

Using Circuits and Systems-Level Research to Drive Nanotechnology

Michael T. Niemier¹, Ramprasad Ravichandran¹, and Peter M. Kogge²

Georgia Institute of Technology¹, University of Notre Dame²

mniemier@cc.gatech.edu, raam@cc.gatech.edu, kogge@wizard.cse.nd.edu

Abstract

This paper details nano-scale devices being researched by physical scientists to build computational systems. It also reviews some existing system design work that uses the devices to be discussed. It concludes with a discussion of how the authors believe system-level research can best be used to positively affect actual device development. This work has led to a more thorough design methodology that will address whether or not computationally interesting and buildable circuits are possible with the Quantum-dot Cellular Automata (QCA), while also providing significant wins over end-of-the-roadmap CMOS.

1. Introduction

The main goal of this paper is to show how research related to nano-scale devices can occur in the realm of computer science and circuit/system design, as opposed to having it reside exclusively in the physical science domain. If done smartly (and ideally in collaboration with physical scientists), systems-level research has the potential to answer the question: “Can a certain nano-scale device perform a computationally interesting and necessary task better than end of the roadmap silicon?” This is the eventual end result that is desired. Interestingly, even if the answer to the above question is “No” – using what is currently seen as implementable as bounds on design – the results from a systems-level study would still form a roadmap for physical scientists. It would detail desirable and needed device characteristics that would have to eventually be built to form computationally interesting systems capable of providing wins over silicon. Thus, the focus of this paper will be to detail how systems-level research can answer the above question, and/or create the aforementioned roadmap. It will first detail nano-scale devices being considered for computational systems (Sec. 2). In Sec. 3, we will discuss circuit/system design efforts for some of the devices in Sec. 2. Sec. 4 will

detail some of our circuit/system design work for QCA – a detailed example of how circuit and system designers could work with physical scientists to help advance the state of the art, and possibly a device’s time to realization. We will conclude in Sec. 5 with a discussion of how initial circuit/systems-level studies have evolved into a design methodology whose purpose is to provide a definitive answer to the question posed above – and will also spur new research and projects.

2. Nano-scale devices

We begin by discussing some of the nano-scale devices being considered to form computational systems. Single Electron Transistors (SETs), carbon nanotube arrays, pure quantum computing, DNA based computation, and the Quantum-dot Cellular Automata will all be discussed. The interested reader may also want to see for [15, 24] for a discussion of quantum transistors, resonant tunneling diodes, and computing with chemical molecules.

2.1. Single-Electron Transistors

One of the building blocks of the Single-Electron Transistor (SET) is the quantum dot – generally defined as a semiconductor or metal solid state structure that can confine a small number of electrons into a small space. Confinement of electrons is achieved by placing some insulating material(s) around a central, well conducting region. The energy needed to place an electron on, or remove an electron from, a quantum dot depends on the size of the dot and how many electrons are already present on it (ideally, for room temperature operation 1-3 nm dots are required). One way to place electrons on, or remove electrons from an island is to add an electron source that is separated from the island via a thin oxide through which electrons can tunnel. A gate over the island can “control” this source, and change the energy state of the dot to determine when electrons are allowed to tunnel. Specifically, applying a voltage at the gate will polarize the island. As the voltage is increased from

0, an electron in the source will initially not have enough energy to charge the island. This *Coulomb blockade* is the basis for all SETs [15], [22].

If a larger voltage is applied, the polarization of the island will increase until it becomes the same as that of one electronic charge. At this point, an electron can tunnel from the source electrode. This feature could be used to turn a device into a transistor. Possibilities include replacing the channel of an FET by an island and separating it from the source and drain by tunneling barriers. If the source-drain voltage were raised, no current would flow until a threshold voltage (high enough to overcome Coulomb blockade) was reached.

Still, implementation issues loom with SETs. Notably, because of the small transconductance, it will take a significant amount of time to charge interconnect. Also, background charge could be a significant problem – namely because we are moving single electrons and even a single charged impurity in a dielectric could render a device non-functional. Finally, fabricating a single nano-meter scale dot and lining it up with a transistor channel is non-trivial [15].

2.2. Carbon Nanotube Arrays

Carbon nanotubes are long, thin cylinders of carbon, first discovered in 1991 by S. Iijima. They can have a very high length-to-width aspect ratio – only a few nanometers in diameter yet up to a millimeter long. Nanotubes can be thought of as a sheet of graphite (a hexagonal lattice of carbon) rolled into a cylinder, and have a wide range of electronic, thermal, and structural properties.

Given that there is much ongoing work involving carbon nanotubes, we only seek to provide a “computational flavor” here. Early work with carbon nanotubes shows that in the presence of an electric field, nanotubes can move current, emitting electrons from their tips. While many electrically conductive materials can provide this functionality, nanotubes can do so at extremely low voltages, thus making them ideal for building small and efficient electron emitters [29].

“Computationally”, it may be possible to generate transistor-like functionality with carbon nanotubes, with “devices” being only a few nanometers in diameter. Nanotube “transistors” can further be cascaded into logic gates which can in turn selectively route electrical signals -- namely the 1s and 0s needed for computation. Additionally, nanotubes offer the promise of building, small, efficient and non-volatile memory structures that could lead to terabits of memory [29], [30].

Like most (if not all) emergent devices, carbon nanotube dependent products are by no means close to being a commodity. Fabrication issues abound and

synthesis methods are still primitive. However, nanotubes offer promise because they could form conventional transistors and serve as an interconnection mechanism. Additionally, there appear to be many roles that carbon nanotubes could play in or with more conventional, MOS based system, allowing for a more gradual integration and time-to-market.

2.3. Pure Quantum Computing

Quantum computing offers the potential to scale computation exponentially with data size. Classically, data is most often represented by a discrete 1 or a discrete 0. However, in quantum computation, the basic building block is a quantum bit (called a *qubit*) which can represent both a 1 and a 0 simultaneously using physical properties such as nuclear spin. Specifically, n qubits can represent 2^n states with a qubit's state determined by probability amplitudes. These probability amplitudes can destructively interfere with each other, and only turn into actual probabilities when a value is “read” [31-34].

The last sentence in the above paragraph introduces a fundamental problem associated with quantum computation. Namely, that it is not possible to observe intermediate steps of computation, and a value can only be “read” after the quantum computation has completed – and then can only a random value can be read from the vector. Another fundamental problem associated with quantum computation centers around the fact that qubits will quickly lose their quantum properties in the presence of a constant amount of noise per qubit (this deterioration occurs at an exponential rate and is called *decoherence*). Quantum computation can tolerate a finite amount of decoherence, but building such a noise-tolerant system remains an engineering problem [33].

At present, small quantum devices (consisting of 5-7 qubits) have been built in laboratories, and the possibility exists for 100-bit devices [33], [35]. Improvements in quantum error correction have allowed for large scale designs and established a threshold theorem: scalable computers can be built from “faulty” components provided that the error probability for each quantum operation is less than 10^{-4} . Nevertheless, the error correction cost in quantum computing still remains very high, and in fact, handling errors and performing error correction is probably the most important task when considering quantum architectures.

2.4. DNA-based Computation

One of the most fundamental and well-known methods of computation is the Turing Machine. Simply, a Turing machine consists of a pair of tapes and some mechanism of finite control which could move along the

input tape, read data, and simultaneously moves along the output tape while writing data [36]. It is possible to duplicate some of this functionality with DNA sequences, and use them to perform computation and solve algorithmically interesting problems.

As an example, an experiment conducted by Adelman used combinations of DNA sequences to solve a 4-city “traveling salesman” Hamiltonian Path problem (specific sequences encoded city and connection route names). Overall, 7 total days of “computation” were required to find the solution. The actual “computation” takes place with extraordinary energy efficiency. Ideally, one Joule is sufficient for approximately 2×10^{19} ligation operations, while existing supercomputer applications can execute at most 10^9 operations per Joule [37]. While this, combined with the potential for extremely dense information storage, can certainly be seen as a positive, it can be outweighed by the fact that this method of computation is realistically only useful for a few classes of problems.

2.5. QCA

The QCA concept represents information by using binary numbers, but replaces a current switch with a cell having a bi-stable charge configuration. A QCA device can consist of 2 or 4 quantum dots and either 1 or 2 excess electrons respectively. One configuration of charge represents a binary ‘1’, the other a binary ‘0’ (Fig. 1a), but no current flows into or out of the cell [1, 2]. In the transistor paradigm, the current from one device charges the gate of the next device and turns the device on or off. In the QCA paradigm, the field from the charge configuration of one device alters the charge configuration of the next device. This basic device-device interaction is sufficient to allow for the computation of any Boolean function (see [2-4]), and also forms interconnect. If a clocking potential is added which modulates the energy barrier between charge configurations, general purpose computing becomes possible with low power dissipation. Systems could conceivably be built from the devices in Fig. 1b-e.

Four major “building blocks” are discussed below: (1) molecular QCA devices, (2) DNA-based substrates to which molecules will attach, (3) a silicon-based clock structure, and (4) a means for integrating the QCA logic with the silicon clock structure (liftoff). By analyzing the interactions of these four parts, a design methodology (to be developed in Sec. 4-5) will tell us whether or not we have potential wins over silicon systems with equivalent functionality – and ideal role for circuit and system designers.

QCA Molecules: In contrast to metal-dot QCA, the small size of molecules (1-5 nm) allows for large

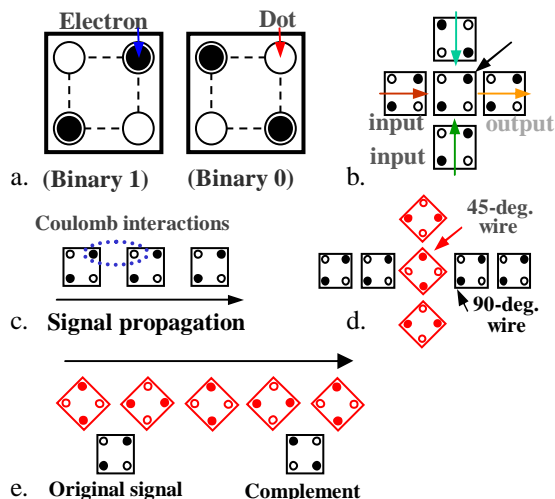


Fig. 1a: (a.) basic QCA device schematic. (b.) Majority gates: Arrangements of cells that implement the logic equation $AB+BC+AC$. Computation occurs by driving the device cell to its lowest energy state (electrostatic repulsion at a minimum), i.e. when it assumes the polarization of the majority of the 3 input cells [2]. (c.) Wires: A binary signal propagates from point A to B because of electrostatic interactions between adjacent cells. (d.) Wire crossings: QCA wires with different orientations can cross in the plane without the destruction of either value on either wire. (e.) Rippers: A binary on a 45-degree wire will alternate between a 1 and 0. By placing a 90-degree cell between 2 45-degree cells, both the original signal value and its complement can be obtained without an explicit inverter circuit. As majority gates can be reduced to an AND or OR, QCA’s logic set is functionally complete.

Coulomb energies and room temperature operation [5]. Also, power dissipation from QCA switching would be low enough that high-density molecular logic circuits and memory are feasible. Projections indicate that 10^{11} QCA devices in a cm^2 would dissipate 100 W of power when switching (with switching speeds ranging from 10^{-12} to 10^{-13} s per device [11, 19]). The role of a “dot” will be played by reduction-oxidation (redox) sites within a molecule. Molecules with at least two redox centers are desired, allowing for 2, 3, and 4 dot cells [5, 20-21].

Molecular QCA and their interactions with a clock are explained using 3-dot cells. In Fig. 2, a QCA molecule forms a ‘v’-shape, and charge can be localized on any one of the three dots at the “vertices” of the ‘v’. If charge is on one of the top two dots, the cell will encode a binary 1 or 0. Whether or not charge is in the top two dots (*active state*) or the lower dot (*null state*) can be determined by an electric field (*clock*) that will

raise or lower the potential of the central dot relative to the top two dots [5]. Binary 1s and 0s are physically represented by the dipole moments of QCA molecules. Besides creating the electric field required for state transitions, the clock also helps to increase the tolerance of individual devices to E_{kink} [8]. (E_{kink} is the amount of external energy that will excite a cell into a mistake state, or create a “kink” in a transmission – i.e. we would get a binary 1 instead of a 0.)

Substrates: A pitch matching problem exists between the substrates to which molecular QCA devices could attach, and the devices themselves [5, 9, 10]. Current optical or x-ray/e-beam lithography cannot create detailed patterns to which devices could attach to form computationally interesting, custom circuits [5].

One mechanism that might allow for selective cell placement and patterning is DNA tiles (branched DNA strands that self-assemble in a regular pattern). DNA tiles can form rigid, stable junctions with well-defined shapes, and can further self-assemble into more complex patterns [27]. Each tile could also contain several points to which a QCA cell could attach. Lieberman et. al. have developed a DNA raft built from four individual tiles, and are working to develop bigger rafts. Each individual tile could hold 8 QCA cells [16, 18]. Each portion of a raft has a different DNA sequence. Consequently, molecular recognition could be used to differentiate locations on the raft to which individual molecules could attach – forming a “circuit board” for molecular components.

Liftoff: Molecular liftoff is a technique for deposition of molecular films of molecules. DNA rafts could be attached to silicon wafers using a thick poly-adhesion layer (probably in EBL etched tracks) – which would be most useful if silicon is used to form the clock circuitry [12-14].

The Clock: A clocking mechanism allowing a QCA device to transition from a monostable, null state, to a bistable, active state, and then back to a monostable, null state is also required. The four phases of a clock signal could take the form of time-varying, repetitious voltages applied to silicon wires embedded underneath a substrate to which QCA cells were attached (see Fig. 2). The charge and discharge of the clocking wires will move the area of activity across the molecular layer of QCA cells and occurs at the “leading edge” of the applied electric field. Computation would move across the circuit in a continuous “wave” [6, 7].

3. Architects and Nano-scale Devices

It is important to remember that all of the devices mentioned above, are being researched because they could potentially form the components of a

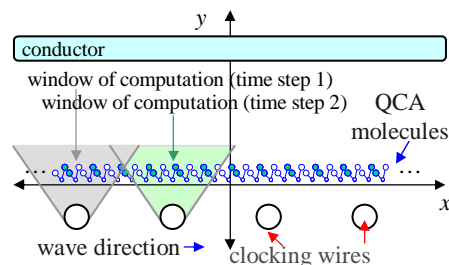


Fig 2: Possible clock implementation.

computational system. However, until recently, most if not all research conducted with any nano-scale device has been limited to the realm of device physics, or has not moved beyond the simplest or basic circuits. Only recently, with some specific nano-scale devices showing significant promise and progress, have researchers - specifically computer architects - begun to study what computational systems of such devices (again, the desired end result) might look like. Some efforts related to the devices discussed earlier are discussed below. The interested reader may also want to see [23, 44] for other studies.

3.1. Carbon Nanotubes

Andre Dehon of Caltech has begun to consider a basic architecture for molecular electronics that assumes a core of carbon nanotubes and silicon nanowires. The goal of this work is to provide a mechanism for some universal logic functionality while still considering issues such as logic and signal restoration at the molecular level. One proposed scheme involves arranging molecular scale wires into interconnected, crossed arrays with non-volatile switching devices at their cross points, with arrays functioning as programmable logic arrays (PLAs) and programmable interconnect. It is envisioned that nano-scale FETs would provide the signal restoration and programming support needed for such a system. This would ultimately result in a programmable logic device that could be configured to compute any logical function and would perform computation at the nano-scale [38].

3.2. Nano-fabrics

Seth Goldstein of Carnegie Mellon University has begun to consider what architectures for Chemically Assembled Electronic Nanotechnology (CAEN) should entail. Specifically, the focus is on reconfigurable computation and defect tolerant systems. CAEN is described as a generic form of electronic nanotechnology that will use self-alignment to construct electronic circuits via nanometer-scale devices, takes advantage of

quantum-mechanical effects, and offers gate densities of up to 10^{10} gate-equivalents per square cm^2 [39-40].

The fundamental “strategy” behind CAEN is to substitute manufacturing precision and fabrication processes (which are expensive and only growing in cost) with compile time. Namely, while CAEN will most likely *not* be used to construct complex and “custom” circuits and systems, it will still provide a great deal of computational power via reconfigurable computing, defect tolerance, architectural abstractions, and compiler technology; and it will do so with high-density and low power substrates that have lower fabrication costs than CMOS. However, it must be emphasized that defect tolerance will have to be a serious consideration. For this reason CAEN focuses on regular structures and developing a methodology to configure and diagnose various computational blocks with an eye toward implementing some desired circuit functionality, while simultaneously routing around defective blocks [39], [40].

The core of the CAEN architecture is called a *NanoFabric*, which is in turn composed of a two-dimensional mesh of *NanoBlocks*. *NanoBlocks* are designed as programmable logic blocks that can implement a three-bit input to a three-bit output Boolean function (they will also generate the function's complement). The blocks can also route signals, and are organized into clusters where each cluster is connected to its nearest four neighbors. Longer wires can span clusters to form “long-range” interconnection schemes and also route signals between clusters [39], [40].

It is worth briefly looking at the “bigger picture” for NanoFabrics. A CMOS implementation is proposed to handle power, ground, clock and configuration wires, I/O mechanisms, and basic control. NanoBlocks would then be constructed on top of the CMOS, with long-line interconnect mechanisms being handled by chemically self-assembled components. The fabrics would then be used in implementable devices either as factory programmed circuits and systems or as reconfigurable computing devices. At the system-level, a significant problem and area of work will be developing compilation software and techniques given the complexity of mapping a circuit design to a fabric as there are potentially 10^{11} configurable switches. Traditional place-and-route will not scale to devices with billions of wires and devices [39], [40].

3.3. Quantum Architectures

While the tasks involved with actually processing the massive overhead required for quantum error correction remain daunting, the proof of the threshold theorem indicates that engineering, not physics could prevent a

quantum computer from actually working. With this in mind, researchers have begun to study what an overall architecture for a quantum computer might look like [31-33].

At present, it appears that quantum computers may best be suited for specific applications. Examples include prime factorizations, Shor's algorithm -- which shows that an n bit integer can be factored in $O(n^3)$ time, and Grover's algorithm -- which can search an unordered n -element list in \sqrt{n} queries [42]. As stated above, while a long way from system-scale quantum computers, current research devoted to physical devices, quantum algorithms, and quantum error correction has provided motivation for beginning to look at the possibilities of quantum computer architectures [33].

Oskin et. al. propose expressing quantum algorithms through a model that performs quantum operations on quantum data but is under the control of a classical computer. Specifically, quantum “programs” would combine quantum unitary transforms (which form quantum “gates”), quantum measurements, classical computation, and classical control-flow decisions into a single instruction stream. A compiler would then read the mixed quantum and classical instructions and further break down complex quantum operations into a small set of universal operators.

This work has been extended by Mark Oskin, Frederic Chong, Isaac Chuang, and John Kubiatowicz [9]. At a high-level, one very important conclusion of their work is that as nanotechnologies move closer to reality, architectural studies become more pressing. At a lower level, Oskin et. al. have chosen to study quantum wires which will be required to move quantum data. In particular, they compare moving information via teleportation to the traditional quantum “swap” operation, concluding that an advanced architecture that uses a teleportation channel overcomes a basic limit with regard to latency and bandwidth associated with the swapping channel method. Like our work, studies were done in a very device independent manner leaving open the possibility that this work could apply to any implementation method for quantum computation. Finally, the authors identified a “pitch matching problem” (we have identified a similar problem in our work with QCA). Namely, classical CMOS logic is needed to control quantum logic; and the CMOS logic exists at a larger scale. The authors assert that a more sparse connection architecture of coarse-grained computational elements is more realistic than the more common “sea of gates” model often put forth.

4. Research Directions – QCA as context

The ability to accurately specify, describe, and verify designs that are more complex than a handful of devices will be crucial as the underlying technology in QCA (or any other nanotechnology) advances. In MOS, the Mead-Conway concept of “design rules” abstracted underlying physics to a point where *engineers* could more easily generate designs from components provided by physical scientists, and computer-aided design (CAD) tools could in turn analyze and verify them. For MOS circuits, if a circuit’s layout conforms to certain geometries (allowable widths, separations, overlaps, etc.), a designer can be assured that a particular layout will conform to the resolution of a particular fabrication process and work as intended post fab. Values used to specify these parameters usually are a function of a given process, take into account lithography limitations, and add a margin for error. QCA design rules are based on potential failure points in the envisioned fabrication process (self-assembly of molecular QCA cells), and how they are reflected in circuits as designed by an engineer (see Fig. 3 for potential defects).

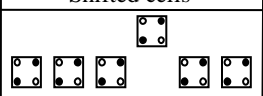
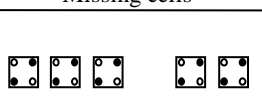
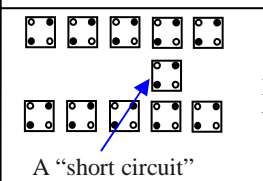
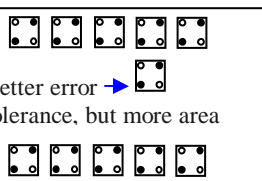
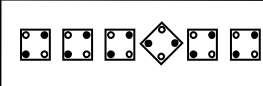

Shifted cells	Missing cells	
 <p>Will cause a logical error – unless an even # of errors occur</p>	 <p>Less probability of a successful switch? Weaker interactions?</p>	
Where the cell shouldn't be...		
 <p>A “short circuit”</p>		 <p>Better error → tolerance, but more area</p>
Rotated cells	Vertical	
 <p>If perfect rotation, no interference; otherwise, “when does it break?”</p>	 <p>What if the substrate isn't level? Is this a problem?</p>	

Fig. 3: Potential defects.

Analyzing the impact of these defects in the context of systems is the goal of a design methodology, and should answer the question of whether or not computationally interesting and buildable QCA circuits and systems are possible. We believe that a similar approach for other devices (i.e. as seen in Sec. 3) would be useful.

A framework for what a design methodology must qualify and quantify is presented below. These are essentially preliminary design rules for QCA.

Cell Spacing: Our first design rule (1A-B, Fig. 4), considers spacing between two molecular QCA cells. Specifically, what is the maximum allowed and minimum required distance between two cells such that they will still transmit data? In Fig. 4, these distances are labeled x_{\max} and x_{\min} and specific values would be governed by substrates to which QCA cells can attach, E_{kink} , and dipole interactions between cells (energy of interaction proportional to $1/d^3$). Also, x_{\min} will provide an initial upper bound on maximum device densities.

Wires: Currently, we envision four design rules when considering wire (see Fig. 4). Rule 2A defines two wire lengths: the maximum physical length (n_{normal}) of a wire, and the maximum length of a wire that is being clocked (n_{clocked}) specified as a number of QCA cells. n_{normal} is simply a function of the substrates to which molecular QCA cells will attach. Using numbers from our discussion of QCA background, an upper bound for the number of cells in a clocked wire is $\exp(E_k \times 28.7)$ assuming a 300K operating temperature. Rule 2B considers the maximum length of a wire with disorder. With regard to design rules, a disordered wire's length ($n_{\text{disordered}}$) would seemingly be dependent on whether or not the greatest kink energy in the system is less than the worst kink energy for cell-to-cell interactions on the wire. Both off-center and rotated cells must be considered. Rule 2C considers cross talk between two parallel wires. Namely, parallel wires will have to be some finite distance apart to ensure that “short circuits” do not occur. ϵ_y is defined to be the maximum error in placement for a cell in a wire (or in other words, the maximum amount of off-centeredness possible). Assuming ϵ_y s that would bring two molecular QCA cells in two parallel wires as close together as possible, d_y is defined as the minimum distance between these two cells to ensure no cross talk or short circuits. However, to incorporate this error, the wires themselves must be separated by a distance of d_{\min} . Rule 2D considers what happens if a cell is missing from a wire. However, this error could be just defined by design rule 1A (the maximum allowable spacing between QCA cells such that a value is still transmitted successfully).

Crossovers: Rule 3 (Fig. 4) considers a 45-degree/90-degree wire crossover. Most of the interactions required to ensure that data is transmitted successfully on both wires are actually defined by previous rules. The distance between the two 45-degree cells is governed by Rule 1A. Errors due to off-centeredness and rotation are defined by rule 2B. Also, we must consider the interaction between the 45-degree

and 90-degree cells as there should be no interference on either value on either wire. 45-degree/90-degree cell interactions can actually be considered by using the fact that kink energy is proportional to $(1/r^5)\cos(2(\theta_1+\theta_2))$. If one cell is rotated 45-degrees and another is not rotated at all ($\theta_2 = 0$), we will take the $\cos(90)$ which is 0. The two cells do not interact. However, any deviation in rotation could disturb a wire crossing.

Majority Gate: When considering a majority gate (rule 4, Fig. 4), it too is essentially governed by previous rules – particularly rule 1A which specifies a maximum distance between cells (x_{max} and y_{max} in Figure 12), and rule 2B which considers error due to off-centeredness and rotation. The designer must ensure that given the sum of all of the “errors” between input cells and the device cell, the device cell still functions as intended.

Rippers: Interactions between 45-degree cells and a 90-degree cell designed to function as a ripper are slightly less defined. Nevertheless, some components of this configuration can be characterized by previous rules. Referring to Fig. 4, the interactions between just the two 45-degree cells could be defined by rules 1A, 1B, and 2B. However, distances d_{min} and d_{max} , as well as angle θ_3 , have also been labeled. Specifically, we will want to determine what the minimum and maximum distances between the 90-degree cell and the two 45-degree cells are allowed to be. Also, we will want to consider how tolerant this circuit is to some rotation in the 90-degree ripper cell (specified by angle θ_3).

The Clock: Rule 6 (not pictured) addresses the fact that the clock is required to produce an electric field of a certain magnitude to ensure that molecules switch between active and null states. This magnitude is a function of the QCA molecule. However, given that $E = V/d$, the circuit designer must consider the distance d between silicon clocking wires and some QCA substrate. Using the required magnitude E and a distance d , we must calculate a required voltage amplitude and design our silicon circuitry accordingly.

This work forms the foundation for a circuit design methodology. We will compile error margins and rates, incorporate this information into circuit schematics, and re-evaluate. This process is summarized in a design methodology. The *circuit design* community should be involved with work related to all aspects of it.

5. A Design Methodology (DM)

The first step (1) of the DM simply involves gathering basic information – a molecule’s inherent tolerance to kink energy, the electric field strength required to turn it on and off, etc. The next step (2) involves laying out cells (using implementable

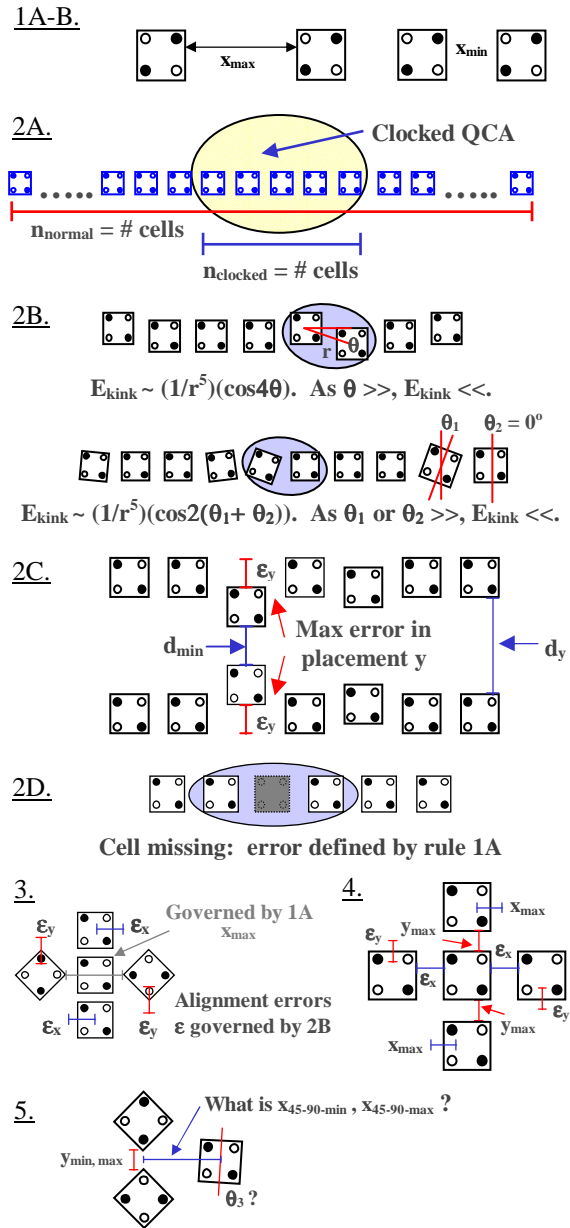


Fig. 4: Design rule schema for (1) cell spacing, (2) wires, (3) crossovers, (4) majority gates, (5) rippers.

constructs) to provide the desired logical output. After simulating for logical correctness (3), we will then introduce defects into our design (4) consistent with statistics provided from self-assembly experiments [16]. We will then re-simulate the design for logical correctness (5), and address any needs for a more robust circuit, redundancy to ensure functionality, etc. (6) [17]. The required design constructs from (6) essentially form micron rules, and will be a function of the yield and area desired from the self-assembly manufacturing process. The next step involves calculating the number of cells allowed in a window of computation (7) – too many cells

that are turned on and switching simultaneously can also include errors/bit flips [2]. This will affect how our clock structure is laid out in silicon. However, before moving on to the design of a clock structure, we must first look at the expected environmental quality of our operating environment (i.e. sources of E_{kink}) and compare it to the tolerance of our design (8). If potential kink energy in the environment is greater than our design's tolerance to kink energy, we must redesign to make our circuit more robust (revisit (2), (6)). Next (9), we will design an adiabatic clock structure to provide the required electric field/clock. This silicon design process will be constrained by (1) and (7) as well as lithographic micron rules. If such a clock structure cannot be built (i.e. because it dissipates too much power, violates (7), etc.) we may need to return to (9), (6), or even (2). If the designed clock structure is feasible, we can move on (this decision is (10)). Finally, we need to ensure that all cells in a critical path of a clock window have time to switch before the window "passes by" (11). If this condition is met, we are finished (12); if not, we may have to revisit (9), (6), or (2). The output of this DM should tell us if interesting circuits are buildable with QCA – or at the very least will tell physical scientists what physical constructs are necessary for them.

6. References

- [1] P.D. Tougaw, C.S. Lent, "Logical Devices Implemented Using Quantum Cellular Automata", *J. of App. Phys.*, Vol. 75, p.1818, 1994.
- [2] C.S. Lent, P.D. Tougaw, "A Device Architecture for Computing with Quantum Dots", *Proc. of the IEEE*, 85, p.541, 1997.
- [3] "The Effects of a New Technology on the Design, Organization, and Architectures of Computing Systems," M.T. Niemier's Ph.D. Dissertation.
- [4] M.T. Niemier, P.M. Kogge, "Exploring and exploiting wire-level pipelining in emerging technologies", Göteborg, Sweden, *Proc. 28th ISCA*, p.166-177, 2001.
- [5] M. Lieberman, S. Chellamma, B. Varughese, Y. Wang, C. Lent, G.H. Bernstein, G. Snider, F.C. Peiris, "Quantum-dot cellular automata at a molecular scale", *An. of the NY Acad. of Sci.*, 960 (Apr 2002), p.225-39.
- [6] R.V. Kumamuru, J. Timler, G. Toth, C.S. Lent, R. Ramasubramaniam, A. Orlov, G.H. Bernstein, "Power gain in a quantum-dot cellular automata latch," *Ap. Phys. Let.*, 81(2002), p.1332-1334.
- [7] K. Hennessy, C.S. Lent, "Clocking of molecular QCA" *J. of Vac. Sci. and Tech. B*, 19,5(Sept-Oct 2001), p.1752-1755.
- [8] C.S. Lent, B. Isaksen, M. Lieberman, "Molecular Quantum-dot Cellular Automata", *J. Am. Chem. Soc.*, 125, (2003), p.1056-1063.
- [9] M. Oskin, F.T. Chong, I. Chaung, J. Kubiawicz, "Building Quantum Wires: The Long and Short of it," *Proc. of 30th ISCA* p.374-85, 2003.
- [10] A. DeHon, "Array-based Architectures for FET-based Nano-scale Electronics", *IEEE Nano.*, 2(1), p.23-32, March 2003.
- [11] J. Timler, C.S. Lent, *J. Appl. Physics*, 91(2), 15 January 2002.
- [12] Q. Hang, Y. Wang, M. Lieberman, G. Bernstein, "Molecular patterning through high-resolution olymethylmethacrylate," *A.P.L.*, 80, p.4220-4222, 2002.
- [13] Q. Hang, Y. Wang, M. Lieberman, G. Bernstein, "A liftoff technique for molecular nanopatterning," *J. Nanosci. Nanotech*, 3, p.309-12, 2003.
- [14] X. Wang, W. Hu, G. Bernstein, R. Rajogopal, G. Snider, M. Lieberman, "Formation, Characterization and Sub-50 nm Patterning of Organosilane Monolayers with Embedded Disulfide Bonds," *Langmuir*, 19(23): p.9748-9758, 2003.
- [15] Linda Geppert, "Quantum Transistors: toward nanoelectronics", *IEEE Spectrum*, Sept., p.46-51, 2000
- [16] "Toward molecular circuit boards: Circuit design, synthesis, and defects in self-assembling DNA structures," submitted to NSF by M. Niemier and M. Lieberman (February, 2004).
- [17] "Tools for the design and simulation of clocked molecular quantum-dot cellular automata circuits," Enrique Pacis Blair, M.S. Thesis.
- [18] Conversations with Marya Lieberman.
- [19] Conversations with Craig Lent.
- [20] J. Jiao, G.J. Long, F. Grandjean, A.M. Beatty, T.P. Fehlner, "Building Blocks for the Molecular Expression of Quantum Cellular Automata. Isolation and Characterization of a Covalently Bonded Square Array of Two Ferrocenium and Two Ferrocene Complexes," *J. Am. Chem. Soc.*; 2003; 125(25); p.7522-7523.
- [21] Z. Li and T.P. Fehlner, "Molecular QCA Cells. 2. Characterization of an Unsymmetrical Dinuclear Mixed-Valence Complex Bound to a Au Surface by an Organic Linker," *Inorg. Chem*, 2003; 42(18); p.5715-21.
- [22] R.H. Chen, A.N. Korotov, K.K. Likharev, "Single electron transistor logic", *Appl. Phys. Lett.*, 68, p.1954, 1996
- [23] J.M. Tour, W.L. Van Zandt, C.P. Husband, S.M. Husband, L.S. Wilson, P.D. Franzon, D.P. Nackashi, "Nanocell Logic Gates for Molecular Computing", *IEEE Nano.*, 1(2), p.101-9, 2002
- [24] M.A. Reed, J.M. Tour, "Computing with Molecules", *Sci. Am.*, June, p.86-93, 2000
- [25] F.A. Bettelheim, J. March, "Introduction to General, Organic, and Biochemistry", Saunders College Publishing, New York, 1988
- [26] C.K. Mathews, K.E. van Holde, K.G. Ahren, "Biochemistry", Addison Wesley Longman, San Francisco, 2000
- [27] Fu, T-J, Seeman, N. C., "DNA double crossover structures," *Biochemistry*, 32, 3211-3220, (1993).
- [28] J.M. Rabaey, "Digital Integrated Circuits: A Design Perspective", Prentice Hall Electronics, New Jersey, 1996
- [29] D. Rotman, "The Nanotube Computer", *MIT's Tech. Rev.*, March, p.37-45, 2002
- [30] Y. Cui, C. Leiber, "Functional nanoscale electronic devices assembled using silicon nanowire building blocks", *Science*, 291, p.851-53, 2001
- [31] S. Swanson, M. Oskin, "Towards a Universal Building Block of Molecular and Silicon Computation", *Proc. of 1st Workshop on Non-Silicon Computation (NSC-1) (@HPCA)*, February 3, 2002
- [32] D. Copey, M. Oskin, F.T. Chong, I. Chaung, K. Abdel-Ghaffar, "Memory Hierarchies for Quantum Data", *Proc. of the 1st Workshop on Non-Silicon Computation (NSC-1) (@HPCA)*, February 3, 2002
- [33] M. Oskin, F.T. Chong, I.L. Chaung, "A Practical Architecture for Reliable Quantum Computers", *IEEE Computer*, January, p.79-87, 2002
- [34] C.P. Williams, S.H. Clearwater, "Explorations in Quantum Computing", Springer-Verlag, New York, 1998
- [35] M. Steffen, L.M.K. Vandersypen, I.L. Chaung, "Toward Quantum Computation: A Five-Qubit Quantum Processor", *IEEE Micro*, March/April, p.24-34, 2001
- [36] E. F. Moore, "A simplified universal Turing machine", Proceedings of the 1952 ACM national meeting (Toronto), 1952, p.50-54.
- [37] L.M. Adleman, "Computing with DNA", *Scientific American*, August, p.54-61, 1998
- [38] A. DeHon, "Array-based Architecture for Molecular Electronics", *Proceedings of the 1st Workshop on Non-Silicon Computation (NSC-1) (@HPCA)*, February 3, 2002
- [39] S.C. Goldstein and M. Buidu, "NanoFabrics: Spatial Computing Using Molecular Electronics", *Proc. 28th ISCA*, 2001, p. 178-191
- [40] M. Mishra, S.C. Goldstein, "Scalable Defect Tolerance for Molecular Electronics", *Proceedings of the 1st Workshop on Non-Silicon Computation (NSC-1) (@HPCA)*, February 3, 2002
- [42] M. Steffen, L.M.K. Vandersypen, I.L. Chaung, "Toward Quantum Computation: A Five-Qubit Quantum Processor", *IEEE Micro*, March/April, p.24-34, 2001
- [43] AJ KleinOowski and David J. Lilja, "The NanoBox Project: Exploring Fabrics of Self-Correcting Logic Blocks for High Defect Rate Molecular Device Technologies", Presented at ISVLSI, February, 2004.