IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 2, Ver. III (Mar-Apr. 2016), PP 72-76
www.iosrjournals.org

Image Restoration Algorithm for Reliable Block Quantum Cellular Automata Gate

Patrick Harrington[1]
[1](Department of Mathematics & Computer Science, Northeastern State University, Tahlequah, OK, United States)

Abstract : *The fault-tolerance properties of the classical Quantum Cellular Automata (QCA) Majority gate, as well as the AND-gate, suffer from precision errors in inherent in manufacturing on a nanometer scale. Due to these implementation problems, a new approach for QCA was developed. A solution using multiple QCA cells in block configuration is an improvement but will may not work depending on the number and pattern of cells that are out of alignment. Image segmentation and restoration algorithms have been previously used successfully in manufacturing automation to correct or compensate for errors in fabrication. This proposal investigates and draws out the mathematical model and approach to apply a neural network-based image restoration algorithm to the QCA block Majority gate, as well as the AND-gate, to produce greater reliability. Study of effectiveness is presented.*

Keywords: *Neural networks, machine learning, Markov Random Fields, Competitive Learning*

I. Introduction

Circuits on a quantum level have been developed and explored, significantly at the University of Notre Dame [1,2,3,4,5]. Using quantum dots, these Quantum-dot Cellular Automata (QCA) have the potential for significant improvement over conventional VLSI circuits, using little power and having greater throughput. By placing quantum cells near one another, data is transmitted between quantum cells by Coulomb interaction between positively and negatively charged quantum dots in each cell, transferring charge, and data, along the array or grid [1]. The negatively charged (-1) quantum dots are used to represent binary 0, with positive (+1) representing binary 1.

Despite their potential, actual realization of QCA as originally proposed is difficult. Tests of QCA, particularly the Majority gate, have shown that they are very sensitive to errors in cell misalignment, diminishing or reversing the Boolean logic between the cells. Fijany and Toomarian [6] demonstrate this and propose a block Majority gate, which is made up of an array of cells that together functions as the Majority gate but is much more robust when manufacturing errors are taken into account. However, changing the location and symmetry of the inputs affected the reliability by altering the logic function of the QCA gate.

A Markov Random Field modeling algorithm was first put forth by Rangarajan and Chellappa [7]. The authors in [8] wrote a competitive learning, neural network-based algorithm to use Markov Random Fields for a Lyapunov function in a Hopfield network, which we shall abbreviate CLRS. It was shown to be effective in restoration of images that have Gaussian noise. This algorithm belongs to a class of algorithms that have been applied to other areas besides imaging, particularly that of manufacturing. Our proposal will show the mathematical model and approach of applying CLRS to the QCA block Majority gate, as well as a specific case of the Majority gate, the AND-gate, to correct errors that cause the block QCA gate to malfunction.

II. Background

QCA research has become more prominent due to the shrinking size of CMOS and VLSI circuits, which are at nanometer scale. At this level, things become so small that the physical limitations become an overriding factor. In addition to expense of manufacturing, other forces besides electric current affect reliability, making the need for alternative circuits more significant. Detailed simulations of these defects are demonstrated in [9], showing that only a few nm are sufficient to destroy the gate logic.

2.1 Quantum Cellular Automata

Using quantum dots, QCA encode binary information based on their magnetic charge. Quantum dots are small groups of atoms, such as silicon, whose size is small enough that its behavior is dominated by quantum forces. Therefore, these "artificial atoms" are easier to manipulate than traditional atoms for quantum computation. The basic unit in QCA is a cell containing four quantum dots in a square. Two mobile electrons exist in the cell, interrupted by barriers of potential between them that can be raised or lowered to control their movement. The electrons are able to move between quantum dots by electron tunneling. The polarization of the electrons, positive or negative, is used to set or read the binary encoding of the cell. If the electrons are negatively charged, then the cell contains a binary 0 value; if it is positively charged, then it contains a binary 1. Interaction between cells is done without electrical currents, depending on the interaction between cells based on the charge of the electrons. By placing a QCA cell adjacent to another cell, the electrons are changed by neighbor cell's polarity. These Coulomb interactions give the QCA its computation power. [1]

The classic or simple Majority gate takes inputs from three cells, uses the majority value of these inputs to set a device cell, which is then transmitted as output (Fig. 1).

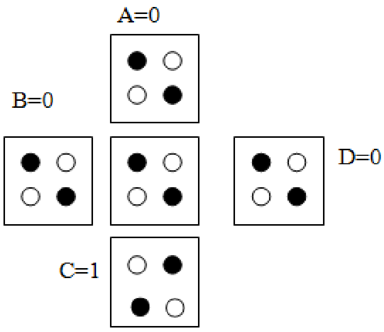

Figure 1: majority gate

Here, we see the three inputs, A, B, and C, having values of binary 0, 0, and 1, respectively. The device cell takes the polarity of the majority of the cells, -1, which is transmitted along to the output cell D. The device cell works this way because it seeks to find the lowest energy state, or ground, among the charges of the three input cells, which are fixed. [3] The AND-gate is a specific case of the Majority gate that can be represented by fixing the binary value of one of the input cells to 0, which is a polarization of -1 charge.

Despite their potential, QCA has a significant weakness because of its lack of fault-tolerance to errors in manufacturing that can occur with respect to defects in assembly precision and lithography. QCA cells have been demonstrated by [6] to be highly sensitive to cell alignment. Vertical imprecision of ½ cell width causes error in cell interaction, leading to the opposite logical effect for any gate. Current manufacturing techniques have difficulty with the precision needed: for quantum dot diameter of 5 nm, cell size of 20 nm, and distance between cells of 14 nm, the margin of horizontal error is less than 10 nm [6, 10] or as small as 5 nm [9] (Fig. 2).

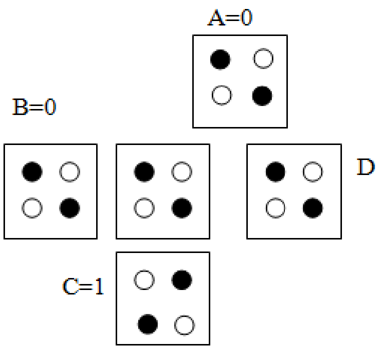

Figure 2: non-functional majority gate

QCA cells were shown to be highly sensitive to other errors in alignment [6]. Vertical misalignment, even when perfectly linear, caused QCA gates to malfunction when the input cells were moved 20 nm from the device cell. Further research has been done by [11] using Hopfield neural network design of QCA for device uncertainty. Additionally, [9] outlines a vector based approach for error detection in manufacturing. The diagram (Fig. 3) shows the layout of a block Majority gate when all of the cells are laid out in a regular geometric pattern. There are three inputs which are either asymmetrical, as in Fig. 3, or symmetrical.

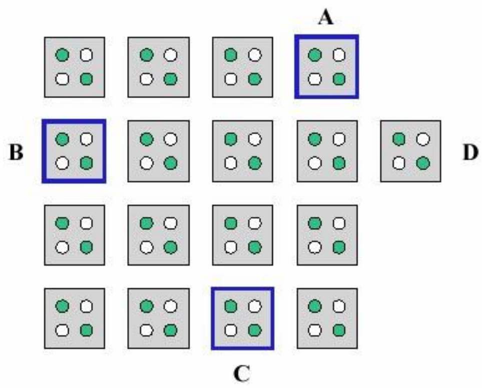

Figure 3: block majority gate

The block Majority gate was shown to be robust when a limited number of cells are defective, along with additional cells that are laid out irregularly. Using multiple cells (three) as input on each side helped keep the defective gate functional. However, changing the location and symmetry of the inputs altered the logic function of the QCA gate. There were also cases where regardless of the input, the pattern of cell misalignment and defect resulted in a non-functional block majority gate. [6]

2.2 Neural Networks

The basis and computation power of the CLRS algorithm is in neural networks. Artificial neural networks are made up of densely interconnected sets of relatively simple units, called perceptrons, with each unit having multiple sets of real-valued input and output traveling between them. Neural networks are sometimes described in terms of knowledge layers, with the more complex having deeper layers that together act as a massively parallel distributed processor. Neural networks are very robust when handling data errors. They often use a learning rule that controls the information flow and minimizes a cost function, energy function, or complex combination of functions that describes the interaction among the nodes. [12]

Perceptrons calculate a combination of inputs based upon some function, and output a 1 (Boolean 1) if the result is greater than a certain threshold and -1 (Boolean 0) otherwise:

$$o(x_1, \ldots, x_n) = \{1 \text{ if } -w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n > 0 \quad -1 \text{ otherwise } \} \tag{1}$$

In (1), each w_i is a weight that determines the contribution of input x_i to the perceptron, with $-w_0$ as a threshold that the weighted combination of inputs must surpass for the perceptron to output a 1. A single perceptron is sufficient to represent the AND- Boolean function or gate, as well as the Majority gate. A two-input perceptron can implement the AND- function by setting the weights $w_0 = .8$, and $w_1 = w_2 = .5$, or the Majority function by $w_1 = 1$ and threshold $n/2$. [12]

In the CLRS algorithm, the principle of competitive learning is derived from the brain's ability to preserve neighborhood relations among neurons as input signals (sound, sight, touch) move between them. This theory is relaxed to prevent one node from dominating other nodes in the network, so that the entire system is able to update together at a regular rate, and thereby also incorporates additional Hebb-type network properties into the algorithm. This combined with Bayesian principles and Markov Random Fields (MRF) allow for restoration of a noise-corrupted image. Image recovery and segmentation algorithms have a range of applicability beyond the scope of imaging into other areas, including manufacturing techniques, which are the core of QCA gate unreliability. The CLRS algorithm adjusts the weights of each of the cells in the neural network as it corrects errors. It works better than most of the classic image segmentation and restoration algorithms, which are often mean- and statistic-based. CLRS is based upon a representation of the low-level mammal vision system, and uses an iterative method to minimize an energy function which in turn makes the image restoration possible. The algorithm works locally between pixels (nodes) and their immediate neighbors, as well as globally across the system of nodes in the grid. [8]

III. Proposed Approach

To perform restoration and segmentation of the block QCA gate by using the CLRS algorithm, its mathematical equations will be incorporated into a program. For the AND-gate and the Majority gate, the threshold values for one will be applied to the system as a whole; that is, since the entire system of cells makes up the logic gate, if the entire system considered as a perceptron is above the threshold values for a given gate, then the corresponding positive charge (Boolean value of 1) is output, with a negative charge (Boolean value 0) output otherwise. For a given node (i, j), the grid of nodes $m \times n$ is denoted by:

$Z_{mn} = \{(i, j): 1 \leq i \leq m, 1 \leq j \leq n \}$ (2)

with the local neighborhood system:

$F = \{ F_{i,j} , (i, j) \in Z_{mn}\}$ on Z_{mn} (3)

where $F_{i,j} \subseteq Z_{mn}$ are the neighbors of (i, j). The node (i,j) has eight nearest neighbors, $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i, j+1)$, $(i+1, j-1)$, $(i+1, j)$, $(i+1, j+1)$. The entire grid of nodes or cells is modeled as the triplet (F, H, V), with the grid of node intensities F, horizontal discontinuities or errors in placement H, and vertical discontinuities V. Thus, we have:

$F = \{ f_{i,j} \} , (i, j) \in Z_{mn}$ (4)

$H = \{ h_{i,j} \} , (i, j) \in Z_{mn}$ (5)

$V = \{ v_{i,j} \} , (i, j) \in Z_{mn}$ (6)

F, H, and V are modeled as MRFs, with a stochastic process over Z_{mn} and neighborhood F, thereby allowing for the algorithm to use an iterative method that takes into account a matrix of prior values for each local neighborhood's update iteration. These prior values are used to obtain a posterior estimate that indicates the likelihood of the solution $f_{i,j}$ at the current iteration, given the prior data $g_{m,k}$ from the previous iteration. The reconstruction of the correct grid pattern using the neural network can thus be solved as an equation that finds an estimate maximizing the likelihood of the solution given the prior data, or in other words, minimizes the expected value of an error.

The algorithm takes the values of each cell represented as a real number, and looks at the relationship of this to the rest of its neighbors, as well as the entire system as a whole. It then attempts to smooth out any inconsistencies that might indicate misleading data, or error in cell placement that give an incorrect strength or weight to the charge of a particular cell in relation to another cell. It also takes into account prior estimates for the cell and its neighbors in calculating what the correct weight and value for the cell should be, so that it does not overcompensate and is able to get closer to the actual value. By doing so, this algorithm resolves discontinuities between estimates of the correct value for the cell and its neighbors as iterations progress. It seeks what it believes to be the correct value for the cell; in this regard, it seeks to find the best ground state among the cell and its neighbors. This will in turn be used to calculate a weighted value based upon all of the cells together acting as one unit, or logic gate.

IV. Results

Below is the mathematical model based on CLRS that will be used as applied to the block QCA gates. This model details the reasoning behind the expected results of compensating for the fabrication errors to produce a QCA gate with greater reliability. The algorithmic process to correct errors is based upon an iterative method to minimize an energy function, which in turn makes the restoration possible. This energy function and its related mathematics are derived from a representation of part of the mammal vision system. The algorithm works locally between pixels (nodes) and their immediate neighbors, as well as globally across the system of nodes in the grid.

For any QCA cell located at node (i, j) in the grid represented in the network, its nearest neighbors are $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i, j+1)$, $(i+1, j-1)$, $(i+1, j)$, $(i+1, j+1)$. The charge of the cell and its neighbors in the local neighborhood system F, either positive +1 (Boolean 1) or negative -1 (Boolean 0), is defined as a matrix (G_{ij}) of $M \times N$, where G_{ij} is the charge of cell (i, j). An update rule that describes the changes and relationship between the charge of a cell and its neighbors takes the following form:

$$\Delta f_{i,j} = \alpha \sum_{mk} \Lambda \, [(i,j), (m, k), K](g_{m,k} - f_{i,j}) + \beta K(f_{i,j-1} + f_{i, j+1} - 4f_{i,j} + f_{i-1,j} + f_{i,j}) \quad (7)$$

In (7) $f_{i,j}$ is the new estimate of the correct (F, H, V) representing the node intensity F, and horizontal and vertical displacement H and V, respectively, for (i, j) at a given iteration. And, $g_{m,k}$ is the estimate from the previous iteration that is used in the calculations for the current iteration; however, $f_{i,j}$ is equal to $g_{m,k}$ during the first iteration. These variables represent the net charge during each iteration, taken from $G_{i,j}$. These prior values are used to obtain a posterior estimate that indicates the likelihood of the solution $f_{i,j}$ at the current iteration, given the prior data $g_{m,k}$ from the previous iteration. Additionally:

$$\Lambda \, [(i,j), (m, k), K] = (e^{-d*d/(2k*k)})/(\textstyle\sum_{m.k} e^{-d*d/2K*K}) \quad (8)$$

K and α are parameters to be adjusted during the implementation, with α denoting the learning rate of the neural network. K is used in representing the strength of the prior estimates of a neighbor node contribution for a given iteration, which is usually set from a range of 0.1 to 1.0 to obtain the most mathematically accurate results. Also:

$d = f_{i,j} - g_{m,k}$ (9)

Equation (7) calculates the best value from the prior estimates that compete to update the new estimate of the correct value of node (i,j). The global optimization function E_0 which is the energy function that is to be minimized in order to reduce the expected value of an error and thereby obtain the best results, is defined as follows:

$$E_0 = \alpha K \sum_{i,j} ln \sum_{m,k} e^{-d*d/2K*K} + \beta \sum_{m,k} (f_{i,j-1} + f_{i,j+1} - 4f_{i,j} + f_{i-1,j} + f_{i,j})^2 \quad (10)$$

The update rule given above for $\Delta f_{i,j}$ in (7) minimizes this global optimization function, which thus allows for the reconstruction of the correct grid pattern.

From here, the QCA gate can be considered as either an AND-gate or Majority gate depending on the threshold values established for the entire system as a gate. From perceptron theory, applying these rules allows for a value of 1 to be output by the system if the result of the system is greater than a certain threshold; otherwise, a -1 or Boolean 0 is output. The mathematics behind this are in (1). Here, the threshold is represented by $-w_0$ that the weighted combination of inputs must surpass for the output to be a 1. For a single node, the AND-gate is implemented by setting the weights $w_0 = .8$, and $w_1 = w_2 = .5$, or the Majority function by $w_1 = 1$ and threshold $n/2$.

V. Conclusion

The simple majority and gate has been shown in other work as unreliable with regard to actual implementation. The alternative solution, the block Majority gate, is more robust to actual errors in manufacturing, but still suffers from several faults. Our alternative solution using the CLRS image restoration and segmentation algorithm builds upon the foundations whereby classes of similar algorithms have been used to correct defects in manufacturing. This algorithm has the potential for making QCA a much more realistic technology by correcting errors in cell placement before the circuit is used. The application of the block majority QCA gate to the AND gate will be an interesting experiment, one which should demonstrate the robustness of the design.

References

[1] P. D. Tougaw and C. S. Lent, Logical device implementation using quantum cellular automata, Journal of Applied Physics, (75), Feb. 1994, 1818-1825.

[2] C.S. Lent and P.D. Tougaw, A device architecture for computing with quantum dots, Proceedings of the IEEE, 85(4), 1997.

[3] M.T. Niemier, M.J. Kontz, P.M. Kogge, A design of and design tools for a novel quantum dot based microprocessor, Design Automation Conference, June 2000, 227-232.

[4] W. Porod, Quantum-dot devices and quantum-dot cellular automata," International Journal of Bifurcation and Chaos, 7(10), 1997, 2199-2218.

[5] G. L. Snider, A. O. Orlov, et. al. , Quantum-dot cellular automata, Journal of Vacuum Science Technology A, 17(4), 1999.

[6] A. Fijany and B. N. Toomarian, New design for quantum dots cellular automata to obtain fault tolerant logic gates, Journal of Nanoparticle Research, 3, 2001, 27-37.

[7] A. Rangarajan, R. Chellappa, Markov random field models in image processing, in The Handbook of Brain Theory and Neural Networks, (MIT Press, 1995) 564-567.

[8] V.V Phoha, W.J.B. Oldham, Image recovery and segmentation using competitive learning in a layered network, IEEE Transactions on Neural Networks, 7(4), 1996, 843-856.

[9] M. B. Tahoori, J. Huang, M. Momenzadeh, F. Lombardi, Testing of quantum cellular automata, IEEE Transactions on Nanotechnology, 3(4), 2004.

[10] C. D. Armstrong, W. M. Humphreys, A. Fijany, The design of fault tolerant quantum dot cellular automata based logic", 11th NASA Symposium on VLSI Design, 2003.

[11] E.N. Ganesh, L. Kishore, M. Rangachar, Simulation of quantum cellular automata circuits using neural networks, International Conference on Computational Intelligence and Multimedia Applications, 2007.

[12] T.M. Mitchell, Machine Learning, (McGraw-Hill, 1997).