Neuromorphic Computation using Quantum-dot Cellular Automata

Enrique P. Blair and Scott Koziol
Electrical and Computer Engineering Department
Baylor University
Waco, TX 76798
enrique_blair@baylor.edu, scott_koziol@baylor.edu

***Abstract*—Quantum-dot cellular automata (QCA) is a paradigm for low-power, general-purpose, classical computing beyond the transistor era. In classical QCA, the elementary device is a cell, a system of quantum dots with a few mobile charges occupying some dots. Device switching is achieved by quantum mechanical tunneling between dots, and cells are interconnected locally via the electrostatic field. Logic is constructed by laying out arrays of QCA cells on a two-dimensional substrate. Several different implementations of QCA exist. Neuromorphic computing is computing which mimics aspects of how our brains compute. This includes parallel processing using highly interconnected primitives which combine local processing and memory. Viable neuron-like devices suitable for neuromorphic computation require a weighted signal fan-in, a way to aggregate signals, and a spike (pulse) output mechanism. The inputs to a neuron can be "excitatory" or "inhibitory" which refers to their ability to encourage or discourage a neuron to fire. We briefly review the concept of QCA and discuss how QCA cells satisfy these requirements. Viable implementations for QCA-based neuromorphism, and challenges that exist for implementing neuromorphic devices in QCA also will be discussed.**

***Index Terms*—quantum-dot, cellular automata, QCA, neuromorphic**

I. Introduction

We seek a new way of efficient computation. A positive energy efficiency trend has been identified with digital computation which shows that the number of computations per unit of energy has been increasing over time. This trend, however, is slowing down and approaching a line or "Efficiency Wall" [1]. If this trend continues it will negatively impact the growth of the computational capability of an all-silicon, all-digital approach to computation. This work examines the concept of neuromorphic computing based on quantum-dot cellular automata (QCA). QCA is a low-power, energy efficient computing paradigm offering scaleability to molecular dimensions [2], [3]. Given this efficiency wall, the significance of this work is twofold: low power and more capable hybrid QCA-neuromorphic solutions to provide longer operation times and more computing capability than currently available for power-budget-constrained systems such as mobile autonomous robots, smart phones, and Internet-of-Things (IoT) devices.

Section II provides bacground on the QCA concept. Section III describes how QCA can be used to implement Neuromorphic computing, and Section IV provides conclusions and challenges in QCA implementations of neuromorphic systems.

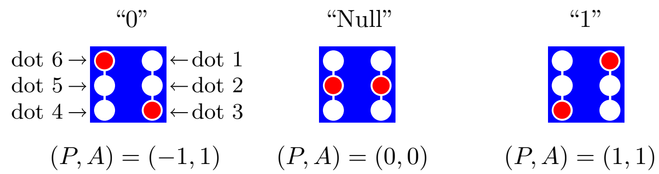

Fig. 1. The states of a six-dot QCA cell are characterized by polarziation (P) and activation (A). Here, each white disc represents a quantum dot, and each red disc represents one mobile electron. White connecting lines between dots indicate tunneling paths, with the consequence that tunneling between active states ("0" or "1") requires an intermediate tunneling through the "Null" state. The cell can be clocked to the Null state by applying a positive voltage ($V_{clk} > 0$) to dots 2 and 5 ("null" dots), or to an active state by applying a negative voltage ($V_{clk} < 0$) to the null dots. Clocking leads connected to the null dots are not shown here.

II. Quantum-dot Cellular Automata Background

QCA is a paradigm for classical computing using transistor-less logic [3], [4]. Here, systems of elementary QCA devices are arranged on a substrate and coupled not via electric current, but rather through Coulomb interaction via the local electrostatic field.

The primitive device in QCA is a cell, in which a system of quantum dots provide charge localization sites for a few mobile charges. A cell's charge configuration encodes a bit, and device switching occurs via quantum charge tunneling between dots. Fig. 1 shows a six-dot QCA cell with two mobile electrons in three states, designated "0", "Null", and "1", respectively. A cell can be set to the Null state by applying a positive voltage V_{clk} to dots 2 and 5 (designated "null dots"). Each state is characterized by two numbers: polarization P and activation A, which are functions of the mobile charge q_k on each dot ($k \in \{1, 2, \ldots, 6\}$):

$$P = \frac{q_1 + q_4 - q_3 - q_6}{Q}, \quad \text{and} \quad A = 1 - \frac{q_2 + q_5}{Q} \tag{1}$$

The valid ranges for P and A are $[-1, 1]$ and $[0, 1]$, respectively. Here, $Q = \sum_{k=1}^{6} q_k = 2q$ is the total mobile charge on all six dots, and q is the electronic charge. A bit is encoded on the sign of P, and signal strength can be measured by taking the absolute value of P. A weak bit has $|P| < 1$, and a full-strength bit has $|P| \simeq 1$.

Intercellular coupling through the electrostatic field provides for a logically complete set of QCA devices [5], [6],

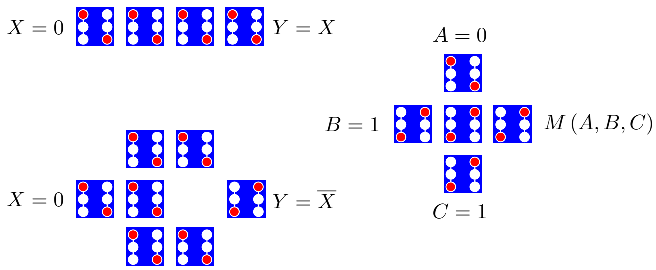

Fig. 2. QCA six-dot-cell logic. ***Upper left***: the output Y of a binary wire matches its input X; ***lower left***: an inverter uses diagonal interactions to flip a bit; and, ***right***: the majority gate is the natural logic gate in QCA. Three inputs A, B, and C exercise equal influence over the central device cell, which takes value $M(A, B, C)$, the bit in the majority on the inputs. This bit is copied to the output.

as depicted in Fig. 2. A row of cells tends to align, forming a binary wire. Diagonal coupling provides bit inversion. The majority gate is the natural logic gate in which three inputs "vote" on state of the central device cell. The majority wins, and this bit is copied to the output. Notably, one input to the gate may be used as a control bit, providing a programmable, two-input AND/OR gate between the two other inputs.

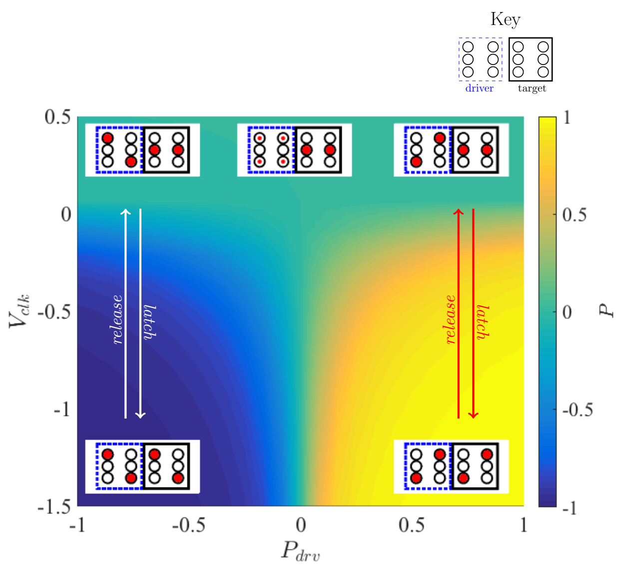

Fig. 3. The clock V_{clk} is used to vary the strength of the polarization response $P(P_{drv}, V_{clk})$ of the target cell. The polarization P_{drv} of a driver cell (cell with dashed blue outline in the insets) is the logical input to the target cell (cell with black outline in the insets), and the voltage V_{clk} is the clocking voltage applied to the central (null) dots of the target cell. The cell's response is its polarization state P, coded in the color gradient shown here. To write a "1" bit on the target cell, the driver is set to $P_{drv} = 1$ while the target cell is in the Null state ($V_{clk} > 0$), and V_{clk} clock is lowered to a strongly negative potential (following the red "latch" arrow). Weak or intermediate values of polarization ($|P| < 1$) result from a weakly negative clock (~ -0.5 V $< V_{clk} < 0$).

Clocking in QCA means controlling the whether a cell is in an active state or the "Null" state. This is achieved by applying a potential V_{clk} to the null dots. Fig. 3 shows the response of a clocked six-dot cell to a single neighboring driver cell. When clocked with a strongly-positive V_{clk}, the cell is driven to the "Null" state regardless of interactions with of neighboring cells. When V_{clk} is strongly negative, the target cell is clocked to an active state as determined by the neighboring driver cell. A cell cannot transition between active states without an intermediate transition through the "Null" state. Therefore, when clocked active, the cell is latched, because the clock prohibits a transition to the "Null" state. Another important result here is that intermediate polarization values are possible (i.e., $P \in [-1, 1]$), and the degree of polarization may be controlled by the strength of the clock: a weakly-clocked cell polarizes only partly, even when its neighbors are strongly polarized.

III. Neuromorphic computing with QCA

Neuromorphic computing traces its early roots to Carver Mead of California Institute of Technology [7], [8] [9]. There have been many analog neuromorphic solutions [10], [11] [12] as well as digital neuromorphic solutions [13] [14] [15]. Our QCA approach is a digital implementation approach. An earlier work considered unclocked QCA cells, however that method experiences a signal degradation problem [16]. In our proposed clocked cell method, signal degradation is not an issue as the clock provides signal gain to restore weakened signals. Also, clocking, as previously discussed in Section II, enables the selective weighting of connections. QCA devices suitable for neuromorphic computation require: 1) an adjustable weighted signal fan-in where the inputs to a neuron can be "excitatory" or "inhibitory" with reference to their ability to encourage or discourage a neuron to fire, 2) a way to aggregate (integrate) signals, and 3) a spike (pulse) output mechanism. The next sections describe candidate implementation methods of these three capabilities.

A. Synaptic weights

This paper presents two methods of synaptic weighting. First, partial clocking can be used to decrease the weight of some signals. Secondly, fan-out can be used to replicate a signal and send multiple copies to another cell.

1) Partial Clocking: Partial polarizations provide for relative weightings to a majority cell, as is shown in Fig. 4. Here, the inputs 1, 2, and 3 each are separated from the central device cell (cell 7) by one linking cell (cells 4, 5 and 6). If full-strength clocking is used on the coupling cells, then these linking cells provide full-strength inputs to the device cell. If, however, partial (weak) clocking is used for some linking cells (here, 4 and 6), those selected linkers polarize only partially and provide an input of diminished strength to cell 7. Under full and equal clocking for all linkers, the two inputs $P_1 = P_3 = -1$ ("0" bits) dominate the minority $P_2 = 1$ (a "1" bit); but, when cells 4 and 6 are only partially clocked, the inputs from 1 and 3 are transmitted only weakly to cell 7 so that the minority "1" bit can dominate the two weaker "0" bits. The simulation data shown here is for 20-nm-by-20-nm metallic QCA cells. An inter-dot tunneling energy of 50 meV was used. The inputs P_1, P_2, and P_3 were fixed, and for each data point, the time-independent Schrödinger equation was solved, and the ground state was calculated for cells 4-8.

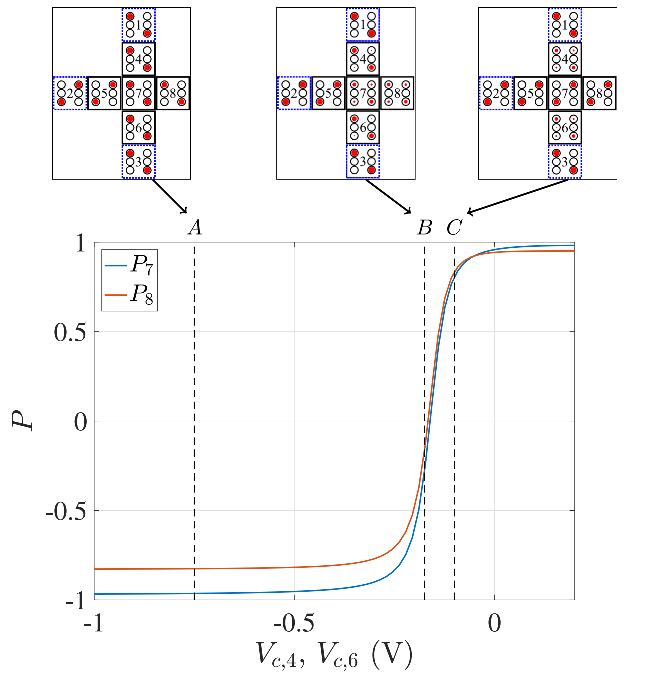

Fig. 4. Partial clocking can be used to modify the relative weights of inputs to a device. Full-bit inputs (cells 1, 2, and 3) couple to the majority cell (7) via cells 4, 5, and 6. Cells 1 and 3 provide a majority "0" bit, and clock $V_{c,M}$ is applied to cells 4 and 6. Cell 2 provides a minority "1" input, and clock $V_{c,m} = -1$ V is applied to cells 5, 7, and 8. Point A represents the case where $V_{c,M} \sim V_{c,m}$. Here, the majority input wins the device cell and is copied to the output. Point B represents partial clocking on cells 4 and 6. Inputs 1 and 3 now have a weaker coupling to cell 7, but the weakened majority "0" still dominates the minority "1". Point C represents the case where the majority bits couple so weakly to cell 7 that the fully-coupled minority "1" bit dominates the weaker majority input.

2) Fan-out replication: In QCA, it is most natural to have three inputs and one output because nearest-neighbor interactions maximize intercellular coupling. Thus, 3^n inputs can be fanned into a majority gate using cascaded majority gates, as in Fig. 5, where n is the number of cascaded stages. An additional way to vary the relative weights of inputs is through fan-out. A single bit can be fanned out using a circuit like that shown in Fig. 5 (right-hand side). Then, multiple copies of that bit can be provided to a dendritic tree, giving that bit more weight.

Additionally, a combination of these two methods can be used: selective input signal weighting through clocking, and fan-out-based signal multiplication.

B. Signal Integration and Spiking

The majority operation is used to implement signal integration and spiking. If the number of 1's exceeds the number of 0's then the QCA cell becomes a 1, which is equivalent to firing a spike. This integration is implementing a simple "Integrate and Fire" neuron [7]. Once the integration reaches a threshold, the neuron fires. Our threshold is adaptable by using fixed Excitatory or fixed Inhibitory cells. The combination of fixed cells and the number of synaptic connections sets the threshold, Fig 6. Fig. 6a shows six pre-synaptic signals arriving

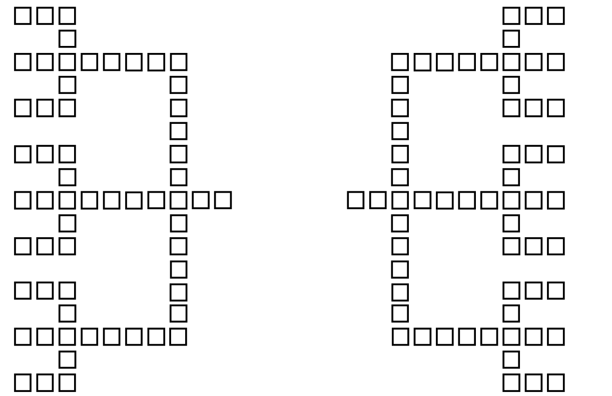

Fig. 5. Fan-in and fan-out circuits: a multi-stage fan-in circuit (left) provides for multi-bit fan-in. Intercellular interactions are maximized by nearest-neighbor coupling. Therefore, it is natural for a fan-in to have three inputs. Many inputs can be fanned in using multiple stages of majority logic, which function as three-input fan-in structures. *Right*: fan-out produces copies of a bit and may be used to increase the weight of a signal to another QCA cell. Multiple bit copies can be fed to a gate, increasing its weight through redundancy.

at this neuron. Each signal has a synapse (E through J). In this example, the threshold is set to 5. This can be deduced by applying (2-5). In this implementation, the synapses are turned on or off (i.e. "1" or "0"). The fixed cells are providing "0" bits. If in this case 2 or more input signals are excitatory, then the neuron fires (which is equivalent to outputting a 1). Fig. 6b shows a QCA abstraction of the fixed inhibitory cells. Fig. 6c shows how the threshold can be adjusted using excitatory fixed cells. This provides a binary (on or off) synapse method.

$$N_E = \text{\# of Excitatory Fixed Cells} \quad (2)$$
$$N_I = \text{\# of Inhibatory Fixed Cells} \quad (3)$$
$$N_S = \text{\# of synaptic connections} \quad (4)$$
$$\text{Threshold} = \left[\left(N_S + N_E - N_I\right)/2\right] - \left(N_E - N_I\right) \quad (5)$$

A second synapse method provides a way to create an adjustable synapse weight by using two QCA concepts: fan-out signal replication, and on-off clocking. In Fig. 7, there are two axons connecting to this neuron. They carry signals S_1 and S_2. Fig. 6 showed how these signals can be weighted using binary weights (i.e they either pass the signal, or attenuate the signal). Fig. 7 shows how axon signals can have non-binary synaptic strength. In Fig. 7, the pre-synaptic signals S_1 and S_2 are each copied three times. In this example, Inhibitory connections can be selectively used to create four different synaptic weight options: 0, 1/3, 2/3 and 1. For proper scaling, it is important that all input axons be copied the same number of times. The weight resolution depends on how many times each pre-synaptic signal is copied.

IV. Conclusions

We have proposed methods by which QCA can implement a few important operations needed for a neuromorphic system. We have shown two ways of implementing synaptic weighting

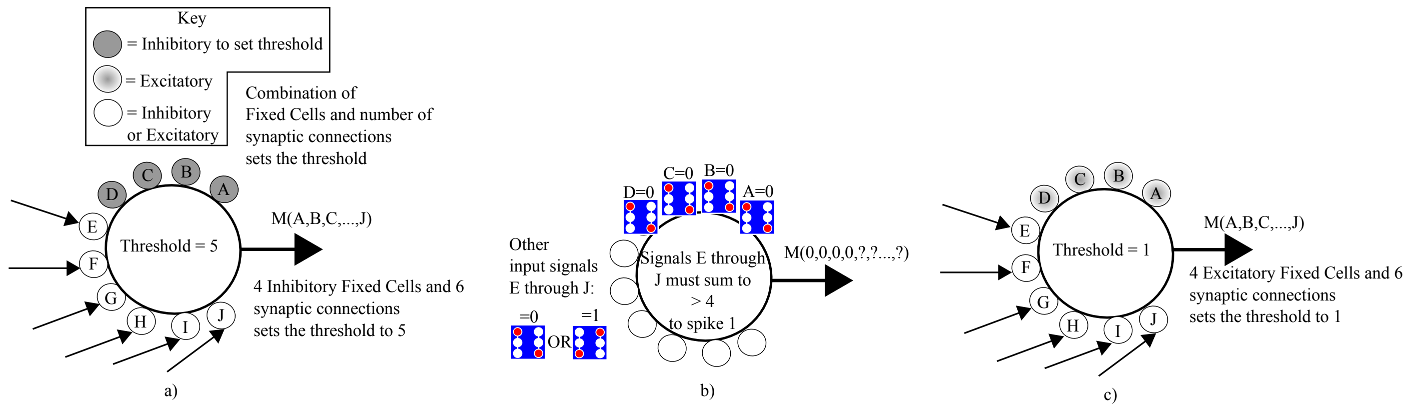

Fig. 6. Excitatory and inhibitory connctions and integrate and fire thresholds are shown in this picture. The thresholds can be adaptively set according to (5).

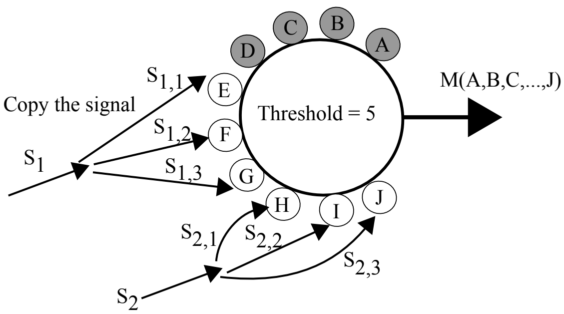

Fig. 7. Excitatory and inhibitory connctions and integrate and fire thresholds are shown in this picture. The thresholds can be adaptively set according to (5).

and described a candidate QCA integrate-and-fire neuron. We anticipate a Spike Timing Dependent Plasticity (STDP) learning method for determining the fan or cell clocking weights.

While there are numerous implementations for QCA, metal-dot [17] or semiconductor [18] implementations are envisioned for this application because cells may be individually clocked. In these implementations, however, fabrication is a significant challenge: presently, circuits with only very few QCA cells can be fabricated, as each additional cell dramatically increases the complexity of the layout. In-situ learning also is currently a challenge. Memristors, a candidate for neuromorphic implementations, have been shown to be immune to their variability [19]. The transition region between B and C in our numerical simulation in Fig. 4 shows that a small-scale weighted input system is sensitive to an imprecise clock. The ramifications of this for a multi-neuron system are an area of future research.

References

[1] B. Marr, B. Degnan, P. Hasler, and D. Anderson, "Scaling energy per operation via an asynchronous pipeline," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 21, no. 1, pp. 147–151, Jan 2013.

[2] G. Snider, A. Orlov, R. Kummamuru, R. Ramasubramaniam, I. Amlani, G. Bernstein, C. Lent, J. Merz, and W. Porod, "Quantum-dot cellular automata: Introduction and experimental overview," in *Proc. of the 2001 1st IEEE Conf. on Nanotechnol.* IEEE, Oct 2001.

[3] C. S. Lent, "Molecular electronics - bypassing the transistor paradigm," *Science*, vol. 288, pp. 1597–1599, Jun. 2000.

[4] C. Lent, P. Tougaw, W. Porod, and G. Bernstein, "Quantum cellular automata," *Nanotechnology*, vol. 4, p. 49, 1993.

[5] P. Tougaw and C. Lent, "Logical devices implemented using quantum cellular automata," *J. Appl. Phys.*, vol. 75, no. 3, pp. 1818–1825, Feb. 1994.

[6] C. Lent and P. Tougaw, "A device architecture for computing with quantum dots," *Proc. of the IEEE*, vol. 85, no. 4, pp. 541–557, April 1997.

[7] C. Mead, "Analog VLSI and neural system," *Ed. Addison Wesley. USA*, 1989.

[8] C. Mead and M. Ismail, *Analog VLSI implementation of neural systems*. Springer, 1989.

[9] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct 1990.

[10] S. Brink, S. Nease, P. Hasler, S. Ramakrishnan, R. Wunderlich, A. Basu, and B. Degnan, "A learning-enabled neuron array ic based upon transistor channel models of biological phenomena," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 7, no. 1, pp. 71–81, 2013.

[11] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[12] S. Moradi and G. Indiveri, "An event-based neural network architecture with an asynchronous programmable synaptic memory," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 1, pp. 98–107, Feb 2014.

[13] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[14] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The spinnaker project," *Proc. of the IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.

[15] A. Cassidy, A. G. Andreou, and J. Georgiou, "Design of a one million neuron single fpga neuromorphic system for real-time multimodal scene analysis," in *2011 45th Annual Conference on Information Sciences and Systems*, March 2011, pp. 1–6.

[16] W. Zhao, D. Querlioz, J. O. Klein, D. Chabi, and C. Chappert, "Nanodevice-based novel computing paradigms and the neuromorphic approach," in *2012 IEEE International Symposium on Circuits and Systems*, May 2012, pp. 2509–2512.

[17] A. Orlov, I. Amlani, G. Bernstein, C. Lent, and G. Snider, "Realization of a functional cell for quantum-dot cellular automata," *Science*, vol. 277, no. 5328, pp. 928–930, 1997.

[18] M. Mitic, M. Cassidy, K. Petersson, R. Starrett, E. Gauja, R. Brenner, R. Clark, A. Dzurak, C. Yang, and D. Jamieson, "Demonstration of a silicon-based quantum cellular automata cell," *Applied Physics Letters*, vol. 89, no. 1, p. 013503, 2006.

[19] D. Querlioz, O. Bichler, and C. Gamrat, "Simulation of a memristor-based spiking neural network immune to device variations," in *The 2011 International Joint Conference on Neural Networks*, July 2011, pp. 1775–1781.