Strategy and Prototype Tool for Doing Fault Modeling in a Nano-technology

Timothy J. Dysart and Peter M. Kogge
Dept. of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
Email: {tdysart,kogge}@nd.edu

***Abstract*— Quantum-dot cellular automata (QCA) has been proposed as a replacement for CMOS circuits. The major difference between QCA and CMOS is that electronic charge, not current, is the information carrier. A complete set of logic gates has been created and some have been experimentally tested with metal-dots acting as quantum dots. Molecular implementations are currently being examined. This work examines the possible defects that may occur in the fabrication of both types of QCA systems. Fault models for these defects are developed, and a prototype tool with a strategy for fault modeling is outlined.**

I. Introduction

As nano-technology devices mature, the engineering issues involved with these technologies demand attention. One of these issues that must be addressed is determining when a nano-technology system will fail. This is a vital question to answer as nano-systems move beyond proof-of-concept experiments. This issue needs to be addressed because these nano-systems are inherently more susceptible to defects due to their small size and fabrication techniques, such as building them with self-assembling molecules. As the knowledge base of determining device and system failures increases, building fault tolerant systems must be considered. However, the knowledge base of fault mechanisms and methods of modeling them need to be developed first. This work focuses on developing an initial strategy and a prototype tool for doing fault modeling in a specific nano-technology, namely quantum-dot cellular automata (QCA).

In QCA, four dots occupy the corners of a square cell with potential barriers between each dot's two nearest neighbors. Molecular QCA uses redox sites within a molecule as dots and a bridging ligand as a junction between them [1]. Two extra electrons are introduced (or are available within the molecule(s)) into the cell, and by raising and lowering the potential barriers with the clock, discussed shortly, an electron can localize on a dot. From the Columbic interactions between these electrons, they will tend to occupy antipodal sites in the square cell. Due to this interaction, two different polarizations are available, *P = 1* and *P = -1* as shown in Fig. 1(a). Respectively, these polarizations provide a logical one and a logical zero, thus maintaining the binary computing paradigm. As Fig. 1(b) shows, several of these cells can then be placed side by side to form a wire. Logic values then pass from cell to cell due to the Columbic interactions.

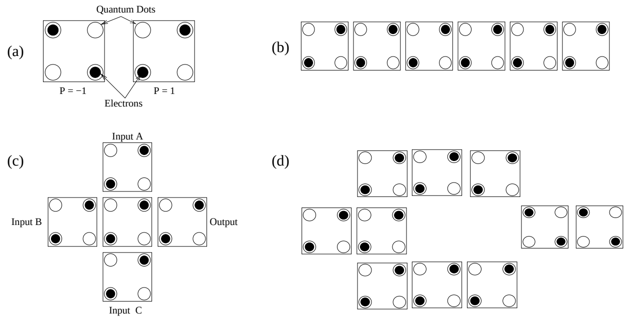

Fig. 1. (a) Shows the two polarization values for QCA cells. (b) A QCA wire. (c) A QCA majority gate, which is the fundamental logic gate implementing AB + BC + AC. (d) A QCA inverter.

By placing groups of cells together in different configurations, logic gates can be constructed. The majority gate in Fig. 1(c) is the fundamental gate used in QCA, and implements the voting logic function AB + BC + AC. Holding one of the inputs to zero forms an AND gate, while holding an input to one forms an OR gate for the remaining two inputs. Since cells that are diagonal from one another tend to hold the opposite polarization, inverters can also be constructed as is shown in Fig. 1(d). Having an inverter, AND gates, and OR gates, a functionally complete logic set is available for general computation. Several systems have been designed in QCA including the data flow for a simple processor and a memory structure [2], [3].

The clock used in QCA consists of four phases: hold, release, relax, and switch. These phases correlate to the action of the potential barriers within the cell. During the hold phase, the barriers are kept high, thus the electrons are kept highly localized on two dots and give the cell a set polarization. This provides a driver cell for a neighbor. In the release phase, the barriers are slowly reduced, which delocalizes the electrons, and the cell loses a distinct polarization value. The barriers in the relax phase are minimized, giving full freedom to the electrons and preventing these cells from influencing neighbors. During the switch phase, the barriers are slowly risen while the cells are driven by neighbors in the hold stage. By the end of this stage, the cells are distinctly polarized. QCA systems are then divided into clocking zones, where a clocking zone is a region where all cells are in one of the four clock phases. By placing four (or more) clocking zones together, where each one starts in the switch, relax, release, and hold phases respectively (repeating the pattern as necessary), values can transfer from one end of a system to another. The reader is directed to [4]–[6] for more information regarding the clocking of QCA systems.

Although QCA proof-of-concept devices have been shown to function properly [7], [8], several researchers have pointed out that improvements need to be made in terms of manufacturability and defect tolerance [9], [10]. This work is focused on the latter of these since previous work has shown how sensitive QCA systems are to various defects. Theoretical work by Fijany and Toomarian found that moving a cell from its intended location by only a half-cell in distance can cause the system to fail [9]. In the case of a molecular QCA implementation, this would require placement accuracies of approximately one nanometer. Governale *et al.* have done some work showing semi-conductor QCA to be sensitive to dot placement and size, but this has yet to be explored in the metal-dot implementation covered in this work [10]. It is expected that metal-dot QCA cells will be substantially more robust to inaccuracies in dot placement and size. Different dot sizes should not be an issue with molecular QCA due to the nature of molecular structures. These results demonstrate knowledge of how some individual defects affect a system, but there is a need for understanding how multiple defects affect a system to aid in the development fault-tolerant architectures.

This paper will outline a strategy for fault modeling by first determining what the various manufacturing defects are in Section II and then developing methods of modeling these defects in Section III. Both metal-dot and molecular implementations will be examined. Semi-conductor implementations could be considered, but metal-dot systems require fewer processing steps using similar fabrication processes. Section IV outlines a strategy and a tool that will be developed to systematically study how multiple defects can influence a QCA system. Conclusions and future work will be discussed in Section V

II. Defects

In this section, fabrication defects will be analyzed for both metal dot and molecular implementations of QCA. The proof-of-concept QCA cells were implemented using metal dots instead of semi-conductors [7]. Even for these simple two to six dot systems, circuitry was needed to balance the fabrication defects. The possible defects of these systems will be analyzed in the first subsection. For molecular QCA cells, proof-of-concept devices have yet to be built, but candidate molecules have been identified [11]. Further research and development of molecular QCA systems will show what the exact defects of these systems are. This, and future, work should aid in the development of these molecular proof-of-concept systems by providing the first step in developing fault tolerant architectures.

One possible error in QCA systems that will not be examined in this section is that of having an energy large enough, due to thermodynamic effects, to cause a cell to switch incorrectly. The energy needed for a QCA cell to switch should be many times larger than the thermodynamic energy provided by the environment. This error is ignored since the thermodynamic energy is proportional to $k_B T$ where k_B is Boltzmann's constant and T is the temperature. It is assumed that the temperature of a QCA system will be held in a region where these thermodynamic errors do not occur.

Also, clocking defects will not be examined in this work since clock signals are implemented either in extra metal dots [4] or wires in the substrate below the QCA cells [5]. In the former implementation, the defects are the same as for the dots containing information. For the latter, the clocking structure should be considered separately.

A. Metal Dot Implementation

Both initial and more recent QCA systems have applied corrective voltages on each dot due to the fabrication differences inherent in each dot and junction [7], [8]. The main problem faced by these systems was parasitic cross-talk capacitance [7]. As this is a noise issue and not a particular defect it is unable to be modeled. Since metal-dot cells use similar processing steps as CMOS circuits, the possible defects will be similar as well. As shown in [7] and [8], the major processing steps are electron-beam lithography (EBL), metal deposition (by shadow evaporation), and oxidation. Each of these steps can create defects, and additional ones can occur from particles in the clean room.

Errors in the EBL processing step are similar to those in current lithography techniques and will tend to either leave dots and/or junctions in the wrong spot or they will be sized incorrectly because the targeted region was either under or over exposed. Making the tunnel junctions too large will have a similar effect as depositing too much metal in the junction, which allows electrons to pass through the junction easily. This could theoretically cause improper switching, but Columbic forces from other dot pairs should force a dot pair with this defect to work properly. Similarly, having a metal oxide layer that is too thick is similar to having a junction with too little metal. An electron trying to pass through a junction like this could be fixed in place due to the high energy needed to pass through it. Another possible defect in the metal deposition stage would be having dots of the wrong size or shape.

B. Molecular Implementation

Self-assembling molecules hold great promise for the fabrication of nano-technology systems. The process proposed to attach candidate molecules to a substrate first grows a SiO_2 layer, uses EBL to create trenches where QCA molecules are attached, and then soaks the wafer in a bath containing QCA molecules [11], [12]. The possible defects associated with EBL are the same as those listed above, thus there could be extra or missing molecules due to an inaccurate trench. Since it is unknown how precisely the molecules will align within and attach to the substrate, defects in all three spatial directions will need to be considered.

Even if the molecules align precisely as desired in the trenches, it appears that they will still have surface attachment angles that vary from molecule to molecule [11]. Due this inherent factor, it is possible that each pair of dots could be in a slightly different than expected location. Thus, having architectures tolerant of this difference will be necessary. This inherent factor of molecular QCA needs to be examined as a defect due to the instability of a QCA system from moving dots as was outlined in [10]. Another factor that needs to be considered is that of stray charges being introduced into the system. These charges will probably interact with the QCA molecules, and will tend to force cells into a fixed polarization.

III. Fault Models

A large majority of the defects discussed in the previous section can be modeled in one of three ways. The first is to remove a cell from the system, which could occur with a clean room defect, incorrect EBL, or a molecule not attaching in a specific location. The second method is by rotating or moving a cell, or at least a pair of dots (in the case of molecules). Again, this could be the case from EBL defects or molecules not attaching precisely where expected. Lastly, modifying dots will also need to be undertaken by either moving them or changing their size. Moving a single dot would be most likely in metal-dot implementations, as the molecular implementations will have dots moving in pairs. A change in the size of the diameter of the dot is not likely to occur in molecular QCA, but in metal-dot QCA applying too much or too little metal could cause the dot to be the wrong size.

There are several other defects that cannot be modeled using the three previous models. One possible defect is having a bad tunnel junction, thus fixing a cell to a specific polarization. This could also occur by having a stray charge in the system. However, both of these defects can be modeled with a fixed cell. A fixed cell may cause an error in a single-cell wide wire system, but should not cause one in a fault-tolerant architecture. The tool presented in the next section will aid in deciding how detrimental this type of defect is in a tolerant architecture.

One proposed defect tolerant architecture for QCA is that of wider gates and wires. Instead of a wire that is a single cell wide as was shown in Fig. 1(c), a wire is n-cells wide. Fijany and Toomarian showed that increasing the input lines of a block majority gate from one cell to three cells wide could turn a non-functioning gate into a functioning one [9]. The block majority gate they used was 11 x 8 cells, and had several missing and rotated cells. This demonstrates that wider wires and gates have improved defect tolerance, and Fig. 2 shows an example of a three cell wide wire.

IV. Strategy and Prototype Tool

The prototype tool discussed in this section must be able to implement the strategy outlined here. The first part of the strategy is to have a complete set of fault models, such as those listed in Sec. III. Determining where the system fails for each of these individual models is necessary. It is imperative that each cell of a small system be tested since defects in specific cells may be more catastrophic than defects in other cells. For example, a defect in the middle cell of a majority gate may cause the output to be undetermined, but a missing cell in a wire may not cause the system to fail. After determining how a single fault affects a system, higher quantities of each fault need to be tested. For example, it will be instructive to know what happens when two, three, or four cells are rotated. To save computational time, it will be useful to know if there is a distance where faults are far enough apart so that their effects are not compounded, and can be treated as individual defects instead. This knowledge will be accumulated to avoid unnecessary testing and aid in the future development of yield models as QCA systems are manufactured in quantity.

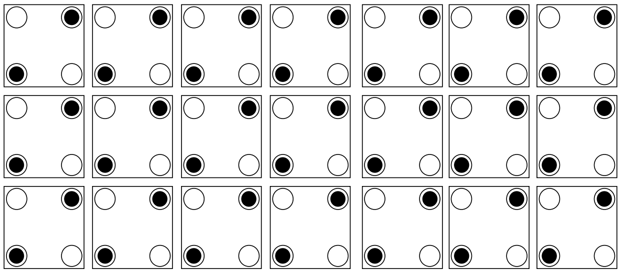

Fig. 2. A three cell wide wire

After understanding how one or more faults of a specific type cause a system to fail, the next step is understanding how groups of different types of faults create problems. For example, does rotating a cell and moving or changing the size of a dot in that, or a neighboring, cell have a different effect on the system than just rotating the cell. Fault types should be tested for all combinations, except for those that have been found not to change a system. As before, finding methods of limiting the computational time is extremely important.

The prototype tool to implement this modeling has only one major constraint on it, and that is it must be able to work with existing QCA design tools. The current basis for these design tools is QCADesigner, which is a CAD tool capable of layout and simulation [13]. A file format, based on the XML standard, is under development so that the fault modeling tool can be either integrated with or stand-alone from QCADesigner. This file format will allow for the separation of the architecture from the technology used to build the system and allow for hierarchical design, which is similar to the goals and purpose of the CIF (Caltech Intermediate Format) file format. Additionally, building a format based on the CIF model provides an understanding that various tools will need to operate on the same circuits, thus a common format between all tools is essential.

Since all information regarding the cell and dot locations will be available to the tool, fault models can be injected into the system. For example, one fault model was a moved cell. By changing where the center of the cell is located within the tool, the result of moving it can be examined. Since the

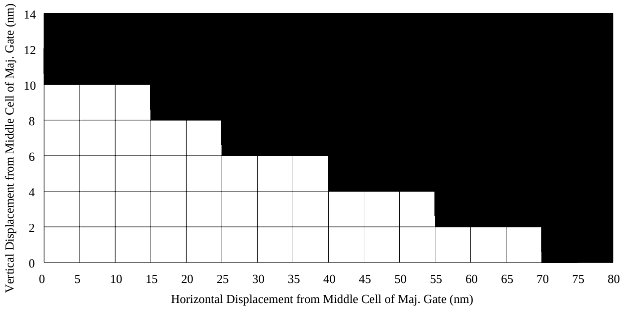

Fig. 3. A schmoo plot showing pass (white) vs. fail (black) for the horizontal and vertical displacements of the horizontal input to a majority gate, such as Input B in Fig. 1(c).

tool will operate by iteratively changing a pair/set of fault models (i.e. moving a cell by the same quantity each iteration), simulating the system with a computationally efficient physical approximation, and determining if the output values are correct, schmoo plots can be created to determine when a system fails. Using results from Fijany and Toomarian's work [9] at the endpoints and linearly interpolating between them, the schmoo plot in Fig. 3 can be created. This plot shows whether a majority gate functions or not based on the horizontal and vertical displacements of a specific input cell (Input B of Fig. 1(c)). These plots will provide a clear indication of where a system fails. To reduce overall computational time, these plots will first be generated on a coarse grain level. After determining what range of values for a specific model cause a system to fail, that range can be tested at a finer grain to provide a more complete picture of system failures. For example, a cell could be initially rotated by five degrees, and then on a one degree level once the first failure range is known.

V. Future Work and Conclusion

As previous research has shown, fabrication defects for QCA systems, particularly for self-assembling molecules, are likely to occur and as such, fault-tolerant architectures will need to be developed. However, without knowing what the defects are and having methods of modeling them, it will be difficult to develop a fault-tolerant architecture. In this work, the likely QCA fabrication defects have been examined and fault models for these defects have been developed. Additionally, a strategy for understanding how groups of defects effect the system and a prototype tool for doing fault modeling have been developed. Work continues on molecular self-assembling monolayers and exactly how the molecules are attached to a substrate. Finalizing the strategy and developing the modeling tool are ongoing. The results of this work will be used in the testing and development of fault tolerant architectures which are of the utmost importance in the development of practical nano-technology systems.

Acknowledgment

This work was supported by The Jet Propulsion Laboratory. The authors would like to thank M. Niemier, M. Lieberman, and C. Lent for their helpful discussions.

References

[1] C. S. Lent, B. Isaksen, and M. Lieberman, "Molecular quantum-dot cellular automata," *J. Am. Chem. Soc.*, vol. 125, no. 4, pp. 1056–1063, 2003.

[2] M. T. Niemier, "Designing digital systems in quantum cellular automata," Master's thesis, U. of Notre Dame, Notre Dame, IN, USA, 2000.

[3] S. E. Frost *et al.*, "Memory in motion: A study of storage structures in qca," in *1st Workshop on Non-Silicon Computing (NSC-1) held in conjunction with High Performance Computing Architecture 8*, Boston, MS, USA, Feb. 2002.

[4] G. Toth and C. S. Lent, "Quasiadiabatic switching for metal-island quantum-dot cellular automata," *J. Appl. Phys.*, vol. 85, no. 5, pp. 2977–2984, Mar. 1999.

[5] K. Hennessy and C. S. Lent, "Clocking of molecular quantum-dot cellular automata," *J. Vac. Sci. Technol. B*, vol. 19, no. 5, pp. 1752–1755, Sept./Oct. 2001.

[6] M. T. Niemier and P. M. Kogge, "Exploring and exploiting wire-level pipelining in emerging technologies," in *28th Annual International Symposium on Computer Architecture*, Goteborg, Sweeden, July 2001, pp. 166–177.

[7] G. L. Snider *et al.*, "A functional cell for quantum-dot cellular automata," *Solid State Electronics*, vol. 42, no. 7-8, pp. 1355–1359, 1998.

[8] I. Amlani *et al.*, "Expiremental demonstration of a leadless quantum-dot cellular automata cell," *Appl. Phys. Lett.*, vol. 77, no. 5, pp. 738–740, July 2000.

[9] A. Fijany and B. N. Toomarian, "New design for quantum dots cellular automata to obtain fault tolerant logic gates," *J. Nanoparticle Research*, vol. 3, pp. 27–37, 2001.

[10] M. Governale *et al.*, "Modeling and manufacturability assesment of bistable quantum-dot cells," *J. Appl. Phys.*, vol. 85, no. 5, pp. 2962–2971, Mar. 1999.

[11] M. Lieberman *et al.*, "Quantum-dot cellular automata at a molecular scale," *Ann. N.Y. Acad. Sci.*, vol. 960, pp. 225–239, 2002.

[12] Q. Hang, Y. Wang, M. Lieberman, and G. H. Bernstein, "Molecular patterning through high-resolution polymethylmethacrylate masks," *Appl. Phys. Lett.*, vol. 80, no. 22, pp. 4220–4222, June 2002.

[13] QCADesigner home page. [Online]. Available: http://www.atips.ca/projects/QCADesigner