FAULT TOLERANT QUANTUM-DOT CELLULAR AUTOMATA MAJORITY GATE DESIGN

By

ZACHARY DANIEL PATITZ
Bachelor of Science in Computer Science

Oklahoma State University

Stillwater, OK

2002

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfullment of
the requirements for
the Degree of
Master of Science
May, 2006

FAULT TOLERANT QUANTUM-DOT CELLULAR AUTOMATA MAJORITY GATE DESIGN

Thesis Approved:

Dr. N. Park

Thesis Advisor

Dr. K. M. George

Dr. V. Sarangan

Dr. Gordon Emslie

Dean of the Graduate College

Preface

This thesis presents reliable geometric QCA cell structures for designing single clock-controlled majority gates with a tolerance to radius of effect-induced faults as a basic building component for carry lookahead adder. Realizable quantum computing is still far in the future due to the complexity of quantum mechanics that govern them. In this regard, QCA-based system design is a challenging task since each cell's state must interact with all the cells that are in its energy-effective range with respect to its corresponding clocking zone, referred to as its *radius of effect*. The proposed geometric design approach for majority gates in this thesis is to overcome the constraints imposed by the radius of effect of each cell with respect to clock controls. We will show majority gate structures that will operate with multiple radius of effect-induced faults under a single clock control. The design approach to a single clock controlled majority gate will ultimately facilitate more efficient and flexible clocking schemes for complex QCA designs. The focus will be on molecular scale designs, as these are the future of QCA structures that will operate at room temperature. It will be shown that these single clock zone majority gates can be used to create a reduced clock cycle carry-look-ahead full-adder that is more flexible with respect to clocking zone size and placement as well as radius of effect faults. The effectiveness of the designs will be show through results simulated under projected system environment properties.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

1 Introduction

One of the major hurdles that needs to be overcome in quantum computing is defect- and fault-tolerance. Quantum-dot cellular automata (QCA) are no different in this respect. QCA are composed of a number of cells each of which contains four dots where electrons may lie. The small structures are subject to manufacturing defects as well as other faults. A QCA cell may interact with too many of its neighboring cells and cause erroneous operation of the most basic functions. This proposal introduces designs that can tolerate radius of effect-induced faults by redesigning some of the basic structures, namely the majority gate.

In section 2, we will discuss QCA cells and basic structures that are used to create complex arrays for computation. Section 3 will cover the clocking scheme that is used to control large arrays of cells. Section 4 shows a carry-look-ahead full-adder. Section 5 will discuss the faults that can occur due to an increase or a decrease within the radius of effect of each cell. Section 6 introduces structures that can tolerate various radius of effect-induced faults while under only a single clock control to perform a majority-gate operation. Simulation results for the structures will be shown. Section 7 shows the structures which will be used for molecular level QCA majority gates and also shows simulation results for the majority gates. Section 8 places the modified majority gates into a large array (namely an adder) to show that they funtion correctly when used in more complex designs. The conclusion will review the work done and its possible impact on the future of QCA design and fault-tolerance.

2 Basic QCA Structures

Each of the computing elements is composed of a number of quantum dot cells, the building blocks of QCA. Each quantum dot cell can represent a binary one or zero, or a superposition of the two states. The next structure is a simple wire composed of a number of quantum dot cells; a QCA quantum wire that can transmit binary information without current. The two most basic computing elements in a QCA are the majority gate and the inverter (the NOT gate). These can be used to realize AND, OR, NOT, and NAND gates when combined. Also, wire crossings are essential to the implementation of a complex two-dimensional array and will also be discussed.

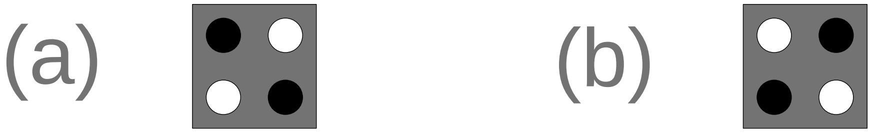

Figure 1: The darkened dots represent the location of the electron in a given cell. The two cell states are represented here, (a) shows polarity P=-1 (binary 0) and (b) with polarity P=+1 (binary 1)

The composition and states of individual quantum dot cells are presented as follows. In Figure 1 cells in binary state 0 (a) and binary state 1 (b) are shown. The darkened quantum dots of the cells in Figure 1 represent the location of two extra mobile electrons in the cell. As can be seen in Figure 1, the mobile electrons migrate to antipodal sites of the cell by their natural electrostatic repulsion to represent their respective binary states [3, 11, 12]. The dot configuration of the cell can also be rotated 45 degrees to represent a rotated cell [11] where the electrons also occupy antipodal sites within the cell. The rotated cell is important for use in wire crossings and inverter wires. We assume that the potential barriers between cells is at a level such that the electrons will not migrate, or tunnel, to neighboring cells. Cells will, however, react with one another through the Coulombic repulsion between their electrons. With these cells we can construct the components of a QCA, the first of which is a binary wire.

In Figure 2 (a) we show a basic QCA binary wire with the input cell on the left and the output cell on the right. We show left to right operations since this is the most typically seen flow of an array of cells, though the direction will not matter

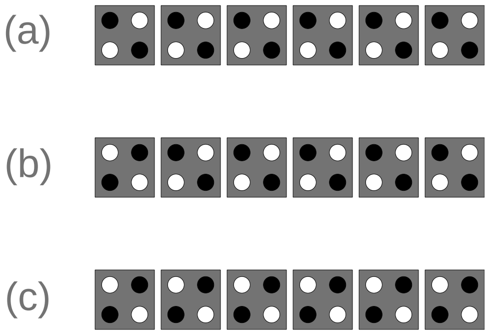

Figure 2: A binary wire in steady state (a), with input P=+1 (b) and after polarity has propagated down the wire (c)

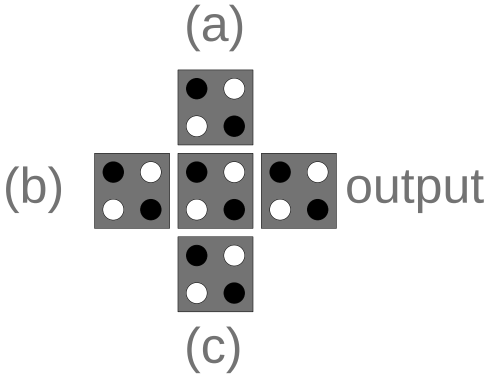

Figure 3: A majority gate with inputs (a), (b) and (c) at polarity P=-1

for simple structures. In (a) the wire is at a stable state with input and output as binary 0. In Figure 2 (b) we change the polarization of the first (input) cell to P=+1 to create an unsteady state in the wire. Figure 2 (c) shows the result of this polarization change as the state of the input cell is propagated down the line of cells via the Coulombic interaction of the cells. That is, when the input cell is forced into polarization state P=+1 (and held in that state) the electrons in the quantum dots are redistributed to the opposite antipodal configuration of the cell. As the input is held at its new polarity its neighboring cell reacts by also changing to polarization state P=+1. This reaction takes place down the line until reaching the output cell, thus transmitting a binary value 1 down the wire without using a current, but instead using the Coulombic repulsion between the cells and their electrons. Note that this process does not happen instantaneously. Instead, there is a certain amount of delay and dissipation which occurs with each cell that must be switched from an unstable state (relative to adjacent cells).

Majority gates are a pivotal part of QCA designs since they can come to represent an OR gate or an AND gate. The majority gate consists of three inputs and a single output. The bit value represented by the majority of the inputs is propagated to

Table 1: Majority gate inputs and resulting outputs

a	b	c	output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

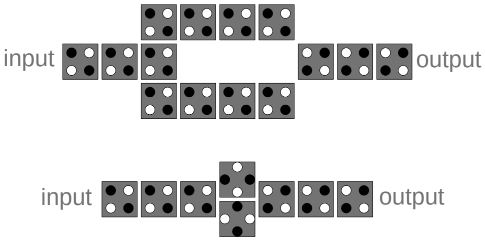

Figure 4: Two inverters with inputs at polarity P=-1 (binary 0) and output polarity P=+1 (binary 1)

the output. The structure of a majority gate is shown in Figure 3. Figure 3 shows a majority gate at steady state with all inputs (a), (b) and (c) at polarity P=-1 (binary 0). The Coulombic repulsion of the three input cells forces the middle cell, or the device cell, to polarity P=-1, which in turn forces the output cell to P=-1 [3, 11, 12, 14]. In Table 1 the behavior of the majority gate function can be seen with all possible input configurations.

As is implied by Table 1, one can create an AND or an OR gate by simply locking one of the inputs in the P=-1 state or P=+1 state, respectively. In Table 1 it can be seen that by setting input (a) to 0, one can create an AND gate for inputs (b) and (c). Conversely, by setting input (a) to 1, one can create an OR gate for inputs (b) and (c). The next major component that is needed is an inverter, or a NOT gate.

The function of the inverter is to take as an input either a binary 0 or 1 and output 1 or 0, respectively. Figure 4 shows two inverters. In the first, the input branches into two wires and on the other end the cells are offset so that when they interact with the output wire to reverse its polarity (with respect to the input) [3, 11, 12]. The second inverter simply uses two offset and rotated cells to reverse the signal traversing the wire. The value of an input can also be inverted by correctly pulling the value off an inverter wire.

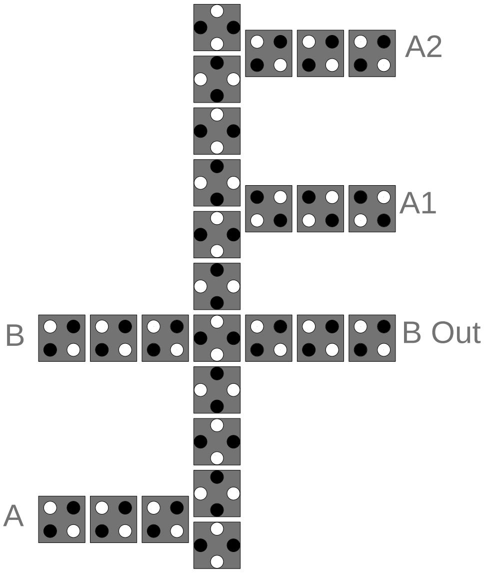

Figure 5: A wire crossing with input A, outputs A1 and A2 and wire B.

Wire crossings and inverter wires structures are shown in Figure 5. The input cell A will determine the polarity of the vertical inverter wire. Output A1 will reflect the inverted value of input A by pulling the value off of the wire an odd number of cell widths from the driving wire. That is, the distance between the top of line A and the bottom of line A1 is $5w + 6s$, where w is the width (height) of a cell and s is the uniform cell spacing distance. Output A2 will be the same as the input since it is an even number of cell widths from the input wire, a distance of $8w + 9s$ from the top of line A to the bottom of line A2. The wire B will be unaffected by the vertical wire since no matter what state the inverter wire is in it will have a perfectly balanced effect on wire B. Pushing values onto, pulling values off of, and crossing a vertical wire are all pivotal in creating a full adder or any other significantly complex 2-D QCA structure.

3 Clocking of QCA

With the simple structures discussed above, it has been shown that one can create wires, logic gates, shift-registers, memories and even a simple microprocessor [12]. For these large circuits we need to be able to clock-control the movement of bits from cell to cell.

The clocking of cells is achieved by electrostatically switching the cell between three different states. The first state is the *null state*, in which the cell holds no binary data. The second state is the *switching state*, in which the cell takes on the value of its neighboring cell through Coulombic interaction between the two. The third state is the *locked state*. In the locked state the cell cannot be affected by neighboring cells and is therefore retained in its current state [9, 21].

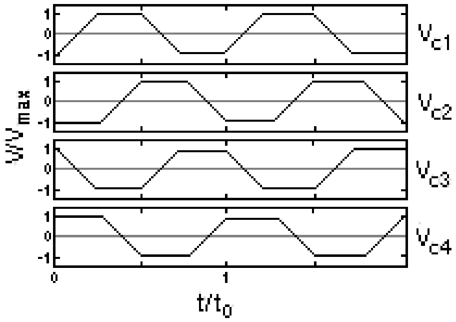

Figure 6: The voltage of four adjacent wires, each offset $\pi/2$ from its neighboring wire. [21]

The clocking scheme is four-phased, consisting of a switching phase, locked phase, switching phase, and null phase, respectively. These four phases are achieved by submitting regions of the QCA array to electric fields. The electric fields are produced by wires running under the QCA array. Each wire has a four-phase signal which corresponds to each of the four clocking phases. The clocking phases can be seen in Figure 6 for four adjacent wires buried under the QCA cells. The signals in adjacent wires have a $\pi/2$ phase shift from neighboring wires so that every fourth wire has the same signal. These wires run under the QCA array as shown in Figure 7.

In the simulator, these clocking zones are manually selected. The selection of

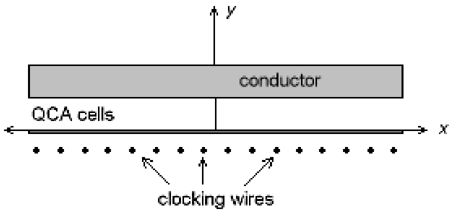

Figure 7: A schematic representation of a clocking model. [21]

zones is mainly by device. For example, each majority gate in an adder must be in a zone isolated from its inputs and outputs for proper operation. The clocking zone discussed above, though, will not have that flexability. We must determine a clocking zone that runs vertically (or horizontally) under the QCA array. The width of these zones will be determined by the wires used, the grounded conductor above the arrary and the power of the current in the wire.

Figure 7 is a schematic representation of a clocking model as proposed by Hennesy and Lent in [21]. The clocking wires located just below the QCA array alternate electric fields. The QCA array lies on the xz plane while the wires run parallel to the z axis (perpendicular to the page). The grounded conductor above the array is used to draw the the electrical fields upward in order to consume each clocking region.

The four phases of a clocking zone (i.e., a clocking zone refers to a set of cells under a common clock control) is starting in the null phase, that is, the wire embedded beneath is at low amplitude. The null phase causes each individual cell to be null. At this time null values are at steady state due to the electromagnetic field surrounding the region. As the wire's amplitude slowly changes from low to high, which is sufficiently long to facilitate electron tunneling, the clocking zone is in the switching phase. While this zone is in the switching phase, its predecessor will be locked. Thus it is the case that the current zone will react with the previous zone to obtain its values with out affecting other zones. Since the next clocking zone is in the null state, it will neither affect or be affected by the zone in the switching state. After reaching a certain high enough amplitude the wire's electromagnetic field locks the current zone to its steady state with respect to the previous zone. The next zone will begin as the current zone did in this example, moving from null state to switching state, and propagate the data through the array.

One major advantage of this clocking scheme is that it facilitates power gain in the QCA array as shown in [4,9]. In each cell interaction there is energy dissipation. The driver cell must push the other cells into their stable states and, thus, energy is lost. The energy that is dissipated must be restored at each stage to prohibit the loss of the data that is carried within the cells. Clocking the cells does just that by driving the electrons into their appropriate stable states and holding them there.

4 QCA Carry-Look-Ahead Full-Adder

A QCA-based implementation of a single carry-look-ahead full-adder is a target design in which the proposed single clock-controlled majority-gate is employed as a basic component. In order to construct the proposed single carry-look-ahead full-adder under reduced clock zones, the proposed majority gates are to be integrated along with proper clocking, wire crossings and inverter wires. In order to create a functional full adder, three inputs are needed: a, b, and C_{in}. A design of the single carry-look-ahead full-adder is shown in Figure 8 as created using QCADesigner, a design, layout and simulation tool for QCA [1,2]. As derived in [6] the formulae used in the creation of the adder are as follows:

$$C_{out} = M(A, B, Cin) \tag{1}$$

$$Sum = M(\overline{Cout}, Cin, M(a, b, \overline{Cin})) \tag{2}$$

Where M defines the three input majority function as previously discussed. This adder, as constructied in [7], uses five clocking zones and can be seen in figure 8. This adder construction obviously does not have uniform clocking zones. The clock zone placement, which is dependent on the majority gates present in the design, is not physically realizable with the previously discussed clocking scheme. This is one of the major problems that we will address in the next few sections, beginning with RoE induced faults in majority gates.

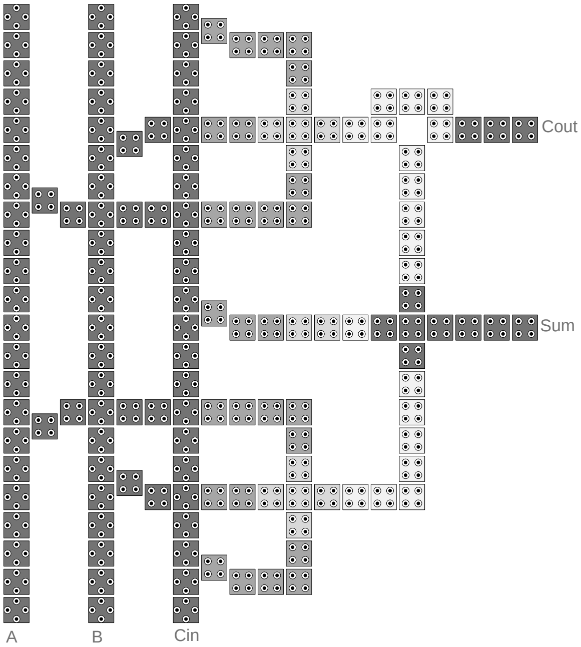

Figure 8: A single carry-look-ahead full-adder. Output of this adder can be seen in table 2

Table 2: Full-adder inputs and resulting outputs

A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

5 Radius of Effect-Induced Faults

The radius of effect of each cell can and will affect the operation of certain structures in a QCA array. In this section we will simulate and analyze a simple majority gate under a single clock control and using different areas within the radius of effect. The radius of effect-induced faults will become apparent through a few simple simulations.

The radius of effect of a cell is the radius at which it will interact with other cells. In the simulator specifications of the radius of effect is from the center of one cell to the center of another. So two inline cells will interact if

$$d = d_N = w + s \tag{3}$$

where d is the radius of effect, w is the width (and height) of the (square) cell, and s is the distance of separation of the cell. Now we will specify the distances that will be used in testing. We will test the majority gate at different areas within the radius of effect. We will assume that the cells are evenly spaced and of uniform width and height. The cells that are being used in these simulations are 20nm x 20nm with 5nm dots. The cells are spaced 5nm apart.

Equation 3 is the distance for nearest neighbor which we will denote as d_N. The radius of effect for next to nearest neighbor will simply be

$$d_{NN} = 2(d_N) = 2w + 2s \tag{4}$$

Using the pythagorean theorem we can find some of the other important distances that need to be considered in testing the area within the radius of effect. For the diagonal cell distances (see Figure 9), which we will call first diagonal, second diagonal, and third diagonal, respectively, we have the three equations

$$d_1 = \sqrt{2((s + w)^2)} \tag{5}$$

$$d_2 = \sqrt{(s + w)^2 + (2s + 2w)^2} \tag{6}$$

$$d_3 = \sqrt{2((2s + 2w)^2)} \tag{7}$$

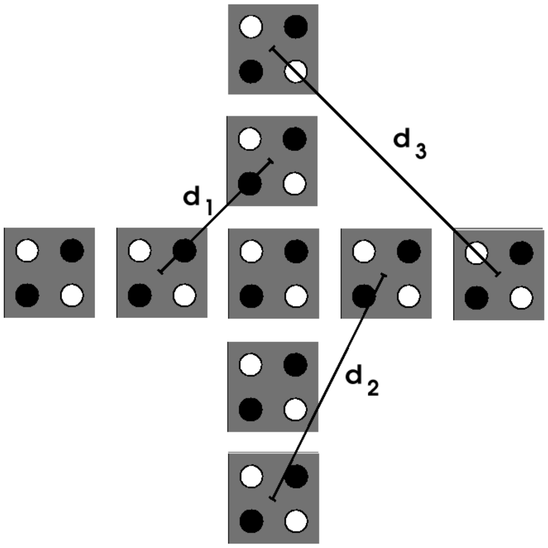

Figure 9: Area of effect distances which require consideration.

We limit the area of effect distances to d_3 due to the rapid decay of kink energy between cells as their distance of separation increases. As is stated in reference [1], it decreases inversly with the fifth power of the distance of cell separation.

Now that we have the distances of note defined, we can analyze their impact on a majority gate. We will use the coherence-vector simulation to test. We choose the coherence-vector simulation due to the fact that it is the most precise simulation available and takes into account the dissipative effects of the system. The i^{th} cell simulated in the coherence-vector simulation is a two-state cell and is defined by the Hamiltonian [1]

$$H_i = \sum_{j \epsilon S} \begin{pmatrix} -\frac{1}{2} P_j E^k_{i,j} & -\gamma_i \\ -\gamma_i & +\frac{1}{2} P_j E^k_{i,j} \end{pmatrix}. \tag{8}$$

The energy needed to tunnel between polarization states is γ. The j^{th} cells indicate those cells that are in the effective radius of the i^{th} cell; S is the effective neighborhood of cell i. $E^k_{i,j}$ is the kink energy between the i^{th} and j^{th} cells and P_j is the polarity of the j^{th} cell. The kink energy (the cost of two cells having opposite polarities) $E^k_{i,j}$ can be found by calculating from the electrostatic interaction of all the charges. For each dot in i we compute its electrostatic interaction with each dot in j by using the equation [1]

$$E^k_{i,j} = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{|r_i r_j|} \tag{9}$$

where ϵ_0 is the relative permittivity of free space and ϵ_r is the relative permittivity

Table 3: Radius of Effect Testing Results

Radius of Effect	5 Cell M-Gate	9 Cell M-Gate	13 Cell M-Gate
$d_N \leq d < d_1$	$fault - free$	$fault - free$	$fault - free$
$d_1 \leq d < d_{NN}$	$fault - free$	$fault - free$	$\overline{A}$
$d_{NN} \leq d < d_2$	$fault - free$	B	$M(\overline{A}, B, \overline{C})$
$d_2 \leq d < d_3$	$fault - free$	B	B
$d = d_3$	$fault - free$	B	B

of the system [1]. For clarity we will expand the equation for kink energy between two cells. The expanded form is

$$E^k_{i,j} = \frac{1}{4\pi\epsilon_0\epsilon_r}\sum_{i=0}^{3}\sum_{j=0}^{3}\frac{q^1_i q^2_j - q^1_i q^1_j}{|r_i - r_j|} \tag{10}$$

for $q^1_i = -0.8e - 19$ for even i and $= 0.8e - 19$ for odd i; $q^2_i = 0.8e - 19$ for even i and $= -0.8e - 19$ for odd i. The constant $0.8e - 19$ is one half of one electron volt (eV), a half charge. The term $|r_i - r_j|$ is simply the distance between dot i in cell 1 and dot j in cell 2.

The results of the testing can be seen in Table 3.

In table 3, the five cell majority gate is the basic gate that can be seen in 3, and the nine cell configuration is like that in Figure 9, with inputs from top, left, and bottom, and output to the right. The thirteen cell majority gate is expanded in the same way. The table gives the resulting outputs from the given configurations. $Fault - free$ indicates that the gate functioned properly and the output was $M(A, B, C)$. Erroneous outputs are indicated by the differing output calulated by the simulator.

Looking at the Table we can see how the radius of effect-induced fault disrupts the larger majority gates. The distance that works in all three instances is the nearest neighbor (d_N). This is obvious since the cells will only be interacting with at most four cells (the middle cell of the majority gate is within d_N of one cell above, below, to its left and to its right). For all disances the 5-cell configuration works, since this is the 'classic' setup for a majority gate the results are as expected.

The 9-cell gate obviously has more potential for more complex interaction and, therefore, more potential for erroneous interaction. As can be seen in Table 3, there are only two distances that work. When setting the distance to d_{NN} or d_2, the output for the gate is equal to the left input value. The results are therefore erroneous. There is a rather serious problem here which we will now address.

Simple wire crossings require that there be at least next to nearest neighbor interaction (see Figure 5 for an illustration). Without a radius of effect greater or equal to d_{NN} a wire crossing simply is not possible.

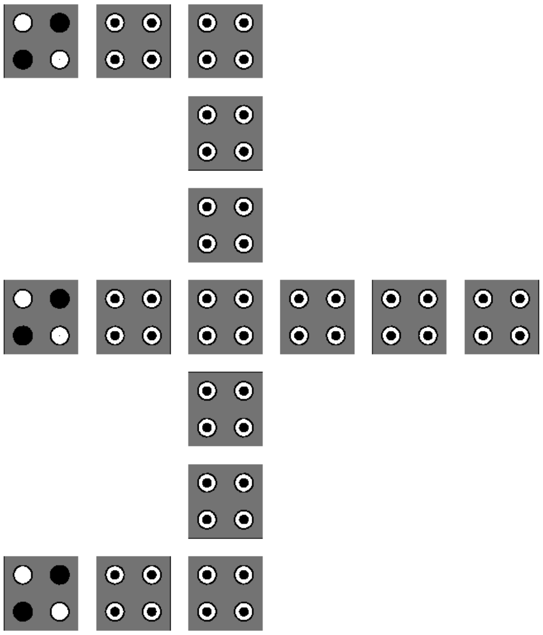

Figure 10: Left to right majority gate as used in an adder constructed of 20nm x 20nm cells with 5nm dots, spaced 5nm apart.

Now we can look back at Figure 8 to further illustrate the radius of effect problem. In the adder we see that the majority gates needed for the computation are constructed as shown in Figure 10, which will be referred to as a *left to right majority gate*. The results of a coherence-vector simulation of this configuration with a radius of effect of d_{NN} can be seen in Figure 15 as erroneous. The resulting output is as if the top input, input A were flipped; $M(\overline{A}, B, C)$. This type of majority gate is used four times in the single carry-look-ahead adder. The adder works, however, due to its clocking zones. However, we will not be able to have clocking zones that are on such a small scale in a desirable design.

The radius of effect-induced faults have now been identified and characterized through simulation. In the next section the new left to right majority gates that are tolerant to such radius of effect-induced faults will be proposed.

6 Radius of Effect-Induced Fault-Tolerant Majority Gate(s)

The projected width of a QCA cell for room temperature operation is somewhere in the 5nm realm. For cells of this size it is not likely that we will be able to have small enough clocking zones (three cell width in our adder, which will be approximately 20nm, at most) to have working majority gates. Also, in previously proposed adders the clocking zones are non-uniform in that they do not follow the constriants of the proposed clocking scheme. They do not have uniform, parallel, vertical clocking zones that are required by the use of wires running under the array. They are also very inflexible in that if clocking zones are offset by one or more cells the array will not properly function. These problems can be solved by creating a majority gate that will operate in a single clocking zone regardless of the radius of effect of the cells.

Therefore, we now have motivation to construct a majority gate that will operate correctly under a single clock control within multiple radius of effect distances. The construction is not complex, in fact it mearly involves adding (or subtracting) a number of cells to the gate in order to even out the three inputs' interactions with the device cell(s). First discussed will be the majority gate that will be used to handle a radius of effect of up to d_{NN}.

The modified majority gate can be seen in Figure 11, which is a left to right majority gate as is needed in an adder. From the simulation results in Figure 16, it can be seen that the output for a radius of effect of d_{NN} results is a correct output for the gate and also works for d_1, both with a single clock. This tolerance is facilitated by the addition of only two cells. However, this configuration does not operate correctly with a radius of effect of d_2.

For radius of effect of d_2 we need to add more cells to the configuration. The modified majority gate can be seen in Figure 12. Once again, the addition of one cell to the design creates a structure that will operate correctly with radius of effects d_{NN} and d_2. The simulation results for this configuration are exactly like those seen in Figure 16. Though this modified majority gate cannot handle radius of effect distances of d_3, majority gates that can accept greater distances can be further engineered. This section has dealt with the operation of larger cells, cells greater than 5nm width and height. Creating fault-tolerant molecular scale left-to-right majority gates has a more simple solution, which we will now discuss.

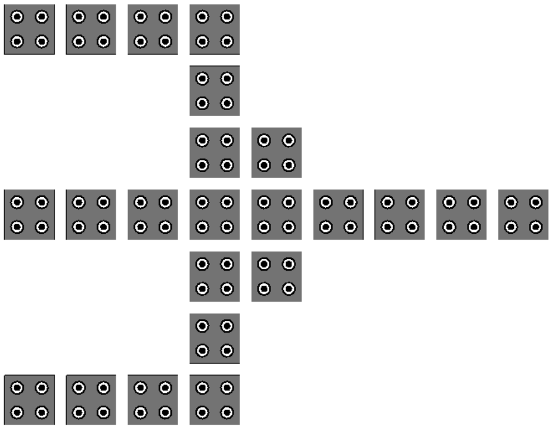

Figure 11: Modified majority gate for radius of effect-induced fault-tolerance under a single clock control for $d = d_1$ and d_{NN}.

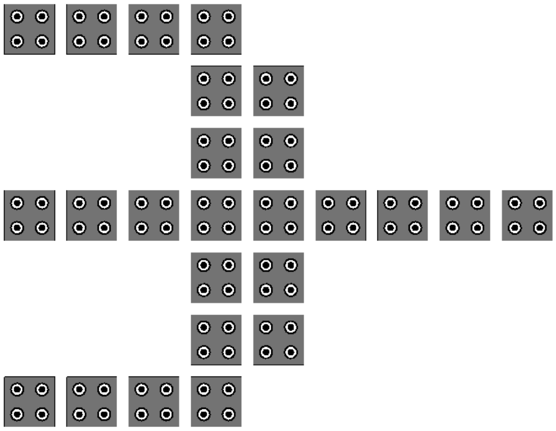

Figure 12: Modified majority gate for radius of effect-induced fault-tolerance under a single clock control for d_1, $d = d_{NN}$ and d_2.

7 Molecular QCA

Creating QCA structures that will operate at room temperature will require reducing the scale of cells to the molecular level, giving cells sizes of around 2nm [24]. Molecular cells are constructed by connecting redox sites, which can hold a charge, by ligands that allow tunneling between the sites. A simple example of such a molecule is shown in [25] (1,4-diallyl butane radical cation) and has two allyl groups which are connected by a butyl bridge which facilitates the tunneling of electrons and, therefore, the switching of the molecule between basis states. The size of this molecule is 7Å in length (0.7nm). Placing two of these molecules side by side creates a cell with a total of four allyl groups. These four allyl groups act a the dots which contain charges. The two-molecule cell is approximately 1nm by 1nm and has the two basis states ("0" and "1") that we need for a typical QCA cell. Cells of this size also have erroneous output in simulation when configured into a left-to-right majority gate.

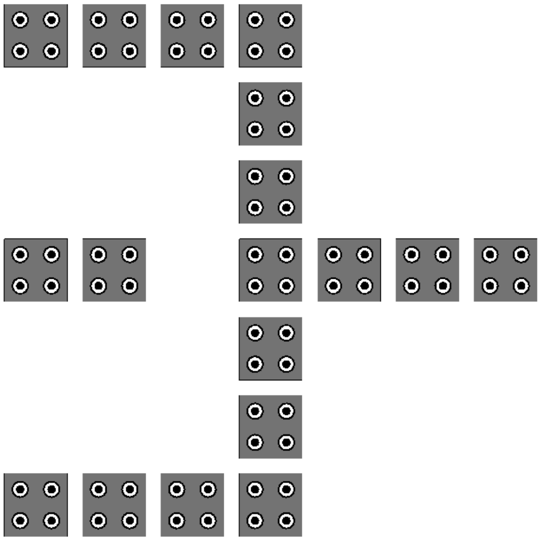

Figure 13: Functioning left-to-right majority gate for cells of size 1, 2, or 4nm.

Cells of width 1, 2 and 4nm were tested in left-to-right majority gates. The results for these configurations were the same for all three cell sizes and radius of effects d_1 to d_3. All tests resulted in an output equal to the middle input cell. This indicates that the middle input cell overpowers the other inputs at the device cell, switching the device cell to the middle input value at all times. This is not unlike the errors that occur in configurations with larger cells. To overcome this erroneous functioning we have constructed a majority gate that uses one less cell that functions correctly for all three cell sizes and for radius of effect greater than or equal to d_{NN}. The configuration is shown in figure 13.

For these simulations we used the coherence-vector simulation as with testing of larger cells. The simulations show that this configuration works only for radii of effect greater than or equal to d_{NN}. This fact is obvious since the middle input will only be able to interact with the device cell if it can interact with cells that are d_{NN} away, due to the missing cell in the middle input wire. Through simulation, we have found that this configuration will work for all radii of effect of concern, and beyond.

More accurate simulations were done for molecular implementations of quantum-dot cellular automata. The coherence-vector simulation was used to simulate the previously discussed molecular construction using 1,4-diallyl butane. It is stated in [8] that molecular implementations will have a kink energy (E_k) greater than 500 meV. With this in mind a kink energy of $E_k = 629.45$ (relative permittivity of 0.3) was chosen as an apporximate value for the molecule in question. The value was calculated using equation 10. The simulations were performed at approximate room temperature (300 K). The cell height and width was 1 nm with dot diameter and uniform cell spacing of 0.25 nm.

The simulations show that, under one clock zone, the construction shown in figure 10 is erroneous, resulting in output equal to the middle input for radii of effect greater than d_{NN}. When the majority gate shown in figure 13 is used under the same constraints, it functions for radii of effect from d_{NN} to d_3 and beyond. The results of the simulation can be seen in figure 17. This shows that the radius of effect fault-tolerant majority gate can operate at room temperature for molecular implementations of QCA.

8 Fault-Tolerant Carry-Look-Ahead Full-Adder

We have seen that the fault-tolerant left-to-right majority gates correctly function in the simulator. The last simulated molecular scale majority gate is the most exact to physical reality, so we will be using it to create a larger array to show that the gates will function within a complex QCA structure, namely a carry-look-ahead full-adder. The adder to be constructed will be a direct mapping of the one seen in figure 8. The modified adder can be seen in figure 14.

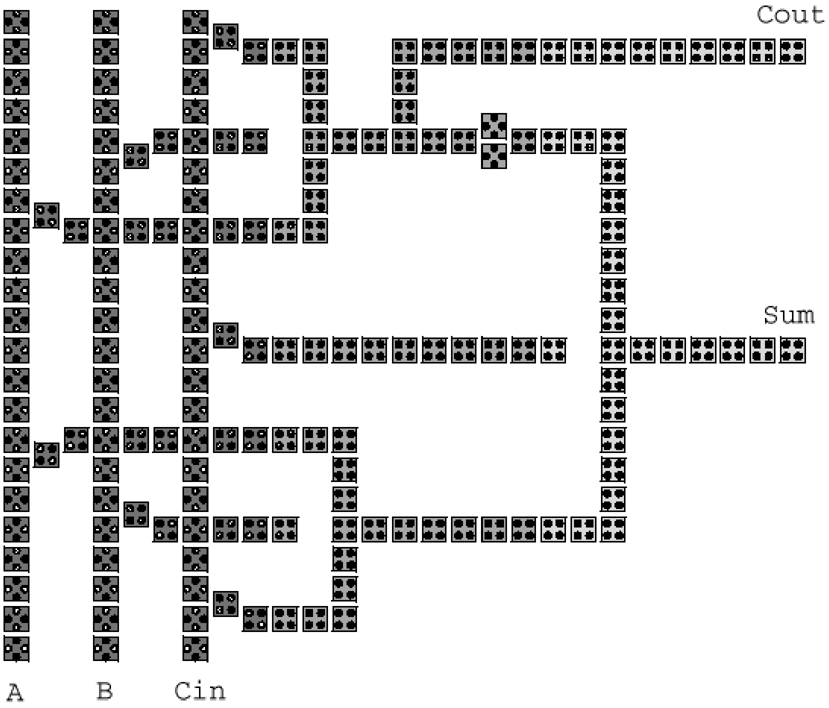

Figure 14: Functioning fault-tolerant carry-look-ahead full-adder with reduced clocking zones. Clocking zones shown by shading.

The fault-tolerant adder uses only three clocking zones to complete operation. In addition to using less clock zones, the adder has uniform, vertical zones that coincide with the clocking scheme that was presented previously. The adder was tested under the strict guidlines presented at the end of the previous section and the simulation results can be found in figure 18. The simulation results show that the adder funcions correctly.

The improved adder functions for all radii of effect of concern, with physically realistic constraints, as well as with the proposed clocking scheme. It can also be easily adjusted for multiple clocking zone sizes, but is show with a clock zone width of approximately 11nm.

9 Conclusion

It has been shown that radius of effect faults occur in the simplest of structures in quantum-dot cellular automata. Under one clock cycle majority gates will provide erroneous results and, therefore, will limit the clocking scheme when placed in larger arrays.

To counter these faults we have made minor adjustments to the majority gate. It has been shown that these changes, which are made according to the radius of effect of each individual cell, result in functioning majority gates. It has also been shown that, under simulation, the radius of effect fault-tolerance majority gates will operate for molecular implementations of QCA, which is important due to the fact that a molecular level cell will be needed to create arrays that will function at room temperature. The use of the modified gates in larger arrays of cells was also shown to be successful and benificial to the larger array. These gates will allow for fault-tolerant designs with respect to radius of effect as we all facilitate more flexable clocking zone placement and sizing. This research will aid in the creation of large QCA designs that are more physically implimentable than ever before.

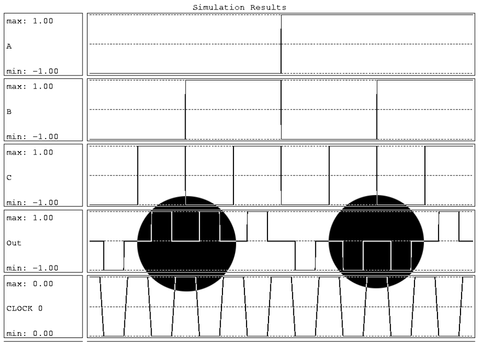

Figure 15: Simulation results for majority gate in figure 10 under one clock cycle from the simulator. Erroneous outputs are highlighted.

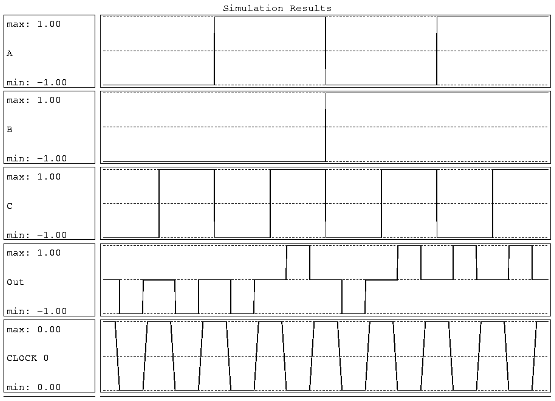

Figure 16: Simulation results for the modified majority gates in figure 11 and figure 12 from the simulator. The results show that the majority gates function properly.

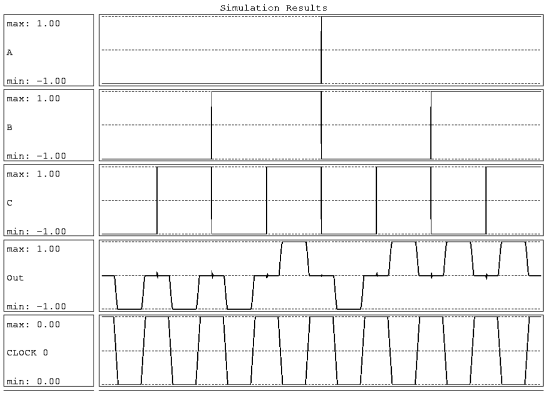

Figure 17: Simulation results for molecular QCA fault-tolerant majority gate.

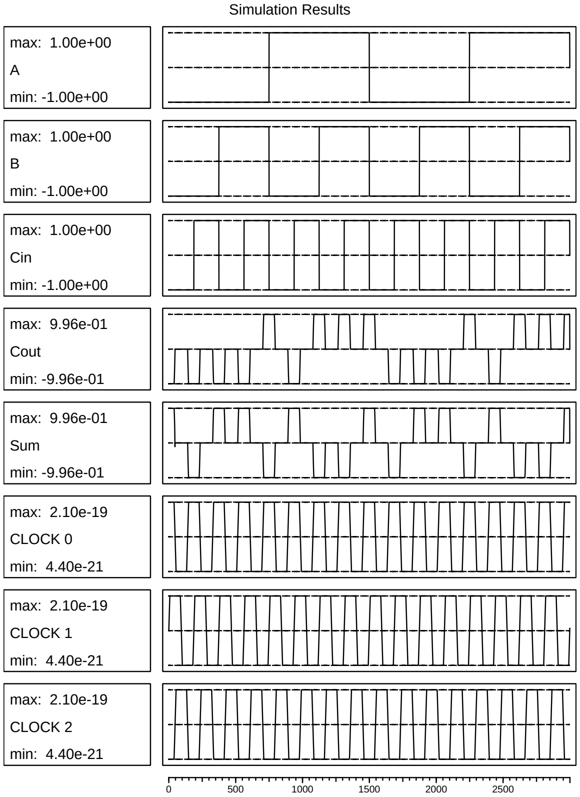

Figure 18: Simulation results for the fault-tolerant adder in figure 14.

References

[1] (2004) QCADesigner: The QCA simulator homepage. [Online]. Available: http://www.atips.ca/projects/qcadesigner

[2] (2004) QCADesigner. [Online]. Available: http://www.qcadesigner.ca

[3] A. Fijany and B. N. Toomarian, "New disign for quantum dots cellular automata to obtain fault tolerant logic gates," *J. Nanoparticle Research,* **3**, (Nov. 2001): 27–37.

[4] J. Timler and C. Lent, "Power gain and dissipation in quantum-dot cellular automata," *J. Appl. Phys.,* **91**, (Jan. 2002): 823–831.

[5] G. Toth and C. Lent, "Quantum computing with quantum-dot cellular automata," *A Physical Review,* **63**, (2001)

[6] R. Zhang and K. Walus and W. Wang and G. Jullian, "A Majority Reduction Technique for Adder Structures in Quantum-dot Cellular," *SPIE International Symposium on Optical Science and Technology,* **5559-10**, (2004)

[7] (2004) Basic Tutorial On Quantum-Dot Cellular Automata (QCA). [Online]. Available: http://www.qcadesigner.ca/tutorials/QCATutorial.html

[8] J. Timler and C. Lent, "Maxwell's demon and quantum-dot cellular automata," *J. Appl. Phys.,* **94**, (Jul. 2003): 1050–1060.

[9] C. S. Lent and B. Isaksen, "Clocked Molecular Quantum-Dot Cellular Automata," *IEEE Trans. Electron Devices,* **50**(9), (Sep. 2003): 1890–1896.

[10] A. I. Csurgay and W. Porod and C. S. Lent, "Signal Processing with Near-Neighbor-Coupled Time-Varying Quantum-Dot Arrays," *IEEE Trans. Circuits Syst. I,* **47**(8), (Aug. 2000): 1212–1223.

[11] P. D. Tougaw and C. S. Lent, "Logical devices implemented using quantum cellular automata," *J. Appl. Phys.,* **75**, (feb. 1994): 1818–1825.

[12] M. T. Niemier and P. M. Kogge, "Logic in wire: using quantum dots to implement a microprocessor," *Proc. IEEE ECS'99,* **3**, (Pafos, Cyprus, Sep. 1999): 1211–1215.

[13] M. T. Niemier, A. F. Rodrigues and P. M. Kogge, "A Potentially Implementable FPGA for Quantum Dot Cellular Automata," *Proc. NSC,* **1**, (Boston, MS, Feb. 2002)

[14] A. Vetteth, K. Walus, V. S. Dimitrov and G. A. Jullien, “Quantum-dot Cellular Automata Carry-Look-Ahead Adder and Barrel Shifter,” *Proc. IEEE ETTC,* (Dallas, TX, Sep. 2002)

[15] C. D. Armstrong and W. M. Humphreys, “The Development of Design Tools for Fault Tolerant Quantum Dot Cellular Automata Based Logic,” *2nd International Workshop on Quantum Dots for Quantum Computing and Classical Size Effect Circuits,* (2003)

[16] C. D. Armstrong, W. M. Humphreys and A. Fijany, “The Design of Fault Tolerant Quantum Dot Cellular Automata Based Logic,” *11th NASA Symposium on VLSI Design,* (2003)

[17] M. Governale, M. Macucci, G. Iannaccone and C. Ungarelli, “Modeling and manufacturability assessment of bistable quantum-dot cells,” *J. Appl. Phys.,* **85**, (Mar. 1999): 2962–2971.

[18] C. G. Smith, “Computation Without Current,” *Science,* **284**(5412), (Apr. 1999)

[19] A. O. Orlov and I. Amlani, “Realization of a Functional Cell for Quantum-dot Cellular Automata,” *Science,* **277**(5328), (Aug. 1997)

[20] P. D. Tougaw and C. S. Lent, “Dynamic behavior of quantum cellular automata,” *J. Appl. Phys.,* **80**, (Oct. 1996): 4722–4736.

[21] K. Hennessy and C. S. Lent, “Clocking of molecular quantum-dot cellular automata,” *J. Vac. Sci. Technol.,* **19**, (Sep. 2001): 1752–1755.

[22] I. Amlani, A. O. Orlov, G. L. Snider and C. S. Lent, “Demonstration of a six-dot quantum cellular automata system,” *Appl. Phys. Letters,* **72**, (Apr. 1998): 2179–2181.

[23] S. E. Frost, A. F. Rodrigues, A. W. Janiszewski, R. T. Rausch and P. M. Kogge, “Memory in Motion: A Study of Storage Structures in QCA,” *Proc. NSC,* **1**, (Boston, MS, Feb. 2002)

[24] M. Lieberman, S. Chellamma, B. Varughese, Y. Wang, C. Lent, G. H. Bernstein, G. Snider, and F. C. Peiris, “Quantum-Dot Cellular Automata at a Molecular Scale,” *Ann. N.Y. Acad. Sci,* **960**, (2002): 225–239.

[25] C. Lent, B. Isaksen, and M. Lieberman, “Molecular Quantum-Dot Cellular Automata,” *J. Am. Chem. Soc.,* **125**, (2003): 1056–1063.

VITA

Zachary Daniel Patitz

Candidate for the Degree of

Master of Science

Thesis: FAULT TOLERANT QUANTUM-DOT CELLULAR AUTOMATA MAJORITY GATE DESIGN

Major Field: Computer Science

Biographical:

Personal Data: Born in Des Moines, Iowa, On June 6, 1979, the son of Daniel and Patricia Patitz.

Education: Graduated from Charles Page High School, Sand Springs, Oklahoma in May 1997; received Bachelor of Science degree in Computer Science from Oklahoma State University, Stillwater, Oklahoma in May 2002. Completed the requirements for the Master of Science degree in Computer Science at Oklahoma State University (May, 2006).

Experience: Summer internship at SupportSoft, Redwood City, California from 2000 to 2004; Teaching assistant for Oklahoma State University, Computer Science Department, Stillwater, Oklahoma 2001 to present.