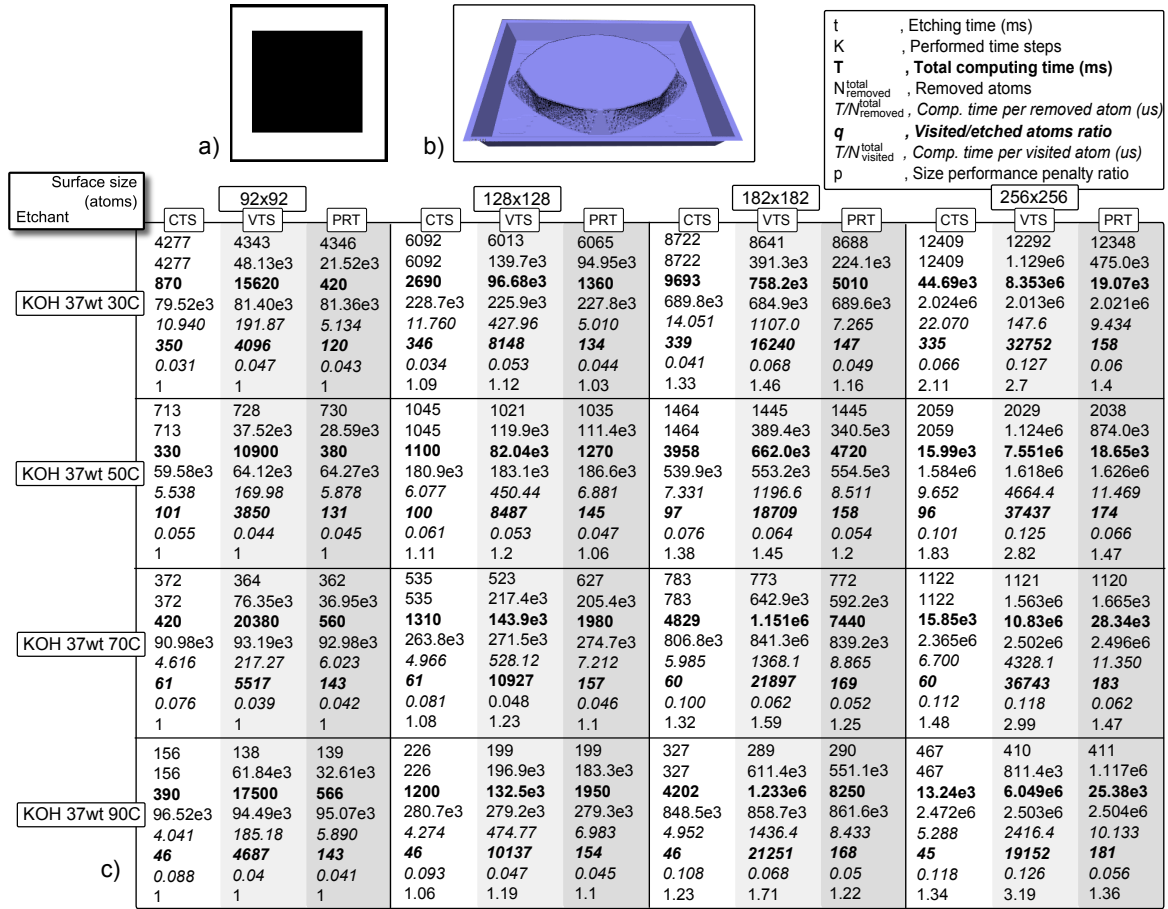


Figure 4. Performance comparison between CTS-CCA, VTS-CCA and PRT-CCA: (a) Mask design used for the simulations. (b) Typical final state for the simulations, showing large underetching at the convex corners of the mask design. (c) Resulting values for the five chosen figures of merit, as indicated in the legend (upper-right box).

does not depend on the particular implementation details of any of the three methods. If the time evolution of the three methods is identical, the number of update visits will also be identical. If the time evolution differs moderately, a slightly different average number of update visits will be effectively added to $N_{visited}^{total}$ for each of the methods. This will increase the value of q but it will not introduce any significant deviations between the q values of the three methods. Although the contribution (ii) can be completely eliminated from $N_{visited}^{total}$, there are reasons to retain it, as described in the next paragraph.

The tree traversal visits considered in contribution (i) for $N_{visited}^{total}$ in the PRT-CCA implementation are theoretically non-essential: a side-effect from choosing a binary tree as the data structure for the actual implementation of the method in practice. Thus, they can be completely eliminated from $N_{visited}^{total}$. Similarly, we can eliminate the count of the atom visits associated to the tree bookkeeping tasks, as they also can be considered theoretically a side-effect from the use of the binary tree data structure. However, if we

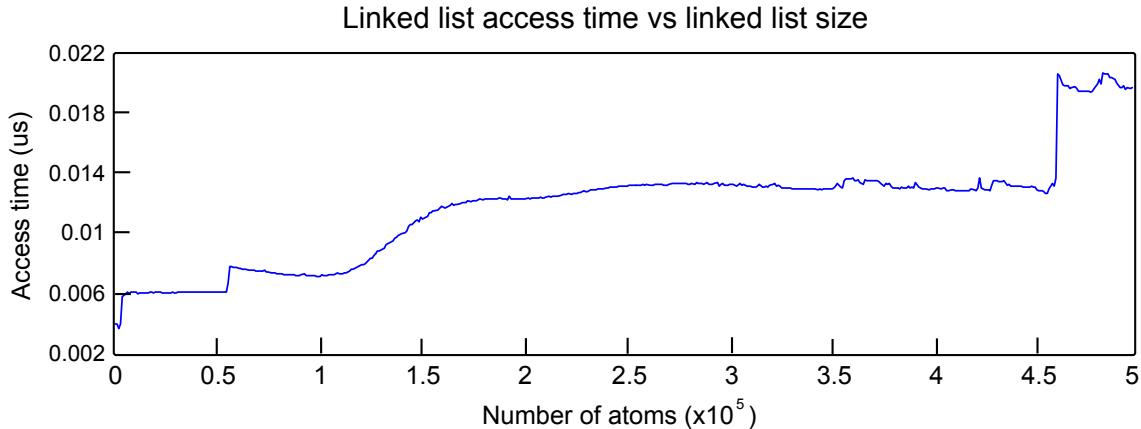


Figure 5. Average access time per node of a linked list as a function of the linked list size.

disregard all contributions (i) through (iii) for PRT-CCA, we arrive to the result that every chosen atom in the PRT-CCA implementation is always removed, thus leading to $q = 1$. While this is theoretically correct, a comparison between the q values for the traditional implementations and $q = 1$ for the newly proposed method becomes pointless. For this reason, we choose to retain all the enumerated contributions to $N_{visited}^{total}$ described at the beginning of this section. This way the reported values of q provide a fair comparison of the number of visit-related operations performed in the three implementations in order to remove one surface atom.

5.3. Computational efficiency of the three methods

For CTS-CCA (white columns in the table of Fig. 4), the efficiency (signaled by small values of q) depends strongly of the applied etchant reactivity r . The simulations at the lowest temperature (smallest r) require visiting hundreds of atoms (≈ 350) in order to remove one atom while almost ten times less visits (≈ 45) are needed at the highest temperature (largest r). This is simply due to the fact that larger r leads on average to larger occupation reductions (see Eq. 1) and thus fewer time steps K are needed to remove the atoms. As demonstrated by the table, the dependence of q on r is completely independent from the surface size (q remains essentially constant for all surface sizes).

The previous behavior of q for CTS-CCA is confirmed by the behavior of the computational time per removed atom $T^{(RA)} = T/N_{removed}^{total}$, which is plotted in Figure 6 using full lines. In agreement with Equation 16, Fig. 6(a) shows that $T^{(RA)}$ increases with decreasing r (= decreasing temperature), for the same reason as explained for q . $T^{(RA)}$ is observed to increase with the system size in Fig. 6(b), thus deviating from the expected constant behavior regarding N according to Equation 16. This is simply due to the increase in the average access time per atom for the linked list data structure in practice, as explained in relation to Figure 5. When the corrected computational time per removed atom $T_c^{(RA)} = T^{(RA)}/p$ is plotted in Figure 6(c), we observe that in

deed $T_c^{(RA)}$ for CTS-CCA is constant (slope = -0.015), thus in perfect agreement with Equation 16.

For VTS-CCA (light gray columns in the table of Fig. 4), the most salient feature is the huge amount of time steps K required to finish the simulations, in comparison to CTS-CCA. This feature is reflected as an extremely large inefficiency q , typically one or two orders of magnitude larger than the inefficiency of CTS-CCA for the same system. The larger the surfaces, the larger the q values. As expected from Equation 18, the dotted lines for VTS-CCA in Figure 6(a) show that the computational cost per atom is independent of the etchant reactivity ($T^{(RA)}$ remains essentially constant as a function of the temperature). The figure confirms the huge difference in computational times between VTS-CCA and the other two methods, typically two orders of magnitude. As expected, Figure 6(b) shows that $T^{(RA)}$ increases with N . In deed, when the corrected computational time per removed atom $T_c^{(RA)} = T^{(RA)}/p$ is plotted in Figure 6(c), we observe that $T_c^{(RA)}$ for VTS-CCA increases linearly with N (slope = 0.922 \approx 1.0), thus in good agreement with Equation 18.

The number of removed atoms per time step in the VTS-CCA method (obtained by dividing the values of $N_{removed}^{total}$ by the values of K) lies typically between 1.5 and 3 for all etchant temperatures and system sizes. This underlines the fact that highly reactive atoms are constantly formed and quickly removed in these simulations and, as a result, only a few atoms can be removed per time step by using the VTS-CCA method.

For PRT-CCA (dark gray columns in the table of Fig. 4), q is much lower than for VTS-CCA, typically by one or two orders of magnitude. This means that PRT-CCA is typically two orders of magnitude more efficient than VTS-CCA, without affecting the accuracy, as described in Section 5.4. This is shown in Figure 6(a), where the broken lines for the PRT-CCA curves lie much lower than the dotted lines for VTS-CCA. In comparison to CTS-CCA, Figure 6(a) shows that PRT-CCA is more efficient for the less reactive etchants (low temperature), mostly due to the loss in efficiency experienced by CTS-CCA when r is decreased. Although CTS-CCA is more efficient than PRT-CCA for the more reactive etchants (high temperature), it also makes larger errors (see Section 5.4).

As shown by Figure 6(b) and the table in Figure 4, the $T^{(RA)}$ and q values for the PRT-CCA method experience a moderate increase with the surface size, mostly due to the fact that the number of tree vertical traversals and tree bookkeeping tasks increases slightly with the system size, as described by Equation 23 (log N dependence). In fact, when the corrected computational time per removed atom $T_c^{(RA)} = T^{(RA)}/p$ is plotted for PRT-CCA in Figure 6(d) we obtain that $T_c^{(RA)}/\log(N)$ remains constant with N (slope = 0.025), thus meaning that the cost per atom follows the log N dependence in deed, in excellent agreement with Equation 23. For a fixed size, the $T^{(RA)}$ and q values for PRT-CCA experience a faint increase with the temperature, indicating that slightly more neighbor updates and related tree bookkeeping tasks need to be performed when the average removal rate r is increased.

The computational cost of the three methods is summarized in Figures 6(e)-(f).

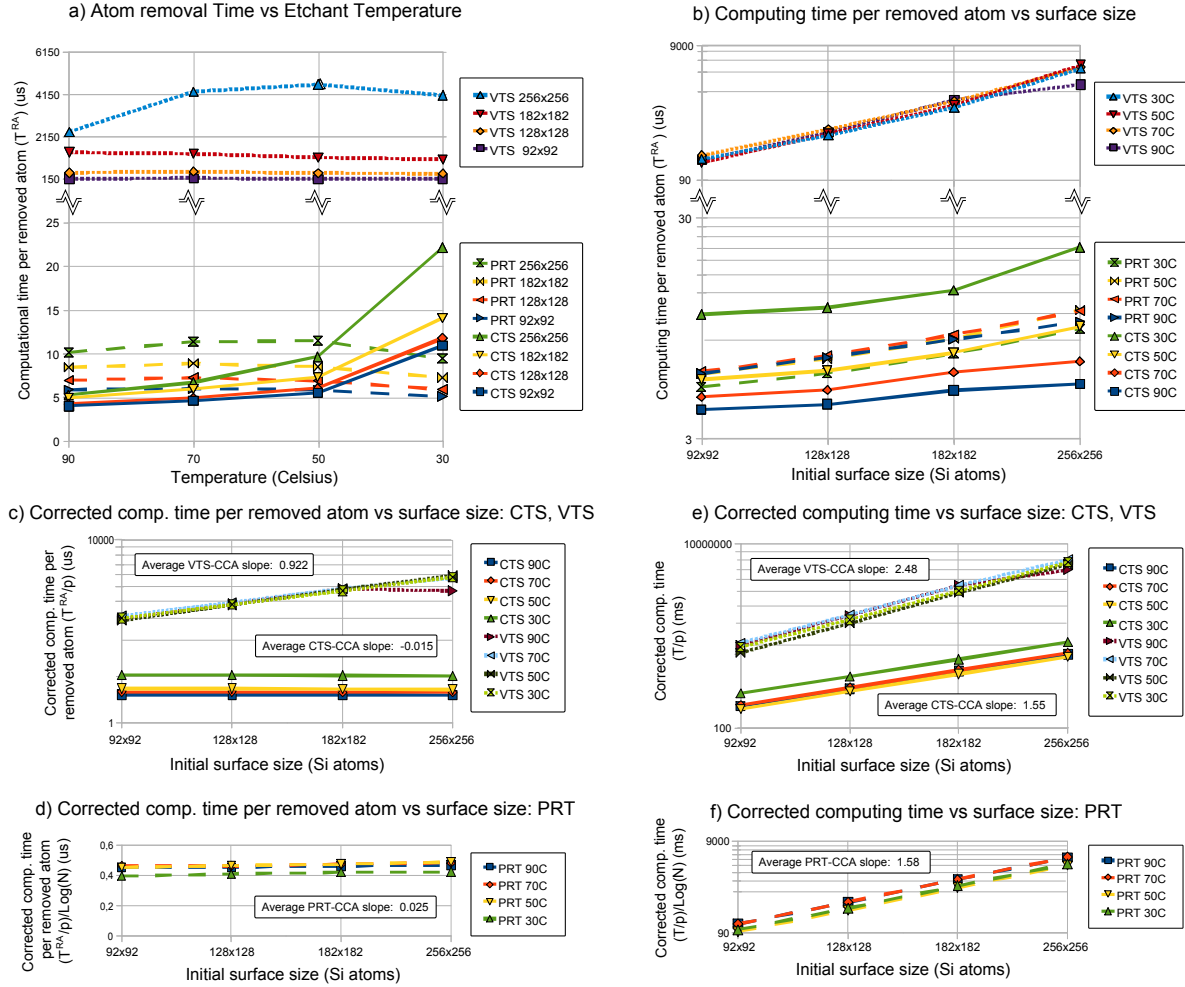


Figure 6. Up: Computational time per removed surface atom ($T^{(RA)}$) as a function of (a) the etchant temperature and (b) the size of the systems, for the CTS-CCA, VTS-CCA and PRT-CCA methods. Down left: Corrected computational time as function of the size of the system for (c) CTS-CCA, VTS-CCA and (d) PRT-CCA. Down right: Corrected computational time per removed atom surface as a function of the size of the system for (e) CTS-CCA, VTS-CCA and (f) PRT-CCA. Each point corresponds to one of the 16 systems of Fig. 4.

Figure 6(e) shows that the corrected total computational time $T_c = T/p$ grows as $\mathcal{O}(N^{2.48}) \approx \mathcal{O}(N^{5/2})$ for VTS-CCA and $\mathcal{O}(N^{1.55}) \approx \mathcal{O}(N^{3/2})$ for CTS-CCA, in excellent agreement with Equations 17 and 15, respectively. Similarly, Figure 6(f) shows that T_c grows as $\mathcal{O}(N^{1.58} \log N) \approx \mathcal{O}(N^{3/2} \log N)$ for PRT-CCA, also in excellent agreement with Equation 22. These results demonstrate that the total number of removed atoms $N_{removed}^{total}$ is proportional to $N^{3/2}$, as described in Section 3. Physically, this results from the fact that the variable N is used to describe the size of a 2D system (the silicon surface) while $N_{removed}^{total}$ describes the size of a 3D volume. In this manner, $N^{1/2}$ represents a characteristic length of the 2D silicon surface and $N^{3/2}$ the characteristic volume of the silicon sample. Effectively, the number of removed atoms is proportional

to the number of atoms in the sample, not only the atoms on the surface.

5.4. Relative error with respect to VTS-CCA

The final value of the total simulated etching time t is a very important parameter in order to reliably simulate wet etching for MEMS applications. Very often the user will define the desired target etch depth and the simulator will be responsible of calculating the etching time correctly and propagating the system to the correct final state. Errors in t are translated directly into errors in the final geometry of the structure.

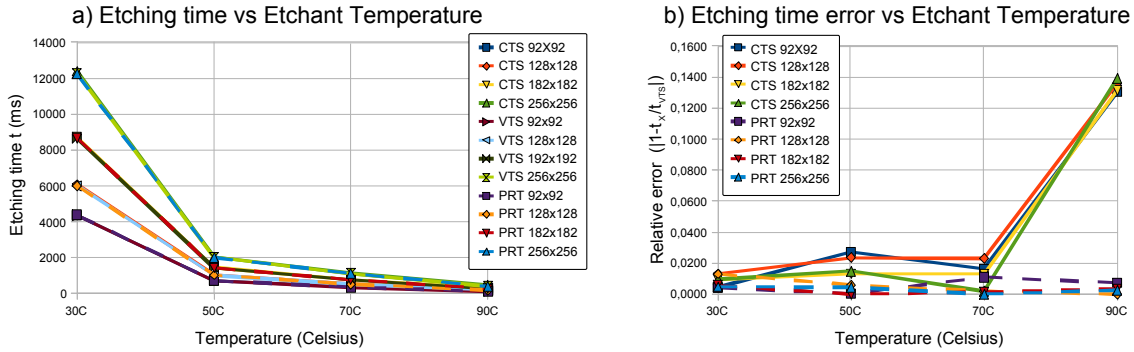


Figure 7. (a) Total simulated etching time t as a function of the etchant temperature for the CTS-CCA, VTS-CCA and PRT-CCA methods applied to the systems of Fig. 4. (b) Relative error in t with respect to the VTS-CCA value: $Error = |1 - t_X/t_{VTS}|$, where $X = CTS$ and PRT .

Figure 7(a) shows the total simulated etching time t to reach a depth of $1/15$ of the chip width as a function of the etchant temperature for the CTS-CCA, VTS-CCA and PRT-CCA methods applied to the systems of Fig. 4. The reduction in t with temperature for the three methods agrees completely with the fact that the etch rate of the bottom floor surface $\{100\}$ depends exponentially on the inverse temperature (Arrhenius law [1]). The increase in t with the simulated system size is also expected, occurring according to the $\sqrt{2}$ factor per surface doubling, as explained in Section 5.1. At first glance, Fig. 7(a) suggests that the CTS-CCA and PRT-CCA methods provide very similar values to the VTS-CCA implementation, which is taken as the reference system. However, the relative error in t with respect to the VTS-CCA value, as shown in Figure 7(b), demonstrates that the error for CTS-CCA increases drastically as the simulated etchant becomes more reactive, reaching a maximum of 14% with respect to VTS-CCA. This error is due to the fact that larger removal rates r for the atoms result in larger occupation reductions (see Eq. 1) and thus more removed atoms per time step and larger errors. In principle, the error can be reduced by reducing the time step size Δt of the CTS-CCA method. However, this will lead to larger computational times (Equation 16).

In the case of PRT-CCA, Figure 7(b) shows that the t values stand very close to VTS-CCA for all temperatures and system sizes. Since PRT-CCA and VTS-CCA

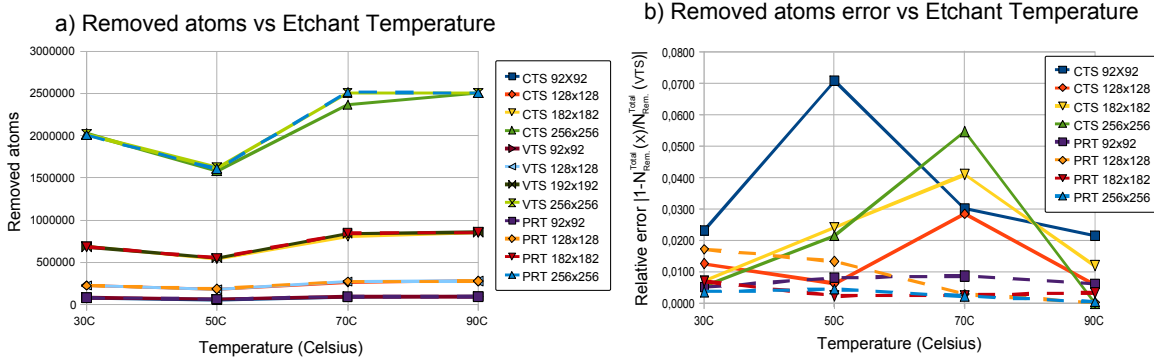


Figure 8. Total number of removed atoms $N_{removed}^{total}$ as a function of the etchant temperature for the CTS-CCA, VTS-CCA and PRT-CCA methods applied to the systems of Fig. 4. (b) Relative error in $N_{removed}^{total}$ with respect to the VTS-CCA value: $Error = |1 - N_{removed}^{total}(X)/N_{removed}^{total}(VTS)|$, where $X = CTS \text{ and } PRT$.

are theoretically completely equivalent methods, the small error (typically less than 1%) has a numerical origin. In both implementations we use single precision floating point variables to store the occupation and PRT values in the computer memory. The precision of this format is relative to the stored absolute value. While the occupation-based simulator is able to store values between 0.0 and 1.0, the PRT-based simulator is required to store increasingly larger values of t and PRT, thus leading to a loss of precision in the decimal part of the stored values. Under these circumstances the differences between $PRT^{(k)}$ and $t^{(k+1)}$ can lead to slightly wrong values for $PRT^{(k+1)}$ when using Equation 21. We have checked that the use of double precision floating point variables for PRT and t leads to completely identical simulation results for the VTS-CCA and PRT-CCA methods. Despite the numerical errors associated to single precision storage, Figure 7(b) shows that the PRT-CCA results are much closer to VTS-CCA than the results from CTS-CCA.

A way to assess with a single number the similarity between the resulting geometries of the simulated systems is to consider the total number of removed atoms $N_{removed}^{total}$. Although the etched depth along the $\langle 100 \rangle$ direction is similar in all the simulations for the same system size, differences in $N_{removed}^{total}$ indicate differences in the etch rates along other directions and thus deviations in the final geometry. Figure 8(a) shows $N_{removed}^{total}$ as a function of the etchant temperature for the CTS-CCA, VTS-CCA and PRT-CCA methods, corresponding to the same systems as in figures 4 and 6. As before, Figure 8(a) suggests that the three simulation methods produce very similar results. However, Figure 8(b) shows that the relative error in $N_{removed}^{total}$ with respect to the VTS-CCA value is larger for CTS-CCA (up to 7%) while it remains low for PRT-CCA (below 2%). The origin of these errors is the same as discussed for Fig. 6, namely, the size of the time step for CTS-CCA (intrinsic error) and single precision data storage for PRT-CCA (numerical error).

In conclusion, the proposed PRT-CCA implementation provides affordable

simulation times –even faster than CTS-CCA for low reactivity systems– while producing the correct simulated geometries within numerical precision errors.

6. Practical Application

In this section we analyze the resulting differences between the CTS-CCA, VTS-CCA and PRT-CCA methods in a real-life, practical application. The chosen MEMS structure is a dual-axis micromechanical probe characterized by having a torsion beam for vertical force detection and a double cantilever beam for horizontal force detection [23]. We choose this microstructure based on its novelty, prevalent use of wet etching and miniature torsion and cantilever beam thicknesses, which underline the need to accurately control the relative etch rates of different surface orientations as well as the overall etching time in the simulations.

For the fabrication of this structure the CCA model is calibrated based on the experimental etch rate data for KOH 40wt at 70C by Sato *et al.* [24]. Since most facets of the etch front correspond to $\langle 100 \rangle$ and $\langle 111 \rangle$ in this system, the errors introduced in the CTS-CCA implementation are expected to appear as an incorrect etch rate for the $\langle 100 \rangle$ facets. Since the CCA calibration explicitly matches the $\langle 100 \rangle$ experimental etch rate, the exact model implementations should simulate the etch rate of this plane without errors.

The simulated system consists on a $5300 \times 3003 \mu\text{m}^2$ silicon chip with a thickness of $180 \mu\text{m}$ obtained from a Si{100} wafer with a $\langle 100 \rangle$ flat. The 2-axis probe can be considered as a high-aspect ratio microstructure since the resulting cantilever beam is only a few microns thick while the cantilever length is several hundreds. In addition, the simulations must achieve enough precision in the thinnest parts of the structure and, thus, the model resolution must be increased as much as possible. The initial surface has a total of $1.59 \cdot 10^6$ atoms and the final microstructure is formed by $766 \cdot 10^3$ surface atoms.

The fabrication procedure used in our CCA simulations is described in Figure 9 and briefed as follows:

- (i) Apply silicon nitride masks 1 and 2 (Figure 9(a)) on the top and bottom sides of the silicon sample, respectively (Figure 9(b)).
- (ii) Wet etch for 45 minutes and remove the masks (Figure 9(c)).
- (iii) Apply a mask of silicon nitride with the shape of mask 3 and mask 1 on the top and bottom sides of the silicon ship, respectively (Figure 9(d)).
- (iv) Etch for another 250 minutes and remove the masks (Figure 9(e)).

The etch time in step (iv) must be accurately chosen as the etching from the top and bottom sides results in the perforation of the chip, leading to the release of the torsion beam and the cantilever beams. Etching beyond this point is highly undesirable as it will easily result in the complete removal of the torsion and cantilever beams. Taking

2-axis probe fabrication steps.

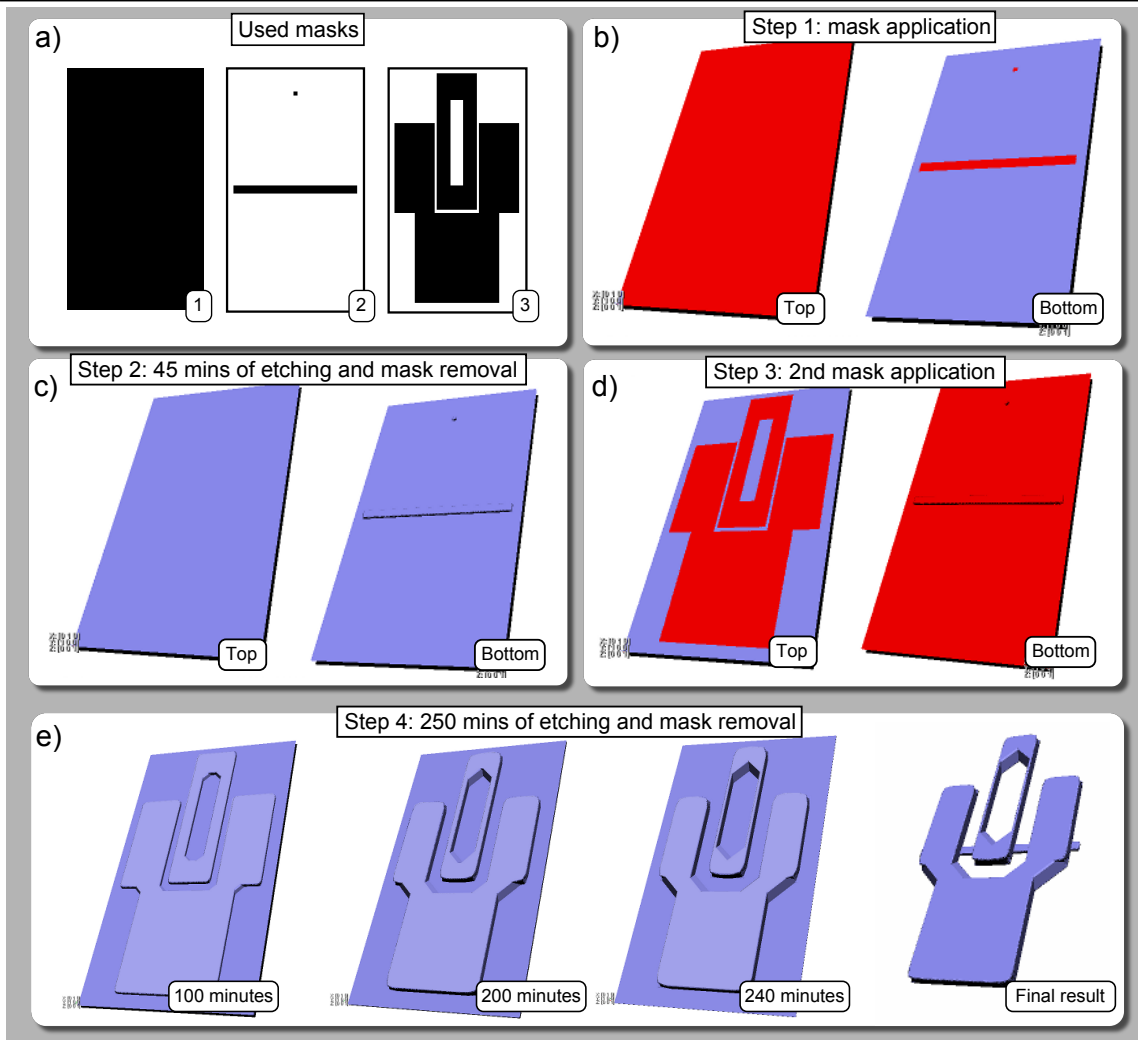


Figure 9. Two-axis probe fabrication procedure: (a) used masks, (b)-(e) proposed fabrication steps.

this into account, the user must adjust the dimensions of the elongated, inner hole of mask 3 in order to control the final cantilever beam thickness correctly.

Figure 10 shows the simulation results using VTS-CCA, PRT-CCA and CTS-CCA. For CTS-CCA, the silicon substrate has not been perforated (a remaining $4.73\mu\text{m}$ thick layer is still present). In addition, the width of the cantilever beams is substantially larger than for VTS-CCA. In order to get similar results with CTS-CCA, 10 extra minutes of etching time must be applied. If this extra time is used in the VTS-CCA method the cantilever beams will be completely etched away, making the structure unusable. The results obtained with PRT-CCA are much closer to VTS-CCA, with small differences in the torsion and cantilever beam thicknesses due to numerical precision errors, as discussed in Section 5.4.

In conclusion, the use of CTS-CCA for the simulation of wet etching on demanding

2-axis probe simulation results. VTS-CCA vs PRT-CCA vs CTS-CCA, 45+250 mins of etching

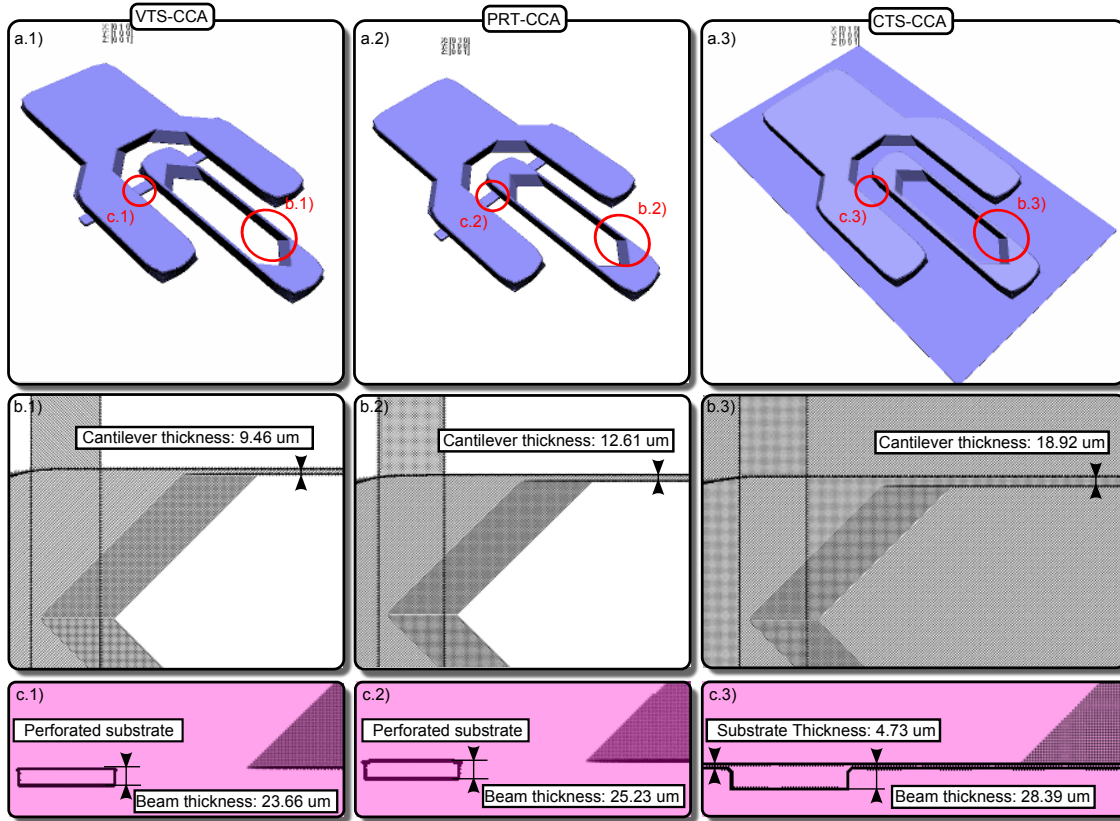


Figure 10. Two-axis probe simulation results for the three considered methods: (a) 3D view, (b) close view of the cantilever beam from the top, (c) cross sectional view of the torsion beam.

real-life scenarios can lead to moderately wrong fabrication parameters for the mask dimensions and/or etching times. The proposed PRT-CCA implementation produces more accurate results while keeping a reduced computing complexity in comparison to VTS-CCA.

7. Conclusions

This study focuses on the importance of a more accurate model implementation for anisotropic wet etching simulations based on the Continuous Cellular Automaton (CCA). Although the traditional, approximate approach based on constant time stepping (CTS) leads to much faster simulations than the exact (also traditional) variable time stepping (VTS) implementation, relevant differences are observed for the simulated etch rates of some surface orientations as well as for the overall simulated times and etched geometries. To correct this situation, the present study proposes an efficient implementation of the VTS approach, based on (i) the introduction of a new variable, referred to as the Predicted Removal Time (PRT), which substitutes the

traditional concept of a gradually decreasing surface site occupation, and (ii) the use of a self-balanced binary search tree (SB-BST) in order to store the PRTs of the surface atoms in the computer memory and efficiently access the minimum PRT value in each time step so as to quickly remove the corresponding atom or group of atoms.

We demonstrate that the total number of removed atoms $N_{removed}^{total}$ (which represents the total volume of removed silicon material) is proportional to $N^{3/2}$ in any CCA implementation, where N is the system size. This is due to the fact that $N^{1/2}$ represents a characteristic length of the 2D silicon surface and, thus, $N^{3/2}$ represents the characteristic volume of the silicon sample. The proposed PRT approach removes the traditional need of the CTS and VTS implementations to visit all the surface atoms in each time step and allows to reduce the simulation cost of the exact implementation from $\mathcal{O}(N^{3/2+1})$ to $\mathcal{O}(N^{3/2} \log N)$ without any model simplifications. Compared to the CTS implementation, whose computational complexity is $\mathcal{O}(\frac{N^{3/2}}{r\Delta t})$, the new approach provides more precise simulations (within numerical precision errors) at a similar computational cost. The PRT implementation becomes even more efficient than CTS for the simulation of etchants whose average reactivity r is low. As depicted by the chosen example application, the new version of the CCA method enables the accurate control of the etch rates of the different surface orientations as well as the overall etching time in real-life simulations for demanding MEMS applications.

Acknowledgements

This work has been supported by Programa de Becas de Excelencia de la Universidad Politécnica de Valencia (PAID-09-09), MEXT Grant in Aid Research (Kakenhi: Silicon etching (A) 19201026), the Global COE program of Japan (GCOE, Wakate JSPS Young Scientist Fund), and the Ramón y Cajal Fellowship Program by the Spanish Ministry of Science and Innovation. MAG is thankful to Dr. Y. Xing, Southeast University, Nanjing, PRC, for useful discussions and tests at the early stages of this study.

References

- [1] M. A. Gosálvez, I. Zubel, E. Viinikka, *Wet etching of Silicon, Ch. 24 in Part IV of Handbook of Silicon Based MEMS Materials and Technologies*, Ch. Ed. H. Seidel, Book Ed. V. K. Lindroos, M. Tilli, A. Lehto and T. Motooka, Elsevier, Micro and Nano Technologies series. 55 pages. (Jan. 2010).
- [2] Pal P, Gosálvez M A and Sato K 2010 *Silicon Micromachining Based on Surfactant-Added Tetramethyl Ammonium Hydroxide: Etching Mechanism and Advanced Applications* (Japanese Journal of Applied Physics 49 056702)
- [3] Zubel I and Kramkowska M 2010 *The effect of alcohol additives on etching characteristics in KOH solutions* (Sensors and Actuators A 101 pp255-261)
- [4] Gosálvez M A, Xing Y, Sato K and Nieminen R M 2009 *Discrete and continuous cellular automata for the simulation of propagating surfaces* (Sensors and Actuators A: Physical 155 1 pp 98-112)
- [5] Gosálvez M A, Xing Y and Sato K 2008 *Analytical Solution of the Continuous Cellular Automaton for Anisotropic Etching* (Journal of MicroelectroMechanical Systems 17 2)

- [6] Zhou Z, Huan Q, Li W and Deng W 2007 *A cellular automaton-based simulator for silicon anisotropic etching processes considering high index planes* (Journal Micromechanics Microengineering 17 (2007) S38S49)
- [7] Gosálvez M A, Sato K, Foster A S, Nieminen R M and Tanaka H 2007 *An atomistic introduction to anisotropic etching* (Journal Micromechanics Microengineering 17 (2007) S1S26)
- [8] Frühauf J, Trautmann K, Wittig J and Zielke D 1993 *A simulation tool for orientation dependent etching* (Journal Micromechanics Microengineering 3 3 (1993) pp113-115)
- [9] Asaumi K, Iriye Y, Sato K 1997 *Anisotropic-etching process simulation system MICROCAD analyzing complete 3D etching profiles of single crystal silicon*, Proc. IEEE Int. Conf. on Micro Electro Mechanical Systems (MEMS 97), Nagoya, Japan, January 26–30, 1997, pp. 412417.
- [10] <http://www.intellisensesoftware.com/modules/Anise.html>
- [11] <http://www.intellisensesoftware.com/modules/IntelliEtch.html>
- [12] M. A. Gosálvez, Y. Xing, K. Sato, and R. M. Nieminen, Atomistic methods for the simulation of evolving surfaces, J. Micromech. Microeng. 18 (2008) 055029.
- [13] Zhu Z and Liu C 2000 *Micromachining Process Simulation Using a Continuous Cellular Automata Method* (Journal of Micromechanical Systems 9 2 pp 252-261)
- [14] Hubbard H and Antonsson E K 1997 *Cellular Automata Modeling in MEMS Design* (Sensors and Materials 9)
- [15] Xing Y, Gosálvez M A and K Sato 2007 *Step flow-based cellular automaton for the simulation of anisotropic etching of complex MEMS structures* (New Journal of Physics 9 (2007) 436)
- [16] Gosálvez M A, Foster A S and Nieminen R M 2002 *Atomistic simulations of surface coverage effects in anisotropic wet chemical etching of crystalline silicon* (Applied Surface Science 202 160-182)
- [17] Wind R A, Jones H, Little M J and Hines M A 2002 *Orientation-Resolved Chemical Kinetics: Using Microfabrication to Unravel the Complicated Chemistry of KOH/Si Etching* (Journal of Physical Chemistry B 106 7 pp 1557-1569)
- [18] Knuth D *The art of computer programming vol 1. Fundamental Algorithms*, (Addison-Wesley, 1997. ISBN 0-201-89683-4.)
- [19] Adelson-Velskii G M and Landis E M 1962 *An algorithm for the organization of information* (Soviet Math. Doklady, 3 pp 12591263)
- [20] bayer R 1972 *Symmetric binary B-Trees: Data structure and maintenance algorithms* (Acta Informatica 1 4 pp 290-306)
- [21] Unit cell size similar to the ones used in VisualTAPAS wet etch simulators: <http://tfy.tkk.fi/mag/VisualTAPAS/Home.html>
- [22] Hennessy J L, Patterson D A 2007 *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann publishers, ISBN 10: 0-12-370490-1, Chapter 5: Memory Hierarchy Design.
- [23] Fukuzawa K, Terada S, Shikida M, Amakawa H, Zhang H and Mitsuda Y 2007 *Mechanical design and force calibration of dual-axis micromechanical probe for friction force microscopy* 2007 (Journal of Applied Physics 101, 034308)
- [24] Sato K, Shikida M, Matsushima Y, Yamashiro T, Asaumi K, Iriye Y and Yamamoto M 1998 *Characterization of orientation-dependent etching properties of single-crystal silicon: effects of KOH concentration* (Sensors and Actuators A 64 pp 87-93)