Modeling and Evaluating Errors due to Random Clock Shifts in Quantum-dot Cellular Automata Circuits

F. Karim[†], M. Ottavi[‡,1], H. Hashempour[‡], V. Vankamamidi[‡,2], K. Walus[†], A. Ivanov[†] and F. Lombardi[‡]

[†]Department of ECE
University of British Columbia
Vancouver, BC, V6T 1Z4, Canada

[‡]Department of ECE
Northeastern University
Boston, MA 02115, USA

Abstract

This paper analyzes the effect of random phase shifts in the underlying clock signals on the operation of several basic Quantum-dot Cellular Automata (QCA) building blocks. Such phase shifts can result from manufacturing variations or from uneven path lengths in the clocking network. We perform numerical simulations of basic building blocks using two different simulation engines available in the QCADesigner tool. We assume that the phase shifts are characterized by a Gaussian distribution with a mean value of $i\frac{\pi}{2}$, where i is the clock number and a standard deviation, σ, which is varied in each simulation. Our results indicate that the sensitivity of building blocks to phase shifts depends primarily on the layout while the performance of all building blocks starts to drop once the phase shifts in the clocking network are characterized by a standard deviation, $\sigma \geq 4°$. A full adder was simulated to analyze the operation of a circuit featuring a combination of the building blocks considered here. Results are consistent with expectations and demonstrate that the C_{out} output of the full adder is better able to withstand the phase shifts in the clocking network than the Sum output which features a larger combination of the simulated building blocks.

Key Words: Quantum-dot Cellular Automata (QCA), clocked QCA, emerging nanotechnologies, phase shift.

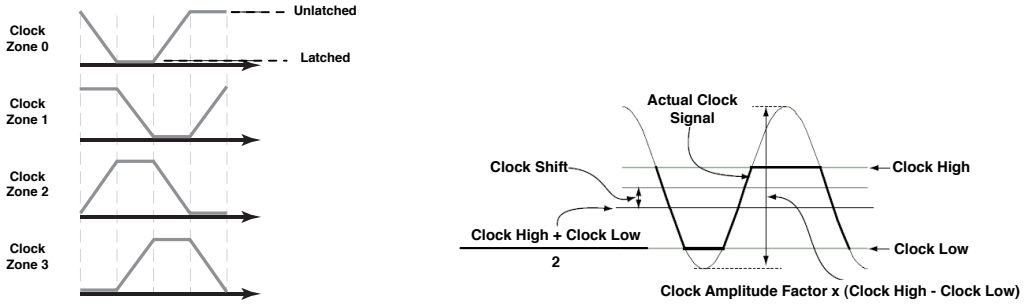
1: Introduction

The feature size of conventional field-effect transistors (FET) has been consistently decreasing in an attempt to increase device density and operating frequency of computing circuits. Today, transistors with gate lengths below 50nm are fabricated and exhibit excellent electrical characteristics [1]. This trend has resulted in an exponential growth in the

[1]Currently with Advanced Micro Devices Inc.
[2]Currently with EMC Corporation

integration level of electronic chips captured through Moore's law [2]. The architectural and fundamental limits to this growth have been revised several times to account for new technologies. Novel technological concepts based on nano-devices and nano-electronics are projected to play a significant role in future systems [3, 4]. Quantum-dot cellular automata (QCA) is a paradigm in which an array of cells, each electrostatically interacting with its neighbors, is employed in a locally interconnected manner to implement general purpose digital circuits [5]. Research into a physical realization of QCA using coupled quantum dots [6–12], nanomagnets [13–18], or various molecular structures [19–26] is ongoing. In practice, a system to clock individual cells is required for QCA operation, this involves additional wiring for clock distribution and field generation. While defect tolerance and testing of QCA circuits has been addressed in the past [27–29], little attention has been paid to variations in the required clock network and its implications on QCA circuit operation. A comprehensive review of QCA can be found in [30–32].

We employ "zone clocking" in this paper, where all the cells in a design are grouped into one of four available clocking zones; each cell in a particular clocking zone is connected to one of the four available phases of the QCA clock shown in Figure 1(a).



(a) Four phases of the QCA clock

(b) QCADesigner clock waveform

Figure 1. (a) The four phases of the QCA clock used to control information flow in the QCA circuit. (b) Implementation of clock waveform in QCADesigner as a clipped sinusoid.

The clock signals act to pump information throughout the circuit as a result of the successive latching and unlatching of cells connected to the different clock phases. Cells latch to a polarization, P, that is driven by the state of neighboring cells and the strength of the electrostatic interaction to those cells [33]. Within the zone clocking scheme, each group of cells connected to a particular phase of the clock can be considered as a D-latch [34]. When the clock value is low, the cells in that clocking zone will become latched (*i.e.*, switch to $\mathrm{P} = \pm 1$) and hold their value until the clock is relaxed (or unlatched), at which point the cell will have $\mathrm{P} = 0$ and remain in this state until the clock returns to its low value once again, independent of changes in the polarization of cells in neighboring clocking zones. An example of a binary wire using zone clocking is shown in Figure 2, where the different clocking zones are represented with different shades of gray. We will use this gray scale to represent the different clocking zones throughout this paper.

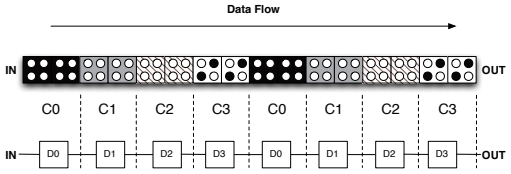


Figure 2. QCA wire shown with cells and schematic representation. C0, C1, C2, C3 are the four phases of the clock. Each of the clocking zones maps to a numbered D-latch in the circuit representation.

2: Contribution

In this paper, we present the results of numerical simulations of QCA building blocks in the presence of random phase variations in the individual clock signals. It is well known that process variations in thickness, alignment, and width do occur in deep sub-micron conventional technologies. These can lead to unexpected variations in resistance, capacitance, and inductance [35]. Clock trees and networks used for field generation and switching in QCA circuits are susceptible to such variations, particularly because of high-speed switching and aggressive scaling of clock lines due to the inherently small size of QCA cells. The current simulation tools enable us to explore the effect of clock phase on the correct operation of QCA circuitry to determine how sensitive these circuits are to these phases. These results will enable us to develop upper bounds on acceptable phase variations which can be used to assess the feasibility of the QCA paradigm.

3: Method

In this section, we present our method of characterizing the different QCA building blocks. The goal of the simulations is to reproduce a random variation in the phase of the clock signals delivered to the different clocking zones and to evaluate its impact on the functional behavior of the building blocks. The simulation algorithm is summarized as follows:

1. For each of the QCA building blocks considered in this work, we generate simulation results for the circuit without introducing any phase shifts. The output traces of these simulations are first digitized and then used to automatically compare against simulations with phase shifts in order to determine if the circuits still operate correctly.
2. A batch simulation with 1000 samples is executed using a random variable, X, to represent the possible phase shifts. The random variable X is characterized by a Gaussian probability distribution function with mean value $\mu = i\frac{\pi}{2}$, where i is the clock number and standard deviation σ.
3. The percentage of successful circuits is recorded. Success is judged based on the ability of the simulated circuit to reproduce the digitized output trace from step 1.
4. Step 2 is repeated for increasing values of σ.

Our objective is to obtain the percentage of successful circuits as a function of σ so that we can determine at what point QCA circuits begin to break down in the presence of these phase variations in their clocking network. The chosen interval is $\sigma \in (0, 45^\circ)$ with a step

of 2.25°.

3.1: Building Blocks of QCA Circuits

The tested QCA building blocks are divided into wires shown in Figure 3 (straight-wire and L-shaped wire), logical devices shown in Figure 4 (inverter and majority gate), and branches shown in Figure 5 (fanout with 2 and 3 outputs). To date, crossover implementation in QCA remains an open discussion as various proposed architectures including coplanar crossover [36, 37], robust coplanar crossover [38] and multi-layer crossover [36] have all been discussed in recent years. However, none of these solutions provide a feasible fabricatable solution to wire crossings. Moreover, recent work has considered the use of logical crossings and gate duplication [39, 40] to avoid physical crossover in QCA circuits suggesting that it may in fact be possible to avoid wire crossings altogether. Given the uncertainty associated with the implementation of wire crossings in QCA, coplanar crossing or any other crossover is not considered in this paper.

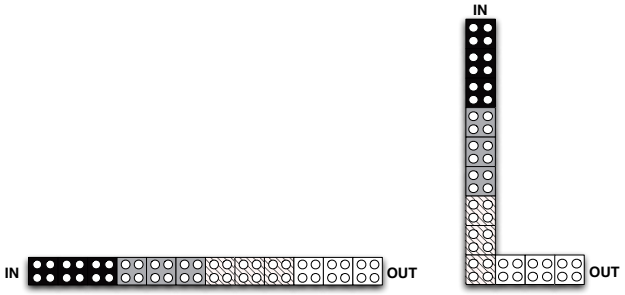


Figure 3. Simulated QCA building blocks: (a) Straight-Wire (b) L-Shaped Wire.

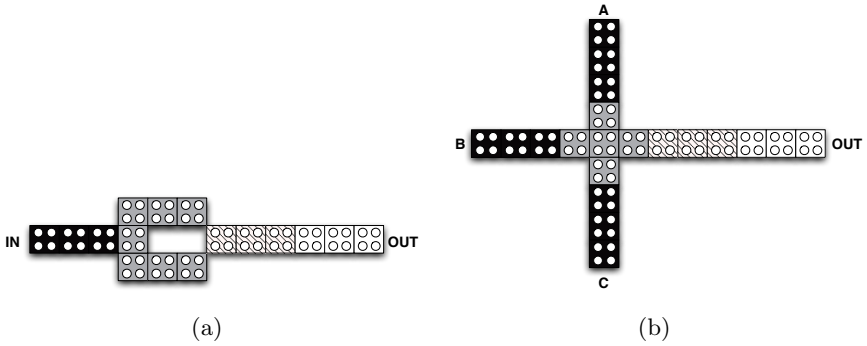


Figure 4. Simulated QCA building blocks: (a) Inverter (b) Majority Gate.

3.2: Simulation setup

Simulations were conducted using both the bistable (time-independent) and coherence vector (time-dependent) simulation engines available in QCADesigner [32] to determine

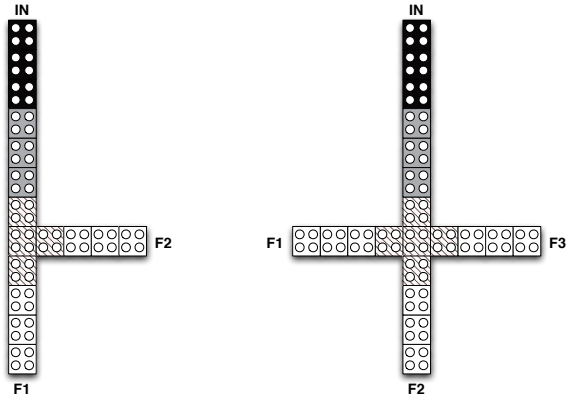


Figure 5. Simulated QCA building blocks: (a) Fanout with 2 outputs (b) Fanout with 3 outputs.

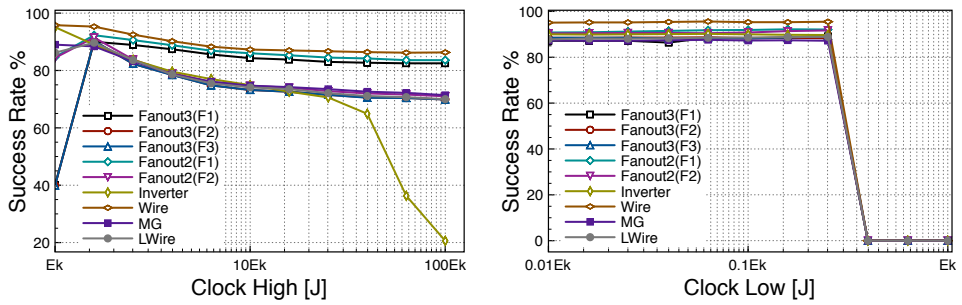
if the behaviour is dependent on the particular choice of model. Both models use the Hartree-Fock approximation which takes an N-cell system and decouples it into a set of N single-cell subsystems that are assumed to interact classically through expectation values without any quantum mechanical coherence between them [41]. The benefit of using this approximation is that for an N-cell system, it is only necessary to solve N 2x2 Hamiltonians as opposed to one $2^N \times 2^N$ Hamiltonian. This allows large circuits, which would otherwise be intractable using a full quantum mechanical model, to be solved on a classical computer [42]. The QCADesigner tool was extended to provide a batch simulation mode in which each individual simulation was executed with a random phase shift added to each of the clock signals. Cell sizes of 2 nm were used throughout with a dielectric constant, ϵ_r, selected to be 1.

3.2.1 Clock Optimization

A set of batch simulations were performed on each of the six building blocks to determine optimal values for both the clock high, C_H, and clock low, C_L, parameters. These values, shown in Figure 1(b), play an important role in the circuit in that they determine the strength of the latched and relaxed state of the cell. The larger C_H, the more relaxed a cell will be and the less likely it is to perturb any neighboring cells. Conversely, the smaller the C_L value, the more latched a cell will be and the less likely it is to *be* perturbed by any of its neighboring cells. The results of these simulations are shown in Figures 6-9. 500 batch simulations were performed on each building block for the worst-case scenario ($\sigma = 45°$), and then the number of successes were plotted versus the different C_H and C_L values as shown in Figures 6 and 8. The C_H and C_L values were kept as a scalar multiple of the kink energy, E_k, which describes the interaction between cells. For the 2 nm cells and dielectric constant used in these simulations, the E_k was found to be 0.293 eV for nearest neighbors. The average number of successes for each C_H and C_L value over the number of devices was calculated (Figures 7 and 9) and then used to determine the optimal values for both simulation engines. We select the optimal value to be the C_H and C_L value that produces the highest average number of successes.

From Figures 6-9, the simulated QCA building blocks behave similarly under both the bistable and coherence vector simulation engines with respect to the various C_H and C_L values with the exception that the bistable simulation engine produces higher success rates. Figures 6(a) and 8(a) show that the optimal C_H value lies at around $1.585E_k$ (≈ 0.465 eV) for both simulation engines since the largest number of successes occur at this C_H value. Values smaller than $1.585E_k$ J may not allow a cell to fully unlatch - allowing it to perturb neighbouring cells in spite of its "relaxed" state. Ideally, large C_H values are desired because as the C_H value approaches infinity, the cells in that clocking zone will have $P = 0$. However, once random phase shifts are introduced into the clock, a clocking zone may enter a relaxed state before the next one has the opportunity to fully latch - creating a gap in the flow of information. In such cases, it is helpful to keep a C_H value slightly lower such that the cells in the relaxed zone retain some residual polarization (weak on-state) that can be used to perturb the neighbouring cells. Further work needs to be done to full understand the mechanisms involved.

Figures 6(b) and 8(b) show that for a QCA circuit to function correctly, the C_L value must be smaller than $0.25E_k$ (≈ 0.073 eV). Larger values do not allow the cells to fully latch and hence the circuit will be unable to propagate the signal forward. While the results shown in Figure 7(b) suggest that there exists no one optimal C_L value for the bistable engine, Figure 9(b) shows that a C_L value of $0.25E_k$ provides, on average, the largest number of successes for the coherence vector simulation engine. Thus we select $0.25E_k$ as a C_L value for both simulation engines.

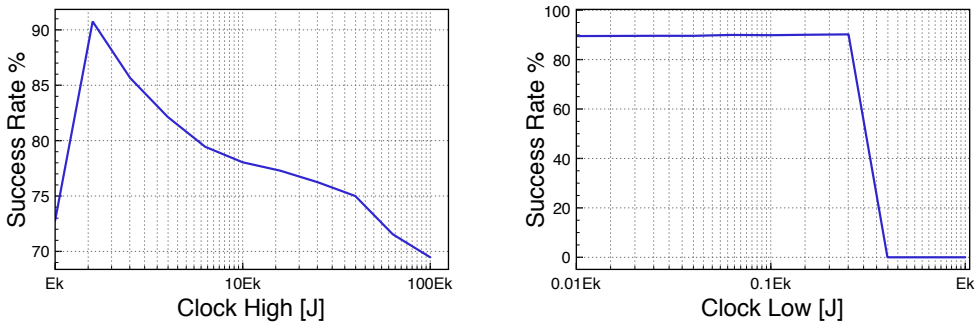


(a) Number of Successes vs Clock High (b) Number of Successes vs Clock Low

Figure 6. Number of successes of each considered QCA building block vs. the C_H and C_L values using the bistable simulation engine.

3.2.2 Simulation Parameters

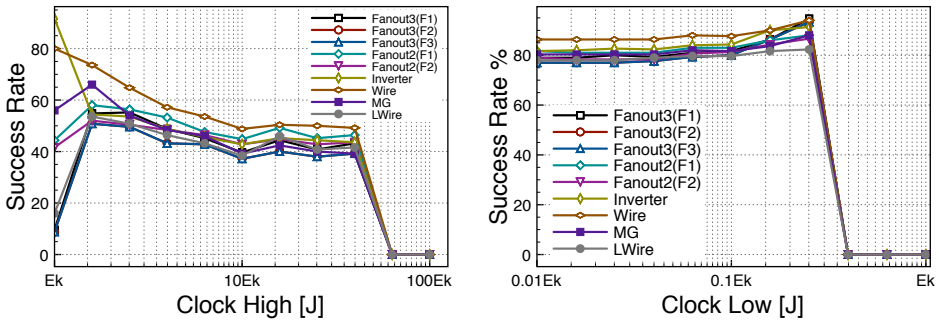
All simulations are performed at a temperature T = 1 K to minimize the influence of thermal noise. A relaxation time $\tau = 1.11 \times 10^{-16}$ s, a time step $t_{step} = 1.11 \times 10^{-18}$ s, and simulation time $t = 1.11 \times 10^{-12}$ s, were used in the coherence vector simulation engine to ensure that the solver converged, while the "radius of effect" parameter, which describes how far the simulator will look for neighboring cells, was kept at 200 nm in order to encapsulate the entire circuit. All other parameters were kept at their default values. A summary of the simulation parameters is shown in Table 1.



(a) Average Number of Successes vs Clock High

(b) Average Number of Successes vs Clock Low

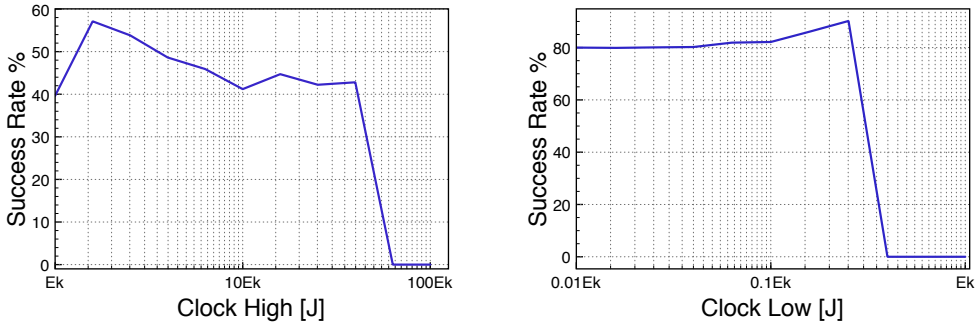
Figure 7. Average Number of successes for the considered QCA building block vs. the C_H and C_L values using the bistable simulation engine. A C_H value of $1.585E_k$ is optimal while any value smaller than $0.25E_k$ provides a working value for C_L.



(a) Number of Successes vs Clock High

(b) Number of Successes vs Clock Low

Figure 8. Number of successes of each considered QCA building block vs the C_H and C_L values using the coherence vector simulation engine.



(a) Average Number of Successes vs Clock High

(b) Average Number of Successes vs Clock Low

Figure 9. Average Number of successes for the considered QCA building block vs. the C_H and C_L values using the coherence vector simulation engine. A C_H value of $1.585E_k$ is optimal while $0.25E_k$ provides an optimal value for C_L.

	Bistable Engine	Coherence Vector
Temperature	N/A	1 K
Relaxation Time	N/A	1.11 e-16 s
Time Step	N/A	1.11e-18 s
Duration	N/A	1.11 e-12 s
Clock High	0.465 eV	0.465 eV
Clock Low	0.073 eV	0.073 eV
Clock Shift	0 J	0 J
Clock Amplitude Factor	2	2
Radius of Effect	200 nm	200 nm
Relative Permittivity	1.0	1.0
Layer Separation	1.15 nm	1.15 nm
Algorithm	N/A	Euler
Number of Samples	12800	N/A
Convergence Tolerance	0.001	N/A
max Iterations per sample	100	N/A

Table 1. Simulation Parameters

4: Simulation Results

The following two sets of simulations were performed:

- *Building Block Level:* Each of the six building blocks described in Section 3.1 were simulated individually using the Bistable and Coherence Vector simulation engines. Results are shown in Figures 10 and 11.
- *Circuit Level:* A full adder circuit, created with several of the underling building blocks, was also simulated. The full adder circuit layout is shown in Figure 15. Simulation results appear in Figure 16.

5: Discussion

5.1: Building Blocks

The results shown in Figures 10 and 11 show that both simulation engines produce the same general trends, confirming that the behavior of the individual building blocks is not dependent on the choice of model. The figures also show that there appears to exist a threshold value of σ before which the probability of success for all building blocks is 100%. For the bistable engine, this σ value is 4°, and for the coherence vector engine, 3.5°. While this threshold value is common to all considered building blocks, the decrease in success rate as a function of σ is different for each individual building block output. This particular characteristic allows us to classify the outputs of each building block into distinct groups based on their ability to tolerate phase variations in the clocking network. For instance, the Fanout2(F2), Fanout3(F1) and Fanout3(F2) building block outputs demonstrate the most dramatic decrease in success rate which is indicative of a high phase variation sensitivity. This is discussed in further detail in Section 5.1.2.

5.1.1 Fault Analysis

Faults due to random phase shifts in the clock can manifest themselves in one of two ways. They can result in either an unwanted delay or an inversion at the primary outputs. The

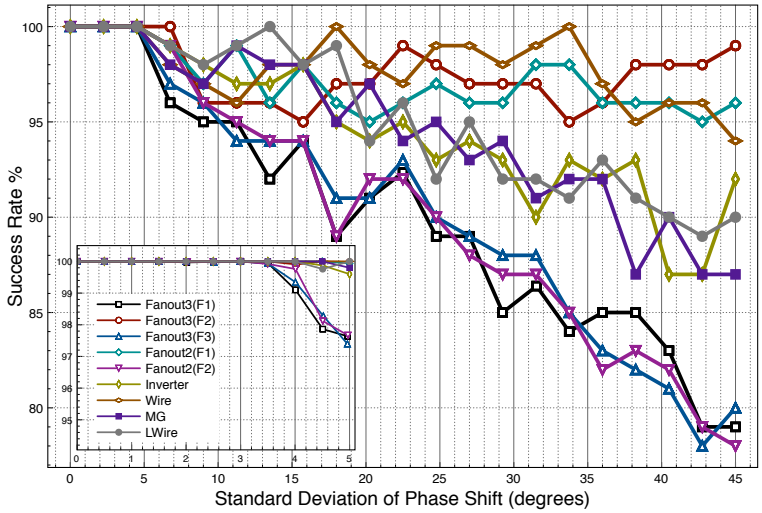


Figure 10. Success rates for building blocks versus standard deviation of phase shift for bistable simulation engine. In-lay figure was generated by simulating for $\sigma \in$ (0, 5°) in increments of 0.5°.

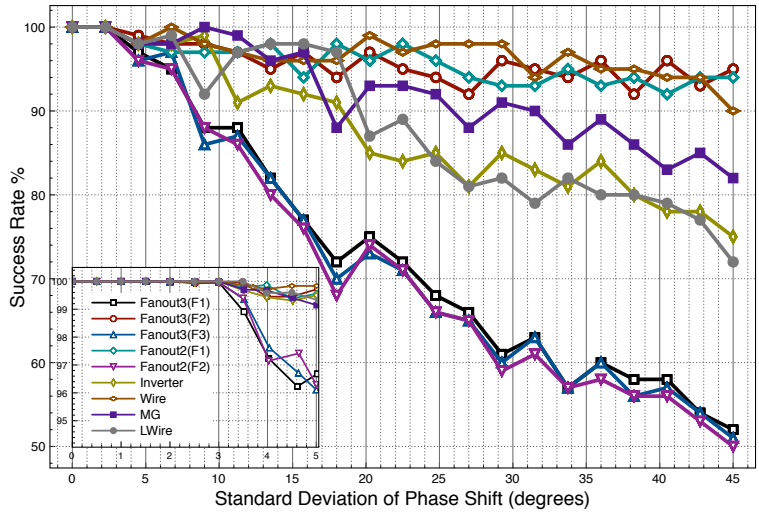
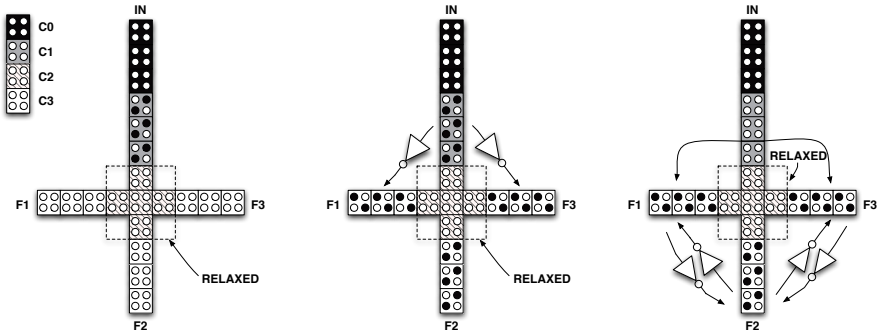


Figure 11. Success rates for building blocks versus standard deviation of phase shift for coherence vector simulation engine. In-lay figure was generated by simulating for $\sigma \in$ (0, 5°) in increments of 0.5°.

delays occur because the clocking zone to which the output is connected latches out of sequence, propagating the information either sooner or later than expected. These delays can often be masked at the output if an unwanted inversion were to occur at the same time and cause us to incorrectly identify a faulty circuit as functional. Thus, it is critical that the input pattern be selected appropriately such that no such false positives occur. Here, the test sequence {0, 0, 1, 1, 0, 1, 0, 1} is selected for all QCA building blocks featuring a single input, and {000, 100, 110, 010, 011, 001, 101, 111} for the majority gate and full adder. These test sequences ensure that the values at the primary outputs do not depend on any of their previous values.

The unwanted inversions are less intuitive. Consider the fanout circuit shown in Figure 12(a). In this figure, the signal has already propagated to the cells in clocking zone C1. Here, we expect C2 to move into a latching state and propagate the signal forward. However, if phase shifts in clocking zones C2 and C3 are significant, then it is possible that the cells in C3 will latch before those in C2. If this occurs, then the cells in C3 will relax to the opposite polarization as those in C1 due to the 45° angle that exists between their cells (which results in a negative interaction energy) as shown by the NOT gates in Figure 12(b). Output F2 is not affected by this phase shift since it is lined up directly with the input and is not subject to any inversion. As a result we expect that F2 will be less sensitive to such phase variations. The same analysis applies to the Fanout2 building block.



(a) Fanout 3 before inversion error

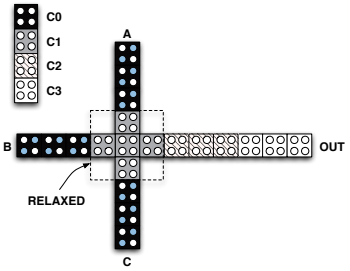
(b) Fanout 3 after inversion error

(c) Fanout 3 exhibiting positive feedback between outputs

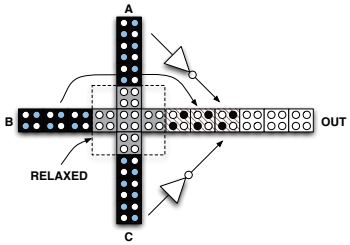
Figure 12. Inversion error in the Fanout3 circuit. If clocking zone C3 latches before C2, then two of the outputs will experience unwanted inversion (F1, F3). The different clocking zones are labeled in the top-left hand corner for reference.

If the C_H value, which determines how relaxed thet cells are, is selected appropriately ($\approx$ 0.465 eV), then there will exist some residual polarization in the cells in C2 which may be sufficient to perturb the cells in clocking zone C3 and overcome the inversion particularly since the interaction between cells in C2 and C3 is much higher than that between cells in C1 and C3. However, the LWire building block does not have any other outputs with which to interact and therefore, may be able to recover its correct output, provided that there exists enough residual polarization in the cells in clocking zone C2.

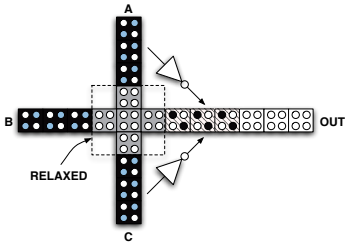
The majority gate behaves similarly to the Fanout3 building block. Consider the majority gate shown in Figure 13(a). Here, the signal has propagated to the cells in clocking zone C0. Assume that the variations in the clock phase have caused C2 to latch before C1. In this case, inputs A and C will have an inverting effect on the output, while input B will drive the output without inversion. Figures 13(b) and 13(c) outline the two possible scenarios that need to be considered under these circumstances. The first results when inputs A and C are logically opposite. If this is the case then the output will be equal to input B and the inverting error that occurs due to inputs A and C cancels out as shown in Figure 13(b). The second scenario occurs when both inputs A and C are logically equivalent. Under these circumstances, the combined inverting interaction with the output of these two inputs cannot be overcome and the output will polarize to the incorrect result as shown in Figure 13(c).



(a) Majority Gate before inversion error



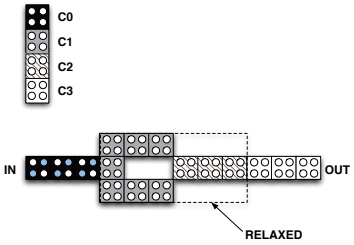
(b) Majority Gate recovering from inversion error



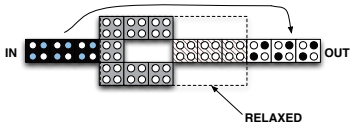
(c) Majority Gate failing due to inversion error

Figure 13. Inversion error in the majority gate circuit. If clocking zone C3 latches before C2, then the output may exhibit an inversion error.

Another example of unwanted inversion can occur in the case of the inverter. Consider Figure 14(a). Here, the cells in clocking zone C0 are holding the signal to be propagated through the inverter. If the phase shifts are such that C3 latches before C1 and C2, then the signal will propagate directly from the cells in C0 directly to the output without undergoing any inversion as shown in Figure 14(b). Under this scenario, the inverter acts like a wire.



(a) Inverter before inversion error



(b) Inverter after inversion error

Figure 14. Inversion error in the inverter circuit. If clocking zone C3 latches before C1 and C2, then the output will simply take on the value of the input, effectively replicating the behavior of a straight wire.

5.1.2 Grouping of QCA Building Blocks

The results shown in Figures 10 and 11 demonstrate that the success rates of the different building block outputs are clustered into three distinct groups. The first group of outputs

are those that have a straight path from input to output. The Wire, Fanout2(F1) and Fanout3(F2) all fall into this group of outputs. This group sees the highest success rate of any other group because the variations in clock phase can only cause delay at these outputs as mentioned in Section 5.1.1, and are not affected by any unwanted inversion.

The second group of outputs are those belonging to either Fanout2 or Fanout3 that do not have a straight path from the input to its output, i.e, there exists a 90^o bend between the input and output. It is clear from Figures 10 and 11 that the outputs that belong to this group - Fanout2(F2), Fanout3(F1) and Fanout3(F2) - produce the worst success rate of any group due to the potential of unwanted inversion at the outputs as discussed in the previous section.

The third group consists of the LWire, MG, and Inverter building blocks. For reasons described in Section 5.1.1, these building blocks have success rates that lie in between that of the other two groups.

5.2: Full Adder

The simulation results for the full adder are shown in Figure 16. C_{out}, whose inputs travel through a fanout, majority gate and straight wires, expectedly performs at a higher success rate than the Sum output whose inputs travel through multiple inverters and majority gates in addition to a fanout block and both straight and L-shaped wires. As σ increases, the success rates of both outputs of the full adder begin to resemble that of the Fanout3(F1)/(F3) building block outputs, suggesting that the overall functionality of the full adder is eventually limited by the performance of the most sensitive QCA building block.

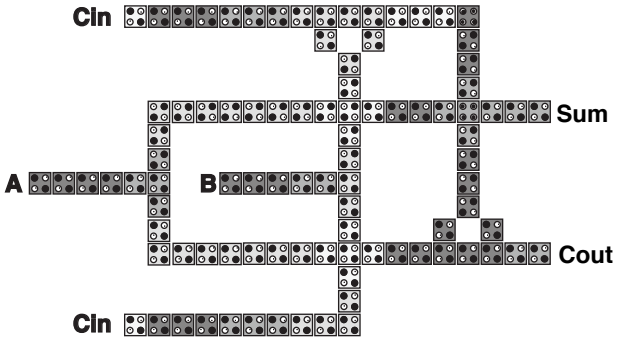


Figure 15. Layout of QCA full adder with no crossover

6: Conclusion

This paper has presented a case study of numerical simulations to evaluate the robustness of QCA circuit building blocks against the effect of random phase shifts on the underlying clocking network. Simulations were performed using a set of universal QCA building blocks, and were repeated using both the bistable and coherence vector simulation engines in QCADesigner. A preliminary assessment of the optimal values of clock strength has

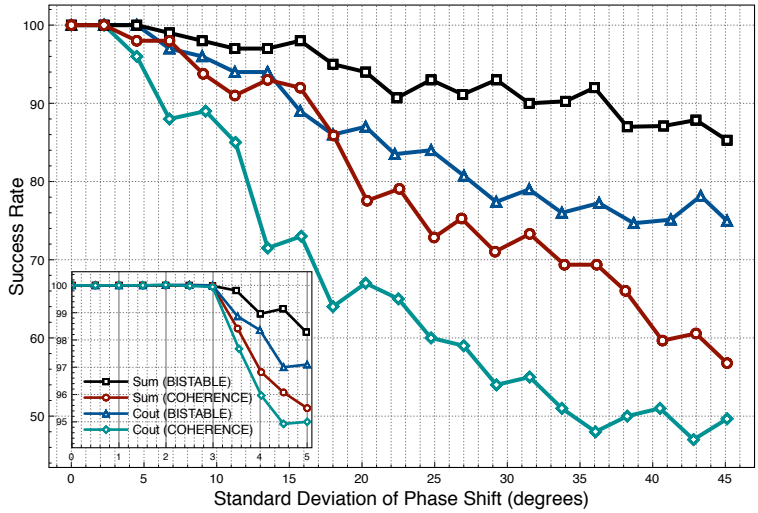


Figure 16. Outputs of QCA full adder: simulation results for both simulation engines. In-lay figure was generated by simulating for $\sigma \in (0, 5^\circ)$ in increments of 0.5°.

been performed in order to have the most meaningful results from the phase dependent simulations. The results from both simulation engines remained fairly consistent, confirming that the behavior of the circuits is not dependent on the choice of model. A threshold value of σ before which the probability of success for all considered building blocks drops below 100% was found to be 4° for the bistable engine and 3.5° for the coherence vector engine. It was also found that the success rate of a given output within a building block is highly layout-dependent. Outputs with a straight path to the associated input displayed more robustness to the phase variations than did those featuring 90° turns. As a result, QCA building blocks can be grouped into distinct classes depending on the number of 90° turns and outputs that they contain making it easier to assess overall circuit sensitivity. In addition to providing critical information for developing fabrication specifications for QCA clocking networks, these results will also help designers to design new QCA circuits while taking the sensitivity information into consideration. Finally, the QCA full adder circuit was also evaluated. The simulation results show that circuit level analysis is consistent with the analysis based on building blocks. As the standard deviation of the phase shift increases, the performance of the outputs of the full adder begin to resemble the performance of the outputs of fanout building block suggesting that the overall functionality of a circuit is eventually limited by its most sensitive building block.

References

[1] R. Chau, J. Kavalieros, B. Roverts, R. Schenker, D. Lionberger, D. Barlag, B. Doyle, R. Arghavani, A. Murthy, and G. Deweyt. 30-nm physical gate length cmos transistors with 1.0-ps nmos and 1.7-ps pmos gate delays. In *IEDM Tech. Dig.*, pages 45–48, 2000.

[2] G. E. Moore. Lithography and the future of moore's law. In *Proc. of SPIE, Advances in Resist Technology and Processing*, volume 2438, pages 2–17, 1995.

[3] L. Risch. How small can mosfets get? In *Proc. of SPIE, Advances in Microelectronic Device Technology*, volume 4600, pages 1–9, 2001.

[4] D. J. Frank, R. H. Dennard, E. Nowak, P. M. Solomon, Y. Taur, and H. S. P. Wong. Device scaling limits of si mosfets and their application dependencies. In *Proc. of IEEE*, volume 89, pages 259–288, 2001.

[5] International Technology Roadmap for Semiconductors. (ITRS). *[Online] http://public.itrs.net*, 2004.

[6] A. O. Orlov, R. K. Kummamuru, R. Ramasubramaniam, C. S. Lent, G. H. Berstein, and G. L. Snider. Clocked quantum-dot cellular automata shift register. *Surf. Sci.*, 532–535:1193–1198, 2003.

[7] I. Amlani, A. O. Orlov, G. Toth, G. H. Bernstein, C. S. Lent, and G. L. Snider. Digital logic gate using quantum-dot cellular automata. *Science*, 284:289–291, 1999.

[8] I. Amlani, A. Orlov, G.L. Snider, and C.S. Lent. Demonstration of a functional quantum-dot cellular automata cell. *J. Vac. Sci. Technol. B*, 16:3795–3799, 1998.

[9] C.S. Lent, G.L. Snider, G.H. Bernstein, W. Porod, A. Orlov, M. Lieberman, T. Fehlner, M.T. Niemier, and P. Kogge. *Quantum-Dot Cellular Automata*. Kluwer Academic Publishers, Boca Raton, FL, USA, 2003.

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

[11] R.V. Kummamuru, J. Timler, G. Toth, C. S. Lent, R. Ramasubramaniam, A. Orlov, and G.H. Bernstein. Power gain and dissipation in a quantum-dot cellular automata latch. *Appl. Phys. Letters*, 81:1332–1334, 2002.

[12] G. Toth and C. S. Lent. Quasiadiabatic switching of metal-island quantum-dot cellular automata. *J. Appl. Phys.*, 85(5):2977–2984, 1999.

[13] Imre, A. and Csaba, G. and Ji, L. and Orlov, A. and Bernstein, G. H. and Porod, W. Majority Logic Gate for Magnetic Quantum-Dot Cellular Automata. *Science*, 311(5758):205–208, 2006.

[14] C. György *et al.* Nanocomputing by field-coupled nanomagnets. *IEEE Trans. Nano.*, 1(4):209–213, 2002.

[15] C. György and W. Porod. Simulation of field coupled computing architectures based on magnetic dot arrays. *J. Comp. Elec.*, 1(1):87–91, 2002.

[16] M. C. B. Parish. *Modeling of Physical Constraints on Bistable Magnetic Quantum Cellular Automata*. PhD thesis, University of London, 2003.

[17] M. E. Welland R. P. Cowburn. Room temperature magnetic quantum cellular automata. *Science*, 287:1466–1468, 2000.

[18] G.H. Bernstein, A. Imre, V. Metlushko, L. Ji, A. Orlov, G. Csaba, and W. Porod. Magnetic QCA Systems. *Microelectronics Journal*, 36:619–624, 2005.

[19] Y. Lu and C. S. Lent. Theoretical Study of Molecular Quantum-dot Cellular Automata. *J. Comput. Elec.*, 4:115–118, 2005.

[20] C. S. Lent, B. Isaksen, and M. Lieberman. Molecular quantum-dot cellular automata. *J. Am. Chem. Soc.*, 125:1056–1063, 2003.

[21] J. Jiao, G. J. Long, F. Grandjean, A. M. Beatty, and T. P. Fehlner. Building blocks for the molecular expression of quantum cellular automata. isolation and characterization of a covalently bonded square array of two ferrocenium and two ferrocene complexes.

J. Am. Chem. Soc., 125(25):7522–7523, 2003.

[22] C. S. Lent and B. Isaksen. Clocked molecular quantum-dot cellular automata. *IEEE Trans. Electron Devices*, 50(9):1890–1896, 2003.

[23] Zengxiao Jin. *Fabrication and Measurement of Molecular Quantum Cellular Automata (QCA) Device*. PhD thesis, University of Notre Dame, Notre Dame, IN 46556, 2006.

[24] Z. Li and T. P. Fehlner. Molecular qca cells. 2. characterization of an unsymmetrical dinuclear mixed-valence complex bound to a au surface by an organic linker. *Inorg. Chem.*, 42(18):5715–5721, 2003.

[25] Z. Li, Alicia M. Beatty, and Thomas P. Fehlner. Molecular qca cells. 1. structure and functionalization of an unsymmetrical dinuclear mixed-valence complex for surface binding. *Inorg. Chem.*, 42(18):5707–5714, 2003.

[26] Qi, H., Sharma, S., Li, Z., Snider, G.L., Orlov, A.O., Lent, C.S., and Fehlner, T.P. 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.*, 125(49):15250–15259, 2003.

[27] M. B. Tahoori, M. Momenzadeh, J. Huang, and F. Lombardi. Defects and faults in quantum cellular automata at nano scale. In *Proc. IEEE VLSI Test Symposium*, pages 291–296, 2003.

[28] M. Momenzadeh, M. Ottavi, and F. Lombardi. Modeling qca defects at molecular-level in combinational circuits. In *Proc. IEEE Conf. on Defect and Fault Tolerance*, pages 208–216, 2005.

[29] G. Schulhof, K. Walus, and G. A. Jullien. Simulation of random cell displacements in qca. *J. Emerg. Technol. Comput. Syst.*, 3(1):2, 2007.

[30] K. Walus and G. A. Jullien. Design tools for an emerging soc technology: quantum-dot cellular automata. *Proc. IEEE*, 94(6).

[31] K. Walus, T. Dysart, G. A. Jullien, and R. A. Budiman. QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata. *IEEE Trans. Nano.*, 3(1).

[32] K. Walus and G. Schulhof. QCADesigner Homepage. *[Online] http://www.qcadesigner.ca/*, 2001.

[33] C. S. Lent. Quantum cellular automata. *Nanotechnology*, 4:49–57, 1993.

[34] K. Walus, M. Mazur, G. Schulhof, and G. A. Jullien. Simple 4-bit processor based on quantum-dot cellular automata (QCA). In *Proc. of Application Specific Architectures, and Processors Conference*.

[35] S. Natarajan, M. A. Breuer, and S. K. Gupta. Process variations and their impact on circuit operation. In *Proc. IEEE Conf. on Defect and Fault Tolerance*, pages 73–81, 1998.

[36] K. Walus, G. A. Jullien, and R. A. Budiman. Qca co-planar wire-crossing and multilayer networks. In *Proc. of iCore Banff Summit*.

[37] P. D. Tougaw and C. S. Lent. Logical devices implemented using quantum cellular automata. *J. Appl. Phys.*, 75(3):1818–1825, 1994.

[38] S. Bhanja, S. Srivastava, M. Ottavi, and F. Lombardi. Novel designs for thermally robust coplanar crossing in qca. In *Proc. of Design, Automation and Test in Europe, 2006. DATE '06.*, 2006.

[39] A. Chaudhary, D.J. Chen, X.S. Hu, M.T. Niemier, R. Ravichandran, and K. Whitton. Eliminating wire crossings for molecular quantum-dot cellular automata implementation. In *IEEE/ACM International Conference on Computer-Aided Design, 2005. IICAD-2005.*, pages 564–571, 2005.

[40] A. Chaudhary, D.J. Chen, X.S. Hu, M.T. Niemier, R. Ravichandran, and K. Whitton. Fabricatable interconnect and molecular QCA circuits. *IEEE Trans. On CAD of IC*, 26(11):1978–1991, 2007.

[41] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman. QCADesigner: a rapid design and simulation tool for quantum-dot cellular automata. *Nanotechnology, IEEE Transactions on*, 3(1):26–31, 2004.

[42] K. Walus and G. A. Jullien. Design tools for an emergng soc technology: Quantum-dot cellular automata. *Proceedings of the IEEE*, 94(6):1225–1244, 2006.