FÁBIO BELOTTI COLOMBO

Aplicação de autômatos celulares para simulação de processos de microfabricação

São Paulo

2017

Aplicação de autômatos celulares para simulação de processos de microfabricação

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Doutor em Ciências.

Área de Concentração:

Engenharia Elétrica

Orientador: Prof. Dr. Marcelo Nelson Páez Carreño

São Paulo

2017

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 04 de janeiro de 2017

Assinatura do autor

Assinatura do orientador

Catalogação da Publicação

Colombo, Fábio Belotti

Aplicação de Autômatos Celulares para Simulação de Processos de Microfabricação / F. B. Colombo – versão corr.-- São Paulo, 2017.

169 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

Orientador: Marcelo Nelson Páez Carreño

1.AUTÔMATOS CELULARES 2.SIMULAÇÃO 3.MICROELETRÔNICA 4.SISTEMAS MICROELETROMECÂNICOS I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

DEDICATÓRIA

À minha esposa Simone.

AGRADECIMENTOS

À Simone, pela ajuda, apoio e amor.

Aos meus pais, pelo apoio ao longo dos meus estudos.

Ao Prof. Dr. Marcelo Nelson Paéz Carreño e ao Prof. Dr. Marco Alayo, pelos conselhos e colaboração ao longo desse tempo.

À CAPES, pelo apoio financeiro para a realização deste trabalho.

RESUMO

COLOMBO, F. B. **Aplicação De Autômatos Celulares para Simulação de Processos de Microfabricação.** 2017. 150 f. Tese (Doutorado). Escola Politécnica - Universidade de São Paulo, São Paulo, 2017.

Autômatos celulares e suas variações são atualmente utilizados para simulação de diversos processos físicos. De especial interesse para o campo de simulação de processos de microfabricação são os autômatos celulares para evolução de frentes circulares e elípticas e os autômatos celulares para corrosão anisotrópica de Si. No presente trabalho é apresentado um autômato celular alternativo para uso na simulação da evolução topográfica de uma superfície. Este autômato celular apresenta diversas vantagens em relação aos autômatos celulares reportados na literatura, como menor vizinhança e melhores resultados para grandes redes de células. Também foi avaliada a viabilidade do uso de um autômato celular multipartículas para simulação de processos de dopagem e oxidação térmica do silício. Este autômato celular multipartículas se mostrou uma alternativa interessante para simulação da dopagem de Si por impurezas. Aqui é apresentado também detalhes do software desenvolvido, o simMEMS, para incorporar estes autômatos celulares em um único ambiente, permitindo assim a simulação do processo de fabricação completo de inúmeros dispositivos como MEMS, MOEMS e dispositivos e circuitos microeletrônicos. São também, como exemplo, o processo de fabricação de um microespelho para projeção digital de luz, uma porta lógica NOR e uma microagulha para uso em uma matriz de eletrodos com aplicações em biologia e medicina.

Palavras-chave: Autômatos celulares. Simulação. Microeletrônica, Sistemas microeletromecânicos.

ABSTRACT

COLOMBO, F. B. **Using Cellular Automata to Simulate Microfabrication Processes.** 2016. 150 f. Thesis (Doctorate). Escola Politécnica - Universidade de São Paulo, São Paulo, 2015.

Cellular automata and their variations are currently used to simulate a large range of physical phenomena. Of special interest to the microfabrication field are cellular automata for the propagation of circular and elliptical fronts as well as cellular automata for the simulation of silicon wet etching. Here an alternative cellular automaton for use in surface propagation is presented. This cellular automaton has several advantages over those reported in the literature, such as a smaller neighborhood e better results in large grid simulations. The results on the study on the viability of using a multiparticle cellular automaton for simulation of reaction diffusion systems in microfabrication simulation are also reported. The multiparticle cellular automaton was used to simulate both dopant diffusion in silicon and silicon thermal oxidation. This multiparticle cellular automaton was found to be of interest to the simulation of dopant diffusion in silicon. Details of the implementation of the software developed to incorporate these cellular automata, called simMEMS, and allow for the simulation of complex MEMS, MOEMS and microelectronic circuits and devices are also presented. The fabrication processes of several complex devices, including a micromirror for digital light projection, a quad NOR logic gate IC and a microneedle for use in a medical electrode array, are presented as an example of the capabilities of the simMEMS software.

Keywords: Cellular automata. Simulation. Microelectronics. Microelectromechanical systems.

LISTA DE FIGURAS

Figura 1 – Vizinhanças de von Neumann (a) e de Moore (b) em uma rede bidimensional quadrada. 16

Figura 2 – Princípio de Huygens aplicado à evolução de uma superfície. 17

Figura 3 – Surgimento de facetas durante a evolução de uma superfície. 21

Figura 4 – Tela para entrada de parâmetros para criação de substratos. 23

Figura 5 – Telas para seleção de processos e entrada de parâmetros para os processos. 24

Figura 6- Camadas do software. 28

Figura 7 – Volume de células para simulação. 29

Figura 8 – Substratos de diferentes tamanhos em volumes de células de mesmo tamanho (10x10x5 células). 30

Figura 9 – Resultado da deposição de um filme com espessura de uma célula sobre os diferentes substratos. 31

Figura 10 – Tela de criação de novo material. 32

Figura 11 – Tela para edição de materiais compatíveis. 33

Figura 12 – Tela de opções do processo PVD mostrando um exemplo de explicação de uma opção. Também é possível ver no canto inferior direito o console com informações para o usuário. 34

Figura 13 – Tela para edição e execução de listas de processos. 39

Figura 14 – Célula unitária tipo diamante. 41

Figura 15 – Célula diamante do Si, com um conjunto de átomos não repetidos destacados em verde. 42

Figura 16 – Uso de uma rede cúbica de células para representar a estrutura cristalina do Si. Cada bola colorida representa uma célula com um átomo do cristal. 44

Figura 17 - Exemplo da rede cristalina do Si em uma lâmina de Si tipo p com orientação (100). 44

Figura 18 – Sobreposição de dois tipos de redes para representar um mesmo substrato.....46

Figura 19 – Diagrama de classes para representar dados dos materiais e processos 48

Figura 20 – Fluxograma para tratar materiais salvos. 50

Figura 21- Estrutura usada para implementar as operações. 52

Figura 22- Padrão de projeto de software Observer aplicado para comunicação entre camadas.....53

Figura 23 – Fluxograma de uma simulação na versão 3.....58

Figura 24 - Separação das células entre os threads. 59

Figura 25 – Fluxograma da simulação na versão 3.....60

Figura 26 – Exemplo do tratamento de bordas para paralelização. 61

Figura 27 - Evolução temporal de uma simulação usando o autômato celular para simular a deposição. 68

Figura 28 – Corrosão isotrópica de uma microponta em Si simulada em diversos passos. As imagens (a) até (e) mostram corrosões de 75 células de profundidade. A imagem (f) compara o perfil da microponta resultante com o perfil de um quarto de disco. A estrutura tem 600x600x610 células. 70

Figura 29 - Imagens de um substrato com material de mascaramento fotogravado no formato de círculos (a) e o mesmo substrato após uma corrosão isotrópica (b) (CHEN et al., 2010). 71

Figura 30 – Exemplo da deposição isotrópica de uma camada de óxido sobre uma microponta de Si. 72

Figura 31 – Exemplo de uma deposição anisotrópica sobre um pedestal. As imagens 1a e 2a mostram o pedestal antes e depois da deposição, respectivamente. As imagens 2a e 2b mostram a vista em perfil antes e depois da deposição, respectivamente. 73

Figura 32 - Comparação entre as espessuras do filme no topo e na lateral do substrato. 74

Figura 33 – A figura mostra um pedestal antes (a) e depois (b) da deposição de óxido de Si por PECVD sobre um pedestal (a) (CARVALHO, 2012). 74

Figura 34 – Exemplos de um processo PVD com espessura sobre estruturas de altura variando de 20, 40 e 60 células. Abaixo das imagens é informado a razão entre as taxas de crescimento lateral e vertical. 76

Figura 35 – À esquerda está o resultado da deposição de um filme metálico de 2,3 µm de espessura por evaporação sobre um pedestal de fotoresiste com 7 µm de altura. À direita pode-se ver a primeira camada metálica de um circuito CMOS fabricada usando a técnica de lift-off (MORITZ, 1985). 77

Figura 36 – Exemplo de eletrodeposição de níquel sobre filme condutor de cobre, usando um molde de fotorresiste isolante para formar uma estrutura em forma de micropinça. Observe que a eletrodeposição pode extravasar a espessura do molde. 78

Figura 37 – A imagem (a) mostra uma imagem de microscopia ótica de micropinças fabricadas utilizando eletrodeposição. As imagens (b), (c) e (d) mostram em maior detalhe uma região da pinça usando um MEV. Pode-se ver na imagem (d) o resultado do crescimento do filme metálico por sobre o fotoresiste, formando uma estrutura que se projeta para além da base da pinça. 79

Figura 38 – Exemplo de um processo de eletrodeposição dentro de um pequeno canal feito em uma única etapa com espessura 100 e o mesmo processo em duas etapas com espessuras 49 e 51. 80

Figura 39 – Esquemático de uma porta lógica NOR no CD4001B (TEXAS INSTRUMENTS, 2016a). 84

Figura 40 – Layout final das quatro portas lógicas. Cada porta lógica ocupa um quarto da área do chip, sendo simétricas e espelhadas entre si. Os contatos seguem o mesmo layout definido no datasheet do CD4001B. 85

Figura 41 – Máscaras geradas para simular a fabricação do CI no simMEMS. 87

Figura 42 – Resultado da análise transiente de uma das portas lógicas. As entradas A1 e B1 foram excitadas com trens de pulsos. O1 mostra a tensão na saída da porta lógica. 88

Figura 43 – Gráficos para medida dos tempos de subida e de descida para a porta lógica. Apenas uma porta é apresentada pois as quatro apresentam comportamento idêntico. 89

Figura 44 – Visão tridimensional parcial do circuito fabricado a partir do layout criado no Microwind. 90

Figura 45- Etapas do processo de fabricação do circuito integrado. 107

Figura 46 – Condutores formando a base do dispositivo. 112

Figura 47 – Passos para formar a base do espelho e a mola de torção. .113

Figura 48 – Apoio e mola de torção finalizados. 114

Figura 49 – Processo de fabricação do espelho sobre o jugo e a mola de torção. No final o topo do espelho foi removido para mostrar o apoio. 115

Figura 50 – Imagens de uma matriz de microespelhos produzida pela Texas Instruments. No topo à esquerda pode-se os espelhos ao final do processo. Na imagem superior direita, um dos espelhos foi removido, expondo os apoios e a mola de torção por baixo. Na imagem inferior esquerda, todos os espelhos foram removidos. A imagem inferior direita mostra em detalhe os apoios e a mola de torção de um dos microespelhos. 116

Figura 51 – Estrutura fatiada após (a) 0, (b) 10, (c) 20, (d) 30 e (e) 40 minutos de corrosão úmida (SHIKIDA et al., 2004). 118

Figura 52 – Passos de fabricação da microagulha na versão 4 do simMEMS. 119

Figura 53 – Vista da ponta da microagulha usando (a) células dos autômatos celulares, (b) átomos da rede cristalina do Si e (c) resultado experimental (SHIKIDA et al., 2004). 120

Figura 54 – Simulação do processo de fabricação de uma microponta na versão 3 do simMEMS. 121

Figura 55 - Vista da ponta da microagulha usando células, à esquerda, e átomos, à direita, nos resultados gerados pela versão 3 do simMEMS. 122

Figura 56 – Resultados de simulação da fabricação de uma matriz de micropontas usando o processo descrito acima. A simulação foi realizada no software VisualTAPAS (XING; GOSÁLVEZ; SATO, 2007). 123

Figura 57 - Distribuições independentes em todas as células.....128

Figura 58 – Funcionamento do autômato celular unidimensional para difusão.....129

Figura 59 – Direções do autômato celular multipartículas em 3D. 134

Figura 60 - Evolução temporal da resposta.....142

Figura 61 – Visualização de um corte no substrato dopado em um dos experimentos. A figura mostra apenas os materiais ocupando as células, com as células em cinza sendo Si não dopado e as células azuis sendo Si dopado. 143

Figura 62 – Concentração de dopantes após uma simulação com escala variando de 0 (vermelho) a 680 (azul). 144

Figura 63 – Visualização do plano XY em Z=50. 144

Figura 64 - Resultado da simulação de 99 passos de difusão a partir de um ponto. São apresentados os números de partículas por célula no plano XY passando por Z=50. Cada série representa uma coordenada Y de 0 a 99 e o seixo horizontal apresenta a posição X, em células. 145

Figura 65 - Comparação entre resultados simulados e o modelo removendo os valores nulos.....146

Figura 66 - Resultados da simulação de uma dopagem em um plano infinito no simMEMS. À esquerda pode-se ver o substrato após a dopagem. A escala da imagem à direita varia de zero (amarelo) a $4x10^5$ (azul) e mostra a segunda escala de cores disponível no software.....148

Figura 67 – Comparação entre resultados experimentais e modelo para difusão em um plano infinito com concentração constante.....148

Figura 68 – Resultado da simulação de dopagem usando o autômato celular multipartículas. 150

Figura 69 – Espessura do óxido de Si em função do tempo de oxidação simulado. 152

LISTA DE TABELAS

Tabela 1-Diferenças na discretização de uma superfície em função do formato das células. 45

Tabela 2 – Processos Disponíveis na versão atual. 63

Tabela 3 – Tabela da verdade para uma porta NOR. 82

Tabela 4 – Valor das variáveis usadas para simulação de difusão pontual. 143

Tabela 5 – Valor das variáveis usadas para simulação de difusão em um plano infinito. 147

SUMÁRIO

1 INTRODUÇÃO 14

1.1 Versões Anteriores do simMEMS 20

1.2 Objetivos 24

1.3 Contribuições e Publicações 24

2 DESCRIÇÃO DO SIMMEMS 27

2.1 Estrutura Geral do Software 27

2.2 Utilizando o simMEMS 29

2.3 A Camada UI 34

2.3.1 Renderização 35

2.3.2 Menus 36

2.3.3 Listas de Processos 37

2.4 A Camada Model 40

2.4.1 Armazenamento dos Dados 40

2.4.2 Armazenamento da Estrutura Atômica do Si 41

2.4.3 Materiais e Processos 47

2.4.4 Persistência dos Dados 49

2.5 A Camada Core 51

2.5.1 Implementação dos Processos Simulados 55

2.5.2 Processos Simulados com Autômatos Celulares 61

3 SIMULAÇÃO IDEALIZADA 63

3.1 Autômato Celular para Evolução Topográfica 65

3.2 Corrosão Isotrópica 69

3.3 Deposição Isotrópica 72
3.4 Deposição Anisotrópica 73
3.5 PVD 75
3.6 Eletrodeposição 78
3.7 Exemplos de Aplicação 81
3.7.1 Circuito Integrado CD4001B 82
3.7.2 Dispositvo de Microespelho Digital (DMD) 111
3.7.3 Corrosão Anisotrópica do Si (Microfabricação em Substrato) 117
4 SIMULAÇÃO USANDO PARÂMETROS FÍSICOS 125
4.1 Caminhada Aleatória 125
4.2 Caminhada Aleatória em um Autômato Celular 128
4.3 Processos de Difusão-Reação 132
4.3.1 Detalhes de Implementação 139
4.4 Validação do Modelo para Difusão 141
4.5 Visualização dos Resultados 149
4.6 Aplicação do Autômato Celular para Oxidação 151
5 CONCLUSÕES 155
6 BIBLIOGRAFIA 163

1 INTRODUÇÃO

Um autômato celular é definido, de forma informal, por um conjunto regular de células que ocupam um espaço n-dimensional. Cada uma dessas células possui um estado dentro de um conjunto finito de possíveis estados que cada célula pode assumir. Os estados das células são associados a um instante de tempo $t = 0,1,2,3, \ldots$. A evolução temporal dos estados de uma célula segue um conjunto fixo, para cada autômato celular, de regras que analisa o estado da célula e de uma vizinhança fixa e finita de células em um número finito de passos no passado (geralmente $t - 1$, mas podendo eventualmente incluir $t - 2, t - 3, \ldots$). Uma definição mais rigorosa de um autômato celular pode ser encontrada na referência Chopard e Droz, 2005.

Autômatos celulares são um tipo de máquina abstrata (HOPCROFT; MOTWANI; ULLMAN, 2006; WOLFRAM, 2002). Máquinas abstratas são modelos abstratos de computadores. Estes modelos são, em geral, mais simples e limitados em sua operação que um computador atual. No entanto podem ter o poder de resolver os mesmos problemas, embora não de maneira tão eficiente (HOPCROFT; MOTWANI; ULLMAN, 2006). O fato de uma máquina abstrata poder ser descrita de maneira simples e direta, como foi feita no parágrafo anterior, as torna, do ponto de vista matemático, mais simples do que trabalhar com o modelo de um computador real

Autômatos celulares, de forma mais específica, foram inicialmente utilizados para estudar a capacidade de máquinas se reproduzirem. John von Neumann propôs um autômato celular que operava sobre uma rede de células bidimensionais quadradas em que cada célula poderia ter até 29 estados. Os estados dessas células são atualizados em paralelo a cada passo usando o seu estado e o estado das células ao norte, sul, leste e oeste seguindo um conjunto de regras. Nesse autômato celular, uma estrutura composta de milhares de células é capaz de gerar uma cópia completa de si mesma, que também é capaz de gerar outra cópia de si mesma, provando assim que é possível desenvolver uma máquina complexa que seja capaz de criar algo tão complexo quanto ela mesma (BURKS, 1970). Esse campo de pesquisa ainda é de interesse e diversos autômatos celulares mais simples que também são capazes de

auto reprodução foram desenvolvidos (BYL, 1989; NEHANIV, 2002; TEMPESTI, 1995; WOLFRAM, 2002).

Autômatos celulares e suas variações são atualmente utilizados para simulação de diversos processos físicos (CHOPARD; DROZ, 2005). O método lattice Boltzmann tem sido utilizado com sucesso para simulação fluídica (AIDUN; CLAUSEN, 2010; BENZI; SUCCI; VERGASSOLA, 1992; CHEN; DOOLEN, 1998), bem como aplicações diversas como a simulação da transmissão de luz (MINK et al., 2016) e de campos eletromagnéticos (HANASOGE; SUCCI; ORSZAG, 2011). De especial interesse para o campo de simulação de processos de microfabricação são autômatos celulares para evolução de frentes circulares e elípticas (SIRAKOULIS; KARAFYLLIDIS; THANAILAKIS, 2005). Estes autômatos celulares têm inúmeras aplicações visto que muitos processos e fenômenos podem ser reduzidos ao problema de determinar como uma superfície evolui com o tempo (KARAFYLLIDIS; THANAILAKIS, 1997).

Embora autômatos celulares para evolução de superfícies sejam de interesse para a simulação de microfabricação, não são os únicos. Existem também autômatos celulares para corrosão anisotrópica de silício que têm sido utilizados com bastante sucesso (FERRANDO; GOSÁLVEZ; COLÓM, 2012; FOSTER; NIEMINEN; GOSA, 2002; GOSÁLVEZ et al., 2008; THAN; BÜTTGENBACH, 1994). Ao longo deste texto serão apresentados outros tipos de autômatos celulares que podem ser utilizados para simulação de processos de dopagem e oxidação, que também são fundamentais para fabricação de dispositivos microeletrônicos.

Autômatos celulares cujo número de possíveis estados é infinito são também chamados de autômatos celulares contínuos. Quando se simula o autômato celular contínuo usando um número fixo de bits para representar um número em ponto flutuante, o número de estados é, a rigor, finito e ainda se trata de um autômato celular. No entanto, autômatos celulares contínuos, como os utilizados no método Lattice Boltzmann, apresentam dificuldades e características próprias e é útil tratá-los como uma classe própria.

A vizinhança usada pela regra de evolução também deve ser fixa e não muda em função da posição da rede sendo analisada. No entanto, incorporando mais informações no estado das células, é possível simular tanto a variação da vizinhança

em função da célula como a regra a ser aplicada. Existem algumas vizinhanças que são muito utilizadas, inclusive no presente trabalho, e tem um nome próprio.

A vizinhança de von Neumann, apresentada na Figura 1a, é composta pelas células ao norte, sul, leste e oeste da célula sendo analisada. Neste trabalho foi feita uma extensão para três dimensões para essa vizinhança que é composta pelas seis células que compartilham faces com a célula sendo analisada. Essa extensão também é chamada de vizinhança de von Neumann.

Um segundo exemplo é a vizinhança de Moore, apresentada na Figura 1b, que é composta em pelas células ao norte, nordeste, leste, sudeste, sul, sudoeste, oeste e noroeste da célula sendo analisada. Também pode-se realizar uma expansão para três dimensões dessa vizinhança, que passa a incluir todas as células que compartilham uma face, aresta ou vértice com a célula sendo analisada.

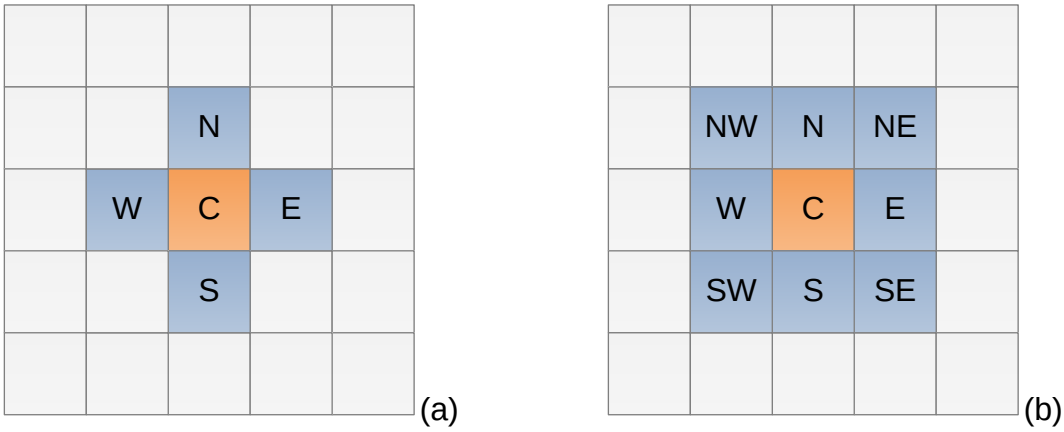


Figura 1 – Vizinhanças de von Neumann (a) e de Moore (b) em uma rede bidimensional quadrada.

O tamanho da vizinhança de um autômato celular é bastante importante visto que uma vizinhança grande leva a uma maior necessidade por processamento a cada atualização. Tal fato se torna mais notável em três dimensões. A vizinhança de von Neumann ocupa 4 e 6 células em duas e três dimensões enquanto a de Moore ocupa 8 e 26, respectivamente. Dessa forma, em 3D, a escolha pela vizinhança de Moore pode levar então a necessidade de analisar o estado de 4 vezes mais células em relação à vizinhança de von Neumann.

Quanto às regras para atualização dos estados, o mais comum para regras de um autômato celular é levar em conta o estado atual para determinar o estado seguinte.

É possível, no entanto, que diversos estados anteriores ($t-1, t-2, \ldots$) sejam utilizados, embora nesse caso seja preciso armazenar várias cópias dos estados das células. Quando os estados são finitos, é comum definir as regras usando cláusulas do tipo “se então” ou, se o número de possíveis opções for pequeno, como uma tabela. Para autômatos celulares contínuos, a regra pode também ser definida por uma equação.

Existe um terceiro tipo de autômato celular no qual a regra incorpora variáveis aleatórias. Nesse caso, a evolução do autômato celular deixa de ser determinística. Esse tipo de autômato celular é chamado de autômato celular probabilístico. Um exemplo simples de uma regra que incorpora uma variável aleatória é o caso onde se tem duas distribuições cujos valores são aleatoriamente invertidos com uma determinada probabilidade a cada passo.

O presente trabalho foca na aplicação de autômatos celulares para simulação de processos de microfabricação. Strasser e Selberherr (1995) propuseram um algoritmo para simulação de processos de corrosão e deposição em uma representação celular. O método proposto, assim como diversos autômatos celulares para simulação de avanço de superfícies, é baseado no princípio de Huygens (SIRAKOULIS; KARAFYLLIDIS; THANAILAKIS, 2001, 2002, 2005; STRASSER; SELBERHERR, 1995). O princípio de Huygens é um método proposto por Christiaan Huygens para explicar a propagação da luz. Ele propôs que cada ponto em uma frente de onda é uma fonte pontual. A cada instante, a frente de onda gera novas ondas infinitesimais. A nova frente de onda é então formada pelo envelope das frentes de ondas pontuais. Esse princípio também descreve a movimentação de uma superfície, seja devido a uma corrosão ou a uma deposição, como mostra a Figura 2.

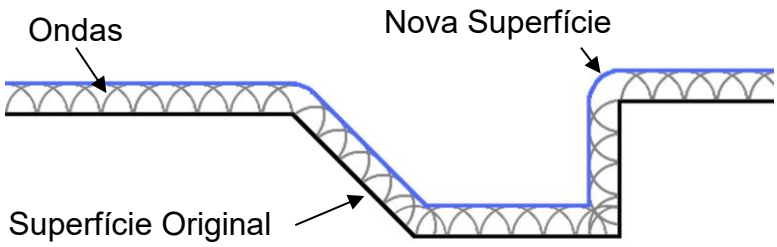


Figura 2 – Princípio de Huygens aplicado à evolução de uma superfície.

O algoritmo proposto aplica diretamente esse princípio de forma discreta. Cada célula na superfície é considerada como uma fonte pontual. O algoritmo consiste em marcar, para cada célula fonte, todas as células dentro de uma elipse. O formato da elipse é determinado pela taxa de corrosão, anisotropia do processo e também pela dimensão das células. Este algoritmo, embora não seja um autômato celular, pode ser facilmente convertido em um autômato celular mudando a lógica para que cada célula atualize o seu estado em função da distância até uma célula fonte. No entanto, para essa implementação a vizinhança de cada célula deve ter o formato da elipse a ser utilizada na aplicação do princípio de Huygens. Essa modificação é descrita em Colombo (2011).

Em trabalhos mais recentes, autômatos celulares contínuos, onde o número de estados das células não é finito, têm sido utilizados para realizar o mesmo tipo de simulação (HUANG et al., 2006; KARAFYLLIDIS, 1999; SIRAKOULIS; KARAFYLLIDIS; THANAILAKIS, 2002). Nestes casos, um valor inteiro entre 0 e 1 representa o volume de uma célula que está ocupada por um determinado material. A cada passo uma equação que leva em conta a ocupação de células vizinhas e sua disposição é aplicada para determinar a nova ocupação da célula atual. Estes autômatos celulares utilizam a vizinhança de Moore, o que reduz bastante o tamanho da vizinhança em relação a uma aplicação direta do princípio de Huygens, como descrito no parágrafo anterior.

Os autômatos celulares mencionados acima necessitam que ajustes nos parâmetros utilizados sejam realizados em função do tamanho da rede de células. Mais especificamente, o algoritmo proposto por Strasser e Selberherr requer a escolha da vizinhança a ser usada em cada passo para evitar o aparecimento de facetas (STRASSER; SELBERHERR, 1995). Já para os autômatos celulares contínuos, os parâmetros utilizados para atualização de estados foram escolhidos para um certo tamanho de rede, e apresentam resultados ruins para redes maiores (COLOMBO, 2011).

No presente trabalho é proposto um autômato celular para simulação da evolução topográfica. O autômato celular pode ser utilizado em duas ou três dimensões e utiliza a vizinhança de von Neumann. O autômato celular também permite rodar simulações sem a necessidade de ajustes pelo usuário para evitar o surgimento de facetas ou

problemas semelhantes. Além disso, permite determinar facilmente o vetor normal da superfície em determinado ponto, algo que não é facilmente disponível em outros autômatos celulares.

Embora qualquer processo de corrosão, deposição ou crescimento possa ser simulado pela simulação do avanço de uma superfície, é preciso saber a velocidade com que a superfície deve avançar para realizar essa simulação. Quando o processo é isotrópico, isso é simples. A superfície deve avançar com a mesma velocidade independentemente da orientação da superfície. Existem processos, como *chemical vapor deposition* (CVD) ou *physical vapor deposition* (PVD), para os quais existem modelos para calcular a taxa de crescimento em função da orientação da superfície[1]. Por outro lado, existem processos como a corrosão úmida de Si, nos quais é preciso se determinar experimentalmente a taxa de corrosão para todos os possíveis planos cristalográficos para se modelar o avanço da superfície.

Para o caso específico da corrosão anisotrópica de Si, existe uma alternativa mais interessante. A anisotropia do processo de corrosão está fortemente ligada à estrutura cristalina do Si. Existem diversos autômatos celulares que se aproveitam desse fato para simular o processo diretamente na rede cristalina do Si. A rede do autômato celular passa a ser a rede cristalina do Si e cada átomo é uma célula do autômato celular.

O modelo mais simples de autômato celular proposto para tratar da corrosão do silício tem uma regra fixa para remover ou não um átomo baseado em quantas ligações este átomo e seus vizinhos tinham ao substrato. Essa regra, embora simples, é capaz de diferenciar entre os principais planos cristalográficos envolvidos nesse tipo de corrosão, os planos (100), (110) e (111) (THAN; BÜTTGENBACH, 1994). Nesse autômato celular, a relação entre as taxas de corrosão dos planos cristalográficos é definida pela geometria da rede e o plano cristalográfico (111) tem taxa de corrosão nula. No mesmo trabalho, também foi proposta a adição de um componente estocástico à regra de remoção para permitir um melhor ajuste das taxas de corrosão de outros planos cristalográficos (THAN; BÜTTGENBACH, 1994). O autômato celular

[1] O presente trabalho não abordará de forma aprofundada o funcionamento dos diversos processos de microfabricação aqui mencionados. Para uma introdução a estes métodos de fabricação, recomenda-se o livros como "The Science and Engineering of Microelectronic Fabrication", por Stephen A. Campbell e "Micro and Nano Fabrication: Tools and Processes", por Hans H. Gatzen, Volker Saile e Jurg Leuthold.

estocástico permite realizar o ajuste das taxas de remoção atomística em função das velocidades de corrosão dos planos (100), (110) e (111). Embora sejam simples, esses autômatos celulares já permitem simular bem processos de corrosão razoavelmente complexos.

Existem também autômatos celulares contínuos que permitem ajustes melhores ainda para diversos outros planos cristalográficos e sem o ruído gerado por regras estocásticas (FERRANDO; GOSÁLVEZ; COLÓM, 2012). Esse ajuste é importante e permite, além de um melhor ajuste, a simulação de corrosão úmida em outros meios (e.g. *tetramethylammonium hydroxide* (TMAH)) (GOSÁLVEZ et al., 2009). Em um autômato celular contínuo define-se uma taxa de remoção para cada átomo em função da sua vizinhança. Em geral, quanto maior o número de ligações entre um átomo e o substrato, menor é a sua taxa de remoção (THAN; BÜTTGENBACH, 1994).

Dado o exposto acima, fica claro que existem duas abordagens possíveis para o uso de autômatos celulares para simulação de processos de microfabricação. A primeira é tratar o problema como um problema de evolução de uma superfície, usando autômatos celulares determinar como a superfície evolui, e modelos externos para determinar a taxa de crescimento ou corrosão da superfície em cada célula. A segunda envolve a utilização de autômatos celulares que simulam o processo em si, como é feito para simular a corrosão anisotrópica do Si.

Ambas as abordagens podem apresentar vantagens, o que leva a necessidade de se trabalhar com autômatos celulares que podem ser bastante distintos. Exemplo claro disso é o fato de que os autômatos celulares propostos para corrosão anisotrópica de Si operam sobre uma rede de células com o formato da rede cristalina do Si, enquanto que os autômatos celulares para evolução de frentes circulares operam sobre redes cúbicas. É preciso, portanto, de alguma forma integrar essas diversas redes e autômatos celulares para permitem a simulação em um mesmo substrato de diversos processos.

1.1 Versões Anteriores do simMEMS

Como será visto, uma parte importante do trabalho descrito aqui envolveu o desenvolvimento de um software para simular a fabricação de dispositivos

microeletromecânicos (microelectromechanical systems (MEMS)). As versões 1 e 2 do software foram desenvolvidas utilizando uma representação geométrica para os substratos e filmes. As superfícies são armazenadas como faces, que são um conjunto de arestas, cada uma composta por dois vértices. Operações como deposição ou corrosão de um filme ou do substrato são simuladas usando operações de deslocamento, remoção e criação de novas faces.

Essa abordagem geométrica traz algumas dificuldades durante a simulação dos processos de microfabricação comumente utilizados para fabricação de MEMS, principalmente no caso da corrosão anisotrópica do Si. Além disso, existem também problemas devido à complexidade geométrica das estruturas à medida que as superfícies evoluem.

Para simular a evolução da superfície, é preciso adicionar ou remover faces de maneira dinâmica. Isso é feito para evitar a perda de detalhes bem como para evitar problemas numéricos devido a arestas demasiadamente pequenas. A Figura 3 exemplifica o problema do surgimento de facetas. Quando uma superfície arredondada, representada por um conjunto de arestas, aumenta de raio mas o número de arestas é constante, começam a surgir facetas em uma superfície que deveria ser arredondada. Para evitar este fenômeno é preciso detectar o aumento das arestas e adicionar vértices intermediários.

Figura 3 – Surgimento de facetas durante a evolução de uma superfície.

Invertendo a ordem temporal na Figura 3 têm se a corrosão de uma superfície. Fica claro nesse caso o segundo tipo de problema, que é a diminuição das arestas. Se as arestas continuarem diminuindo, eventualmente pode-se ter vértices muito próximos uns aos outros, o que gera erros numéricos. Nesse caso, é preciso remover vértices para evitar o problema.

Um terceiro problema está ligado à interseção de faces. É comum que durante uma simulação ocorra a interseção de superfícies. É preciso detectar que essa interseção

ocorreu e tratar corretamente a evolução da superfície nesse caso, o que inclui a remoção de faces ou arestas desnecessárias.

A simulação por autômato celular, por outro lado, não sofre dos problemas geométricos listados anteriormente, visto que a colisão de superfícies pode ocorrer apenas entre células vizinhas e é facilmente tratada. Além disso, como mencionado previamente, também permite simular com melhores resultados alguns processos de microfabricação, como a corrosão anisotrópica de Si.

Com isso em mente, uma nova versão do software, o simMEMS versão 3, foi desenvolvida (COLOMBO, 2011). Nessa nova versão, todos os processos de simulação são implementados usando abordagens celulares, e autômatos celulares são utilizados sempre que possível. Assim sendo, nesse desenvolvimento diversos autômatos celulares foram estudados como alternativas para simular alguns processos de microfabricação e foram levantados os desafios para implementação de um software capaz de simular diversos tipos de processos sobre um mesmo substrato.

Em uma abordagem celular, as células devem ter um tamanho suficientemente pequeno para representar os menores detalhes de interesse. No entanto é comum que MEMS tenham um tamanho consideravelmente maior que o menor detalhe de interesse, principalmente se circuitos microeletrônicos são embutidos na mesma lâmina. Nesse caso, o número de células necessárias para a simulação pode ser bastante grande. Como consequência dessa grande quantidade de células, os tempos de processamento e o consumo de memória são bastante elevados.

A versão 3 do simMEMS permite a simulação da corrosão anisotrópica de silício usando um autômato celular conhecido como Basic Cellular Automaton (BCA), já mencionado acima (THAN; BÜTTGENBACH, 1994). Também permite a simulação de processos de deposição e corrosão seletiva de filmes, deep reactive ion etching (DRIE) de silício e o processo de fotogravação. Esse conjunto básico de funções foi escolhido porque permitia simular a fabricação de um grande número de dispositivos (COLOMBO, 2011).

Na versão celular do simMEMS as operações são feitas por meio de menus e caixas de diálogo. As opções nessas caixas de diálogo estão ligadas aos autômatos celulares implementados, e não às características físicas do processo. Por exemplo, durante a

criação de um substrato, o usuário escolhe o tamanho do substrato em número de células ao invés de informar uma dimensão.

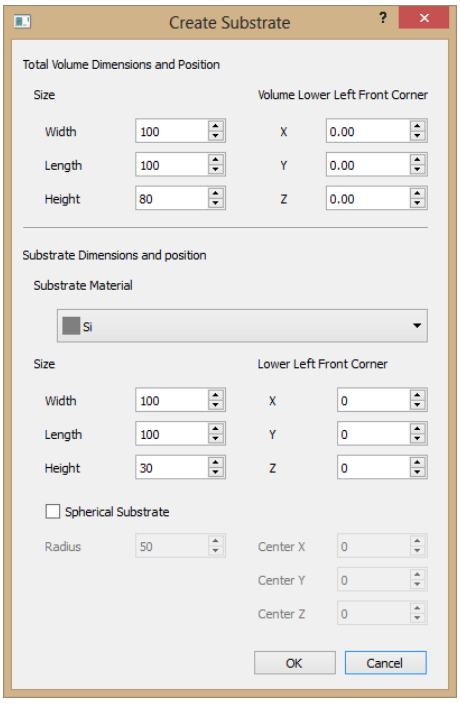


Figura 4 – Tela para entrada de parâmetros para criação de substratos.

De maneira semelhante, as opções para simulação são também os parâmetros para a simulação do autômato celular e não do processo. As telas de opções de simulação são mostradas na Figura 5. A caixa de diálogo da esquerda permite selecionar o processo a ser simulado. Algumas opções que são específicas a cada autômato celular são também selecionadas nessa caixa de diálogo. Por exemplo, as imagens usadas como máscaras fotolitográficas são selecionadas nessa caixa de diálogo. A caixa de diálogo à direta permite selecionar quantos passos do autômato celular serão executados. O autômato celular pode ser executado passo a passo ou o usuário pode selecionar um número de passos que serão executados em sequência. Nesse caso a superfície só é atualizada ao final da simulação de todos os passos.

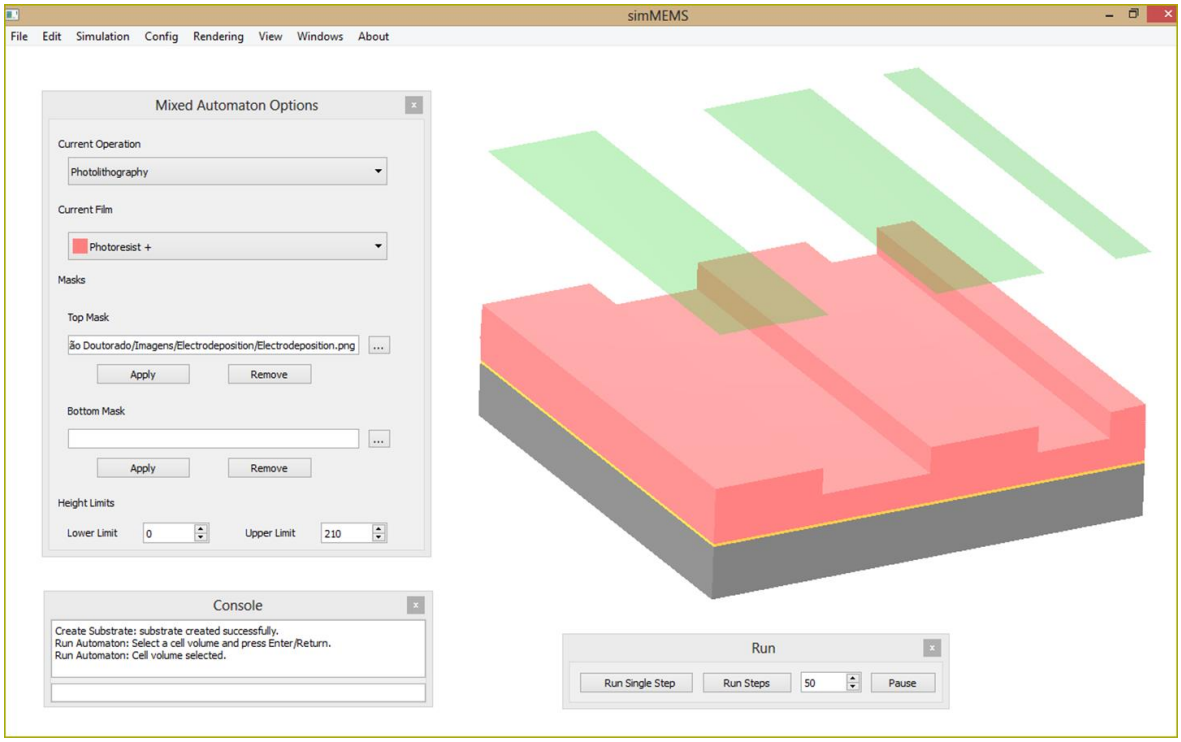


Figura 5 – Telas para seleção de processos e entrada de parâmetros para os processos.

1.2 Objetivos

Diante do exposto, o objetivo geral do presente trabalho é estudar a utilização de autômatos celulares na simulação de processos de microfabricação para a área de MEMS e microeletrônica. Isto será feito, por um lado, dando continuidade ao desenvolvimento do software de simulação (simMEMS) e, por outro lado, abordando o estudo de autômatos celulares específicos que permitam incluir parâmetros físicos nas simulações. Com isso, busca-se desenvolver uma plataforma de simulação que possa ser usada em âmbito acadêmico como ferramenta de auxílio ao ensino e aprendizado, mas que ao mesmo tempo permita estudos científicos rigorosos e confiáveis na área de microfabricação.

1.3 Contribuições e Publicações

O presente trabalho apresenta contribuições inovadoras na implementação das simulações por autômatos celulares e nos autômatos celulares utilizados para simulação dos processos de fabricação. Um dos objetivos do SimMEMS é fornecer

uma plataforma que permite a simulação de diversos tipos de processos de fabricação sobre um mesmo substrato (COLOMBO; CARREÑO, 2011). Nesse sentido, bastante esforço foi dirigido no sentido de criar uma ferramenta capaz de simular, de maneira rápida e simples, o processo de microfabricação de um dispositivo. O estado inicial deste software foi descrito em Colombo e Carreño (2011).

Embora o desenvolvimento em si não seja inovador, alguns problemas resolvidos são. Um dos problemas que surgem ao se implementar um software como o simMEMS é relativo ao uso de autômatos celulares que utilizam redes de células diferentes. No presente trabalho é apresentada uma possível solução para o problema, com usas vantagens e desvantagens, que é uma contribuição do presente trabalho. A solução aqui adotada é sobrepor as redes diferentes para permitir representar filmes e substratos em qualquer formato de rede.

No presente trabalho também é apresentado um novo autômato celular desenvolvido ao longo deste trabalho que apresenta diversas vantagens em relação aos autômatos celulares já descritos na literatura (COLOMBO; CARREÑO, 2013). Este autômato celular apresenta uma vizinhança menor que os demais, o que o torna interessante para uso em simulações tridimensionais. Além disso, é capaz de gerar melhores resultados, no que tange ao formato dos filmes e substratos corroídos ou depositados. O funcionamento e exemplos de aplicação foram publicados em Colombo e Carreño (2013).

Outra contribuição do presente trabalho é o estudo e aplicação de um autômato celular multipartículas já existente para uso na simulação de processos de dopagem de Si. Embora o autômato celular utilizado seja relativamente antigo, sua aplicação para este tipo de processo é inovadora. O autômato celular se mostrou adequado para simulação deste processo nas escalas usuais de circuitos microeletrônicos.

O autômato celular multipartículas também foi modificado no presente trabalho para permitir a simulação do processo de oxidação térmica do Si. Embora o autômato celular tenha sido usado para simular processos de difusão-reação, a reação era descrita entre duas espécies que se difundem no meio. O presente trabalho apresenta uma modificação que permite simular a reação entre a espécie que se difunde e o próprio meio.

Embora não descrito no presente trabalho, variações sobre outros autômatos celulares também foram desenvolvidas e testadas. Uma dessas variações envolve a simulação do efeito de obstruções sobre um processo de deposição (COLOMBO; CARREÑO, 2012). O algoritmo descrito por Strasser e Selberherr (1995) foi modificado de forma a modificar a taxa de deposição do filme em função de obstruções, como a presença de uma parede próxima à superfície, bem como variações para simular deposições não conformes de filmes de óxido de Si. Os resultados foram publicados em Colombo e Carreño (2012).

2 DESCRIÇÃO DO SIMMEMS

Neste capítulo, será descrita a estrutura do software bem como alguns aspectos da implementação dos algoritmos de simulação dos processos de fabricação. Primeiramente a estrutura do software será descrita, explicando a função das diferentes camadas e as classes implementadas. Em seguida serão descritos alguns detalhes da implementação dos autômatos que foram estudados para melhorar o desempenho do software e diminuir o consumo de memória durante as simulações.

2.1 Estrutura Geral do Software

O simMEMS foi implementado em três camadas, seguindo o modelo implementado na versão anterior. Essa abordagem foi escolhida inicialmente para separar a implementação da interface gráfica da implementação da simulação. Por questão de organização, os modelos para armazenar os dados do software também foram separados em uma camada.

A implementação em camadas é uma maneira de organizar o código fonte de um programa. O simMEMS é dividido em três camadas, como mostrado na Figura 6. A camada UI, que implementa os menus, utiliza a interface disponibilizada pela camada Core para realizar a tarefa escolhida pelo usuário. A camada Core, por sua vez, utiliza interface da camada Model para realizar a tarefa. Implementar um software em camadas implica que durante a implementação uma camada tem acesso apenas às funções da camada de baixo. Dessa forma o código que realiza determinada tarefa é encapsulado, de forma que eventuais mudanças na interface da camada Model (por interface entenda-se as funções que podem ser chamadas pela camada acima), por exemplo, não se propagam para o código escrito na camada UI. De forma semelhante, mudanças na camada que não alteram a interface não se propagam além da própria camada modificada.

Na versão 4 do simMEMS, a camada Model é responsável por armazenar os dados referentes aos volumes de células, materiais e processos. As funções para salvar e ler os dados ficam nessa camada. As funções para gerar e desenhar os meshes também ficam nessa camada. Todos os meshes são renderizados usando a

Application Programming Interface (API) OpenGL devido ao caráter multiplataforma do software (SELLERS; WRIGHT JR.; HAEMEL, 2013). A camada Core implementa as funções para simulação de processos de microfabricação. É nessa camada, portanto, que os autômatos celulares são implementados. Além disso, também implementa atualmente toda a lógica de obtenção dos dados para simulação, herança das versões anteriores do software. A camada UI implementa os menus e telas de diálogo para interação com o usuário. Também trata a entrada de dados pelo usuário por meio do mouse (como a rotação da cena e seleção de objetos).

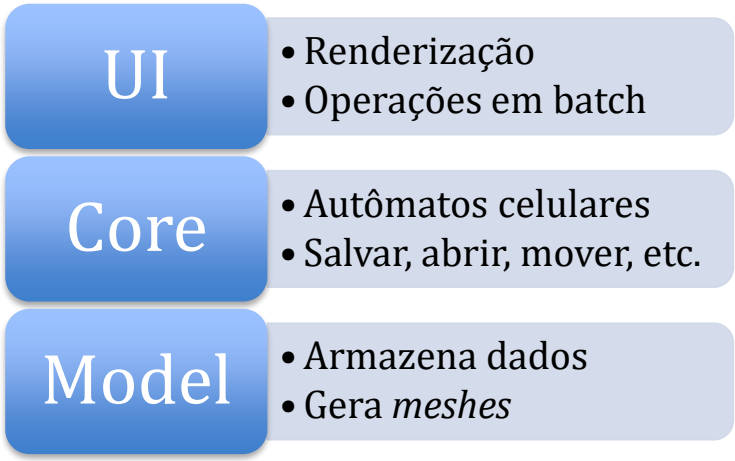


Figura 6- Camadas do software.

Todas as camadas se apoiam fortemente no framework Qt (QT COMPANY, 2016). Trata-se de um conjunto de bibliotecas que facilita o desenvolvimento de software multiplataformas, permitindo criar interfaces gráficas para Windows, Mac OS X e Linux sem a necessidade de escrever código específico para cada sistema operacional. Também fornece funcionalidades para leitura de arquivos, análise sintática de arquivos no formato Extensible Markup Language (XML) e criação e gerenciamento de threads. A versão 4 do simMEMS se baseia na versão 5.2 do Qt. O simMEMS também usa as bibliotecas Boost, que fornecem algumas funções avançadas que ainda não fazem parte da biblioteca padrão do C++ (“Boost C++ Libraries”, 2016). Entre elas destacam-se geradores de números pseudoaleatórios para uso nos autômatos celulares probabilísticos. O simMEMS também usa a implementação do Boost para *shared pointers*. Embora essa funcionalidade faça parte no novo padrão C++14 (“ISO International Standard ISO/IEC 14882:2014(E) – Programming Language C++”, 2014), a mesma não estava disponível no início do desenvolvimento.

2.2 Utilizando o simMEMS

Um usuário do simMEMS simula processos de fabricação sobre substratos. Como o simMEMS é desenvolvido com uma abordagem celular, todas as operações são realizadas sobre um volume de células cúbicas. Cada célula é “preenchida” por um único material, que inclui a possibilidade do material ser vazio. O substrato é um subconjunto de células dentro do volume de células, como pode ser visto na Figura 7.

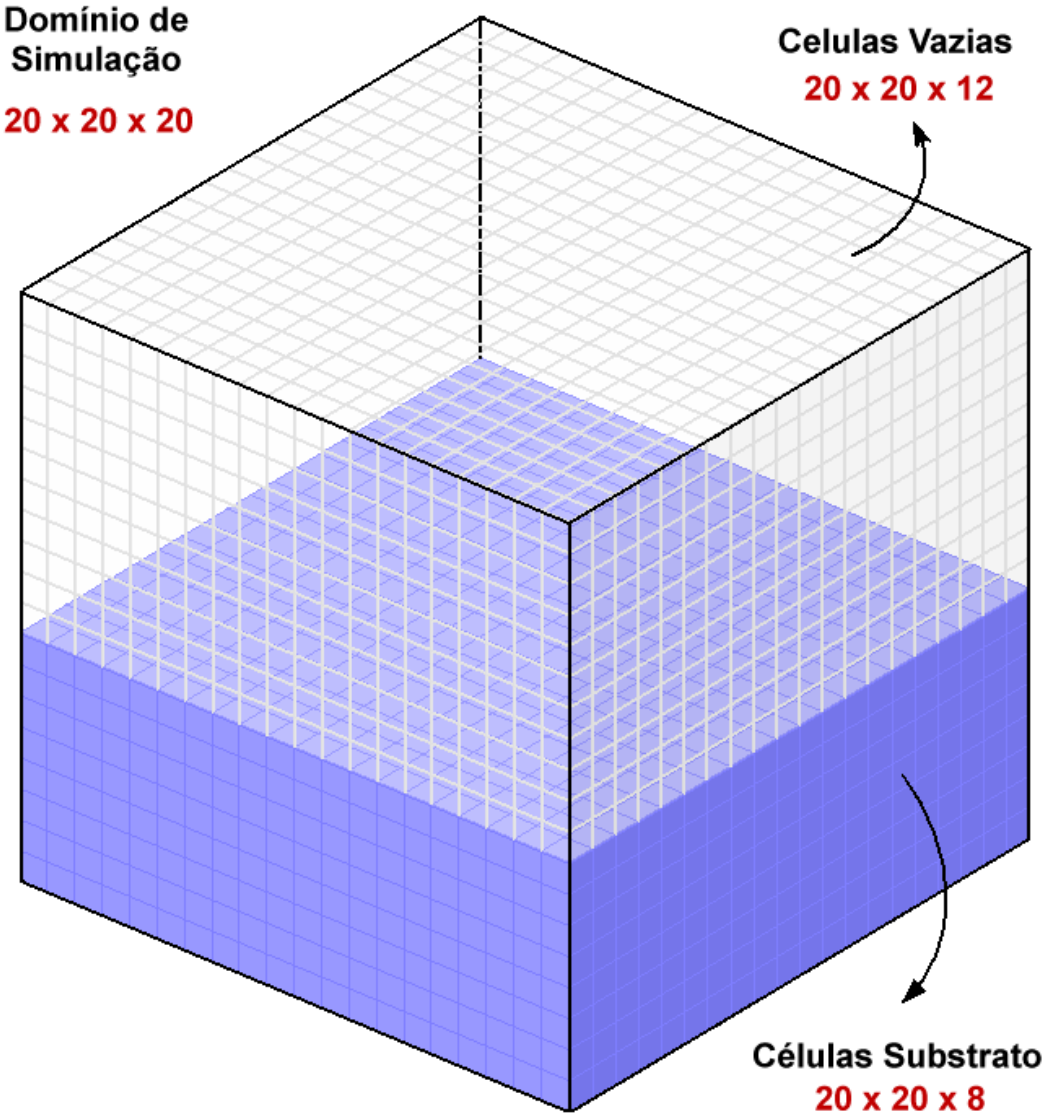


Figura 7 – Volume de células para simulação.

Para fazer uma simulação, o primeiro passo deve ser sempre a criação de um volume de células com um substrato dentro. O simMEMS permite criar substratos elípticos ou em forma de paralelepípedos. O usuário também pode escolher o tamanho do substrato dentro do volume bem como sua posição dentro do volume. O usuário também tem a opção de selecionar o material do qual o substrato é feito, por exemplo Si ou vidro. A Figura 8 mostra, a título de exemplo, quatro volumes de células de mesmo tamanho com substratos de tamanhos diferentes.

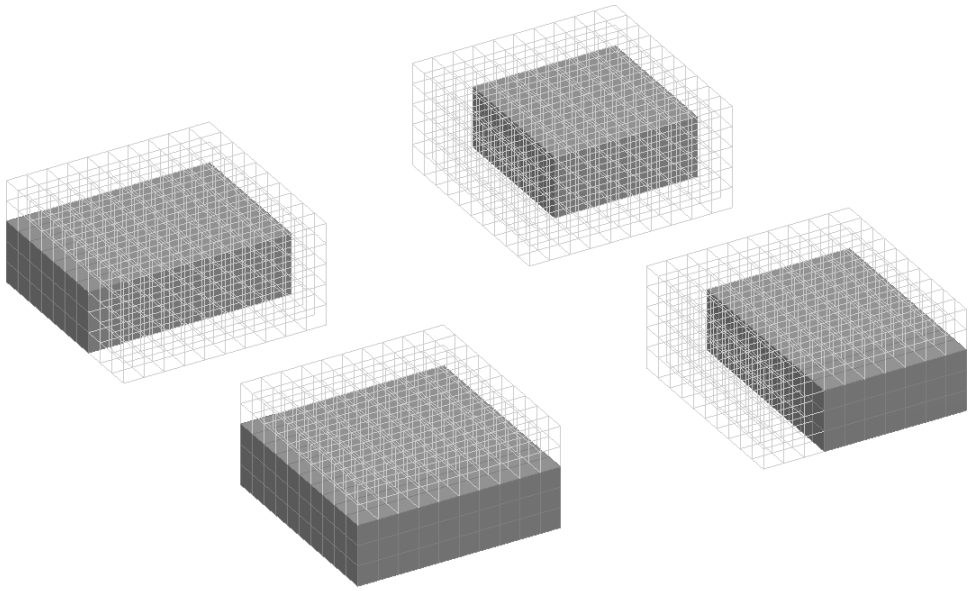


Figura 8 – Substratos de diferentes tamanhos em volumes de células de mesmo tamanho (10x10x5 células).

Todos os volumes têm 10x10x5 células. Em cada volume de células foi instanciado um substrato de tamanho e posição diferente. Do canto inferior esquerdo em sentido anti-horário os substratos têm, respectivamente, 10x10x3, 7x10x3, 7x7x3 e 10x7x3 células. Dessa forma, cada substrato expõe uma região diferente para os processos a serem realizados. A Figura 9 mostra o resultado de se realizar uma mesma operação de deposição sobre os diferentes substratos. Como se pode ver, o usuário tem a liberdade de selecionar as superfícies do substrato que estarão expostas modificando o tamanho do substrato e sua posição dentro do volume de células.

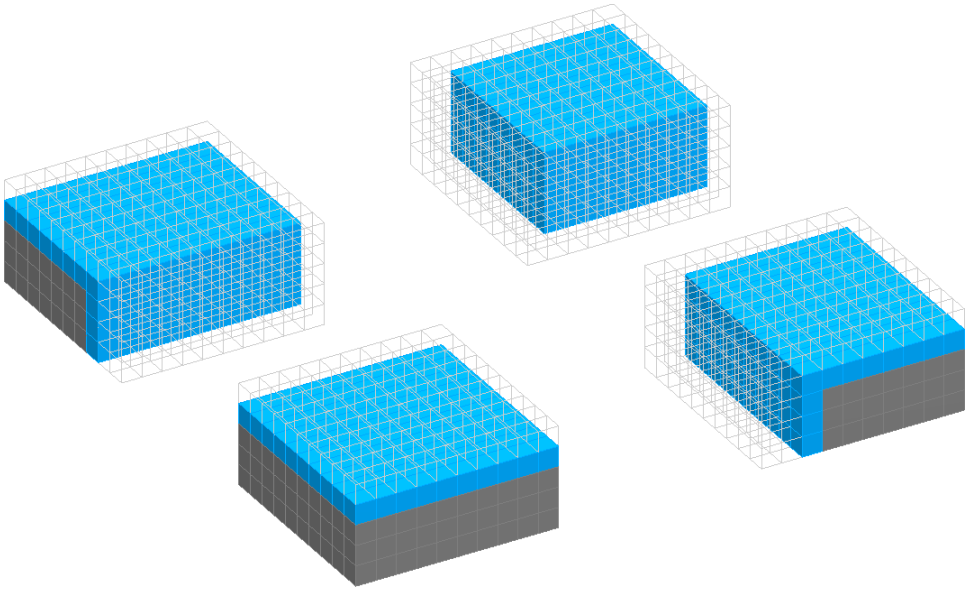


Figura 9 – Resultado da deposição de um filme com espessura de uma célula sobre os diferentes substratos.

Após criar um volume de células com um substrato dentro, o usuário pode então começar a realizar operações sobre esse substrato, como a deposição de um filme ou a corrosão do substrato.

O modo básico de visualização dos resultados é semelhante ao mostrado na Figura 9, no qual o substrato e filmes são desenhados como um conjunto de células cúbicas. As células vazias em geral não são mostradas, assim como as bordas entre as células. Caso o usuário deseje visualizar esses dados, existe a possibilidade de habilitar essas opções nos menus. Além do modo apresentado acima, existem ainda a possibilidade de usar um algoritmo marching cubes para gerar a superfície e a possibilidade de utilizar uma octree para gerar a superfície visível.

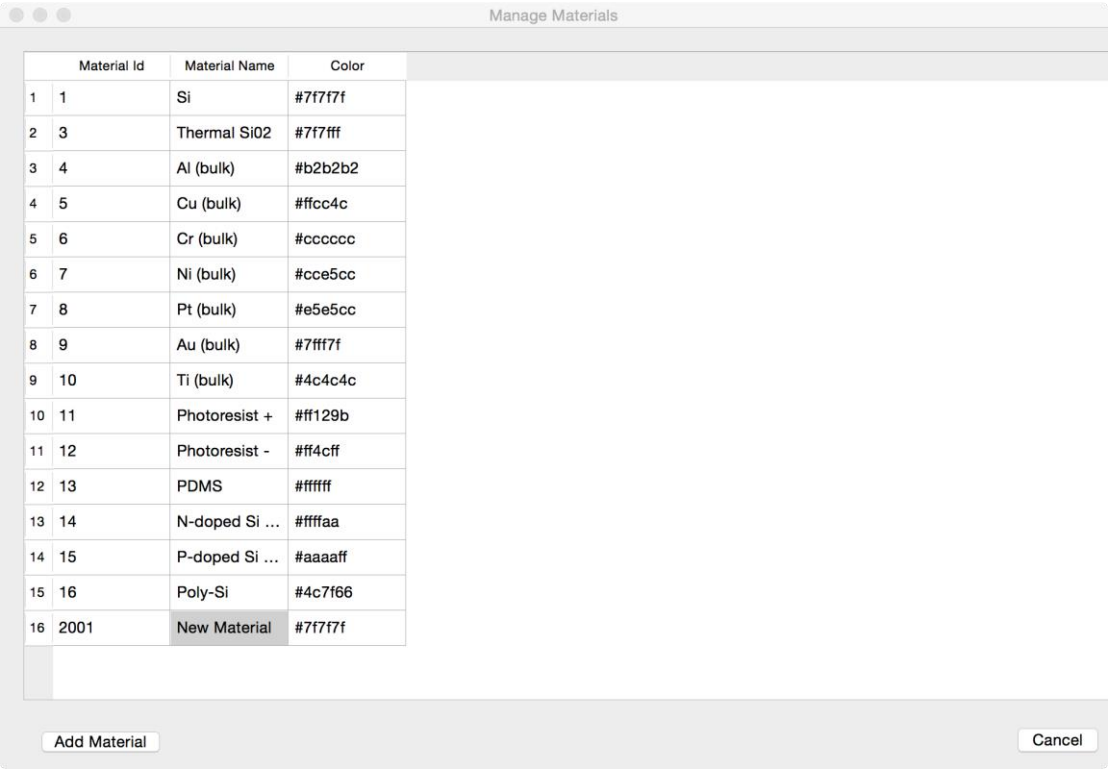
O usuário tem acesso a todas as funções de simulação disponíveis, inclusive a criação de substratos, no menu “Simulation”. O menu permite ao usuário selecionar operações a serem realizadas e o substrato sobre o qual realizar. Outra opção para utilização é a de criar uma lista de processos, na qual uma sequência de operações (como a deposição de um filme ou uma fotogravação) é criada. O usuário pode então executar todas as operações da lista sobre um substrato. O uso dessas listas de processos permite salvar a sequência de operações para fabricar um dispositivo, criando assim a possibilidade de realizar uma mesma sequência de processos diversas vezes. Por outro lado, a possibilidade de realizar as operações uma a uma, de forma dinâmica,

usando o menu “Simulation” permite visualizar mais rapidamente o resultado de uma possível sequência de fabricação, tornando assim mais fácil encontrar problemas na mesma.

Na versão atual do simMEMS, foi também introduzida a possibilidade da criação de novos materiais pelo usuário. O usuário pode criar um novo material e utilizar esse material em diversos processos. Atualmente os materiais são genéricos, não há nenhuma característica física atrelada a cada material. Quem define quais processos são compatíveis com quais materiais é o próprio usuário. Dessa forma, o comportamento padrão para criar e usar um novo material é:

1. Criar um novo material;
2. Editar a lista de materiais compatíveis com o processo para incluir o novo material;
3. Executar o processo.

Por exemplo, pode-se adicionar um novo metal chamado para deposição PVD criando o material com um novo nome, por exemplo, Alumínio. A Figura 10 mostra a tela para criação de materiais.



	Material Id	Material Name	Color
1	1	Si	#7f7f7f
2	3	Thermal Si02	#7f7fff
3	4	Al (bulk)	#b2b2b2
4	5	Cu (bulk)	#ffcc4c
5	6	Cr (bulk)	#cccccc
6	7	Ni (bulk)	#cce5cc
7	8	Pt (bulk)	#e5e5cc
8	9	Au (bulk)	#7fff7f
9	10	Ti (bulk)	#4c4c4c
10	11	Photoresist +	#ff129b
11	12	Photoresist -	#ff4cff
12	13	PDMS	#ffffff
13	14	N-doped Si ...	#ffffaa
14	15	P-doped Si ...	#aaaaff
15	16	Poly-Si	#4c7f66
16	2001	New Material	#7f7f7f

Figura 10 – Tela de criação de novo material.

Em seguida o novo material deve ser adicionado à lista de materiais compatíveis com o processo PVD. A Figura 11 mostra a tela de edição de materiais compatíveis. Do lado esquerdo são listados os materiais disponíveis, e do lado direito estão listados os materiais compatíveis com o processo selecionado da caixa de opções acima. Selecionando o material e usando os botões com setas no centro pode-se mover materiais de uma lista para outra. Uma vez que o novo material é adicionado à lista de materiais compatíveis (e após clicar em OK), ao tentar realizar um processo PVD, o novo material estará disponível na lista de seleção do material a ser depositado.

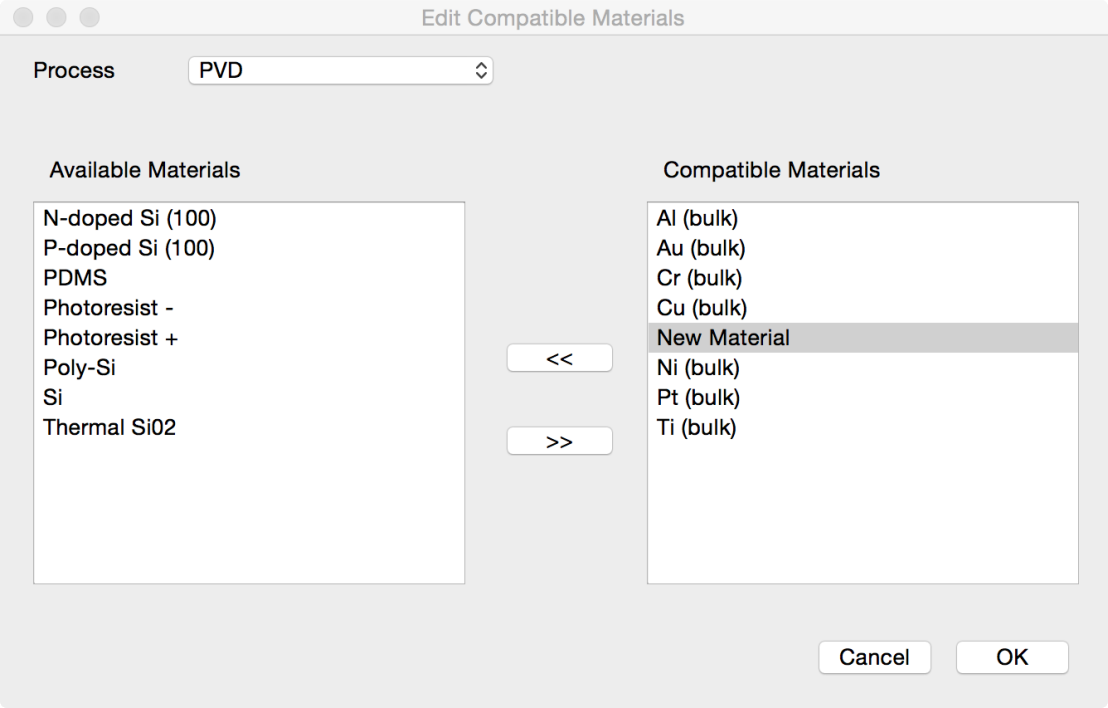


Figura 11 – Tela para edição de materiais compatíveis.

Cada processo pedirá ao usuário, através da interface gráfica, as informações necessárias para realizar o mesmo. O software tenta também, por meio da barra de status e do console disponível, orientar o usuário no uso do software. Também na tentativa de melhorar a experiência do usuário, diversas telas possuem textos para explicar as opções disponíveis e o significado dos valores pedidos que são mostrados ao deixar o mouse sobre o controle. A Figura 12 mostra um exemplo de uma tela onde essa dica foi adicionada. Também é possível ver o console com algumas informações para o usuário.

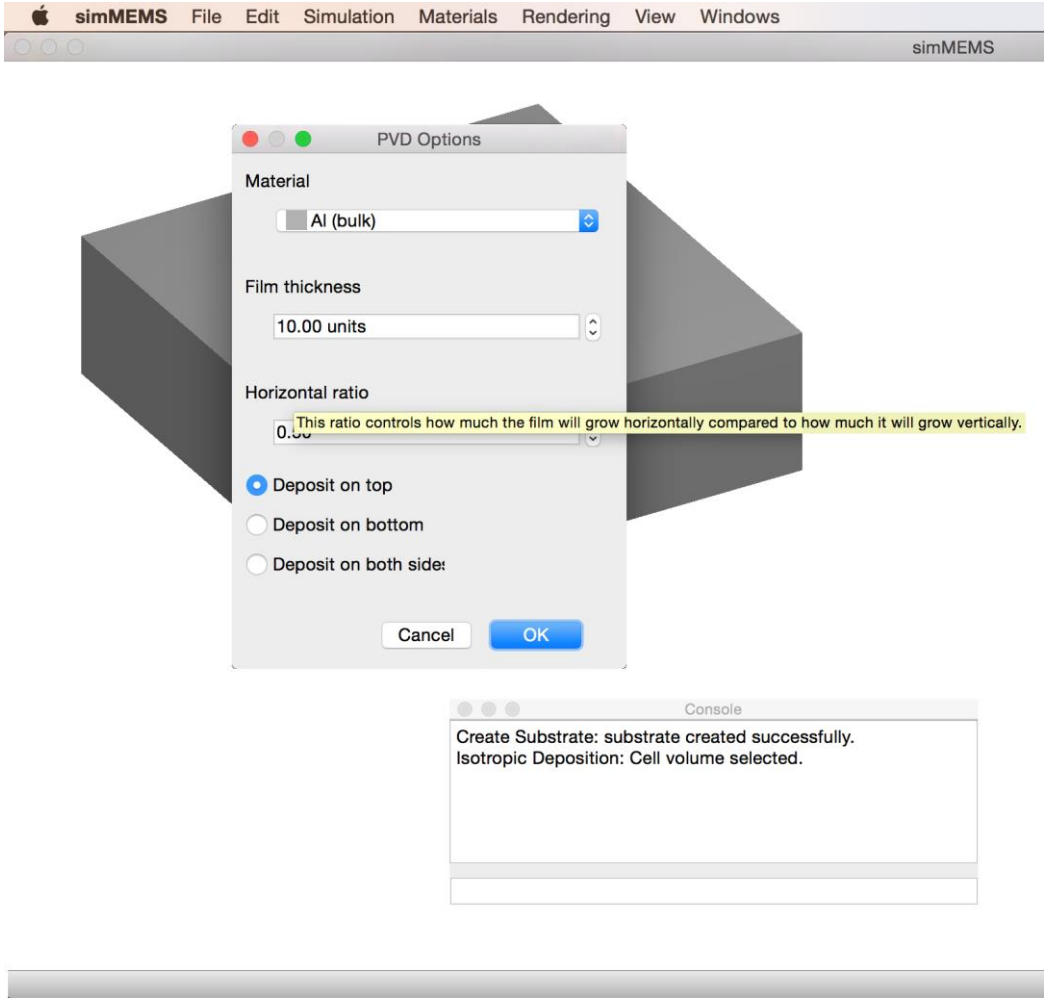


Figura 12 – Tela de opções do processo PVD mostrando um exemplo de explicação de uma opção. Também é possível ver no canto inferior direito o console com informações para o usuário.

A seção 3 inclui uma lista dos processos atualmente implementados na versão 4 do simMEMS. Com exceção dos processos de deposição e corrosão de Moore, corrosão anisotrópica de Si e do processo de fotolitografia, os demais processos foram implementados na versão 4 do simMEMS, durante o presente trabalho.

2.3 A Camada UI

A camada UI contém a lógica da interface gráfica do software. Isso inclui os menus e todas as janelas com as opções dos processos. Além disso, também contém a lógica para execução em lista dos processos. Na versão atual do software, a camada Core foi refatorada para permitir que seja usada como uma biblioteca. Isso permite seu uso

para simulações sem uma interface gráfica. Isso também permitiu a execução de uma lista de processos.

2.3.1 Renderização

O framework Qt fornece toda funcionalidade necessária para criar a interface gráfica do software. Também fornece um widget que inicializa e controla o contexto OpenGL, permitindo renderizar gráficos usando a implementação do OpenGL. Além disso, o framework Qt também fornece funcionalidades para carregar dinamicamente as funções necessárias para renderização usando versões mais novas do OpenGL.

O mais comum é que apenas a interface da versão 1.0 do OpenGL seja disponibilizada. Extensões das versões mais novas podem estar disponíveis no sistema, mas é preciso usar funções específicas a cada sistema para carregar as respectivas funções da biblioteca OpenGL. Existem algumas bibliotecas multi-plataforma que também realizam essa tarefa, mas o framework ajuda nesse sentido. Além disso, o Qt também fornece algumas funcionalidades para compilar o código dos shaders do OpenGL.

A renderização de todos os modelos até a versão 3 usa a interface da versão 1.0 do OpenGL. Como versões da API que utilizam um pipeline fixo (como a versão 1.0) foram definidas como obsoletas pelo Khronos Group, que controla e desenvolve a API OpenGL, foi iniciado neste trabalho o processo de conversão do código existente para uma versão mais moderna da API. A atualização está sendo feita em partes (KHRONOS GROUP, 2016; OPENGL.ORG, 2016).

Atualmente, o código que utiliza chamadas para glBegin, glVertex e glEnd está sendo convertida para utilizar vertex buffers. O mesmo está sendo feito para os poucos casos nos quais *display lists* eram utilizadas. No futuro, também será preciso modificar o código responsável por configurar as matrizes ModelView e Projection do pipeline fixo para que shaders passem a ser utilizados. Isso implica na mudança de chamadas para funções como glColor, glRotate e glTranslate, por exemplo.

2.3.2 Menus

Todas as operações são feitas através da interface gráfica. Os menus foram separados em 8 opções: “File”, “Edit”, “Simulation”, “Materials”, “Rendering”, “View”, “Windows” e “About”. A distribuição das operações dentro desses menus é importante visto que afeta diretamente a facilidade de uso do software. Atualmente a estrutura dos menus é plana; nenhum menu possui um submenu. Essa estrutura funciona atualmente pelo número de simulações que podem ser realizadas.

O menu “File” fornece operações ligadas a arquivos. É possível salvar ou carregar dados dos substratos para um arquivo. Outra opção é carregar diretamente um arquivo com a lista de processos para fabricar um dispositivo. Também é possível salvar imagens dos resultados de uma simulação através desse menu. Finalmente, é possível exportar alguns tipos de dados específicos, como a concentração de dopantes por célula.

O menu “Edit” permite realizar operações que tratam o volume de células como um objeto genérico. Isso inclui mover o volume de células e desfazer ou refazer operações que tenham essa opção.

Todos os processos que podem ser simulados são acessados pelo menu “Simulation”. Além disso, a opção de criar uma lista de processos pode também ser acessada através desse menu.

Opções de visualização dos volumes de células são acessadas através do menu “Rendering”. Esse menu permite escolher diversas opções sobre como os volumes de células são desenhados. A cor dos materiais, o uso ou não de bordas nas células e cortes nos volumes de células podem ser feitos aqui.

O menu “View” controla a câmera. Permite definir a posição da câmera, o zoom e o ângulo de visualização. Também permite configurar a posição das luzes, que podem ser adicionadas ou removidas e movimentadas para melhorar a visualização dos resultados.

A janela “Windows” permite abrir ou fechar a tela do console. Essa tela fornece informações de ajuda para o usuário e também pode ser usada, em alguns casos,

para obter informações do usuário. O último menu “About” fornece informações sobre o software, como versão e licenças, para o usuário.

2.3.3 Listas de Processos

Também incluído nessa camada está a lógica que permite criar uma lista com a sequência de processos de fabricação de um ou mais dispositivos. Todos os processos disponíveis no simMEMS que operam sobre um único volume de células podem ser adicionados à lista de processos, com exceção da extração de um filme (a seção 3 contém uma lista dos processos disponíveis no simMEMS). Além disso, é possível também criar substratos novos e salvar imagens dos resultados intermediários. Trata-se de uma funcionalidade bastante interessante, visto que permite realizar a simulação da fabricação de dispositivos complexos com um grande número de etapas de fabricação.

Essa funcionalidade poderia ser implementada na camada Core. No entanto, para garantir que a camada Core possa ser usada como uma biblioteca, optou-se por implementar a lógica na camada acima. Além disso, existem diversos dados para os processos que são necessários para a interface gráfica, mas não para simulação em si. Finalmente, a renderização em si é realizada na camada UI, e a opção de salvar imagens requer a lógica dessa camada.

Cada processo é representado por uma instância da classe ProcessData. A classe ProcessData armazena dois objetos do tipo Map, que são containers associativos da biblioteca padrão do C++ onde uma chave é mapeada para um valor. Um dos containers armazena valores, como profundidade ou tipo de material, usando o nome deste valor (e.g. “Thickness”) como chave. O segundo container armazena para as mesmas chaves usadas acima (e.g. “Thickness”) uma variável booleana que indica a visibilidade do valor, ou seja, se o valor deve ou não ser mostrado para o usuário na interface gráfica. A visibilidade é usada para permitir armazenar valores como a cor de um material, que serve exclusivamente para visualização e não afeta a simulação.

Os valores em si são armazenados dentro de instâncias da classe QVariant. A classe QVariant é disponibilizada pelo framework e age como uma união dos tipos mais comuns de dados do Qt e do C++. Uma união no contexto de C++ é uma classe que

pode armazenar apenas um dos seus membros de cada vez. O compilador aloca apenas a memória necessária para armazenar o maior tipo entre os tipos dos membros definidos. Dessa maneira, é possível definir uma variável para armazenar vários tipos de dados sem precisar alocar memória para todos os possíveis tipos. O uso de instâncias da classe Map que armazenam objetos do tipo QVariant permite armazenar de forma genérica os atributos de um processo.

Como os valores das chaves são mostrados para o usuário para identificar os atributos, devem ser escolhidos identificadores que são descritivos do parâmetro salvo. As chaves podem conter espaços e mais de uma palavra. Isso também facilita a edição direta do arquivo XML gerado a partir das listas de processos.

A visibilidade descrita previamente implica que alguns parâmetros não são apresentados para o usuário. Quando um material precisa ser armazenado para simulação, é preciso armazenar seu nome e cor (para mostrar para o usuário) e o seu identificador numérico (para uso na simulação). Nesse caso, os três dados são armazenados, e apenas um deles é visível: o nome do material.

No caso especial dos materiais, toda vez que uma lista de materiais é mostrada, a cor do material acompanha o nome para facilitar a seleção pelo usuário. Para esse caso, foi adotado um padrão nas chaves. Para mostrar uma cor ao lado da chave de um parâmetro para o usuário, um outro parâmetro é criado, com o mesmo nome sufixado do valor "_color". O software procura por este parâmetro e se o encontrar, mostra a cor que estará armazenada no valor associado à chave. É possível, portanto, dar qualquer nome à chave do material e mostrar mais de um material (com sua cor) por processo.

A tela de edição de processos é bastante simples, e foi reproduzida na Figura 13. A caixa de opções no canto superior direito lista todos os processos disponíveis para o usuário. O usuário seleciona um processo e clica no botão adicionar processo. A tela de opções do respectivo processo será mostrada, permitindo ao usuário selecionar os parâmetros do processo. Se o usuário confirmar as opções, o processo é adicionado à lista, mostrada no lado esquerdo da tela. A lista de processos permite modificar a ordem de processos já criados bem como a remoção de processos. Também é possível visualizar os dados do processo nessa lista bem como editar os valores, caso

necessário. Os botões no canto inferior esquerdo da tela permitem salvar e abrir listas de processos.

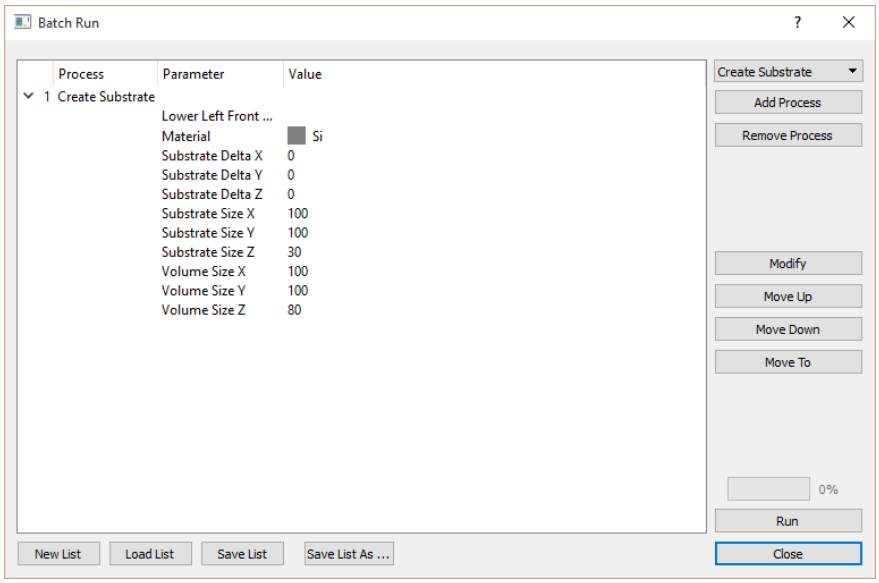


Figura 13 – Tela para edição e execução de listas de processos.

A simulação pode ser executada clicando no botão “Run”. A barra de progresso sobre o botão “Run” indica o andamento da simulação através da relação entre processos executados e processos totais. Fechar essa janela durante a execução da lista encerra a simulação após terminar a simulação do processo atualmente em execução.

Listas de processo podem executar uma simulação mais rapidamente se os resultados intermediários não forem desenhados. Nesse caso, as simulações serão realizadas em sequência e apenas o resultado final será desenhado. Isso pode ser muito mais rápido que uma simulação feita selecionando os processos um a um.

Um dos processos disponíveis é a criação de um substrato. Diversos substratos podem ser criados em uma mesma lista, mas as operações serão realizadas apenas no último substrato criado. Não é necessário que uma lista crie um substrato. O processo pode ser simulado sobre um substrato criado previamente, independentemente de processos de simulação já terem sido realizados sobre esse substrato.

2.4 A Camada Model

2.4.1 Armazenamento dos Dados

Os dados das células são armazenados em uma classe própria chamada TotalVolume. Essa classe encapsula a lógica de controle de acesso aos dados das células dos autômatos celulares. Um mutex é utilizado para controle de acesso aos dados, garantindo que apenas uma thread de cada vez possa acessar e modificar os dados. Isso impede, por exemplo, que a UI tente desenhar a estrutura durante uma simulação. Informações adicionais referentes ao volume de células, como seu tamanho e cortes feitos, também são gerenciados por essa classe.

O desenho dos substratos é realizado por uma classe separada, a TotalVolumePainter. O usuário pode escolher entre algumas opções de algoritmos para gerar a superfície que é desenhada. Cada algoritmo de geração de superfície é implementado por uma classe separada, visto que cada um tem as suas opções e atributos. Para simplificar o uso dessas classes, todas implementam uma mesma interface.

A maioria dos dados referentes às células utilizadas pelos autômatos celulares são armazenados em um vetor de objetos do tipo Voxel. Toda instância da classe Voxel representa uma célula e possui um número inteiro que serve como identificador do material que ocupa a célula. Também armazena as três distâncias e um booleano usados pelo autômato celular para deposição e corrosão de filmes desenvolvido no presente trabalho, que será descrito no item 3.1.

A rede cristalina de átomos usada pelo BCA para simular a corrosão anisotrópica do Si e os dados do autômato celular multipartículas para simular dopagem do Si são armazenados em outras estruturas. Esses dados são também armazenados na classe TotalVolume. A melhor localidade dos dados tem um impacto bastante positivo sobre o tempo de simulação, que é bastante menor na versão 4 do simMEMS em relação à versão 3. Armazenar os dados em vetores independentes, e não na classe Voxel, também permite alocar os dados somente se um autômato celular que necessita dos dados for utilizado pelo usuário.

2.4.2 Armazenamento da Estrutura Atômica do Si

O autômato celular BCA, como já mencionado, opera sobre os átomos da rede cristalina do Si. Dessa forma, cada célula do BCA corresponde a um átomo na rede cristalina. É preciso, portanto, representar um substrato de Si através das células cúbicas mostradas na Figura 7 bem como por uma rede cristalina que ocupa o mesmo volume. A primeira representação é usada para processos de corrosão isotrópica e para depositar filmes sobre o substrato do Si e a segunda para simular a corrosão úmida do Si em KOH. Como se tratam de duas representações do mesmo substrato, também é preciso relacionar as duas representações de forma que ao remover átomos, as respectivas células são removidas e vice-versa. A seguir será descrita a maneira como a rede cristalina é armazenada e como essa relação entre as duas representações é feita.

A estrutura atômica do Si é armazenada aproveitando a repetição apresentada por toda rede cristalina. A rede cristalina do Si possui uma célula unitária que pode ser repetida um número qualquer de vezes para montar um cristal de tamanho arbitrário. A célula unitária do Si é mais comumente descrita como uma célula diamante, apresentada na Figura 14.

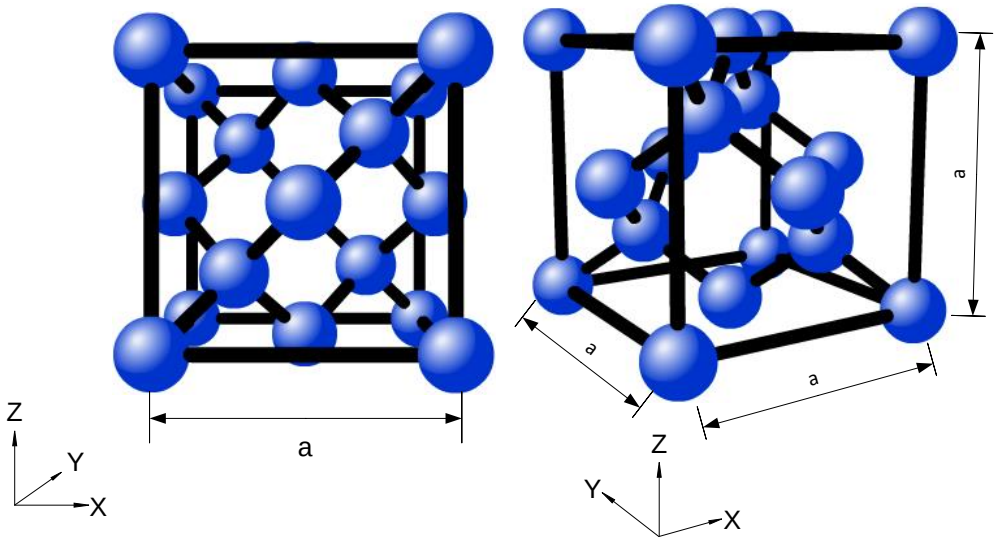


Figura 14 – Célula unitária tipo diamante.

A célula unitária tipo diamante é uma estrutura cúbica de face centrada onde cada ponto da rede possui dois átomos idênticos. Nessa estrutura, cada átomo forma quatro

ligações com os vizinhos mais próximos. Todas as ligações possuem o mesmo comprimento $a\sqrt{3}/4$, em que a constante a representa a constante de rede, que é a dimensão de uma das arestas da célula unitária apresentada na Figura 14.

A célula unitária tipo diamante é geralmente utilizada para representar a estrutura cristalina do Si. Essa célula unitária pode repetidas nas direções X, Y e Z para criar um cristal de Si de tamanho arbitrário. A Figura 15 destaca em verde um conjunto de oito átomos dentro da célula unitária do Si. Estes átomos representam um possível conjunto de átomos que permite criar cristais de tamanho arbitrário, como descrito anteriormente, mas sem repetir átomos. Os átomos em azul são compartilhados com as células unitária vizinhas. Do ponto de vista computacional, evitar essa repetição é útil porque evita ter que tratar a remoção de átomos repetidos.

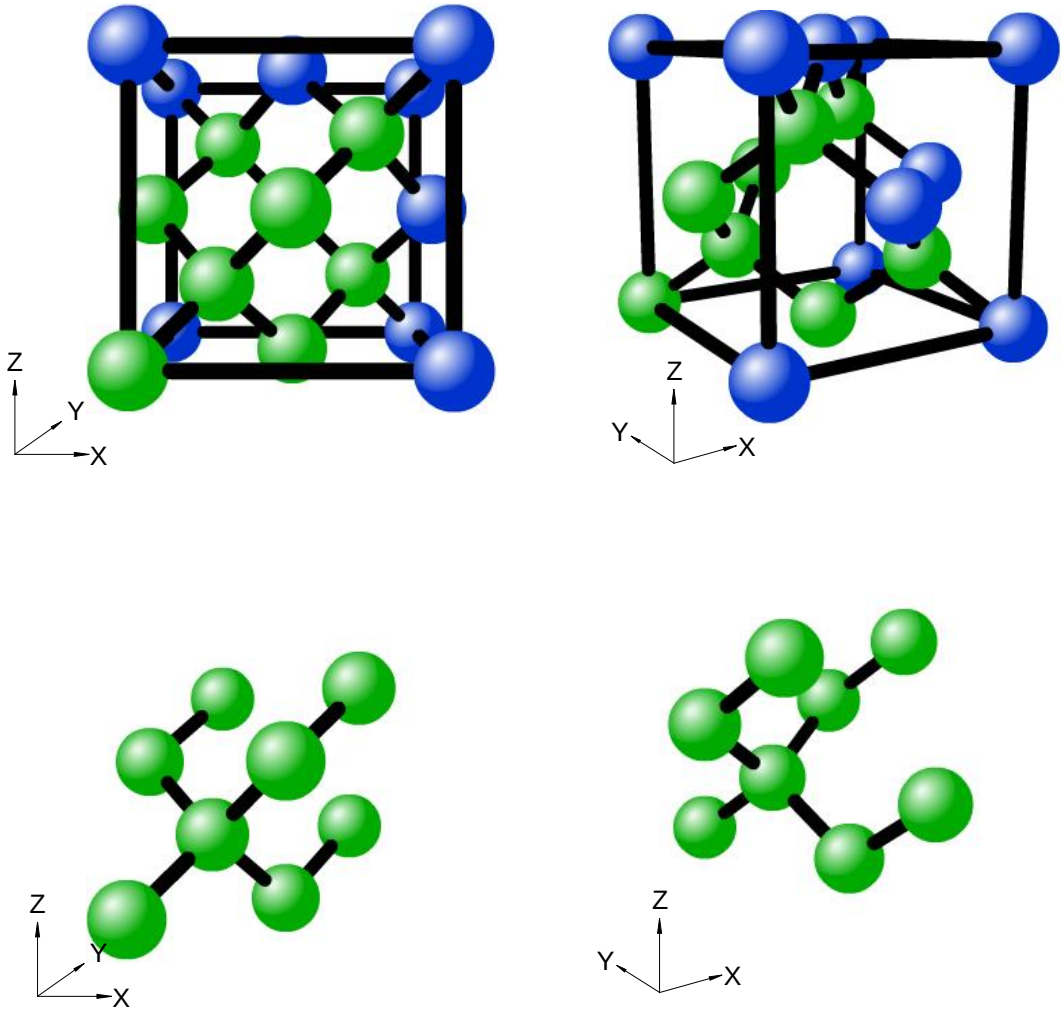


Figura 15 – Célula diamante do Si, com um conjunto de átomos não repetidos destacados em verde.

A estrutura cristalina do Si também pode ser utilizada para diminuir a quantidade de dados a ser armazenados por célula unitária. Considere que o cubo que contém os em verde da célula unitária tipo diamante da Figura 14 tem dimensões de 4x4x4 unidades. Os átomos de interesse (aqueles em verde na Figura 15) se encontram nas posições (0,0,0), (0,2,2), (2,0,2), (2,2,0), (3,3,3), (3,1,1), (1,3,1) e (1,1,3). Essas posições podem ser representadas de forma mais genérica para toda a rede cristalina pelas equações 1 e 2 a seguir:

$$X = Y = Z \ (mod\ 2) \tag{1}$$

$$X + Y + Z = 0 \ ou\ 1 (mod\ 4) \tag{2}$$

Ou seja, um cristal de Si também pode ser criado colocando um átomo de Si em cada posição que satisfaz uma das duas equações acima e se encontra dentro do cristal. De forma análoga, sabendo onde se encontra um dos vértices que formam o cubo da célula unitária tipo diamante da Figura 14, é possível calcular onde se encontram todos os átomos dentro da célula unitária usando as fórmulas descritas.

Pelo que foi descrito acima, fica claro que é possível representar um cristal de Si em uma rede de células cúbicas de maneira simples e direta. Nesse caso, as células que satisfazem as equações 1 e 2 terão átomos de Si dentro, e as demais serão vazias. A

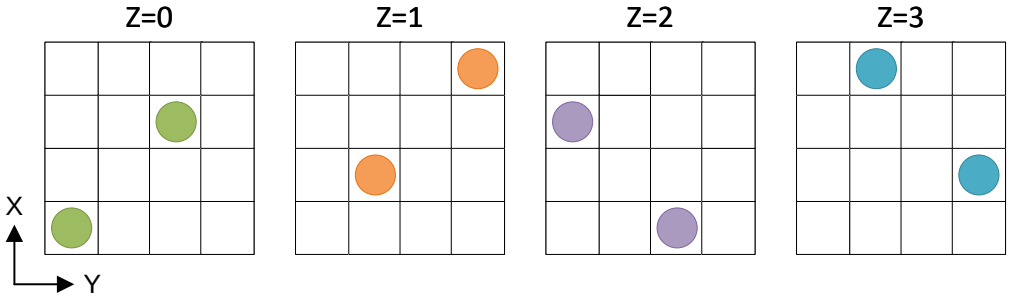


Figura 16 exemplifica isso para uma pequena rede de 4x4x4 células cúbicas.

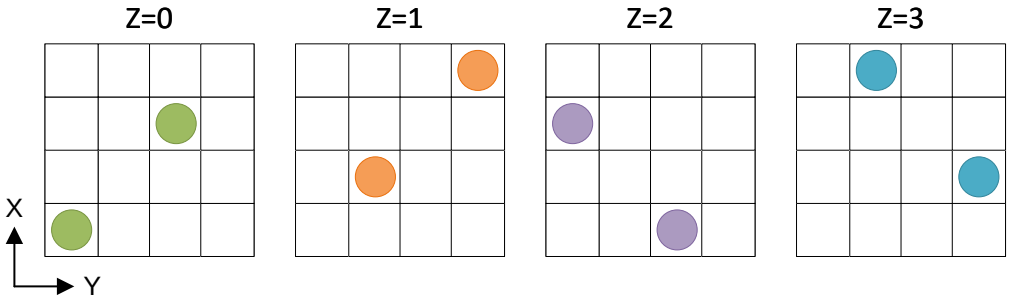


Figura 16 – Uso de uma rede cúbica de células para representar a estrutura cristalina do Si. Cada bola colorida representa uma célula com um átomo do cristal.

As lâminas de Si usadas em microeletrônica podem ser adquiridas com diversas orientações cristalográficas. Essas orientações são especificadas usando índices de Miller[2]. A orientação mais comum é chamada (100), por possuir na superfície da lâmina um plano cristalográfico da família {100}. As lâminas desse tipo possuem um chanfro maior para alinhamento na lateral direita, que segue o plano cristalográfico (110), bem como um segundo chanfro para indicar o tipo de dopagem. Assim, uma lâmina (100) possui uma rede cristalina onde a célula unitária tipo diamante da Figura 14 se encontra girada 45 graus, de forma que o topo da lâmina de Si se alinha com um plano da família {100} e o chanfro à direita se alinha com um plano da família {110}, como mostra a Figura 17.

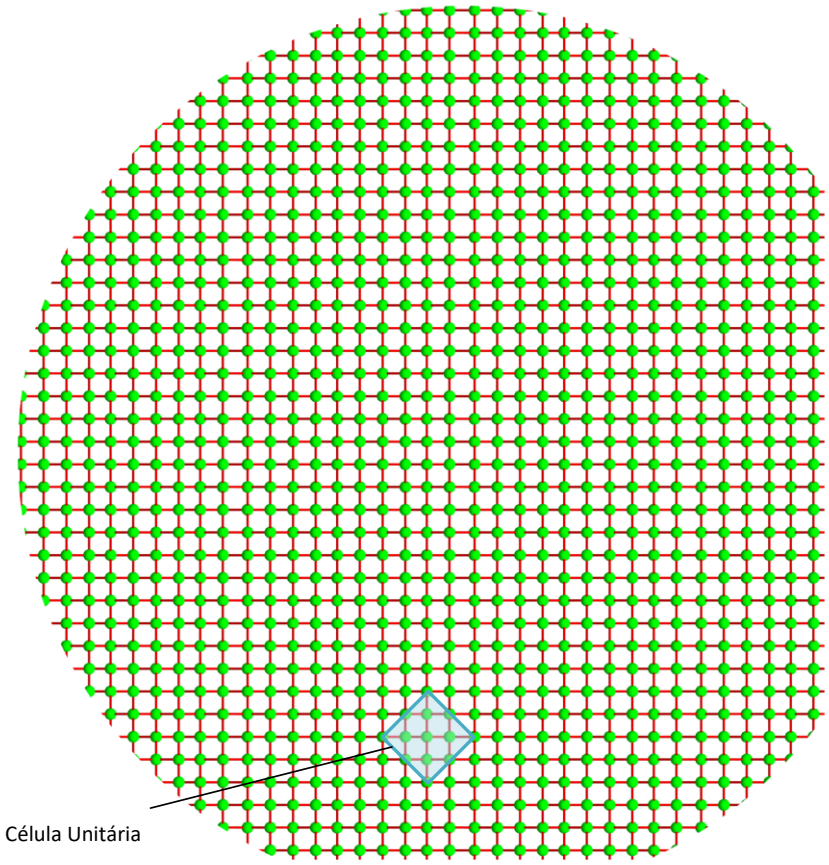


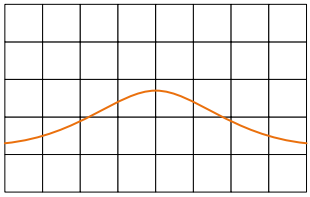
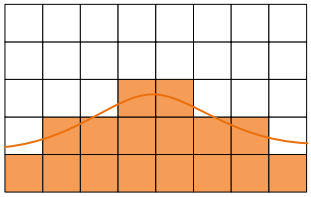
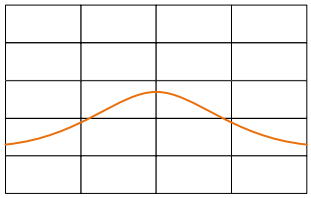
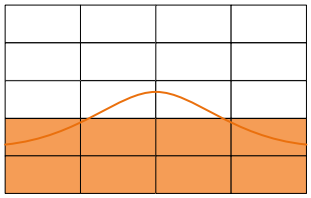
Figura 17 - Exemplo da rede cristalina do Si em uma lâmina de Si tipo p com orientação (100).

[2] Uma melhor descrição dos índices de Miller pode ser encontrada no livro Solid State Physics (ASHCROFT; MERMIN, 1976).

Essa mudança de orientação apresenta então uma complicação, visto que a estrutura cristalina na Figura 17 não se sobrepõe perfeitamente ao volume de células cúbicas alinhado com o chanfro. É possível, usando um esquema semelhante ao apresentado previamente para representar este cristal em um volume de células ortorrômbicas alinhadas com o chanfro e superfície da lâmina (GOSÁLVEZ et al., 2008). Além disso, a razão entre as dimensões X, Y e Z dessas células podem ser irracionais, o que ocorre para as células na Figura 17, não sendo possível então representar o cristal em um volume de células cúbicas.

Para os autômatos celulares usados na simulação de deposição ou corrosão de filmes, uma célula não cúbica não é recomendável. Isso porque, quando se discretiza um volume em células, passa-se a trabalhar com um erro devido à discretização do volume. Isso pode ser visualizada na Tabela 1. Para gerar as imagens nessa tabela, as células cujo centro estava abaixo da linha da superfície ideal foram coloridas. Embora o resultado para a célula retangular esteja correto, trata-se de um resultado que requer consideração sobre como ocorre a discretização para sua compreensão.

Tabela 1-Diferenças na discretização de uma superfície em função do formato das células.

Tipo de Célula	**Discretização**
Quadrada	Superfície ideal  Superfície discreta 
Retangular	Superfície ideal  Superfície discreta 

Quando uma célula não é cúbica, passa-se a ter um erro de discretização maior em uma direção em relação a outra. Essa diferença pode gerar resultados corretos, mas que parecem errados. As células dos autômatos celulares utilizados na versão 3 do simMEMS têm um formato não cúbico. Essa abordagem foi adotada para permitir

representar a estrutura cristalina do silício usando a rede de células dos demais autômatos celulares. No entanto isso leva a confusão na interpretação dos resultados pelos usuários.

A solução adotada neste trabalho é utilizar uma representação paralela para a estrutura atômica do silício. Nessa representação paralela, o cristal de Si é representado por um conjunto de células unitárias tipo diamante. A célula unitária tipo Si é girada de maneira a obter a direção desejada e é então repetida até preencher todo o volume do cristal. Isso permite armazenar apenas a posição da aresta da célula unitária tipo Si e calcular a posição dos átomos dentro de uma célula unitária a partir desse valor quando necessário. Essa estrutura cristalina é então sobreposta ao volume de células usado pelos demais autômatos celulares. A Figura 18 mostra um exemplo dessa sobreposição.

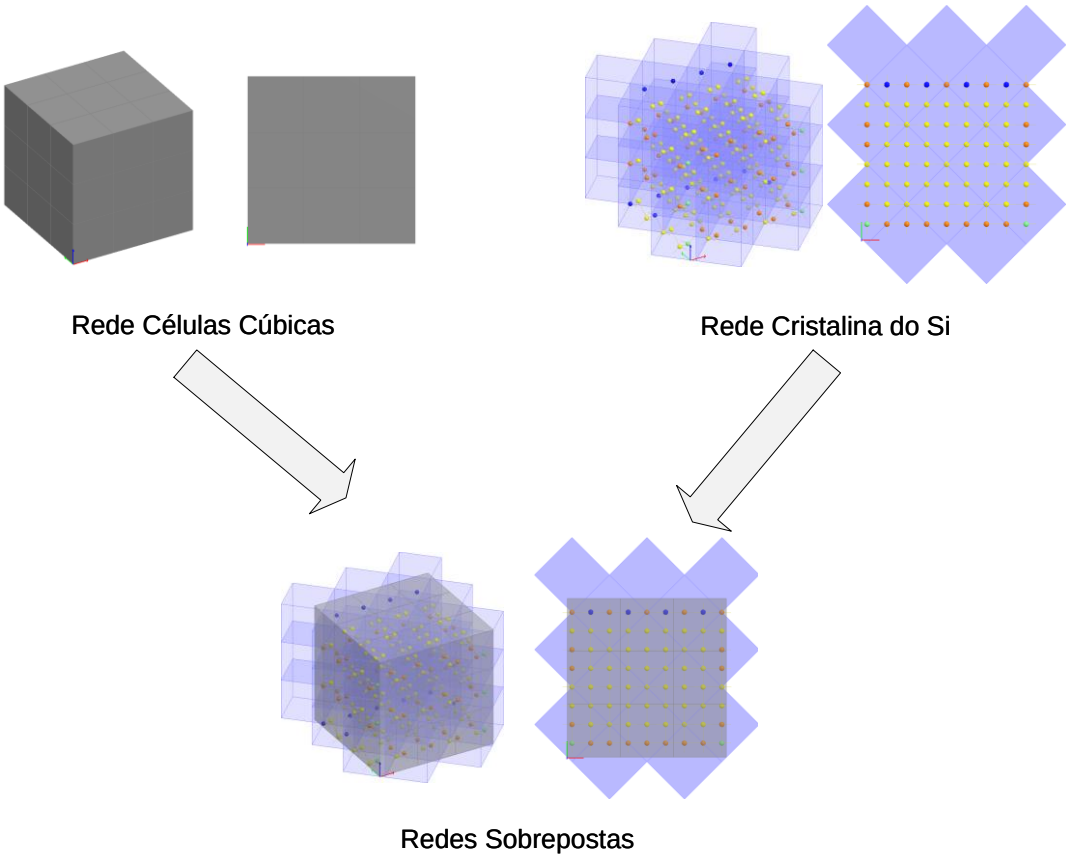


Figura 18 – Sobreposição de dois tipos de redes para representar um mesmo substrato.

Para determinar quais células da rede cúbica são corroídas durante uma simulação de corrosão anisotrópica de Si, a simulação é realizada na rede cristalina do Si.

Quando todos os átomos da rede cristalina de Si dentro de uma célula cúbica do substrato de Si são removidos, a célula cúbica é corroída.

Para que essa lógica de corrosão funcione é preciso ter pelo menos um átomo dentro de cada célula do substrato de Si. Para garantir isso, basta ajustar a razão do tamanho da célula unitária tipo diamante. Essa razão entre os tamanhos deve ser tal que seja impossível colocar uma célula do volume de células nos espaços entre os átomos do cristal. Para a orientação da lâmina da Figura 17 - Exemplo da rede cristalina do Si em uma lâmina de Si tipo p com orientação (100). o simMEMS usa uma razão entre os tamanhos de 1,4, isto é, a célula unitária tipo diamante tem uma dimensão 40% maior que a célula usada pelos demais autômatos celulares.

Embora a razão acima possa ser inferior a 1,4, quanto menor for a célula unitária, maior o número de átomos a ser armazenado para um substrato de mesmo tamanho. Assim, é interessante que essa razão seja a maior possível.

2.4.3 Materiais e Processos

Como descrito previamente, os autômatos celulares operam em um volume de células onde cada célula possui um material que a ocupa, como Si ou Al. Esse material é armazenado na classe Voxel usando um identificador, que é um número inteiro. Além disso, existe também uma classe Material, que armazena outras informações do material (como cor e transparência) e outras grandezas físicas de interesse (como densidade e coeficiente de difusão).

Nas versões anteriores do simMEMS, o usuário não tinha a possibilidade de criar novos materiais, e apenas os materiais fornecidos com o programa poderiam ser utilizados. O usuário tinha apenas a possibilidade de modificar a cor e transparência dos materiais, mas essas mudanças não eram salvas quando o programa era fechado.

Os processos que podem ser simulados no simMEMS na versão 4, em geral, bastante genéricos e podem ser usados com diversos materiais. Para não restringir o usuário aos materiais fornecidos pelo software foi adicionada a opção de criar novos materiais e modificar sua compatibilidade com os processos disponíveis, que foi descrita previamente. A estrutura da Figura 19 é utilizada para essa implementação.

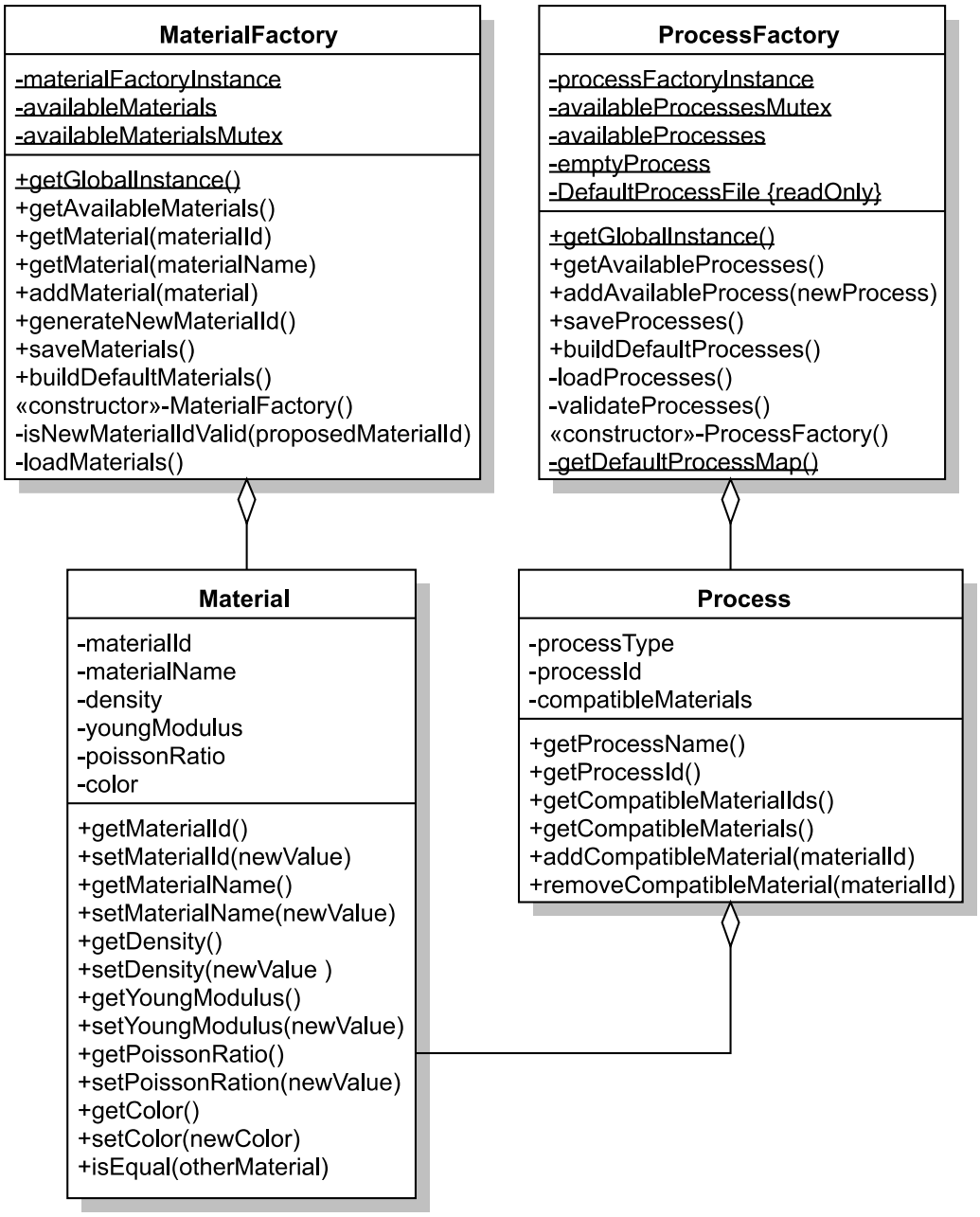


Figura 19 – Diagrama de classes para representar dados dos materiais e processos

A classe MaterialFactory é um singleton que gerencia os materiais existentes. Quando se deseja acessar informações sobre materiais, a instância dessa classe é obtida por meio da função membro getGlobalInstance. Um mutex é utilizado para gerenciar a adição ou remoção de materiais. O singleton é inicializado pela camada UI, que informa o arquivo onde os dados sobre os materiais disponíveis estão armazenados. Se o arquivo não está presente, é criado um arquivo com os materiais fornecidos com o software. A classe MaterialFactory também é utilizada quando se cria um novo

material, para criar um novo identificador para o material e para salvar os dados do novo material.

A classe Process representa um processo, que é uma operação que pode ser realizada pelo usuário. Exemplos de processos são a deposição de um filme ou a corrosão anisotrópica de Si. Alguns processos pedem ao usuário que informem quais materiais serão utilizados no processo.

Após criar um novo material, o usuário pode configurar também quais processos são compatíveis com esse material, adicionando o material à lista de matérias do processo. Um singleton também é utilizado para gerenciar os processos. A classe ProcessFactory é inicializada pela camada UI, que indica onde o arquivo com os dados dos processos estão armazenados. Caso o arquivo não exista, é criado com as definições padrão dos simMEMS. O singleton também é utilizado para salvar mudanças nos materiais compatíveis feitas pelo usuário.

2.4.4 Persistência dos Dados

Atualmente, todos os dados salvos pelo simMEMS utilizam o formato XML. Esse formato apresenta a vantagem de ser razoavelmente simples e pode ser lido e editado pelo usuário, se preciso. O framework Qt fornece funcionalidades para ler, escrever e analisar sintaticamente arquivos XML.

O esquema dos documentos é bastante simples visto que não existem muitos tipos de dados a serem salvos. Para o arquivo com os resultados de simulação, primeiramente, uma tag “Header” para armazenar a versão do arquivo é salva. Isso permite que novas versões sejam criadas e que as versões antigas ainda possam ser carregadas pelo software. Quando um arquivo não contém essa tag, é considerado como sendo gerado pela versão 3 do simMEMS e é convertido para a nova versão.

A tag mais importante é a “Document” que contém uma ou mais tags do tipo “Object”, cada uma descrevendo volume de células. A tag do objeto armazena o tamanho do volume em seus atributos. Uma tag filha “State” contém os dados serializados descrevendo o material que ocupa cada célula. Uma segunda tag filha “Position” armazena em seus atributos a posição do volume de células.

Além disso, a versão atual também salva em cada arquivo, junto com os dados dos volumes de células, uma lista com os materiais disponíveis. Como cada usuário pode criar novos materiais, cada usuário do sistema terá, à princípio, sua própria lista de materiais. Isso poderia apresentar um problema se um arquivo criado por um usuário for aberto por outro usuário em outro computador que não possui o novo material.

Antes de carregar os dados dos volumes de células, os materiais salvos no arquivo são carregados. Os dados são salvos dentro de uma tag “Materials” que contém diversas tags “Material” que contem em seus atributos os dados do material. Quando um material é carregado, é feita uma comparação com os materiais já salvos seguindo a lógica do fluxograma na Figura 20. Essa mesma lógica é usada sempre que materiais são importados pelo programa, como no caso de uma lista de processos descrita no item 2.3.3.

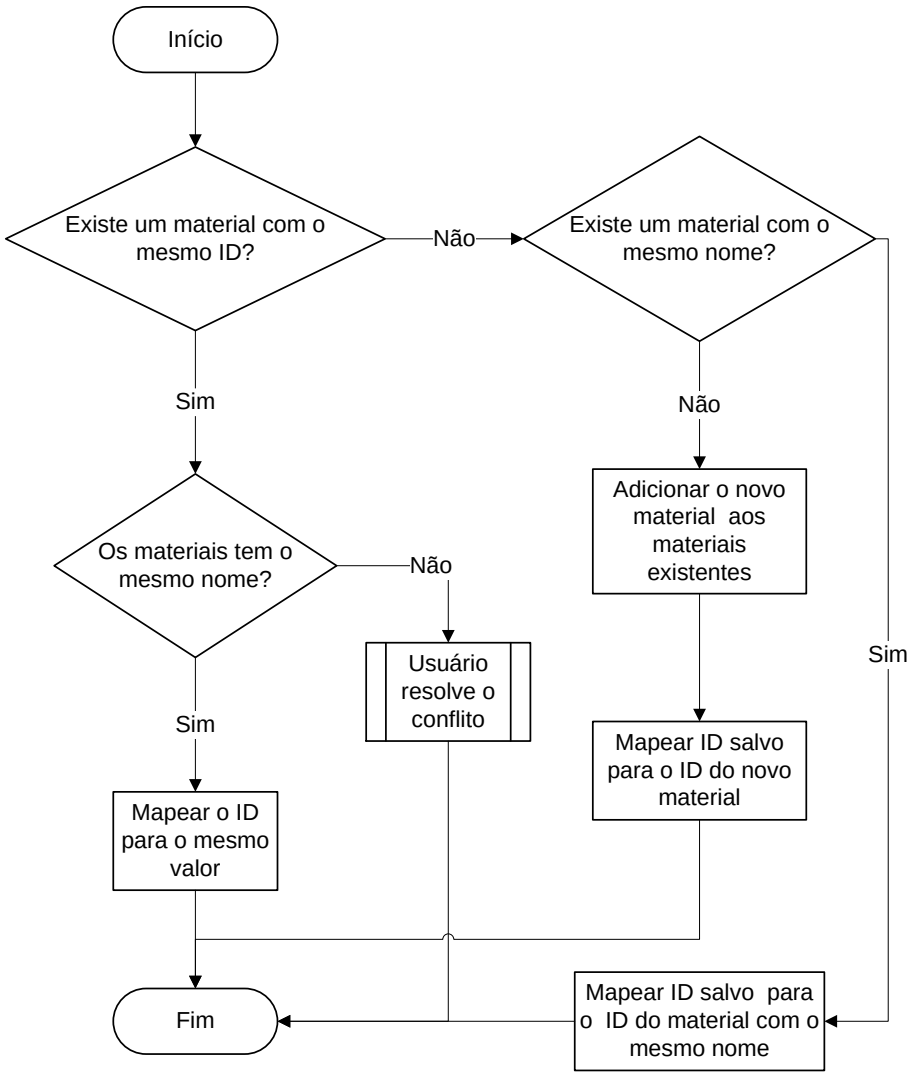


Figura 20 – Fluxograma para tratar materiais salvos.

Na Figura 20, o sub-processo para o usuário resolver o conflito consiste em perguntar ao usuário o que deve ser feito. Uma janela é aberta mostrando ao usuário os dois materiais em conflito e dando duas opções: criar um novo material com os dados do arquivo ou usar o material que já existe e descartar os dados vindos do arquivo.

2.5 A Camada Core

A camada Core contém a lógica das operações sobre os volumes de células, como salvar ou carregar dados bem como a lógica dos processos simulados. Para manter compatibilidade com o software desenvolvido em versões anteriores do simMEMS, a mesma estrutura para entrada de dados para simulações foi mantida. Assim, na versão 3 do simMEMS a camada Core possui diversas classes responsáveis por obter os dados do usuário para simulação. Por exemplo, se o usuário deseja criar um substrato, é preciso saber o formato e as dimensões desse substrato.

Mantendo essa estrutura para a camada Core permitiu o uso de classes desenvolvidas na versão anterior com poucas modificações. As limitações são as mesmas que a da versão anterior, e se resumem a limitações sobre quando os dados podem ser requisitados e usados. Como a obtenção dos dados é feita de maneira assíncrona, é preciso requisitar os dados e esperar até que os mesmos estejam disponíveis antes de usá-los para realizar as operações.

A Figura 21 mostra como as operações são implementadas. A classe Operation descreve genericamente como uma operação deve adquirir seus dados. Uma classe que herda de Operation, como a IonImplantationOperation, deve ser implementada para cada processo. A função dessa classe filha é obter todos os dados necessários para realizar a operação. Já o algoritmo que realiza a operação em si (e.g. o autômato celular) é implementado em outra classe (a classe IonImplantation no caso do processo IonImplantationOperation).

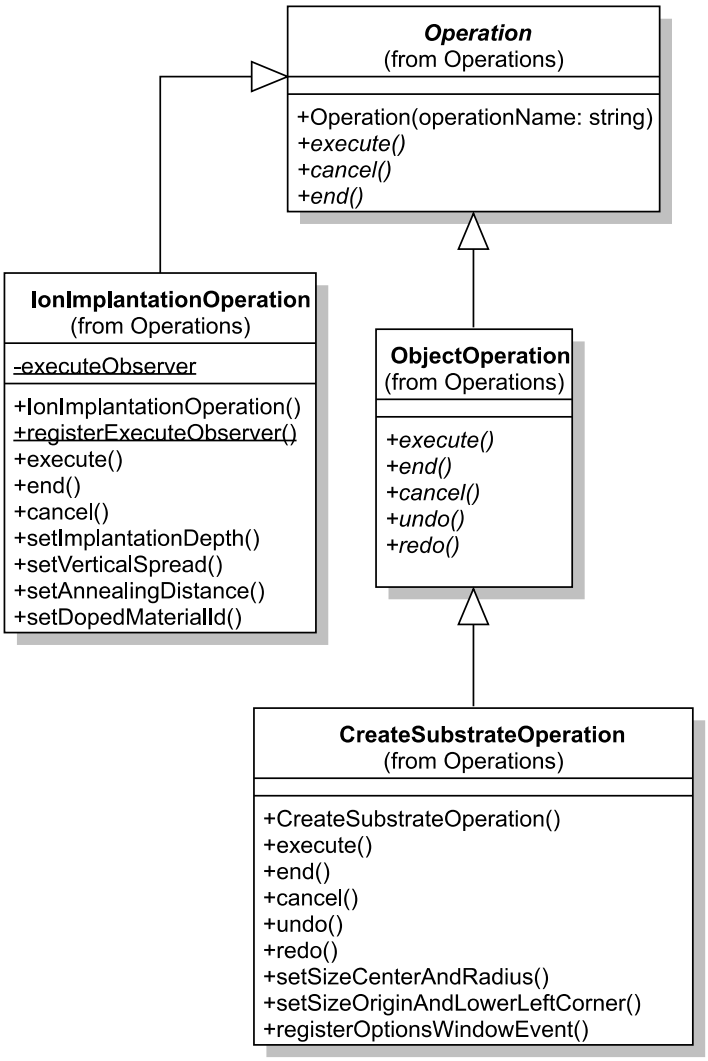


Figura 21- Estrutura usada para implementar as operações.

A classe ObjectOperation, apresentada na Figura 21, é também uma classe filha de Operation. É idêntica com a exceção dos métodos *undo* e *redo*. Processos que possam ser desfeitos ou refeitos devem ser filhos de ObjectOperation e implementar os respectivos métodos. Atualmente a maioria dos processos não podem ser desfeitos. A criação de substratos é uma das exceções, a respectiva classe CreateSubstrateOperation é apresentada no diagrama de classes da Figura 21.

Outra diferença entre as versões anteriores do simMEMS e a atual é em como o usuário entra com as informações necessárias para realizar uma simulação. A maior parte dos dados era fornecida utilizando o mouse, selecionando objetos na tela, ou digitando dados num console e era feita diretamente pela camada Core. Na versão atual do software (a versão 4) a maior parte das entradas de dados pelo usuário é

feita pela interface gráfica. Como os dados são obtidos pela interface gráfica, na camada UI, mas a requisição de dados é feita pela camada Core, é preciso ter uma comunicação tanto no sentido da camada UI para camada Core como no sentido inverso, para indicar quando e que telas devem ser mostradas. A Figura 22 mostra a solução adotada para este problema.

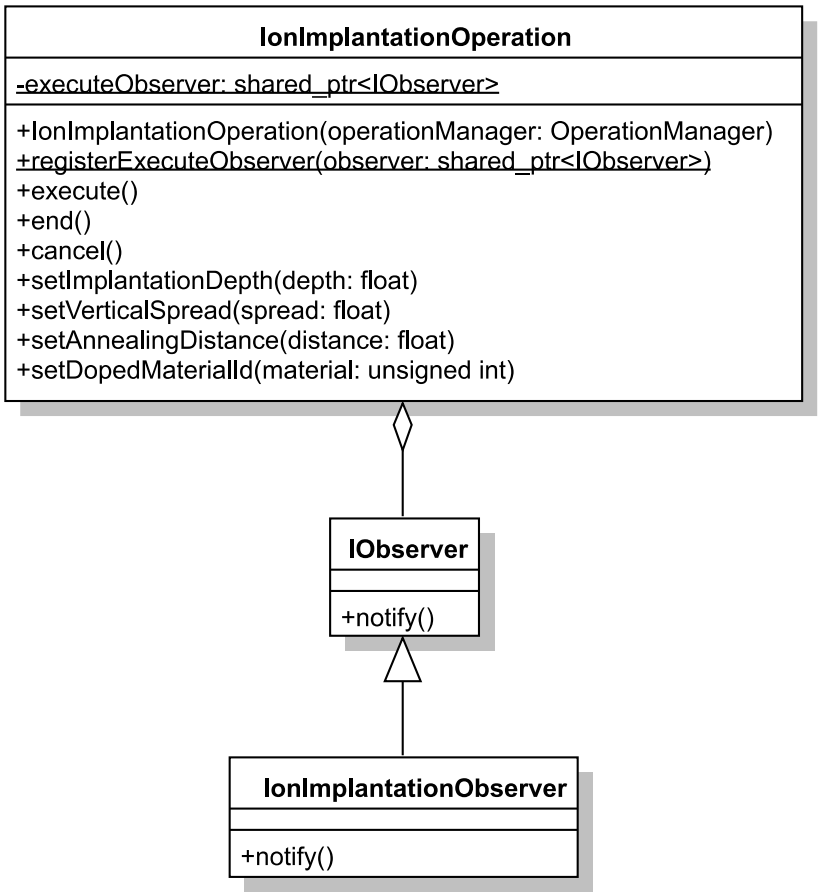


Figura 22- Padrão de projeto de software Observer aplicado para comunicação entre camadas.

Como mostra a Figura 22, o padrão de projeto de software (design pattern) Observer foi utilizado. Padrões de projeto de software são soluções gerais para problemas comuns em determinados contextos. Factories, singletons e observers são exemplos disso. O livro Design Patterns (GAMMA et al., 1994) oferece uma boa descrição de diversos desses padrões de projeto de software. Com o padrão de projeto de software Observer, quando a classe IonImplantationOperation necessita de um dado, chama o

método *notify* da sua instância da classe IonImplantationObserver. Cabe à camada UI registrar uma instância da classe IonImplantationObserver para ser chamada quando algum dado é necessário. A camada UI implementa as interfaces IObserver que são notificadas pela camada Core.

Como mencionado no começo desta seção, a implementação dos algoritmos de simulação dos processos em classes separadas permite separar a aquisição de dados para uma simulação de sua execução. O simMEMS possui também a opção de criar uma lista de processos para realizar as simulações, descrita no item 2.3.3. Nesse modo de execução, todos os dados são obtidos pela camada UI quando o usuário cria a lista. Essa lista pode ser posteriormente salva em arquivo, carregada de um arquivo e executada. Nesse tipo de execução em lista, não é necessário pedir nenhum dado ao usuário, visto que a lista possui os dados necessários. Nesse segundo modo de execução, a camada Core funciona como uma biblioteca de simulação sendo acessada pela camada UI e as chamadas de métodos e funções seguem o sentido usual para uma implementação em camadas.

No que tange à simulação há alguns detalhes de implementação que também merecem menção. A camada Core também utiliza bastante o framework Qt bem como alguns elementos da biblioteca Boost. O framework do Qt foi utilizado principalmente para paralelização dos autômatos celulares. A biblioteca Boost também foi considerada, visto que fornece uma implementação para criar e gerenciar threads semelhante à do novo padrão C++14, à época ainda não aprovado (“ISO International Standard ISO/IEC 14882:2014(E) – Programming Language C++”, 2014). No entanto, a implementação do Qt se mostrou mais amigável.

A biblioteca Qt também fornece uma classe para gerenciar threads, a QThread. Cada instância de QThread representa um thread. Para usar essa classe, o programador cria uma classe filha e uma nova implementação do método *run*. Outra alternativa é usar o sistema de signals e slots do Qt. O programador cria slots e liga aos signals de uma instância da classe QThread. Os slots serão executados no thread do QThread. Nesse caso, cabe ao programador também criar e gerenciar os threads em uso.

O Qt também fornece um thread pool (QThreadPool) que permite executar tarefas simples de maneira paralela reutilizando threads. Nesse caso, o gerenciamento dos threads é feito pelo próprio framework. O thread pool tem uma função que recebe o

ponteiro para o método a ser executado no thread. Para os autômatos celulares, em que uma mesma tarefa simples é executada repetidas vezes sobre diversos dados, essa abordagem é vantajosa visto que o gerenciamento e a reutilização de threads são essenciais para um bom desempenho. Assim, essa foi a abordagem escolhida para o simMEMS.

2.5.1 Implementação dos Processos Simulados

Os processos simulados pelo simMEMS podem ser divididos de maneira genérica em dois tipos: os que sempre modificam todas as células no volume e os que modificam apenas células na superfície. No primeiro caso, é possível ainda diferenciar entre os que são simulados usando autômatos celulares e os que não são. Serão tratados inicialmente os processos que não utilizam autômatos celulares e em seguida serão abordados os processos que utilizam autômatos celulares.

2.5.1.1 Processos Simulados sem Autômatos Celulares

Existem quatro processos que não são simulados por autômatos celulares. A princípio, a aplicação de uma simulação não é necessária ou o uso de autômatos celulares não é indicada. São esses:

1. O processo de criação de um substrato;
2. O processo de planarização (ou polimento);
3. O processo de extração de um filme;
4. O processo de cola de dois substratos.

Todos esses processos têm em comum o fato de que podem ser realizados em um só passo. São, portanto, processos para os quais o uso de autômatos celulares não é interessante.

O processo de criação de um substrato é nada mais do que a criação de um volume de células onde algumas dessas células são ocupados pelo material do substrato. Esse processo pouco mudou desde a versão anterior.

No caso do processo de polimento, o que se deseja fazer, idealmente, é remover todo material da superfície do substrato até uma certa profundidade. Em geral, esse

polimento é feito em equipamentos próprios para isso. Na versão atual, o processo é idealizado. Nesse caso, não há uma verdadeira simulação. O software simplesmente muda o estado de uma faixa de células para vazio. Isso efetivamente remove qualquer material acima ou abaixo de uma região definida pelo usuário.

Os processos de extração e cola de substrato são bastante semelhantes. Nesse caso o processo também é idealizado. Quaisquer dois substratos podem ser colados perfeitamente. Um filme de formato arbitrário também pode ser extraído, o que nem sempre é possível. O processamento em ambos os casos se limita a copiar os dados de interesse para um novo volume de células.

É interessante notar que a independência da ordem de processamento das células é mantida em todos esses casos. Além disso, como o processo é realizado em um único passo, não há preocupação com as bordas das células nem com vizinhanças. Portanto, a execução desses processos pode ser facilmente separada entre os processadores disponíveis. São, portanto, os processos mais fáceis de se implementar.

2.5.1.2 Processos Simulados com Autômatos Celulares para Deslocamento de Superfície

O segundo caso mais simples é quando um autômato celular é utilizado, mas apenas a superfície muda de estado. Processos de corrosão e deposição de filmes em geral se enquadram nesse caso. Esses processos são tratados de forma genérica como o deslocamento de uma superfície, e utilizam os mesmos autômatos celulares.

A principal vantagem é que a simulação de um passo em um substrato de N x N x N passa de complexidade $O(N^3)$ para $O(N^2)$, no caso médio[3]. Essa diminuição na complexidade já torna essa implementação interessante. Além disso, como apenas as células na superfície mudam de estado, é possível também fazer uma otimização no espaço necessário para armazenar os dados de simulação. Na versão 3 do software, duas cópias do volume de células eram mantidas. Uma representava o

[3] A notação O é utilizada para descrever a complexidade temporal de algoritmos. Seja $f(n)$ o tempo de execução de um algoritmo, em segundos, para uma entrada de tamanho n. Pode-se dizer que $f(n)$ é da ordem de $O(g(n))$ se para todo n com exceção de um número finito de valores positivos de n existe uma constante c tal que $f(n) \leq cg(n)$ (JOHNSONBAUGH, 2009).

estado atual, e a outra, o estado futuro. Durante a simulação, as mudanças eram salvas na cópia das células, e depois os estados eram sincronizados. No entanto, é possível armazenar apenas as mudanças que ocorrem, e depois do passo de simulação aplicar essas mudanças, armazenando assim apenas as $O(N^2)$ mudanças a cada passo.

O fluxo da simulação usando um autômato celular na versão 3 do software é apresentado na Figura 23. Uma lista com todas as células na superfície era mantida para cada volume de células. Como a superfície era conhecida, não era necessário determinar as células na superfície antes de rodar os passos do autômato celular. A cada passo de simulação, o estado de todas as células da superfície era copiado da matriz de estados seguintes para a matriz do estado atual. Como as matrizes inicialmente tem o mesmo estado, e apenas as células na superfície podem mudar de estado, isso garante que as duas matrizes de estado mantêm o mesmo valor. A lista de superfície é então atualizada baseado nas mudanças que ocorrem no estado anterior. Por exemplo, para uma deposição, quando uma célula vazia muda de estado, todas as células vazias em sua vizinhança devem ser adicionadas à lista de células na superfície.

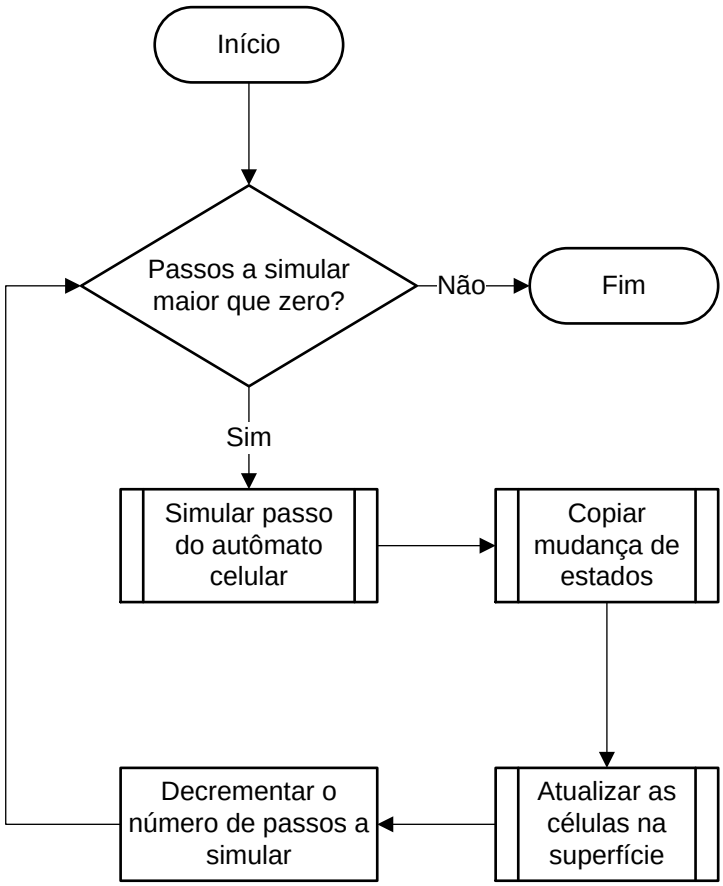


Figura 23 – Fluxograma de uma simulação na versão 3.

Essa abordagem trazia alguns problemas. Primeiramente, quais células pertencem à superfície depende do processo sendo simulado. Tanto um processo de deposição como um processo de corrosão são, genericamente, uma superfície se deslocando. No caso da deposição, a superfície que se desloca é a do filme sendo depositado. Nesse caso, as células de interesse, que devem pertencer à lista de células na superfície, são as células vazias com pelo menos uma célula vizinha não vazia. Já em uma corrosão, a superfície que se desloca é a vazia. Nesse caso, as células de interesse são as preenchidas pelo material a ser corroído que possuem pelo menos uma vizinha vazia. A solução adotada na versão 3 foi armazenar sempre as células vazias, e alterar a lógica para os casos nos quais o oposto é válido. Nessa versão, o processamento não era feito em paralelo.

Na versão atual, diversos processos foram adicionados, sendo simulados com novos autômatos celulares. Alguns desses autômatos celulares não disfrutam das mesmas vantagens de se utilizar uma lista de superfície global, visto que requerem uma inicialização de todas as células do volume antes da simulação. Nesse sentido, uma

nova abordagem foi adotada, que consiste em dividir o volume de células através de cortes no plano YZ, como mostra a . Para simulação paralela, os dados auxiliares de simulação, como a lista de células na superfície, são criados e atualizados independentemente por cada thread.

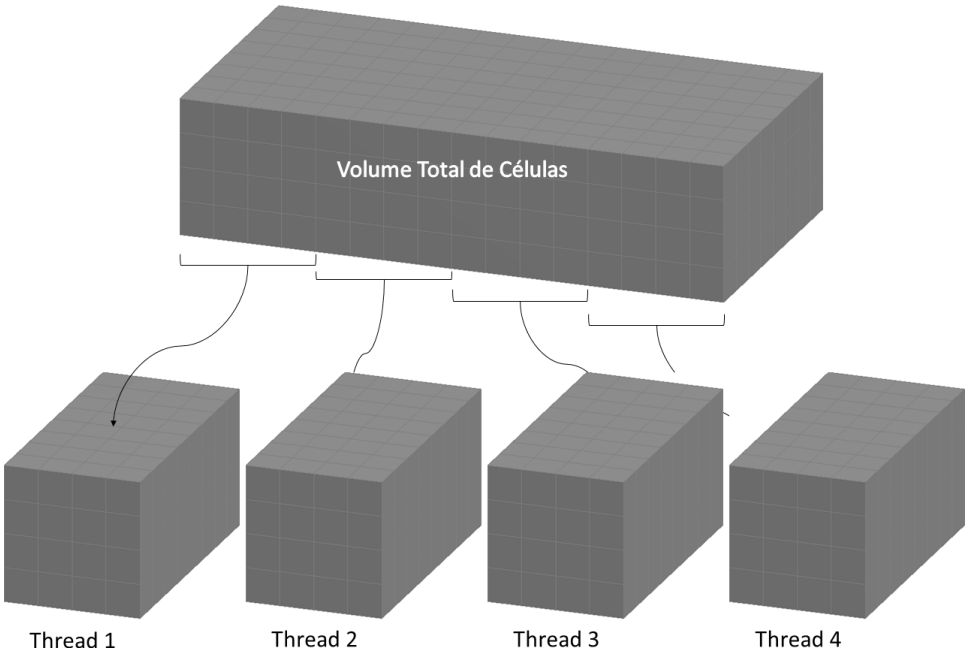


Figura 24 - Separação das células entre os threads.

A escolha do plano YZ é baseada no uso das versões anteriores do software e de outras implementações de autômatos celulares para simulação de processos semelhantes. No caso de simulações de microfabricação em superfície, é comum o usuário seguir a orientação do software e criar um substrato com dimensões X e Y na mesma ordem e um valor bastante menor para a altura Z. O software também pede primeiro a dimensão X, seguida por Y e finalmente Z. Como o comprimento, a maior dimensão, costuma vir primeiro, acaba sendo dado ao valor X. O corte ao longo do plano YZ, portanto, terá a menor área. Já no caso de simulações de microfabricação em substrato, o substrato pode ter um formato que se assemelha mais a um cubo. Nesse caso, a escolha do plano de corte tem pouca variação. Visto o previamente descrito, e como fixar um plano simplifica a implementação, o plano YZ foi escolhido.

O fluxo de simulação agora segue a Figura 25. Inicialmente todas as células são percorridas para inicializar a lista de células na superfície. Além disso, para os autômatos celulares que requerem alguma inicialização nos estados das células, isso é feito. Os passos do autômato são então simulados.

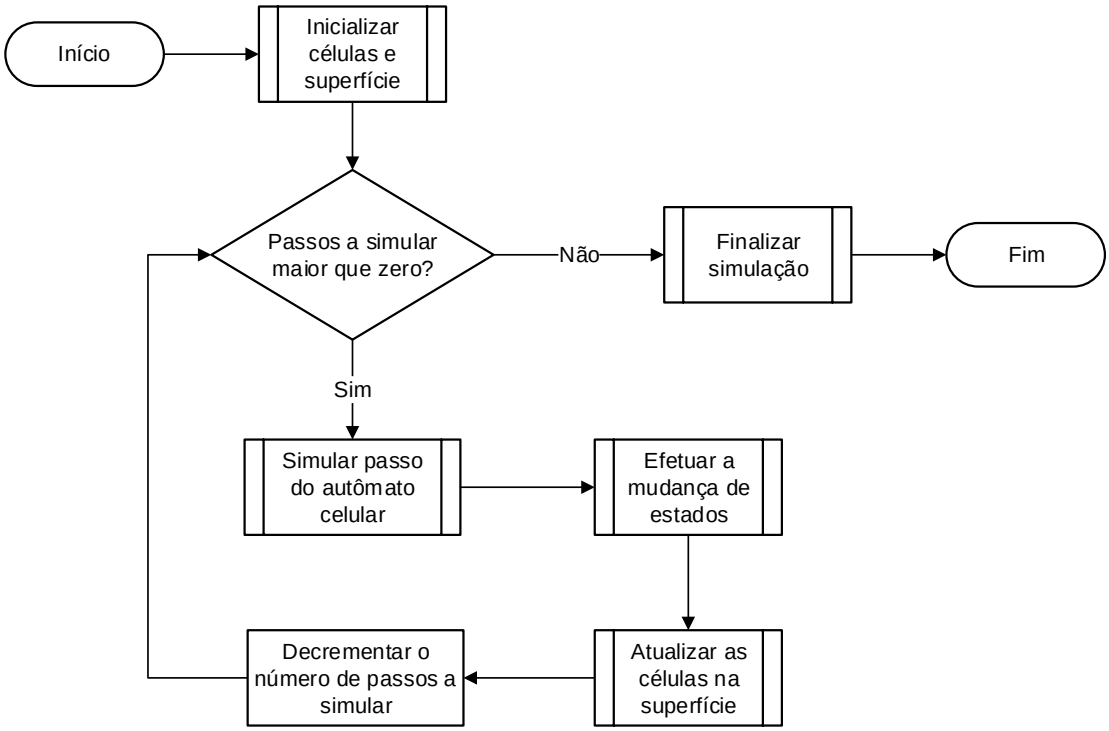


Figura 25 – Fluxograma da simulação na versão 3.

A simulação de um passo consiste em atualizar o estado de todas as células na lista de superfície. O estado seguinte de cada célula é armazenado em uma lista separada; quando o processamento dos blocos vizinhos termina, os estados das células na superfície são atualizados com os dados dessa lista.

É comum que a mudança de estado de uma célula na borda de um bloco adicione à superfície uma célula em um bloco vizinho. Inicialmente todas as células da borda eram consideradas como na superfície para resolver esse problema. No entanto, essa abordagem é ruim, pois o número de células na borda de um bloco pode ser da mesma ordem ou até maior que o número de células efetivamente na superfície de interesse. A solução ideal é adicionar somente as células que efetivamente estiverem na vizinhança de uma célula cujo estado mudou.

Para tanto, cada thread armazena duas listas adicionais de superfície. Quando uma célula do bloco muda de estado, sua vizinhança é analisada. Para cada célula na vizinhança, sua posição relativa ao bloco *i* é analisada (à esquerda do bloco, à direita do bloco, ou no bloco) e é adicionada a uma das três listas. Quando todas as células são simuladas, o thread espera até que os threads vizinhos (*i-1* e *i+1*) terminem os seus respectivos processamentos. Após efetivar as mudanças de estado, a lista com

as células na superfície dentro do bloco *i* é unida com a lista de células na superfície do bloco *i* definidas pelos vizinhos *i-1* e *i+1*, como exemplificado na Figura 26.

Com essa implementação, foi possível diminuir quase pela metade a memória necessária para simular um mesmo volume de células. Além disso, essa abordagem é menos propensa a erros de programação na implementação de novos processos e autômatos celulares.

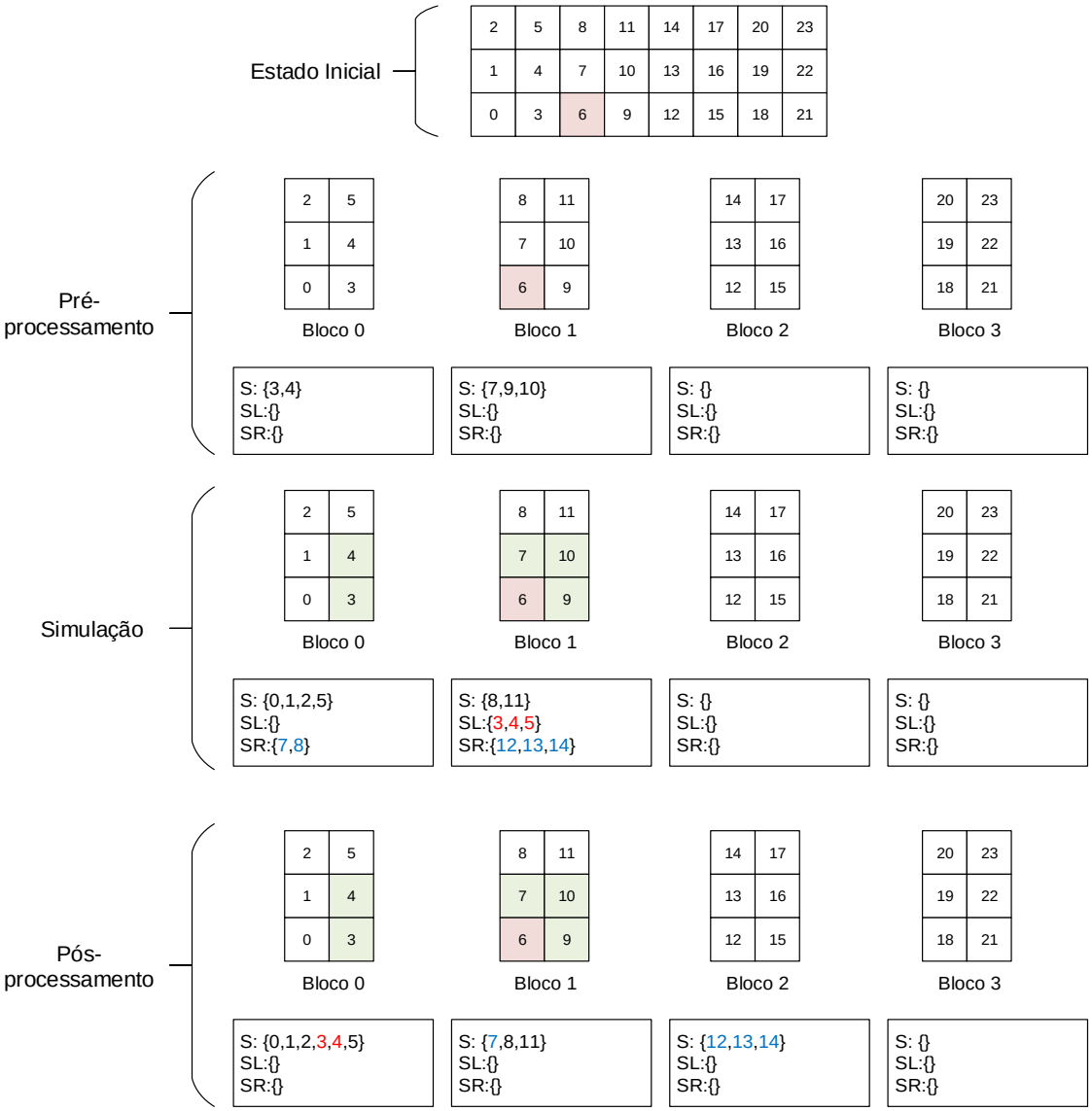


Figura 26 – Exemplo do tratamento de bordas para paralelização.

2.5.2 Processos Simulados com Autômatos Celulares

A versão 4 do simMEMS permite simular a difusão por meio de um autômato celular estocástico. Os detalhes desse autômato celular serão descritos mais adiante, no item

4.3. Diversos processos simulados, como CVD e PVD, são simulados usando o mesmo autômato celular com pequenas modificações nas condições iniciais e do processo. Esse autômato celular é descrito em maiores detalhes no item 3.1. Os demais processos usam autômatos celulares já descritos em trabalho prévio (COLOMBO, 2011).

3 SIMULAÇÃO IDEALIZADA

Nesta seção, serão descritos de forma breve processos simulados pelo software. O objetivo é mostrar os tipos de dispositivos cujos processos de fabricação podem ser simulados pelo software. A versão 3 do software visava principalmente a simulação de processos para fabricação dos aspectos mecânicos de dispositivos MEMS. Na versão atual do simMEMS, além da adição de processos para fabricação de MEMS também foram adicionados processos para permitir a simulação da fabricação de circuitos microeletrônicos.

Atualmente, o software pode simular cerca de 21 processos. Os processos estão disponíveis no menu Simulation. A tabela abaixo descreve brevemente o que cada opção faz.

Tabela 2 – Processos Disponíveis na versão atual.

Item	Opção	Descrição
1	Create Substrate	Cria um substrato. O usuário pode selecionar o material do substrato e seu formato
2	Doping	Simulação qualitativa de uma dopagem térmica em um substrato de Si
3	Random Walk Doping	Simulação de uma dopagem térmica em um substrato de Si
4	Electroplating	Simulação qualitativa do crescimento de um filme por eletrodeposição
5	Ion Implantation	Simulação qualitativa de uma dopagem de Si por implantação iônica
6	PVD	Simulação qualitativa que se aproxima de processos PVD, como evaporação
7	CVD	Simulação qualitativa que se aproxima de processos CVD conformes e isotrópicos
8	Moore Deposition	Deposição para crescimento rápido de filmes onde o formato não é importante

9	Moore Etching	Corrosão rápida de filmes onde o formato não é importante
10	Photolithography	Processo de corrosão seletiva idealizada, semelhante a revelação de fotorresiste
11	Planarization	Processo idealizado de polimento
12	Si Oxidation	Simulação simplificada de um processo de oxidação térmica
13	Si Wet Etching	Simulação simplificada de uma corrosão anisotrópica de Si usando KOH
14	Simple RIE	Simulação idealizada de um processo DRIE
15	Pour Material	Simulação idealizada da deposição que se obtém ao se depositar um material líquido sobre o substrato. Semelhante ao spin coating
16	Anisotropic Deposition	Simulação genérica de um processo de deposição ou crescimento conforme e anisotrópico
17	Anisotropic Etching	Simulação genérica de um processo de corrosão anisotrópico
18	Isotropic Depostion	Simulação genérica de um processo de deposição ou crescimento conforme e isotrópico
19	Isotropic Etching	Simulação genérica de um processo de corrosão isotrópico
20	Extract Film	Processo que gera um novo substrato com o formato de um filme selecionado, como a remoção do material sobre um molde
21	Glue Substrates	Cola de dois substratos, como um processo flip chip

Com exceção dos processos para criar substratos, corrosão anisotrópica de Si, corrosão e deposição de Moore, fotolitografia, extração de filme e cola de substratos, todos os demais processos foram adicionados na versão atual. A maioria desses processos é baseada no mesmo autômato celular, o autômato celular contador de distância proposto no presente trabalho, com diferentes parâmetros ou condições iniciais. O autômato celular contador de distância será descrito a seguir. Em seguida serão descritos, de forma breve, os processos que utilizam este autômato celular.

3.1 Autômato Celular para Evolução Topográfica

Como parte do desenvolvimento da versão 3 do simMEMS, diversos autômatos celulares foram estudados para simular a deposição e corrosão de filmes (COLOMBO, 2011). Embora satisfatórios em diversos aspectos, esses autômatos celulares ainda apresentam limitações e há interesse em melhorar o seu desempenho no que tange à qualidade dos resultados gerados e custo computacional. Neste sentido, foi desenvolvido um novo autômato celular para simular a evolução topográfica de uma superfície.

O autômato celular é baseado na aplicação do princípio de Huygens para simular a evolução da superfície, semelhante ao trabalho de Strasser e Selberherr (STRASSER; SELBERHERR, 1995)**.** Trata-se de um autômato celular contínuo, onde o estado das células é composto por três valores reais, um valor inteiro e um bit. O valor inteiro é utilizado para representar o material que ocupa a célula (por exemplo, Si, Al, SiO_2). Os valores reais representam distâncias e o bit indica se os demais valores podem ou não ser modificados. O autômato utiliza uma vizinhança de von Neumann. Existe uma variável que representa a distância máxima D_{max} e deveria ser armazenada por todas as células. Por eficiência, é implementada como uma variável global.

O sucesso da simulação do autômato celular é dependente da inicialização da rede, que será abordada primeiro. A inicialização é feita em função do processo a ser simulado. Para uma deposição, as células cujo material representa o substrato sobre o qual será depositado o filme têm as distâncias mudadas para 0 e o bit que impede a mudança de estado é definido como verdadeiro. As demais células, que representam o espaço vazio onde o filme pode crescer, tem a distância mudada para o valor máximo e o bit que impede mudanças de estado definido como falso. Para evitar problemas de overflow durante operações matemáticas subsequentes, o valor máximo pode ser o valor da espessura final do filme.

No caso de uma corrosão, a lógica é invertida; células que representam o substrato que pode ser corroído tem a distância mudada para o valor máximo e o bit que impede mudanças no estado definido como falso. As células que representam o espaço onde o material corrosivo se encontra tem a distância mudada para 0 e o bit que impede mudanças de estado definido como verdadeiro.

A evolução temporal dos estados ocorre da mesma maneira em ambos os casos. No caso da deposição, a superfície que evolui é a do filme sendo depositado. Já para a corrosão a superfície que evolui é a do espaço vazio. A regra para evolução temporal dos estados é definida como:

1. Se o bit que impede a mudança das distâncias for verdadeiro, as distâncias, o material e o bit são mantidos.
2. Se o bit que impede a mudança das distâncias for falso, esse valor é mantido. Em seguida:
    a. Para cada célula na vizinhança, determinar a distância $\overrightarrow{D_k} = \overrightarrow{D_o} + \overrightarrow{\delta_k}$, onde $\overrightarrow{D_o}$ é o vetor formado pelas distâncias das células vizinhas e $\overrightarrow{\delta_k}$ é a distância para a célula.
    b. Se o menor valor $\left|\overrightarrow{D_k}\right|$ for menor que o $\left|\vec{D}\right|$, o módulo da distância atual da célula, e que a distância máxima, o valor das distâncias é mudado para $\overrightarrow{D_k}$ e o tipo de material é modificado para o novo valor.

No caso de uma deposição de um filme, o novo valor é o valor correspondente ao filme. Para uma corrosão, o novo valor é o valor que representa o espaço vazio. A Figura 27 mostra um exemplo da execução desse autômato celular para a deposição de um filme. Os estados de uma rede de 7x7 células para os passos T=0,1,2,3 e 4 são apresentados. As células em azul representam o material do substrato. As células verdes representam o material sendo depositado. As células brancas indicam o espaço vazio onde o filme pode crescer. Um filme com espessura máxima $D_{max} = 3$ unidades foi depositado. Os valores entre parênteses representam o vetor distância. O valor acima da distância indica se o estado é ou não fixo (1 e 0, respectivamente).

De acordo com a inicialização descrita previamente, em T=0 as células do substrato têm o bit que impede mudanças verdadeiro e o vetor distância é definido como o vetor nulo. As células vazias, por sua vez, tem o valor máximo da distância, indicado por "(-,-)", e o bit que impede mudanças no estado falso.

O número de passos do autômato que devem ser executados depende da espessura do filme a ser depositado, da dimensão da simulação (2D ou 3D) e do tamanho das células. O simMEMS recebe como entrada a distância máxima D_{max} e a menor

dimensão das células d_{min} e determina automaticamente o número de passos N_p a serem executados através da equação 3.

$$N_p = \left\lceil \frac{D_{max}}{d_{min}} \frac{\left(1+\sqrt{2}\right)}{\sqrt{2}} \right\rceil \tag{3}$$

A equação 3 é válida para três dimensões e superestima o valor necessário para duas dimensões. Essa equação indica o número máximo de passos necessários para, andando apenas nas direções cardinais norte, sul, leste e oeste, atingir uma célula a uma distância D_{max} da origem. No exemplo sob consideração, usando a espessura $D_{max} = 3$ e considerando células quadradas com tamanho $d_{min} = 1$ são necessários pelo menos 4 passos.

0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
1 (0,0)	1 (0,0)	1 (0,0)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
1 (0,0)	1 (0,0)	1 (0,0)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)

T=0

0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (0,1)	0 (0,1)	0 (0,1)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
1 (0,0)	1 (0,0)	1 (0,0)	0 (1,0)	0 (-,-)	0 (-,-)	0 (-,-)
1 (0,0)	1 (0,0)	1 (0,0)	0 (0,1)	0 (0,1)	0 (0,1)	0 (0,1)
1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)

T=1

0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (0,2)	0 (0,2)	0 (0,2)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (0,1)	0 (0,1)	0 (0,1)	0 (1,1)	0 (-,-)	0 (-,-)	0 (-,-)
1 (0,0)	1 (0,0)	1 (0,0)	0 (1,0)	0 (0,2)	0 (0,2)	0 (0,2)
1 (0,0)	1 (0,0)	1 (0,0)	0 (0,1)	0 (0,1)	0 (0,1)	0 (0,1)
1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)

T=2

0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (0,3)	0 (0,3)	0 (0,3)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (0,2)	0 (0,2)	0 (0,2)	0 (1,2)	0 (-,-)	0 (-,-)	0 (-,-)
0 (0,1)	0 (0,1)	0 (0,1)	0 (1,1)	0 (1,2)	0 (0,3)	0 (0,3)
1 (0,0)	1 (0,0)	1 (0,0)	0 (1,0)	0 (0,2)	0 (0,2)	0 (0,2)
1 (0,0)	1 (0,0)	1 (0,0)	0 (0,1)	0 (0,1)	0 (0,1)	0 (0,1)
1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)

T=3

0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (0,3)	0 (0,3)	0 (0,3)	0 (-,-)	0 (-,-)	0 (-,-)	0 (-,-)
0 (0,2)	0 (0,2)	0 (0,2)	0 (1,2)	0 (2,2)	0 (-,-)	0 (-,-)
0 (0,1)	0 (0,1)	0 (0,1)	0 (1,1)	0 (1,2)	0 (0,3)	0 (0,3)
1 (0,0)	1 (0,0)	1 (0,0)	0 (1,0)	0 (0,2)	0 (0,2)	0 (0,2)
1 (0,0)	1 (0,0)	1 (0,0)	0 (0,1)	0 (0,1)	0 (0,1)	0 (0,1)
1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)	1 (0,0)

T=4

Figura 27 - Evolução temporal de uma simulação usando o autômato celular para simular a deposição.

É importante mencionar que nesse autômato celular o número de passos simulados não tem nenhuma relação com a duração do processo sendo simulado. Isto é, o estado do filme no passo T=2 não representa necessariamente o estado do filme na metade do processo de deposição. No presente autômato celular o tempo de deposição é incorporado à simulação quando se faz o cálculo da espessura do filme ou da profundidade da corrosão.

Outro detalhe interessante de se notar é que o valor normalizado do vetor distância é também o vetor normal da superfície do filme em cada célula. Essa característica é interessante visto que a orientação da superfície pode ser utilizada para modelar processos como remissão de partículas durante um processo de deposição ou corrosão (KWON; LEE, 2003).

Modificando as condições iniciais aqui descritas é possível utilizar as mesmas regras para simular, por exemplo, uma deposição não conforme. Também é possível variar a velocidade de evolução da superfície em função da direção através da variação dos valores dos vetores $\overrightarrow{\delta_k}$. A seguir serão descritas algumas condições iniciais e definições para os vetores $\overrightarrow{\delta_k}$ implementadas no simMEMS.

3.2 Corrosão Isotrópica

O processo de corrosão isotrópica utiliza o autômato de contagem de distância. As condições iniciais nesse caso consistem em fixar o vetor distância de todas as células vazias como o vetor nulo. As células preenchidas pelo material que podem ser corroídas permitem a modificação do estado (vetor distância e material), e os componentes do vetor distância são inicialmente definidos como a profundidade da corrosão. Células com outro material tem a distância fixada na profundidade da corrosão.

Com as condições iniciais descritas acima, a corrosão só pode começar a partir de células vazias. Além disso, células que não estejam preenchidas por um material que pode ser corroído podem mudar de estado. Como a corrosão é isotrópica, as dimensões das células são iguais em todas as direções.

A Figura 28 mostra como exemplo o processo de produção de uma microponta em Si pela corrosão isotrópica do Si utilizando um material de mascaramento com formato circular. O volume de células na imagem possui dimensões de 600x600x610 células. O substrato de Si tem 600 células de espessura. Sobre este substrato, uma camada de uma célula de espessura de material de mascaramento é depositado sobre o substrato e fotogravado.

Nas imagens da Figura 28, são apresentados o substrato inicial com o material de mascaramento. Em seguida, quatro simulações de corrosão isotrópica, cada uma usando uma profundidade de 75 células, são realizadas em sequência. O resultado final é a microponta, cujo perfil é apresentado na última imagem.

O resultado da corrosão acima é levemente diferente do que se obteria se uma corrosão com profundidade de 300 células fosse realizada. Isso porque, o resultado de realizar diversas pequenas corrosões não é o mesmo de realizar uma grande corrosão. A diferença fica mais pronunciada quanto menor for o passo, isto é, 10 corrosões de 30 células resultam em uma diferença maior do que 4 corrosões de 75 células em relação a uma única corrosão de 300 células.

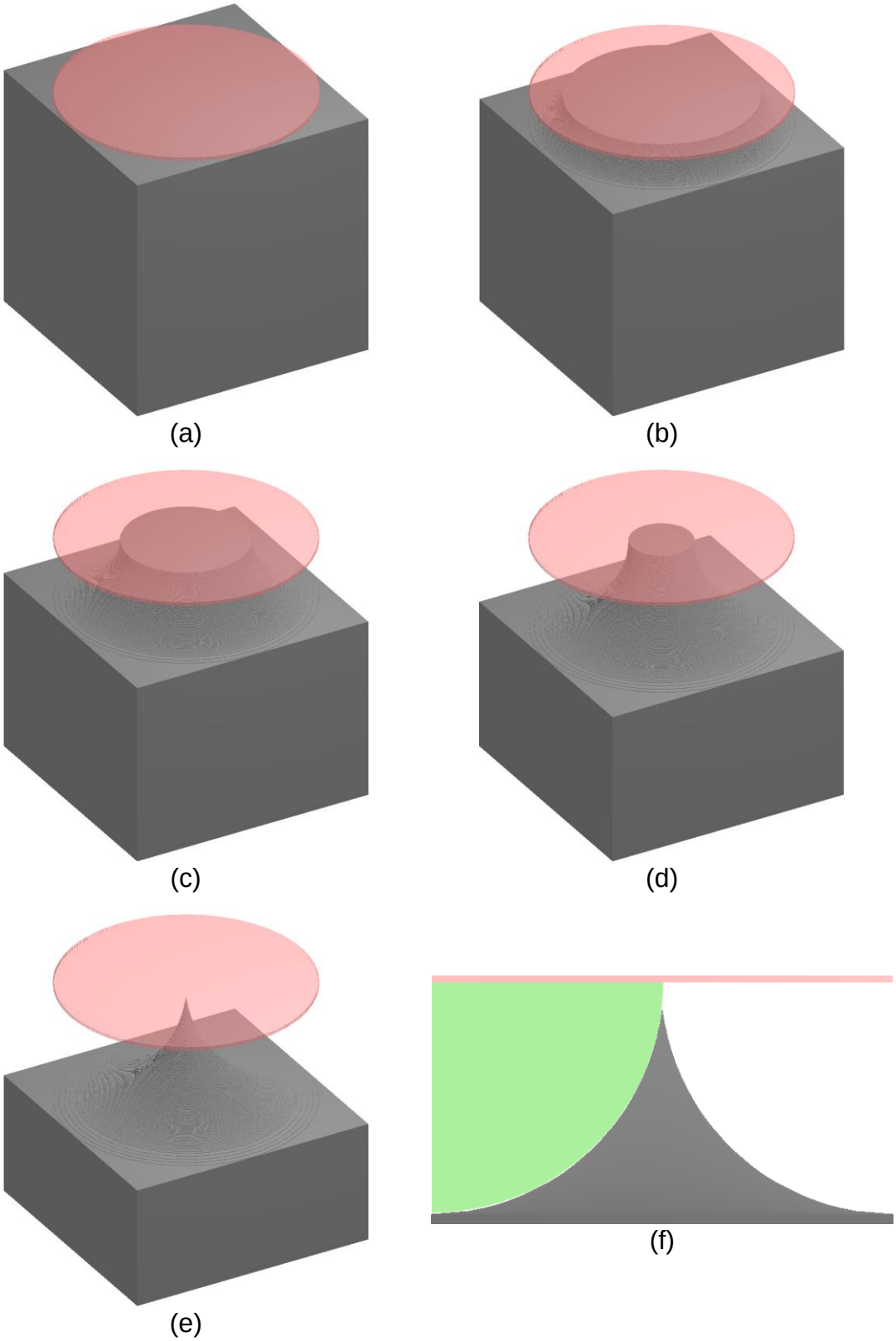
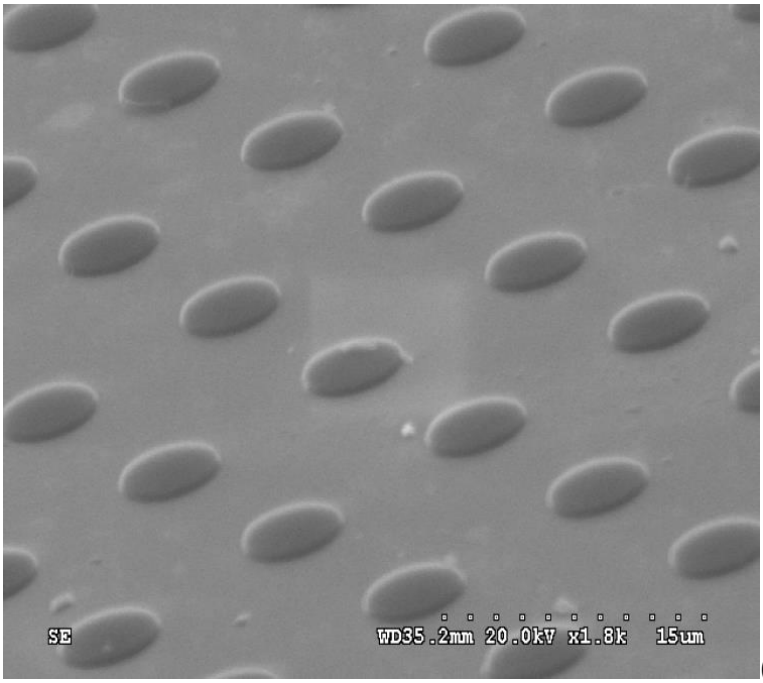
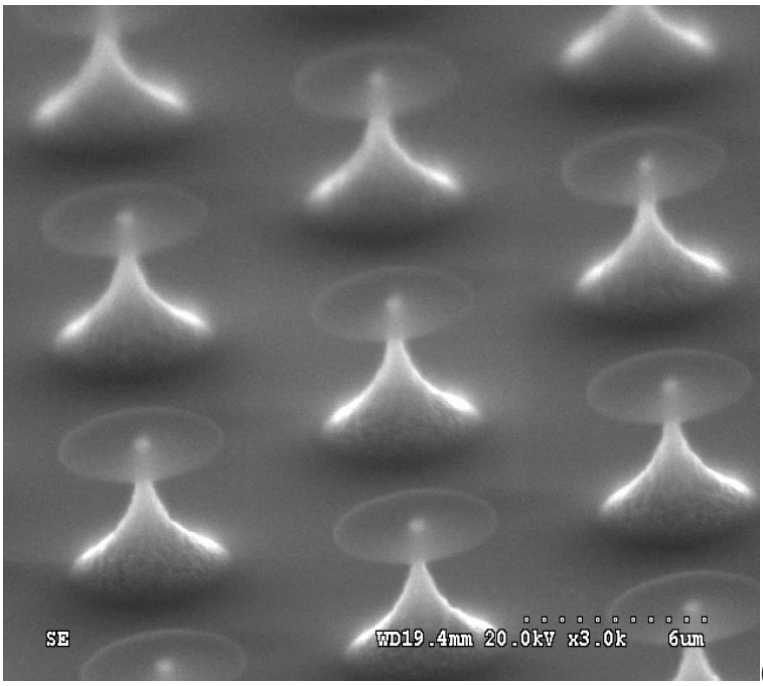


Figura 28 – Corrosão isotrópica de uma microponta em Si simulada em diversos passos. As imagens (a) até (e) mostram corrosões de 75 células de profundidade. A imagem (f) compara o perfil da microponta resultante com o perfil de um quarto de disco. A estrutura tem 600x600x610 células.

Este processo é semelhante ao utilizado para fabricar micropontas. Micropontas têm diversas aplicações, entre elas a microscopia de força atômica. A Figura 29 mostra resultados experimentais de uma corrosão semelhante à apresentada na Figura 28. Os resultados experimentais mostram uma grande semelhança aos simulados pelo simMEMS.



(a)



(b)

Figura 29 - Imagens de um substrato com material de mascaramento fotogravado no formato de círculos (a) e o mesmo substrato após uma corrosão isotrópica (b) (CHEN et al., 2010).

3.3 Deposição Isotrópica

Os processos de CVD e deposição isotrópica na verdade rodam a mesma lógica com as mesmas condições iniciais. A mudança é apenas no nome e é fornecida porque o processo CVD não é a única forma de se obter um filme uniforme.

O autômato celular utilizado é o de contagem da distância. As condições iniciais zeram a distância de todas as células ocupadas. Dessa forma, o filme cresce inicialmente sobre todas as células expostas. As distâncias horizontais e verticais entre células são consideradas iguais, e o crescimento é, portanto, isotrópico. Cabem as mesmas limitações, ligadas ao crescimento sobre cantos vivos. A Figura 30 mostra um exemplo de um filme depositado por este processo.

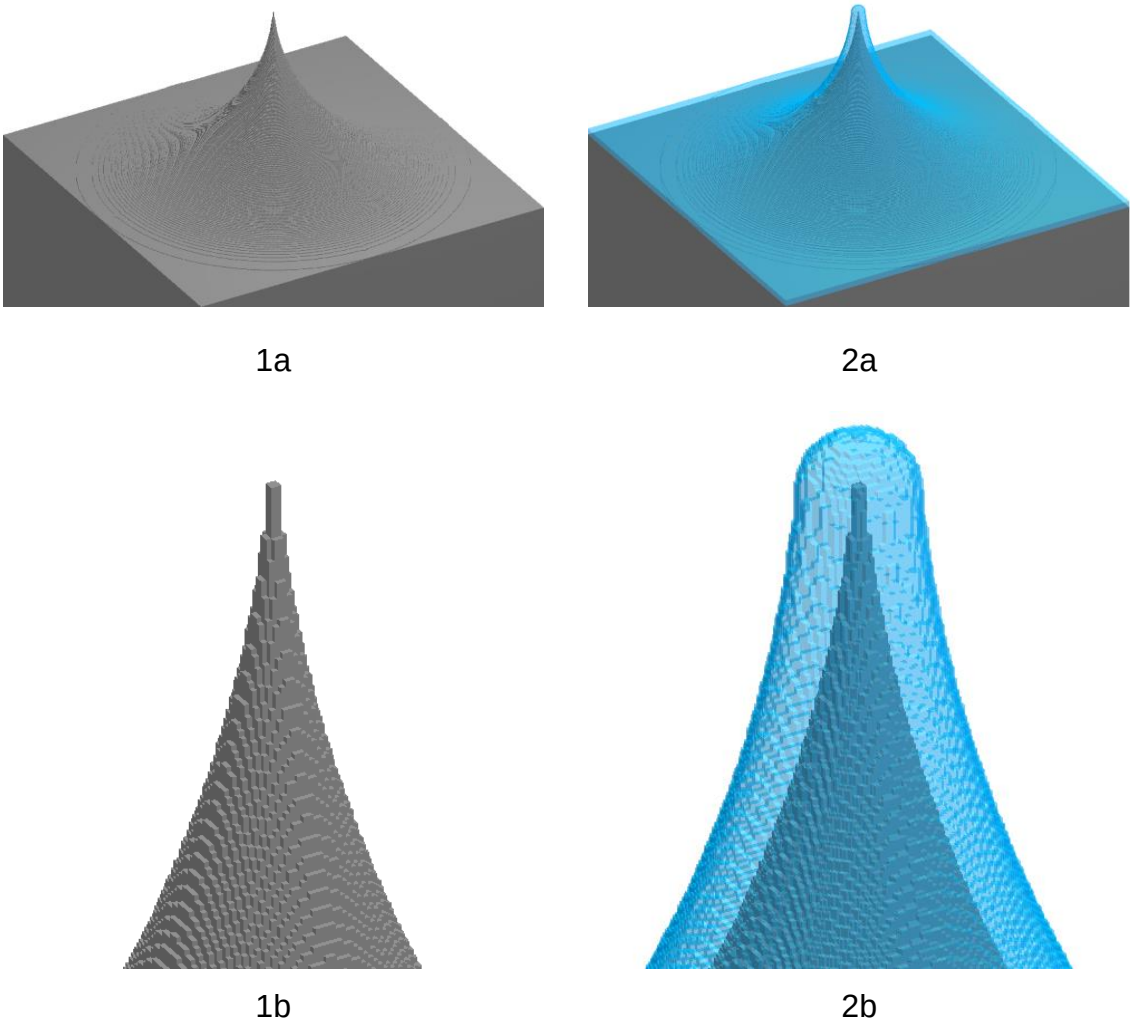


Figura 30 – Exemplo da deposição isotrópica de uma camada de óxido sobre uma microponta de Si.

No exemplo, o processo CVD foi utilizado para depositar uma camada fina de óxido sobre uma microponta de Si. Na Figura 30, as imagens 1a e 2a mostram a microponta antes e depois da deposição, respectivamente. As imagens 1b e 2b da Figura 30 mostram em detalhe o topo da ponta antes e depois da deposição, respectivamente.

3.4 Deposição Anisotrópica

A deposição anisotrópica, assim como o processo CVD, também utiliza o autômato celular de contagem de distância. A condição inicial também considera a distância de todas as células ocupadas como nula, permitindo o crescimento sobre qualquer material.

A diferença é que o usuário tem a possibilidade de definir dimensões diferentes para os tamanhos das células. Quando o usuário define um tamanho maior para uma dimensão, o crescimento nessa direção é mais lento. Dessa forma, é possível ter algum controle sobre o crescimento, fazendo o filme crescer mais no sentido vertical do que horizontal, por exemplo. A Figura 31 mostra um exemplo da deposição anisotrópica de um filme sobre um pedestal.

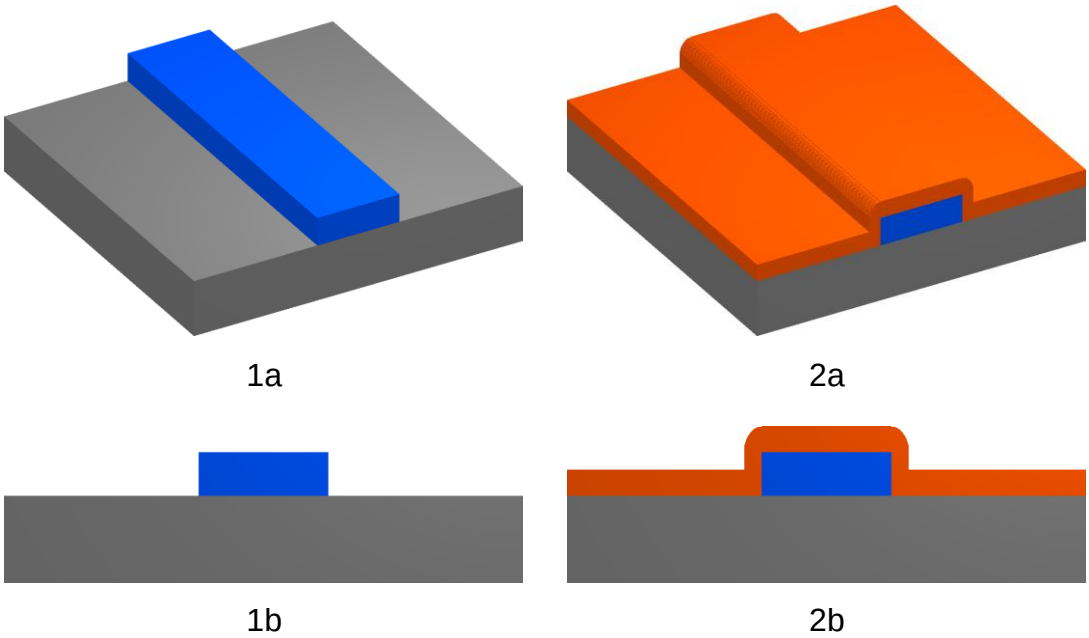


Figura 31 – Exemplo de uma deposição anisotrópica sobre um pedestal. As imagens 1a e 2a mostram o pedestal antes e depois da deposição, respectivamente. As imagens 2a e 2b mostram a vista em perfil antes e depois da deposição, respectivamente.

Na Figura 31, as imagens 1a e 1b mostram de dois ângulos o pedestal antes da deposição do filme. As imagens 2a e 2b mostram o substrato depois da deposição dos mesmos ângulos. A deposição realizada no exemplo foi de um filme com espessura de 30 unidades e coeficiente de anisotropia de 0,7. A Figura 32 mostra em maior detalhe, e com as bordas das células individuais, o resultado visto na imagem 2b da Figura 31. O fato do filme ser menor do que 21 unidades (o esperado) se deve à imprecisão devido ao uso de valores em ponto flutuante para representar as espessuras e o coeficiente de anisotropia. Esse é o erro

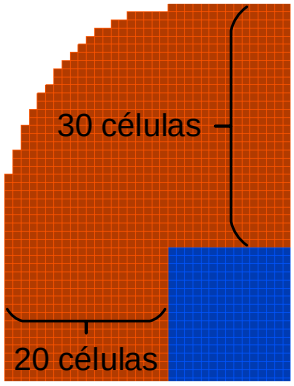
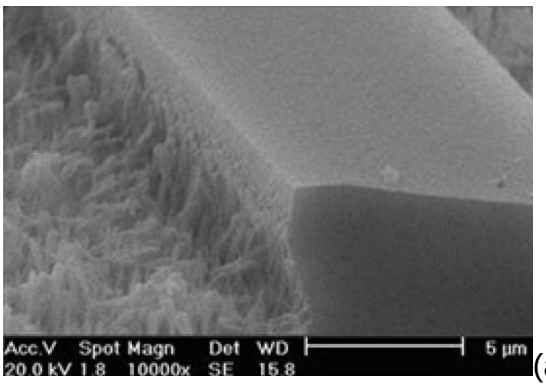
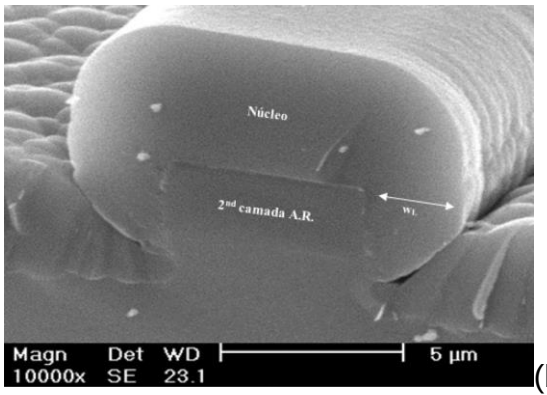


Figura 32 - Comparação entre as espessuras do filme no topo e na lateral do substrato.

A Figura 33 mostra um exemplo de uma deposição de uma camada de óxido de silício por PECVD sobre um pedestal para formar uma guia de onda tipo pedestal (CARVALHO, 2012). As imperfeições na superfície final são resultado das imperfeições do substrato sobre o qual o filme foi depositado. Os resultados se assemelham aos simulados pelo simMEMS na Figura 31.



(a)



(b)

Figura 33 – A figura mostra um pedestal antes (a) e depois (b) da deposição de óxido de Si por PECVD sobre um pedestal (a) (CARVALHO, 2012).

3.5 PVD

O processo de PVD também usa o autômato celular de contagem de distância. Uma deposição PVD pode ser realizada de diversas maneiras, como por evaporação ou por sputtering. Nesse tipo de deposição não ocorre uma reação química; a deposição é realizada através de processos fisícos. No caso da evaporação, por exemplo, uma amostra do material a ser depositado é aquecida até evaporar. O vapor gerado atinge o substrato, onde se solidifica novamente. Este tipo de deposição tem uma direcionalidade grande, visto que as partículas que atingem o substrato todas tem Da mesma maneira que para o processo de deposição anisotrópica, é possível alterar a relação de crescimento lateral para o crescimento vertical.

A diferença se encontra nas condições iniciais. Processos PVD, como a evaporação, tem uma direção preferencial para as partículas que atingem o substrato e formam o filme (MORITZ, 1985). Isso gera um crescimento não conforme; o crescimento em superfícies perpendiculares à superfície do substrato é mais lento.

Para emular esse comportamento, às condições iniciais foram modificadas de forma que apenas células que não tenham um vizinho acima tenham inicialmente distância nula. Dessa forma, o crescimento do filme só pode começar nas superfícies paralelas ao plano do substrato.

É importante notar que o crescimento sobre as células já depositadas não é limitado da mesma maneira. O filme pode crescer normalmente sobre a lateral de células do filme já depositado. De maneira a melhor emular o comportamento de um processo de evaporação é preciso também alterar a taxa de crescimento na direção horizontal.

A Figura 34 exemplifica o resultado de um processo PVD. Em todas as imagens foi depositado um filme de alumínio com 20 células de espessura sobre um substrato de vidro com três estruturas de alturas de 20, 40 e 60 células previamente depositadas.

Os valores abaixo de cada coluna identificam o valor escolhido para a anisotropia no crescimento do filme. No primeiro caso, 0,0, não ocorre nenhum crescimento para os lados. No segundo caso, 0,5, o filme cresce para as laterais com metade da velocidade que cresce sobre as estruturas. Pode-se perceber que nesse caso há a junção do filme sobre a estrutura de 40 unidades com o material depositado sobre o substrato. No caso final, 1,0, não há diferença entre as taxas de crescimento por

direção. Nesse caso, percebe-se que novamente ocorre a junção do filme que cresce sobre o substrato com o filme que cresce sobre as estruturas de 20 e 40 unidades de altura.

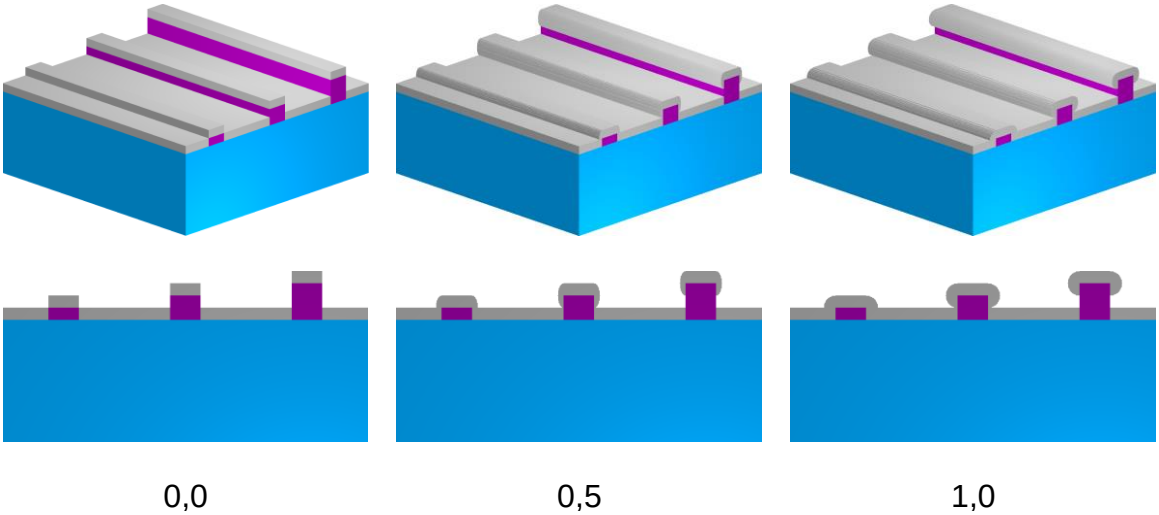


Figura 34 – Exemplos de um processo PVD com espessura sobre estruturas de altura variando de 20, 40 e 60 células. Abaixo das imagens é informado a razão entre as taxas de crescimento lateral e vertical.

Para os pedestais mais altos da Figura 34 o resultado obtido se assemelha ao desejado para um processo de lift-off. Neste processo, uma camada espessa de fotoresiste é fotogravada e sobre essa camada é depositado um filme cujo processo é não conforme. Ao remover o fotoresiste, o material depositado sobre o mesmo é levado junto, restando assim apenas o material depositado sobre o substrato. Esse processo apresenta a vantagem de não necessitar de uma corrosão do filme depositado, que dever ser feita por um processo com boa seletividade para não corroer as demais camadas envolvidas.

A Figura 35 mostra resultados experimentais para um processo de lift-off obtidos por Moritz (1985). Neste trabalho os autores estudam métodos para gerar o perfil de fotoresite visto na Figura 35, que é mais vantajoso para um processo de lift-off. Embora o simMEMS não simule o processo de fotogravação com o nível de detalhamento necessário para obter estruturas como as da Figura 35, fica claro que o processo de lift-off em si pode sim ser simulado com as funcionalidades disponíveis no simMEMS, o que pode ser confirmado pelos resultados da Figura 34.

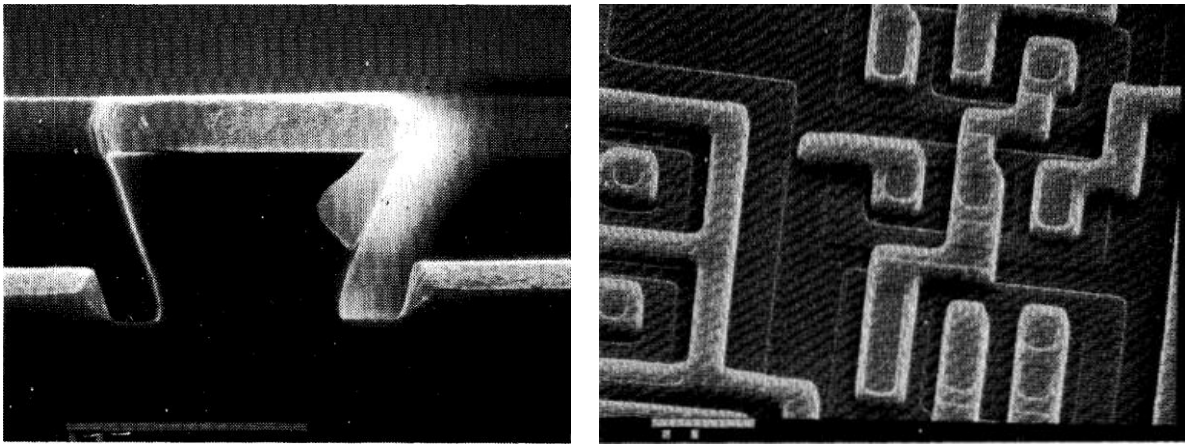


Figura 35 – À esquerda está o resultado da deposição de um filme metálico de 2,3 µm de espessura por evaporação sobre um pedestal de fotoresiste com 7 µm de altura. À direita pode-se ver a primeira camada metálica de um circuito CMOS fabricada usando a técnica de lift-off (MORITZ, 1985).

3.6 Eletrodeposição

O processo de eletrodeposição também utiliza o autômato celular de contagem de distância. Neste caso, o crescimento é isotrópico e pede duas informações ao usuário: qual material será depositado e sobre qual material condutor será feita essa deposição. Assim, o filme eletrodepositado cresce somente sobre áreas expostas do material informado como condutor. Se trata, portanto, de uma simulação simplificada, em que a condutividade elétrica dos materiais não é levada em conta realmente. Tampouco se considera a ligação elétrica do material condutor a alguma fonte de tensão.

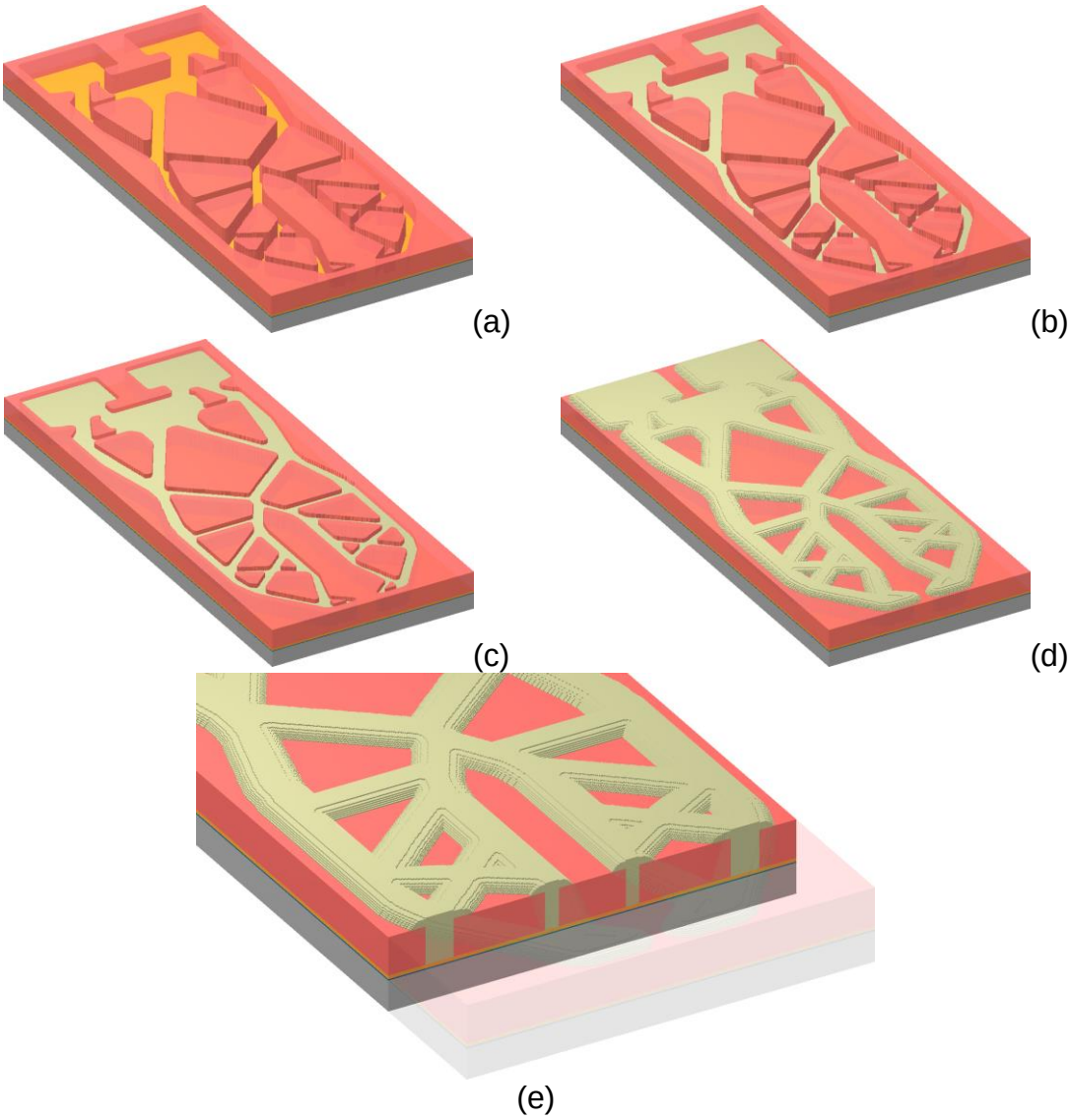


Figura 36 – Exemplo de eletrodeposição de níquel sobre filme condutor de cobre, usando um molde de fotorresiste isolante para formar uma estrutura em forma de micropinça. Observe que a eletrodeposição pode extravasar a espessura do molde.

Para que o processo de eletrodeposição ocorra, as condições inicias são definidas da seguinte maneira. Se uma célula está preenchida com o material definido como condutor, sua distância é fixada em zero. Células preenchidas com outros materiais têm suas respectivas distâncias fixadas na espessura do filme. As células vazias têm as distâncias mutáveis inicializadas para a espessura do filme. Dessa maneira, apenas células vazias vizinhas ao material condutor ou ao filme depositado podem mudar de estado. Um exemplo desse tipo de deposição é mostrado na Figura 36.

Para comparação, a Figura 37 apresenta micropinças fabricadas usando um processo semelhante. Na Figura 37d é possível ver a estrutura sobressalente que se forma quando o filme começa a crescer por sobre o fotoresiste. Esse crescimento pode ser comparado com o resultado na Figura 36e.

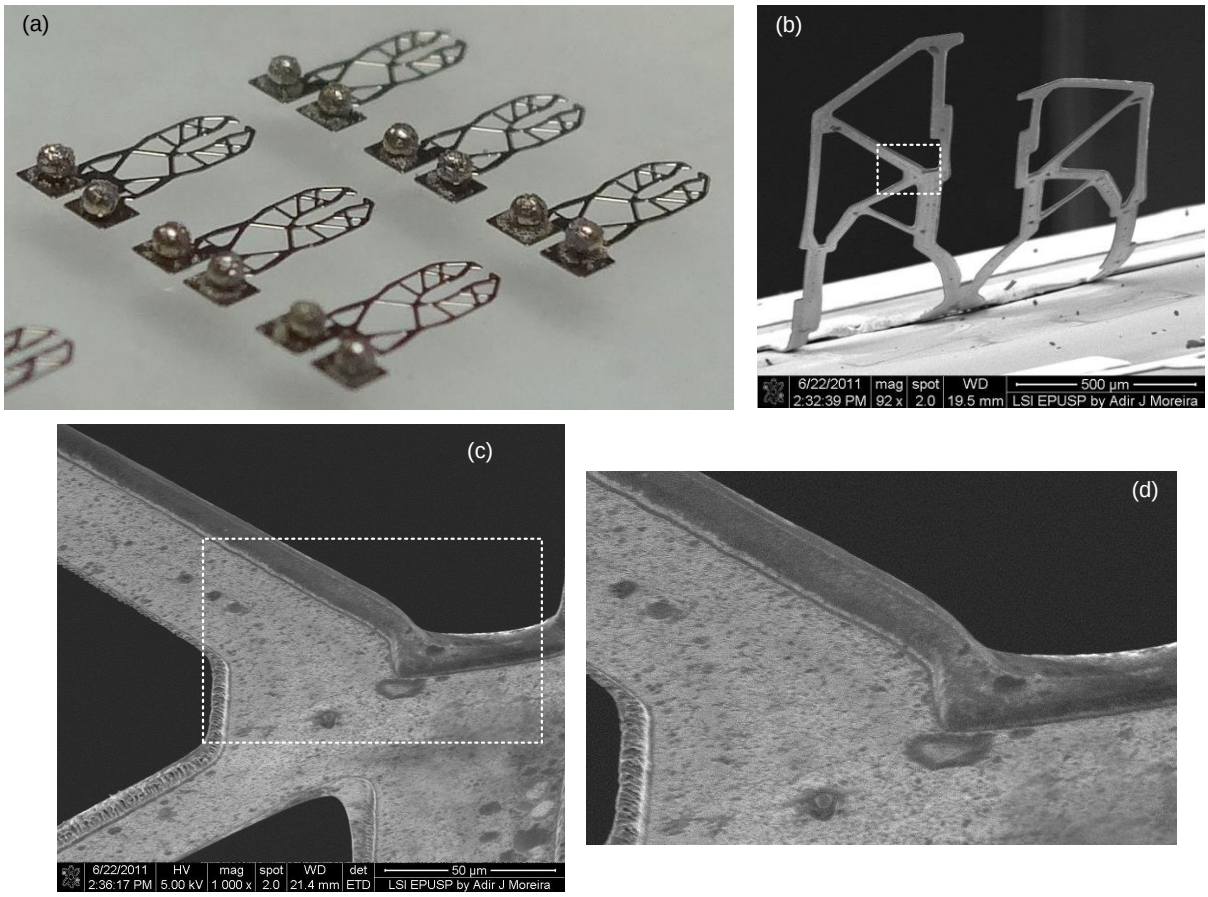


Figura 37 – A imagem (a) mostra uma imagem de microscopia ótica de micropinças fabricadas utilizando eletrodeposição. As imagens (b), (c) e (d) mostram em maior detalhe uma região da pinça usando um MEV. Pode-se ver na imagem (d) o resultado do crescimento do filme metálico por sobre o fotoresiste, formando uma estrutura que se projeta para além da base da pinça.

Cabe aqui ressaltar algumas limitações desse algoritmo que se originam não na simplicidade da simulação, mas são frutos de limitações do autômato celular escolhido. Como descrito previamente, o que o autômato celular faz é determinar, de forma distribuída e paralela, células que se encontram a uma determinada distância de uma superfície. Essa distância é medida de forma direta, não levando em conta obstáculos. O problema se torna bastante claro quando se tem uma deposição como na Figura 38.

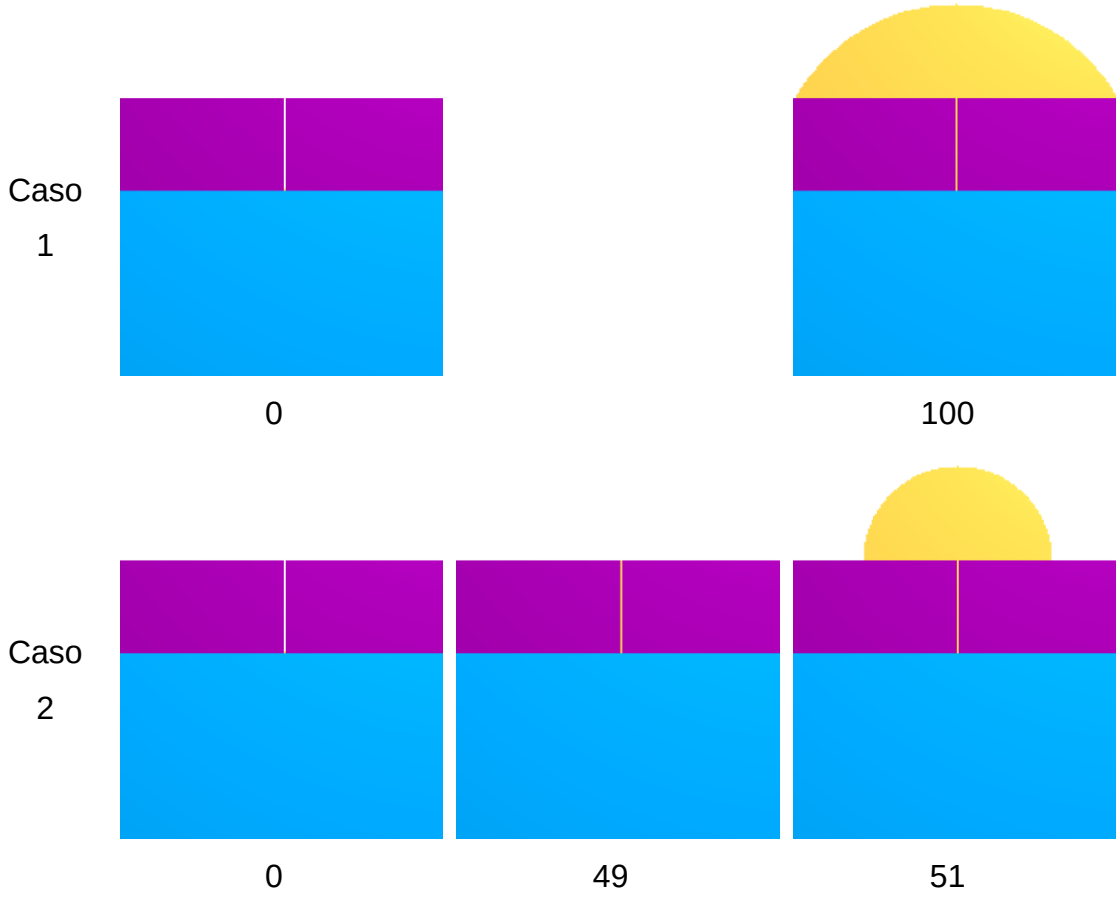


Figura 38 – Exemplo de um processo de eletrodeposição dentro de um pequeno canal feito em uma única etapa com espessura 100 e o mesmo processo em duas etapas com espessuras 49 e 51.

Em ambos os casos, o crescimento se inicia no fundo do canal, que tem 49 unidades de profundidade. As distâncias que compõe os estados das células serão determinadas partindo das células condutoras na base do canal. Enquanto a deposição tiver um valor igual ou inferior à profundidade do canal, o número de passos de simulação utilizados não leva a diferenças no resultado.

Quando o crescimento ultrapassa a profundidade do canal, começam a aparecer as diferenças. O crescimento deveria se dar de forma isotrópica para fora do canal. Assim, se o filme crescer 3 unidades acima do canal, deve também crescer três unidades para a lateral. É trivial ver, no entanto, que isso não ocorre no Caso 1. Isso acontece porque o resultado final da simulação terá preenchido todas as células dentro de um determinado raio da superfície inicial, e esse resultado não é o correto.

Para se obter o resultado isotrópico esperado, o filme deve ser crescido como no Caso 2. O crescimento é realizado em dois passos. No primeiro passo, o canal é inteiramente preenchido. Em seguida o crescimento é resumido até atingir a altura esperada.

3.7 Exemplos de Aplicação

Para exemplificar a aplicação do software, a seguir são apresentados alguns estudos de caso em que a sequência de fabricação exige diversas etapas e técnicas de microfabricação. O primeiro exemplo envolve um circuito com tecnologia complementary metal-oxide-semiconductor (CMOS) com 40 transistores baseado no circuito integrado (CI) CD4001B Texas Instruments (TEXAS INSTRUMENTS, 2016a), um circuito lógico desenvolvido como alternativa à série 7400. O segundo exemplo envolve a tecnologia do Digital Micromirror Device (DMD, também da Texas Instruments (LEE, 2013)) para sistemas de processamento digital de luz (Digital Light Projection ou DLP). Nos dois exemplos, o uso das listas de processos descritas previamente é especialmente importante, dado o grande número etapas necessárias e de parâmetros que devem ser ajustados.

Ao final, são também apresentados dois exemplos relacionados à corrosão anisotrópica do Si cristalino (parte essencial dos processos de Microfabricação em Substrato (KOVACS; MALUF; PETERSEN, 1998; PAL et al., 2009)). Os exemplos em questão envolvem a fabricação de microagulhas de Si e a corrosão de substratos esféricos de Si (GOSÁLVEZ et al., 2011; SATO et al., 1998; SHIKIDA et al., 2004; XING; GOSÁLVEZ; SATO, 2007).

3.7.1 Circuito Integrado CD4001B

A série 7400 de portas lógicas foi introduzida na década de 60 (COMPUTER HISTORY MUSEUM, 2016) e ainda hoje é utilizada e fabricada comercialmente(TEXAS INSTRUMENTS, 2016b). A família 7400 era composta inicialmente de circuitos integrados baseados na tecnologia transistor-transistor logic ou TTL, baseada transistores bipolares(BUIE, 1966). Com o advento da tecnologia CMOS, uma nova série foi introduzida no mercado, a série 4000.

Os CIs da série 4000 implementam funções lógicas semelhantes às oferecidas pela série 7400. Por utilizarem a tecnologia CMOS, no entanto, apresentam vantagens sobre a tecnologia TTL, como menor consumo, maior faixa de tensão de alimentação (3 a 15 V, comparado com 4,75 a 5,25 V para as primeiras famílias da série 7400), e maior simplicidade no design de circuitos. A última é resultado direto da possibilidade de ligar um número muito maior de entradas em uma mesma saída quando se usa portas lógicas com tecnologia CMOS, medida conhecida como fan-out (TOCCI; WIDMER, 2003).

A série 4000 apresenta também desvantagens em relação à tecnologia TTL, em especial sua velocidade significativamente menor, que limita seu uso a aplicações estáticas ou de baixa velocidade. Enquanto as primeiras versões operavam a até 1 MHz, os circuitos bipolares TTL da mesma época poderiam chegar a 10MHz (TEXAS INSTRUMENTS, 2016a, 2016b). O modelo CD4001B pertence à série 4000. Trata-se de um CI com quatro portas lógicas NOR. A tabela da verdade para essa porta lógica é apresentada na Tabela 3.

Tabela 3 – Tabela da verdade para uma porta NOR.

Entrada A	**Entrada B**	**Saída**
1	1	0
1	0	0
0	1	0
0	0	1

O processo de fabricação exato do CI não é disponibilizado pelo fabricante. Um possível processo de fabricação para esse dispositivo foi então desenvolvido com o auxílio do software Microwind (versão 2.3) (SICARD, 2016). O Microwind é um

software bastante utilizado para ensino de projeto de circuitos integrados (AZIZ; SICARD, 2007; AZIZ; SICARD; DHIA, 2010; UYEMURA, 2005). Também é utilizado na literatura para simulação de diversas implementações de circuitos eletrônicos (BELORKAR; BHANDE, 2015; LORENZO; CHAUDHARY, 2013; UMA et al., 2012). Usando o Microwind é possível criar um layout para um circuito microeletrônico e verificar automaticamente se esse layout não segue alguma regra de projeto. Para simplificar a tarefa de projetor um circuito CMOS é possível adotar um conjunto de regras que garantem o bom funcionamento do circuito projetado (BAKER, 2010). Essas regaras incluem detalhes como distâncias entre trilhas metálicas ou vias. Cada fábrica de circuitos, ou *foundry*, possui um conjunto próprio de regras ligadas aos detalhes do seu processo de fabricação. O Microwind permite definir essas regras e verificar se as regras foram seguidas.

O Microwind tem algumas regras para *foundries* com processos de fabricação definidos por detalhes como tamanhos mínimos, espaçamentos, características dos transistores e o número de camadas de metalização. Para este exemplo, foi escolhida uma *foundry* para tecnologia CMOS de 1,2 µm, que era a maior disponível. Essa escolha foi baseada no fato do CI a ser simulado ser bastante antigo e também pelo fato de que uma tecnologia demasiadamente pequena aumentar o consumo de memória para simulação do processo no simMEMS. Projetado o layout, é possível exportar o conjunto de máscaras necessário para fabricação do dispositivo.

Outra funcionalidade interessante do Microwind é a possibilidade de visualizar em 3D o processo de fabricação do dispositivo. O ângulo de visão é fixo, mas o usuário tem controle sobre qual parte do dispositivo será visualizada. O usuário pode navegar por uma lista dos processos para fabricação do dispositivo de acordo com as regras da tecnologia escolhida e visualizar o que ocorre a cada etapa.

O projeto do layout foi baseado no esquemático da porta lógica disponível no datasheet do dispositivo (TEXAS INSTRUMENTS, 2016a). O esquemático é reproduzido na Figura 39, e mostra que a porta lógica possui 10 transistores. Os números ao redor das bolinhas indicam o pino do CI a que o ponto do circuito se conecta. Como o CI tem quatro portas idênticas, os pinos para as demais portas são informados em parênteses abaixo das bolinhas. Os asteriscos marcam os dois pinos de entrada da porta lógica. O datasheet também indica como é o circuito de proteção

contra descargas elétricas, mas essa parte do circuito foi omitida para simplificar o projeto.

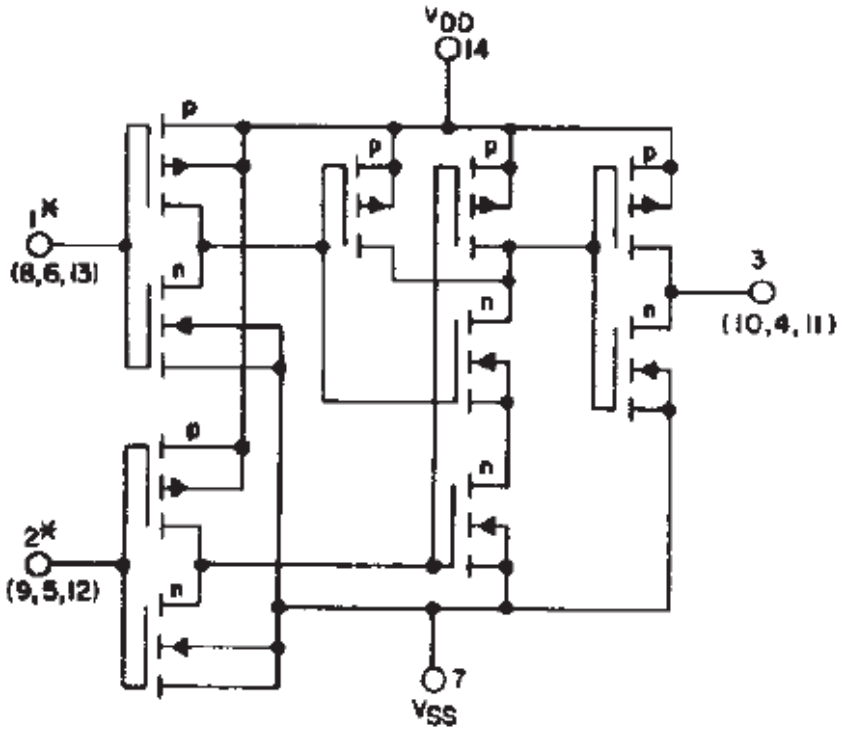


Figura 39 – Esquemático de uma porta lógica NOR no CD4001B (TEXAS INSTRUMENTS, 2016a).

Devido às limitações no tamanho do dispositivo que pode ser simulado em memória no simMEMS, é importante escolher um tamanho de célula que permita simular bem o processo, mas que não seja demasiadamente grande a ponto de tornar a simulação inviável. Como uma tecnologia de 1,2 microns foi utilizada nesse exemplo, valor de λ (o menor espaçamento possível) é de 0,6 microns. Optou-se então por fazer transistores com canal de 3 microns, um múltiplo inteiro do valor de λ do processo. O tamanho final do circuito foi limitado pelo tamanho dos pads de contato e ficou em aproximadamente 860x800 μm^2. A tecnologia escolhida utiliza duas camadas de metal, que é suficiente para implementar o CI escolhido. A Figura 40 mostra o layout final do dispositivo no Microwind, incluindo os pads para ligação posterior dos pinos do encapsulamento ao circuito integrado.

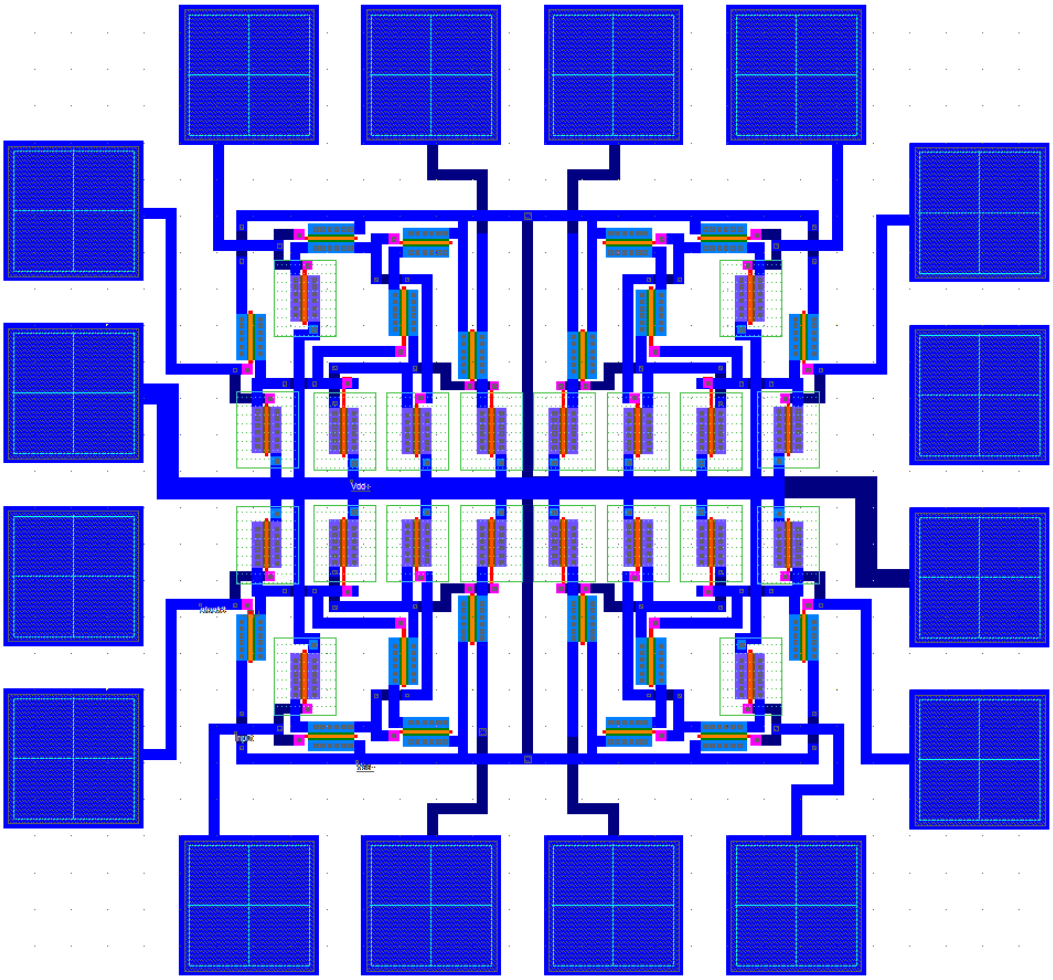
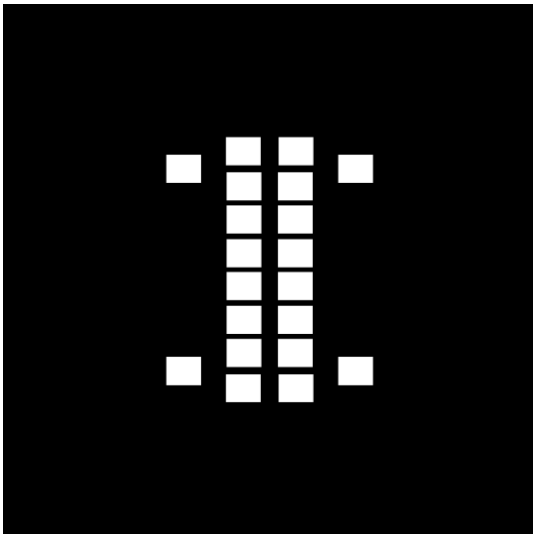
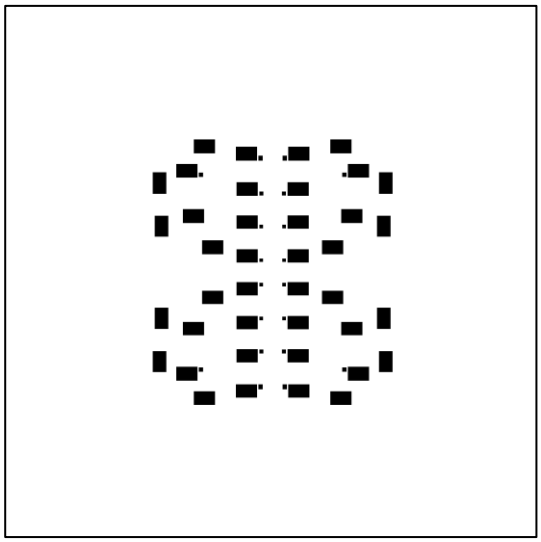


Figura 40 – Layout final das quatro portas lógicas. Cada porta lógica ocupa um quarto da área do chip, sendo simétricas e espelhadas entre si. Os contatos seguem o mesmo layout definido no datasheet do CD4001B.

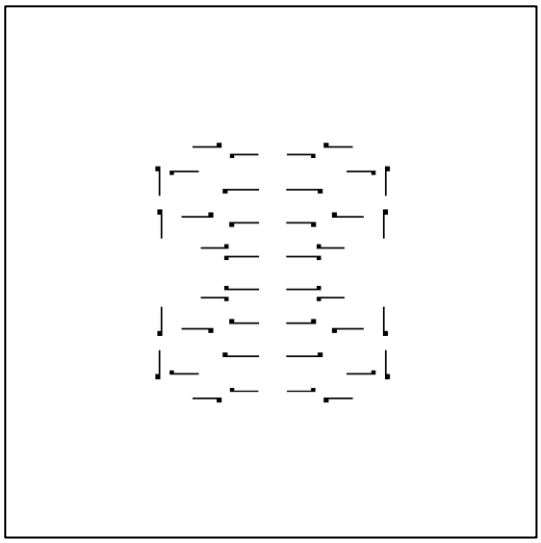
Após a criação do layout da Figura 40, o conjunto de máscaras para fabricação do dispositivo foi gerado e exportado pelo Microwind. Com a ajuda do software AutoCAD, as máscaras geradas foram transformadas em imagens para uso como máscaras no simMEMS. Para simulação do circuito completo foi adotado um tamanho de célula de 0,6 µm, levando a máscaras de 1600x1600 pixels (as máscaras têm um espaço adicional ao redor dos contatos) que são apresentadas na Figura 41. O processo de conversão das máscaras e escolha do tamanho de célula é descrito no item 3.7.1.1.



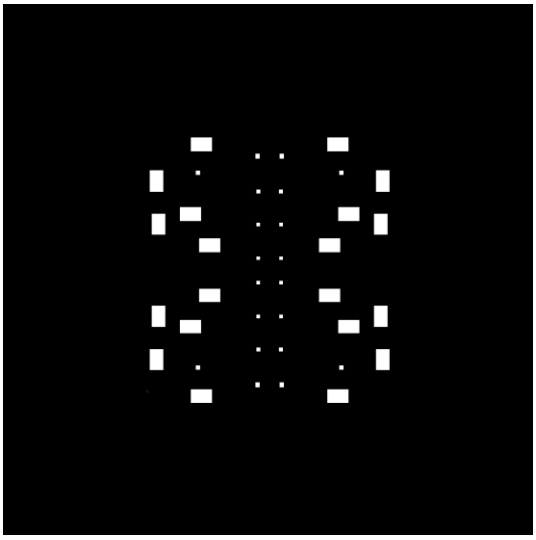
1-Máscara para formar o poço N



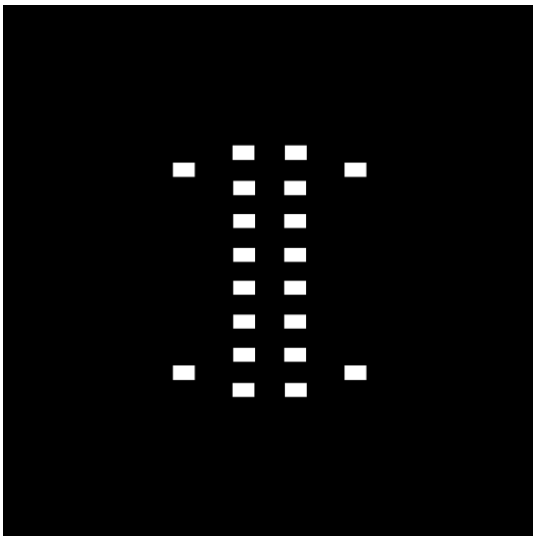
2-Mácara para formar os trenches



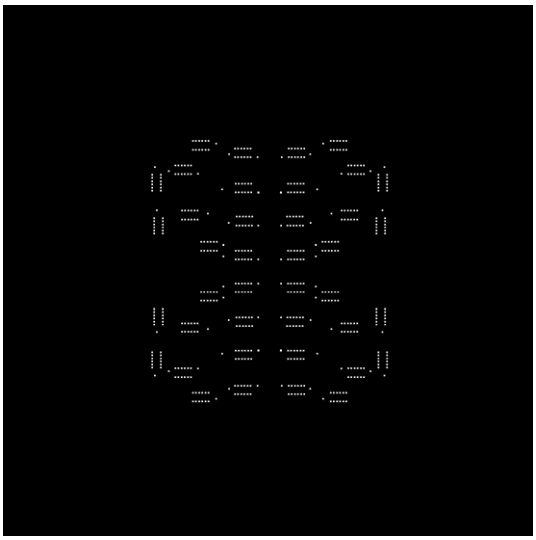
3-Máscara para formar os gates



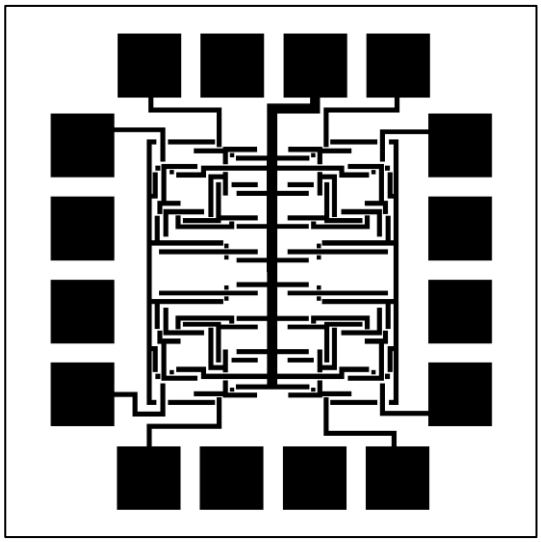
4-Máscara para dopagem N+



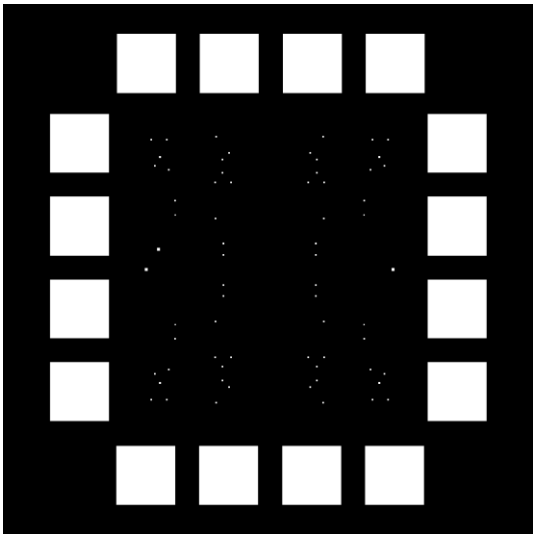
5-Máscara para dopagem P+



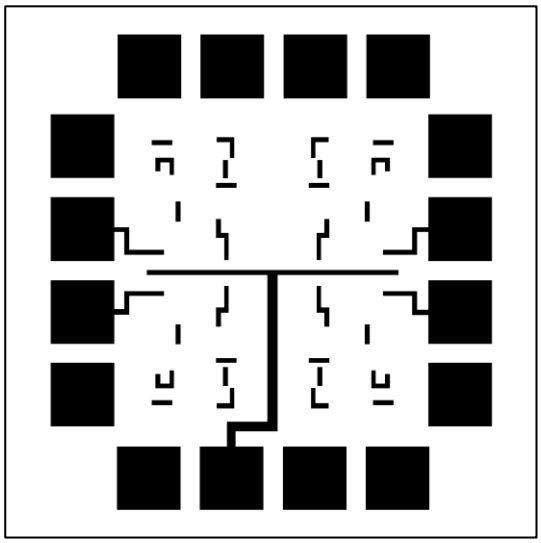
6-Máscara para contatos



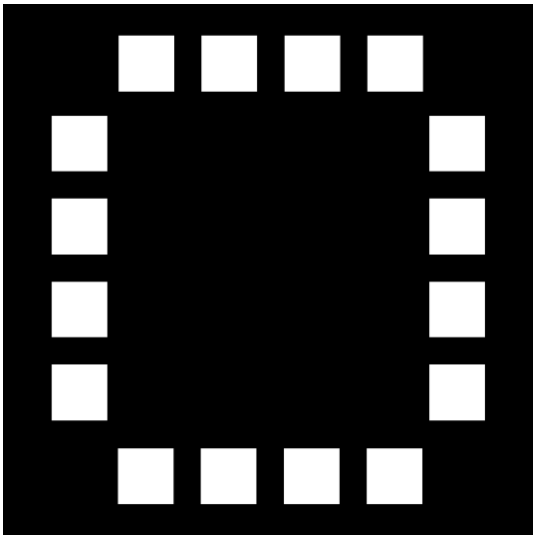
7-Máscara da primeria camada metálica



8-Máscara das vias



9-Máscara da segunda camada metálica



10-Máscara para abrir acesso aos pads

Figura 41 – Máscaras geradas para simular a fabricação do CI no simMEMS.

Um detalhe adicional do Microwind é que este software permite gerar uma netlist para simulação do elétrica do circuito projetado no SPICE, um software para simulação de circuitos elétricos. Para realizar um teste básico do funcionamento do circuito projetado e garantir que o mesmo cumpre sua função de porta NOR, o netlist do circuito mostrado na Figura 40 foi exportado e posteriormente simulado no NGSpice. Algumas pequenas modificações foram feitas no netlist antes da simulação. Primeiro, os pads não utilizados geram alguns elementos não conectados que foram removidos. As fontes de excitação também foram adicionadas, embora o Microwind permita fazer isso antes de exportar o circuito. Na Figura 42 são apresentados os resultados da

análise transiente de uma das portas lógicas, localizada no canto inferior esquerdo do layout. O comportamento das demais portas também foi simulado, mas como todas apresentaram comportamentos idênticos os demais resultados foram omitidos.

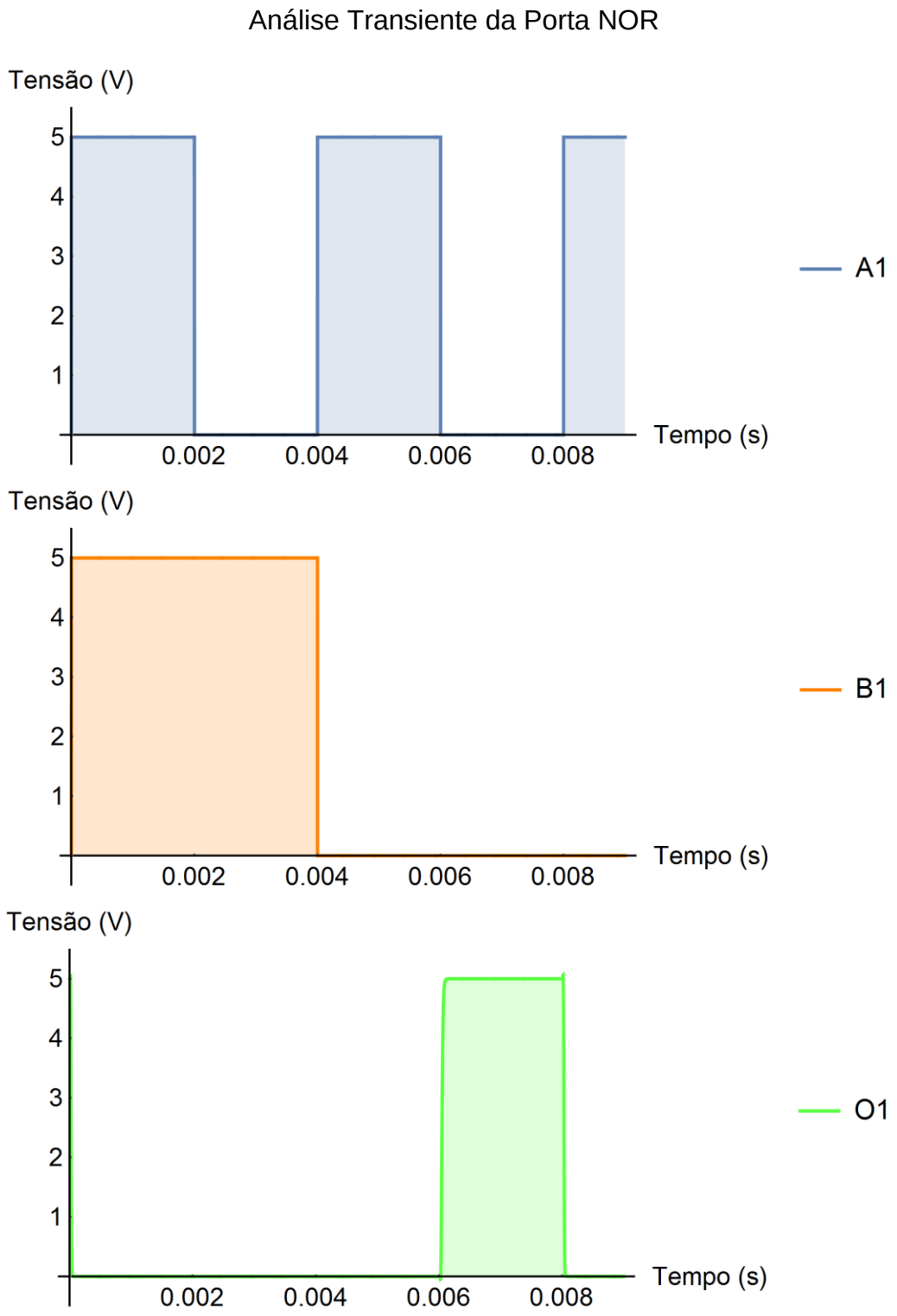


Figura 42 – Resultado da análise transiente de uma das portas lógicas. As entradas A1 e B1 foram excitadas com trens de pulsos. O1 mostra a tensão na saída da porta lógica.

Na Figura 42, os pulsos nas entradas seguem o comportamento descrito na Tabela 3. Pelos resultados apresentados, fica claro que a porta lógica se comporta efetivamente como uma porta NOR. Pelos valores de tempo na escala também fica claro que a porta lógica projetada não é muito rápida.

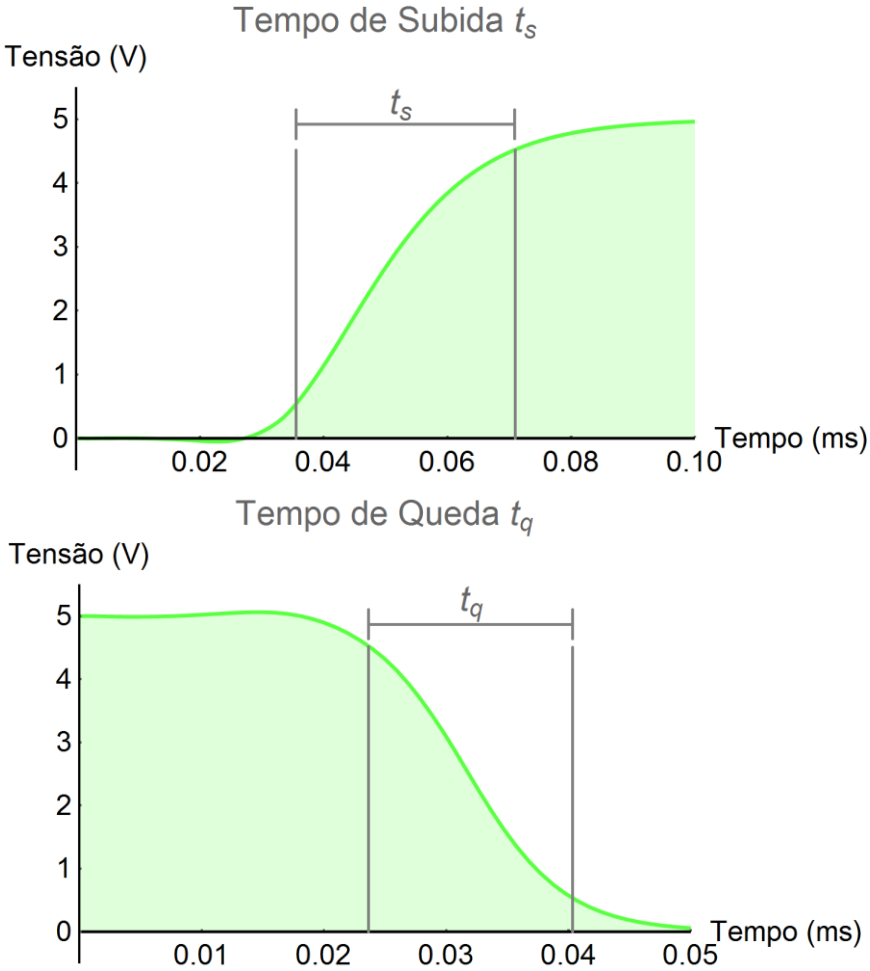


Figura 43 – Gráficos para medida dos tempos de subida e de descida para a porta lógica. Apenas uma porta é apresentada pois as quatro apresentam comportamento idêntico.

A Figura 43 mostra a medida dos tempos de subida e de descida do sinal. Os dois valores são razoavelmente altos, aproximadamente 35,5 µs e 16,7 µs para o tempo de subida e de descida, respectivamente. Desse modo, a frequência máxima de entrada é de aproximadamente 19,2 KHz, que é bastante baixa para um dispositivo desse tipo. Isso ocorre porque os transistores são relativamente grandes (cerca de 25x34 μm^2), o que aumenta as capacitâncias parasitárias e os tempos de subida e descida. As definições de tempo de subida e descida bem como a fórmula para o cálculo da frequência máxima podem ser encontradas no livro Sistemas Digitais (TOCCI; WIDMER, 2003).

No que diz respeito à fabricação, o Microwind lista a sequência de etapas necessária para fabricação do dispositivo usando uma dada tecnologia (em nosso caso, CMOS de 1,2 µm). Essa lista pode ser visualizada na tela do programa junto com o aspecto geral em 3D do circuito em cada etapa. Essa tela foi reproduzida na Figura 44. Como pode-se ver, para a tecnologia escolhida o Microwind utiliza 15 etapas começando com o substrato inicial. Como será visto a seguir, isto representa uma grande simplificação do processo de fabricação.

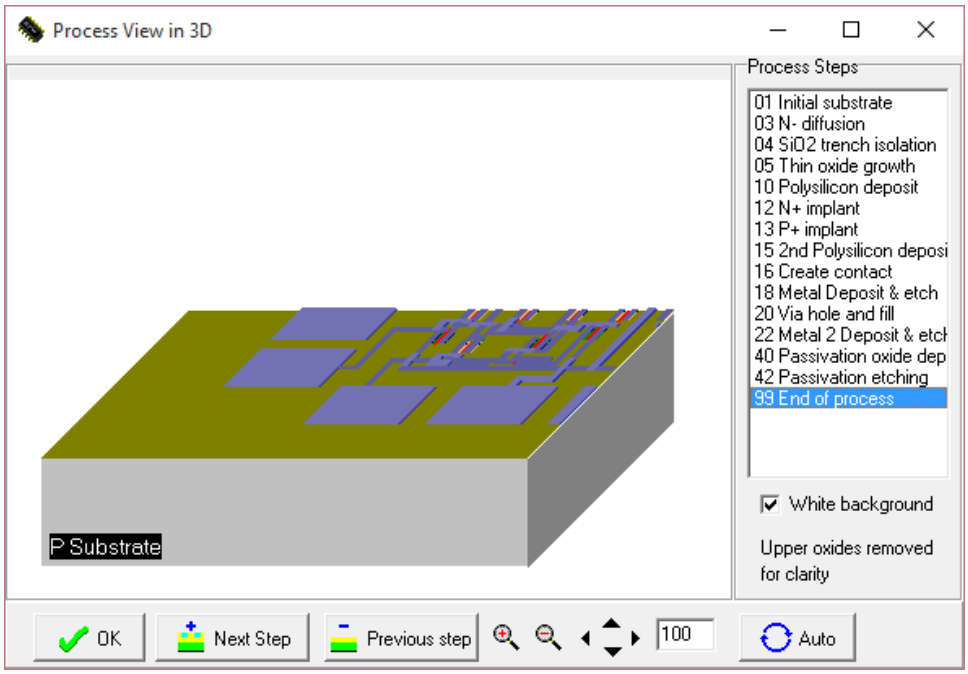


Figura 44 – Visão tridimensional parcial do circuito fabricado a partir do layout criado no Microwind.

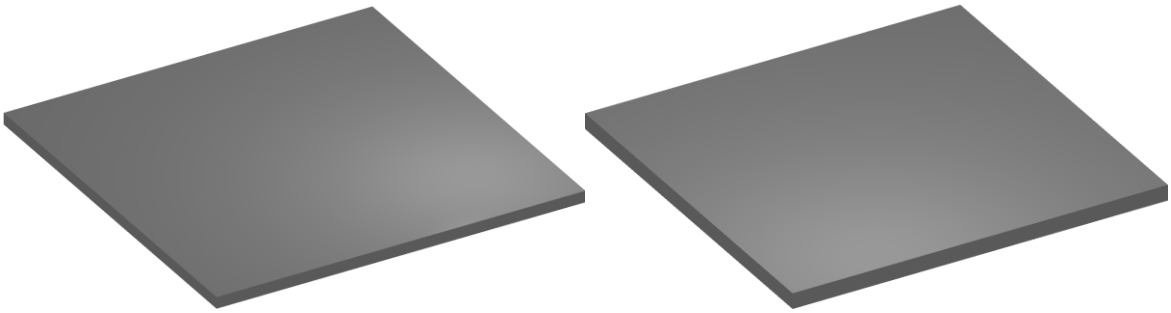
A partir dessa lista básica de etapas no Microwind, foi montada uma lista de processos a ser usada dentro do simMEMS. Porém, processos que no Microwind são indicados como uma única etapa, envolvem na verdade várias etapas que, por sua vez, são indicadas individualmente no simMEMS. Por exemplo, o processo de difusão N, que diz respeito à criação dos poços N no substrato tipo P, é indicado como uma única etapa no Microwind, mas no simMEMS corresponde a sete etapas, a saber:

1. O crescimento de uma camada de óxido térmico, que servirá como material de mascaramento para a dopagem;
2. A deposição do fotorresiste;
3. A revelação do fotorresiste, usando a máscara apropriada;

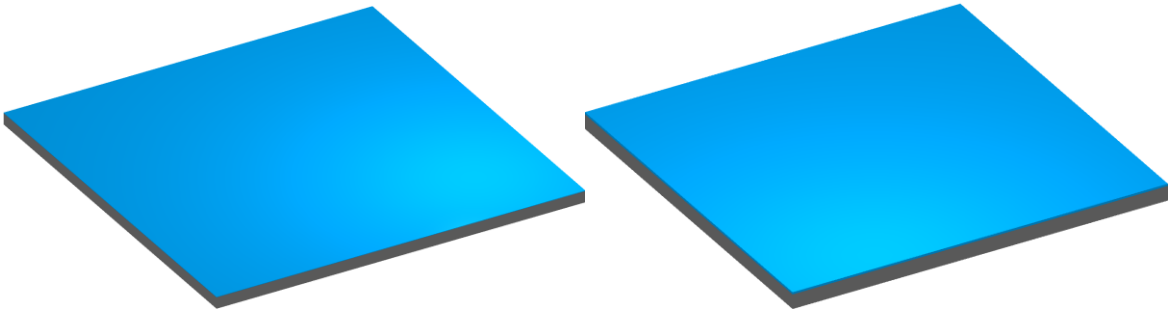
4. A corrosão da camada de óxido, usando o fotorresiste revelado como material de mascaramento;
5. A remoção do fotorresiste;
6. A dopagem térmica do substrato;
7. A remoção pelo processo de polimento da camada de óxido térmico.

A simulação de todas as 15 etapas apresentadas pelo Microwind necessita de 58 etapas no simMEMS, o que evidencia que o nível de detalhamento apresentado pelo simMEMS é bem maior do que o apresentado pelo Microwind. Por outro lado, é importante mencionar também que os processos no simMEMS são simplificados. Por exemplo, o processo de deposição e revelação do fotorresiste, inclui algumas etapas de cozimento do fotorresiste que por simplicidade são omitidas no simMEMS, mas que não deixam de ser essenciais para o processo real.

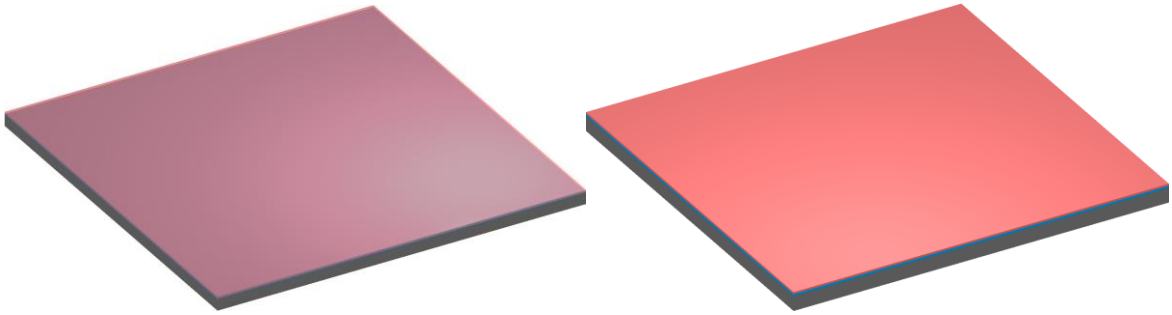
A Figura 45 apresenta todos os 58 processos e uma breve descrição do que está ocorrendo. A seguir os processos serão melhor descritos. Também serão apresentadas as justificativas para escolha de alguns processos. A Figura 45-1 mostra apenas o substrato inicial, nesse caso um substrato tipo P, com dimensões de 1600x1600x50 células. Para dar conta do crescimento dos diversos filmes sobre o substrato de Si, o volume de células para simulação é um pouco maior no eixo Z, resultando em um total de 1600x1600x80 células, ocupando aproximadamente 8 GB, incluindo os dados para desenhar a superfície. A simulação da fabricação completa do CI, salvando uma imagem após cada etapa, demora aproximadamente 23 minutos e leva cerca de 9 minutos se as imagens não forem salvas.



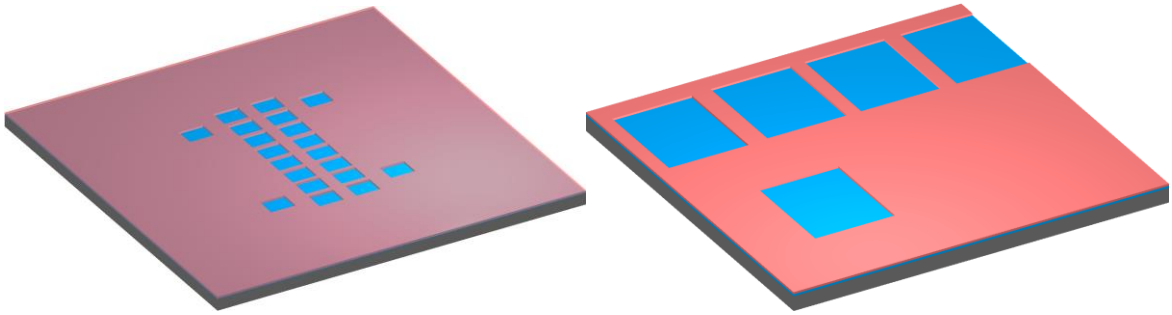
1-Substrato inicial



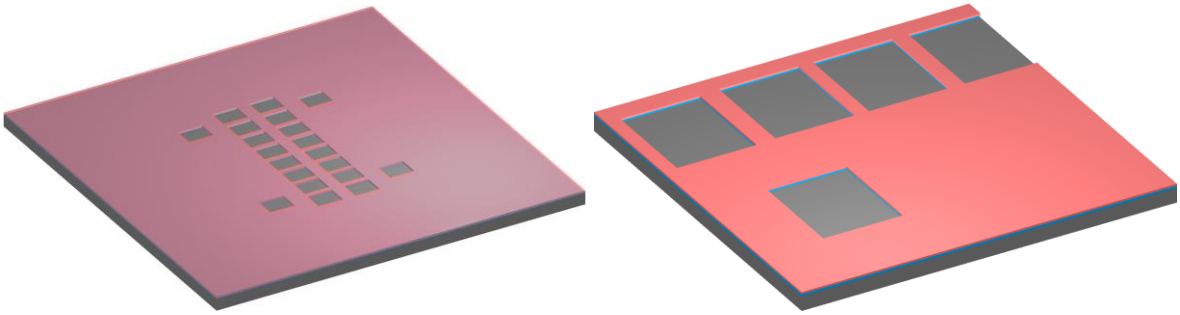
2-Crescimento de um óxido térmico



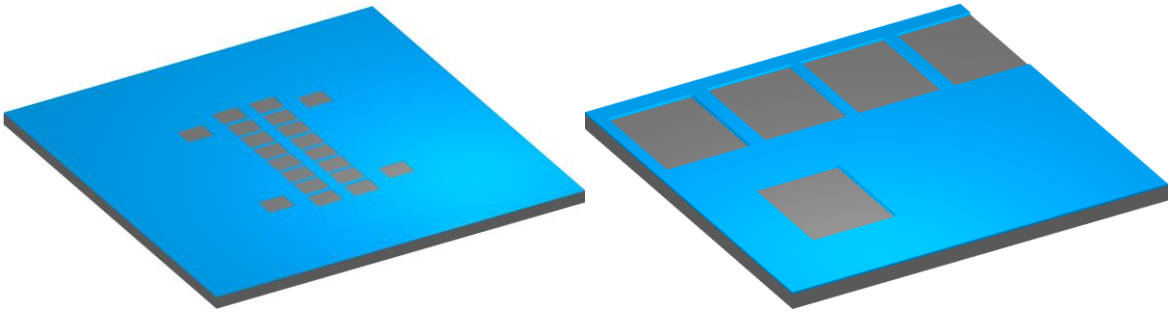
3-Spin coating do fotorresiste



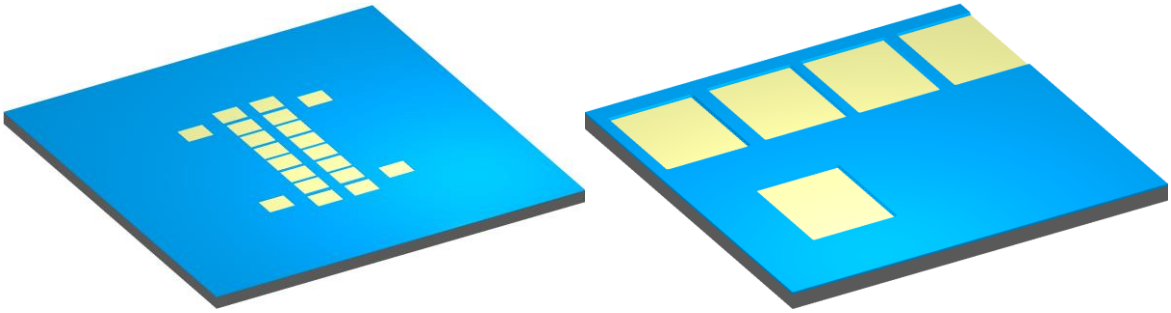
4-Revelação do fotorresiste



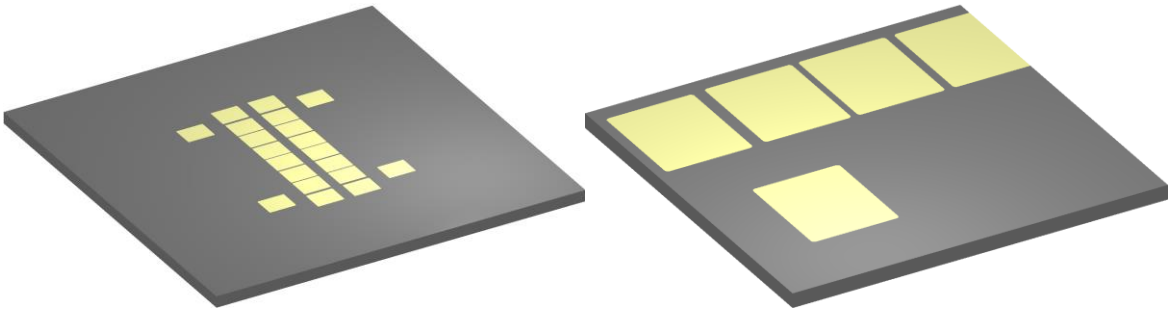
5-Corrosão da camada de óxido



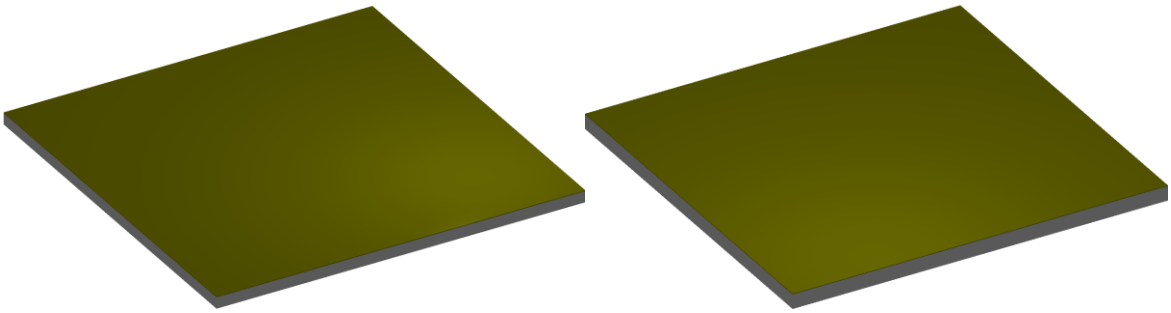
6-Remoção do fotorresiste



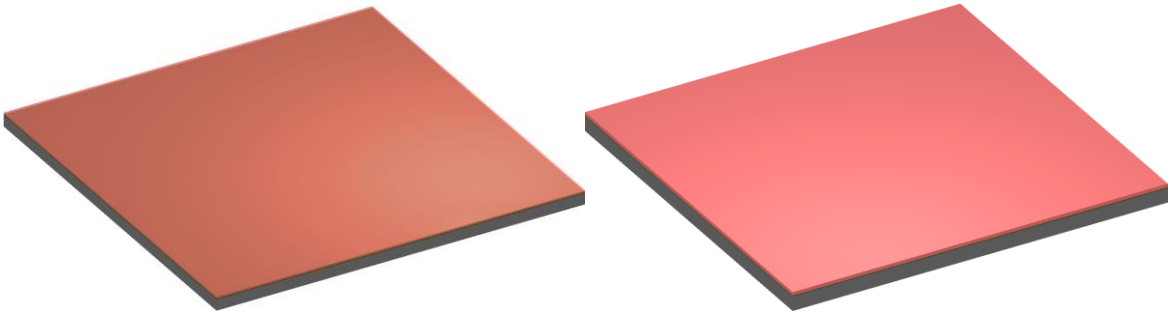
7-Dopagem térmica tipo N



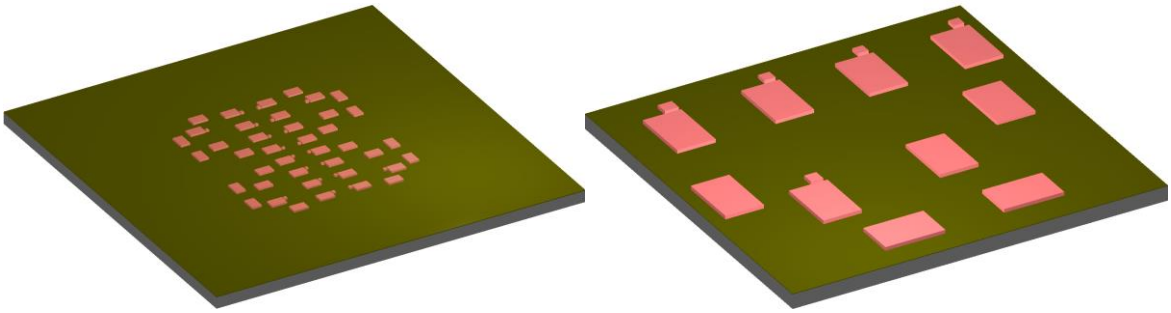
8-Remoção da camada de óxido



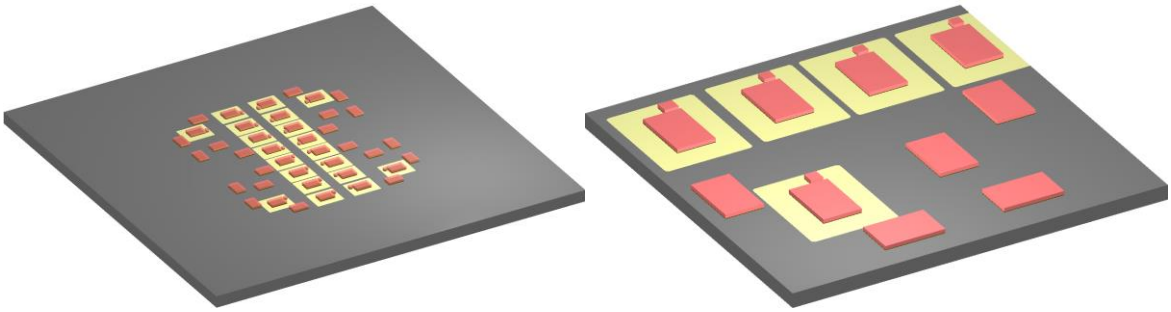
9-Deposição de filme de Si_3N_4



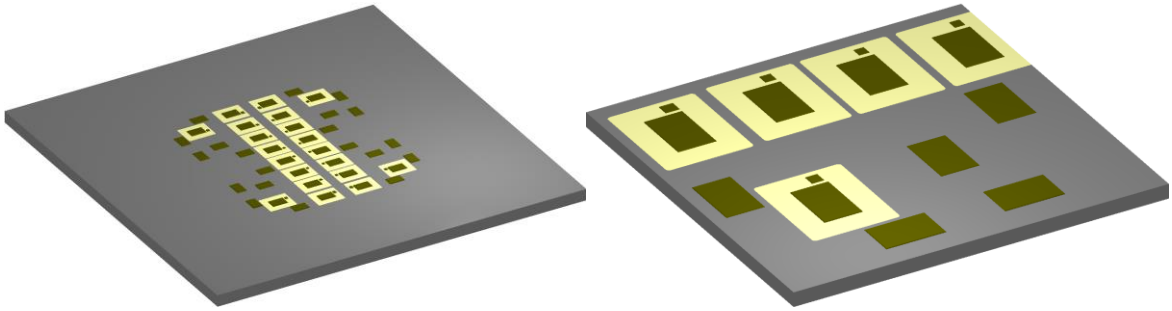
10-Spin coating do fotorresist



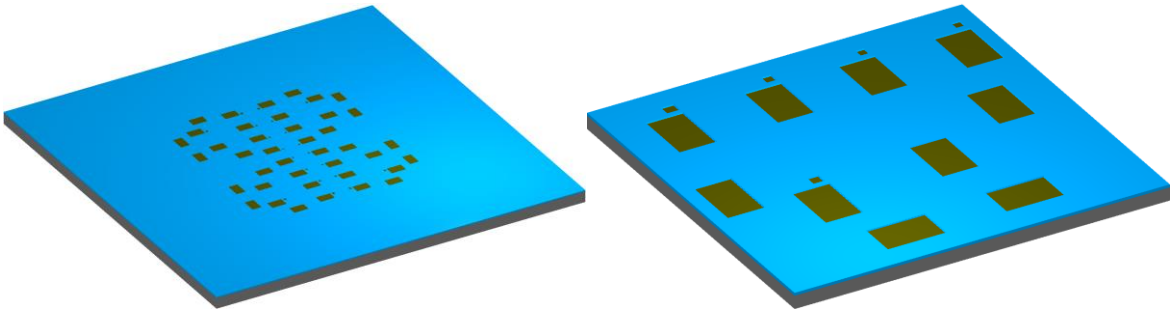
11-Revelação do fotorresiste



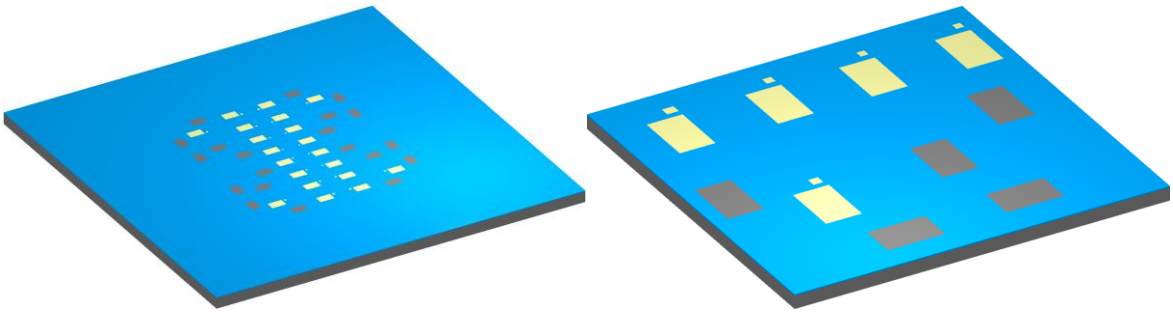
12- Corrosão do Si_3N_4



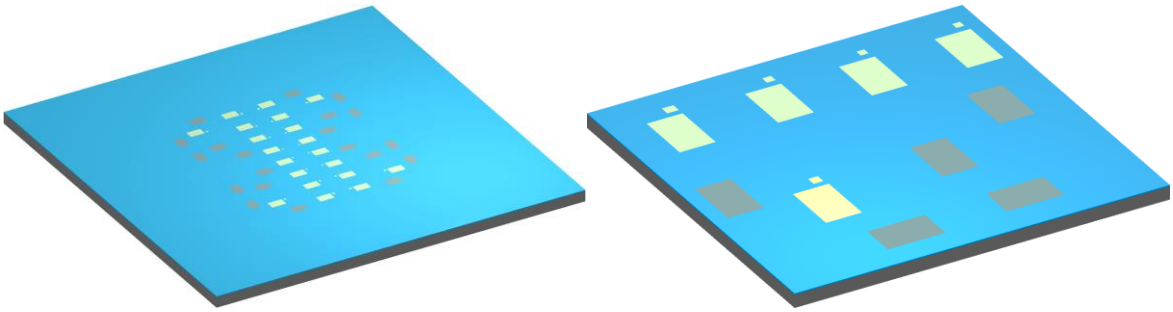
13-Remoção do fotorresiste



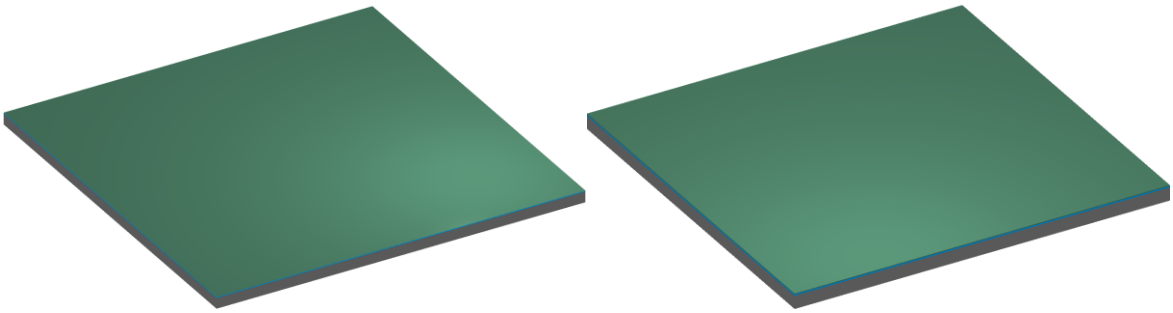
14-Crescimento de SiO_2



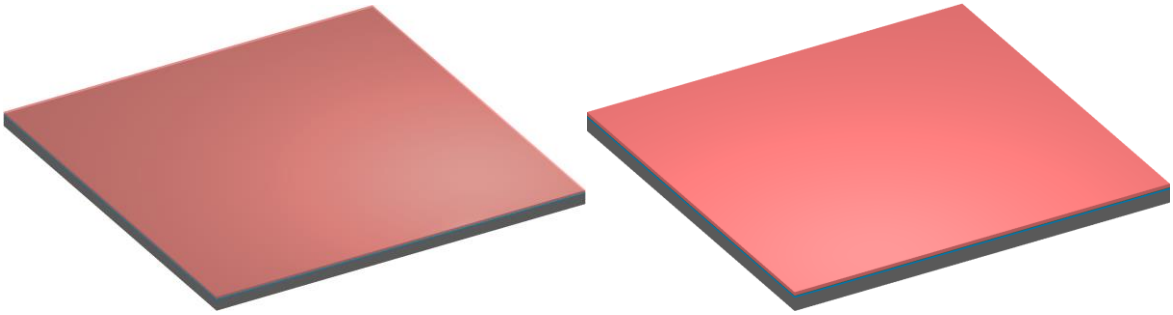
15-Remoção do Si_3N_4 por polimento



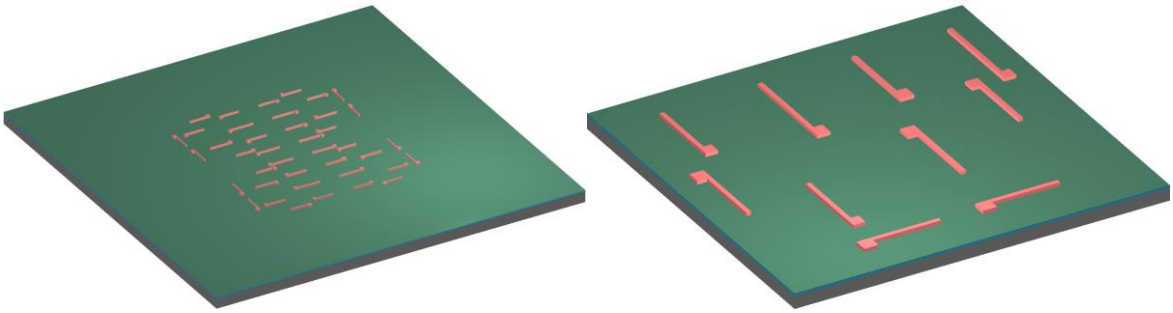
16-Deposição de SiO_2



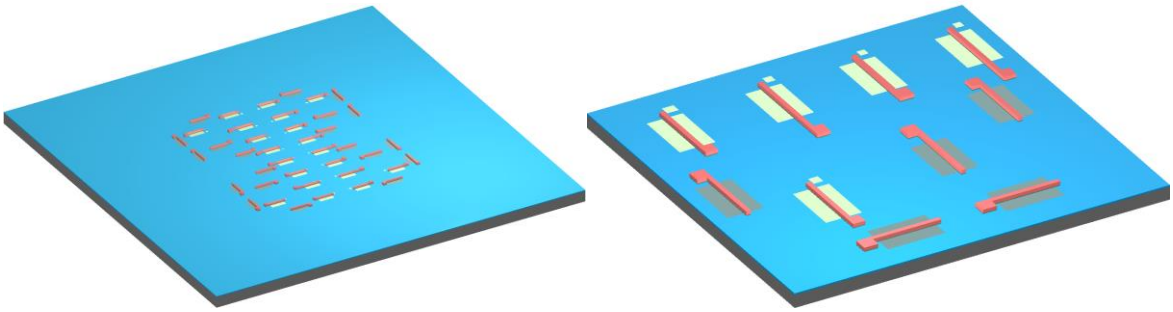
17-Deposição de Si policristalino



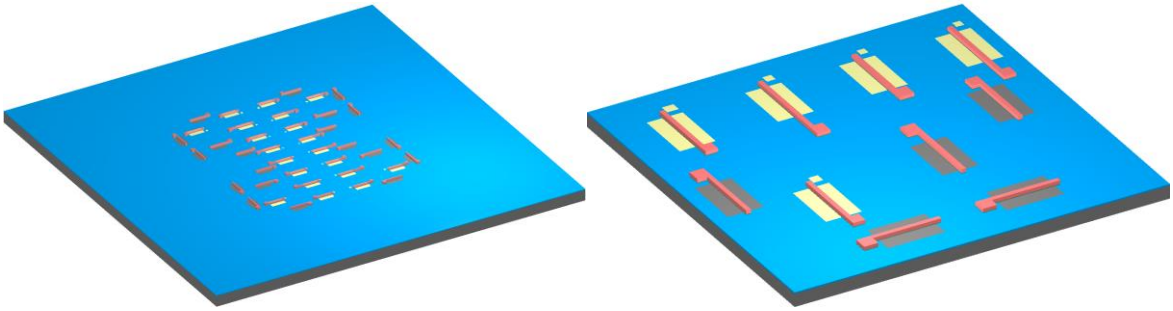
18-Spin coating do fotorresiste



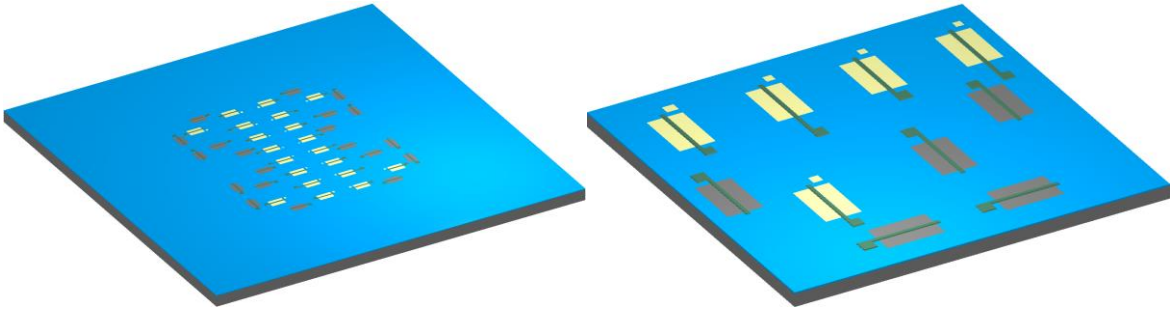
19-Revelação do fotorresiste



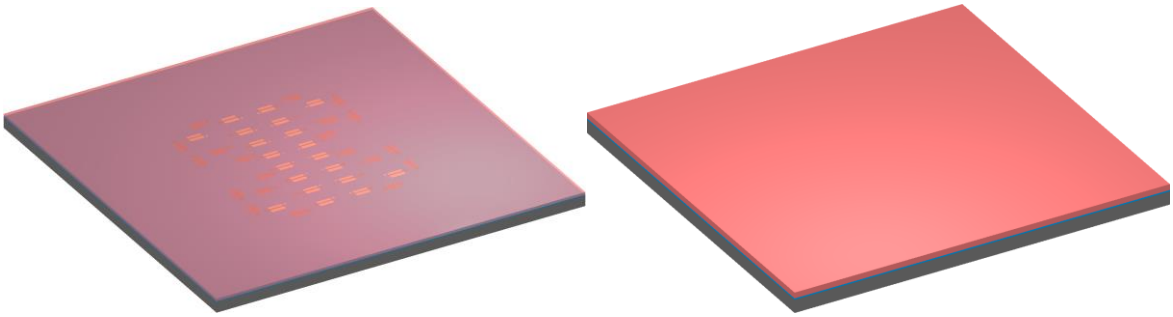
20-Corrosão do Si policristalino



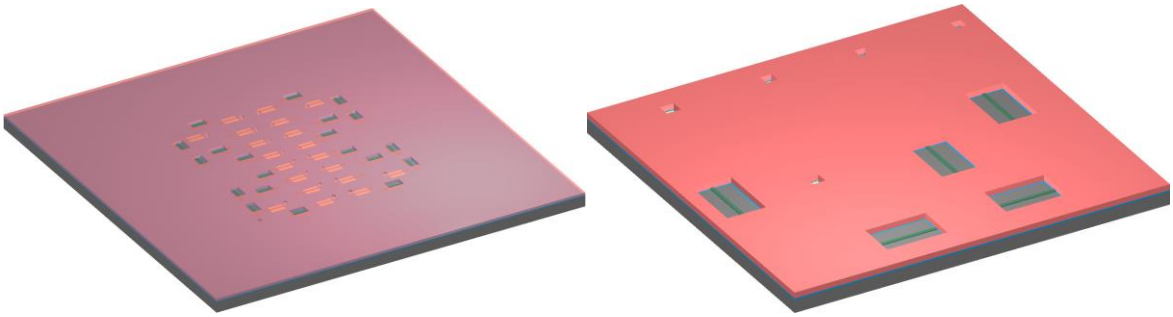
21-Corrosão do SiO_2



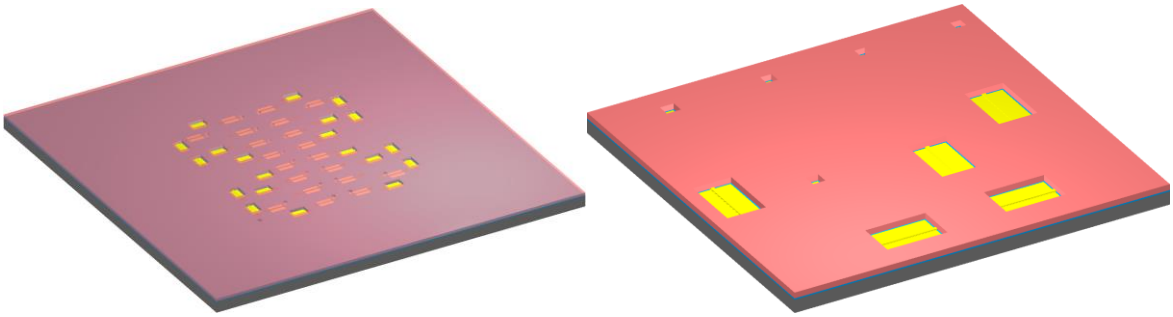
22-Remoção do fotorresiste



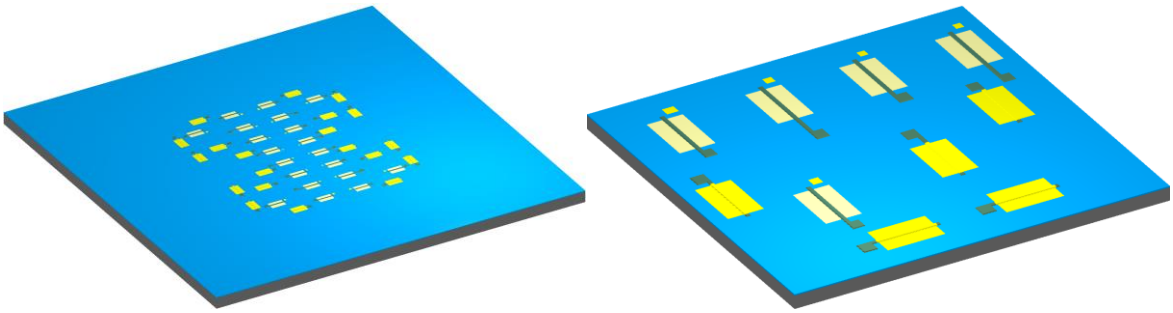
23-Spin coating do fotorresiste



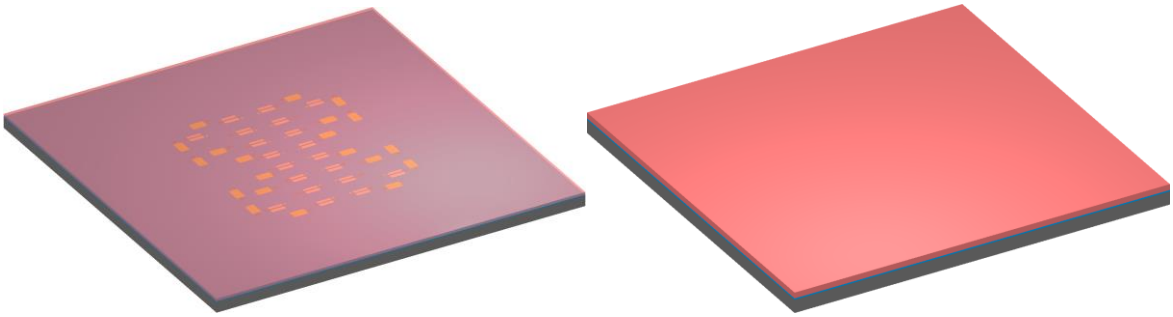
24-Revelação do fotorresiste



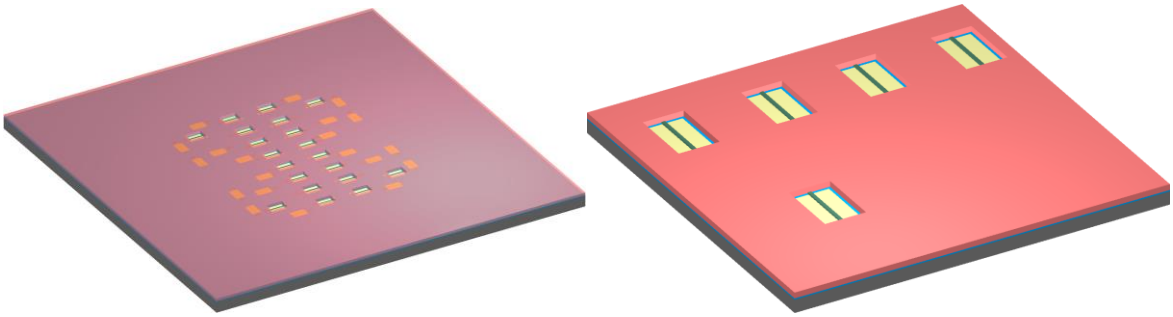
25-Implantação iônica (N+)



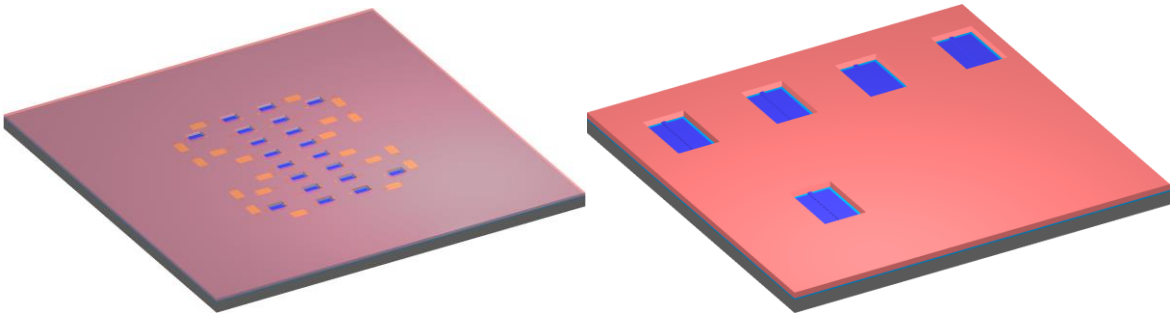
26-Remoção do fotorresiste



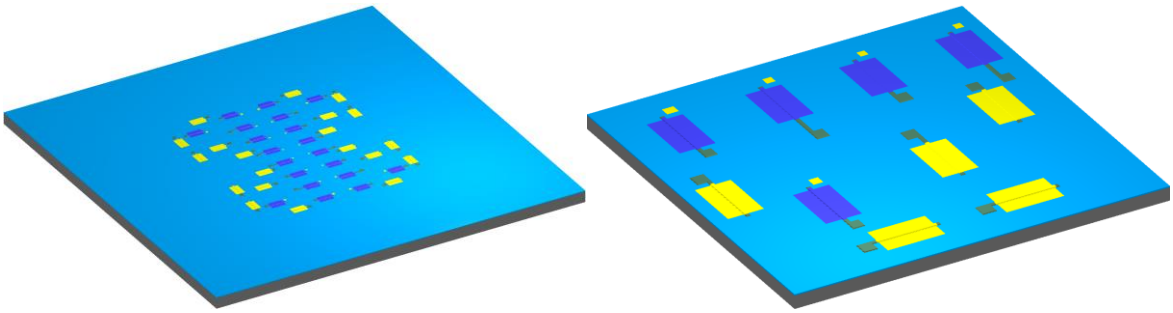
27-Spin coating do fotorresiste



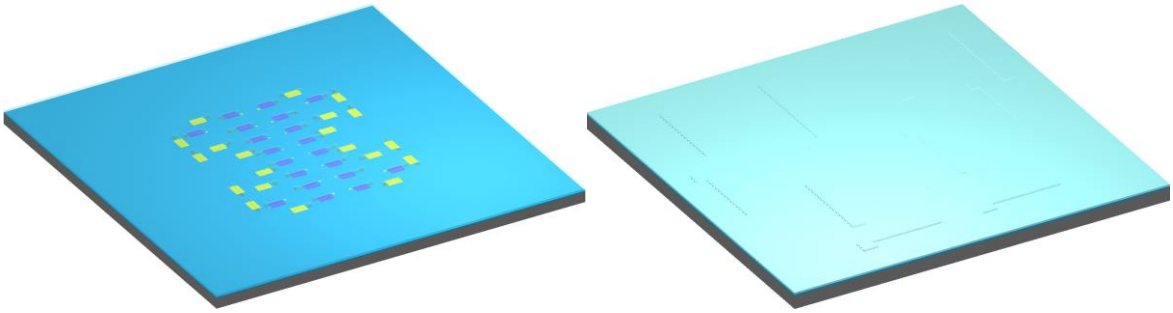
28-Revelação do fotorresiste



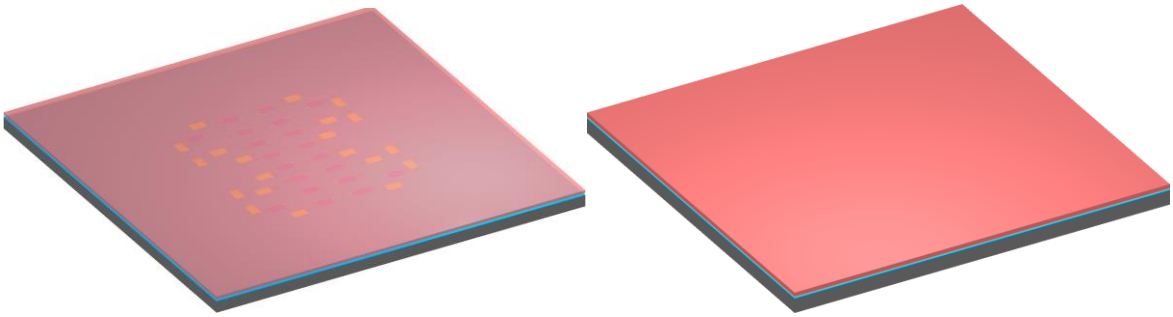
29-Implantação iônica (P+)



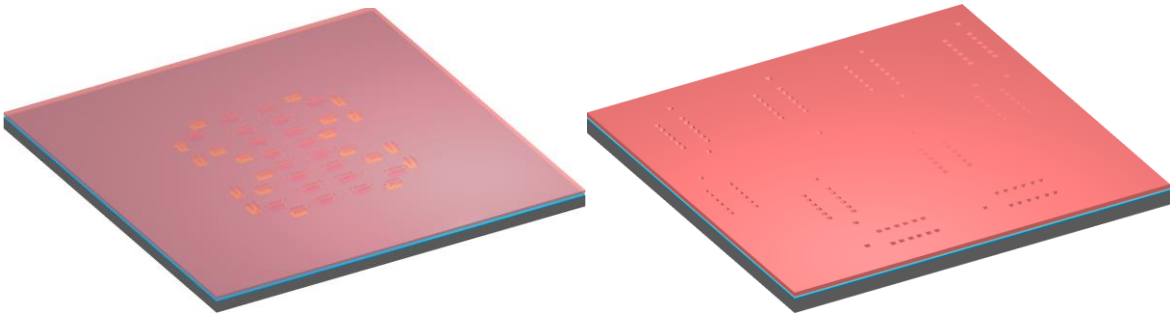
30-Remoção do fotorresiste



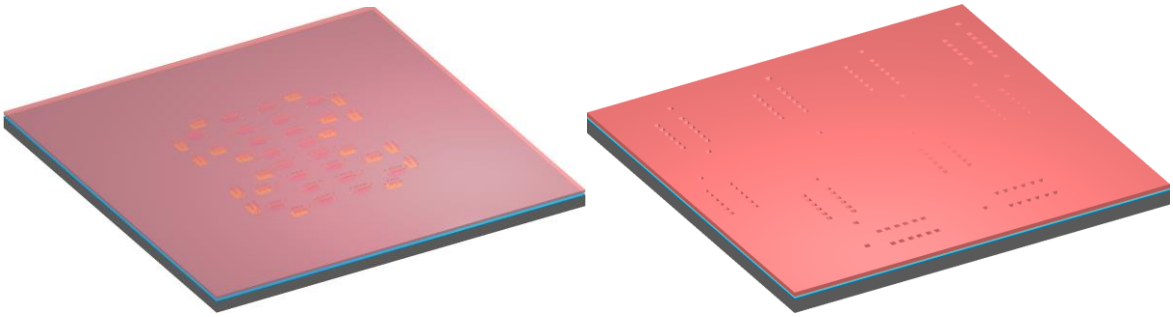
31-Deposição de SiO_2



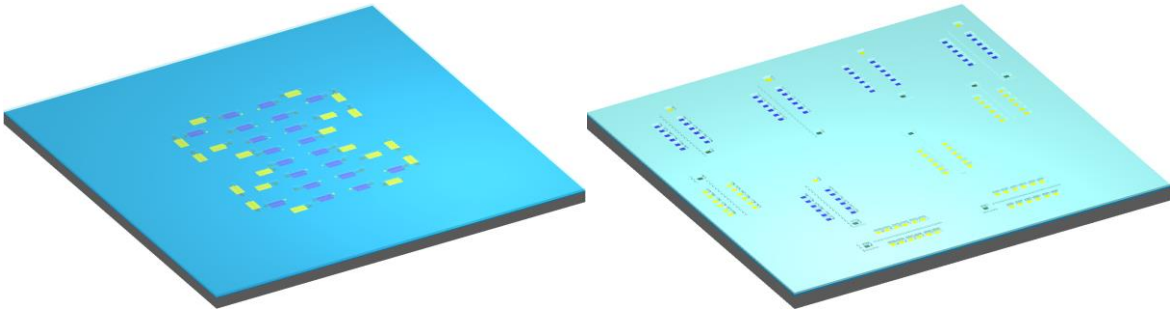
32-Spin coating do fotorresiste



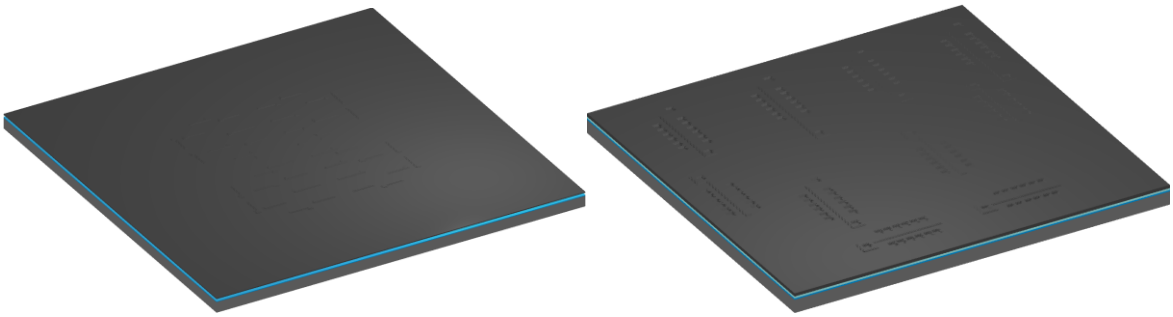
33-Revelação do fotorresiste



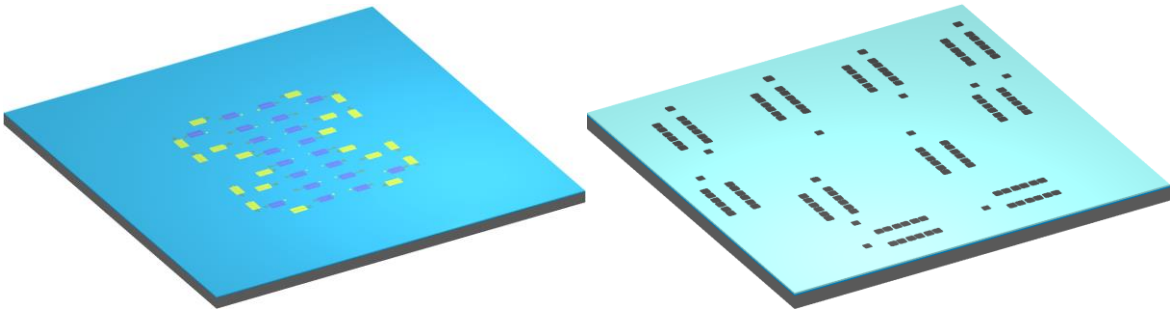
34-Corrosão do SiO_2



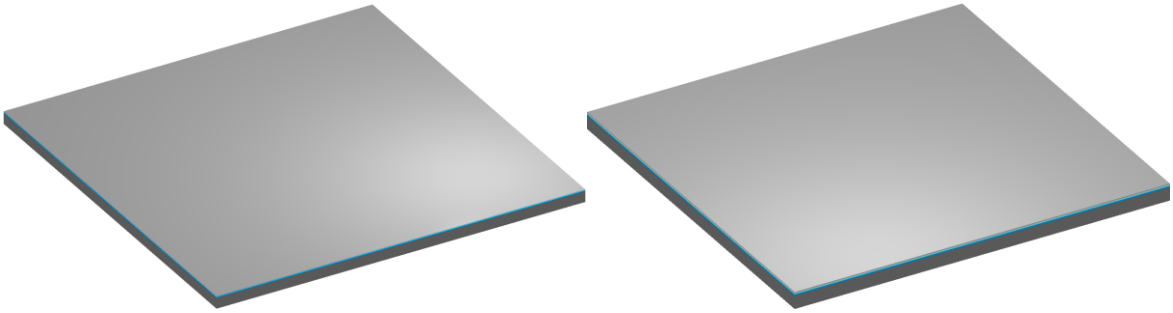
35-Remoção do fotorresiste



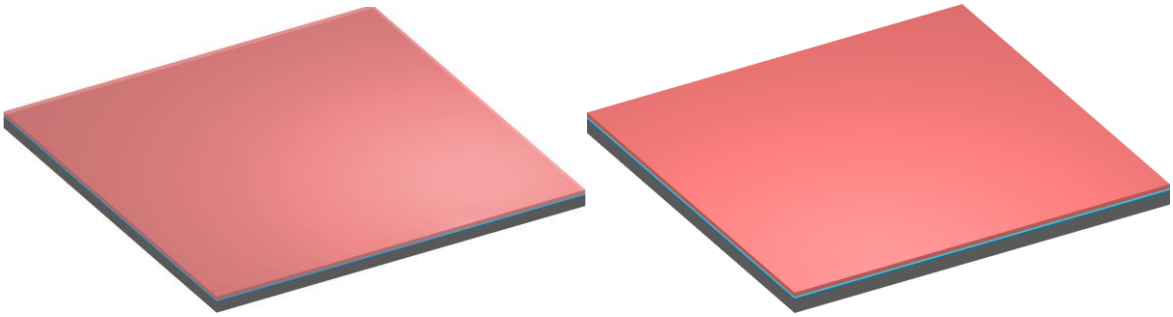
36-Deposição do metal da via



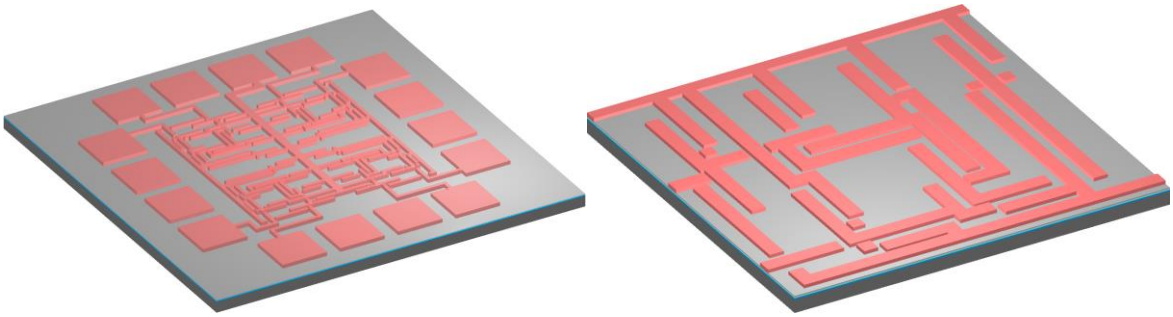
37-Polimento do substrato



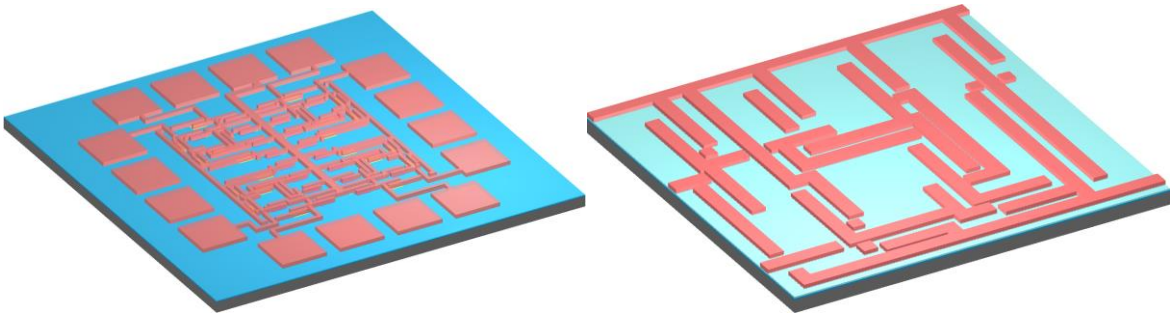
38-Deposição de Al



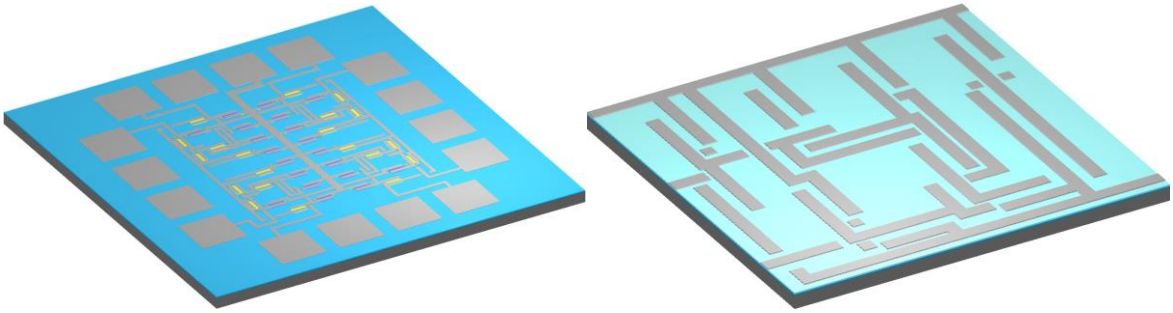
39-Spin coating do fotorresiste



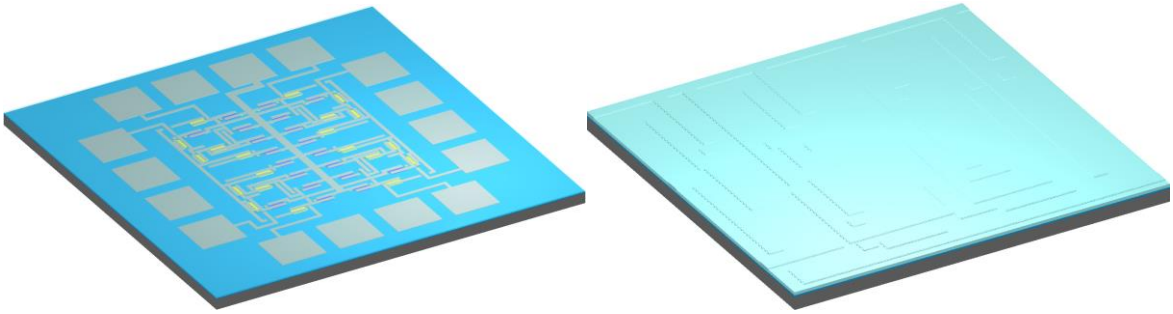
40-Revelação do fotorresiste



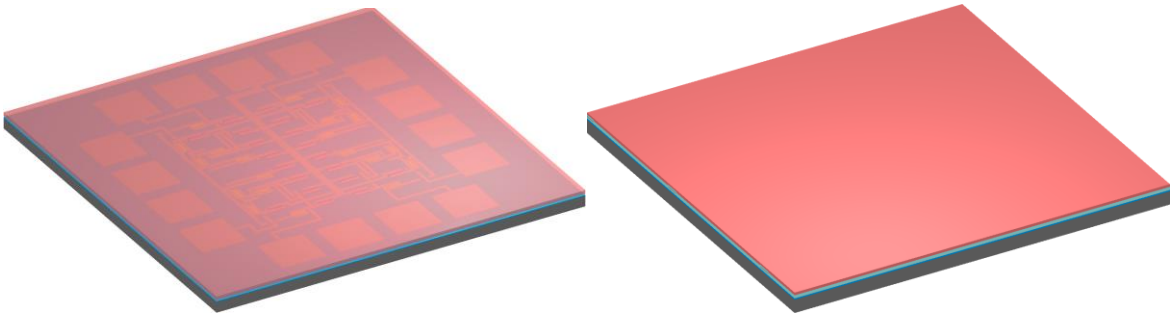
41-Corrosão do Al



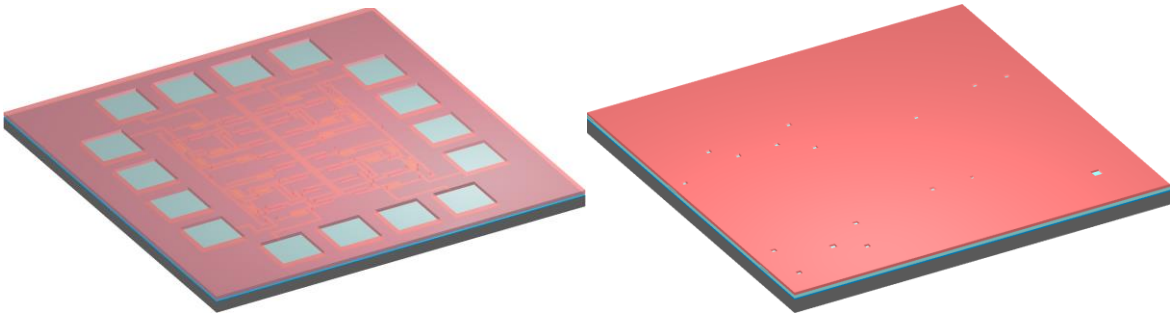
42-Remoção do fotorresiste



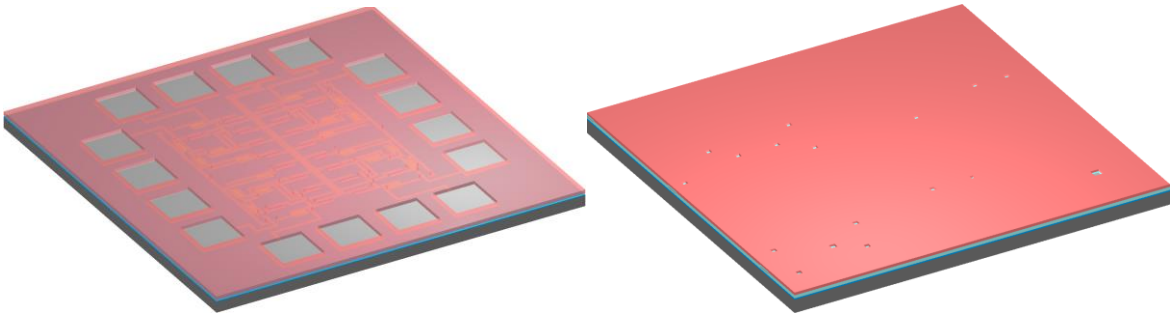
43-Deposição de SiO_2



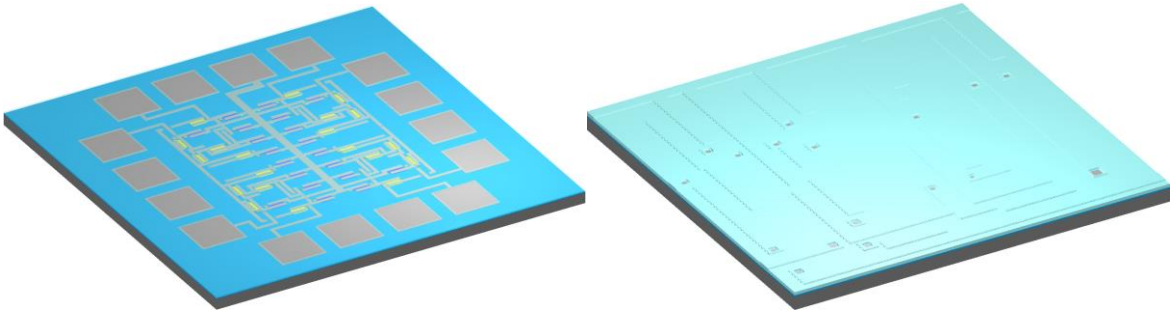
44-Spin coating de fotorresiste



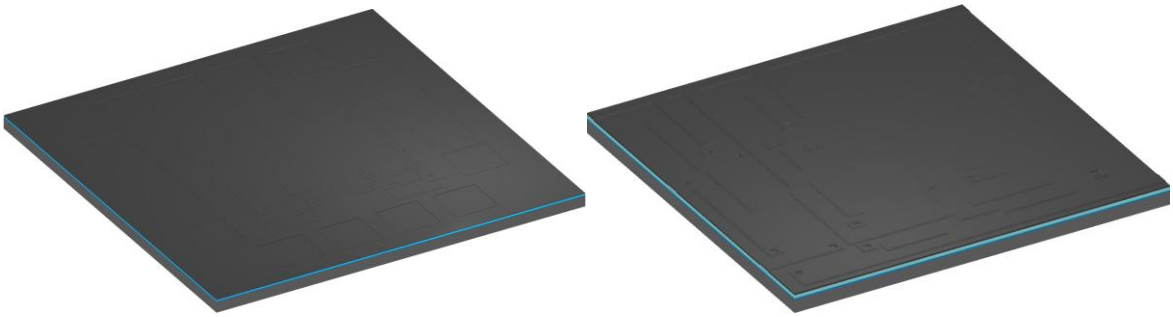
45-Revelação do fotorresiste



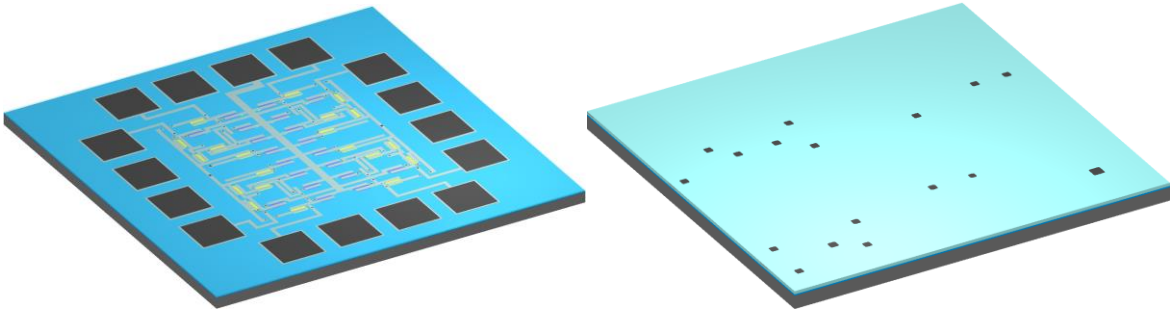
46-Corrosão do SiO_2



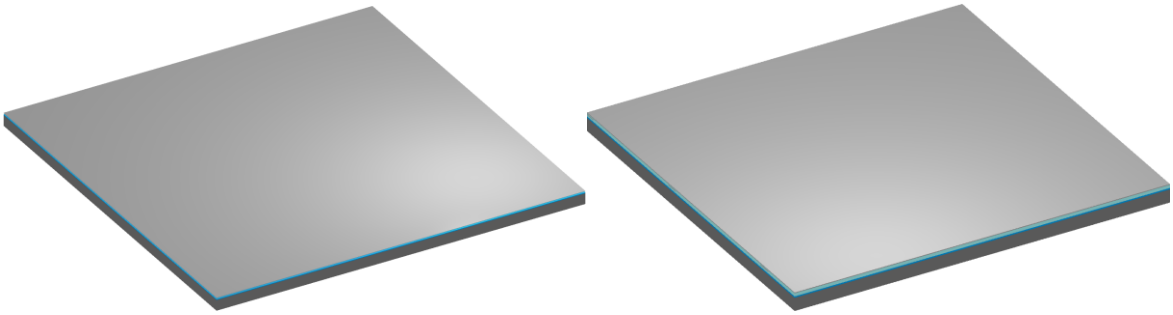
47-Remoção do fotorresiste



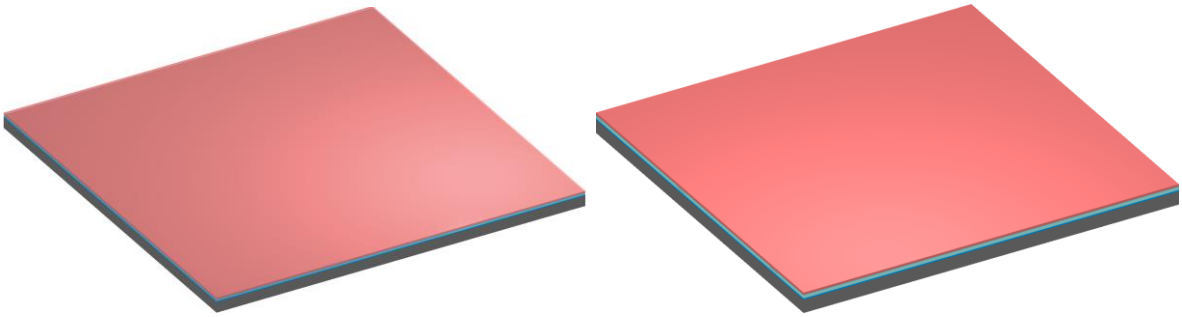
48-Deposição do metal da via



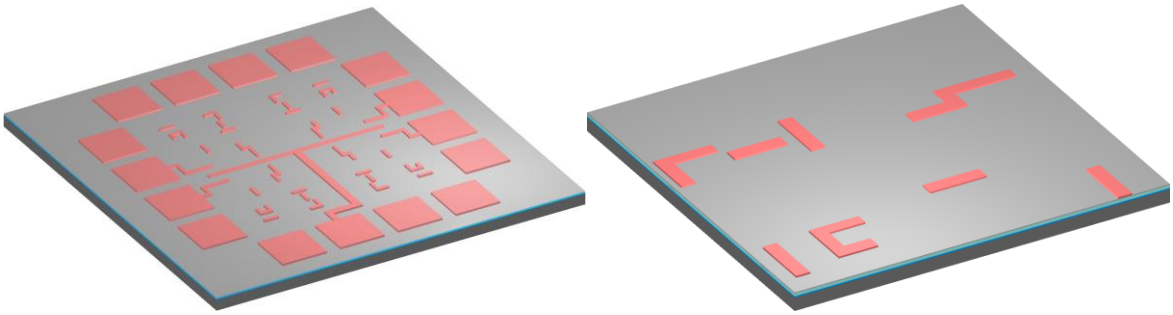
49-Polimento do substrato



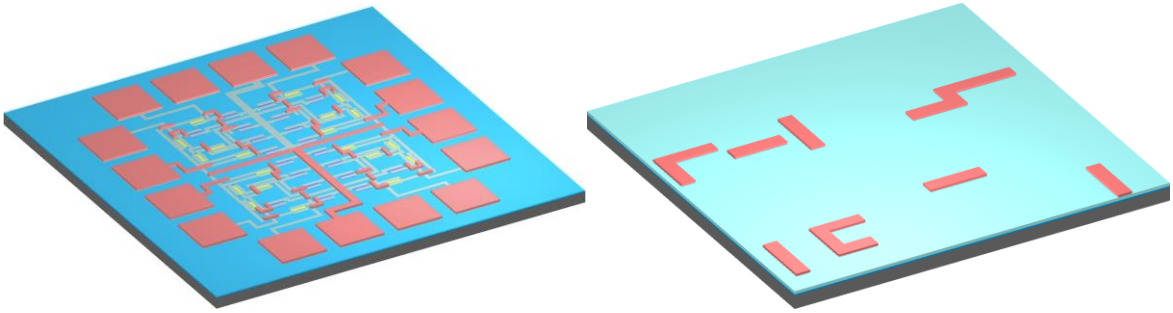
50-Deposição do Al



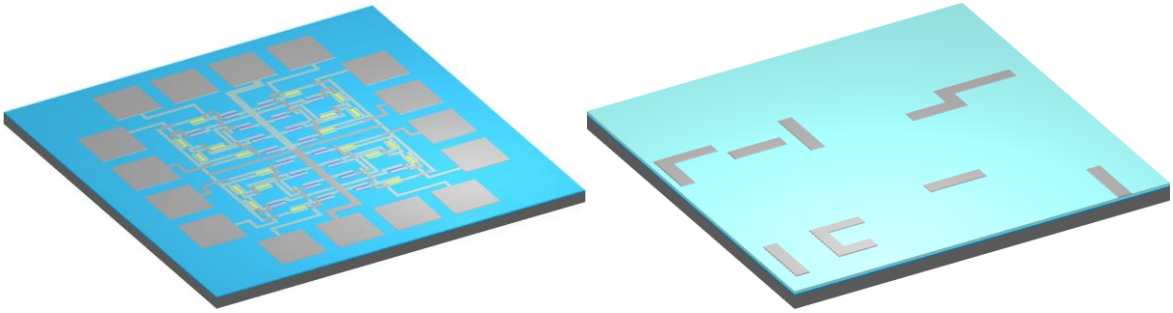
51-Spin coating do fotorresiste



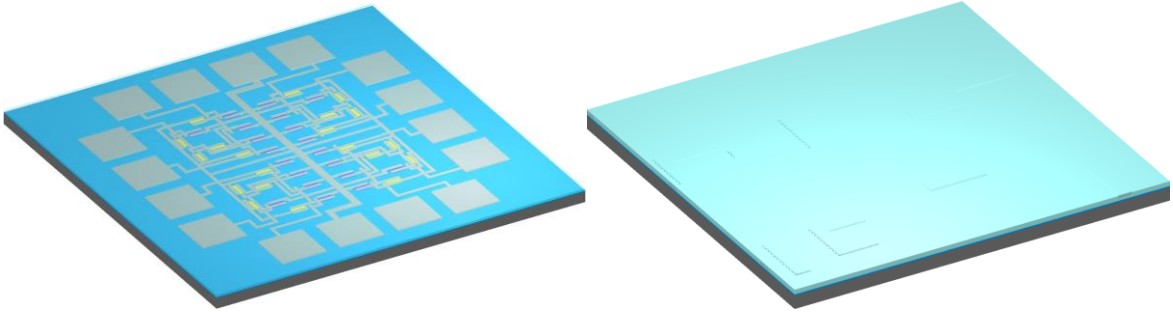
52-Revelação do fotorresiste



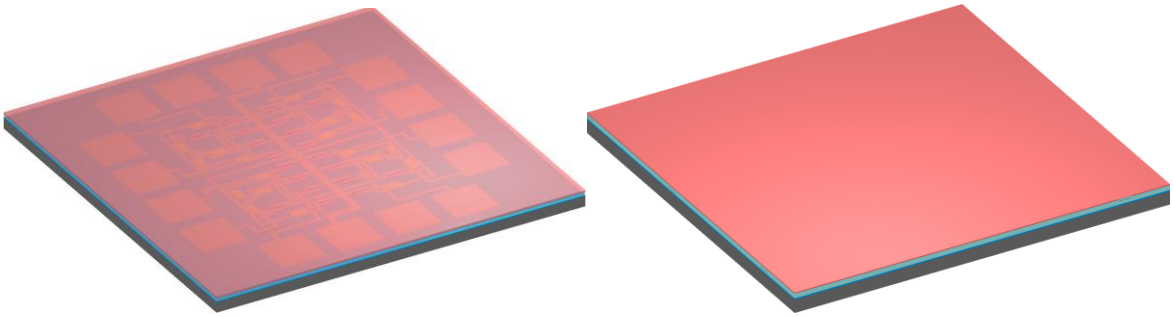
53-Corrosão do Al



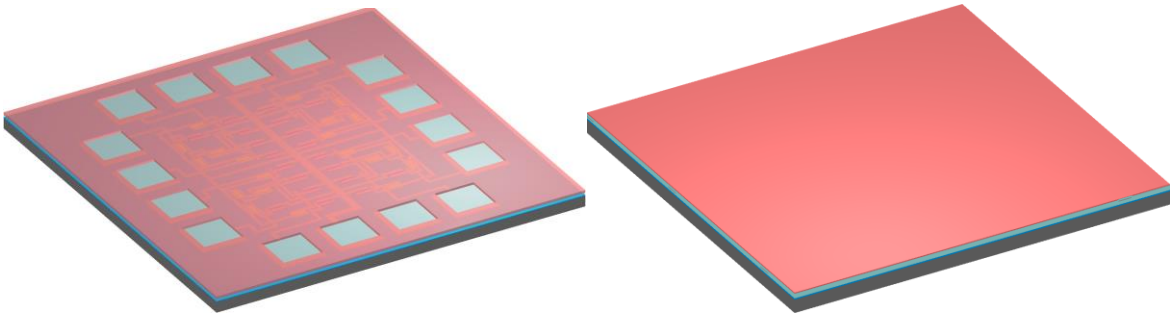
54-Remoção do fotorresiste



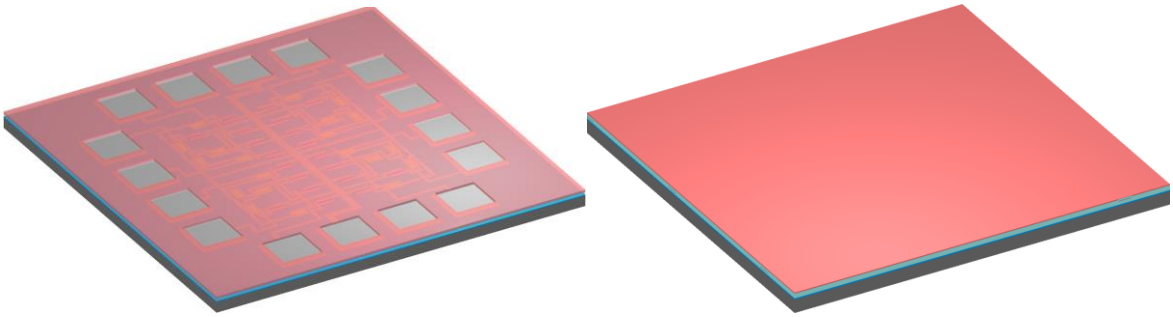
55-Deposição de SiO_2



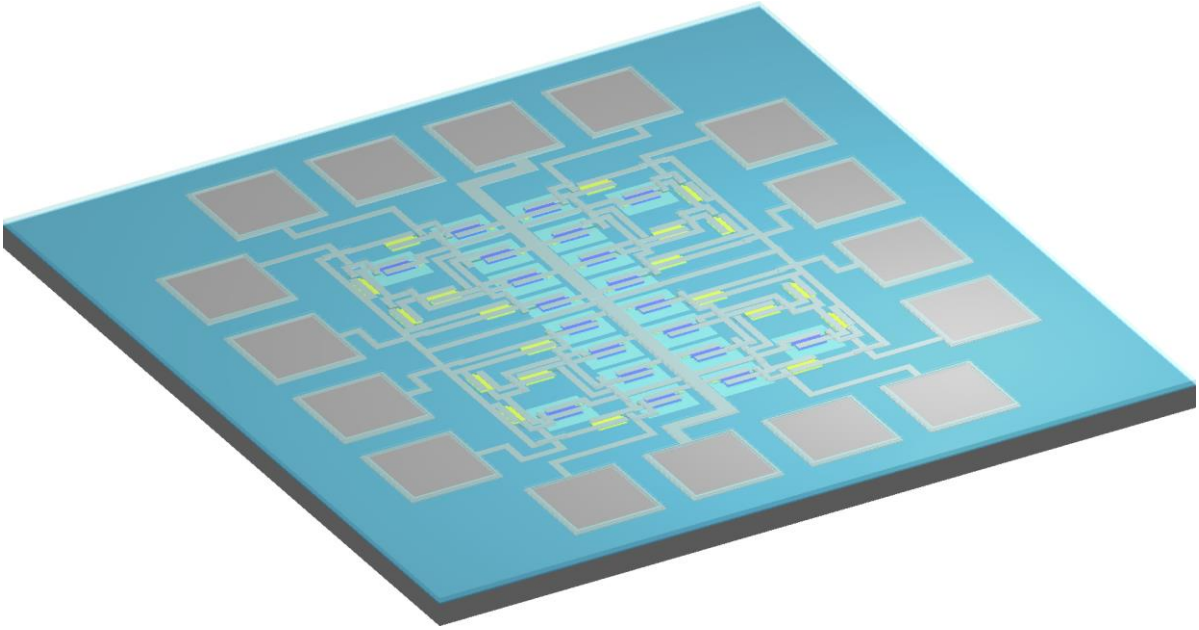
56-Spin coating do fotorresiste



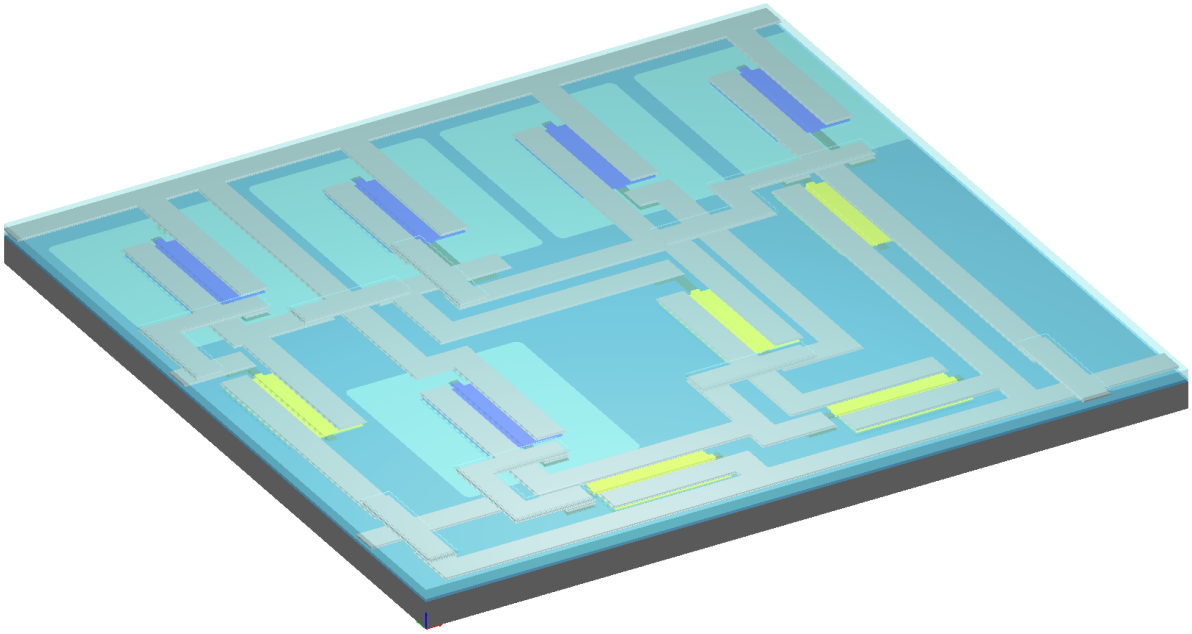
57-Revelação do fotorresiste



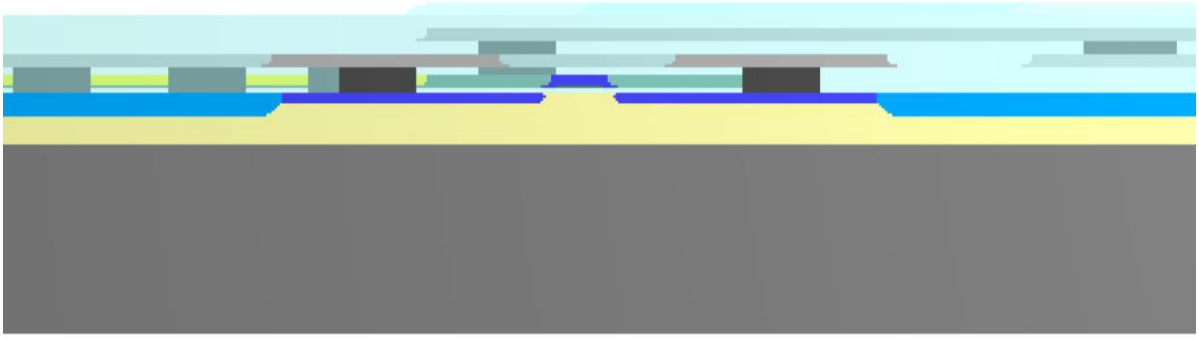
58-Corrosão do SiO_2



59-Remoção do fotoresiste, dispositivo final



59b-Remoção do fotorresiste, corte do dispositivo final



60-Corte lateral de um transistor

Figura 45- Etapas do processo de fabricação do circuito integrado.

O primeiro processo, mostrado na Figura 45-2, é uma oxidação térmica. O objetivo dessa etapa é formar uma camada de óxido que será usada como material de mascaramento para a dopagem dos poços N. As imagens 3 a 5 da Figura 45 mostram a maneira mais tradicional de corrosão seletiva no simMEMS. Primeiro é depositada uma camada de fotorresiste, em seguida o fotorresiste é revelado usando uma máscara. Finalmente, o material a ser corroído seletivamente, nesse caso o óxido crescido previamente, é corroído usando o fotorresiste como material de mascaramento. A Figura 45-6 mostra o resultado da corrosão quando o fotoresite é removido.

A Figura 45-7 mostra a dopagem térmica para formar os poços N. Nesses poços serão produzidos os transistores pMOSFET. Na Figura 45-8, a camada de SiO2 usada para mascaramento durante a dopagem foi removida. Isso foi feito para fornecer uma superfície lisa para os demais processos. Na Figura 45-9, foi depositada uma camada de nitreto de silício. As imagens 10 a 13 da Figura 45 mostram a fotogravação dessa camada, que foi usada como material de mascaramento para a produção de uma camada espessa de óxido de Si que forma o trench (Figura 45-14), isolando os transistores que serão fabricados. A camada de nitreto é então removida, novamente por polimento para manter o substrato liso (Figura 45-15).

Na Figura 45-16, uma fina camada de óxido de Si é depositada por CVD. Essa camada formará o óxido do gate nos transistores MOSFET. Em cima dessa camada, é crescido Si policristalino que formará o contato dos gates (Figura 45-17). Esse Si deve ser dopado para garantir boa condutividade. É preciso manter em mente também que os processos posteriores de implantação iônica para dopagem também irão afetar a condutividade desse material.

As imagens 18 a 22 da Figura 45 mostram a fotogravação dessas duas camadas, dando forma aos contatos e expondo as regiões que serão dopadas para formar o dreno e a fonte de todos os transistores. A dopagem por implantação iônica usa nesse caso o próprio fotorresiste para o mascaramento. O mascaramento evita a dopagem tipo N das regiões que devem ser tipo P e vice-versa. As imagens 22 até a 30 da Figura 45 mostram essas dopagens.

Feitas as dopagens, uma camada mais espessa de óxido de Si é depositada sobre todo o material por CVD, como mostra a Figura 45-31. Essa camada tem a função de isolar os componentes das camadas de metal que serão depositadas sobre o óxido de Si.

As imagens 32 a 35 da Figura 45 mostram a fotogravação dessa espessa camada de óxido de Si para permitir a deposição do material de via, que fará o contato dos drenos, fontes e gates à primeira camada de metal. Também são feitas as conexões dos poços N ao V_{DD}. Para tanto uma camada de metal é depositada por evaporação sobre o substrato (Figura 45-36).

Para formar a primeira camada de metal, primeiramente o substrato é novamente polido, removendo o excesso de metal das vias, como pode ser visto na Figura 45-37.

Em seguida, uma camada de Al é depositada (Figura 45-38) e fotogravada (imagens 39 a 42 da Figura 45). Sobre essa camada de metal é novamente depositado uma nova camada de óxido de Si, como visto na Figura 45-43. Essa segunda camada é também fotogravada (imagens 44 a 47 da Figura 45) e o metal da via é novamente depositado por evaporação (Figura 45-48). Mais uma vez o substrato é polido (Figura 45-49) e a segunda e última camada de metal é depositada por evaporação (Figura 45-50).

A segunda camada de metal é então fotogravada (imagens 51 a 54 da Figura 45) e uma camada final de óxido é depositada sobre os dispositivos (Figura 45-55). Essa camada final terá a função de proteger o dispositivo. No entanto, é preciso abrir a região sobre os pads para permitir acesso ao dispositivo, o que é feito nas imagens 56 a 59 da Figura 45.

O uso dos processos de polimento e de um metal para via e outro para as camadas de metais refletem o processo usado no Microwind. Dependendo do tamanho das aberturas para os contatos das vias, é possível usar apenas uma deposição de metal para fazer o mesmo processo, e isso efetivamente era feito nos primeiros processos CMOS propostos. O resultado difere do aqui apresentado principalmente no formato final do dispositivo, visto que as diferenças de altura em uma camada são propagadas para todas as camadas acima.

As dopagens realizadas utilizam tanto dopagem térmica, em um forno, como o processo de implantação iônica. O Microwind não especifica um processo em especial, então ambos foram usados visto que se trata de um exemplo do que pode ser feito no software.

3.7.1.1 Importando Máscaras

O Microwind exporta máscaras no formato Caltech Intermediate Format (CIF) (SPROULL; LYON; TRIMBERGER, 1980). É um formato comum para descrição de máscaras, mas não é reconhecido pelo simMEMS. Para usar as máscaras geradas, foi preciso converter as máscaras para imagens.

Essa tarefa foi realizada com o auxílio do software AutoCAD, versão 2013 para Windows (AUTODESK, 2016). O formato CIF é semelhante a uma linguagem de

programação. O formato das máscaras é definido através do uso de algumas formas básicas e suas junções. No caso especial das máscaras exportadas pelo Microwind, apenas um tipo de forma é utilizado: um retângulo.

No arquivo gerado pelo Microwind, cada máscara ocupa uma camada. As diferentes camadas são identificadas e são seguidas pela definição dos retângulos que formam a máscara referente a essa camada. Esse arquivo foi convertido em um script para criar as máscaras dentro do software AutoCAD que gera as formas equivalentes, cada uma em sua respectiva camada, através de uma operação de substituição em um editor de texto. Os comandos para gerar retângulos no formato CIF foram substituídos pelo comando para gerar retângulos no AutoCAD. O mesmo foi feito para definir as camadas. Uma operação de união foi então realizada manualmente em cada camada, unindo todos os retângulos que formam a máscara daquela camada.

O uso do AutoCAD é de interesse visto que o passo de geração das imagens pode ser feito em diversas escalas facilmente. Após importar as máscaras para dentro do AutoCAD, cada camada pode ser impressa em forma de uma imagem usando uma escala fixa. No exemplo mostrado na Figura 45, a escala é de um pixel para cada 600 nanômetros, gerando máscaras de 1600x1600 pixels.

A simulação final foi realizada, portanto, com células de 0,6 µm. Cada transistor tem então um gate com 5 células de espessura, ou 3 µm. Embora o tamanho do transistor entre na escolha da escala, ele não é o fator determinante. Isso porque as espessuras dos filmes são bem menores do que as larguras dos dispositivos.

Para se obter um bom resultado visual quando se analisa os transistores de perto, com a visualização de cantos arredondados devido aos diversos processos de corrosão e deposição, é preciso ter cerca de 10 ou mais células de espessura para o menor filme. No caso desse dispositivo, existiam camadas de até 0,4 µm de espessura. Estas camadas foram representadas por uma única célula.

Cabe ressaltar que uma escala de 100 nm por célula (ainda insuficiente para uma espessura de 10 células) já eleva o tamanho das máscaras para 9600x9600 pixels. O volume de células chegaria a 9600x9600x480 células (mais de 44 bilhões de células), o que inviabilizaria a simulação. Portanto optou-se pela menor escala e a consequente queda de detalhamento. No entanto, como pode ser verificado nas imagens da Figura

45, fica claro que devido ao tamanho do dispositivo, não é possível ver claramente pequenos detalhes ao se olhar o CI completo.

Para mostrar os transistores em maior detalhe, o mesmo processo foi realizado em uma escala de 1 pixel para cada 100 nm, mas em uma região reduzida. As máscaras geradas possuíam apenas 2400x2300 pixels, para um volume de simulação de 2400x2300x150 células, aproximadamente 820 milhões de células. As imagens dessa simulação são apresentadas no **Erro! Fonte de referência não encontrada.**.

3.7.2 Dispositvo de Microespelho Digital (DMD)

O simMEMS foi inicialmente desenvolvido para simular a fabricação de MEMS. Como exemplo, abaixo é apresentado o processo de fabricação de um DMD (Digital Mirror Device). Trata-se de um pequeno espelho que pode ser atuado para redirecionar luz para duas direções diferentes. Em geral utiliza-se uma matriz de microespelhos, que têm diversas aplicações, como espectrografia, leitores de códigos de barras e projetores digitais.

O princípio de funcionamento é simples, luz de uma fonte incide sobre o espelho. Os microespelhos chaveiam a luz entre duas possíveis direções. Pode-se alternar, por exemplo, entre direcionar a luz para um alvo ou em direção a uma superfície que absorve essa luz. Dessa forma, é possível escolher, de forma discreta quais espelhos direcionarão a luz para o alvo.

O chip em si possui células de memória CMOS que controlam a posição do microespelho (LEE, 2013). Sobre essas células são fabricadas as estruturas móveis dos microespelhos. No processo a seguir, apenas a estrutura do microespelho, que fica sobre o circuito CMOS, será fabricada. Pelo que foi apresentado no item anterior, simular a fabricação também do circuito CMOS é possível. Isso não foi feito por duas razões. Primeiramente, embora haja alguma documentação do circuito CMOS que controla o espelho, ainda seria preciso desenvolver um processo para simular esse circuito e informações sobre esse processo são mais escassas. Além disso, um exemplo da fabricação de um circuito integrado com tecnologia CMOS já foi apresentado.

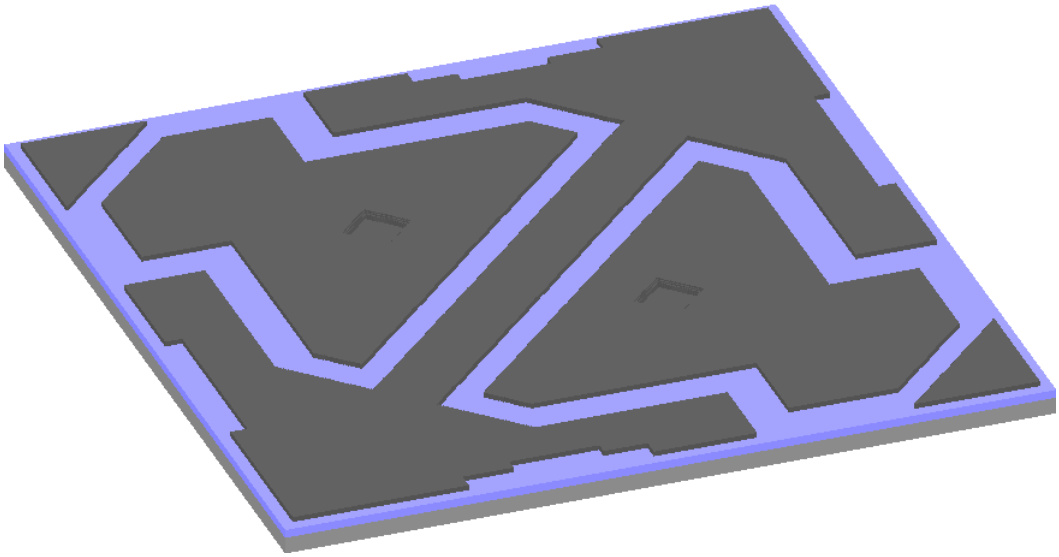


Figura 46 – Condutores formando a base do dispositivo.

O processo aqui apresentado foi desenvolvido baseado em datasheets, patentes, reports e apresentações que tratam sobre o funcionamento desses dispositivos (HORNBECK, 1991, 1998; HUFFMAN; HORNBECK; KNIPE, 2002; LIVERMORE, 2007; VAN KESSEL et al., 1998). O processo de fabricação, incluindo os materiais utilizados não é descrito nessas fontes e o aqui apresentado é, portanto, apenas uma alternativa.

A Figura 46 mostra a base do dispositivo, com os eletrodos que serão usados para operar o microespelho. As duas peças metálicas simétricas no centro do dispositivo fazem contato com o substrato de Si, onde estão as células de memória que controlam a posição do microespelho. O contato atravessando o modelo diagonalmente será ligado ao espelho. Os dois contatos simétricos ao seu lado estão ligados às células de memória abaixo do microespelho.

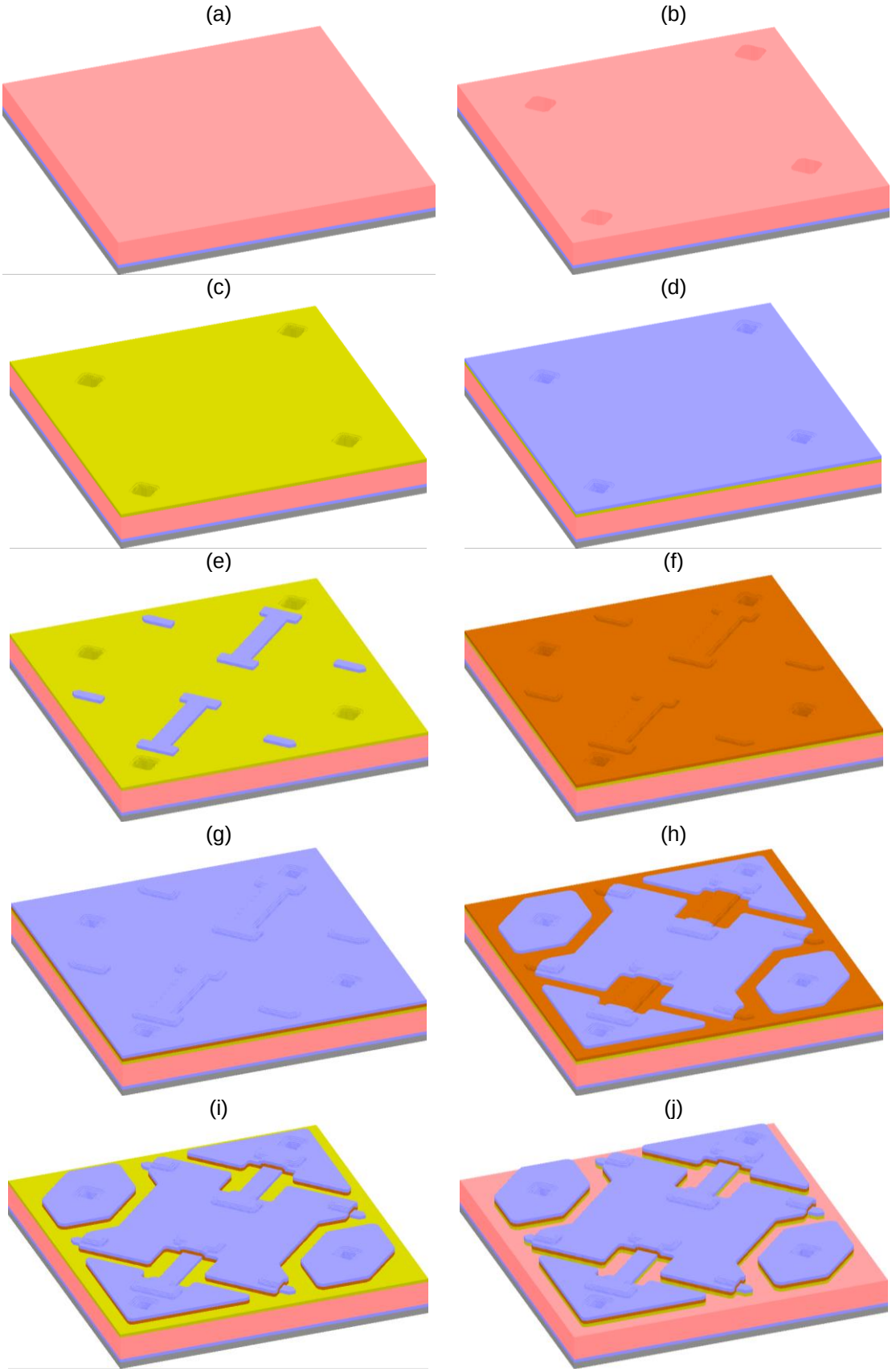


Figura 47 – Passos para formar a base do espelho e a mola de torção.

Sobre o substrato da Figura 46, é depositada uma camada espessa de fotorresiste (Figura 47a). O fotorresiste serve de base para a fabricação da base de apoio do espelho e da mola de torção. Também fornece o espaçamento necessário para permitir a rotação do espelho. Após fotogravar o fotorresiste para criar o espaço da via (Figura 47b), o metal que forma a base dos apoios e a mola de torção em si é depositado (Figura 47c). Em seguida uma camada de óxido é depositada sobre o metal (Figura 47d) e fotogravada (Figura 47e). Essa camada será usada como material de mascaramento em uma etapa futura. Um dielétrico é então depositado sobre o metal e o óxido de Si (Figura 47f). No passo seguinte uma nova camada de óxido de Si é depositada sobre o dielétrico (Figura 47g) e fotogravada (Figura 47h). Em seguida, o dielétrico e o metal são corroídos usando o óxido como material de mascaramento (Figura 47i e Figura 47j).

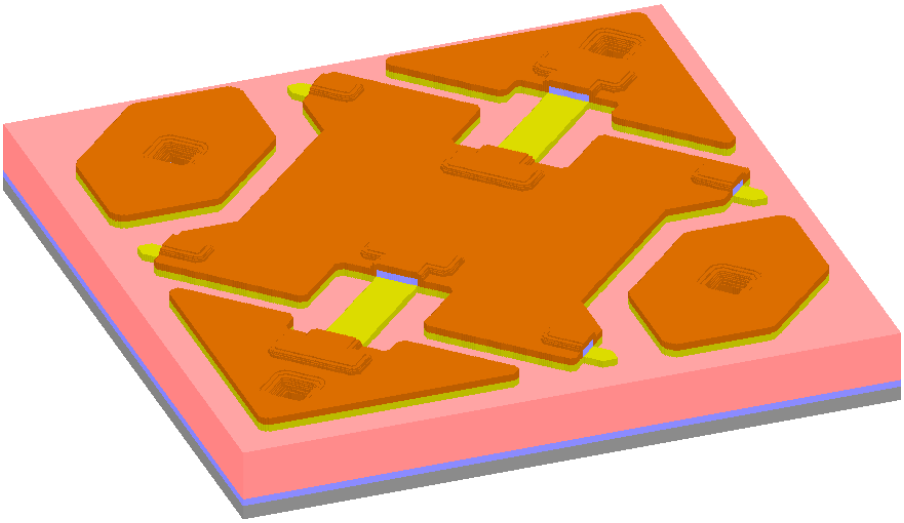


Figura 48 – Apoio e mola de torção finalizados.

As quatro pontas amarelas expostas na Figura 48 limitam o movimento do espelho, servindo também como molas para frear o movimento do espelho quando ele se choca com a base. As vias não ligadas à mola de torção também podem eventualmente entrar em contato com o espelho e a camada dielétrica sobre o metal evita um curto circuito nesse caso.

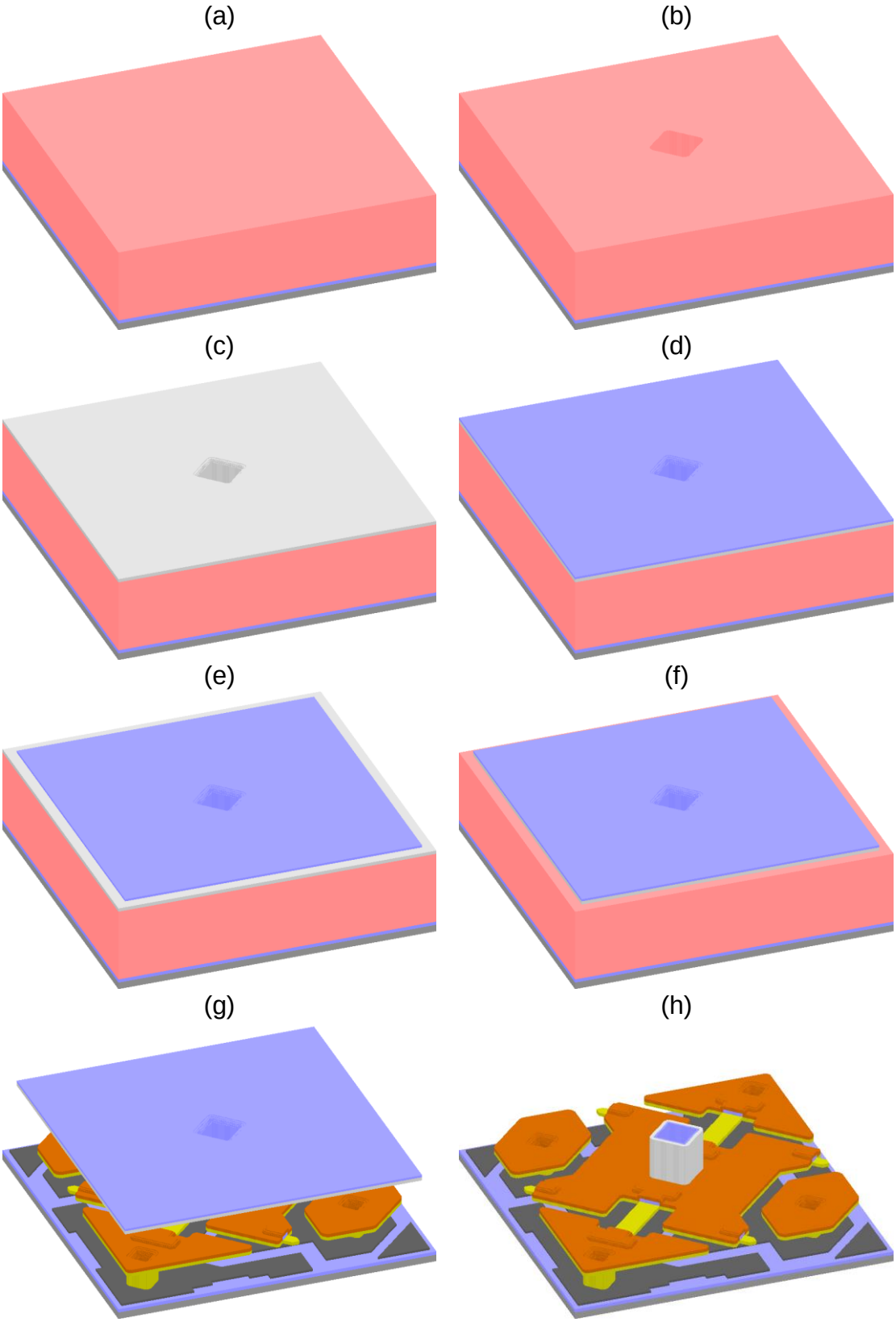


Figura 49 – Processo de fabricação do espelho sobre o jugo e a mola de torção. No final o topo do espelho foi removido para mostrar o apoio.

Sobre o substrato da Figura 48 é depositada outra camada de fotorresiste (Figura 49a). Essa camada de fotorresiste é usada como apoio para o espelho. O uso do fotorresiste é para garantir uma base reta sobre o qual depositar o metal do espelho, garantindo assim a boa qualidade do espelho. O espelho também precisa ser alto o suficiente para poder girar em torno da mola cerca de 10 graus antes de entrar em contato com as camadas abaixo.

Nas etapas finais do processo de fabricação, o fotorresiste é fotogravado para formar o espaço para a via (Figura 49b). Uma camada de alumínio é depositada então, formando a base do espelho (Figura 49c). Sobre essa camada, uma camada de óxido de Si é depositada (Figura 49d). Em seguida, ambas as camadas são fotogravadas (Figura 49e e Figura 49f), separando os diversos microespelhos. Finalmente, as camadas de sacrifício, usadas como apoio para as demais camadas, são removidas, liberando assim toda a estrutura (Figura 49g). Na Figura 49h, um corte foi feito removendo o topo do espelho de maneira a melhor exibir a mola de torção e os eletrodos do dispositivo.

Para comparação, a Figura 50 mostra uma matriz de DMDs produzida pela Texas Instruments, seguindo um processo semelhante ao aqui apresentado. Comparando os resultados As imagens abaixo mostram que é possível produzir um resultado bastante semelhante aos obtidos em um processo real.

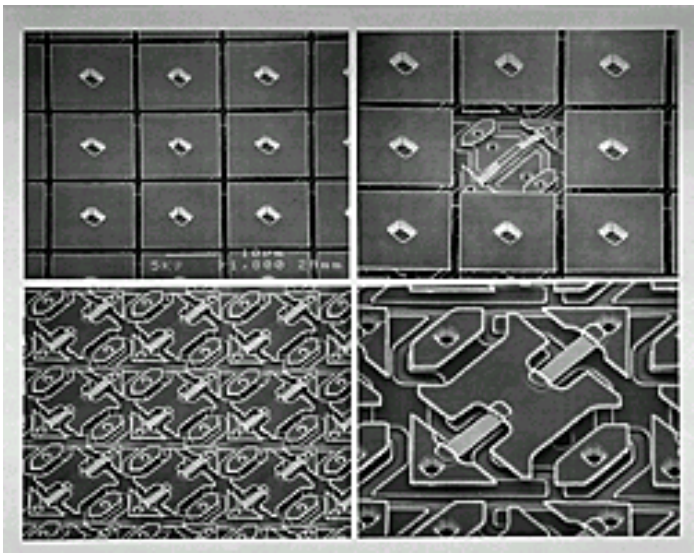


Figura 50 – Imagens de uma matriz de microespelhos produzida pela Texas Instruments. No topo à esquerda pode-se os espelhos ao final do processo. Na imagem superior direita, um dos espelhos foi removido, expondo os apoios e a mola de torção por baixo. Na imagem inferior esquerda, todos os espelhos foram removidos. A imagem inferior direita mostra em detalhe os apoios e a mola de torção de um dos microespelhos.

3.7.3 Corrosão Anisotrópica do Si (Microfabricação em Substrato)

O primeiro exemplo que foi analisado é a fabricação de uma microagulha seguindo o processo descrito em Shikida et al. (2004). Matrizes de microagulhas tem aplicações médicas e biológicas como eletrodos (CAMPBELL et al., 1991). O método de fabricação proposto por Shikida et al. visa diminuir o custo de fabricação dessas matrizes, o que é interessante principalmente visto que as mesmas devem ser descartáveis, enquanto mantém a possibilidade de controlar o tamanho e o espaçamento das agulhas na matriz (SHIKIDA et al., 2004). Alguns dos passos envolvidos na fabricação dessa matriz são apresentados na Figura 51.

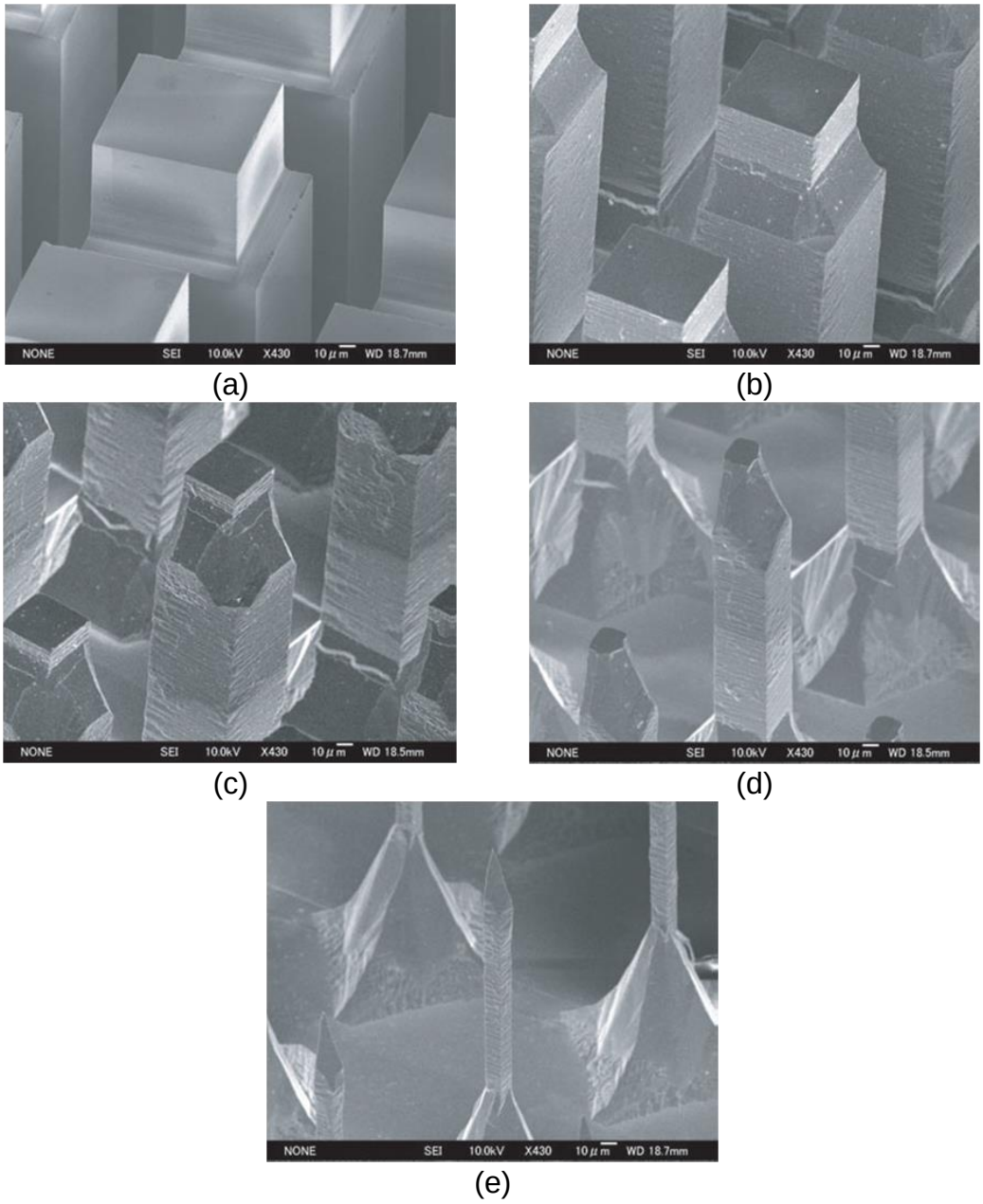


Figura 51 – Estrutura fatiada após (a) 0, (b) 10, (c) 20, (d) 30 e (e) 40 minutos de corrosão úmida (SHIKIDA et al., 2004).

O método de fabricação começa com uma oxidação térmica para proteção da superfície inferior da lâmina. Em seguida, o topo da lâmina é mecanicamente “fatiado” separando as futuras agulhas e formado um degrau (Figura 51a). Finalmente, a corrosão anisotrópica úmida é utilizada para formar a ponta afiada das agulhas (Figura 51b-e).

Na Figura 52 são mostrados os mesmos passos de fabricação da Figura 51 simulados dentro do simMEMS. Como a matriz é simétrica, apenas uma agulha foi produzida. O

simMEMS não possui um processo de corte mecânico como o usado em Shikida et al. (2004) que permita fatiar o substrato. No entanto, a mesma estrutura pode ser produzida por Deep Reactive Ion Etching (DRIE), esta sim disponível no simMEMS.

Comparando as imagens com seus respectivos resultados experimentais (apresentados na Figura 51), pode-se verificar que os mesmos são bastantes parecidos. As dimensões finais da agulha também são semelhantes as dimensões medidas experimentalmente.

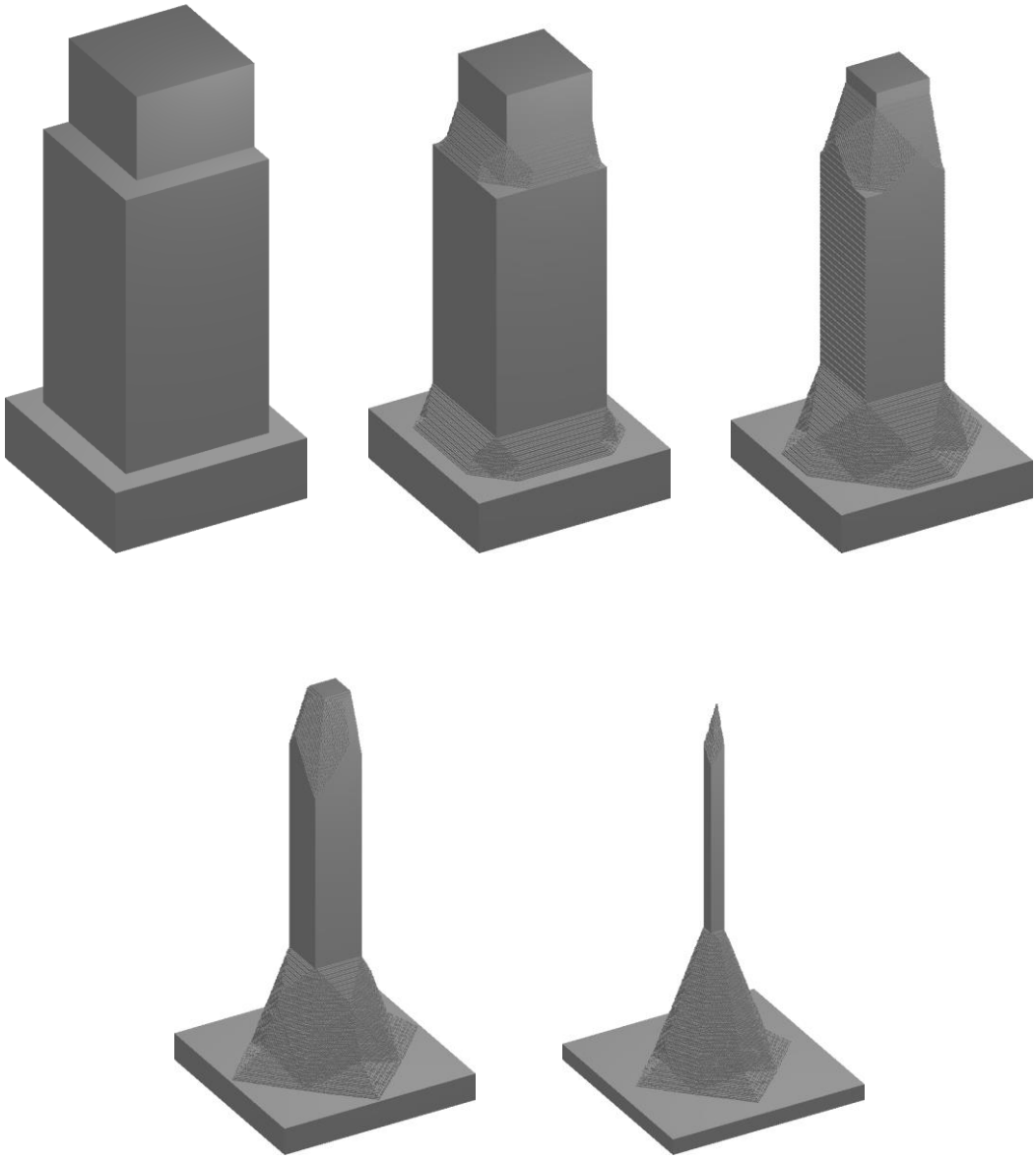
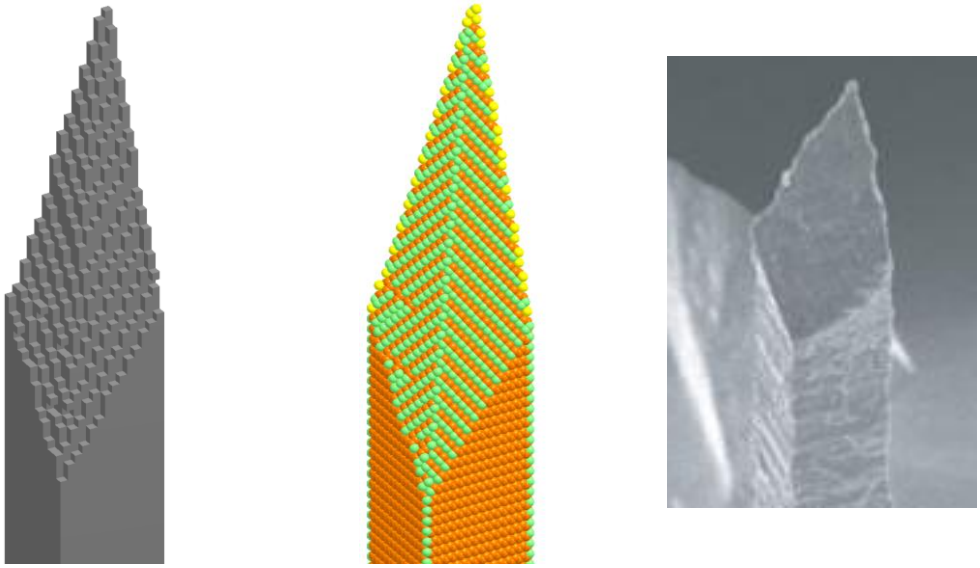


Figura 52 – Passos de fabricação da microagulha na versão 4 do simMEMS.

A Figura 53 mostra em maior detalhe a ponta da agulha. Como se trata de um processo de corrosão de Si, o simMEMS também permite visualizar a estrutura cristalina do substrato. O intuito dessa funcionalidade é facilitar a identificação dos planos cristalográficos que formam a estrutura. Como pode ser visto pela comparação com os resultados experimentais apresentados na Figura 53c, o formato final da ponta é bastante próximo do esperado. A agulha apresenta paredes retas formadas por planos {110} (são os planos compostos apenas por átomos laranjas) e a ponta afiada é formada por planos {311} (identificados pelo padrão intercalado de verde e laranja), o que era esperado (SHIKIDA et al., 2004).



(a) (b) (c)

Figura 53 – Vista da ponta da microagulha usando (a) células dos autômatos celulares, (b) átomos da rede cristalina do Si e (c) resultado experimental (SHIKIDA et al., 2004).

Para comparação, a Figura 54 também apresenta os resultados do mesmo processo gerado com a versão 3 do simMEMS. Verifica-se que os resultados são bastante semelhantes. Isso é esperado visto que o autômato celular utilizado é o mesmo. Para esta comparação, a dimensão dos substratos foi mantida aproximadamente constante nas duas simulações. A versão antiga do simMEMS trabalha com células não cúbicas, com razão de aproximadamente 1,0x1,0x0,35 entre as dimensões X, Y e Z, respectivamente. A versão atual trabalha com células cúbicas, então as profundidades foram mantidas e o número de células nas direções X e Y foram aumentadas. Dessa forma, o substrato passou de 70x70 para 197x197 na versão atual.

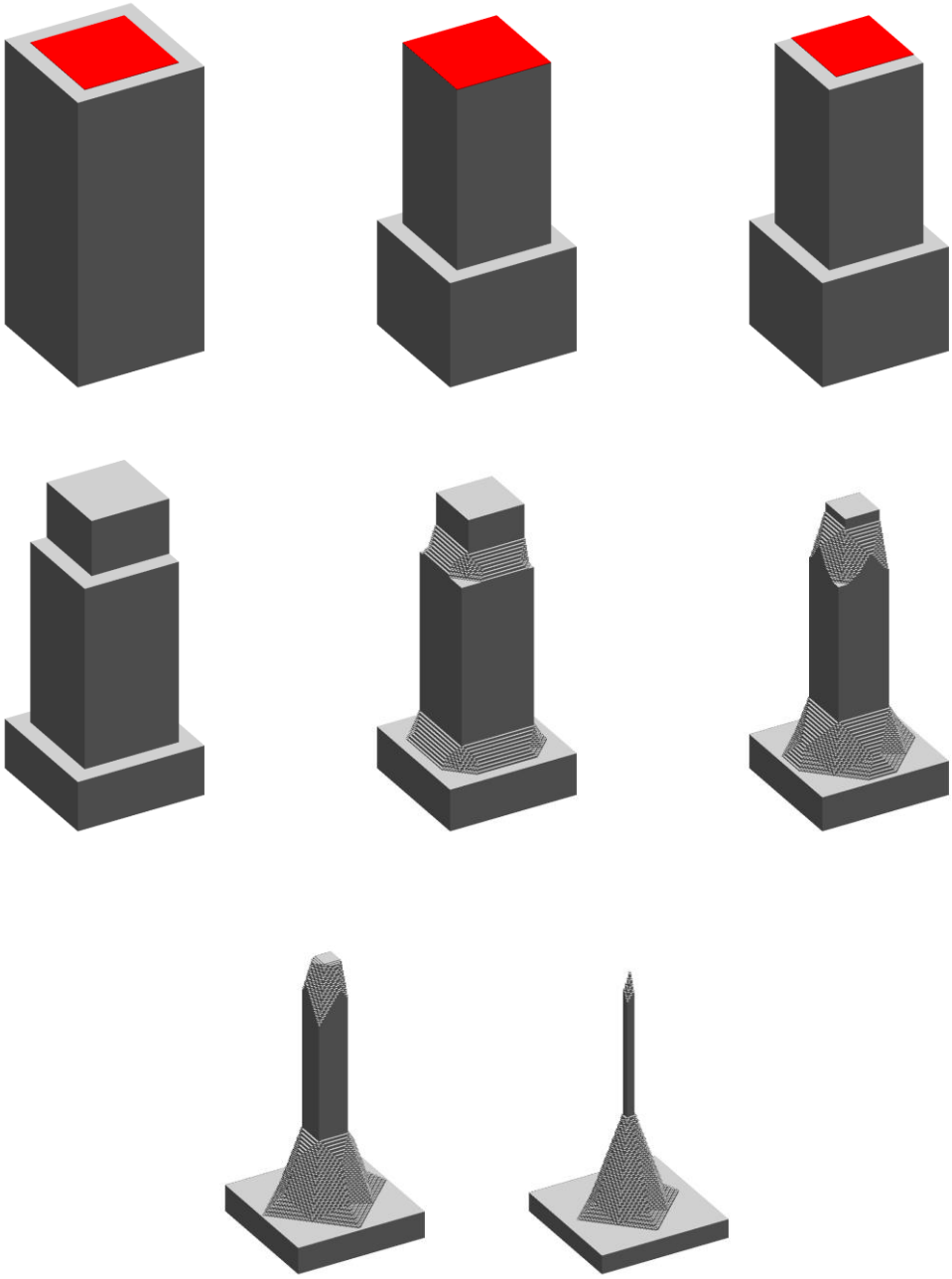


Figura 54 – Simulação do processo de fabricação de uma microponta na versão 3 do simMEMS.

O número de átomos usados na simulação de corrosão úmida em KOH aumentou consideravelmente. Isso pode ser visto pela comparação entre a ponta da agulha resultante da simulação na versão 4 do simMEMS com a gerada pela versão 3, mostrada na Figura 55. Enquanto na versão anterior a região afiada da ponta possuía comprimento equivalente à cerca de 12 células de diamante, a versão atual possui quase o dobro disso.

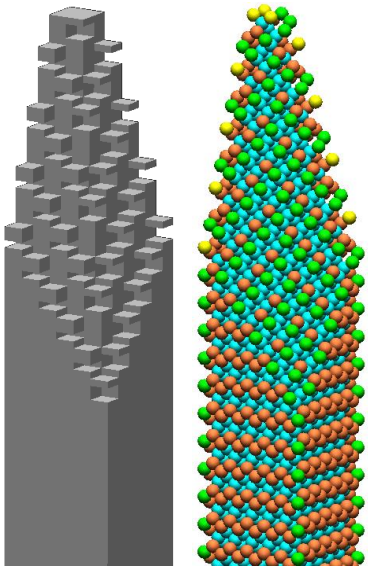


Figura 55 - Vista da ponta da microagulha usando células, à esquerda, e átomos, à direita, nos resultados gerados pela versão 3 do simMEMS.

Outro detalhe que pode chamar bastante a atenção é o formato das células do autômato celular dessa ponta. Na versão 3, cada célula em um substrato de Si contém exatamente um átomo em seu interior. A posição do átomo dentro da célula depende da posição da célula, mas pode ser calculada facilmente. Comparando as duas visualizações, tanto na versão atual como na versão antiga, é possível ver que em ambos os casos, a rede cristalina possui superfícies bastante planas. No entanto, enquanto a visualização em células da versão atual também apresenta faces planas, a da versão anterior é bastante errática. O resultado da Figura 55 é bastante diferente de um plano, com diversas reentrâncias e algumas células saindo para fora. Já o resultado na versão atual acompanha muito melhor a visualização atomística dos resultados.

Essa melhora na simulação está ligada ao maior número de átomos dentro de cada célula preenchida por Si. O resultado da simulação se assemelha mais a uma média da superfície formada pelos átomos do Si. A nova implementação apresentada, no entanto, algumas desvantagens. A mais clara é o aumento no consumo de memória. Embora os tamanhos das estruturas sejam semelhantes, a nova versão ocupa bem mais memória que a versão anterior, devido à necessidade de armazenar as duas representações do substrato de Si em paralelo, bem como ao maior número de átomos de silício.

A Figura 57 apresenta os resultados de uma simulação semelhante realizada em outro software para simulação de corrosão anisotrópica de Si. O software VisualTAPAS usa um método step-flow para simulação dessa corrosão, que permite melhor ajuste das taxas de corrosão dos planos cristalográficos do Si.

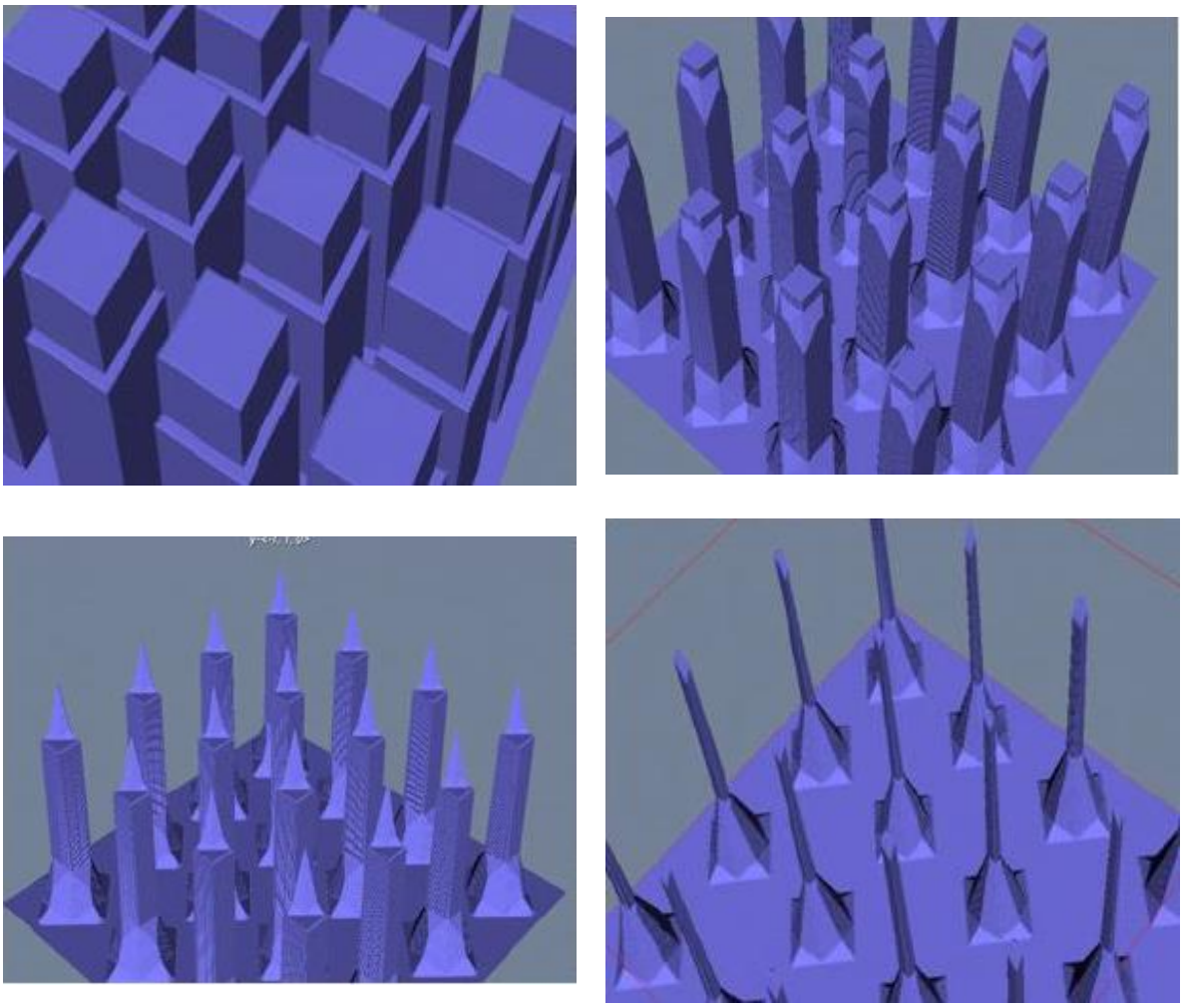


Figura 56 – Resultados de simulação da fabricação de uma matriz de micropontas usando o processo descrito acima. A simulação foi realizada no software VisualTAPAS (XING; GOSÁLVEZ; SATO, 2007).

Comparando os resultados na Figura 57 com os resultados gerados pelo SimMEMS, apresentados na Figura 52, é possível notar algumas diferenças. No entanto, o formato final da microponta é bastante parecido com o obtido pelo VisualTAPAS. Dessa forma, embora o autômato celular empregado seja bastante simples, fica claro que ainda é possível obter resultados bastante interessantes para simulação da corrosão anisotrópica do Si utilizando o SimMEMS.

No entanto, outro problema que surge pode ser claramente visto nos resultados apresentados na Figura 52. Note que na segunda imagem da corrosão úmida, aparece na face esquerda um padrão alternado de células a mais corroídas. Isso é resultado de erros de arredondamento no mapeamento da estrutura cristalina para a estrutura de células cúbicas. Átomos localizados muito próximos da borda entre duas células do substrato podem ser classificados de forma diferente, o que leva a essas variações.

Esse problema pode se manifestar de maneira mais crítica quando se realiza uma simulação onde todos os átomos na superfície são classificados erroneamente. Nesse caso, não ocorre nenhuma mudança de estados pois a superfície exposta é considerada não existente. Isso ocorre principalmente quando a altura do substrato em número de células cúbicas é um múltiplo inteiro de 14, pois a superfície do cristal (e respectivamente das células unitárias) será exatamente sobre a borda de duas células cúbicas dos autômatos celulares.

Finalmente, deve-se mencionar que a atual abordagem não é facilmente extensível. Como o Si pode ser corroído por diversas formas, como o DRIE utilizado nos passos iniciais do processo, é preciso que a estrutura cristalina que é armazenada em paralelo também seja atualizada. O mesmo deve ocorrer com qualquer outro valor que venha a ser armazenado de forma paralela. Isso significa que qualquer autômato celular novo implementado deve necessariamente levar em conta a existência desses dados e, de forma análoga, todo novo dado requer atualização de diversos autômatos celulares já implementados.

Dessa forma, embora a atual abordagem seja satisfatória, é preciso desenvolver mecanismos mais avançados para a atualização de dados paralelos usados por autômatos celulares específicos. Isto é, um novo valor armazenado por célula deve ser tratado como uma entidade independente que seja capaz de se atualizar automaticamente baseada no comportamento de um autômato celular genérico que esteja modificando o material que ocupa a célula ligada aquele valor.

4 SIMULAÇÃO USANDO PARÂMETROS FÍSICOS

Os autômatos celulares implementados no simMEMS são em geral idealizados. Como visto pelos exemplos apresentados até aqui, esses processos idealizados permitem simular a fabricação de diversos dispositivos. Apesar de não retratarem de maneira fidedigna todos os detalhes de todos os processos, ainda é possível projetar e entender a sequência de processos necessária para fabricação de dispositivos microeletrônicos bem como de MEMS e MOEMS.

Por outro lado, também é interessante poder simular um processo levanto em conta os parâmetros como temperatura, pressão ou taxa de corrosão e visualizar a influência dessas grandezas no processo. Existem duas maneiras de fazer isso. A maneira mais simples é usar um modelo conhecido que define, por exemplo, a taxa de corrosão de um material em função da concentração da solução corrosiva e da sua temperatura, por exemplo. Em seguida pode-se utilizar um autômato celular como o aqui descrito, de contagem de distância, para resolver o problema de evolução da superfície. A segunda opção é usar um autômato celular específico para o processo. Nesse caso, é utilizado um autômato celular cujo comportamento macroscópico é descrito pela mesma equação que descreve o comportamento do processo a ser simulado.

Nesse sentido, foi estudado um autômato celular que permite incluir parâmetros físicos na simulação da dopagem de um substrato de Si. Esse autômato celular também permite a simulação de processos de difusão-reação, e seu uso para simular um processo de oxidação térmica do Si também foi estudado.

4.1 Caminhada Aleatória

A simulação do processo de difusão no simMEMS é baseada no processo de caminhada aleatória. Em um processo de caminhada aleatória, uma entidade tem uma posição inicial $\vec{P}$. A entidade então se move em incrementos $\overrightarrow{l_n}$ aleatórios. Após N passos a posição da entidade é dada pela equação 4:

$$\overrightarrow{P_N} = \vec{P} + \sum_1^N \overrightarrow{l_n}. \quad (4)$$

O conjunto de deslocamentos $\{\overrightarrow{l_n}\}$ é um conjunto de variáveis aleatórias independentes que definem o caminho tomado pela entidade, que recebe o nome de caminhada aleatória.

Diversos fenômenos podem ser simulados pelo processo de caminhadas aleatórias. Por exemplo, processos como difusão e transferência de calor, entre outros, podem ser simulados usando essa técnica. No caso da difusão, isso é bastante simples de entender. À medida que uma partícula se difunde em um meio, ela interage com os demais átomos que formam o meio. Embora essas interações ocorram de maneira muito bem definida, o resultado final dessas interações é bastante complexo. O resultado final faz com que o movimento da partícula seja aleatório.

Para que uma entidade alcance uma posição $\vec{R}$ qualquer após N passos, é preciso que essa entidade se encontre no ponto $\vec{R}-\vec{r}$ e realize no passo N o deslocamento $\vec{r}$. Dessa maneira, é possível calcular $P_N(\vec{R})$, a probabilidade de se encontrar a entidade em $\vec{R}$ após N passos, usando a equação 5:

$$P_N(\vec{R}) = \int p(\vec{r}|\vec{R}-\vec{r}) P_{N-1}(\vec{R}-\vec{r})\, d^d\vec{r}\ . \tag{5}$$

A equação 5 pode ser simplificada se a probabilidade de se deslocar na direção $\vec{r}$ for independente da posição atual. Nesse caso $p_N(\vec{r}|\vec{R}-\vec{r}) = p_N(\vec{r})$ e a equação resultante é apresentada abaixo:

$$P_N(\vec{R}) = \int p(\vec{r})\, P_{N-1}(\vec{R}-\vec{r})\, d^d\vec{r}\ . \tag{6}$$

A seguir é analisado um caso especial, bastante simplificado, para essa caminhada aleatória em uma dimensão ($\mathrm{d} = 1$). Será mostrado o limite do contínuo, quando $N \to \infty$, para a equação 6. Para auxiliar nessa tarefa, define-se a função $\rho(x,t)$ tal que $\rho(x, N\tau) = P_N(x)$, onde τ é a duração de um passo, para N muito grande e $\rho(x,t)$ é diferenciável em x e t. Em seguida, é preciso encontrar a equação parcial diferencial para $\rho(x,t)$:

$$P_{N+1}(x) = \int p(x-x')\, P_N(x')\, dx'\ . \tag{7}$$

Assumindo $t \gg \tau$, $\sigma^2 = \langle x^2 \rangle = \int x^2 \rho(x) dx$ e que $\langle x_N^2 \rangle \sim \sigma^2 N$, pode-se assumir que a probabilidade $P_N(x)$ varia em uma escala L tal que $L \gg \sigma$. Dessa forma, quando $N \to \infty$ a probabilidade de dar um passo em uma determinada direção, $p(x-x')$, não varia

muito com x. Pode-se então substituir $P_N(x)$ por sua expansão de Taylor. Para facilitar, também é feita a substituição $x' = x - y$, logo:

$$P_{N+1}(x) = \int p(y) P_N(x-y)\, dy$$
$$= \int p(y) \left(\sum_{i=0}^{\infty} \frac{(-y)^i}{i!} \frac{\partial^i P_N(x)}{\partial x^i} \right) dy \,. \tag{8}$$

A equação acima pode então ser simplificada usando os momentos da variável aleatória $p(y)$ que por definição são $m_i = \int y^i p(y) dy$. Fazendo as respectivas substituições e mantendo em mente que $m_0 = 1$, obtém-se a equação abaixo:

$$P_{N+1}(x) = P_N(x) + \sum_{i=1}^{\infty} \frac{(-1)^i m_i}{i!} \frac{\partial^i P_N(x)}{\partial x^i}. \tag{9}$$

Em seguida define-se $D_i = \frac{m_i}{i!\tau}$. Passando $P_N(x)$ para o lado esquerdo da equação e dividindo ambos os lados por τ, obtém-se a equação 10 a seguir:

$$\frac{P_{N+1}(x) - P_N(x)}{\tau} = \sum_{i=1}^{\infty} (-1)^i D_i \frac{\partial^i P_N(x)}{\partial x^i}. \tag{10}$$

Tomando o limite infinitesimal dos passos, isto é, passos com comprimento e duração tão pequenos quanto se queira, obtém-se a equação abaixo, conhecida como a expansão Kramers-Moyall:

$$\frac{\partial \rho(x,t)}{\partial t} \sim -D_1 \frac{\partial \rho(x,t)}{\partial x} + D_2 \frac{\partial^2 \rho(x,t)}{\partial x^2} - D_3 \frac{\partial^3 \rho(x,t)}{\partial x^3} + \cdots \,. \tag{11}$$

Se a relação entre o passo de tempo e o tamanho do passo ao quadrado for definida como sendo constante e mantendo apenas os primeiros elementos da expansão, a equação para convecção-difusão é obtida. Nesse caso, o coeficiente de difusão é dado por D_2 e a velocidade média do deslocamento é dada por D_1 na equação abaixo:

$$\frac{\partial \rho(x,t)}{\partial t} = -D_1 \frac{\partial \rho(x,t)}{\partial x} + D_2 \frac{\partial^2 \rho(x,t)}{\partial x^2}. \tag{12}$$

Analisando o coeficiente de difusão, é possível verificar que o mesmo é ligado ao segundo momento da posição e a duração do passo de tempo. Essa é a relação encontrada para o movimento Browniano (EINSTEIN, 1956), e indica que o deslocamento de uma partícula é proporcional à raiz do tempo de deslocamento, como pode ser verificado abaixo:

$$\langle X^2 \rangle = D_2 2\tau. \tag{13}$$

4.2 Caminhada Aleatória em um Autômato Celular

O conceito de uma caminhada aleatória pode também ser aplicado a autômatos celulares. Nesse caso, as possíveis posições das partículas são limitadas pele rede do autômato celular. Inicialmente, será novamente analisado o caso unidimensional. As células do autômato celular formam uma linha infinita por onde as partículas podem se movimentar.

Na abordagem descrita anteriormente, cada partícula é analisada individualmente e movimentada. A cada passo de tempo τ, uma partícula pode se deslocar uma célula para esquerda ou para direita, com probabilidade p_0 e $1 - p_0$, respectivamente, como mostra a Figura 57.

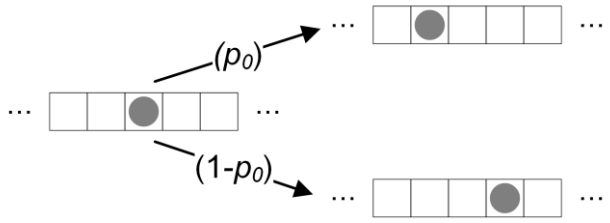


Figura 57 - Distribuições independentes em todas as células.

Essa abordagem não é facilmente implementada através de um autômato celular. Como todas as partículas se movimentam ao mesmo tempo, duas partículas podem optar por se movimentar para mesma célula desocupada ao mesmo tempo. Para limitar o número de partículas em cada célula, é preciso implementar uma lógica que analisa não somente as células vizinhas, mas também as células adjacentes às vizinhas, também chamadas de segundos vizinhos. Essa mudança também implica em uma mobilidade diferenciada em função da concentração local de partículas, o que pode ser problemático.

Uma forma simples de resolver esse problema é utilizar duas variáveis para representar a ocupação de uma célula. Cada célula possui uma variável para indicar se há uma partícula entrando na célula pela lateral esquerda, enquanto a outra variável indica se há uma partícula entrando na célula pela lateral direita da célula. A movimentação aleatória das partículas é obtida através da inversão, com probabilidade p_0, das duas variáveis de cada célula a cada passo de tempo.

Dessa forma, a cada passo de tempo a atualização dos estados segue os seguintes passos. Primeiramente, os valores das duas variáveis são invertidos ou não baseados em uma variável aleatória com probabilidade p_0 de ser verdadeira. Isso é feito para todas as células. Em seguida, o valor das duas variáveis é atualizado copiando o respectivo valor das células vizinhas. A Figura 58 exemplifica o funcionamento do autômato celular para passos de tamanho e duração unitários.

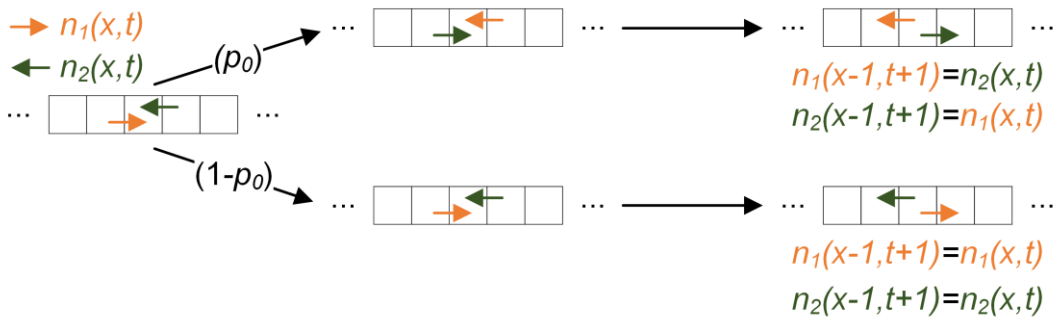


Figura 58 – Funcionamento do autômato celular unidimensional para difusão.

É possível então escrever um conjunto de equações para descrever o comportamento dessas duas variáveis em função da posição da célula e do passo de tempo. Para tanto, é introduzida a variável aleatória $p(x,t)$ que apresenta o valor 1 com probabilidade p_0 e 0 com probabilidade $1 - p_0$.

Uma partícula só pode chegar à célula na posição x no instante t se ela se encontrar na posição $x - 1$ no instante $t - 1$. Dessa forma, o comportamento do autômato celular pode ser descrito através do sistema de equações abaixo. A variável aleatória $p(x,t)$ seleciona qual número de ocupação será repassado às células vizinhas.

$$\begin{bmatrix} n_1(\mathrm{x}+\lambda,\mathrm{t}+\tau) \\ \mathrm{n}_2(\mathrm{x}-\lambda,\mathrm{t}+\tau) \end{bmatrix} = \begin{bmatrix} \mathrm{p}(\mathrm{x},\mathrm{t})\mathrm{n}_1(\mathrm{x},\mathrm{t}) + \big(1-\mathrm{p}(\mathrm{x},\mathrm{t})\big)\mathrm{n}_2(\mathrm{x},\mathrm{t}) \\ \big(1-\mathrm{p}(\mathrm{x},\mathrm{t})\big)\mathrm{n}_1(\mathrm{x},\mathrm{t}) + \mathrm{p}(\mathrm{x},\mathrm{t})\mathrm{n}_2(\mathrm{x},\mathrm{t}) \end{bmatrix}. \qquad (14)$$

Partindo do sistema de equações em 14, que descreve o comportamento microscópico das partículas, é possível encontrar uma equação para descrever o comportamento médio do sistema de forma macroscópica. Inicialmente é preciso descrever o comportamento médio microscópico desse autômato celular. A média do número de ocupação é o resultado de se tirar a média dos números de ocupação em um determinado instante t para diversas simulações do autômato celular a partir de uma mesma condição inicial.

Como a variável aleatória $p(x,t)$ é estatisticamente independente dos valores de ocupação $n_i(x,t)$, a média dos seus produtos é simplesmente o produto das suas médias. Além disso, o valor médio de $p(x,t)$ é p_0. Assim, o comportamento médio microscópico dos números de ocupação desse autômato celular, $N_i(x,t)$, são dados pelo sistema de equações em 15:

$$\begin{bmatrix} N_1(x+\lambda, t+\tau) \\ N_2(x-\lambda, t+\tau) \end{bmatrix} = \begin{bmatrix} p_0 N_1(x,t) + (1-p_0) N_2(x,t) \\ (1-p_0) N_1(x,t) + p_0 N_2(x,t) \end{bmatrix}. \tag{15}$$

Analisando o caso periódico, onde as M células formam um disco de comprimento $L = M\lambda$ de forma que $N_i(r,t) = N_i(r+L,t)$, existem algumas maneiras de se obter o comportamento macroscópico do sistema partindo do sistema apresentado em 15. A abordagem aqui escolhida é escrever $N_i(r,t)$ como uma superposição de ondas planares com a forma $N(r,t) = A(t)e^{ikr}$. Nesse caso, k assume os valores $0, \frac{2\pi}{L}, \dots, \frac{m2\pi}{L}, \dots, \frac{(M-1)2\pi}{L}$ e $A(t)$ é uma função a ser determinada. Substituindo essa nova representação em 15, obtém-se o novo sistema de equações abaixo:

$$\begin{bmatrix} \mathrm{A_1(t+\tau)} \\ \mathrm{A_2(t+\tau)} \end{bmatrix} = \begin{bmatrix} \mathrm{p_0 e^{-ik\lambda}} & \mathrm{(1-p_0)e^{-ik\lambda}} \\ \mathrm{(1-p_0)e^{ik\lambda}} & \mathrm{p_0 e^{ik\lambda}} \end{bmatrix} \begin{bmatrix} \mathrm{A_1(t)} \\ \mathrm{A_2(t)} \end{bmatrix}. \tag{16}$$

Em seguida pode-se escrever as funções $A_i(t)$ como $A_i(t) = A(0)e^{i\omega t}$. Substituindo essa nova forma no sistema em 16, obtém-se então o sistema apresentado em 17:

$$\begin{pmatrix} p_0 e^{-ik\lambda} - e^{i\omega\tau} & (1-p_0)e^{-ik\lambda} \\ (1-p_0)e^{ik\lambda} & p_0 e^{ik\lambda} - e^{i\omega\tau} \end{pmatrix} \begin{pmatrix} A_1(0) \\ A_2(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{17}$$

Para que o sistema tenha uma solução diferente de $A_i(0) = 0$, o determinante da matriz deve ser diferente de zero. Calculando o determinante dessa matriz, é possível também encontrar a relação de ω e k nesse autômato celular, que é a relação de dispersão (CHOPARD; DROZ, 2005; EINSTEIN, 1956), apresentada a seguir:

$$\mathrm{e}^{2\mathrm{i}\omega\tau} - 2p_0 \cos(k\lambda) e^{i\omega\tau} + 2p_0 - 1 = 0. \tag{18}$$

Partindo da equação 18 é possível determinar o valor de $e^{i\omega\tau}$, mostrado abaixo:

$$\begin{aligned} \mathrm{e}^{\mathrm{i}\omega\tau} &= p_0 \cos(k\lambda) \pm \sqrt{p_0^2 \cos^2(k\lambda) - 2p_0 + 1} \\ &= p_0 \cos(k\lambda) \pm (1 - p_0) \sqrt{1 - \frac{p_0^2}{(1 - p_0)^2} \sin^2(k\lambda)}. \end{aligned} \tag{19}$$

O passo final para determinar o limite macroscópico do autômato celular é determinar a evolução temporal do sistema quando o passo de tempo τ e o tamanho da célula τ tendem a zero. A evolução temporal do sistema é determinada pelo termo $e^{i\omega t} = \left[e^{i\omega\tau}\right]^{t/\tau}$. Logo é possível utilizar o resultado obtido na equação 19 para calcular os limites desejados, como mostrado a seguir:

$$\lim_{\tau\to 0,\lambda\to 0} [e^{i\omega\tau}]^{\mathrm{t}/\tau} = \lim_{\tau\to 0,\lambda\to 0} \left[p_0 \cos(k\lambda) \pm (1 - p_0) \sqrt{1 - \frac{p_0^2}{(1 - p_0)^2} \sin^2(k\lambda)} \right]^{t/\tau}. \tag{20}$$

Um desses limites é simples de calcular. Quando se subtrai o segundo termo entre colchetes, o valor dentro dos colchetes tende a $2p_0 - 1$. Quando $\tau \to 0$, esse valor tende a 0, visto que $0 < p_0 < 1$. O limite para o outro caso é um pouco mais complexo, visto que o valor dentro dos colchetes tende a 1, resultando em um limite do tipo 1^∞. Para remover essa incerteza, pode-se tirar o logaritmo da expressão e em seguida aplicar a regra de l'Hôpital para encontrar o valor do limite. Em seguida, mantendo em mente a relação entre τ e λ encontrada na equação 19, é possível chegar ao resultado apresentado em 21:

$$\begin{aligned} \ln\left(\lim_{\tau\to 0,\lambda\to 0} [e^{i\omega\tau}]^{\frac{\mathrm{t}}{\tau}}\right) &= \lim_{\tau\to 0,\lambda\to 0} \ln\left([e^{i\omega\tau}]^{\frac{\mathrm{t}}{\tau}}\right) \\ &= t \lim_{\tau\to 0,\lambda\to 0} \frac{\ln(e^{i\omega\tau})}{\tau} \\ &= t \lim_{\tau\to 0,\lambda\to 0} \partial_\tau \ln(e^{i\omega\tau}) \\ &= t \lim_{\tau\to 0,\lambda\to 0} e^{-i\omega\tau} \partial_\lambda \ln(e^{i\omega\tau}) \frac{d\lambda}{d\tau}. \end{aligned} \tag{21}$$

Finalmente, é possível simplificar ainda mais o valor desse limite substituindo $e^{i\omega\tau}$ pelo valor encontrado na equação 19 abaixo:

$$\begin{aligned} t \lim_{\tau\to 0,\lambda\to 0} e^{-i\omega\tau} \partial_\lambda \ln(e^{i\omega\tau}) \frac{d\lambda}{d\tau} &= t \lim_{\tau\to 0,\lambda\to 0} \partial_\lambda \ln(e^{i\omega\tau}) \frac{d\lambda}{d\tau} \\ = t \lim_{\tau\to 0,\lambda\to 0} \partial_\lambda \ln\left(p_0 \cos(k\lambda) \pm (1 - p_0) \sqrt{1 - \frac{p_0^2}{(1 - p_0)^2} \sin^2(k\lambda)} \right) \frac{d\lambda}{d\tau}. \end{aligned} \tag{22}$$

4.3 Processos de Difusão-Reação

Na seção anterior, foi descrita uma maneira de obter o comportamento macroscópico de um autômato celular simples unidimensional. Durante o desenvolvimento das equações, foi tomado o valor médio do número de ocupação nas células. Esse valor médio pode ser tomado de diversas maneiras. Uma delas é reduzindo a resolução do sistema, seja texzsxzxmporal ou espacial, e tomando o valor médio de diversas células ou de diversos passos de tempo. Outra alternativa é realizar a simulação a partir das mesmas condições iniciais diversas vezes e tomar como resultado final a média das diversas simulações. Se o valor médio não for tomado, o resultado final se torna bastante ruidoso, e pode ser bem diferente do resultado esperado. Outro potencial problema é a imposição do princípio de exclusão ao sistema.

Uma alternativa para resolver o problema de ruído é o uso de um autômato celular baseado no método lattice Boltzmann. Nesse caso, os autômatos celulares usam valores reais na simulação. Nessa abordagem as equações de Boltzmann descrevendo o sistema são simuladas diretamente. O autômato celular passa a tratar com distribuições de partículas e não com partículas individuais. Dessa forma se trabalha com o comportamento médio do sistema ao invés de se tirar a média do comportamento posteriormente. Isso diminui o trabalho de simulação em troca da perda do caráter discreto dos estados do sistema. O método lattice Boltzmann apresenta maior flexibilidade na configuração de parâmetros para simulação. É preciso, no entanto, tomar cuidado na escolha dos parâmetros visto que alguns valores podem levar a instabilidades na simulação. Além disso, por tratar o problema através de distribuições de partículas ao invés de partículas individuais, é preciso ter certeza de que essa aproximação é válida. No caso da difusão, isso é equivalente a garantir que o número médio de partículas é grande o suficiente para que essa afirmação seja verdadeira.

Para aplicações em microeletrônica pode-se supor uma concentração de dopantes na faixa de $1x10^{15}$ átomos por cm^3. Nesse caso, uma célula de 10 nm contém, em média, 0,001 átomos. Assim, embora a abordagem pelo método lattice Boltzmann seja válida em casos como o exemplo de fabricação do CI CD4001B, onde uma célula possui em média 216 átomos do dopante, essa abordagem para de ser válida quando o tamanho das células cai abaixo de 100 nm.

A alternativa aqui adotada foi utilizar um outro tipo de autômato celular, que tenta unir as vantagens do método lattice Boltzmann às vantagens dos métodos que trabalham com partículas discretas. A base dessa abordagem é simplesmente remover a imposição do princípio de exclusão. Dessa forma, cada célula pode ter até N partículas. A remoção dessa imposição também remove a necessidade de se usar duas distribuições de partículas. A cada passo de tempo, as partículas presentes em uma célula são espalhadas de maneira aleatória, semelhante a uma caminhada aleatória. Para diminuir o custo computacional de cada passo, células com um número suficientemente alto de partículas podem aproximar o número de partículas que segue em cada possível direção através de distribuições, semelhante ao que se faz no método lattice Boltzmann. De forma rigorosa, não se trata mais de um autômato celular. Como uma célula pode possuir um número qualquer de partículas, o número de possíveis estados é infinito. No entanto, por trabalhar apenas com números inteiros, não sofre de problemas de instabilidade por erros numéricos, como é o caso de alguns autômatos celulares contínuos.

Cada célula do autômato celular possui uma variável que indica quantas partículas estão dentro da célula. A cada passo de tempo, para cada partícula dentro de uma célula é escolhida aleatoriamente uma direção em que a partícula irá seguir dentro das possíveis direções, que inclui a possibilidade da partícula continuar na célula atual. Em seguida o número de partículas é atualizado considerando quantas partículas chegam das células vizinhas e quantas partículas ficaram na célula.

Em três dimensões, existem sete possíveis direções: ficar na célula, para cima, para baixo, para esquerda, para direita, para frente ou para trás. Uma variável aleatória é utilizada para definir qual direção cada partícula deve tomar. A probabilidade de uma partícula continuar na célula é chamada de p_0. As probabilidades de se seguir nas demais direções $\overrightarrow{u_i}$ são definidas como p_i. As direções $\overrightarrow{u_i}$ são definidas de tal forma que $\overrightarrow{u_3} = \overrightarrow{u_1} \times \overrightarrow{u_2}$ e $\overrightarrow{u_i} = -\overrightarrow{u_{i+3}}$ com $1 \leq i \leq 3$, como mostrado na Figura 59.

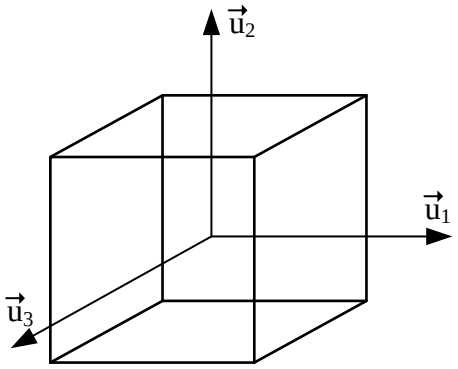


Figura 59 – Direções do autômato celular multipartículas em 3D.

O funcionamento básico do autômato celular é determinar o destino de cada partícula e mover a partícula para o seu destino. Como se trata de um autômato celular, todas as partículas se movimentam no mesmo instante. É possível implementar esse autômato de forma semelhante a autômatos celulares como o FHP e HPP, onde a lógica do autômato celular é implementada em dois passos (CHOPARD; DROZ, 2005). Primeiramente, a cada passo de tempo, itera-se sobre todas as partículas em uma posição $\vec{r}$ e determina-se qual dos sete possíveis destinos a partícula irá tomar. Em seguida o número de partículas em cada célula é atualizado para todas as células utilizando as informações de destino das partículas em células vizinhas. Dessa forma, o número de ocupação $n(\vec{r},t)$ de uma posição $\vec{r}$ no instante t pode ser descrito pela equação 23 a seguir:

$$n(\vec{r},t+\tau) = \sum_{l=1}^{n(\vec{r},t)} p_{0l}(\vec{r},t) + \sum_{k=1}^{6}\left[\sum_{l=1}^{n(\vec{r}+\lambda\overrightarrow{e_k},t)} p_{kl}\left(\vec{r}+\lambda\overrightarrow{e_{k_i}},t\right)\right]. \qquad (23)$$

Na equação 23 $p_{kl}(\vec{r},t)$ é uma variável booleana estocástica que seleciona se a partícula l da posição $\vec{r}+\lambda\overrightarrow{e_{k_i}}$ irá para a posição $\vec{r}$, tendo o valor 1 com probabilidade p_k. O coeficiente $k_i = (k+3)mod\ 7$ é o índice da direção oposta de k. Cada partícula pode seguir para apenas um destino, para toda partícula l é preciso que $\sum_{k=0}^{6} p_{kl}(\vec{r},t) = 1$.

A ocupação macroscópica de uma posição $N(\vec{r},t) = \langle n(\vec{r},t)\rangle$ é obtida novamente calculando o valor médio da ocupação para um conjunto de sistemas equivalentes, isto é, sistemas com os mesmos valores para $n(\vec{r},0)$ e p_k mas não para as variáveis

aleatórias booleanas $p_{kl}(\vec{r},t)$. Novamente, tem-se que $\langle\sum_{l=1}^{n(\vec{r},t)} p_{kl}(\vec{r},t)\rangle = p_k N(\vec{r},t)$, que aplicado à equação 23 gera a equação 24 abaixo:

$$N(\vec{r},t+\tau) = p_0 N(\vec{r},t) + \sum_{k=1}^{6} p_k N\left(\vec{r} + \lambda\overrightarrow{e_{k_l}}, t\right). \tag{24}$$

Assume-se agora que N varia lentamente ao longo da rede. Como se trata de simular um processo físico, no caso difusão, onde as partículas não se movimentam muito rapidamente, trata-se de uma suposição válida. Nesse caso, é possível aplicar uma expansão de Taylor, para o tempo e espaço, aos termos da equação 24 resultando na equação 25 a seguir:

$$\begin{gathered}\tau\partial_t N(\vec{r},t) + \frac{\tau^2}{2}\partial_t^2 N(\vec{r},t) + \mathcal{O}(\tau^3) \\ = \lambda[(p_4-p_1)\overrightarrow{e_1} + (p_5-p_2)\overrightarrow{e_2} + (p_6-p_3)\overrightarrow{e_3}]\cdot\nabla N(\vec{r},t) \\ + \frac{\lambda^2}{2}[(p_1+p_4)\overrightarrow{e_1}\cdot\nabla^2\mathrm{N}(\vec{\mathrm{r}},\mathrm{t}) + (p_2+p_5)\overrightarrow{e_2}\cdot\nabla^2\mathrm{N}(\vec{\mathrm{r}},\mathrm{t}) + (p_3+p_6)\overrightarrow{e_3}\cdot\nabla^2\mathrm{N}(\vec{\mathrm{r}},\mathrm{t})] \\ + \mathcal{O}(\lambda^3).\end{gathered} \tag{25}$$

Para garantir que a escolha das direções em que as partículas se movimentam pela rede não será visível, é preciso impor a condição representada na equação 26:

$$p_1 + p_4 = p_2 + p_5 = p_3 + p_6 = \frac{1-p_0}{3}. \tag{26}$$

Isto é, o autômato celular terá um comportamento isotrópico em uma rede cúbica se a condição apresentada em 26 for mantida. Note que a condição acima não impõe que as probabilidades em um mesmo eixo sejam iguais, mas apenas que sua soma seja igual às somas dos demais eixos. Tornando $p_1$diferente de p_4, por exemplo, é possível simular o efeito de uma velocidade de advecção no sistema. Dessa forma é possível simular não somente o fenômeno de difusão, mas também o efeito de outras influências, como um fluxo no meio ou o efeito de um campo elétrico sobre íons e difundindo.

A equação 25 pode então ser simplificada ainda mais definindo o vetor $\vec{V}$, que representa a velocidade de advecção, e o coeficiente de difusão D. Lembrando que pela definição das direções $\overrightarrow{e_l}$, vale a relação $(\overrightarrow{e_1}\cdot\nabla)^2 + (\overrightarrow{e_1}\cdot\nabla)^2 + (\overrightarrow{e_3}\cdot\nabla)^2 = \nabla^2$, os valores de $\vec{V}$ e D são apresentados abaixo:

$$\vec{V} = \frac{\lambda}{\tau}[(p_4-p_1)\overrightarrow{e_1} + (p_5-p_2)\overrightarrow{e_2} + (p_6-p_3)\overrightarrow{e_3}], \tag{27}$$

$$D = \frac{\lambda^2}{6\tau}(1 - p_0). \tag{28}$$

A equação 25 toma então a forma abaixo:

$$\partial_t N(\vec{r}, t) + \frac{\tau}{2}\partial_t^2 N(\vec{r}, t) + \mathcal{O}(\tau^2) = \vec{V} \cdot \nabla N(\vec{r}, t) + \mathrm{D}\nabla^2\mathrm{N}(\vec{\mathrm{r}}, \mathrm{t}) + \mathcal{O}(\lambda^3). \tag{29}$$

O passo final é considerar o que ocorre quando o tamanho da célula λ e a duração do passo de tempo τ tendem a zero. Ao tomar esse limite, novamente se considera que a relação λ^2/τ é constante, como se espera para um processo de difusão e como foi visto previamente. Para que o limite exista, também é preciso impor uma nova condição sobre as probabilidades p_i. Caso a velocidade de advecção $\vec{V}$ seja não nula, a intensidade dessa velocidade deve diminuir proporcionalmente ao tamanho da célula. Em resumo, ao diminuir o passo de tempo τ pela metade, λ, $p_3 - p_1$, $p_4 - p_2$ e $p_6 - p_3$ devem também diminuir por um fator de quatro. Nesse caso, o limite macroscópico é dado pela equação 30:

$$\partial_t N(\vec{r}, t) = \vec{V} \cdot \nabla N(\vec{r}, t) + \mathrm{D}\nabla^2\mathrm{N}(\vec{\mathrm{r}}, \mathrm{t}). \tag{30}$$

Pelo visto acima, o comportamento macroscópico do autômato celular descrito tem o mesmo comportamento que um processo de difusão onde o coeficiente de difusão não varia com a posição. Como descrito anteriormente, a cada passo de tempo, para cada partícula, é preciso definir o destino dessa partícula. A vantagem do autômato celular multipartículas sobre o autômato celular que impõe o princípio da exclusão vem da possibilidade de se ter muitas partículas em uma mesma célula. Quando cada célula tem diversas partículas, o ruído é reduzido significativamente. Por outro lado, um grande número de partículas implica em um custo de processamento muito maior. Se uma célula possui 1000 partículas, será preciso repetir o processo 1000 vezes. Para um número grande de células, isso pode se tornar proibitivo rapidamente.

É possível, no entanto, limitar o processamento quando se tem muitas partículas. Para um número razoavelmente grande de partículas, o número de partículas que segue para um determinado destino a cada passo possui uma distribuição normal. De forma mais específica, dados dois possíveis destinos para n partículas com probabilidade p e $(1 - p)$, respectivamente, o número de partículas que segue para o destino com probabilidade p pode ser aproximado por uma distribuição normal com média pn e variância $np(1 - p)$(SCHADER; SCHMID, 1989). Assim, é possível implementar uma

lógica específica para o caso onde a célula tem muitas partículas que é substancialmente mais rápido e apresenta resultados idênticos.

Dado o exposto previamente, a lógica para esse caso de muitas partículas é a seguinte. Primeiramente separa-se as n partículas em uma célula em dois conjuntos: o de partículas que fica na célula e o de partículas que saem da célula. Para tanto, uma variável aleatória n_0 com valor médio np_0 e variância $np_o(1-p_0)$ é gerada. Esse é o número de partículas que continuam na célula. Em seguida, as $n-n_0$ partículas restantes são divididas em três conjuntos, referentes aos três possíveis eixos. As probabilidades de cair em cada um desses conjuntos são iguais, visto que é preciso manter a relação imposta na equação 26.

A divisão realizada da mesma forma, uma $n_{1,4}$ variável aleatória com distribuição normal com média $1/3(n-n_0)$ e variância $2/9(n-n_0)$ é gerada. Esse valor representa o número de partículas que segue na direção $\overrightarrow{e_1}$ ou na direção $\overrightarrow{e_4}$. As partículas restantes são então divididas em dois grupos novamente equiprováveis, gerando uma variável aleatória $n_{2,5}$ (o número de partículas que seguem para as direções $\overrightarrow{e_2}$ e $\overrightarrow{e_5}$) com média $1/2(n-n_0-n_{1,4})$ e variância $1/4(n-n_0-n_{1,4})$. Isso também gera a terceira variável aleatória $n_{3,6}$ que indica quantas partículas seguem para as direções $\overrightarrow{e_3}$ e $\overrightarrow{e_6}$.

O passo final é separar os valores $n_{1,4}$, $n_{2,5}$ e $n_{3,6}$ entre os dois respectivos destinos para cada direção. Caso o vetor $\vec{V}$ seja nulo, novamente serão dois conjuntos equiprováveis. Caso contrário, será preciso calcular a probabilidade relativa para cada direção, e.g. $p_1/(p_1+p_4)$ é a probabilidade de uma partícula no conjunto $n_{1,4}$ seguir na direção $\overrightarrow{e_1}$.

As variáveis aleatórias com distribuição normal geradas são valores reais que são arredondadas para um número inteiro. Além disso, um teste de sanidade é aplicado, para garantir que os valores são sempre maiores ou iguais a zero e menores que o número máximo de partículas no conjunto sendo dividido. Finalmente, a aproximação por uma distribuição Gaussiana só é adotada quando o número de partículas é suficientemente grande. No simMEMS, essa condição é que o conjunto de partículas tenha mais do que 40 partículas (CHOPARD; DROZ, 2005).

É possível simular a difusão independente de mais de uma espécie usando o mesmo conjunto de regras, mas uma distribuição separada. Em se tratando de mais de uma espécie, pode-se adicionar mais um passo na simulação para simular reações entre as espécies se difundindo. A simulação da reação é dependente da reação em si; o autômato celular não simula a interação entre as espécies, apenas trata a reação entre as espécies como um processo aleatório. Fatores como a concentração dos reagentes e do produto final também podem ser considerados. Se distribuições específicas para os produtos das reações forem mantidas, também é possível simular uma reação química reversível, bem como simular a difusão da espécie gerada. Se o intuito é apenas medir a taxa de consumo dos reagentes, essa concentração pode eventualmente ser desprezada.

Tomando como exemplo a reação não reversível A + 2B → C, e ignorando a espécie resultante C, existem duas concentrações independentes: a concentração da espécie A e a da espécie B. Define-se então p_r como a probabilidade de, estando presentes um elemento da espécie A e dois elementos da espécie B em uma mesma célula, a reação para gerar C ocorrer.

Considera-se que a cada passo, todos os elementos A dentro de uma célula podem reagir com todos os possíveis pares de elementos B dentro da mesma célula apenas. Existem $C_B = \frac{N_B!}{2!(N_B-2)!}$ combinações de elementos do tipo B em uma célula, onde $N_B \geq 2$ é o número de elementos da espécie B na célula. Se $N_B < 2$ a reação não pode ocorrer. Da mesma forma, existem $C_A = \frac{N_A!}{1!(N_A-1)!}$, com $N_A \geq 1$, possíveis combinações de um elemento do tipo A. Define-se então o valor $N_{máximo}$ como o número de todas as possíveis combinações de 1 elemento da espécie A com 2 elementos da espécie B.

$$N_{máximo} = C_A C_B \tag{31}$$

Cada elemento da espécie A presente na célula pode reagir com qualquer um dos possíveis pares de elementos da espécie B presentes na mesma célula. No entanto, após essa reação ocorrer, tanto o par quanto o elemento não poderão mais reagir. Dessa forma, o número de reações que efetivamente ocorre será necessariamente inferior a $N_{máximo}$. A terceira parte da simulação de um passo do autômato celular se resume então a escolher uma variável aleatória booleana com valor 1 com

probabilidade p_r e 0 com probabilidade $(1 - p_r)$. Se a variável aleatória for 1, o número de elementos da espécie A é decrementado por 1 e o número de elementos da espécie B é decrementado em 2. Esses elementos foram, dessa forma, consumidos para gerar o elemento da espécie C. A geração da variável aleatória é repetida $N_{máximo}$ vezes ou até o número de elementos das espécies A e B ficar menor que o necessário para realizar a reação.

4.3.1 Detalhes de Implementação

A implementação do autômato celular no simMEMS é baseado em parte na biblioteca Boost. A implementação do autômato celular multipartículas requer o uso de um gerador de números aleatórios para a escolha de destinos. A biblioteca Boost fornece diversos geradores de números aleatórios. O gerador de números aleatórios utilizados é a especialização mt19937, que usa o algoritmo Mersenne Twister (MATSUMOTO; NISHIMURA, 1998). O gerador de números aleatórios é usado em conjunto com distribuições para simulação com um grande número de células, usando a classe *normal_distribution*. Quando as partículas são tratadas individualmente, o gerador é usado diretamente para gerar um valor entre 0,0 e 1,0. O destino é escolhido baseado na faixa em que o valor cai. As probabilidades não são escolhidas pelo usuário para garantir que seus valores não serão demasiadamente pequenos. O coeficiente de difusão só pode ser modificado através do tamanho da célula e da duração do passo de tempo. Atualmente nenhum processo permite utilizar uma velocidade de advecção. No simMEMS, um valor fixo de p_0= 0,5 é utilizado. A escolha foi feita simplesmente para evitar que qualquer uma das probabilidades envolvidas fosse demasiadamente pequena (a probabilidade para cada direção é de aproximadamente 0,083).

O estado de uma célula é composto por 7 valores inteiros, representando uma distribuição de partículas em seus 7 possíveis destinos. Para simulação no simMEMS, portanto, cada célula precisa armazenar sete variáveis do tipo unsigned int a mais. Um vetor com tamanho 7 x D_x x D_y x D_z, onde D_x, D_y e D_z são as dimensões do volume de células, é alocado dinamicamente quando um processo de dopagem é iniciado pela primeira vez.

Diferentemente dos demais processos descritos até aqui, o processo de dopagem não se limita a uma superfície. Dessa forma, seria necessário armazenar duas cópias do estado das células para atualização, visto que a cada passo o estado de todas as células deve ser atualizado no mesmo instante. No entanto, é possível limitar a memória necessária à apenas uma cópia do estado, em um esquema semelhante ao padrão A-A para o método Lattice-Boltzmann (BAILEY et al., 2009).

Para tanto, é preciso notar que pela equação 24, o número de partículas presentes em uma célula no instante t que continuam na célula no passo seguinte, em $t + \tau$, só é utilizado para atualização do estado da própria célula. Esse valor não influencia a atualização do estado de nenhuma célula vizinha e pode, portanto, ser sobrescrito durante a atualização do estado da própria célula. A implementação adotada faz uso dessa característica do autômato celular.

Cada célula possui um vetor de sete números inteiros que armazenam o número de células que irá seguir para os sete possíveis destinos no passo seguinte. O número de partículas que irão ficar na célula é armazenado na posição de índice 0 no vetor. Inicialmente, o número total de partículas na célula é armazenado na posição de índice 0 do vetor. Usando as regras descritas acima, esse conjunto de partículas é separado entre os sete possíveis destinos e o vetor é totalmente sobrescrito. Isso é feito para todas as células. O número de partículas na célula passa a ser então a soma dos sete valores nesse vetor.

A segunda parte da simulação de um passo é a movimentação das partículas. Para cada célula então se aplica a equação 24. O resultado é o número total de partículas na célula no passo seguinte, que é novamente armazenado na posição de índice 0 no vetor. Essa posição só é usada na soma da própria célula, como já dito, e seu valor pode ser modificado sem problemas. Após o término da segunda parte da simulação para todas as células, tem-se novamente a situação descrita inicialmente, onde a posição de índice 0 do vetor contém o número total de partículas na célula, que podem então ser espalhadas entre os possíveis destinos.

4.4 Validação do Modelo para Difusão

Na seção anterior, foi desenvolvida a equação que descreve o comportamento macroscópico do autômato celular multipartículas. O uso mais simples para esse autômato celular é a simulação de um processo simples de difusão, como o utilizado para formar um poço N+, por exemplo.

É possível calcular uma solução analítica para diversos exemplos de difusão, como uma fonte de concentração fixa e um poço ou para uma concentração pontual inicial. Para validar nosso autômato celular, foram comparados os resultados obtidos com os resultados teóricos. O primeiro exemplo explorado é o de uma concentração pontual inicial que se espalha por um volume infinito. Primeiramente é apresentada de forma resumida a solução analítica, seguida da comparação com os resultados obtidos pelo autômato celular para uma simulação análoga. Esse exemplo é semelhante ao que ocorre durante o passo de recozimento após uma implantação iônica, sendo, portanto, uma comparação interessante.

Seja $c(x,t)$ a função que descreve a concentração de dopantes em função da posição x e do tempo t. Impõe-se então as seguintes condições de contorno:

$$c(x,0) = N\delta(x); \tag{32}$$

$$\int_{-\infty}^{+\infty} c(x,t)dx = N. \tag{33}$$

A função $c(x,t)$ descreve então uma concentração inicial de N partículas na posição $x = 0$ no instante $t = 0$. Além disso, o número de partículas se mantém constante. A solução para a função $c(x,t)$ assume então a forma descrita na equação 34 (URSELL, 2007) para $t > 0$:

$$c(x,t) = N\frac{1}{\sqrt{4Dt}}e^{-x^2/4Dt}. \tag{34}$$

A solução para 3 dimensões pode ser obtida multiplicando a solução para uma dimensão. Nesse sentido, a difusão tridimensional se assemelha a três difusões independentes, cada uma ocorrendo em uma dimensão (URSELL, 2007). A função $c(x,y,z,t)$ assume então a forma descrita na equação 35 para $t > 0$:

$$c(x,y,z,t) = N\frac{1}{(4Dt)^{3/2}}e^{-\frac{(x^2+y^2+z^2)}{4Dt}}. \quad (35)$$

A equação 35 tem a forma de um pico que diminui de amplitude e se alarga com o passar do tempo. A Figura 60 mostra de maneira qualitativa a evolução de uma curva semelhante à equação 35 para três instantes de tempo.

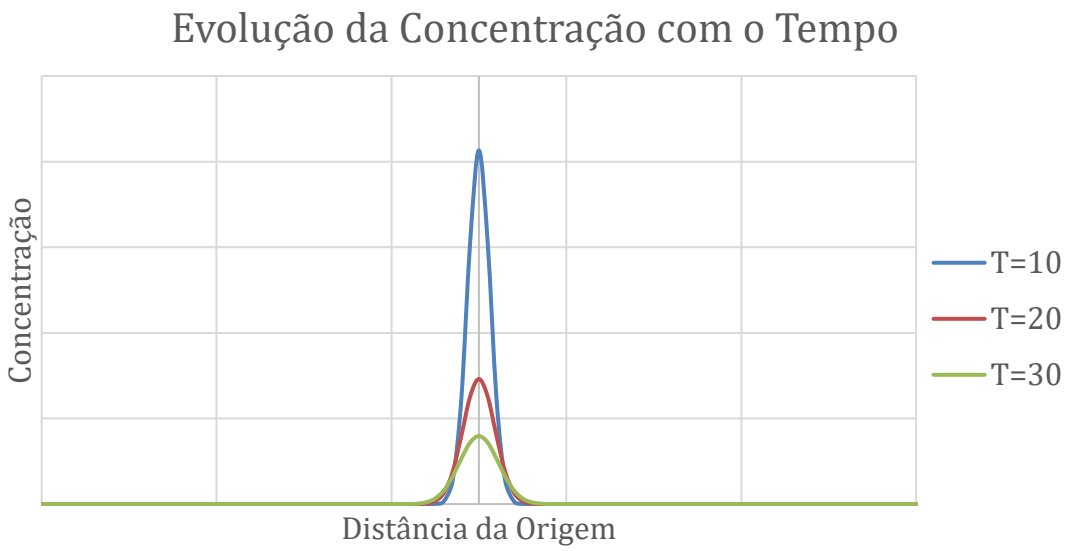


Figura 60 - Evolução temporal da resposta.

Como comparação, a difusão de um conjunto de 700000 partículas inicialmente localizadas na célula de índices (50,50,50) de um bloco de 100x100x100 células foi simulada. Esse número de partículas garante que durante a simulação células nos dois modos de simulação sejam encontradas. A Tabela 4 mostra os valores dos parâmetros usados para simulação.

Tabela 4 – Valor das variáveis usadas para simulação de difusão pontual.

Variável	**Valor**
D	$2{,}6\times10^{-14}\mathrm{cm}^2/\mathrm{s}$
τ	0,1 µm
τ	320,513 s
p_0	0,5
Passos	99
N	$7{,}0\times10^5$

Após 99 passos, o número de partículas em cada célula foi extraído. A simulação foi realizada 5 vezes, e o valor médio de células em cada célula, bem como seu desvio padrão, foram calculados. Isso foi feito para diminuir o ruído e permitir uma melhor comparação com a solução analítica. O número de passos foi escolhido para evitar que partículas cheguem às bordas da estrutura, o que permite ignorar as condições de contorno, visto que o substrato não é infinito. A soma do número de partículas confirma que nenhuma partícula chegou às bordas. A Figura 61 mostra o resultado da dopagem visualizado dentro do simMEMS.

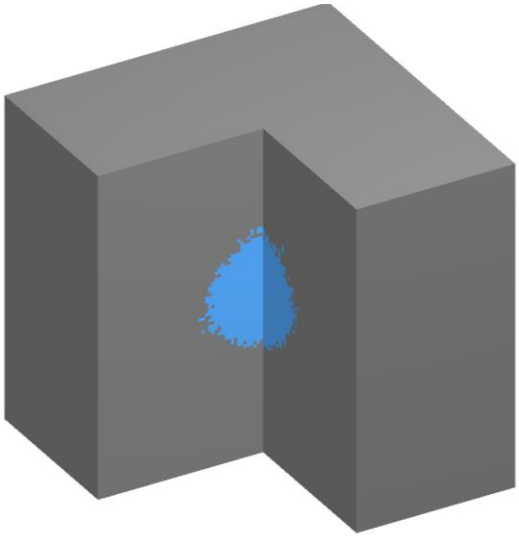


Figura 61 – Visualização de um corte no substrato dopado em um dos experimentos. A figura mostra apenas os materiais ocupando as células, com as células em cinza sendo Si não dopado e as células azuis sendo Si dopado.

O simMEMS permite também visualizar as concentrações de dopantes dento das células. Ao visualizar a concentração o software também disponibiliza uma escala para as cores. A escala abaixo varia de 0 a 680, que é a maior concentração de partículas em uma célula. Pode-se notar assim que o número de partículas por célula cai rapidamente com o andar da difusão.

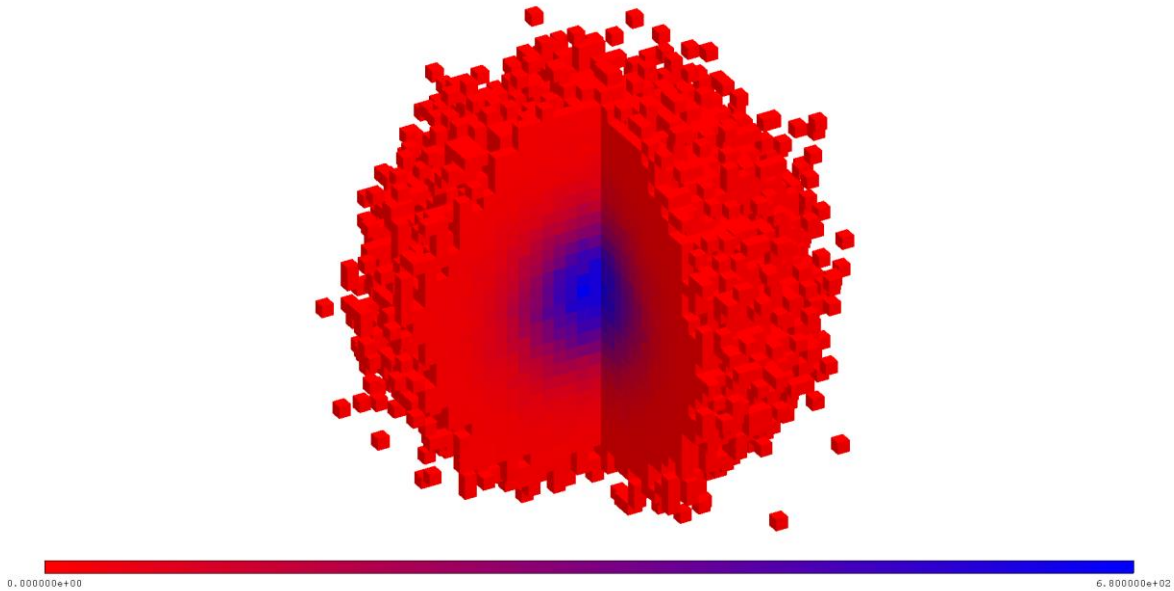


Figura 62 – Concentração de dopantes após uma simulação com escala variando de 0 (vermelho) a 680 (azul).

O gráfico da concentração ao longo do plano XY passando por Z=50 é apresentado na Figura 64. A Figura 63 mostra o plano do gráfico. É possível verificar que as curvas apresentam o formato esperado, uma curva de Gauss com concentração máxima no centro. É possível também visualizar o ruído inerente a esse tipo de simulação.

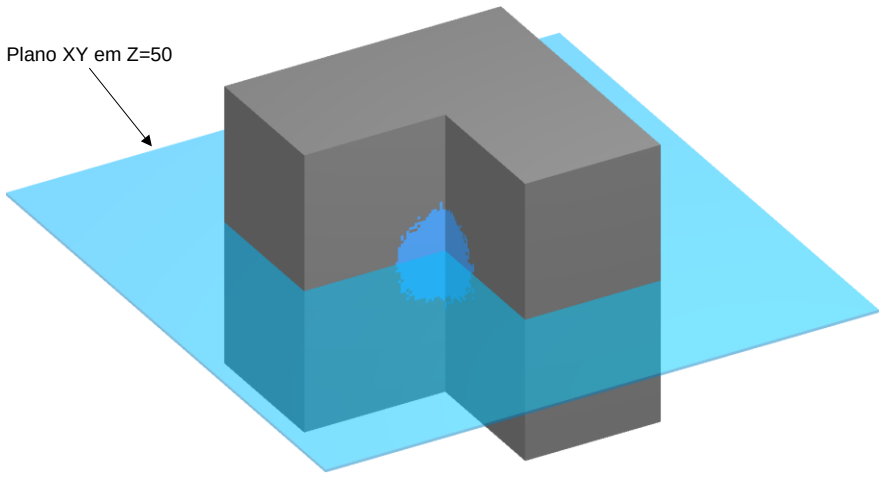


Figura 63 – Visualização do plano XY em Z=50.

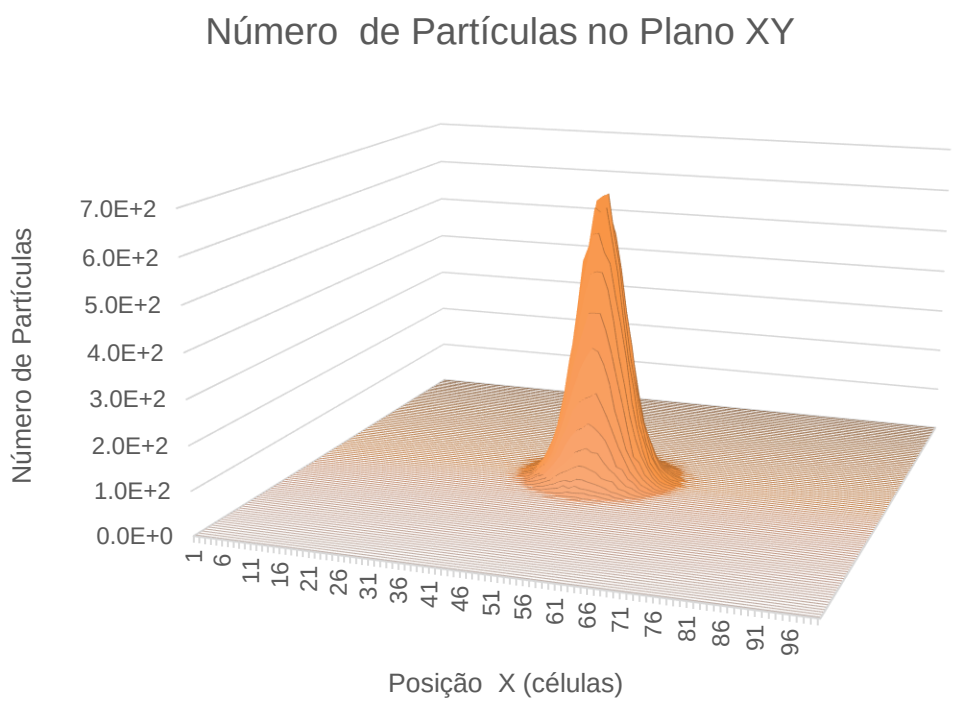


Figura 64 - Resultado da simulação de 99 passos de difusão a partir de um ponto. São apresentados os números de partículas por célula no plano XY passando por Z=50. Cada série representa uma coordenada Y de 0 a 99 e o seixo horizontal apresenta a posição X, em células.

Há uma simetria nos resultados obtidos, o que é esperado. Na Figura 65 são comparados os resultados obtidos com a solução teórica para algumas direções. Por simplicidade são apresentados apenas a região com alguma partícula. Visto que se trata de uma curva simétrica ao redor da origem das partículas em X=Y=Z=50, apenas metade da curva foi reproduzida.

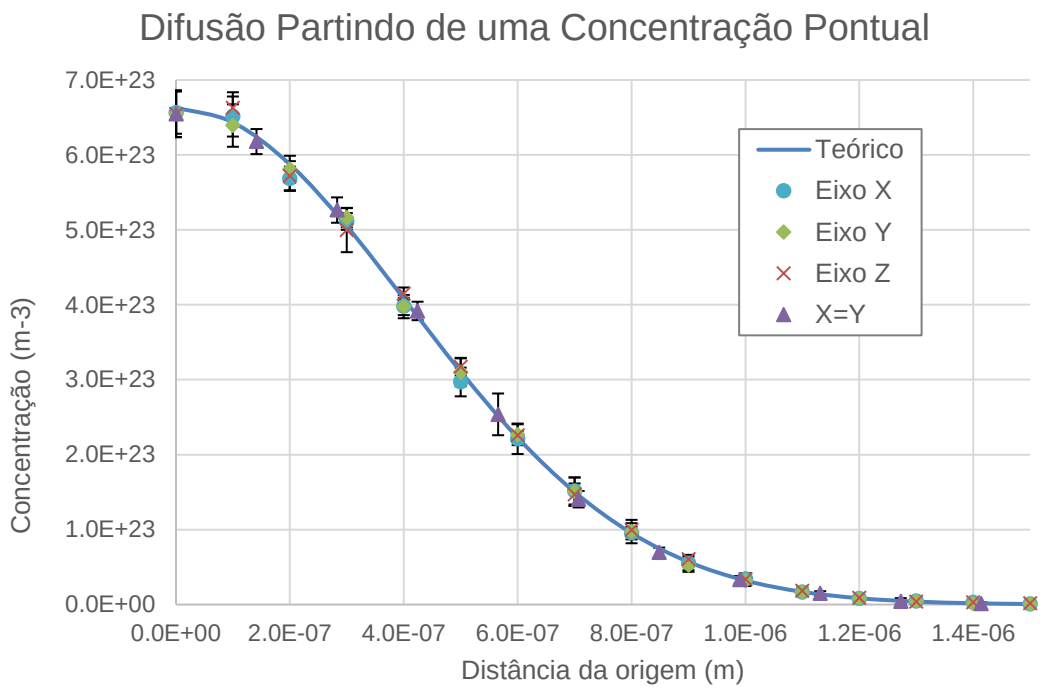


Figura 65 - Comparação entre resultados simulados e o modelo removendo os valores nulos.

A Figura 65 mostra também que ao manter a relação entre as probabilidades expressas na equação 26, esse autômato celular não sofre de anisotropias devido à rede cúbica utilizada. Alguns autômatos celulares, como o HPP e FHP sofrem com anisotropia devido à rede quadrada e hexagonal utilizadas, respectivamente (CHOPARD; DROZ, 2005).

Outro processo de interesse na microeletrônica é a dopagem térmica, onde a superfície da lâmina é exposta ao dopante. Um exemplo teórico que se assemelha a essa situação é o de um plano infinito exposto a uma concentração fixa na superfície, a concentração em função da profundidade no substrato e do tempo é dada por:

$$c(x,y,z,t) = C\ erfc\left(\frac{z}{2\sqrt{Dt}}\right). \tag{36}$$

Na equação 36, C é a concentração na superfície do plano, Dé o coeficiente de difusão do meio e $erfc$ é a função erro complementar. Para simular uma situação semelhante no software, uma pequena modificação foi feita para impor condições de contorno cíclicas nas laterais do substrato. Assim, as células da borda esquerda do substrato têm como vizinhas à esquerda as células da borda direita do substrato, e

vice-versa. O mesmo ocorre com as células à frente e ao fundo do substrato. Dessa maneira é possível simular um substrato infinito com um número finito de células.

As células na superfície do substrato têm uma concentração fixa de partículas C. Foram simulados 100 passos no autômato celular em um substrato de 10x10x100 células de Si. Ao final da simulação, a média do valor de partículas em todas as 100 células de cada uma das 100 colunas foi calculada. A partir do valor médio de partículas em cada célula foi calculada uma concentração média para a respectiva profundidade. Os valores usados para simulação são apresentados na Tabela 5. A concentração foi definida para que durante a simulação existam células operando nos dois modos de simulação descritos previamente.

Tabela 5 – Valor das variáveis usadas para simulação de difusão em um plano infinito.

Variável	**Valor**
D	$2{,}6\times10^{-14}\ \mathrm{cm}^2/\mathrm{s}$
λ	$0{,}1\ \mu\mathrm{m}$
τ	$320{,}513\ \mathrm{s}$
p_0	$0{,}5$
Passos	100
C	$4{,}0\times10^{20}\ \mathrm{cm}^{-3}$

A Figura 66 mostra os resultados da simulação realizada. Neste caso apenas uma simulação foi realizada visto que os dados das diversas colunas podem ser usados para calcular a média para o plano. A imagem à esquerda da Figura 66 mostra também as bordas das células; pode-se verificar que um bloco de 10x10 células foi simulado. A imagem à esquerda mostra novamente a visualização da concentração por célula de dopantes.

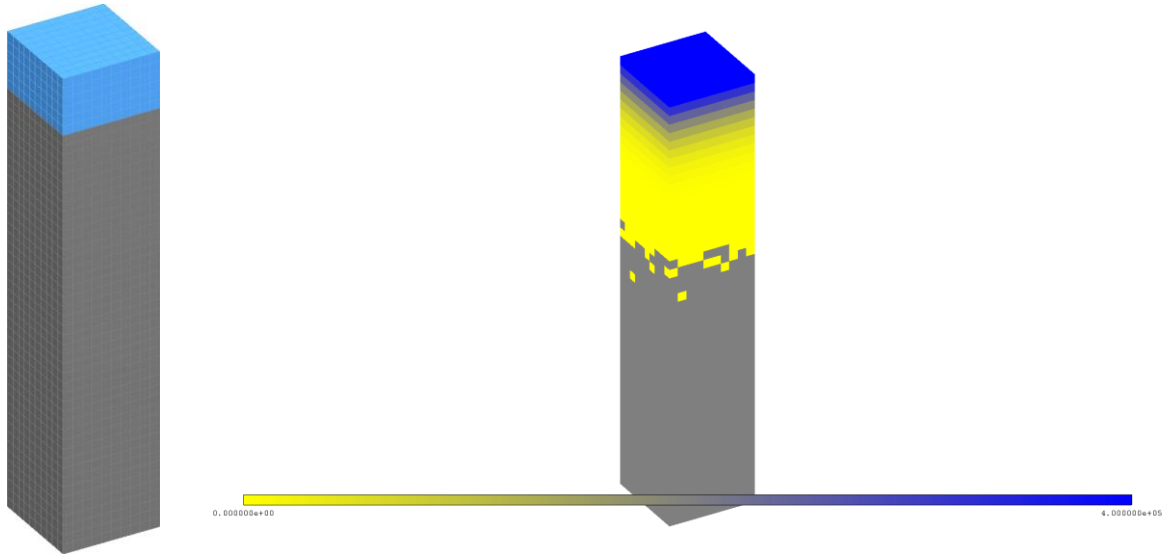


Figura 66 - Resultados da simulação de uma dopagem em um plano infinito no simMEMS. À esquerda pode-se ver o substrato após a dopagem. A escala da imagem à direita varia de zero (amarelo) a $4x10^5$ (azul) e mostra a segunda escala de cores disponível no software.

A comparação entre os valores médios e a solução teórica para esse caso é apresentada na Figura 67. Novamente, os valores se aproximam bastante da solução analítica. Os resultados obtidos pela simulação estão próximos do esperado, estando dentro do desvio padrão calculado.

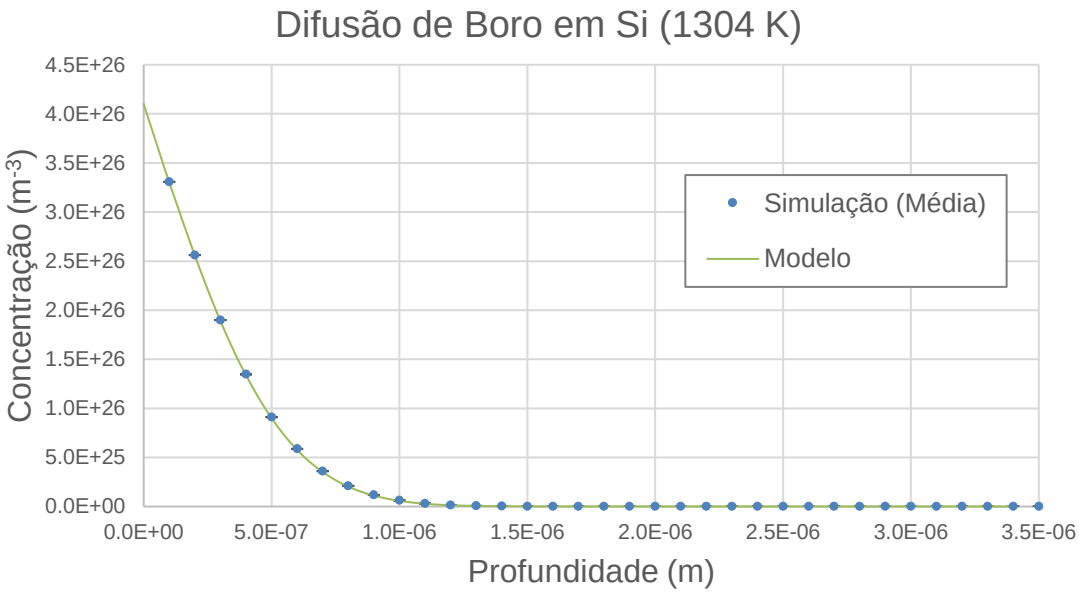


Figura 67 – Comparação entre resultados experimentais e modelo para difusão em um plano infinito com concentração constante.

Esses resultados também demonstram que para a escala em que se pretende trabalhar, com células de tamanhos abaixo de 1µm, o autômato celular apresenta resultados muito próximos aos teóricos. A duração de um passo de tempo é bastante alta, cerca de 5 minutos, para um valor de coeficiente de difusão na ordem de grandeza de um processo de dopagem em microeletrônica. Os valores usados para o coeficiente de difusão são os que seriam utilizados para um processo de difusão de boro em silício a 1000ºC. A simulação acima representa um processo de difusão bastante longo, durando quase 9 horas. A simulação em si foi bastante rápida, levando cerca de 100 ms.

Tanto a duração do passo de tempo quanto o número de partículas por célula dependem do tamanho da célula. Em ambos os exemplos acima, cada célula tem dimensões de 0,1µm x 0,1µm x 0,1µm. Embora essas células sejam pequenas, em muitos processos, principalmente os voltados para microeletrônica, as espessuras de filmes são muito menores. Portanto as células podem facilmente chegar a dimensões na ordem de 1nm. Nesse caso, pela relação desenvolvida para o cálculo do coeficiente de difusão, o passo de tempo teria que ser 10000 vezes menor para manter o coeficiente de difusão constante. Seria então necessário simular 1000000 de passos para simular um processo real de mesma duração (aproximadamente 9 horas) em uma rede de células 10000 vezes maior. Embora isso ainda seja factível, a simulação deixa de ser quase instantânea, levando cerca de 20 horas para terminar.

4.5 Visualização dos Resultados

Os resultados das simulações podem ser visualizados de duas maneiras. A mais simples é pelo material. Quando o usuário realiza uma simulação de dopagem, deve definir o tipo de material usado para dopagem. O usuário também deve definir a concentração de dopantes na superfície e uma porcentagem limite. Durante a simulação, células com uma concentração de dopantes superior ao limite definido pelo usuário tem seu estado modificado. Essa mudança de estado pode ser visualizada pela cor da célula. Dessa forma, é possível visualizar de forma simplificada as regiões dopadas, que são representadas de maneira semelhante à usada em livros de circuitos microeletrônicos.

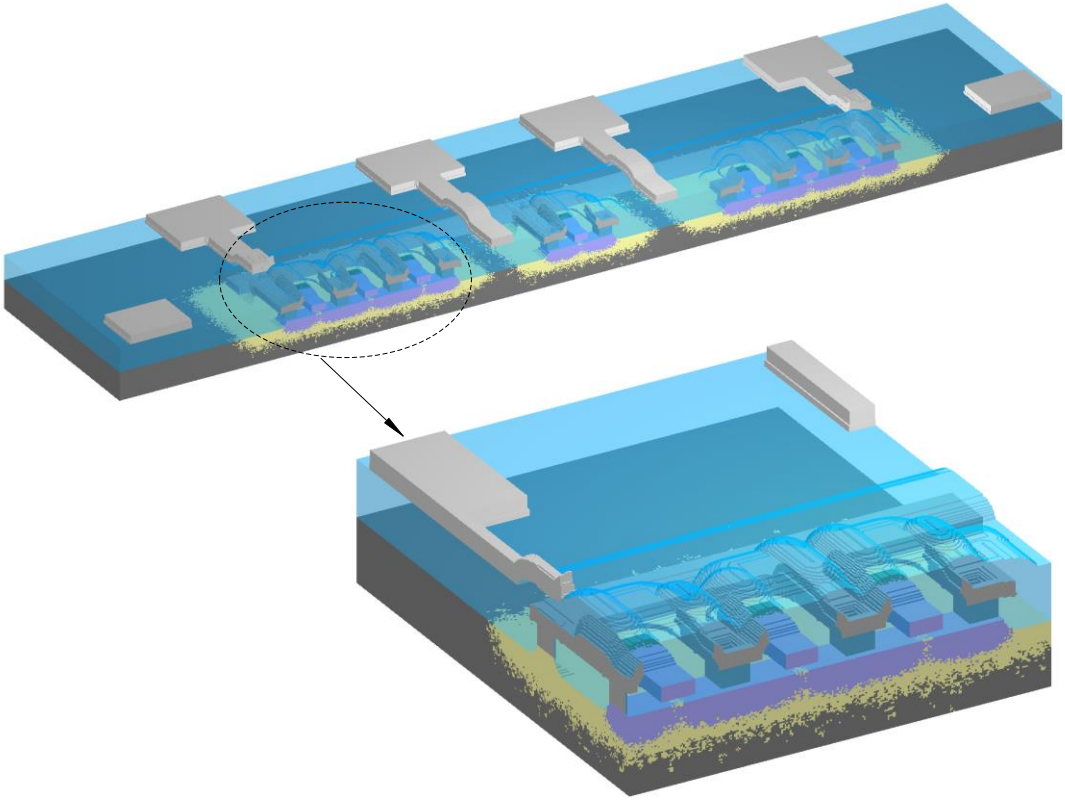


Figura 68 – Resultado da simulação de dopagem usando o autômato celular multipartículas.

A Figura 68 mostra o resultado da fabricação de transistores onde o processo de dopagem para formar o poço N (em amarelo) e as regiões P+ (em azul) foram feitas usando o autômato celular multipartículas. As camadas de óxido de silício foram deixadas semitransparentes para facilitar a visualização. O material em verde é silício policristalino e também é dopado durante o processo, por isso sua ponta fica azul. As demais conexões em cinza claro são de alumínio.

Essa visualização por estados busca permitir ao usuário rapidamente verificar quais regiões estão dopadas e como estão dopadas. No entanto, o resultado de um processo de dopagem é na verdade bem mais complexo que um resultado binário. Para visualizar as concentrações, um novo método de gerar superfícies foi implementado. Baseado no método de faces visíveis, as cores dos materiais foram modificadas de forma a permitir renderizar um espectro de cores para representar as diferentes concentrações.

O usuário pode selecionar qual distribuição quer visualizar. É possível visualizar apenas os dopantes tipo N, apenas os dopantes tipo P, ou a soma dos dopantes.

Algumas configurações de cores também foram disponibilizadas, variando de preto a branco, de vermelho a azul ou de azul a amarelo.

4.6 Aplicação do Autômato Celular para Oxidação

Como descrito no início dessa seção, o autômato celular multipartículas pode ser utilizado para simular sistemas como difusão e reação. Como mencionado inicialmente, duas ou mais espécies se difundem pelo meio. Essas espécies reagem entre si, formando novos compostos.

Para simular a oxidação do Si, uma pequena modificação foi realizada. Apenas a difusão do oxigênio (O_2) no óxido de Si e no Si é simulada. Dessa forma, simula-se a oxidação a seco. O oxigênio, ao se difundir para uma célula preenchida por Si pode então reagir com os átomos de Si. Cada vez que uma reação ocorre, um átomo de Si e uma molécula de O_2 são consumidos. Quando todos os átomos de Si em uma célula reagem, o estado da célula passa a indicar que a célula é ocupada por óxido de Si. Essa primeira abordagem foi testada para determinar a viabilidade da simulação de oxidação usando o autômato celular baseado no autômato celular multipartículas.

O modelo clássico utilizado para descrever a oxidação térmica do Si é o modelo Deal-Grove (DEAL; GROVE, 1965). Nesse modelo, a espessura da camada de óxido X está relacionada com o tempo de oxidação t pela equação quadrática abaixo:

$$t = \frac{X^2}{B} + \frac{X}{B/A}. \tag{37}$$

Os coeficientes A e B são determinados a partir do tipo de oxidação (a seco ou úmida) e da temperatura de oxidação T através das equações abaixo (DEAL; GROVE, 1965) :

$$B = B_0 e^{-E_{A_B}/kT}; \tag{38}$$

$$A = A_0 e^{-E_{A_A}/kT}. \tag{39}$$

O valor dos coeficientes B_0, A_0, E_{A_B}, e E_{A_A} definem o tipo de oxidação (a seco ou úmida) e também variam com a pressão ambiente durante a oxidação. Támbém é comum definir uma constante τ_{ox} equivalente ao tempo de oxidação necessário para

se obter a camada de óxido presente antes da oxidação. Isso se faz necessário visto que em geral há a presença de um óxido nativo sobre a lâmina. A relação entre τ_{ox} e X_0, a espessura do óxido inicialmente sobre a lâmina cujo valor pode ser calculado pela equação:

$$\tau_{ox} = \frac{X_0^2}{B} + \frac{X_0}{B/A}. \tag{40}$$

Definido τ_{ox} é possível obter as soluções para a equação parabólica acima e determinar o valor da espessura final em função do tempo de oxidação t, expressa na equação 41:

$$X(t) = \frac{-A + \sqrt{A^2 + 4B(t + \tau_{ox})}}{2}. \tag{41}$$

Dado o modelo acima, foi realizado um primeiro teste, que buscou verificar se o autômato modificado segue o comportamento parabólico descrito acima. Para tanto, novamente optou-se por uma simulação mais simples, de uma simples superfície plana com condições de contorno cíclicas. Dessa maneira, novamente foi simulada a oxidação de um plano infinito, com uma concentração constante de oxigênio na superfície do substrato. Por simplicidade, o substrato não possuía uma camada de óxido nativo. Os resultados de uma primeira simulação são apresentados na Figura 69.

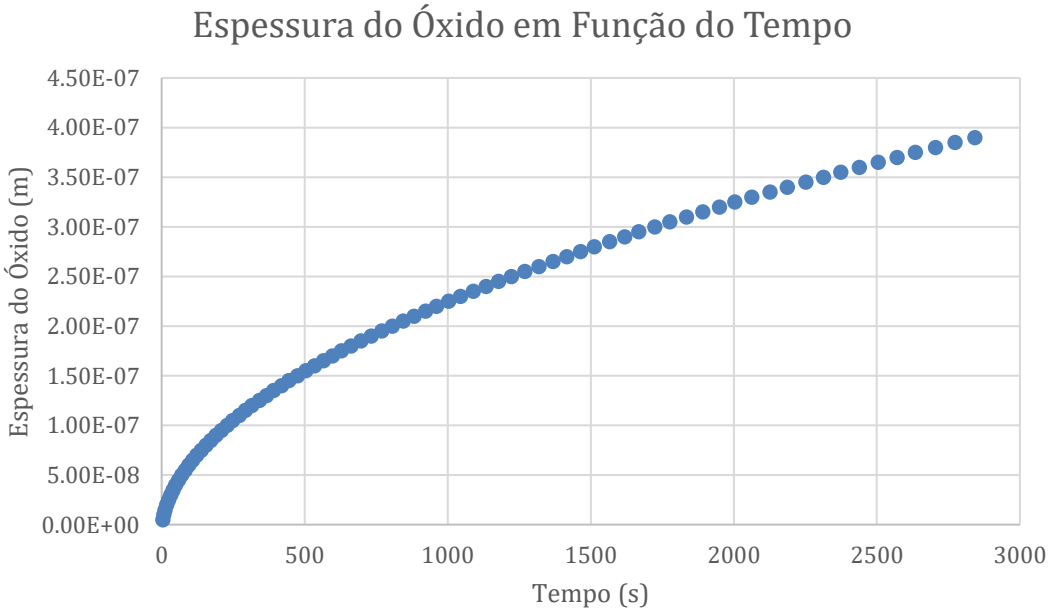


Figura 69 – Espessura do óxido de Si em função do tempo de oxidação simulado.

Ao fazer o ajuste dos dados a uma parábola, encontra-se que os dados podem ser descritos pela equação 42 com $R^2 = 0{,}999996$:

$$t = 1{,}7132\times10^{16}\, X^2 + 6{,}1682\times10^{8} X - 2{,}6004. \qquad (42)$$

Na equação 42, t é o tempo de oxidação e X é a espessura do óxido. Conclui-se que o comportamento do autômato celular segue de forma bastante próxima o comportamento quadrático descrito no modelo Deal-Grove, apresentado na equação 37.

Como passo seguinte, é interessante determinar a relação entre os parâmetros do autômato celular e os parâmetros reais de uma oxidação. O tamanho da célula depende do coeficiente de difusão e da espessura do óxido a ser crescido. É interessante ter mais de uma célula para representar o óxido, mas é preciso levar em conta também o fato de que o passo de tempo diminui com o quadrado do tamanho de célula, como visto anteriormente na equação 28.

É preciso também definir o coeficiente de difusão D da molécula de oxigênio no óxido de Si. Sabendo este valor e o tamanho da célula, é trivial definir a duração de um passo de tempo. O coeficiente de difusão é mais comumente descrito através dos valores de D_0e E_a para uma faixa de temperaturas. A equação 43 é então utilizada para encontrar o coeficiente de difusão (LAMKIN; RILEY; FORDHAM, 1992):

$$D(T) = D_0 e^{-E_a/RT}. \qquad (43)$$

Os valores de D_0e E_a para uma oxidação a seco reportados por Norton são de D_0 = 2,8 x 10^{-8} m^2s^{-1} e E_a=-113 $kJmol^{-1}$ na faixa de 950°C a 1078 °C (NORTON, 1961). O parâmetro final que entra na conta do coeficiente de difusão é a probabilidade de uma partícula ficar na célula. Embora qualquer valor entre 0 e 1 seja válido, é preciso manter em mente alguns limites. As probabilidades não podem ser tão próximas de 0 ou de 1 que seja impossível para o gerador de números aleatórios efetivamente representar essa probabilidade.

A probabilidade de ocorrer uma reação entre o oxigênio e o Si depende de diversos fatores. No entanto, foi assumida como sendo 1. Ou seja, sempre ocorre uma reação entre o Si e uma molécula de O_2 que chega à interface Si/SiO_2. Essa decisão foi tomada visto que o modelo Deal-Grove assume que não há O_2 se difundindo no Si cristalino e que essa reação ocorre sempre na interface entre os materiais.

O método desenvolvido tem algumas desvantagens. Tomando como exemplo a simulação de um processo de oxidação a 1000 °C, o coeficiente de difusão a ser utilizado é de $6{,}47 \times 10^{-9}$ cm^2/s, de acordo com os dados apresentados previamente. Nesse caso, para uma célula de 100 nm, o passo de tempo tem uma duração de 1,28 ms. A 1000 °C são precisas cerca de 120 horas de oxidação para se obter um filme de 1 μm (ou dez células). Para simular essas 120 horas, por sua vez, são precisos cerca de 337 milhões de passos.

A atual implementação do autômato celular descrito leva cerca de 0,79 ms para simular um passo. Esse valor é o tempo mínimo para rodar a lógica do autômato celular. Não pode ser diminuído aumentando o número de processadores, embora seja possível simular uma rede maior no mesmo tempo com mais processadores.

Conclui-se então que a simulação de 120 horas de oxidação a 1000 °C também demora cerca de 74 horas para ser realizada. Este problema se agrava para células menores pois, como foi apresentado antes, ao diminuir o tamanho da célula para 10nm, por exemplo, o tempo simulado fica 100 vezes menor. Nesse caso, simular 120 horas de oxidação demora muito mais que 120 horas; leva cerca de 7400 horas ou 308 dias.

Dessa forma, embora o autômato celular tenha apresentado um comportamento bastante próximo do modelo Deal-Grove, o mesmo não se mostrou muito eficiente para simulação deste processo nas escalas usuais da microeletrônica. No entanto, o modelo ainda fornece uma ferramenta interessante para o estudo do processo de oxidação em si, o que será o foco do estudo do mesmo daqui em diante.

5 CONCLUSÕES

Neste trabalho foram apresentados diversos aspectos relacionados à utilização de autômatos celulares na simulação de processos de microfabricação para a área de MEMS, MOEMS e microeletrônica. Isto foi feito através do desenvolvimento de um software de simulação (o simMEMS), que integra em um único ambiente de simulação, diferentes algoritmos e implementações de autômatos celulares para simular as diversas técnicas de fabricação abordadas no trabalho, tais como corrosão anisotrópica do Si, deposição e corrosão de filmes, fotolitografia, processos de difusão, entre outras. Assim sendo, o trabalho envolveu não apenas o desenvolvimento do software em si, mas também o estudo dos autômatos celulares como metodologia de simulação de processos físicos.

Essa escolha por duas frentes de trabalho desenvolvidas paralelamente teve consequências importantes no trabalho realizado. Por exemplo, foi possível desenvolver um ambiente único de trabalho e simulação (o simMEMS), que permite trabalhar não apenas com uma ou duas técnicas específicas, mas com uma diversidade de métodos de microfabricação e que, conforme mostrado na seção 3, podem ser utilizadas em conjunto para simular a fabricação de sistemas complexos, como circuitos integrados e Dispositivos de Microespelhos Digitais (DMS's). Porém, apesar dos grandes avanços já conseguidos até aqui, o presente trabalho é apenas o início de um longo estudo. Ainda há muito a ser feito em termos de usabilidade do software e de recursos de visualização e análise dos resultados. Além disso, a simulação dos processos é feita de forma idealizada e semi quantitativa, sem relação direta com os parâmetros físicos que determinam esses processos (como temperatura, concentrações e tempo) e na maioria dos casos, em unidades de células do autômato e não em micrômetros.

Por outro lado, e conforme foi mostrado na seção 4, a relação entre os processos de fabricação e os parâmetros físicos (temperatura, concentrações e tempo) e a forma como essa questão pode ser tratada no âmbito da simulação computacional utilizando autômatos celulares foram estudadas. Para isto, o foco foi no processo de difusão de partículas, pois esse é o fenômeno por trás de processos de fabricação essenciais, como a oxidação térmica do silício e a difusão térmica de dopantes em silício. Nesse

estudo foi estabelecido, por exemplo, que a difusão de dopantes pode ser simulada de forma rápida e eficiente, o que viabiliza a inclusão desse processo na lista dos disponíveis no simMEMS.Também foi visto que, devido aos valores do coeficiente de difusão do oxigênio em SiO_2, a simulação da oxidação térmica do Si é extremamente lenta e que, para viabilizar sua inclusão no simMEMS, será necessário continuar as pesquisas e desenvolver novos modelos de autómatos celulares.

Nesse contexto, o que se tem é uma plataforma de simulação que permite trabalhar com um grande número de processos de fabricação, embora faça essas simulações de forma idealizada e semi quantitativa. De fato, por enquanto as simulações baseadas em parâmetros físicos envolvem apenas a difusão de impurezas dopantes no Si. Como alternativa, o foco do trabalho poderia ter sido nos aspectos relativos ao software, produzindo uma plataforma mais amigável e com mais recursos. Alternativamente, o foco poderia ter sido na simulação de um processo específico (a corrosão anisotrópica do silício, por exemplo) gerando assim um simulador que leva em conta o tipo de solução corrosiva, a temperatura, a concentração dessas soluções e a energia de ligação entre os átomos de Si. Porém, essas seriam abordagens limitadas em si e de menor alcance. A opção aqui tomada foi abordar o problema de forma mais geral e, além de estudar a utilização de autômatos celulares na simulação de processos de microfabricação, desenvolver uma solução que fosse de interesse para um universo mais amplo de usuários. Uma solução que, por um lado, pudesse ser usada em âmbito acadêmico, para auxiliar o ensino e aprendizado em cursos de pós-graduação, por exemplo, mas por outro lado, que também permitisse estudos científicos mais rigorosos, quantitativos e confiáveis. Em função do exposto, acredita-se que o trabalho aqui descrito é o resultado de uma proposta inovadora, sem paralelo em âmbito nacional, e com apenas alguns projetos semelhantes em âmbito internacional, como o próprio projeto do software Microwind, utilizado na seção 3.

Feitas essas considerações, é importante comentar em maiores detalhes alguns aspectos que merecem destaque neste trabalho. Como mencionado, um dos objetivos do desenvolvimento do simulador simMEMS foi ampliar o leque de processos de fabricação disponíveis para o usuário. O pré-requisito aqui foi garantir que os novos processos funcionem em paralelo com os demais processos já implementados. Além disso, muito trabalho foi realizado para garantir uma boa experiência para o usuário e diversas mudanças foram implementadas de forma a melhorar o desempenho da

simulação, diminuindo os tempos de processamento e a quantidade de memória necessária para realizar as simulações.

De um modo geral, os resultados aqui apresentados mostram que o processo de paralelização da simulação e de otimização dos autômatos celulares implementados foi bem-sucedido. Por exemplo, em relação à versão 3 do software, foi possível diminuir pela metade a memória necessária para realizar a mesma simulação. Os tempos de simulação também foram diminuídos de forma significativa, o que é fundamental numa ferramenta de auxílio ao ensino. Em paralelo novos métodos para gerar a superfície dos resultados foram adicionados e melhorias na estabilidade do software como um todo foram implementadas. Nesse sentido também foi iniciado o trabalho de desenvolvimento de material de apoio para uso do software.

A seção 2 apresentou uma descrição da implementação do software. A descrição foi focada nos aspectos que mais causaram problemas e nas soluções escolhidas. Embora adequadas, nem todas as soluções escolhidas são ótimas. A constante adição de novos autômatos celulares implica na constante criação de novos conflitos e problemas. Contudo, algumas observações podem ser feitas sobre as implementações realizadas, principalmente no que tange à implementação dos autômatos celulares e à visualização dos resultados.

Uma primeira conclusão pode ser tirada sobre a maneira como os dados são armazenados. Existem dados que necessariamente devem ser compartilhados entre os diversos autômatos celulares, como o tipo de material que ocupa cada célula. Por outro lado, existem dados que são usados por apenas alguns tipos de autômatos celulares. Podem também existir dados que precisam ser armazenados para algumas células, mas não para outras. Além disso, a modificação dos dados se dá de maneira diferente para diferentes processos. Por outro lado, é preciso manter em mente que em geral, o número de células será sempre grande e que, a cada passo de simulação, as operações sobre as células serão realizadas sobre todas elas ou sobre um subconjunto delas. Portanto, armazenar os dados diretamente em um vetor melhora a localidade dos dados e o aproveitamento da memória cache dos processadores atuais, diminuindo, e bastante, o tempo de processamento.

Outra melhora implementada foi alocar e armazenar dados de forma mais dinâmica. Assim, um eventual novo autômato celular poderá alocar dados para simulação da

forma que fizer mais sentido para sua implementação. No caso da simulação de corrosão anisotrópica, por exemplo, é conhecido que uma estrutura do tipo octree pode ser utilizada para diminuir significativamente o consumo de memória para armazenar a estrutura cristalina do Si. Essa mesma estrutura, no entanto, seria prejudicial para armazenar os dados necessários para simulação do processo de dopagem usando o autômato celular multipartículas, ou para armazenar as distâncias usadas para deposição ou corrosão de um filme. Como cada autômato celular pode armazenar dados numa estrutura otimizada para a sua implementação, esta abordagem facilita agregar novos autômatos celulares ao software. Dessa forma o desenvolvimento do simMEMS está agora direcionado para este tipo de armazenamento.

Finalmente, as mudanças realizadas na implementação dos autômatos celulares usados no simMEMS permite que todos os autômatos celulares agora em uso operem de um modo semelhante ao esquema de memória A-A para o método Lattice Boltzmann (ver item 4.3.1). Neste esquema, apenas uma cópia de cada estado é mantida em memória para a simulação, reduzindo quase pela metade a memória usada para a simulação de uma mesma rede. Para os tamanhos de rede com que se pretende trabalhar, esse ganho é bastante significativo. Vale destacar que a implementação do autômato celular multipartículas neste trabalho utiliza este esquema, que foi adotado para evitar a necessidade de armazenar uma segunda cópia do estado das células. É importante mencionar também que esta abordagem é uma contribuição do presente trabalho, pois a implementação original não utiliza este esquema.

Quanto à lógica de paralelização dos autômatos celulares aqui apresentado, é possível afirmar que a mesma se mostrou bastante eficaz. Embora seja bastante simples, não sendo feito, por exemplo, nenhum tipo de balanceamento de carga entre os threads, o resultado obtido é muito bom. Em geral, se o usuário se limitar a simular processos em redes cujo tamanho não sejam significativamente maiores do que o necessário, é bastante incomum que a evolução da superfície ocorra em uma área muito limitada. Dessa forma, o trabalho entre os threads tende a se distribuir de forma mais ou menos uniforme se o usuário realizar simulações de forma racional. De fato, verifica-se que a forma como o usuário utiliza o software afeta o desempenho do simulador. Por exemplo, a possibilidade de gerar superfícies usando octrees é

altamente conveniente em processos que envolvem substratos ou camadas espessas. Já processos planares, como a fabricação de circuitos CMOS, só serão beneficiados se o usuário tomar o cuidado de selecionar espessuras que permitam uma efetiva compactação dos dados.

A implementação da paralelização através da replicação dos recursos necessários para simulação, como listas ou geradores de números aleatórios, para cada thread, se mostrou como uma forma simples e confiável de implementar os diversos autômatos celulares aqui apresentados. De fato, neste trabalho foi possível implementar todos os processos utilizando este método, o que também facilita a leitura e compreensão do código.

Dessa forma, o desenvolvimento do simMEMS deve se basear em migrar o armazenamento dos diversos dados que formam o estado de um autômato celular para estruturas específicas para cada autômato celular. A escolha da representação deve manter o atual critério de minimizar o tempo de processamento sobre o tamanho da estrutura armazenada. No futuro, também se cogita a possibilidade de separar a simulação entre diversos nós, visto que muitas vezes essa é uma maneira mais fácil de aumentar a quantidade de memória e processamento disponível para simulação de estruturas maiores.

Na seção 3 foi apresentado um processo de fabricação baseado em máscaras geradas pelo software Microwind. O Microwind também é um software voltado para uso educacional, o que permite visualizar onde o simMEMS se insere do ponto de vista de um software para projeto, fabricação e teste de um MEMS, dispositivo ou circuito microeletrônico. Enquanto o Microwind permite criar um layout do circuito e simular seu comportamento, o simMEMS permite simular e visualizar a fabricação do dispositivo em si. Um usuário do simMEMS pode, dessa forma, obter um entendimento muito mais profundo da complexidade do processo de fabricação e do número de passos e processos envolvidos na fabricação de um dispositivo que pode ser, do ponto de vista funcional, simples ou complexo (como as portas lógicas NOR mostradas na seção 3).

A seção 3 também apresentou os processos idealizados de simulação disponíveis, bem como diversos exemplos de processos de fabricação que podem ser simulados no simMEMS. Os resultados mostram que a gama de dispositivos cujo processo de

fabricação pode ser executado pelo software é bastante ampla. O simMEMS tenta não limitar o que o usuário pode fazer. Por um lado, isso aumenta a chance de que o usuário utilize um processo de uma forma impossível na vida real, o que pode ser negativo para um software voltado para aprendizado. Por outro lado, visto que o simMEMS pode ser usado como uma ferramenta para apenas visualização de processos e dispositivos genéricos, optou-se por permitir essa liberdade.

Quanto ao exemplo da porta NOR apresentada, trata-se de um resultado bastante interessante. O tamanho da estrutura, em células, é cerca de quatro vezes maior que a maior estrutura simulada em versões anteriores do software. Além disso, a simulação só foi possível graças a implementação de novos processos como dopagem do silício, oxidação térmica e planarização. Esse exemplo também é interessante porque mostra como o simMEMS pode ser utilizado em conjunto com outro software também voltado para fins educacionais (no caso, o Microwind).

A maioria dos processos apresentados na seção 3 é baseada no mesmo autômato celular (que conta distâncias). Este autômato utiliza uma vizinhança muito pequena, de apenas 6 vizinhos, e permite obter resultados muito bons, em termos do formato final das superfícies. Isso resulta em um autômato celular que é rápido e permite gerar bons resultados independentemente da resolução da rede; não é preciso ajustar parâmetros para que se obtenha, por exemplo, uma deposição isotrópica em uma rede de 30x30x30 células ou em uma rede de 300x300x300 células. Além disso, o autômato celular contador de distância pode simular de forma idealizada diversos processos através de variações nas condições iniciais. Também é importante destacar que este autômato celular também é uma contribuição original do presente trabalho.

Finalmente, na seção 4 apresentamos a simulação baseada em parâmetros físicos. De fato, enquanto os autômatos celulares apresentados previamente (para simulação idealizada dos processos) se restringem a resolver o problema da evolução de uma superfície, o autômato celular multipartículas usa um processo simples para simular a difusão de partículas em um dado meio. No caso do autômato celular multipartículas, o modelo descreve o funcionamento do autômato celular, e não o oposto. Isto é, o autômato celular não busca resolver a equação diferencial que descreve o processo de difusão. Nesse caso, a equação diferencial que descreve o comportamento

macroscópico do autômato celular é a mesma que descreve macroscopicamente o processo de difusão.

Já para o processo de oxidação térmica do Si, o autômato celular se mostrou menos eficiente. Embora os resultados obtidos mostrem que a evolução da espessura do óxido segue o formato esperado (modelo de Deal-Grove), a relação entre o tempo simulado de oxidação e o tempo que demora a simulação é muito baixa. Ou seja, são necessárias muitas horas de simulação para obter espessuras de apenas alguns nanômetros de óxido. Isto acontece porque o coeficiente de difusão do oxigênio no óxido de Si é muito baixo. Por exemplo, é cerca de três ordens de grandeza menor que o do boro no Si (usado no processo de difusão de impurezas dopantes). O resultado disso é que cada passo de simulação corresponde a tempos de oxidação extremamente curtos e de fato, quando se usa células muito pequenas (na ordem de dezenas de nanômetros), o tempo de simulação é muito superior ao tempo real da difusão em si num forno de laboratório. Como exemplo, nas simulações realizadas aqui, para obter filmes de 400 nm foi realizada uma simulação que levou cerca de 90 dias.

Por outro lado, verificamos que o autômato celular multipartículas se encaixa muito bem na simulação da difusão de dopantes em Si. Para os tamanhos de célula de interesse e as concentrações típicas de dopantes, trata-se de uma alternativa válida. É preciso, de qualquer forma, manter em mente a relação entre tamanho de células e a duração do passo de tempo, bem como o fator atenuante de que a necessidade de uma célula menor estará associada, em geral, a dopagens com menor profundidade (menos células) e menor tempo (menos passos).

Para finalizar, os resultados apresentados mostram que o uso de autômatos celulares para simulação de processos de microfabricação não só é factível, como apresenta diversas vantagens. O autômato celular contador de distância se mostra como uma ótima alternativa para uso em simulações baseadas no problema de evolução de superfícies. Autômatos celulares que simulam o processo em si, levando em conta parâmetros físicos e não somente a evolução da superfície em função de outro modelo, também são de grande interesse e podem facilmente ser utilizados em paralelo com os anteriores. Pode-se incluir nesse grupo tanto o autômato celular multipartículas como autômatos celulares atomísticos para simulação da corrosão

anisotrópica do Si. Durante a seleção de autômatos celulares para uso em simulação de processos de microfabricação é importante manter em mente as limitações impostas pela escala em que os processos de simulação se dão. Como mencionado acima, dependendo do tamanho escolhido das células algumas das suposições feitas podem não ser válidas.

6 BIBLIOGRAFIA

AIDUN, C. K.; CLAUSEN, J. R. **Lattice-Boltzmann Method for Complex Flows**. Annual Review of Fluid Mechanics, v. 42, n. 1, p. 439–472, 2010.

ASHCROFT, N. W.; MERMIN, N. D. **Solid State Physics**. 1. ed. Philadelphia, PA, USA: Brooks Cole, 1976.

AUTODESK. **AutoCAD for Mac & Windows |CAD Software|Autodesk**. Disponível em: <http://www.autodesk.com/products/autocad/overview>. Acesso em: 25 ago. 2016.

AZIZ, S. M.; SICARD, E. **Industry relevant learning in VLSI Design : the 65 nanometre CMOS technology in Microwind3**. Anais International Conference on Engineering Education & Research. Melbourne, Australia. 2007.

AZIZ, S. M.; SICARD, E.; DHIA, S. BEN. **Effective Teaching of the Physical Design of Integrated Circuits Using Educational Tools**. IEEE Transactions on Education, v. 53, n. 4, p. 517–531, 2010.

BAILEY, P. et al. **Accelerating Lattice Boltzmann Fluid Flow Simulations Using Graphics Processors**. 2009 International Conference on Parallel Processing. Anais...Vienna, Austria: IEEE Computer Society, 2009

BAKER, R. J. **CMOS : circuit design, layout, and simulation**. 3. ed. Piscataway, NJ, USA: IEEE Press, Hoboken, NJ, USA: Wiley, 2010.

BELORKAR, U. A.; BHANDE, A. E. VLSI **Implementation of Back Propagated Neural Network for Signal Processing**. International Journal of Electronics, Communication and Soft Computing Science & Engineering, v. 4, p. 111–115, 2015.

BENZI, R.; SUCCI, S.; VERGASSOLA, M. **The lattice Boltzmann equation: theory and applications**. Physics Reports, v. 222, n. 3, p. 145–197, 1992.

Boost C++ Libraries. Disponível em: <http://www.boost.org/>. Acesso em: 23 ago. 2016.

BUIE, J. L. Coupling transistor logic and other circuits – US. Pat. US3283170, 1 nov. 1966. 4p.

BURKS, A. W. **Essays on Cellular Automata**. 1. ed. Champaign, IL, USA: University of Illinois Press, 1970.

BYL, J. **Self-reproduction in small cellular automata**. Physica D: Nonlinear Phenomena, v. 34, n. 1–2, p. 295–299, 1989.

CAMPBELL, P. K. et al. **A silicon-based, three-dimensional neural interface: manufacturing processes for an intracortical electrode array**. IEEE transactions on bio-medical engineering, v. 38, n. 8, p. 758–68, 1991.

CARVALHO, D. O. **Desenvolvimento de processos de fabricação de dispositivos óptico integrados em tecnologia de silício para aplicação em sensoriamento**. São Paulo, 2012. 128p. Tese de Doutorado – Escola Politécnica - Universidade de São Paulo.

CHEN, K. J. et al. **Fabrication and Characteristics of Silicon Micro-Tip Arrays**. International Journal of Modern Physics B, v. 24, n. 28, p. 5601–5611, 2010.

CHEN, S.; DOOLEN, G. D. **Lattice Boltzmann method for fluid flows**. Annual Review of Fluid Mechanics, v. 30, n. 1, p. 329–364, 28 jan. 1998.

CHOPARD, B.; DROZ, M. **Cellular Automata Modeling of Physical Systems**. 1. ed. Cambridge: Cambridge University Press, 2005.

COLOMBO, F. B. **Desenvolvimento de um software para simulação atomística de processos de microfabricação baseado em autômatos celulares**. São Paulo, 2011. 133p. Dissertação de Mestrado - Escola Politécnica - Universidade de São Paulo.

COLOMBO, F. B.; CARREÑO, M. N. P. **A cellular automata based multi-process microfabrication simulator**. Journal of Integrated Circuits and Systems, v. 6, n. 2, 2011.

COLOMBO, F. B.; CARREÑO, M. N. P. **Simulation of PECVD SiO2 deposition using a cellular automata approach**. ECS Transactions, v. 49, n. 1, p. 297–304, 2012.

COLOMBO, F. B.; CARREÑO, M. N. P. **A Novel Cellular Automaton for Simulation of Film Deposition and Etching Processes**. Microelectronics Technology and Devices (SBMicro), 2013 Symposium on. Curitiba, 2013. **Anais**. Curitiba, Brasil: IEEE.

COMPUTER HISTORY MUSEUM. **1963: Standard Logic IC Families Introduced**. Disponível em: <http://www.computerhistory.org/siliconengine/standard-logic-ic-families-introduced/>. Acesso em: 25 ago. 2016.

DEAL, B. E.; GROVE, A. S. **General Relationship for the Thermal Oxidation of Silicon**. Journal of Applied Physics, v. 36, n. 12, p. 3770, 1965.

EINSTEIN, A. **Investigations on the Theory of the Brownian Movement**. 1. ed. Mineola, NY, USA:Dover Books on Physics, 1956.

FERRANDO, N.; GOSÁLVEZ, M. A.; COLÓM, R. J. **Evolutionary continuous cellular automaton for the simulation of wet etching of quartz**. Journal of Micromechanics and Microengineering, v. 22, n. 2, p. 25021, 2012.

FOSTER, A. S.; NIEMINEN, R. M.; GOSÁLVEZ, M. A. **Atomistic simulations of surface coverage effects in anisotropic wet chemical etching of crystalline silicon**. v. 202, p. 160–182, 2002.

GAMMA, E. et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. 1. ed. Boston: Addison-Wesley Professional, 1994.

GOSÁLVEZ, M. A et al. **Simulating anisotropic etching of silicon in any etchant: evolutionary algorithm for the calibration of the continuous cellular automaton**. Journal of Micromechanics and Microengineering, v. 21, n. 6, p. 65017, 2011.

GOSÁLVEZ, M. A. et al. **Atomistic methods for the simulation of evolving surfaces**. Journal of Micromechanics and Microengineering, v. 18, n. 5, p. 55029, 2008.

GOSÁLVEZ, M. A. et al. **Discrete and continuous cellular automata for the simulation of propagating surfaces**. Sensors and Actuators A: Physical, v. 155, n. 1, p. 98–112, 2009.

HANASOGE, S. M.; SUCCI, S.; ORSZAG, S. A. **Lattice Boltzmann method for electromagnetic wave propagation**. EPL (Europhysics Letters), v. 96, n. 1, p. 14002, 1 out. 2011.

HOPCROFT, J. E.; MOTWANI, R.; ULLMAN, J. D. **Introduction to Automata Theory, Languages, and Computation**. 3. ed. Harlow, England: Pearson, 2006.

HORNBECK, L. J. **Spatial light modulator and method.** US. Pat. US5061049, 29 out. 1991.

HORNBECK, L. J. **Method of making a support post for a micromechanical device.** US. Pat. US5784212, 21 jul. 1998.

HUANG, Q. A. et al. **A modified cellular automata algorithm for the simulation of boundary advancement in deposition topography simulation**. Journal of Micromechanics and Microengineering, v. 16, n. 1, p. 1–8, 2006.

HUFFMAN, J.; HORNBECK, L.; KNIPE, R. **Memory architecture for micromirror cell**. US. Pat. US6775174, 10 ago. 2004.

ISO International Standard ISO/IEC 14882:2014(E) – **Programming Language C++**. International Organization for Standardization, 2014.

JOHNSONBAUGH, R. **Discrete Mathematics**. 7. ed. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2009.

KARAFYLLIDIS, I. **A three-dimensional photoresist etching simulator for TCAD A three-dimensional photoresist etching simulator for TCAD**. Modelling and Simulation in Materials Science and Engineering, v. 7, n. 2, p. 157–168, 1999.

KARAFYLLIDIS, I.; THANAILAKIS, A. **A model for predicting forest fire spreading using cellular automata**. Ecological Modelling, v. 99, n. 1, p. 87–97, 1997.

KHRONOS GROUP. **About The Khronos Group**. Disponível em: <https://www.khronos.org/about>. Acesso em: 25 ago. 2016.

KOVACS, G. T. A.; MALUF, N. I.; PETERSEN, K. E. **Bulk micromachining of silicon**. Proceedings of the IEEE, v. 86, n. 8, p. 1536–1551, 1998.

KWON, U.-H.; LEE, W.-J. **Three-dimensional deposition topography simulation based on new combinations of flux distribution and surface representation algorithms**. Thin Solid Films, v. 445, n. 1, p. 80–89, 2003.

LAMKIN, M. A.; RILEY, F. L.; FORDHAM, R. J. **Oxygen mobility in silicon dioxide and silicate glasses: a review**. Journal of the European Ceramic Society, v. 10, n. 5, p. 347–367, 1992.

LEE, B. **DMD 101: Introduction to Digital Micromirror Device (DMD) Technology**. Disponível em: <http://www.ti.com/lit/an/dlpa008a/dlpa008a.pdf>. Acesso em: 11 dec. 2016.

LIVERMORE, C. **Course materials for 6.777J / 2.372J Design and Fabrication of Microelectromechanical Devices**. Disponível em: <http://ocw.mit.edu/>. Acesso em: 25 ago. 2016.

LORENZO, R.; CHAUDHARY, S. **A novel all NMOS leakage feedback with data retention technique**. International Conference on Control, Automation, Robotics and Embedded Systems (CARE). **Anais**. Jabalpur, India: IEEE, 2013.

MATSUMOTO, M.; NISHIMURA, T. **Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator**. ACM Transactions on Modeling and Computer Simulation, v. 8, n. 1, p. 3–30, 1998.

MINK, A. et al. **A 3D Lattice Boltzmann method for light simulation in participating media**. Journal of Computational Science, v. 17, Part 2, p. 431-437, 2016.

MORITZ, H. **Optical Single Layer Lift-Off Process**. IEEE Transactions on Electron Devices, v. 32, n. 3, p. 672–676, 1985.

NEHANIV, C. L. **Self-reproduction in asynchronous cellular automata**. Proceedings 2002 NASA/DoD Conference on Evolvable Hardware, p. 201–209, 2002.

NORTON, F. J. **Permeation of Gaseous Oxygen through Vitreous Silica**. Nature, v. 191, n. 4789, p. 701–701, 12 ago. 1961.

OPENGL.ORG. **Legacy OpenGL --- OpenGL.org**. Disponível em: <http://www.opengl.org/wiki_132/index.php?title=Legacy_OpenGL&oldid=13721>. Acesso em: 25 ago. 2016.

PAL, P. et al. **Study of corner compensating structures and fabrication of various shapes of MEMS structures in pure and surfactant added TMAH**. Sensors and Actuators A: Physical, v. 154, n. 2, p. 192–203, set. 2009.

QT COMPANY. **Qt for developers by developers | Cross-platform development**. Disponível em: <https://www.qt.io/developers/>. Acesso em: 23 ago. 2016.

SATO, K. et al. **Characterization of orientation-dependent etching properties of single-crystal silicon: effects of KOH concentration**. Sensors and Actuators A: Physical, v. 64, n. 1, p. 87–93, 1998.

SCHADER, M.; SCHMID, F. **Two Rules of Thumb for the Approximation of the Binomial Distribution by the Normal Distribution**. The American Statistician, v. 43, n. 1, p. 23–24, 1989.

SELLERS, G.; WRIGHT JR., R. S.; HAEMEL, N. **OpenGL SuperBible: Comprehensive Tutorial and Reference**. 6. ed. Boston: Addison-Wesley Professional, 2013.

SHIKIDA, M. et al. **Non-photolithographic pattern transfer for fabricating pen-shaped microneedle structures**. Journal of Micromechanics and Microengineering, v. 14, n. 11, p. 1462–1467, 2004.

SICARD, E. **Microwind**. Disponível em: <http://microwind.net/index.php>. Acesso em: 25 ago. 2016.

SIRAKOULIS, G. C.; KARAFYLLIDIS, I.; THANAILAKIS, A. **A TCAD tool for the simulation of the CVD process based on cellular automata**. Le Journal de Physique IV, v. 11, n. Pr3, p. 205-212, 2001.

SIRAKOULIS, G. C.; KARAFYLLIDIS, I.; THANAILAKIS, A. **A cellular automaton methodology for the simulation of integrated circuit fabrication processes**. Future Generation Computer Systems, v. 18, n. 5, p. 639–657, 2002.

SIRAKOULIS, G. C.; KARAFYLLIDIS, I.; THANAILAKIS, A. **A cellular automaton for the propagation of circular fronts and its applications**. Engineering Applications of Artificial Intelligence, v. 18, n. 6, p. 731–744, 2005.

SPROULL, R.; LYON, R.; TRIMBERGER, S. **The Caltech Intermediate Form for LSI Layout Description**. [s.l.] Addison-Wesley Publishing Company, 1980.

STRASSER, E.; SELBERHERR, S. **Algorithms and models for cellular based topography simulation**. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, v. 14, n. 9, p. 1104–1114, 1995.

TEMPESTI, G. **A new self-reproducing cellular automaton capable of construction and computation**. In: Advances in Artificial Life. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. v. 929, p. 555–563.

TEXAS INSTRUMENTS. **CD4001B | NOR Gate | Gate | Description & parametrics**. Disponível em: <http://www.ti.com/product/CD4001B>. Acesso em: 24 ago. 2016a.

TEXAS INSTRUMENTS. **SN7400|NAND Gate|Gate|Description & parametric**s. Disponível em: <http://www.ti.com/product/SN7400/description>. Acesso em: 25 ago. 2016b.

THAN, O.; BÜTTGENBACH, S. **Simulation of anisotropic chemical etching of crystalline silicon using a cellular automata mod**el. Sensors and Actuators A: Physical, v. 45, n. 1, p. 85–89, out. 1994.

TOCCI, R. J.; WIDMER, N. S. **Sistemas Digitais**. 8. ed. São Paulo: Prentice Hall, 2003.

UMA, R. et al. **Area, Delay and Power Comparison of Adder Topologies**. International Journal of VLSI design & Communication Systems, v. 3, n. 1, p. 153–168, 2012.

URSELL, T. S. **The Diffusion Equation A Multi-dimensional Approach**. Out. 2007. Tutorial. Department of Applied Physics, California Institute of Technology. Pasadena, CA 91125.

UYEMURA, J. P. **Chip Design for Submicron VLSI: CMOS Layout and Simulation**. 1. ed. Independence, KY: Cengage Learning, 2005.

VAN KESSEL, P. F. et al. **A MEMS-based projection display**. Proceedings of the IEEE, v. 86, n. 8, p. 1687–1704, 1998.

WOLFRAM, S. **A New Kind of Science**. 1. ed. Champaign, IL, USA: Wolfram Media, 2002.

XING, Y.; GOSÁLVEZ, M. A.; SATO, K. **Step flow-based cellular automaton for the simulation of anisotropic etching of complex MEMS structures**. New Journal of Physics, v. 9, n. 12, p. 436–436, 12 dez. 2007.