PLAs in Quantum-dot Cellular Automata

Michael Crocker and Xiaobo Sharon Hu
Dept. of Comp. Sci. & Eng.
University of Notre Dame
Notre Dame, IN 46556, USA
Email: {mcrocker, shu}@nd.edu

Michael Niemier
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
Email: mniemier@cc.gatech.edu

Minjun Yan and Gary Bernstein
Dept. of Elec. Eng.
University of Notre Dame
Notre Dame, IN 46556, USA
Email: {myan, gbernste}@nd.edu

***Abstract*— Various implementations of the Quantum-dot Cellular Automata (QCA) device architecture may help many performance scaling trends continue as we approach the nano-scale. Experimental success has led to the evolution of a research track that looks at QCA-based design. The work presented in this paper follows that track and looks at *implementation friendly*, programmable QCA circuits. Specifically, we present a novel, QCA-based, Programmable Logic Array (PLA) structure. Our PLA is capable of providing defect tolerance at both the device and architectural level, and limits the amount of determinism required in any fabrication process. The design is compact, exploits properties unique to QCA devices in order to ease programmability, and is relevant to all implementations of QCA.**

I. Introduction

The 2005 edition of the ITRS roadmap stresses that in order to achieve long-term MOSFET scaling (2014-2020, L_g=15nm), experimental advances relating to physical devices are clearly needed [1]. For silicon-based systems, much of this experimental work is being done with the belief that technology requirements for which there are currently "no known solutions" will ultimately be achieved. However, another research track is currently working to address a second possibility – one where red areas of the roadmap will *never* be achieved, and any workaround will indeed be irrelevant. Much of this work has focused on computational models and emerging technologies that are not based on CMOS MOSFETS, but can still offer their own unique "wins."

For MOSFETS, the ITRS roadmap stresses that "the choice of optimum device structures, their physical characteristics, and the constraints of cost-effective processing will become very important *along with the construction of their circuit architectures*." We believe that the same will be true for emerging technologies too. This paper looks at circuit constructs and system architectures for the QCA device architecture. Various implementations of the QCA concept will be considered (as each *implementation* can potentially offer its own unique performance wins). All designs will be considered within the context of what is physically realizable.

First proposed in the early 1990s [2], QCA accomplishes logical operations and moves data via nearest-neighbor interactions rather than with electric current flow. Initial (and current) experiments have focused on a *metal-dot* implementation of a QCA device [3], [4], [5], but two other implementations are possible as well[1].

A theoretical analysis of a *molecular* implementation, has shown that QCA-based circuits could be clocked at an extremely high frequency (adiabatically at 1 THz [6], [7], [8]), potentially lead to circuits with densities that are 3-4 orders of magnitude beyond what end-of-the-curve CMOS can provide [9], [10], and should dissipate very little power [6]. *Experiments* have shown that a molecule is capable of switching between configurations that could represent binary 0 and 1 states.

In a *magnetic* implementation device switching times are slower, but this implementation offers the advantages of large energy differences between states, tolerance to the effects of radiation, and the *demonstrated* room temperature operation of a functionally complete logic set. A theoretical analysis shows that the amount of power dissipated by magnetic QCA devices should also be quite low (0.1 W if 10^{10} devices in $1cm^2$ switch simultaneously) [11], [12], [13]. (Essentially, most applications should require no power other than the power required to switch their state [14].)

The experiments and theoretical analyses discussed above represent important milestones on the path to a functional and realizable *system*. The work presented here considers a different milestone on this critical path – the "construction of circuit architectures".

It is well recognized that any realized circuit or architecture at the nano-scale will have to be able to tolerate higher percentages of defects than current CMOS circuits. The task of ensuring that a system is still functional post-fabrication falls largely to the circuit designer or computer architect who, in many instances, have leveraged reconfigurable/reprogrammable structures. Largely for this reason, reconfigurable logic has been studied by a number of research groups for different nanoscale technologies. For example, DeHon and Wilson proposed a nanowire-based sublithographic PLA design [15]. Likharev and Strukov presented the CMOL FPGA [16]. Hogg and Snider showed interesting algorithmic approaches to map logic functions to PLAs with defective crossbar switches [17]. We present a novel PLA structure based on the QCA device architecture. It is fully re-programmable and allows defects to be readily detected and isolated. The design is compact and

[1] *Semiconductor-based* QCA devices have also recently been realized. We will discuss semi-conductor QCA briefly in Sec. II.

easily extensible.

Before continuing, we note that in addition to addressing the viability of a technology at the systems-level, there are two additional reasons why this research is especially timely. First, for some implementations of QCA, all of the constructs needed for a functionally complete logic set have been experimentally demonstrated. The next logical step is to consider architectures that could provide system-level wins and to begin experimentally verifying their core components. This work is a first step in that direction – and should allow computer scientists to further refine research in the physical sciences (ideally by specifically defining what characteristics and feature sizes are needed for wins at the systems-level).

Second, recent work with a molecular implementation of QCA has shown that devices and substrates should *not* be considered independently of circuits and systems [18]. Acceptable device, scaffolding, and substrate characteristics should be determined at least in part by considering how all of these components will form logic. Making and placing one device is difficult enough. We should ensure that once fabricated, it will facilitate logic for a variety of applications.

We will address all of these issues in this paper for an architecture that is well-suited for nano-scale devices. We begin in Sec. II with a discussion of the QCA concept, implementation specific background, and a review of related work. In Sec. III we discuss our reconfigurable PLA structure. Sec. IV will address how we can maintain the necessary state to ensure that our PLA will repeatedly execute the same logic function. Work discussed in Sec. III and IV is supported by simulation where relevant. In Sec. V we will discuss how our PLA structure would operate. We conclude and discuss future work in Sec. VI.

II. Background and Related Work

In this section we introduce the QCA device architecture, detail the experimental state of the art for each implementation of QCA, explain how a clock structure is used to facilitate reliable computation, and review related work.

A. QCA Basics

1) 1's and 0's: QCA represents information by encoding binary numbers into cells that have a bi-stable charge configuration. A QCA cell can consist of 2 or 4 "charge containers" (i.e. quantum dots) and 1 or 2 excess charges respectively. One configuration of charge represents a binary '1' and the other a binary '0' (Fig. 1a) [19]. Logical operations and data movement are accomplished via Coulomb (or nearest-neighbor) interactions. QCA cells interact because the charge configuration of one cell alters the charge configuration of the next cell.

2) Circuit Constructs: Fig. 1b–Fig. 1e illustrate several basic QCA circuit elements [2], [10]. A QCA wire (Fig. 1b) is just a line of QCA cells. The wire is driven at the input cell by a cell with a fixed/held polarization. The cells do not need to be spaced exactly the same distance apart. The majority gate (Fig. 1c) implements the logic function $AB + BC + AC$.

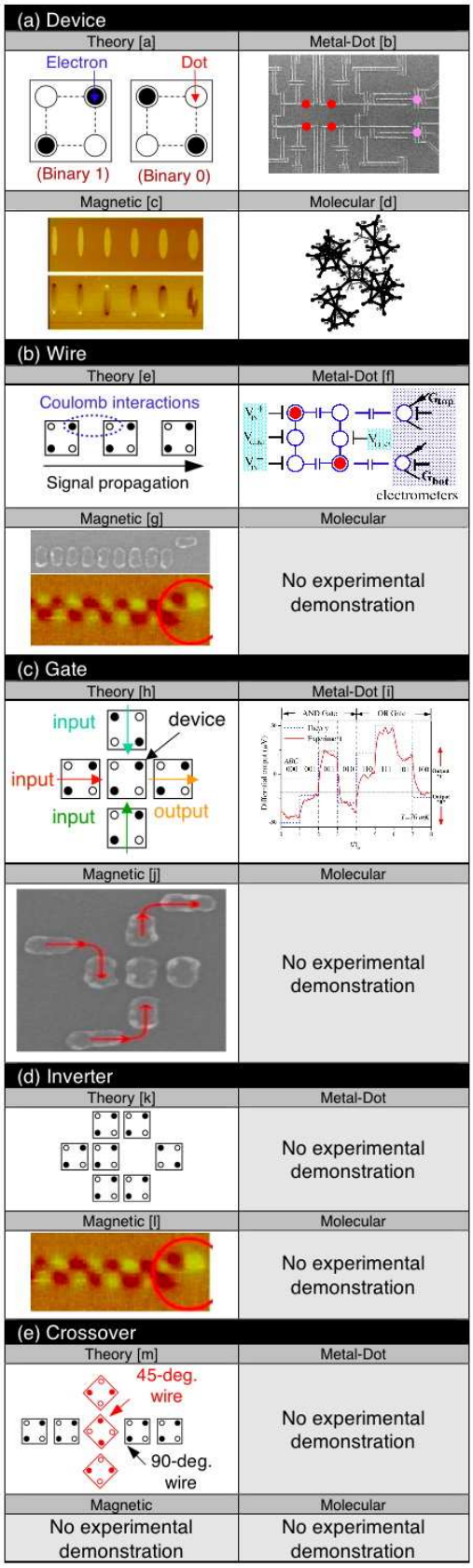

Fig. 1. Schematics of the fundamental structures needed to build QCA circuits and their experimental state of the art: (a) basic devices, (b) wires, (c) logic gates, (d) inverters, and (e) crossovers. Constructs that have not yet been experimentally demonstrated are noted.

The output cell assumes the polarization of the majority of the 3 input cells [19]. By setting one input of a majority gate to a logic '0' or '1', the gate will execute an AND or OR function respectively. An inverter can also be easily built with QCA devices (Fig. 1d). QCA wires with different orientations (Fig. 1e) can theoretically cross in the plane without destroying the binary value on either wire[2].

B. QCA Implementations

There are at least four different material systems that can be used to make a QCA device. Because of space limitations, Fig. 1 illustrates the state-of-the-art of only 3 of the 4 implementations (and also because there are multiple ways to realize a semiconductor-based QCA device).

1) Metal-dot QCA: Many metal dot QCA devices and circuit elements have been implemented using lithography at a fairly large scale (see Fig. 1a-d) [3], [4], [5], [20], [21]. These prototypes demonstrate that the basic QCA concept is valid – but they only operate at temperatures of about 1K. There are ongoing research efforts working to raise the operating temperature of metal-dot devices.

2) Semiconductor QCA: Silicon-germanium quantum fortresses [22], silicon p-doping [23], gallium arsenide [24], and silicon-dot SETS [25] are all possible material systems that could be used to make QCA devices – and could simultaneously raise the operating temperature to about 77K. One "killer application" for this technology might be to use these devices to facilitate the classical part of a quantum computation [26], [27].

A recent paper by Hollenberg et al. [27] describes a 2D implementation where the qubits are integrated on a single substrate with driving circuitry and other classical computational elements. The spin-based qubits are in the central section of a tile and are coupled through spin-to-charge transducers [28] to the classical circuitry. Connecting these tiles on a substrate creates a two-dimensional quantum and classical computing array. Such a design, while optimizing the interactions of the quantum and classical sections, places great demands on the overall system. To maintain reasonable coherence times, the computer must operate at cryogenic temperatures. However the required high-speed classical logic will dissipate a significant amount of power near the qubits, threatening to raise their temperature. In contrast, QCA will make an excellent match for the classical computing needs of a quantum computer.

3) Molecular QCA: A QCA device could also be made from a single chemical molecule. Molecular QCA devices can be made from mixed-valence compounds which contain multiple redox centers in different oxidation states. Each "quantum dot" would be a single redox center, and the redox centers that make up a two- or four-dot QCA cell would be rigidly held together by covalent bonds [29]. A recent experiment demonstrates that applying reasonable electric fields can move a charge between two redox sites of a molecule engineered to function as a two-dot QCA cell [30]. This room temperature experiment shows a self-assembled monolayer of molecules switching between configurations that could be used to chemically represent a binary 0 or 1. Four-dot QCA molecules have also been made [29] (see Fig. 1a) and promising I/O methodologies exist [26].

There are several methods that could be used to deterministically place QCA devices to form circuits. One viable target assumes DNA tiles fabricated by Seeman and Winfree [31] as the scaffolding. QCA molecules would attach covalently to modified DNA nucleotides (see [32] for more detail). Assuming the tiles retain the B-DNA duplex, sites in the major groove could serve as attachment points (with approximately 2 nm between sites in the y-dimension and 3.6 nm between sites in the x-dimension). There are good precedents for rigid placements of metal complexes on duplex DNA [33] (meaning most cells should have the same orientation) and the attachment of nanoparticles to a DNA-based scaffold has been experimentally demonstrated [34]. Other promising mechanisms for deterministically patterning molecules are detailed in [35], [36], and [37]. Our designs should be applicable to any of these methods.

4) Magnetic QCA: Finally, QCA devices can be constructed with nano-scale magnets [11], [12] – where information is transferred via magnetic, rather than with electric dipoles. While current implementations are larger (on the order of 100 nm per cell), and slower (a maximum cell switching time of about 0.1-1.0 ns is expected [13]), the energy difference between two states is enormous (100-200 kT at room temperature), and cell sizes could be reduced to approximately 20 nm [11]. Experiments have demonstrated lines of 64 and 69 cells with perfect coupling (see Fig. 1b), as well as an inverting majority gate (see Fig. 1c) [13]. Inversion is possible by creating a line with an odd number of devices (see Fig. 1d).

C. Clock Structure

Regardless of implementation, a circuit or system made from QCA devices will realistically require some kind of clock structure that directs a computation and provides gain. Here we briefly explain how a clock structure will facilitate computation for each implementation.

1) Charged-based QCA: For charge-based QCA, a physical manifestation of a clock will most likely take the form of lithographically defined, conducting, metal wires [38] (see Fig. 2a). We will discuss what such a clock might look like in the context of a molecular implementation.

QCA devices will sit above the wires and a ground plane would sit above the devices. The wires will produce an electric field that will help to induce dataflow in the QCA logic and interconnect (see [38] and [18] for more detail). When a strong, positive electric field is applied to a device, electrons will be drawn out of the active region of the cell as negatively charged electrons are attracted to a region of positive charge

[2]Note that while crossover functionality has not yet been experimentally demonstrated for any implementation, signals can be crossed logically in the plane with a combination of majority gates and only one cell type [9]. Additionally, crossings could be accomplished in a magnetic implementation by using out-of-plane magnetization. We are working with other physical scientists to experimentally demonstrate these structures.

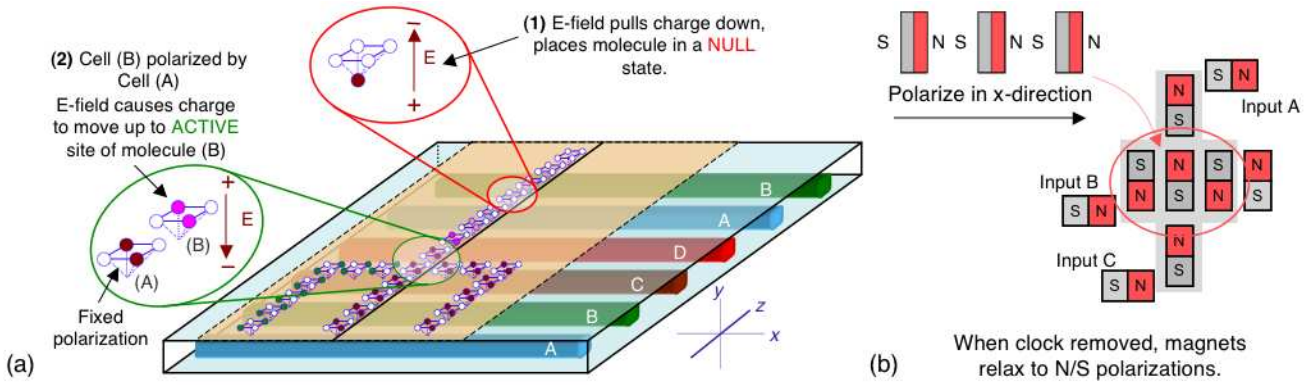

Fig. 2. (a) A three-dimensional view of the proposed clock structure for charge-based QCA devices, (b) how a magnetic field (or "clock") will be used to facilitate computation in a magnetic implementation of QCA.

(see Fig. 2a inset (1)). This puts the molecule in a NULL state where it will not participate in a computation or hold information. When the field becomes strongly negative, the electrons will be forced up into a *locked* state.

Which of the two upper "dots" are occupied is determined by a driving cell that splits the degeneracy and determines whether or not a QCA device is a 1 or 0 (see Fig. 2a inset (2)). By applying time-varying voltages on the metal wires, we can "clock" the cells' activity to produce a flow of information. The required voltages are periodic and adjacent wires have a $\pi/2$ phase shift between them. Every fourth wire will have the same applied signal. The four-phase signal on the buried wires should induce sinusoidal regions of activity (i.e. computation).

2) Magnetic QCA: For the structures illustrated in Fig. 1, the clock took the form of a periodically oscillating external magnetic field that drove a system to an initial state, and then controlled the relaxation of the said system to a ground state. The role of the external clock field is to overcome the energy barriers between metastable states and the ground state. Clocking can be performed by applying the magnetic field along the short axis of the dots (as illustrated in Fig. 3 and discussed in [14]). In Fig. 3b, the external field turns the magnetic moments of all magnets horizontally into a neutral logic state (the equivalent "null" state) against the preferred magnetic anisotropy. This is an unstable state of the system, and when the field is removed, the nano-magnets relax into the antiferromagnetically ordered ground state as illustrated in Fig. 3d. If the first dot of the chain is influenced by an input device during relaxation, then its induced switching sets the state of the whole chain due to the bipolar coupling [14].

In the majority gate experiments discussed in [13], the inputs are represented by horizontally-oriented magnets (see Fig. 1b) – and the magnet's input value is actually determined by the *position* of the input magnet. Referring to Fig. 1c, notice that each input of the majority gate sees a different value (based on position) even though these inputs magnets have the same polarization. To effect line switching, we would want to have more local control. This local control to effect line switching can be provided by an input device that produces

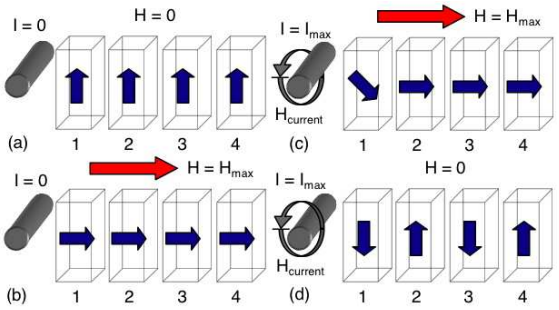

Fig. 3. Operating scheme of a wire: (a) initial configuration, (b) high-field ("null") state, (c) after the application of the input, and (d) the final ordered state [14].

a local magnetic field, only influences the first nano-magnet in the chain, and which is oriented parallel with the long geometrical axes. This could also be realized by a current carrying electrical wire (for example) as illustrated in Fig. 3.

Now, we briefly consider these methods in the context of a larger circuit. Experimental work has shown that there is a critical/maximum length of such a chain such that dataflow progresses without error [14]. If the critical path in a circuit is larger than this distance (and it most certainly will be), then a computation will take multiple clock cycles. However, if an external field is applied to a *chip*, we would need to take care that intermediate values are not erased when the next line of magnets is relaxed. (The analog to conventional processing is that we need a latch between pipe stages). We envision creating more local fields by simply moving current through wires underneath the devices as discussed above (simulation is ongoing). Magnetic fields would no longer be global, but rather local – and the necessary pipelatches will be inherent.

As one final note, unlike charge-based QCA, as evidenced by the majority gate experiment, there is an "inherent capacitance" associated with magnetic QCA. A clock facilitates computation, but it does not necessarily direct it.

D. Related Work

In the context of PLA designs for the QCA device architecture, to the best of our knowledge, related work is limited

to (a) [39] which utilizes crossbar networks to cross signals serially in time (this will have a significant – and adverse – effect on latency), (b) [40] which discusses the building blocks for a self-assembled FPGA design (but would require precise alignment of the clock structure with self-assembled QCA circuitry), and (c) [41] in which a versatile And-Or-Inverter gate is logically produced by precisely placing two majority gates together (but would require precise placement of individual QCA devices – especially difficult if QCA molecules are assumed). Thus, all of these designs are either difficult to implement, sacrifice performance, or do not consider realistic implementation constraints. In contrast, the work presented here continues in the vein of our existing work (i.e. [9]) which examines the implications of physically implementing QCA-based circuits. It also enhances a feedback loop that has helped to shape existing experimental work (regardless of implementation) to best target architectures well-suited for nano-scale devices and the QCA device architecture.

III. A PLA Structure for QCA

Programmable Logic Arrays (PLAs) can easily implement any logic function in a two-level sum-of-products representation – or in other words, groups of AND terms ORed together. PLAs could be used to implement state machines, lookup tables in FPGAs, complex control logic, components needed for processor datapaths, and many other computationally interesting structures (see [15], [42], [43]). Unlike mask-programmable CMOS PLAs which can be programmed once at fabrication time (but like the NW-based PLAs described in [15]), our QCA-based PLA could be reprogrammed repeatedly to target different applications [44]. The structure allows circuit defects to be readily detected and isolated, the design is compact and easily extensible, and could be made almost entirely from AND gates, OR gates, and wire segments (essentially three simple parts reducing the amount of determinism required in any fabrication process). In this section we discuss our design at the logic level in Sec. III-A and discuss implementation specific experiments in Sec. III-B.

A. Reprogrammable PLA Cells and the Array Structure

In an NMOS PLA, each intersection of orthogonal wires can function as either an AND or OR gate or as a simple wire. In our design, both AND and OR planes will consist of QCA devices arranged in such a manner that they are capable of mimicking this functionality. We have created a design where crosspoint functionality is determined by a programmable bit. The state of the bit will allow groups of QCA devices that act as crosspoints to function as a gate or wire. This configuration has the added benefit of being reprogrammable post-fabrication.

We will discuss our PLA in the context of "cells". Here, the term "cell" does not refer to a QCA device, but rather the group of devices that implements a programmable crosspoint. A schematic of one cell appears in Fig. 4a. This structure contains the programmable *select bit* (denoted by S) and two majority gates – one configured to act as an AND gate and the other configured to function as an OR gate. A CMOS analog appears in Fig. 4b while a QCA schematic appears in Fig. 4c.

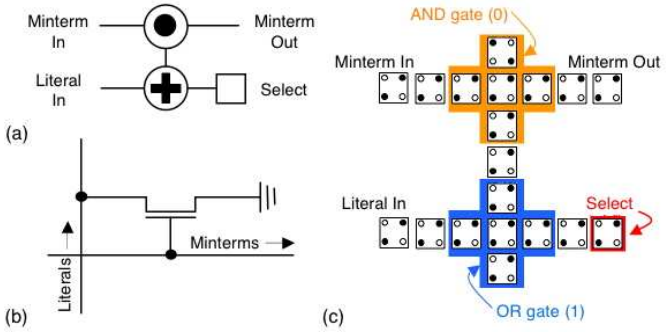

Fig. 4. (a) QCA cross point schematic, (b) CMOS equivalent, (c) a QCA "implementation".

We consider one crosspoint of an AND plane to demonstrate how our design will function. Referring to Fig. 4c:

- if S=1, MINTERM_OUT = *(literal in)* • *(minterm in)*
- if S=0, MINTERM_OUT = MINTERM_IN

In other words, if S=1, the PLA cell will act as an AND gate (we refer to this as *logic* mode), and if S=0, the PLA cell will act as a wire (we refer to this as *wire* mode). The ability to conditionally set each select bit makes our PLA reprogrammable.

For sake of completeness, we briefly comment on how the OR plane would function. For the OR plane the position of the AND and OR gates in one "cell" would be reversed. Thus, referring to Fig. 4c, MINTERM_IN would be an input to an OR gate and LITERAL_IN would be an input to an AND gate. Additionally, the select bit should be set to 0 for *logic* mode and 1 for *wire* mode. Therefore:

- if S=0, MINTERM_OUT = *(literal in)* • *(minterm in)*
- if S=1, MINTERM_OUT = MINTERM_IN

By leveraging the structures just discussed, it is relatively easy to construct the logic required to form a PLA of arbitrary size. Fig. 5 illustrates how the "cell" construct in Fig. 4 can be used to make entire AND and OR planes. The T_{XY} labels represent the terminal QCA cells at the PLA boundaries. X has a value of T (top), B (bottom), L (left), or R (right) and Y will have a value corresponding to the row or column number. If we wanted to configure the PLA in Fig. 5 to perform the majority voting function, we would set our select bits as shown by the values in the inset triangles.

B. Implementation Specific Experiments

It should be fairly obvious that this arrangement of QCA devices will *logically* allow us to construct a PLA. However, as mentioned earlier, we are also interested in the eventual physical realizations of our designs. In this regard, we have leveraged two different (physical-level) simulations suites to determine how this logic might function if it were implemented with molecular QCA devices. (We do not consider metal-dot or magnetic QCA here because, as seen in Fig. 1,

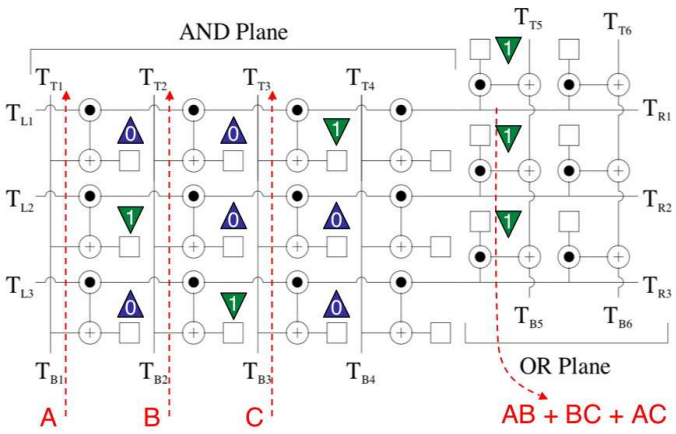

Fig. 5. A PLA array that implements the majority voting function

Minterm-In	Literal-In	Minterm-Out	Probability
0	0	0	99.90%
0	1	0	99.99%
1	0	0	98.86%
1	1	1	99.99%

(a)

Minterm-In	Literal-In	Minterm-Out	Probability
0	0	0	99.99%
0	1	0	99.94%
1	0	1	99.99%
1	1	1	99.88%

(b)

Fig. 6. Statistical mechanics simulation results. (a) `AND` plane cell in Logic Mode, (b) `AND` plane cell in Wire Mode.

the wires and gates needed to form the structure shown in Fig. 4c have already been *experimentally* demonstrated for both of these implementations.)

The two most widely used tools for studying small to medium-sized systems of QCA devices are QCA Designer [45] and M-AQUINAS [46]. Both are clock driven. QCA Designer assumes a discrete, 4-phase clock (as discussed in [46]) while M-AQUINAS assumes the wave-based clocking scheme that is more suitable for a molecular implementation. These simulations methods are good in that they incorporate the physics of a clock which adds a helpful time component to the simulation. We have used M-AQUINAS to generate some of the results to be discussed here.

That said, M-AQUINAS and QCA Designer also only allow a QCA device to be in one of two states (a binary 1 or 0). With a molecular implementation, any combination of the four redox sites may be reduced or oxidized – resulting in six possible active states (see [9]). Because states 2-5 are higher in energy than states 0 and 1, it is unlikely that a single, isolated QCA cell would settle into any of the four degenerate cases. However, when many QCA cells are placed in close proximity to one another (i.e. to form a circuit), it is possible that a locally higher energy state might be favored because it reduces the energy of the entire QCA circuit – especially in the presence of defects. If this happens in the cell that serves as the output of the circuit, it will be impossible to tell whether or not the output is a 1 or a 0.

1) Statistical Mechanics Simulations: To study these effects in the context of circuits, we have extended the work of [47] (developed by chemists working to manufacture QCA devices and scaffolds) to consider charge-balanced, 4-dot QCA molecules and the 6 possible active states that would be associated with the synthesized 4-dot QCA molecule shown in [29]. While beyond the scope of this paper, a more detailed discussion of our simulation methodology can be found in [18] and [47]. We know of no other work that leverages a 6 state model.

We have used both a statistical mechanical analysis (and M-AQUINAS) to verify whether or not the core building block of our PLA (Fig. 4c) will function as intended. For both simulations, we assume molecular QCA devices with 1 nm between redox sites and a 2 nm center-to-center cell spacing. The 1.0 nm distance was selected as it closely corresponds to the distance between redox sites of the Creutz-Taube (CT) ion (or 0.9 nm). The physically realized CT ion has two electrochemically reversible and chemically stable redox centers, and a strong coupling between them to allow for tunneling of the charge that will represent a binary 1 or 0 [48]. For center-to-center spacing, we do not target any of the specific patterning mechanisms discussed earlier. In short, exact center-to-center spacings have not yet been established for any potential patterning mechanism. For this reason, we have chosen a 1 nm edge-to-edge spacing (on the order of what we would like, but also a distance that is plausible as seen in the discussion of potential DNA scaffolds) which leads to the aforementioned 2.0 nm center-to-center spacing (see [18] for more detail).

Our statistical mechanical analysis considered the PLA cell in both 'logic' and 'wire' modes as well as all possible input combinations for `MINTERM_IN` and `LITERAL_IN`. Simulation results indicate that our PLA core should function as intended. All tests confirmed ground states matching the desired behavior with high probability – which should only increase in the presence of a clock (see [49])[3].

2) M-AQUINAS Simulations: For charge-based, molecular QCA devices, stray charges, cell shifts, cell rotations, and missing cells might all cause a circuit construct to produce the wrong result. While (experimentally) no group has attempted to deterministically place individual QCA molecules, there is more than ample evidence to justify these assumptions. As one example, [34] looks at attaching gold nano particles to a DNA scaffold. The AFM imagery in [34] illustrates that some particles are either missing or misaligned. While these problems could be caused by perturbation from the AFM tip, incomplete hybridization between the nanoparticles and the DNA scaffold is the most likely source of missing particles,

[3]Additionally, these results consider no device-level redundancy. As will be seen next, "thicker" wires and gates will further increase the probability of seeing a correct output.

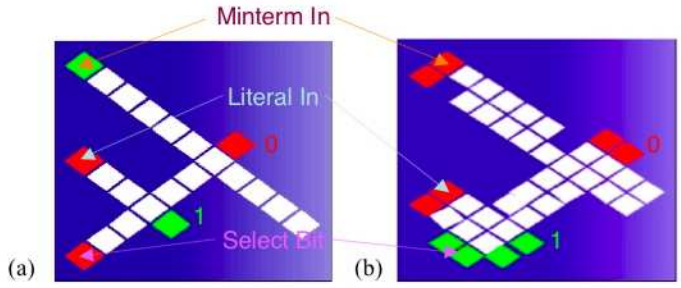

Fig. 7. (a) PLA (1x), (b) Example design with 20% defects and device-level redundancy. Simulations show that this design functions correctly.

and the relatively floppy duplex DNA used to attach the particles to the DNA scaffold is the most likely source of particle misalignment. Another defect of particular concern is stray charge, which could come from a scaffolding molecule (i.e. the sugar-phosphate backbone of DNA) or from the substrate surface (i.e. the ionizable silanol groups on silicon native oxide).

While clearly PLAs offer defect tolerance at the architectural-level, with QCA, redundancy at the *device-level* can provide a measure of defect tolerance too. Device-level redundancy in wires and gates was considered in the context of the above defects (minus stray charge) in [50]. Simulations show that in a wire one cell wide, if one cell flips to the wrong state, the remaining cells in the wire will continue to propagate the wrong data. If there are only rotation defects, a wire once cell wide functions properly when the rotation of one cell is $< 21^o$. If rotations and cell shifts are both possible, tolerance to rotation decreases. With only cell shifts, a wire functions correctly when the displacement factor is ≤ 35.

Adding redundancy can improve defect tolerance. For example, with a wire that is three cells wide, the tolerance to rotational disorder improves by $\pm 5^o$ even *with* a displacement factor of 0.3. Additionally, if one (or even several) of the cells in the QCA array flip to the wrong state, the rest of the cells copy the right information to the output.

While a detailed discussion of fault models and yield estimates for our PLA structure is beyond the scope of this paper, we do briefly describe a set of MAQUINAS simulations to discuss how interconnect in our design might be affected by defects as well as device-level redundancy. Specifically we have laid out a cell core with parts that are 2 cells wide (reflected in Fig. 7b) instead of 1 cell wide (i.e. Fig. 7a), and randomly inserted defects into this design. We then simulated defective designs with M-AQUINAS. Preliminary results indicate that with overall defect rates ranging from 5-20% (divided evenly between shifted cells, missing cells, cells stuck in a polarization – which could be caused by a stray charge, and rotated cells), an individual PLA cell with 2 cell thick parts functioned correctly 34 out of 44 times. <**Update with Zack's new numbers? + 1 cell results?**>

In the remaining 10 cases, we saw behavior equivalent to a "stuck at" fault. This could be overcome by simply not

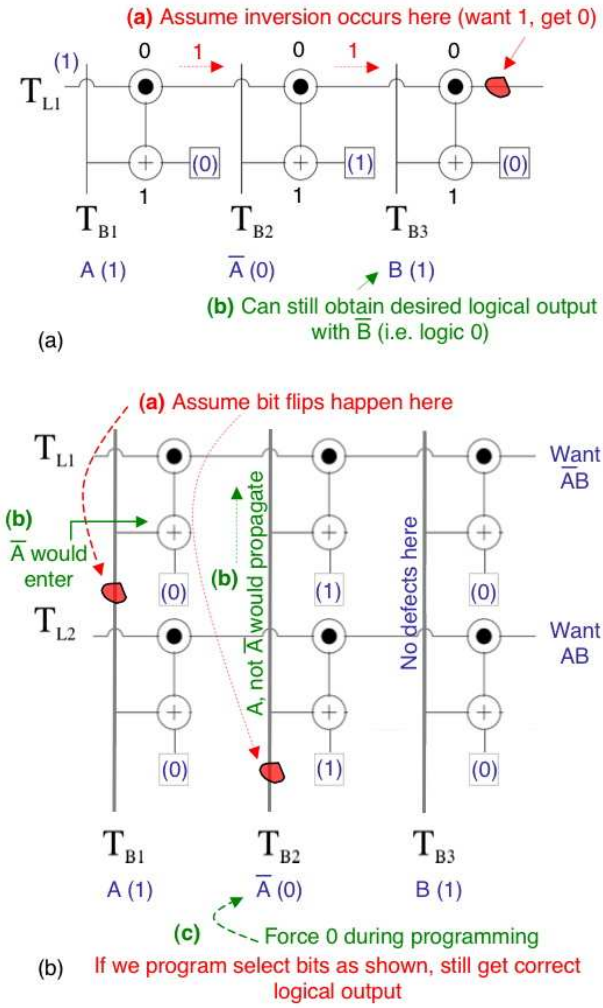

Fig. 8. Examples of how wires with inversion faults can still be used in a QCA-based PLA. (Broken down into steps (a)-(b) in Fig. 8a and steps (a)-(c) in Fig. 8b.

using a given row or column of a PLA. The other fault that appeared was an inversion. It is important to note however that when a bit was inverted, it was *consistently* inverted regardless of input. This is important because if this defect is readily detectable, we would still be able to ensure that we saw the desired logical output of the PLA simply by changing our logic function to account for a flip. (An example is shown in Fig. 8.) While this is very preliminary work and needs further study, it appears that in at least come cases, a very defective QCA wire is not necessarily broken and is still usable. Finally, while these simulations assumed dimensions associated with a molecular implementation, at the logic and systems level, all implementations of QCA could avoid this potential fault in a similar manner.

IV. Maintaining Select Bit State

If we want our PLA to repeatedly perform the same logical function, we will have to indefinitely maintain the state of a select bit. How this might be accomplished is implementation

dependent and will be discussed for charge-based QCA in Sec.IV-A and for magnetic QCA in Sec.IV-B.

A. Charge-based QCA

1) Clocking Requirements: As seen in Sec. II-C a QCA device can be in at most one of three states – a *null state* where the device holds no information and does not participate in computation, an *active state* where the device should be in a configuration that represents a binary 1 or 0, or a *switching state* that occurs when a device is transitioning between the *null* and *active* states. Which of these three states the device is in is determined in part by the electric field applied to a device at a given point and time (see Fig. 2a).

It is important to note that if a negative electric field is *continuously* applied to a charge-based QCA device, it should remain in an active state[4]. More importantly, an active cell should also retain its *binary* state provided that the clock keeps the device(s) in an active state. What this means is that we should be able to maintain select bit state indefinitely – but to do so we must be able to control the fields associated with the relevant bits. In other words, we must clock select bits and logic separately.

While we will want to independently clock select bits (or at least groups of select bits) as it is not desirable from an implementation standpoint to have to independently clock the logic needed for each PLA cell (i.e. each crosspoint). Clocking each individually would require that each PLA cell have at least four clock wires associated with it – each with a different signal with a $\pi/2$ phase shift applied to it. In short, signal routing for the underlying clock structure would become quite complex. Additionally, the QCA logic associated with each cell should have much smaller feature sizes than the smallest metal wire – even with end-of-the-roadmap CMOS pitches. The area requirements for QCA logic would have to be scaled up significantly in order to match the pitch/size of the clock wires. This would only increase latency and adversely affect potential density gains.

Fortunately, PLA cells should not have to be clocked individually. Furthermore, it also appears that our design facilitates the requirements that (a) wires that keep select bits active generate the required electric fields, and (b) these fields are not distorted by the more global clock that moves data in the AND and OR planes from the terminals. We will illustrate this via experimental data next.

2) Experiments: Studies of clock structures for QCA circuits are extremely limited. Existing work consists of a study performed by Hennessy and Lent that investigated electric fields produced by evenly spaced point charges [38]. This work demonstrated that electric fields with the desired magnitude and distributions could be generated by applying the proper charges on infinitesimally small wires with finite sizes. They also demonstrated that these results should apply to wires with finite sizes too. However, to the best of our knowledge, no experimental results have been published regarding this, and there is no experimental study on the possibility of supplying the more complex electric field distribution for our proposed PLA structure.

[4]The field might also be positive depending on whether or not electrons or holes are used to represent binary state.

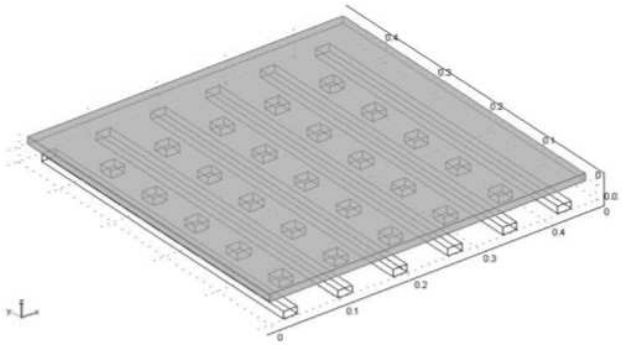

Fig. 9. FEMLAB experimental setup.

To validate whether or not our clocking requirements will also ensure that the state of programmed select bits is kept indefinitely – and is not reset by normal operation – we have conducted a number of experiments to investigate how electric fields might be distributed by lithographically defined metal wires. Fig. 9 illustrates what the lithographically defined, electric-field generating wires might look like. The small square shapes (*select wires*) represent clock wires that keep select bits active. The larger, rectangular shapes (*logic wires*) represent clock generating circuitry that would generate the electric fields necessary to induce dataflow in the rows and columns of PLA logic cells.

While we have used different sizes for the wires, as well as different separations between them, we discuss results associated with the minimum metal pitch according to the 2004 ITRS for the 2018 technology node (40 nm) [1]. A metal plane was added 20 nm above the logic and select wire plane and serves as the ground plane that directs electric fields [38]. It is envisioned that the QCA-based PLA would be sandwiched between the two metal planes. A similar structure has been experimentally demonstrated in the molecular switching experiments discussed in [30] (a ground plane should not be necessary in a magnetic implementation will may not be necessary in other charge-based implementations. We used silicon dioxide as the dielectric with a relative permittivity of 3.9 and gold as the metal for the wires. We note that at most 2 layers of metal should be needed to build this clock structure.

To study the electric field distribution, two distinct clocking signals were applied. The select wires were driven by a constant, positive voltage while the logic wires were driven by a quasi-adiabatic 4-phase clock signal [51]. This setup emulates how the PLA would operate in logic mode. FEMLAB software [52] was used to obtain the electric field distribution at the plane where QCA cells reside. As only the component of the electric field that is perpendicular to the device plane affects the states of the QCA devices [38], we only examine the distribution of the vertical electric field (i.e., the y-

component). We applied a 5 volt signal to the select wires, and applied five different voltages (-5 Volts, -2.5 Volts, 0 Volt, 2.5 Volts, and 5 Volts) to the quasiadiabatic signal (logic wires). The selection of 5 Volts as the maximum voltage is based on the assumption that the intensity of 2 Mv/cm is sufficient to maintain the states of QCA devices. Based on conversations with chemists, this is a realistic assumption for molecular QCA devices [53].

Simulation results show that the intensity of the vertical electric field 10 nm above the logic/select wire plane is unchanged. The field intensity above the select wires exceeds 2Mv/cm, which indicates that QCA cells on top of these wires should stay in the active/bistable state [38]. If these cells are not driven, they should maintain their previous logic values. This is precisely what is required to guarantee the correct functionality of our PLA structure for charge-based devices. Our simulations show that the field distributions remain essentially the same regardless of the sizes of the wires and distances between them – implying that this clocking scheme should scale to larger charge-based devices too.

What must still be determined is the granularity required for the logic wires. For example, it is theoretically possible to treat the AND plane as one large clocked region – and four larger logic wires might be used to induce dataflow through this group of QCA devices. Alternatively, we may want and/or need finer-grained logic wires. Whether we take a more global (coarse grain) or local (fine grain) approach to clocking will ultimately be determined by a number of variables. The magnitude and distribution of the electric field required, the number of devices that can be active simultaneously (if there are too many active devices, energy from switching activity can induce mistake states), and the technologies used to implement the circuit will all have an effect. Ultimately, we need to be able to move data inputs through the QCA logic – inputs enter through the AND plane, outputs of the AND plane and external inputs enter the OR plane, and outputs of the OR plane are either recycled through the PLA or to other logic – that comprises our PLA structure while maintaining the set values of the select bits. As mentioned, our FEMLAB simulations have shown that the latter is possible regardless of wire sizes. In future work, we will investigate what granularity will be necessary.

B. Magnetic QCA

1) Clocking Requirements: As seen in Fig. 2, as with metal-dot and molecular implementations, a "clock" will be used to induce dataflow in magnetic QCA devices too. However, ensuring that a select bit keeps its state indefinitely is (at least conceptually) easier with a magnetic implementation. We briefly outline one likely mechanism for accomplishing this below.

Sec.II-C.2 reported that after a group of nano-magnets has been relaxed (i.e. after the field in the x-direction has been removed), they have a tendency to settle into some new state – and must be relaxed again before they can change state again (when a new computation occurs). In other words, nanomagnets do not have to be active to maintain their state.

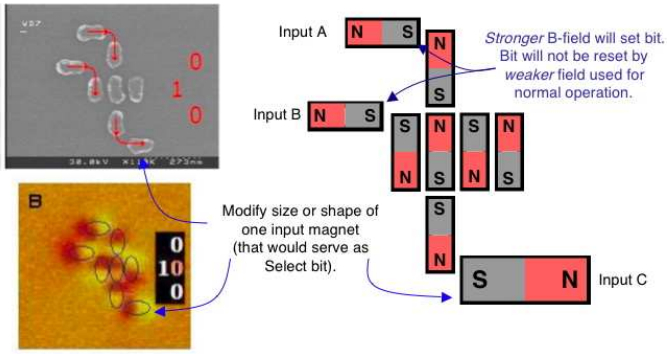

Fig. 10. Possible mechanism for programming a magnetic select bit.

2) Experiments: We intend to take advantage of this property to indefinitely maintain the state of a given select bit – specifically by modifying the size and/or shape of the nanomagnet used to represent it. For example, the left side of Fig. 10 illustrates a fabricated magnetic QCA (inverting) majority gate (as reported in [13]) as well as an MFM image showing the state of the device cell after one particular input combination is applied. By varying the size of one of the inputs, we should be able to maintain select bit state indefinitely. A stronger magnetic field will set the bit, but the bit will not be reset by a weaker field when used for normal operation. We are currently conducting micromagnetic simulations [54] to determine the ideal size and shape of this magnet and are also working with physical scientists to fabricate this structure upon completion (we should be able to experimentally verify this with external magnetic fields in lieu of an implemented clock). We note that this is an excellent example of using design to define experimental milestones that must be reached in order to achieve performance wins at the application-level. <**Mike Putney may have already shown this. I just need to touch base with him.**>

V. Operating a PLA

In Sec. III we showed that by deterministically setting select bits in our PLA structure, we could generate and evaluate arbitrary sum-of-products expressions. In Sec. IV, we illustrated how we could *maintain* state associated with each select-bit. Here, we show ways to initially program select-bit state. We also discuss how this design would function during normal operation. As in the last two sections, we will discuss programming in an implementation dependent fashion. The methods discussed in this section should be viewed as possible ways to physically accomplish programming – which method will ultimately work will be determined by more physics-level simulations and experimental verification. Nevertheless, we believe that it is (a) important to show viable methodologies that reflect implementation reality, and (b) that these methods will again form the basis for both future simulations and experimental prototypes.

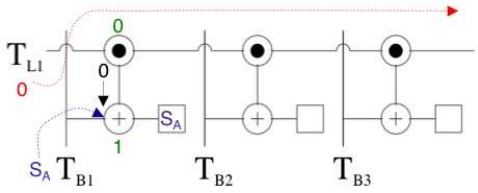

Fig. 11. Programming a PLA made from lithographically-defined or molecular QCA devices (first pass).

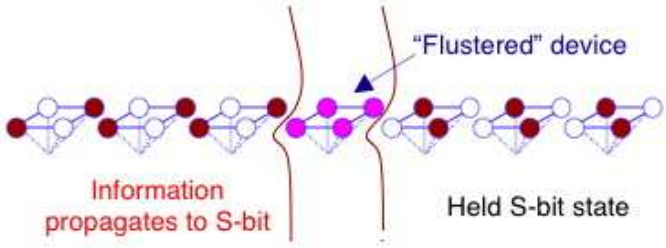

Fig. 12. Driving a value into a *held* select bit (S-bit).

A. Programming Metal-Dot and Molecular QCA

Conceptually, setting the state of a select bit is actually quite simple. Referring to Fig. 11, assume we wanted to program the select bit state for row T_{L1} – one row of an AND plane. If we input a 0 at T_{L1}, the desired select bit values at T_{B1}-T_{B3}, and clock this circuit from left-to-right, data should be driven into the select bit locations (for example S_A). (The OR gate in each column will have input values of a 0 and 1 with the bit input at T_{Bx} deciding the majority vote.) As mentioned, if the select wires remain active, the QCA devices should also stay in an active state – and keep their binary state as well.

This *programming* procedure is directly analogous to the dataflow requirements for operating this PLA design in logic mode (assuming that all select bits have been previously set). T_{L1} will have an initial input of 1 (this corresponds to MINTERM_IN in Fig. 4c) and T_{B1}-T_{B3} would be set with our desired data inputs (LITERAL_IN in Fig. 4c). The only difference between a programming procedure and the PLA operating in logic mode is that during normal operation, select bits would have already been set and would need to act as *drivers*. During *programming*, they need to become active and will be driven.

One potential problem with the programming procedure just discussed is that when the PLA operates in logic mode, data is driven from left-to-right. With the layout shown in Fig. 11, we would also be driving data into active QCA devices that hold select bit state. Of concern would be whether or not this select bit state would be over-written by the new information being driven. Work by Isaksen, et. al. [55] shows that this should not be the case – and if new information is driven, the *active* devices should not lose their state. (If the two driven values are different, at some middle point in the wire, a given cell should become frustrated and/or unpolarized as illustrated in the cartoon in Fig. 12). Thus, to program multiple rows, we could successively apply our desired values to the columns of the PLA (i.e. T_{B1}-T_{B3} in Fig. 13, raise the select bit pads for that row as data is driven, and repeat this process for each row of the PLA.

Still, operating this configuration in logic mode would prove to be problematic. If our PLA was clocked from left-to-right, the select bit would never have a chance to participate in the computation of the AND and OR functions associated with each cell – as the other two input bits would be processed before the the select bit path to the logic ever becomes active (both of these characteristics have been verified with the MAQUINAS simulator).

An alternative would be to move the select bit below the OR gate as shown in Fig. 13. Simulations show that if clocked from left-to-right during programming, select bit values to be programmed will fan out to the proper location. However, this method could also encounter problems during logical operation. If the select bit is placed close enough to the OR gate, the circuit will function as intended. However, if the bit is farther away (as it realistically will be due to the constraints of lithography) it will again not have time to influence the computation of the OR gate. (Again, these results have been verified by simulation).

The above discussion might make it appear that programming select bit state is tied to race conditions and/or complex timing. However, we have devised a method that not only functions logically, but also considers the constraints of the QCA device architecture and the realistic fabrication constructs for any necessary support circuitry – namely the directional dataflow enforced by the clock.

Perhaps the easiest way to to ensure that select bits are driven when programming our PLA – and drive logic when the PLA is operating in logic mode – is to clock our PLA *diagonally* (see Figs. 7 and 13). Referring to Fig. 13, if we clock our structure from top-left to bottom-right during programming, the output of the OR gate terminates in the groups of QCA cells that represent select bit state. If we clock our PLA structure from bottom-left to top-right when the PLA is operating in logic mode, there is no driving path of QCA devices into the select bit – and there is not chance of even the situation depicted in Fig. 12 occurring. Another advantage is that if our PLA cells are roughly square, simple trigonometry tells us that both inputs to AND and OR gates will almost assuredly arrive at the same time. (Among other things, this should allow for faster clock rates as the critical paths within each cell will be roughly equal).

This method has only one minor drawback: a slightly more complex clock structure might be required. In the worst case, another layer of clock wires would be needed so that data could be moved in two different directions. This would require an extra layer of metal – and one layer would be physically farther away from the surface where QCA devices sit. We would need to ensure that the electric field at the surface would still be sufficiently strong.

Finally, we briefly comment on the expected area of our

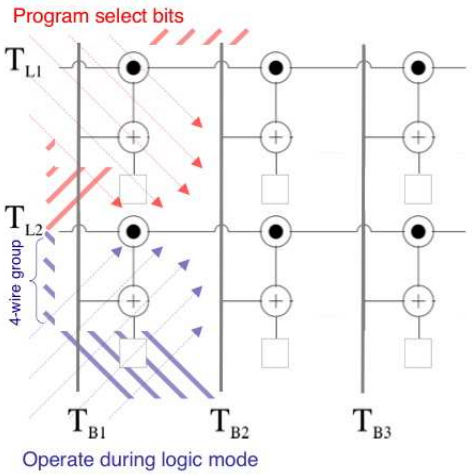

Fig. 13. Clocking a PLA diagonally.

design. By now, it should be fairly obvious that area will ultimately be determined by the granularity of the underlying lithography. Assuming that we clock the PLA diagonally, in the worst case we will require one 4-wire group per diagonal. On the edge of the PLA, this would require an area of $5\times$ (*wire pitch*). (The constant 5 comes from the fact that there are four clock wires per diagonal in addition to a select bit pad.) Thus, an upper bound for the area of an AND or OR plane would be: $(5 \times pitch \times rows) \times (5 \times pitch \times cols)$ where *rows* and *cols* are the number of rows and columns in the given plane. However, as mentioned earlier, we may not need this level of granularity. A "best case" would only have to account for the select pads – especially if the coarser grained clock wires could be fabricated in an additional layer of metal. With this option, the area would simply be $(pitch \times rows) \times (pitch \times cols)$. For a molecular implementation, with no consideration for yield and assuming the wire pitches discussed in Sec. IV-A.2, a 60×60 AND plane would require between 5.76×10^6 nm^2 and 144×10^6 nm^2.

A much more careful analysis of field strengths required to switch charge-based devices and field distributions produced by lithographically defined wires is obviously required (and beyond the scope of this paper). Nevertheless, the numbers above do provide first-order estimates for lower and upper bounds on area.

B. Programming Magnetic QCA

Programming the magnetic QCA device(s) that will hold select-bit state should be much more straight forward. (We will essentially need to set input C in Fig. 10). While we could devise a programming sequence similar to that discussed above, we could also leverage an electrical input structure similar to that used to read/write a bit in MRAM, or different shaped and/or positioned magnets as discussed in Sec. IV-B.2. Considering these multiple options within the context of implementation constraints and the PLA design is the subject of future work.

VI. Conclusions and Future Work

In conclusion, we have presented a novel, QCA-based PLA structure and have demonstrated how it operates as well as how it can be reprogrammed. All of this has been done within the context of realistic fabrication constrains (leveraging simple QCA design constructs and regular, replicable clock structures) for all implementations of QCA.

In any technology, a functional device or logic gate in isolation is useless. Functional units, interconnect, and efficient information flow within the constraints of a given device architecture are ultimately necessary. By considering how QCA devices must interact in order to achieve meaningful computation, we will also help to characterize and quantify potential performance wins. Additional insight into system-level architecture and potential performance will ultimately determine the viability of a new technology and how (or if) device-level research should proceed at all.

This work will serve as a "jumping off point" for more detailed, implementation specific studies. Work is already ongoing for a molecular implementation – where this design will serve as a fabrication target for physical scientists trying to place individual components in computationally interesting patterns. Similarly, we are actively targeting the implementation of this design with a magnetic implementation as well. One immediate application of such a technology would be to use the devices in spacecraft or satellites. Reprogrammable devices on spacecraft offer the same advantages as reprogrammable hardware on the ground (i.e. they can more easily cope with changing interface standards and provide upgrades during the system life-cycle). There is the added benefit that magnetic devices are resilient to the effects of radiation. As the proposed clock structure will most likely result in inherently pipelined dataflow, systolic architectures [56] are also being explored. Any application that requires low power dissipation and non-volatility could benefit from magnetic QCA devices.

In future work, we will present implementation dependent fault models currently being developed, defect detection schemes, and yield and performance estimates for relevant benchmarks.

References

[1] I. T. R. for Semiconductors, "International technology roadmap for semiconductors," http://www.itrs.net/Common/2004Update/2004Update.htm, December 2005, 2005 ITRS Update.

[2] P. Tougaw and C. Lent, "Logical devices implemented using quantum cellular automata," J. of App. Phys., vol. 75, p. 1818, 1994.

[3] I. Amlani, A. Orlov, G. Snider, and C. Lent, "Demonstration of a functional. quantum-dot cellular automata cell," J. Vac. Sci. Tech. B., vol. 16, pp. 3795–99, 1998.

[4] C. Lent, G. Snider, G. Bernstein, W. Porod, A. Orlov, M. Lieberman, T. Fehlner, and a. P. K. M.T. Niemier, "Quantum-dot cellular automata," Chapter in Electron Transport in Quantum Dots, vol. Johathan P. Bird (ed.), Kluwer Academic Publishers, pp. 397–433, 2003.

[5] G. Snider, A. Orlov, I. Amlani, G. Bernstein, C. Lent, J. Merz, and W. Porod, "Quantum-dot cellular automata: Line and majority gate logic," Jpn. J. of Applied Physics, vol. 38, pp. 7227–7229, 1999.

[6] J. Timler and C. Lent, "Power gain and dissipation in quantum-dot cellular automata," J. of App. Phys., vol. 91, pp. 823–831, 2002.

[7] P. Barbara, T. Meyer, and M. Ratner, "Contemporary issues in electron transfer research'," J. Phys. Chem., vol. 100, pp. 13 148–13 168, 1996.

[8] P. Perez-Tejeda, F. Sanchez-Burgos, and M. Galan, "Calculation of rate constants from uv-vis spectroscopic data: an application of the marcus-hush model," J. of Molec. Struct. (Thermochem), vol. 371, pp. 153–160, 1996.

[9] A. Chaudhary, D. Chen, X. Hu, M. Niemier, R. Ravichandran, and K. Whitton, "Eliminating wire crossings for molecular quantum-dot cellular automata implementation," in In Proc. of ICCAD, 2005, pp. 565–571.

[10] M. Niemier, "The effects of a new technology on the design, organization, and architectures of computing systems," Ph.D. Dissertation, January, 2004.

[11] R. Cowburn and M. Welland, "Room temperature magnetic quantum cellular automata," Science, vol. 287(5457), pp. 1466–1468, February, 2000.

[12] G. Bernstein, A. Imre, V. Metlushko, A. Orlov, L. Zhou, G. C. L. Ji, and W. Porod, "Magnetic qca systems," Microelectronics Journal, vol. 36, pp. 619–624, 2005.

[13] A. Imre, G. Csaba, L. Ji, A. Orlov, G. Bernstein, and W. Porod, "Majority logic gate for magnetic quantum-dot cellular automata," Science, vol. 311, No. 5758, pp. 205–208, January 13, 2006.

[14] A. Imre, "Experimental study of nanomagnets for magnetic quantum-dot cellular automata (mqca) logic applications," Disseration, vol. University of Notre Dame, April 2005.

[15] A. DeHon and M. Wilson, "Nano-wire based sublithographic programmable logic arrays," Proceedings of the International Symposium on Field-Programmable Gate Arrays, vol. February 22-24, pp. 123–132, 2004.

[16] D. Strukov and K. Likharev, "Cmol fpga: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices," Nanotechnology, vol. 16, pp. 888–900, 2005.

[17] G. Snider, P. J. Kuekes, T. Hogg, and R. S. Williams, "Nanoelectronic architectures," Applied Physics A, vol. 80, pp. 1183–1196, 20056.

[18] M. Niemier, M. Crocker, X. S. Hu, and M. Lieberman, "Using cad to shape experiments in molecular qca," Int. Conf. on Comp. Aided Design, pp. 907–914, November 8, 2006.

[19] C. Lent and P. Tougaw, "A device architecture for computing with quantum dots," Proc. of the IEEE, vol. 85, p. 541, 1997.

[20] I. Amlani, A. Orlov, G. Toth, G. Bernstein, C. Lent, and G. Snider, "Digital logic gate using quantum-dot cellular automata," Science, vol. 284, p. 1999, 289-291.

[21] R. Kummamuru, J. Timler, G. Toth, C. Lent, R. Ramasubramaniam, A. Orlov, and G. Bernstein, "Power gain in a quantum-dot cellular automata latch," Applied Physics Letters, vol. 81, pp. 1332–1334, 2002.

[22] T. Vandervelde, P. Kumar, T. Kobayashi, J. Gray, T. Pernell, J. Floro, R. Hull, and J. Bean, "Growth of quantum fortress structures in Si1-xGex/Si via combinatorial deposition," Appl. Phys. Lett., vol. 83, pp. 5205–7, 2003.

[23] M. Mitic, M. Cassidy, K. Petersson, R. Starrett, E. Gauja, R. Brenner, R. Clark, A. Dzurak, C. Yang, and D. Jamieson, "Demonstration of a silicon-based quantum cellular automata cell," Applied Physics Letters, vol. 89, July 5, 2006.

[24] C. Smith, S. Gardelis, A. Rushforth, R. Crook, J. Cooper, D. Ritchie, E. Linfield, Y. Jin, and M. Pepper, "Realization of quantum-dot cellular automata using semiconductor quantum dots," Superlattices and Microstructures, vol. 34, pp. 195–203, 2003.

[25] A. Fujiwara, S. Horiguchi, M. Nagase, and Y. Takahashi, "Threshold voltage of si single-electron transistor," Japanese Journal of Applied Physics Part 1-Regular Papers Short Notes and Review Papers, vol. 42, pp. 2429–33, 2003.

[26] G. Snider, "Personal communication," 2006.

[27] T. dimensional Architectures for Donor-based Quantum Computing, "L.c.l. hollenberg and a.d. greentree and a.g. fowler, and c.j. wellard," Physical Review B, vol. 74, pp. –, 2006.

[28] L. Hollenberg, C. Wellard, C. Pakes, and A. Fowler, "Single-spin readout for buired dopant semiconductor qubits," Physical Review B, vol. 69, pp. –, 2004.

[29] J. Jiao, G. Long, F. Grandjean, A. Beatty, and T. 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., vol. 125, pp. 1522–1523, 2003.

[30] H. Qi, S. Sharma, Z. Li, G. Snider, A. Orlov, C. Lent, and T. Fehlner, "Molecular quantum cellular automata cells. electric field driven switching of a silicon surface bound array of vertically oriented two-dot molecular quantum cellular automata," J. Am. Chem. Soc., vol. 125, pp. 15 250–15 259, 2003.

[31] E. Winfree, F. Liu, L. Wenzler, and N. Seeman, "Design and self-assembly of two-dimensional dna crystals," Nature, vol. 394, pp. 539–544, 1998.

[32] W. Hu, K. Sarveswaran, M. Lieberman, and G. H. Bernstein, "M," IEEE Trans. on Nano., vol. 4, pp. 312–316, 2005.

[33] D. Hurley and Y. Tor, "Ru(ii) and os(ii) nucleosides and oligonucleotides: Synthesis and properties'," J. Am. Chem. Soc., vol. 124, pp. 3749–3762, 2002.

[34] J. D. Le, Y. Pinto, N. Seeman, K. Musier-Forsyth, T. Taton, and R. Kiehl, "Dna-templated self-assembly of metallic nanocomponent arrays on a surface," Nano Lett., vol. 4(12), pp. 2343–2347, 2004.

[35] M. de Wild, S. Berner, H. Suzuki, H. Yanagi, D. Schlettwein, S. Ivan, A. Baratoff, H.-J. Guentherodt, and T. Jung, "A novel route to molecular self-assembly: Self-intermixed monolayer phases," ChemPhysChem, vol. 10, pp. 881–885, 2002.

[36] M. Stoykovich, M. Muller, S. Kim, H. Solak, E. E. J. J. de Pablo, and P. Nealey, "Directed assembly of block copolymer blends into nonregular device-oriented structures," Science, vol. 308(5727), pp. 1442–46, June 3, 2005.

[37] L. Demers, D. G. S. Park, and et al., "Direct patterning of modified oligonucleotides on metals and insulators by dip-pen nanolithography," Science, vol. 296(5574), pp. 1836–1838, June 7, 2002.

[38] K. Hennessy and C. Lent, "Clocking of molecular quantum-dot cellular automata," J. of Vac. Sci. & Tech. B, vol. 19(5), pp. 1752–55, Sep-Oct 2001.

[39] C. Graunke, D. Wheeler, D.Tougaw, and J. Will, "Implementation of a crossbar network using quantum-dot cellular automata," IEEE Trans. on Nano., vol. 4(4), July, 2004.

[40] M. Niemier and P. Kogge, "The '4-diamond' circuit - a minimally complex nano-scale computational building block in qca," In Proceedings of the IEEE Computer Society Symposium on VLSI, vol. IEEE Computer Society Press, Lafayette, LA, pp. 3–10, February 2004.

[41] J. Huang, M. Momenzadeh, M. Tahoori, and F. Lombardi, "Design and characterization of an and-or-inverter (aoi) gate for qca implementation," ACM Great Lakes Symposium on VLSI, pp. 426–29, 2004.

[42] K. McElvain, "Lgsynth benchmark set: Version 4.0," http://www.cbl.ncsu.edu:16090/benchmarks/LGSynth93, May 1993, lGSynth93 reference from DeHon.

[43] U. Group, http://ic.eecs.berkeley.edu/pub/Espresso/ espresso-book-examples.tar.gz, June 1993, espresso reference from DeHon.

[44] S. Hu, M. Crocker, M. Niemier, M. Yan, and G. Bernstein, "Plas in quantum-dot cellular automata," to appear in Proceedings of International Symposium on VLSI, March 2-3, 2006.

[45] K. Walus, T. Dysart, G. Jullien, and R. Budiman, "Qca designer: A rapid design and simulation tool for quantum-dot cellular automata," IEEE Trans. on Nano., vol. 3(1), pp. 26–31, 2004.

[46] E. P. Blair, "Tools for the design and simulation of clocked molecular quantum-dot cellular automata circuit," M.S. Thesis, vol. Univ. of Notre Dame, 2003.

[47] Y. Wang and M. Lieberman, "Thermodynamic behavior of molecular-scale quantum-dot automata (qca) wires and logic devices," IEEE Trans. on Nano., vol. 3(3), pp. 368–376, Sept. 2004.

[48] G. Tom and H. Taube, "The mixed valence state based on μ-cyanogen-bis(pentaammineruthenium)," J. of Am. Chem. Soc., vol. 97, pp. 5310–5311, 1975.

[49] M. Niemier, M. Crocker, X. S. Hu, M. Lieberman, T. Dysart, Z. Capozzi, and P. Sadarangani, "Using dna as a circuitboard for a molecular qca pla," Proceedings of Foundations of Nanoscience (FNANO), April 23-27, 2006.

[50] M. Liu, "Robustness and power dissipation in quantum-dot cellular automata," Ph.D. Dissertation, vol. University of Notre Dame, 2006.

[51] G. Toth and C. S. Lent, "Quasiadiabatic switching for metal-island quantum-dot cellular automata," J. Appl. Phys., vol. 85, p. 2977, 1999.

[52] I. COMSOL, "Femlab 3.0 software," http://www.comsol.com, Oct 1984, femlab software reference.

[53] M. Lieberman, "Personal communication," 2006.

[54] M. Donahue and D. Porter, "Oommf user's guide, version 1.0, interagency report nistir 6367," http://math.nist.gov/oommf.

[55] C. Lent, B. Isaksen, and M. Lieberman, "Molecular quantum-dot cellular automata," J. Am.. Chem. Soc., vol. 125(4), pp. 1056–1063, 2003.

[56] H. Kung, "Why systolic architectures," IEEE Computer, pp. 37–46, 1982.