

Simulador de Quantum-Dot Cellular Automata (QCA) Utilizando Redes de Hopfield

Omar Paranaíba Vilela Neto, Marco Aurélio C. Pacheco, Carlos R. Hall Barbosa, Leone P. Masiero
PUC-Rio – Pontifícia Universidade Católica
do Rio de Janeiro
Departamento de Engenharia Elétrica
Rua Marquês de São Vicente, 225, Gávea,
Rio de Janeiro, RJ, Brasil, 22453-900
{omar,marco,hall,masiero}@ele.puc-rio.br

Abstract

Quantum-dot cellular automata (QCA) têm sido propostos como uma alternativa à tecnologia CMOS. A principal diferença do paradigma QCA é a obtenção da informação pelo armazenamento de cargas elétricas em células, ao invés do fluxo de corrente. Células de QCA podem ser obtidas de várias maneiras, sendo que a implementação molecular tem sido investigada e promete ser uma grande promessa de desenvolvimento da nanoeletrônica. Contudo, o uso de simuladores é de fundamental importância no desenvolvimento desse tipo de tecnologia, já que sua implementação física ainda é difícil de ser obtida. Este trabalho descreve o projeto de um simulador de QCA utilizando redes neurais de Hopfield. O desenvolvimento dessa ferramenta mostra que as técnicas de inteligência computacional podem trazer um grande auxílio no desenvolvimento de novos projetos de nanotecnologia.

1. Introdução

Os avanços no tamanho e na velocidade dos dispositivos de computador têm sido de grande importância no desenvolvimento de novas tecnologias. Porém, a miniaturização dos dispositivos baseados em transistores está atingindo seu limite físico. A representação binária da informação através do chaveamento de corrente talvez tenha sido uma das mais brilhantes idéias da história da tecnologia, mas esse paradigma possui sérios inconvenientes quando o tamanho dos dispositivos é reduzido. Quanto menor é um *switch*, menor sua capacidade em chavear a corrente. Outro problema é que a corrente entre os *switches* é reduzida, dificultando a

divisão da mesma entre as linhas de interconexão de um dispositivo. Finalmente, o chaveamento de corrente precisa que elétrons se movam de uma fonte de potência passando por uma resistência, resultando em uma grande dissipação de energia. Para se ter uma idéia, se o transistor tradicional puder ser minimizado até o tamanho de uma molécula, um chip com uma fenomenal densidade poderia ser construído, mas este provavelmente se derreteria assim que fosse ligado.

Assim, fica claro que se os dispositivos eletrônicos forem minimizados até o limite da escala molecular, um novo paradigma deve ser criado. O paradigma do Quantum-dot cellular automata (QCA)[1]-[4] mantém a noção de representação binária da informação, mas essa informação é obtida pela configuração das cargas em uma célula e não pelo chaveamento de corrente.

Importantes estudos de confiabilidade e tolerância a falhas têm sido feitos [5]-[8], porém muito mais necessita ser desenvolvido. Outro desafio é repensar a arquitetura de computadores baseada em células QCA, e não em transistores. Alguns passos importantes já foram dados na evolução de novas estratégias [9]-[12] e aplicações [13]-[15] mas, novamente, muito ainda deve ser feito. Nesse aspecto o uso de um simulador pode acelerar bastante as pesquisas nesta área.

A rede neural de Hopfield [25] foi escolhida para o desenvolvimento do simulador por causa de sua característica de apresentar bons resultados nos problemas de otimização e minimização de energia, o que é de fundamental importância para a simulação de QCA.

A próxima seção descreve em mais detalhes o Quantum-dot cellular automata. A seção 3 descreve a rede de Hopfield e seu funcionamento. As seções 4 e 5 descrevem o simulador e os resultados obtidos, respectivamente. A seção 6 conclui o trabalho.

2. Quantum-dot cellular automata (QCA)

No paradigma de QCA os dispositivos são compostos por células, cada uma contendo um número de *quantum dots* (pontos quânticos). Um ponto, neste caso, é apenas uma região fixa onde a carga pode estar localizada. A figura 1 ilustra uma célula QCA com quatro pontos, onde existem dois elétrons (ou buracos) livres que podem se mover e que tendem a minimizar sua interação Coulombiana ocupando pontos em diagonais opostas [16]-[18]. Logo, a célula possui dois estados de mínima energia, que são associados aos dígitos binários “0” e “1”.



Figura 1: Célula com quatro dots

A interação Coulombiana entre as células permite a transmissão da informação. Os dispositivos QCA são criados arranjando as células de forma apropriada. Por exemplo, arranjando-se as células em uma linha, como na figura 2(a), obtém-se um fio, que transmite a informação de uma extremidade à outra. Células posicionadas diagonalmente umas às outras tendem a se alinharem de forma assimétrica. Essa característica é empregada na construção de inversores, como mostrado na figura 2(b). Uma distribuição de um sinal é mostrada na figura 2(c). O dispositivo de lógica natural é uma porta de três entradas conhecida como “majority gate”, mostrada na figura 2(d). Três linhas de entradas convergem em uma saída, cujo estado é determinado pelo estado da maioria nas entradas.

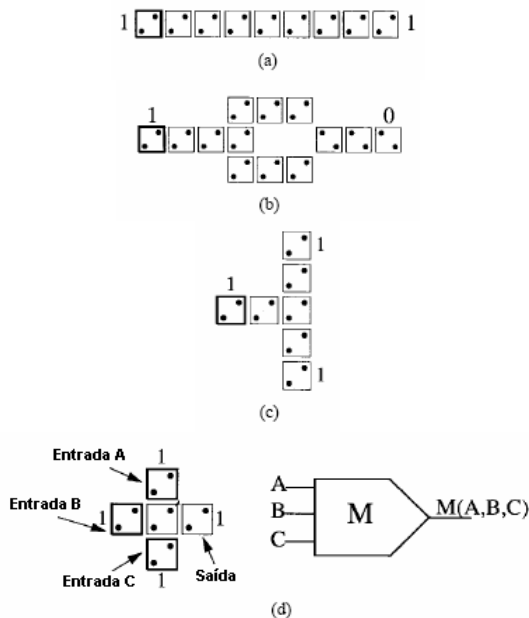


Figura 2: Dispositivos fundamentais de QCA: (a) fio; (b) inversor; (c) fan-out; (d) majority gate.

A partir desses, circuitos mais complexos podem ser desenvolvidos, tais como somadores [3] e até mesmo alguns microprocessadores simples [4].

Dispositivos QCA usando *single-electron switching* foram demonstrados com pequenos pontos metálicos a baixas temperaturas [19]-[22]. Nestes dispositivos, o papel do ponto quântico é feito por uma pequena ilha de metal conectada a outras ilhas por barreiras de junções de tunelamento. Estes experimentos mostraram que o rearranjo das cargas em uma célula pode efetivamente mudar o estado das células vizinhas e possibilitar tarefas computacionais. Dispositivos como *Majority Gates*, fios, memórias e *shift-registers* foram demonstrados.

Para grandes circuitos é necessária a implementação de “zonas de clock”. O *clock* fornece um mecanismo para sincronizar o fluxo da informação. Ele também proporciona controle sobre a direção desse fluxo. Para que esse controle seja feito, existem quatro sinais de *clock*, cada um com um deslocamento de fase de 90 graus, como mostrado na figura 3. Isto permite que a informação seja transmitida através do circuito devido às sucessivas polarizações e despolarizações das células pertencentes a diferentes “zonas de clock”. Além dessas características descritas acima, o *clock* proporciona um ganho de potencial nos dispositivos de QCA, como pode ser visto em [23][24].

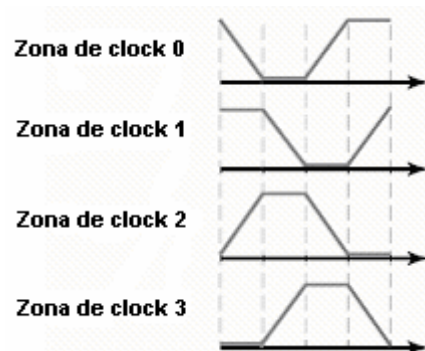


Figura 3 – Os quatro sinais de clock disponíveis. Cada sinal tem um deslocamento de fase de 90 graus.

Outra característica importante de QCA é possibilidade de cruzamento de fios em um plano. Desta forma, circuitos totalmente planos podem ser construídos. Este cruzamento de informações é baseado no fato de que quando colocamos células adjacentes, uma célula normal, com os pontos nas diagonais, e uma célula rotacionada de 45°, estas possuem uma interação mútua muito pequena entre si, como pode ser visto na figura 4.

Esta característica permite que os cruzamentos transmitam as informações independentemente por dois

fos: um fio com células normais e outro com células rotacionadas, como ilustrado na figura 5.



Figura 4: interação entre células normais e rotacionadas.

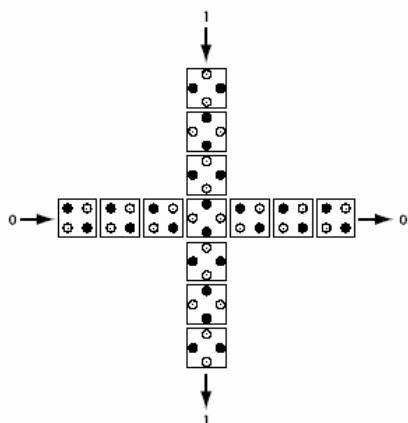


Figura 5: Cruzamento de informações no plano

3. Rede neural do tipo hopfield

Redes neurais artificiais são geralmente baseadas no cérebro humano. Com algumas simplificações o trabalho de um neurônio artificial na rede pode ser comparado ao de um neurônio biológico e os pesos das conexões entre os neurônios da rede às conexões sinápticas dos cérebros humanos.

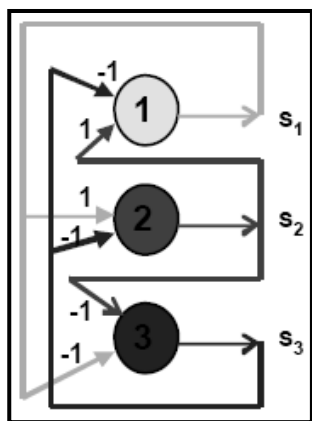


Figura 6 – Modelo de uma rede neural tipo hopfield

Redes neurais do tipo hopfield [25][26] são altamente interconectadas e são usadas com sucesso para resolver

problemas de otimização e minimização de energia. O modelo da rede do tipo hopfield é mostrado na figura 6. A estrutura da rede e os pesos entre os neurônios depende da especificação de cada problema. Um neurônio da rede simplesmente soma os sinais enviados pelos demais neurônios, multiplicados pelos pesos das conexões, e em seguida envia um sinal de resposta para eles. A soma dos sinais recebidos, com seus respectivos pesos, U , e a resposta do neurônio, $V(U)$, são chamados de potenciais de entrada e saída, respectivamente. A função de resposta, $V(U)$, é dada por:

$$V(U) = 1/2[1 + \tanh(\alpha U)] \quad (1)$$

onde α é o ganho da função sigmóide. Para valores suficientemente altos de α , $V(U)$ tem um caracter binário, como neste trabalho, isto é:

$$V(U) = \begin{cases} 0 & \text{if } U < 0 \\ 1 & \text{if } U > 0 \end{cases}$$

Vamos considerar uma rede composta por m neurônios. A função de energia da rede pode ser definida como:

$$E = -1/2 \sum_{i=1}^m \sum_{j=1}^m T_{ij} V_i V_j \quad (2)$$

onde T_{ij} ($i, j = 1, \dots, m$) é o peso da conexão sináptica entre o j -ésimo e o i -ésimo neurônios. Todos os valores de T_{ij} ($=T_{ji}$) formam uma matriz simétrica de pesos de conexões. T_{ij} pode ser positivo, quando há um estímulo que excita, negativo, quando há um estímulo que inibe, ou zero, quando não há conexão entre os neurônios j e i . O potencial de entrada U_i do i -ésimo neurônio é definido pela relação:

$$U_i = -\partial E / \partial V_i \quad (i = 1, \dots, m) \quad (3)$$

e da equação (2) chega-se a:

$$U_i = \sum_{j=1}^m T_{ij} V_j \quad (i = 1, \dots, m) \quad (4)$$

As equações (1) – (4) determinam a evolução da rede para um mínimo local de energia, que é um estado meta-estável da rede.

A rede neural descrita acima foi usada para o desenvolvimento do simulador de QCA. As equações apresentadas, acrescidas de uma lógica de interação da rede, são suficientes para se resolver os problemas do simulador.

4. O simulador

Com o objetivo de desenvolver o simulador utilizando uma rede neural do tipo hopfield deve-se fazer algumas considerações: cada neurônio representa uma célula do circuito QCA e essas células podem assumir os valores 0 e 1 (por esta razão o ganho da função sigmóide é alto). O usuário deve fornecer ao simulador uma matriz que informa a topologia do circuito, ou seja, a matriz deve dizer onde existe e qual o tipo (normal ou rotacionada) das células. Além dessa matriz, o usuário deve informar, através de outra matriz, a qual zona de clock cada célula pertence. O usuário também informa a posição das células de entrada no circuito e a polarização das mesmas.

Nesta versão do simulador foi definido que o raio de vizinhança de uma célula i atinge no máximo a célula após a célula ao lado de i , como mostrado na figura 7. Este raio de vizinhança é o suficiente para o correto funcionamento do simulador, já que a magnitude da interação diminui muito com o aumento da distância entre as células. Com essas informações foi calculado o grau de interação entre todas as possíveis posições vizinhas, levando-se em conta o tipo de células e a polarização. Com esses valores calculados obtém-se a função de energia E , que não é apresentada aqui pois sua expressão é muito grande, possuindo oitenta termos. A derivada dessa função é o potencial de entrada de i , U_i , conforme indicado na seção 3.

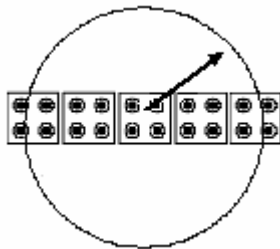


Figura 7 – Raio de vizinhança de uma célula.

O processo de simulação é descrito a seguir:

(i) todos os neurônios são inicializados com o valor igual a “-1” e a quantidade de zonas de clock que se deseja polarizar é definida;

(ii) inicializam-se as células de entrada com os valores fornecidos pelo usuário;

(iii) despolarizam-se os neurônios (células) que estejam em uma zona de clock deslocada de 180 graus da zona de clock atual;

(iv) cada neurônio da zona de clock atual com valor igual a “-1” é escolhido aleatoriamente e, usando a equação 4, seu correspondente valor U_i é calculado. Consequentemente, seu valor V_i é obtido;

(v) ordenam-se todos os neurônios (células), alterados no passo anterior, por ordem crescente do módulo de U_i .

Este passo é importante para evitar que, no passo a seguir, um neurônio com menor energia de interação influencie na polarização de um outro com energia superior; causando erro no funcionamento do simulador;

(vi) cada neurônio da zona de clock atual com valor diferente de “-1” é escolhido, com base na ordenação feita no passo anterior e, usando a equação (4), seu correspondente valor U_i é calculado. Consequentemente, seu valor V_i é obtido;

(vii) cada neurônio da zona de clock atual com valor diferente de “-1” é escolhido aleatoriamente e, usando a equação (4), seu correspondente valor U_i é calculado. Consequentemente, seu valor V_i é obtido;

(viii) caso os valores do neurônios sejam os mesmos que no início do passo (iv), continue. Caso contrário volte ao passo (iv). A alteração dos valores dos neurônios indica que podem existir células de uma zona de clock que ainda não foram polarizadas;

(ix) o simulador fornece o gráfico mostrando a situação das células neste determinado momento. Caso a quantidade de zonas de clock que se deseja polarizar seja atingida o simulador encerra a execução;

(x) muda a zona de clock com um deslocamento de 90 graus, lê os novos valores de entrada e volta ao passo (ii).

A próxima seção mostra os resultados de alguns experimentos, como por exemplo um somador, executados com este simulador.

5. Estudos de casos

Os estudos de casos que são mostrados nesta seção ilustram o funcionamento real do simulador. Os resultados obtidos foram satisfatórios, alcançando o objetivo desejado.

Os primeiros casos estudados mostram que o simulador é capaz de resolver os circuitos básicos conhecidos da literatura, descritos na seção 2.

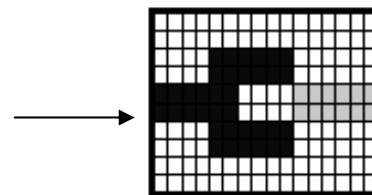


Figura 8 – Resultado da simulação de um inversor. A entrada é mostrada por uma seta e cada célula corresponde a quatro quadrados do grid.

O caso do inversor é ilustrado na figura 8, para a entrada igual a 0. Para ambas as entradas, “0” ou “1”, os resultados foram os esperados. Este circuito foi testado sem o uso de zonas de *clock* e a entrada está indicada por uma seta. A cor mais escura representa uma célula normal

polarizada com o valor 0 e a mais clara polarizada com valor 1.

Uma consideração importante é que os circuitos baseados em *majority gates* são as melhores opções de implementação nesse paradigma de QCA. Isto ocorre porque o *majority gate* e o inversor são os dois elementos lógicos fundamentais disponíveis no QCA. Outras portas lógicas são criadas com a combinação desses dois.

A operação de adição é um exemplo de circuito utilizando o *majority gate* e o inversor. Uma observação importante deve ser feita antes de mostrar o circuito. Para que o circuito funcione corretamente, deve-se garantir que todos os dados cheguem ao mesmo tempo ao *majority gate*. Caso contrário a informação que chegar primeiro será propagada para o restante do circuito. Para garantir o funcionamento correto deve-se usar zonas de clock. A figura 9 mostra o esquema do somador simulado, onde cada zona de clock é representada por diferente tons de cinza [27]. Além desse circuito, um somador serial também foi simulado com sucesso. Este último simplesmente utiliza uma saída do circuito anterior como uma das entradas, como pode ser visto na figura 10.

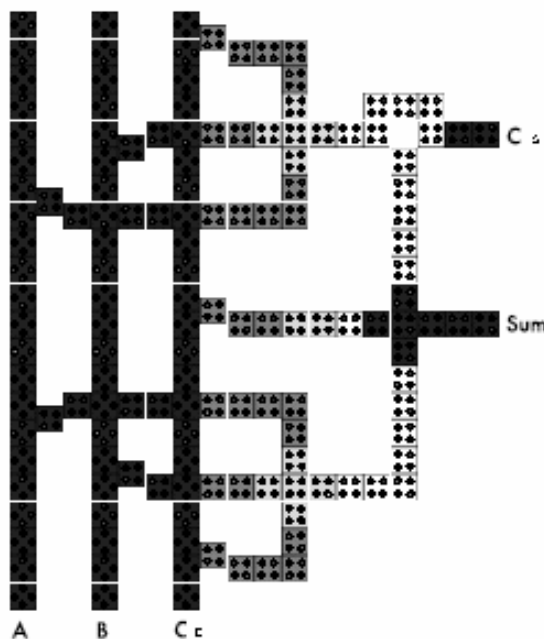


Figura 9 – Esquema do *full-adder* simulado.

6. Conclusão

Este trabalho propôs um simulador de Quantum-Dots Cellular Automata (QCA) utilizando uma rede neural do tipo hopfield. Os resultados dos testes foram satisfatórios. A utilização de simuladores para a pesquisa de circuitos lógicos utilizando o paradigma QCA é muito importante, visto que sua implementação real não é facilmente obtida.

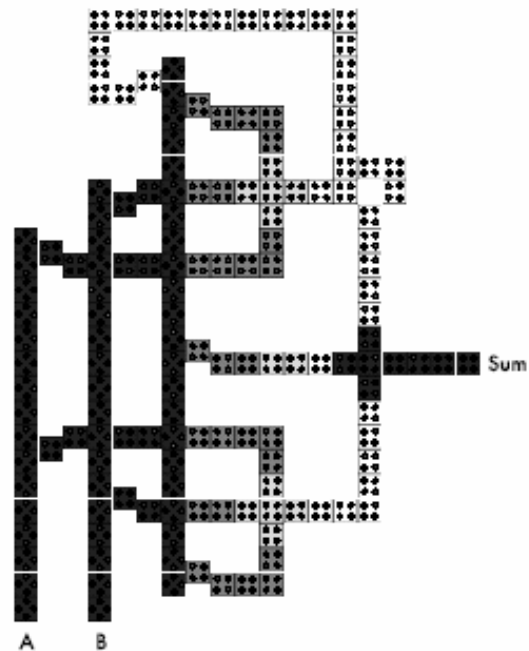


Figura 10 – Esquema do *full-adder* serial simulado.

Este simulador é uma ferramenta de QCA que possibilita ao usuário testar vários tipos de circuitos de forma fácil e correta.

Este trabalho mostra que a utilização de inteligência computacional para resolver problemas de nanotecnologia é um complemento às técnicas já utilizadas. Além das Redes Neurais, os Algoritmos Genéticos e a Lógica Fuzzy, por exemplo, podem auxiliar muito na pesquisa e desenvolvimento de novas tecnologias.

Para o futuro há a expectativa da utilização de outras ferramentas de inteligência computacional que possibilitem ainda mais o desenvolvimento dessa tecnologia. Ainda há muitos aspectos a serem estudados para o aprimoramento da arquitetura de QCA e este simulador deverá ajudar muito nestas novas descobertas.

7. Referências bibliográficas

- [1] C. S. Lent, P. D. Tougaw, and W. Porod, "Bistable saturation in coupled quantum dots for quantum cellular automata," *Appl. Phys. Lett.*, vol. 62, p. 714, 1993.
- [2] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, "Quantum cellular automata," *Nanotechnol.*, vol. 4, pp. 49-57, 1993.
- [3] P. D. Tougaw and C. S. Lent, "Logical devices implemented using quantum cellular automata," *J. Apply. Phys.*, vol. 75, pp. 1818-1825, Feb. 1994.
- [4] C. S. Lent and P. D. Tougaw, "A device architecture for computing with quantum dots," *Proc. IEEE*, vol. 85, pp. 541-557, Apr. 1997.

- [5] A. Fijany and B. N. Toomariam, "New design for quantum dots cellular automata to obtain fault tolerant logic gates," *J. Nanoparticles Res.*, vol. 3, pp. 27-37, 2001.
- [6] I. I. Yakimenko, I. V. Zozoulenko, C. K. Wang, and K. F. Berggren, "Influence of imperfections on the dynamical response in model quantum cellular automata," *J. Apply. Phys.*, vol. 85, pp. 6571-6576, 1999.
- [7] L. Bonci, G. Iannaccone, and M. Macucci, "Performance assessment of adiabatic quantum cellular automata," *J. Apply. Phys.*, vol. 89, pp. 6435-6443, 2001.
- [8] P. D. Tougaw and C. S. Lent, "effect of atray charge on quantum-dot cellular-automata," *Jpn. J. Apply. Phys.*, vol. 34, no. 8B, pp. 4373-4375, 1995.
- [9] M. T. Niemier, M. J. Kontz, and P. M. Kogge, "A design of and Design tools for a novel quantum dot based microprocessor," in *Proc. Design Automat. Conf., 2000*, IEEE Cat. No. 00CH37106, pp. 227-232
- [10] J. R. Pasky, L. Henry, and P. D. Tougaw, "Regular arrays of quantum-dot cellular automata macrocells," *J. Appl. Phys.*, vol. 87, pp. 8604-8609, 2000.
- [11] M. T. Niemier and P. M. Kogge, "Problems in designing with QCAs: layout=timing," *Int. J. Circuit Theory Appl.*, vol. 29, pp. 49-62, 2001.
- [12] J. C. Lusth and B. Dixon, "A characterization of important algorithms for quantum-dot cellular automata," *Inform. Sci.*, vol. 113, pp. 193-204, 1999
- [13] J. L. Cardenas-Barrera, K. N. Platoniotis, and A. N. Venetsanopoulos, "QCA implementation of a multichannel filter for image processing," *Math. Problems Eng.*, vol. 8, pp. 87-99, 2002.
- [14] M. Helsingius, P. Kuosmanen, and J. Astola, "Quantum-dot cells and their suitability for nonlinear signal processing," in *Proc. IEEE EURASIP Workshop Nonlinear Signal Image Processing*, vol. 2, 1999, pp. 659-663.
- [15] A. I. Csurgay, W. Porod, and C. S. Lent, "Signal processing with nearneighbor-coupled time-varying quantum-dot arrays," *IEEE Trans. Circuits Syst. I*, vol. 47, pp. 1212-1223, Aug. 2000.
- [16] P. D. Tougaw, C. S. Lent, and W. Porod, "Bistable saturation in coupled quantum-dot cells," *J. Appl. Phys.*, vol. 74, pp. 3558-3566, Sept. 1993.
- [17] C. S. Lent and P. D. Tougaw, "Lines of interacting quantum-dot cells: A binary wire," *J. Appl. Phys.*, vol. 74, pp. 6227-6233, Nov. 1993.
- [18] C. S. Lent and P. D. Tougaw, "Bistable saturation due to single electron charging in rings of tunnel junctions," *J. Appl. Phys.*, vol. 75, pp. 4077-4080, Apr. 1994.
- [19] A. O. Orlov, I. Amlani, G. H. Bernstein, C. S. Lent, and G. L. Snider, "Realization of a functional cell for quantum-dot cellular automata," *Science*, vol. 277, pp. 928-930, 1997.
- [20] I. Amlani, A. Orlov, G. Toth, G. H. Bernstein, C. S. Lent, and G. L. Snider, "Digital logic gate using quantum-dot cellular automata," *Science*, vol. 284, pp. 289-291, 1999.
- [21] A. O. Orlov, I. Amlani, R. Kumamuru, R. Rajagopal, G. Toth, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Experimental demonstration of clocked single-electron switching in quantum-dot cellular automata," *Appl. Phys. Lett.*, vol. 77, pp. 295-297, 2000.
- [22] A. O. Orlov, R. Kumamuru, R. Ramasubramaniam, G. Toth, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Experimental demonstration of latch in clocked quantum-dot cellular automata," *Appl. Phys. Lett.*, vol. 78, pp. 1625-1627, 2001.
- [23] A. O. Orlov, I. Amlani, R. Kumamuru, R. Rajagopal, G. Toth, J. Timler, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Power gain in a quantum-dot cellular automata latch," *Appl. Phys. Lett.*, vol. 81, pp. 1332-1334, 2002.
- [24] J. Timler and C. S. Lent, "Dissipation and gain in quantum-dot cellular automata," *J. Appl. Phys.*, vol. 91, pp. 823-831, 2002.
- [25] Hopfield J.J., (1984), "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci. USA*, 81:3088-3092.
- [26] Hopfield J.J., Tank D.W., (1985), "Neural computation of decisions in optimization problems," *Biological Cybernetics*, 52:141-152.
- [27] K. Walus, G.A. Jullien, V. Dimitrov, "Computer Arithmetic Structures for Quantum Cellular Automata" Copyright IEEE Asilomar Conference on Signals, Systems, and Computers, November 9-12, 2003, Pacific Grove, CA.