The "4-Diamond Circuit" – A Minimally Complex Nano-scale Computational Building Block in QCA

Michael T. Niemier[1] and Peter M. Kogge[2]
Georgia Institute of Technology[1], University of Notre Dame[2]
mniemier@cc.gatech.edu[1], kogge@wizard.cse.nd.edu[2]

Abstract

One of the key problems in bringing nanotechnology to commercial reality is in finding design methodologies that permit such devices to be combined into computationally useful circuits with favorable density gains over CMOS, but with minimum assembly complexity. This paper develops one such structure for one of the more rapidly maturing nanotechnologies termed QCA. The "4-Diamond" Structure is shown to have interconnect and logical completeness properties similar to a classical FPGA cell, but which if implemented with molecular QCAs on a tiled DNA substrate, has the potential for self-assembly into large structures. When applied to a sample design problem, it has density at least equal to what might be expected from end of the roadmap silicon, but with the potential for significant further improvements.

1. Introduction

One approach to nano-scale computation is the quantum-dot cellular automata (QCA) concept that uses a binary representation of information, but replaces a current switch with a cell having a bi-stable charge configuration. A QCA device usually consists 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', but no current flows into or out of the cell. In the transistor paradigm, the current from one device charges a gate on the next device, the interconnect between them, and thus 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. If a clocking potential is added which modulates the energy barrier between charge configurations, general purpose computing becomes possible with very low power dissipation.

QCA has moved beyond the realm of theory, and experimental work generally falls under one of two different categories – work with prototype metal dot systems and work with molecules targeted for larger-scale QCA implementations. Basic QCA devices, a majority gate, a clocked cell, a shift register, and a wire have all been constructed in metal. A molecular implementation of QCA could offer room temperature operation and lower heat dissipation such that high-density molecular logic circuits and memory are feasible. Experimentally, scientists are working to construct 2-dot and 4-dot QCA molecules, and have extensively simulated 3-dot QCA molecules – as well as their interactions with an electric field clocking structure [2]. Researchers are also addressing means for attachment. Thus, at present, there is much ongoing research concerning nano-scale *devices* geared for computation; but most work is exactly that – *device* work. By in large, only the simplest circuits and systems comprised of such devices (which are the desired end result), have been proposed, let alone simulated and built. Furthermore, virtually all existing circuits and systems for any emergent device have been proposed by the device engineers themselves whose background lies largely in the physical sciences, not engineering.

One of the most fundamental problems in QCA fabrication is deterministically placing individual devices in order to garner specific functionality. As this may persist into the foreseeable future, we seek to determine if we can somehow create determinism "post-fabrication" via QCAs clock. This could facilitate modifiable, computationally interesting circuits independent of the self-assembly process. Ideas, results, and conclusions will be presented and discussed in the context of a QCA Field Programmable Gate Array (FPGA). FPGAs are seen as more feasible by physical scientists as they should offer regular and replicable structures that will be easier to build. System designers may view them as being more computationally feasible than the few logic gates or adders that have been proposed for most nano-scale devices envisioned for computation. This paper will present what is perhaps the simplest possible QCA FPGA design strategy. However, it addresses logical completeness and connectivity. Area numbers for this design are comparable with end of the roadmap CMOS

2. Related Work

Regular array structures are a natural target for nanotechnology design architects, and have received

Proceedings of the IEEE Computer Society Annual Symposium on VLSI Emerging Trends in VLSI Systems Design (ISVLSI'04)
0-7695-2097-9/04 $20.00 © 2004 IEEE

considerable recent attention. Beckett describes a programmable logic cell using a stacked RTD and PROM bit, but has not addressed the interconnectivity needed for large circuits or timing issues [19]. DeHon has studied systems of OR/NOR arrays where the arrays themselves are built from arrays of carbon nanotubes, but interconnect is left to the "micro-scale" [10]. Goldstein & Budiu have introduced a similar idea of "nanofabrics" where compilation of a program is done "spatially" over a large set of arrays [1]. In CMOS, the TERAMAC [16] demonstrated the potential for reconfigurability of FPGA-like structures where fault densities are large, as they are liable to be in any early generation nanotechnology. In terms of QCA circuits, our own earlier work [15] investigated a spectrum of basic routing elements for FPGA like circuits, but did not consider the next level of complexity in terms of timing and arbitrary interconnect.

3. Preliminaries

This section will review some of QCA's basic device physics, how logical devices are formed, identify potential sources for error, and explain how QCAs clock structure works. It will provide context for the rest of this paper.

3.1. QCA Devices

A high-level diagram of a "candidate" four-dot metal QCA cell appears in Fig. 1a. Characteristics related to metal dot QCA cells were essentially discussed in the introduction of this paper. While experiments with metal-dot QCA formed much of the initial experimental work with the technology, molecular QCA is seen as a more natural implementation mechanism. As this paper has a decidedly implementation-related slant, metal dot work will only be referenced here (see [3], [4], [5]). With molecular QCA cells, binary information is encoded in molecular dipole moments [5]. In contrast to metal-dot cells, the small size of molecules (on the order of 1-5 *nm*) means that Coulomb energies are much larger, so room temperature operation is possible [6]. In addition, the power requirements and heat dissipation of QCA are low enough that high-density molecular logic circuits and memory are feasible. Information about specific molecular QCA implementations is readily available in literature (see [6], [7], [8]), with 2-"dot", 3-"dot", and 4-"dot" implementations all under investigation. Binary information is moved and stored via dipole moments and their interactions. A device schematic is shown in Fig. 1b. Finally, while molecules are seen as a more natural implementation for QCA, experiments continue in both veins of research. Metal-dot QCA experiments have been used to prototype molecular QCA devices, and ideas transfer between implementations. Given this, QCA's logic functionality will be explained in terms of "generic" 4-dot cells.

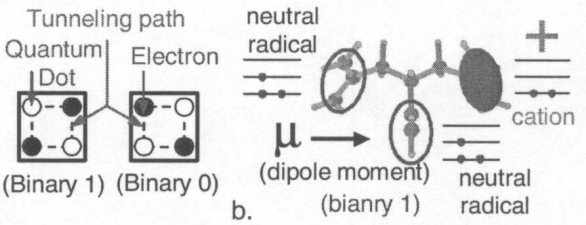

Figure 1: Metal-dot (a) and molecular (b) QCA cells.

3.2. QCA Logical Elements

A 4-dot molecular QCA cell could simply be formed by two adjacent 3-dot or 2-dot cells, but are also being engineered as explicit molecules [11]. 4-dot cells are ideal because of symmetry. Binary information is stored and moved with quadropole moments. The fundamental QCA logical gate is a three-input majority gate (Fig. 2a). Computation is performed by driving the *device* cell to its lowest energy state, which will occur when it assumes the polarization of the majority of three inputs. Data can be moved in a QCA circuit with a row of QCA cells. A binary signal propagates from left-to-right because of electrostatic interactions between adjacent cells (Fig. 2b). A QCA "wire" can also be comprised of cells rotated 45-degrees (Fig. 2b,d). Here, a binary signal will alternate between a binary 1 and a binary 0. The majority voting function can be reduced to an AND or OR function by setting an input to a 0 or a 1. Inversion is also possible and QCA's logic set is thus functionally complete. QCA wires of different orientations possess the unique property that they are able to cross in the plane without destroying the value being transmitted on either wire (Fig. 2b). This is most important as, at present, all layout is assumed to be two-dimensional. By placing a 90-degree cell between and adjacent to two 45-degree cells, both the original signal value and its complement can be obtained (Fig. 2d).

QCA cells do not have to be perfectly aligned to transmit binary signals correctly. "Wires" have some tolerance for fabrication errors caused by misalignment, improper cell rotation, improper cell-spacings, etc. Although imperfect, a wire might still transmit a binary value successfully. External energy (defined as E_{kink}) can cause a cell in a wire or a system to switch into a mistake state. More specifically, the kink energy is the amount of energy that will excite a cell into a mistake state (or create a "kink" in the wire). Referring to Fig. 2c, the kink energy for off-center cells is proportional to $(1/r^5)\cos(4\theta)$. Thus, as the distance between cells

increases, the kink energy will decrease indicating that a smaller amount of external energy could excite a cell into a mistake state which is undesirable. If the angle of off-centeredness between cells increases, E_{kink} will also decrease. Disorder can also arise because of cells with improper rotation. In this situation, kink energy is proportional to $(1/r^5)\cos(2(\theta_1+\theta_2))$ where θ_1 and θ_2 are the angles at which two cells are rotated.

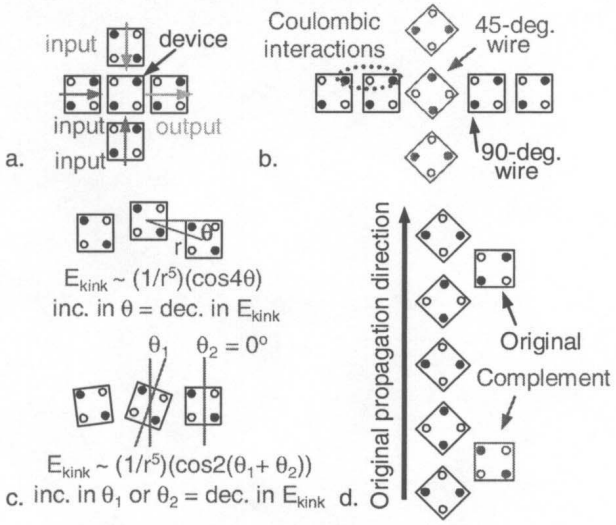

Figure 2: A QCA majority gate (a), 90-degree wire and wire cross (b), disordered cells (c) and complemented/un-complemented copies (d).

3.3. The QCA Clock

In QCA, the clock is not a separate wire or port that would be fed into a circuit like any other signal. Rather, it is typically viewed as an electric field that controls barriers within a QCA cell, which in turn control whether or not excess charge in a QCA cell can represent a binary 1 or 0.

Because of the clock, at the start of a computation QCA cells (a.k.a. "the system") would begin in a mono-stable, *restore* state. During the *switch* phase, an input potential is applied and the system is converted from a mono-stable to a bi-stable state by means of an external energy source – the clock. The binary state of the cell is then determined by some applied input potential (an external driver or another QCA cell). The input signal should be small enough that it alone cannot determine the binary state of the cell and some external clocking mechanism is also required. During the end of the *switch* clocking phase, binary information is latched in the system in the *hold* phase. A cell in the *hold* phase can be used to drive another QCA cell. Finally, the system is *restored* to its original mono-stable state and the process can begin again.

Some clocking mechanism that can allow a QCA device to transition from a monostable state to a bistable state, and then back to a monostable state is desired for molecular QCA cells. An applied external electric field could provide this functionality and will be explained using the three-dot cell as context [7], [17] (also refer to Fig. 3b). The molecule forms a 'v'-shape and charge can be localized on any one of the "dots" at the points of the 'v'. If charge is localized on one of the top two "dots", the cell will encode a binary zero; if charge is localized on one of the bottom "dot", the cell is considered turned off. (Note that in Fig. 1, the charge that moves from "dot-to-dot" is not actually an electron, but rather a hole). Silicon wires (Fig. 3a) that are buried under some substrate to which the QCA cells are attached could provide the electric field needed to control charge movement in QCA devices. The four phases of a clock signal would take the form of time-varying, but repetitious voltages applied to the silicon wires. Every fourth wire would receive the same voltage at the same time [2]. Neighboring wires see delayed forms of the same signal.

As a molecular cell transitions to the *active* state, it will take on the information (polarization) of some driving cell. The cell transitions to an *active*, bi-stable state when an electric field pushes the charge that will encode information up to the active region (or "dots") of the three-dot cell. Thus, again, the clock determines whether or not a cell is *active* or *null* and some external input will determine whether or not the cell should encode a binary 1 or 0. The charge and discharge of the embedded silicon wires will move the area of activity across the molecular layer of QCA cells with computation occurring at the "leading edge" of the applied electric field. Computation moves across the circuit in a "wave" [2], [7].

4. Physical Constraints (QCA)

Before detailing our design target, we will first discuss the technological building blocks that will constrain our designs: substrates (DNA tiles), devices (QCA molecules), and support mechanisms (a silicon clock).

A pitch matching problem exists between the substrates to which molecular QCA devices could attach and the devices themselves (and with any I/O mechanism for nano-scale devices as well [9] [10]). As stated above, molecular devices are at most a few nanometers in length or width. However, lithography that might be used to etch attachment paths is limited to 200 *nm* and 10 *nm* in the cases of optical or x-ray/e-beam lithography. Given these current resolutions, it would be either most difficult or impossible to create the detailed patterns desired (to

which devices could attach) to form computationally interesting, custom circuits [6].

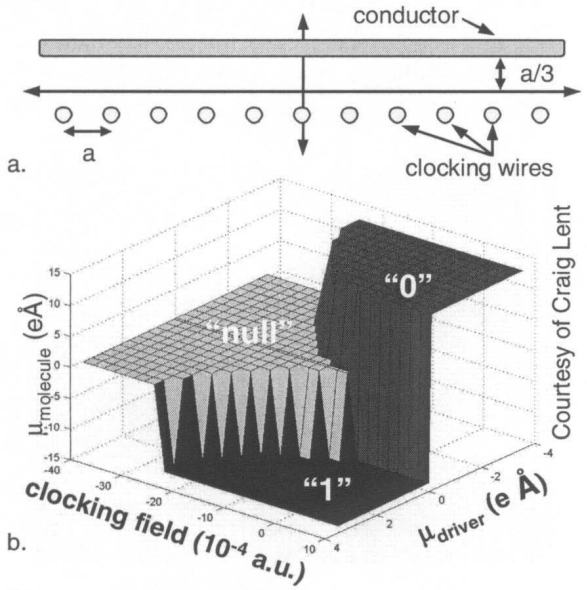

Figure 3: A 3D view of a molecular clock (a.) and an active/null view (b.)

One mechanism that might allow for selective cell placement is DNA tiles. First proposed by Seeman et al, DNA tiles (branched DNA strands that self-assemble in a regular pattern) can form rigid, stable junctions with well-defined shapes, and can further self-assemble into more complex patterns. Additionally, each DNA tile could also contain several points to which something (i.e. a QCA cell) could attach. Lieberman et. al. have developed a DNA raft 37 *nm* long and 8 *nm* tall built from 4 *nm* x 12.5 *nm* x 2 *nm* individual DNA tiles (Fig. 4). Each individual tile could hold 8 QCA cells [11]. 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. Individual tiles self-assemble and parts of the circuit board could further self-assemble in a similar manner. Finally, DNA rafts could be attached to silicon wafers using a thick poly-adhesion layer – ideal if silicon is used to form the clock circuitry.

QCA molecules are envisioned to be the real building blocks of any circuit and have been discussed at length in Section 3.1. Thus, the final major component required for a functional design is some kind of clocking mechanism. The current vision for the implementation of a clock structure for a QCA circuit calls for silicon wires embedded underneath a DNA substrate to which actual devices would have attached (see Fig. 3a). These could be charged adiabatically (charge is recycled) which should help with power dissipation [12].

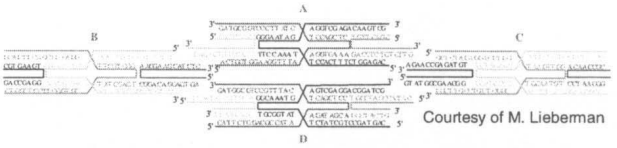

Figure 4: DNA tiles for possible FPGA substrate

5. Desired Design Attributes

These short descriptions of the essential components required for design represent the constraints that are imposed on a system designer. In order for the designs we generate to have a realistic chance of implementation, they must conform to the building blocks just discussed. In more general terms, the FPGA design target to be considered here consists of: 1. only 90-degree cells, 2. no wire crossings[1], 3. interconnect and logic blocks that will be nearly identical and simple, greatly easing the burden on chemists who must build them, and 4. cells that will be placed in the exact positions dictated by DNA tiles. These are the boundaries imposed by physical scientists with which we will work.

Before discussing specific designs, we will first explain the logic block functionality chosen for this FPGA. It will be compared to various XILINX families to provide context. Interconnection mechanisms from the XILINX 2000-4000 series will be discussed for similar reasons. These designs will be particularly useful in explaining an innovative new routing mechanism for QCA circuits in general, but especially for QCA FPGAs.

5.1. An Implementable Logic Block

When considering the functionality that an implementable FPGA design target should have, the combinational logic blocks (CLBs) of the XILINX 2000, 3000, and 4000 series FPGAs were considered (as they have interesting, yet manageable, complexity that we seek to duplicate in QCA). While varying in complexity, each CLB usually had some memory storage element, a static CMOS RAM, and multiplexors to conditionally

[1] Note that even though theoretical physics tells us that QCA wires with different cell orientations can cross in the plane with no disruption on either value being transmitted on either wire (Fig. 2d), such a configuration is not seen as realizable in near-to-midterm QCA experiments. One process envisioned for creating systems of QCA cells is as follows: first, a molecular QCA cell would be engineered that will pack and assemble properly on a self-assembled monolayer (SAM) on top of a silicon surface. Second, I/0 structures would be constructed lithographically. Third, tracks would be etched into the SAM on top of a silicon surface with EBL. Finally, the resulting "chip" would then be dipped into a bath of QCA cells for self-assembly with devices binding to the etched tracks [6]. Currently, the simple tasks of making QCA cells attach to some substrate, in some deterministic pattern, with the same cell rotations is non-trivial. Allowing for selective rotation would only complicate this process even more.

connect outputs of the RAM to the flip-flops or various outputs of the entire CLB. The RAM was used to program various logical functions. For example a function box for the XILINX 2000 series could have at most four input variables, and thus each logic block's functionality was limited to a sum of products expression of no more than four variables. Still, even this basic functionality still permits the design of small counters, etc. and such functionality is ideal for sequential machines that rely on local interconnect – shift registers, systolic arrays, etc. Consequently, if a similar and implementable logic block can be designed, it might match the architectural features of QCA quite well [14].

Still, the logic block used in this new FPGA design will just consist of a single NAND gate. While it does not have the functionality discussed above, it is functionally complete. Most importantly, the layout of a NAND-based logic block will be almost identical to that of the interconnection block that has been designed, greatly simplifying the manufacturing problem for chemists. (We should address the fact that in the introduction of this work we espoused the need to explore designs that moved beyond a functionally complete logic set, and seemingly we are not doing this here. However, we note that this work especially addresses determinism via connectivity – which will require more complex interconnect blocks and an interface to them.)

5.2. An Implementable Interconnect Block

The interconnection scheme for the XILINX 4000 series (a network of pass transistors) is illustrated in Fig. 5. The interconnection mechanisms for the 2000 and 3000 series FPGAs are somewhat simpler, and also have slightly less functionality. For example, the 2000 series allows one output of a CLB to be connected to specific inputs on its intermediate North, South, East, and West sides. Another output of the CLB has a direct connection to just the East neighbor; dedicated long line interconnects are also present. The XILINX 3000 series has a switching matrix that under different configurations permits certain connections to individual wires at certain points [14]. Interestingly, the switching matrix of the XILINX 4000 series actually permits fewer possible connections than that of the XILINX 3000 series, but makes up for it in terms of versatility. Consequently, we will focus on providing functionality akin to that of the XILINX 4000 family.

6. The 4 Diamond Structure

This section will discuss QCA implementations of logic and interconnect blocks as well as interfaces to proposed clock circuitry. Means for generating deterministic routing and area comparisons will also be discussed.

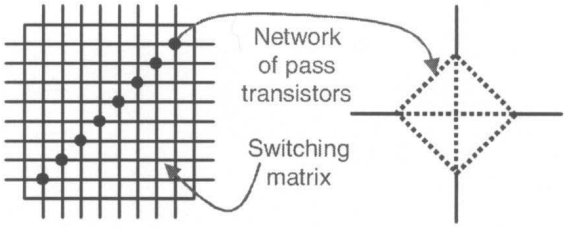

Figure 5: XILINX 4000 interconnect functionality

6.1. QCA Routing and Interconnect Blocks

The interconnection mechanism (or "network of pass transistors") envisioned for this design target is almost identical to that of Fig. 5. However, in an effort to avoid complications such as wire crossings, no wire will run vertically from top-to-bottom of the switching matrix (however, this functionality is obtainable by turning on two, adjacent side connections – i.e. two adjacent pass transistors); turning groups of QCA cells on or off will be accomplished with the clock. This switching matrix encompasses much of the functionality discussed above for various series of commercial FPGAs. Consequently, it should be possible to design interesting circuits given this "system-level constraint."

A schematic and a snapshot of what a physical design might look like appear in Fig. 6a and Fig. 6b respectively. Both depict three interconnect blocks and illustrate possible ways that they can interact with the NAND-gate logic block. Given the way this circuit is wired, there are three entry points to a logic block – however, user controlled clock circuitry must ensure that only two of those inputs actually propagate to the NAND gate. Entry points are labeled A, B, and C. Thus, possible logical functions the block can perform are A NAND B, A NAND C, B NAND C, and B NAND B. With the design envisioned for the interconnection block, any combination of the five paths could be turned on at any given time. However, if these connections were improperly programmed (i.e. too many paths are "on"), signals collisions would occur. Consequently, this design will still not compensate for programmer error.

Fig. 6b provides some initial insight into what a physical design might look like. Individual rectangles represent single DNA tiles (capable of holding 8 cells per tile); squares represent QCA cells. Note that the actual placements of QCA cells needed to implement the logic block and the interconnect block are nearly identical. The only significant difference is that while the interconnection blocks have a wire of QCA cells that spans the middle of the block, in the logic block, this

wire is replaced by a single cell that will be engineered to always represent the equivalent of a binary 0 in order to implement the NAND function. Similarly, the output of the logic block must be inverted.

Before discussing specifically how an interconnect block can perform selective routing, we should express one word of caution: namely, with this design, we must make sure that every QCA cell is clocked at all times. If cells are not directly involved in a computation, they should be turned off. Otherwise "noise" will be latched and would then become garbage data that could interfere with useful computation. However, with the left-to-right, "wave" clock scheme envisioned, this should not be a problem.

6.2. Routing with the Clock

We envision that silicon circuitry will be used to handle the conditional routing and switching for the design shown above in Fig. 6. A discussion of how this functionality will be provided will take place in the context of Fig. 7 – which depicts a "cross section" of what an interconnection cell or switching element might physically look like (including underlying circuitry needed to control it). The design will be considered in layers. The first layer would simply be the DNA substrate to which molecular QCA cells have been attached. Underneath this would be a layer of silicon wire segments that form a pattern identical to the pattern formed by molecular QCA cells attached to DNA tiles (i.e. a diamond with a line across the middle). Each one of these wire segments would then be connected via a pass transistor (i.e. in another layer of metal) to additional silicon wires that would actually form the computational wave and would move charge from one wire to another. However, the QCA cells would be directly controlled by the shorter wire segments directly underneath the DNA substrate. By selectively turning pass transistors on or off, groups of QCA cells can also be selectively turned on or off. The pass transistor will ensure that these groups of QCA cells are not influenced by the electric field propagated by the silicon wires in the lowest layer. Or in other words, switches are created because an artificial "gap" can be made in a QCA wire, putting a group of cells into the null state.

While this is by no means seen as a final, definitive solution, it does help to facilitate data movement in QCA circuits given the constraints currently faced (no wire crossings, one cell type), and should be most useful in moving toward real and computationally useful QCA circuits. Scaling problems associated with pass transistors may eventually be avoided by using nanowires or diodes to "program" routing. Additionally, routing could be permanently defined by replacing transistors with vias between the two layers of wire. Removing cells with an AFM tip is an option for smaller systems [11].

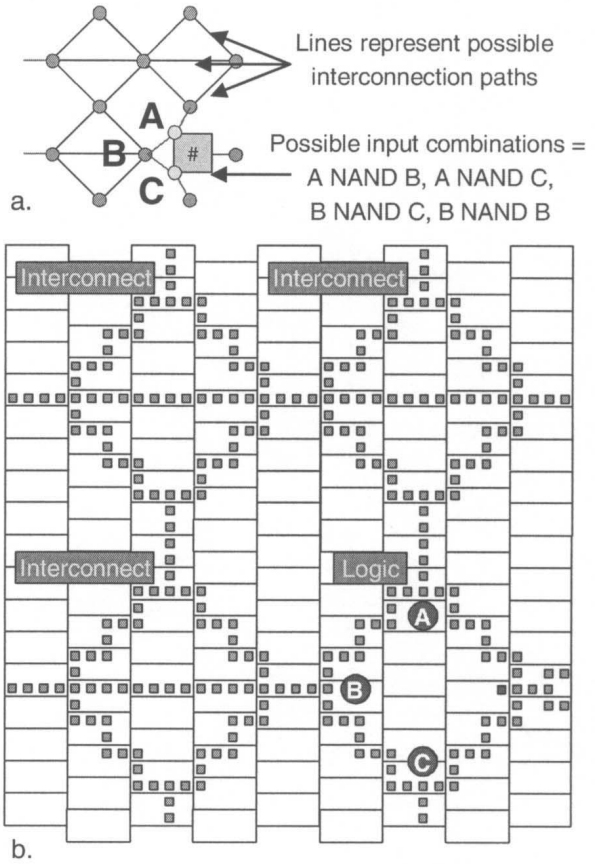

Figure 6: "4-diamond" schematic (a) "4-diamond" assuming specific DNA attachment (b)

A grid of the blocks like those shown in Fig. 6 will form the FPGA envisioned for this first design target. By selectively turning connections on or off, and by feeding in appropriate inputs, interesting circuits and systems can be created. For example, Fig. 8 illustrates how this FPGA could be programmed to implement an ALU for an accumulator-based processor called Simple 12[2]. (Although not illustrated here, feedback is also possible).

6.3. Area results (and the clock structure)

Area requirements for this FPGA will be considered using the implementation/programming connections of

[2] Note that Simple 12 has 12-bit data words, an 8-bit addressable memory, and uses minimal hardware. Consequently when considering design in QCA, much of the physical layout can be performed by hand. Its instruction set includes arithmetic instructions, loads, stores, and jumps. Having A and B inputs, the ALU can generate the outputs A+B, A-B, A AND B, A OR B, B+1, B, and 0.

one bit slice of a Simple 12 ALU (Fig. 8) as context. While we could consider area in terms of QCA cells just

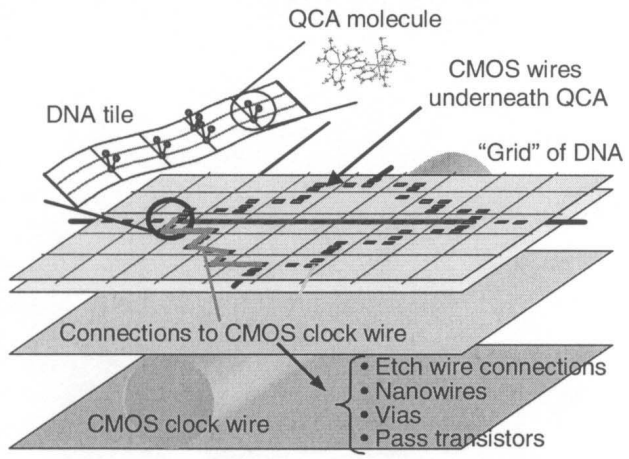

Figure 7: A 3D view of how to create selective routing elements in a computationally useful FPGA

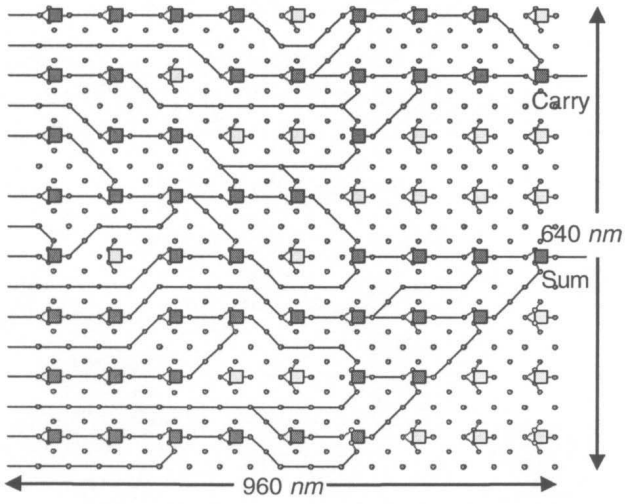

Figure 8: Connections to implement the Simple 12 ALU in an FPGA.

by looking at the ALU schematic, realistically this number would be meaningless – we are now considering specific substrates with specific dimensions. The design shown in Fig. 6b is really 9 DNA tiles wide and 18 DNA tiles tall. As a single DNA tile is 8 *nm* x 4 *nm*, Fig. 6b would require a total area of 7.77 x 10^{-3} μm^2. Using this number as a base, the Simple 12 ALU requires 0.55 μm^2 of area.

Unfortunately, even these area numbers are not completely accurate. We are not constrained just by the dimensions associated with DNA tiles, but also by the lithography resolutions of CMOS. Specifically, when considering just one of the interconnection blocks (i.e. the upper left quadrant of Fig. 6), we will need a silicon wire that is small enough to control a small group of QCA cells (i.e. one edge of the "diamond"). One edge of the "diamond" shown in this picture is only about 7 QCA cells in length. Given 1 *nm* molecular cell spacing, one edge would thus require approximately 7 *nm* of space. (And as two edges of the "diamond"' form a 45-degree right triangle, the wire across the center will require 7 sqrt(2) *nm*). Not even a projected lithographic process can provide this resolution and consequently, this design must be scaled up for silicon. To provide a flavor of how much area will realistically be required, we could just make each wire segment in the snapshot (Fig. 6) five times longer (as 7 *nm* x 5 = 35 *nm* which corresponds to a 0.035 μm process). As a result, the total area required for the Simple 12 ALU in the context of the first FPGA design target grows to 13.9 μm^2.

For reference, the densest design of one bit slice of a QCA Simple 12 ALU (custom, non-silicon scaled, and no wire crossings) is 88 cells tall, 35 cells wide, and requires just 3.08 x 10^{-3} μm^2 of area [13]. Thus, the non-silicon scaled, NAND-based ALU is approximately 178 times larger than a custom design, while the silicon-scaled design is approximately 4500 times larger than the custom design. Still, this is not exactly a fair comparison as the FPGA-based schematic consist of only a single NAND gate. However, a more interesting comparison might be between our QCA NAND-based ALU bit slices, and one bit slice of a CMOS implementation of the same logic – scaled down to a 0.035 micron process. The CMOS implementation requires 14.1 μm^2 of area. When scaled to a 0.022 μm process, density numbers are similar – 5.41 μm^2 for a QCA version of the Simple 12 ALU, 5.56 μm^2 for a CMOS version. All results are summarized in Table 1.

Thus, the area of a scaled, *NAND-based* QCA ALU bit slice is essentially identical to near end of the curve CMOS implementation.

7. Conclusions

Before commenting on our results, we should first note that DNA is not viewed as the most optimal substrate for systems of QCA cells. Background charge could disrupt computation[3] and as systems get larger,

[3] To explain, we must consider possible electrostatic interactions between chemical molecules and their environment, and to what degree they are a function of the distance (d) between entities. Examples include: interactions between two charges ($1/d$), interactions between a charge and a dipole ($1/d^2$), and charge induced dipole interactions (an anion/cation may induce a dipole in a polarizable molecule and thus be attracted to it) ($1/d^3$) [18]. To understand how some of these interactions can affect QCA cells, we will first considered kink energy as a function of driver dipole strength [17]. In the presence of a stronger driver dipole, the energy required to induce a mistake state increases (a good thing). As dipoles are a function

Table 1: Summary of area results

QCA Non-Si scaled, NAND-based	0.55μm^2
QCA Si-scaled (.035 μ proc.), NAND-based	13.9 μm^2
CMOS (.035 μ proc.)	14.1 μm^2
QCA Si-scaled (.022 μ proc.), NAND-based	5.41 μm^2
CMOS (.022 μ proc.)	5.56 μm^2
QCA, Completely custom	0.00308 μm^2
QCA Non-Si scaled / QCA, Completely custom	178
QCA Si-scaled (.022) / QCA, Completely custom	1756
CMOS (.022) / QCA, Completely custom	1805

"sheets" of DNA become less stable [11]. However, it can be most useful for prototyping small to medium sized QCA systems. Additionally, our CMOS implementation of Simple 12 uses only 2 layers of metal. Realistically, an implementation with a state of the art fabrication process (7-8 layers of metal) would be denser. However, a NAND-based QCA implementation is competitive with end of the road map CMOS – and CMOS is the limiting factor in QCA's density scaling.

To conclude, in this paper we have demonstrated that if physical scientists cannot deterministically place QCA devices to create deterministic circuits, determinism can be generated via QCA's clocking structure – independently of the "nano" part of fabrication. The problem becomes one of electric field control and we propose this as an important area of research to advance systems of QCA cells. Our most simple design strategy addresses both logical completeness *and* connectivity and the results are comparable with end of the roadmap CMOS. Finally, and perhaps most importantly, these conclusions were reached because of systems-level research, not physical science – indicating that systems-level work should help to guide physical device development, should be part of research in nanotechnology, and should help us reach working nano-systems sooner.

8. Acknowledgements

The authors would like to acknowledge NSF and SRC.

of the distance between them (d), intuitively closer dipoles imply stronger energies of interaction, while greater distances imply weaker dipole interactions and lower kink energies. DNA has a background charge associated with it. As stated above, the energy of interaction between a charge (i.e. from DNA) and a dipole (i.e. a QCA molecule encoding a 1 or 0) is proportional to $1/d^2$ – and thus will have a greater (and negative) influence on the desired dipole-dipole interactions that would move binary information from QCA cell-to-QCA cell. Charge-dipole interactions could provide a kink energy, and hence induce a mistake state.

9. References

[1] S.C. Goldstein and M. Budiu, "NanoFabrics: Spatial Computing Using Molecular Electronics", *Proc. of the 28th Int. Symp. on Computer Architecture*, 2001, pp 178-191, ACM Press, Göteborg, Sweden.

[2] K. Hennessy and C.S. Lent, "Clocking of molecular quantum-dot cellular automata", *J. of Vacuum Science and Technology B*, Vol. 19(5), pp 1752-1755, Sep-Oct 2001.

[3] P.D. Tougaw and C.S. Lent, "Logical Devices Implemented Using Quantum Cellular Automata", *J. of App. Phys.*, Vol 75, pp 1818, 1994.

[4] C.S. Lent and P.D. Tougaw, "A Device Architecture for Computing with Quantum Dots", *Proc. of the IEEE*, vol 85, pp541, 1997.

[5] C.S. Lent, "Molecular Electronics: Bypassing the Transistor Paradigm", Science, Vol 288, pp 1597-1599, 2000.

[6] 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", *Annals of the New York Academy of Science*, vol 960, pp 225-239, 2002.

[7] C.S. Lent, G.L. Snider, G. Bernstein, W. Porod, A. Orlov, M. Lieberman, T. Fehlner, M. Niemier and P. Kogge, *Quantum-Dot Cellular Automata*, 2003.

[8] A. Aviram, "Molecules for Memory, Logic, and Amplification", *J. Am. Chem. Soc.*, vol 110, pp 5687-92, 1988.

[9] M. Oskin, F.T. Chong, I.L. Chuang, and J. Kubiatowicz, "Building Quantum Wires: The Long and the Short of it", *Proc. of the 30th Int. Symp. on Computer Architecture*, pp 374-385, 2003, ACM Press, San Diego, CA.

[10] André DeHon, "Array-based Architectures for FET-based Nano-scale Electronics", *IEEE Trans. on Nanotechnology*, Vol. 2, No 1, pp 23-32, Mar 2003.

[11] Personal communications with Marya Lieberman, 2003.

[12] Jay Moon and Bill Athas, "Energy-recovery CMOS Design", Presentation at Apple Computer, March 5, 2001.

[13] M.T. Niemier and P.M. Kogge, "Exploring and Exploiting Wire-Level Pipelining in Emerging Technologies", *Proc. of the 28th Int. Symp. on Computer Architecture*, pp 166-177, 2001, ACM Press, Göteborg, Sweden.

[14] Jenkins, "Designing with FPGAs and CPLDs", PTR Prentice Hall, New Jersey, 1994.

[15] M.T. Niemier, A.F. Rodrigues, and P.M. Kogge, "A Potentially Implementable FPGA for Quantum Dot Cellular Automata", *1st Workshop on Non-Silicon Computation*, at 8th Int. Symp. Of High Performance Computer Architecture, Boston, MA, Feb. 3, 2002.

[16] J.R. Heath, P.J. Kuekes, G.S. Snider, and R.S. Williams, "A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology," *Science*, vol. 280, p. 1716-21, June 12 1998.

[17] C.S. Lent, B. Isaksen, and M. Lieberman, "Molecular Quantum-dot Cellular Automata", *J. Am. Chem. Soc.* 125, pp. 1056-63, 2003.

[18] C.K. Mathews, K.E. van Holde, and K.G. Ahren. *Biochemistry*. San Francisco: Addison Wesley, 2000.

[19] Paul Beckett, "Towards a Reconfigurable Nanocomputer Platform", *1st Workshop on Non-Silicon Computation*, at 8th Int. Symp. on High Performance Computer Architecture, Boston, MS. Feb. 3, 2002.