Towards Designing Robust QCA Architectures in the Presence of Sneak Noise Paths

Kyosun Kim[1], Kaijie Wu[2], Ramesh Karri[3]

[1]Department of Electronic Engineering, University of Incheon, Incheon, Korea kkim@incheon.ac.kr

[2]Department of Electrical and Computer Engineering, University of Illinois at Chicago, USA kaijie@ece.uic.edu

[3]Department of Electrical and Computer Engineering, Polytechnic University, Brooklyn, USA ramesh@india.poly.edu

Abstract[*]

Quantum-dot Cellular Automata (QCA) is attracting a lot of attentions due to their extremely small feature sizes and ultra low power consumption. Up to now there are several designs using QCA technology have been proposed. However, we found not all of the designs function properly. Further, no general design guidelines have been proposed so far. A straightforward extension of a simple functional design pattern may fail. This makes designing a large scale circuits using QCA technology an extremely time-consuming process. In this paper we show several critical vulnerabilities in the structures of primitive QCA gates and QCA interconnects, and propose a disciplinary guideline to prevent any additional plausible but malfunctioning QCA designs.

1. Introduction

Scaling of CMOS devices is being aggressively pursued by shrinking transistor dimensions, reducing power supply voltages and increasing operating frequencies. Such aggressive scaling adversely results in a series of non-ideal behaviors such as high leakage current and high power density levels. These issues will eventually become road blocks and slow down the scaling trend that exists for years [1]. Quantum-dot Cellular Automata (QCA) is attracting a lot of attention due to their extremely small feature sizes (at the molecular even atom level) and ultra low power consumption [2]. A quantum cell shown in Figure 1 (a) consists of four dots at the corners with two excess electrons that can tunnel between the dots. Due to Coulomb repulsion the two excess electrons always occupy diagonally opposite dots. There are two configurations with energetically equivalent polarizations designated as +1 and -1. Tunneling out of a cell is suppressed due to high inter-cell barriers. In a second type of QCA cells, the dots are located at the middle of the sides of cells as shown in Figure 1 (b). The basic logic element in QCA logic is a majority gate and shown in Figure 1 (c). Cells **A**, **B** and **C** serve as drivers or input cells. **F** is the output cell and is polarized according to the polarization of the majority of the driver cells. In this example since polarization of 2-out-of-3 input cells are -1, the polarization of the output cell is -1. The cell arrangement in Figure 1 (d) implements an inverter since the polarization of the output **Out** is the opposite of the polarization of input **In**. The wires

constructed using the two types of cells are shown in Figure 1 (e) and Figure 1 (f). When an input is applied to the input cell, the binary information propagates from left to the right due to the Coulomb repulsion between the electrons of neighboring cells. When all cells in a wire settle down to their ground states, they have the same polarization. In Figure 1 (f), when all cells settle down each cell has a different polarization than its neighbors in the wire array.
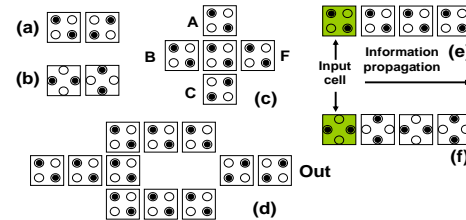

Figure 1: (a) Two 90-degree quantum cells with opposite polarizations (b) Two 45-degree quantum cells with opposite polarizations (c) A majority gate (d) An inverter (e) A QCA wire using 90-degree cells and a (f) QCA wire using 45-degree cells

Polarization switch of a cell is caused by electron tunneling between neighboring dots within the cell. However, when the inter-dot barrier is high, the cell will remain its polarization and will not react to polarization changes of its neighbors. The inter-dot barrier of a cell can be modulated as a clock to allow or deny the polarization changes by the environment. Usually one clock cycle is divided into four phases, namely, switch, hold, release, and relax. During the switch phase, the inter-dot barrier is raised and the cell gradually settles down to its ground state. During the hold phase, the inter-dot barrier remains high, thus suppressing electron tunneling and freezing the cell at its current ground state. During the release and relax phases, the inter-dot barriers are lowered down while the electrons gain mobility gradually. The cell becomes un-polarized and can react to polarization changes of its neighbors. Therefore, the polarization of a cell is determined during the switch phase by the neighbors that are currently in the hold phase, or being newly polarized in the switch phase. The un-polarized neighbors in the release and relax phases do not affect the polarization of the switching cell.

In general, a clocked QCA design uses four pipeline clocks ϕ_1, ϕ_2, ϕ_3 and ϕ_4. Each of the clocks has a 90-degree phase delay to its previous clock. Each cell in a QCA design is assigned one of the pipeline clocks. A cell that is assigned a clock ϕ_i is polarized during the switch phase mostly by its neighbor cells that are assigned the same clock. Since this

cell also contributes to the polarization of its neighbors that are assigned the same clock, the information flows bi-directionally and forms a feedback among the cells with the same clock. The neighbor cells that are assigned clock ϕ_{i-1} (in the hold phase) also contribute to the polarization of the cell (in the switch phase of ϕ_i). However, the cell that is assigned clock ϕ_i, does not affect the polarization of its neighbors that are assigned clock ϕ_{i-1}. This property allows only unidirectional signal flow at the interface between cells that are assigned different pipeline clocks.

After the basic operations of the QCA cell were demonstrated on a hardware implementation in late 1980s, a variety of QCA designs spanning from small scale circuits like an adder to a large scale integration like a micro-processor have been reported. Tougaw and Lent first proposed the design of a QCA-based 1-bit full adder [3]. The full adder takes A, B and carry-in Cin. The Sum is generated as M(M(A', B, Cin), M(A, B', Cin), M(A, B, Cin')) where A' B' Cin' are the complementary of A B Cin respectively and M is a majority gate. Similarly the carry out Cout is generated as M(A, B, Cin). Overall this full adder takes five majority gates, three inverters and requires 192 cells in all. Another QCA full adder with fewer cells is proposed in [4]. This design generates Sum by using M(Cout', Cin, M(A, B, Cin')) and the total number of cells has been reduced to 145. A bit-serial adder proposed in [5] modifies the full adder implementation of [4] to include a feedback connection between Cout and Cin. A QCA-based carry-look-ahead adder is obtained by connecting the carry out of a full adder to the carry in of the next full adder [6]. A microprocessor is proposed on [7].

On the other hand, design tools and simulators have been developed to facilitate the design entry and verification. There are four types of simulation models that have been used so far [9]: Coherence Vector, Bistable, Nonlinear Approximation and Digital. The Coherence Vector model calculates the timing-dependent state of a cell based on the kink energy between this cell and all the other cells. The kink energy between two cells is the energy cost of these two cells having opposite polarizations. The accuracy of Coherence Vector model depends on the granularity of the timing step and can be used to evaluate the dynamic behavior of cell's polarization switching. Bistable and Nonlinear Approximation models also use the kink energy to calculate the state of the cell in a time-independent way thus reducing the total time of simulation. Digital model works like a binary logic analyzer and is the fastest but the least accurate simulation engine.

Unlike the asserted simplicity of device and interconnect structures that are introduced in previous work, one can be easily frustrated by the failures on the simulation of the QCA designs. We have found out that most of the QCA designs that are presented in previous work are not operational. One may have managed to succeed in simulating a small circuit on a single simulation model by tweaking parameters of the simulator, and redrawing circuit parts. Unfortunately, the simulation of the QCA design using other models may fail again. We have found several critical vulnerabilities in the structures of primitive QCA gates and QCA interconnects. We will describe each of them in the rest of this paper. In order to prevent any additional plausible but malfunctioning QCA designs, a disciplinary guideline for robust QCA designs are also provided.

2. Sneak Noise Paths in QCA Designs

Coherence Vector model calculates the state of a cell based on accumulated kink energy. The kink energy of cell i and j represents the energy cost of cells i and j that have opposite polarizations. It is calculated from the electrostatic interaction between all the charges. For each dot in cell i, the electrostatic interaction between this dot and each dot in cell j is calculated as follows:

$$E_{i,j} = \frac{1}{4\pi\varepsilon_0\varepsilon_r} \frac{q_i q_j}{|r_i - r_j|}$$

where ε_0 is the permittivity of free space and ε_r is the relative permittivity of the material system. This is accumulated for all i and j. The overall kink energy is the summation of the all the individual kink energy. Therefore, the state of a cell is determined by all its neighboring cells, not only the ones that deliver the desired information.
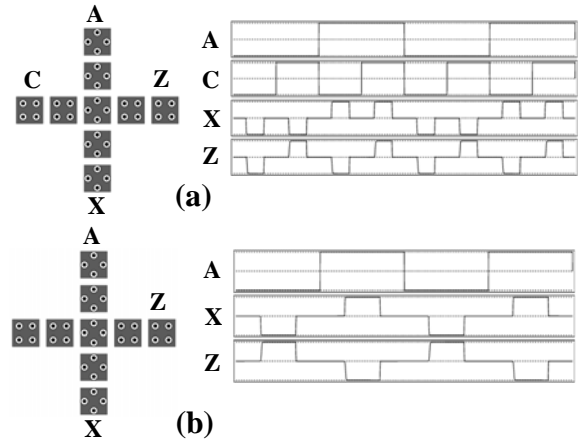

Figure 2: (a) a crossover and the simulation result (b) a crossover with input C absent and the simulation result

Consider a crossover shown in Figure 2 (a). The input applied to cell C crosses over the wire with an input applied to cell A, and is observed at cell Z. The simulation result confirms the functional correctness. However, input at A also participates in determining the state of cell Z, and actually all the cells on the horizontal wire. The simulation without input at C shown in Figure 2 (b) confirms that the state of cell Z is determined by the input at A when input at C is absent. From a designer's point of view the effect on cell Z cast by input at C is signal while the effect cast by input at A is noise. In a QCA design when multiple inputs are present, the signal of a cell is defined as the cell's logic

input while noises are defined as the effects cast by all other inputs.

While in this example signal beats the noise and Z carries the signal, it may not always be true. In this section we will identify several design patterns with hidden noise paths that will cause circuit fail. And we will analyze the reason of failures and propose appropriate design rules.

2.1. An Extended Crossover Structure

The horizontal wire of the crossover shown in Figure 2 (a) is extended by adding one more cell before output Z. The extended crossover is shown in Figure 3 (a). However, the simulation result using the coherence vector model shows that the signal input at cell C fails being transferred to cell Z. The information carried by cell Z is actually the inversion of input at cell A.
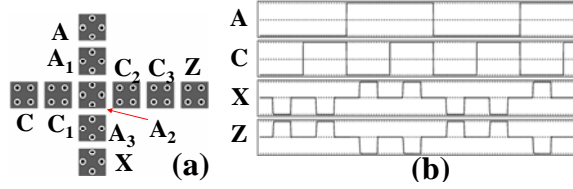

Figure 3: (a) An extended crossover (b) Simulation result

Figure 4 lists the four possible polarization patterns between A_1 and C_2. The kink energy between A_1 and C_2, which is calculated by combining the electrostatic interactions of all possible situations, is 0. In another word, the polarization of cell A_1 has no effect on the polarization of cell C_2.
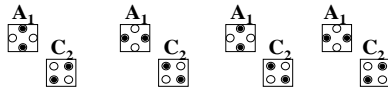

Figure 4: The 4 polarization patterns between A_1 and C_2

i	j	Energy (J)	i	j	Energy (J)
C_2	A	-2.10e-24	A_3	C	+2.10e-24
	A_1	0		C_1	0
	A_2	0		A_2	-1.57e-22
	A_3	0		C_2	0
	X	+2.10e-24		C_3	-2.10e-24
	C_1	+3.48e-24		A_1	-4.20e-24
	C_2	+1.17e-22		X	-1.57e-22

Table 1: The kink energy between cells in the crossover

Similarly the polarizations of cells A_2 and A_3 in the vertical array do not affect the polarization of cell C_2. The kink energies between cells are summarized in Table 1[1]. In our design, the diameter of a dot is 5 nm and the cell size is 18 nm × 18 nm. The cell distance is 5 nm and the grid space is 23 nm. The horizontal signal jumps from cell C_1 to cell C_2 crossing over cell A_2. Unfortunately, the cell pairs {C_2, A}, and {C_2, X} have non-zero values of the kink energy since the dot polarization patterns are asymmetric. We call this the

---

[1] The kink energy is obtained by printing the internal variables of QCADesigner [9].

sneak noise path since it conducts the noise from the input at A to cell C_2.

The effect that the state of one cell has on that of its neighbors can be quantified by a cell-cell response function. The nonlinearity and bistable saturation of the cell-cell response serves the same role as gain in a conventional digital circuit [11]. A very slight polarization of a cell induces a much larger polarization of its neighbor. The neighbor also feedbacks a larger polarization to the cell even before the neighbor's polarization is saturated. Such synergic effect amplifies not only the polarization of a signal, but also that of a noise which propagates through the sneak noise path. Consider the cell arrangement shown in Figure 5 (a). Two inputs are applied at A and B. From the designer's point of view, the input from A acts as signal while the input from B acts as noise. Although the kink energy between A_7 and A_8 is about 30 times stronger than the one between B_1 and A_8, the noise from B arrives at cell A_8 earlier than the signal due to its shorter propagation path, and then propagates down to cells A_{10} and A_{11}. The positive polarization feedback between these cells amplifies the noise so that the signal is stuck at A_8, and propagates no further. However, if cell A_{11} is removed from the end of the wire, the noise-induced polarization is not fully amplified, and the noise disappears as shown in Figure 5 (b). This experiment shows that the noise amplification is successful when both conditions are met: noises arrive earlier than signals, and the wire segment at the noise injection point is long enough. In other words, the noise amplification can be prevented by either limiting this length or letting signal arrive first.
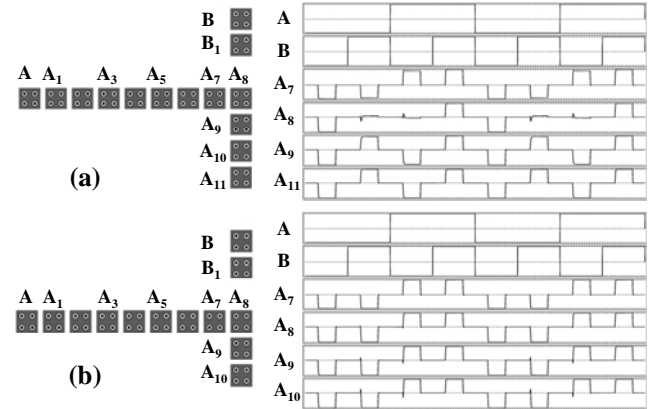

Figure 5: The amplification of noise due to the cell-cell response

Consider again the crossover pattern shown in Figure 3. To prevent the noise amplification on cells C_2, C_3 and Z, the signal has to arrive at cell C_2, C_3 and Z no later than noise. This requires a clocked QCA design. The revised crossover and the simulation result are shown in Figure 6. The horizontal wire is segmented into two phases with a 90-degree phase delay in between. The QCA pipeline clocks are represented by different gray levels. The states of cells

C_2, C_3 and Z will not be determined until the hold phase of cells C and C_1. During the hold phase of cells C and C1, which is also the switch phase of cells C_2, C_3 and Z, the polarizations of cells C_2, C_3 and Z are determined simultaneously by signal from C and noise from A and are eventually settle down to signal. The simulation result confirms that the signal on cell **C** has been successfully transferred to cell **Z**. Extended simulation shows that the results are consistent in all abstraction levels of the models, although the results are not shown here for simplicity.
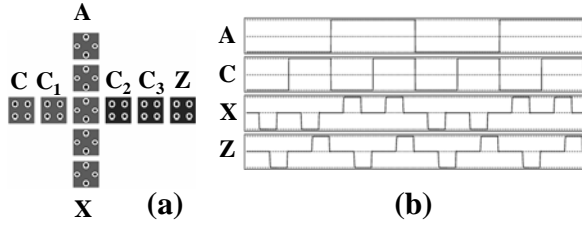

Figure 6 A working crossover and the simulation result

2.2. Majority Gate Structures

Consider a majority gate implementation shown in Figure 7 (a). Cells A, B and C serve as the inputs and cell Y is the output. All the cells are in a single phase. The simulation using Coherence Vector model, however, shows that this gate does not work as a majority gate at all as shown in Figure 8 (a). Due to the unbalanced input paths, signal from A and B arrive at gate device G earlier than signal from C. The gate device will gain its polarization from cells GA and GB, and then propagate the polarization down to C. Signal from C will lose its chance of voting at gate device G and eventually stuck at somewhere between GC and C.
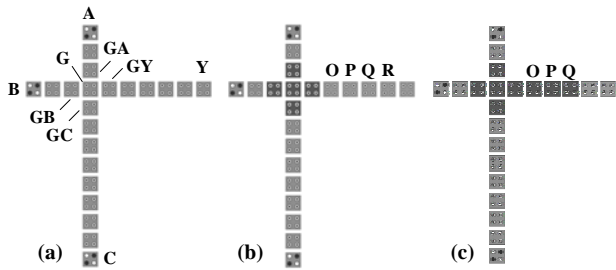

Figure 7: Majority gates with (a) all cells in a phase (b) a proper clock assignment (c) noise amplification due to an improper clock assignment

In order to have a fair voting, all the signals should arrive at the gate device simultaneously. A functional QCA majority gate with a proper clock assignment and its simulation are shown in Figure 7 (b) and Figure 8 (b), respectively. Cells GA, GB, GC, GY and G are in a new phase with a 90-degree phase delay than cells A, B and C. Notice that output cells O, P, Q, R, and Y are assigned to another phase with a 90-degree phase delay than cell GY. Cells GA, GB and GC will gain their polarizations and vote on gate device G at the same time, no matter how unbalanced the three input paths are.

However, if cells O, P, Q have the same phase with cells GA, GB, GC and GY, e.g. the shape of the phase at the cross is extended toward the output as shown in Figure 7 (c), faults occur when the signals on cells A and C are all -1 at the third clock cycle, and all 1 at the sixth clock cycle as shown in Figure 8 (c). At the third clock cycle, cells GA and GC are temporarily polarized to -1. Since the placement between cells GA, GC and GY works like an inverter as shown in Figure 1 (d), this in turn polarizes cell GY to 1. Due to the synergic effect of the cell-cell response between cells GY, O, P and Q, this noise is successfully amplified, and cell GY casts a vote for 1 at the majority gate. The fault at the sixth clock cycle can be similarly explained. It is noticeable that neither Bistable nor Nonlinear Approximation models detects this dynamic behavior since they calculate the state of a cell in a time-independent way.
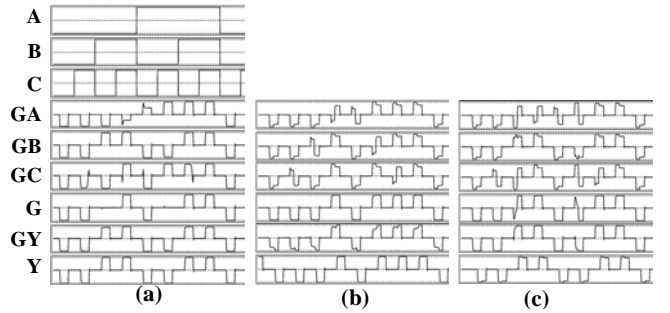

Figure 8: Simulation results of the majority gates

2.3. The Minimum Wire Length of a Phase Block

A **phase block** can be defined as a group of cells that are connected, and assigned the same QCA pipeline clock. Two cells are connected if their diagonal distance is less than $2^{0.5}$ grids. A phase block may consist of a single cell. However, the simulation shows that the waveform of the signal on a single cell phase block becomes distorted, and cascading of such blocks causes functional failures. The schematic and simulation result are shown in Figure 9. While signal A_1 still keeps the waveform of signal A with a clock phase delay, signal A_2 is distorted, and signal A_3 is inverted.
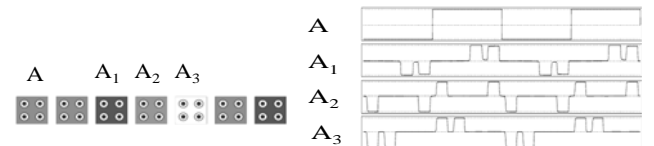

Figure 9 A wire with cascaded single-cell phase blocks and the simulation result
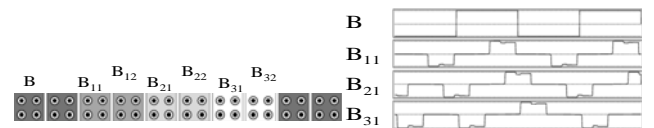

Figure 10 A wire with cascaded double-cell phase blocks and the simulation result

This vulnerability can be made up by letting the minimum length of a phase block be 2 cells so that the synergic effect of the cell-cell response amplifies the weak signal. A wire

consisting of cascaded double-cell phase blocks and the simulation result are shown in Figure 10. The distortion of the waveform has disappeared, and cascading of double cell phase blocks results in no functional failures.

2.4. The Minimum Wire Spacing

As shown in Table 1, the kink energy is 1.16×10^{-22} J and 3.48×10^{-24} J when the spacing between two cells is zero grids and one grid, respectively. Since the kink energy between two cells with zero grid spacing is about thirty times larger than the one with one grid, one grid is enough for the minimum spacing between cells carrying different signals. However, since a horizontal wire sometime may cross over a vertical wire, not all the cells in horizontal wire have zero grid with their neighbors. Therefore the spacing between cells carrying different signals should be at least two grids for safety.

2.5. The Maximum Wire Length

Towards searching for the maximum wire length that can successfully propagate a signal from an end to the other end, consider an experimental setup shown in Figure 11. A wire is implemented by a phase block of 90-degree cells in a row. Signal A is injected from a phase block at the left side of the wire, and measured at a phase block on the other side. This wire is simulated at clock rates of 1 THz and 2 THz and the wire length is increased gradually until the signal fails to propagate to the other side. Also, a wire of 45 degree cells is simulated. The simulation shows that a signal can propagate up to 28 90-degree cells, or 27 45-degree cells at a clock rate of 1 THz, and 12 90-degree cells or 10 45-degree cells at a clock rate of 2 THz. The maximum length of a wire is limited by the clock rates, and should not exceed the corresponding limits.
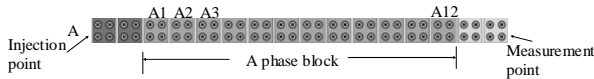

Figure 11: The maximum length of a phase block

The propagation of a signal can be delayed by jogs and rippers on the interconnect wires. Towards evaluating the delay of a jog, consider a wire with five jogs as shown in Figure 12 (a). The simulation shows that the signal that is injected to cell A_1 is propagated up to cell A_{23} at a clock rate of 1 THz. Notice that the number of cells that a signal is propagated to has been reduced from 28 to 23. Since the five jogs are responsible for this reduction, the delay of a jog can be calculated as 1 cell (= 5 cells / 5 jogs).

This delay can be explained in the following way. Assume that 1 is injected at cell A_1. At the second jog, cell A_7 polarizes cell A_8 to 1, and concurrently, cell A_9 to -1. Once cell A_8 is polarized to 1, cell A_8 polarizes cell A_9 to 1. The earlier propagation of -1 to A_9 hinders the propagation of 1. Similarly, the delay of a ripper is also calculated as 2 cells (= (28 – 20) cells / 4 rippers) at a clock rate of 1 THz by simulating the wire shown in Figure 12 (b). The insertion of

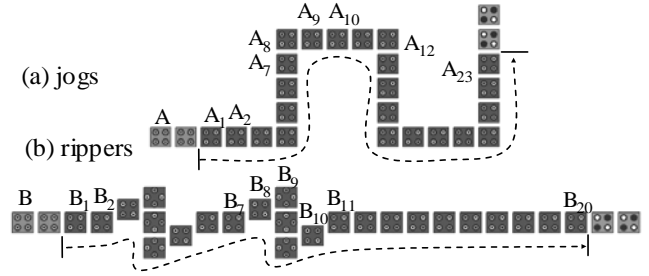jogs and rippers on a wire shortens the maximum allowable length as much as the delays multiplied by their counts

Figure 12: The propagation delay induced by jogs and rippers

2.6.Synchronization

The phase delay of a path can be defined as the number of clock phase changes that have been experienced by a signal to propagate down the path. The input signals arriving at a gate should be synchronized. The phase delay of each path from a primary input to an input of a gate should be the same. The synchronization incurs the area overhead since additional phase blocks need to be inserted to balance the phases. Since the insertion of a phase block necessitates the phase shifts of the cells at the logic stages that follow, the design process becomes complicated. Also, the phase delay is very difficult to estimate during the logic design phase until the schematic diagram is completed since the interconnect structures also increase phase delays. This also complicates the top-down style hierarchical design.

3. ALU Case Study

Towards validating the proposed disciplinary rules for robust QCA designs, we redesigned the bit slice of the Simple 12 ALU which was presented in [7]. The original design was not operational mostly due to the sneak noise path in the crossover structure, and the asynchronous signal flow of the gate structure. The ALU consists of three units – adder unit, logic unit, and complement-zero unit – as shown in Figure 13. It has three data inputs – A, B and Carry In (CI) – and three control inputs – Zero A (ZA), Invert B (IB, also used as OR/AND select), and Logic/Arithmetic select (L/A). The data outputs are Carry Out (CO) and OUT which is selected out of Logic Output (LO) and Sum (S) by a multiplexer. The QCA pipeline clocks are assigned to the cells so that the noise in crossovers can be tolerated, and the signal flows in gates can be synchronized. The control inputs that are fed at the left side of the design are extended to the right side so that an *n*-bit ALU can be constructed by cascading *n* such bit slices. These feed-through outputs are also synchronized with data outputs. This bit slice of the Simple 12 ALU is implemented in the area of 58×81 grids^2 using 1030 cells, and operates at a clock rate of 1 THz. The latency of a 1-bit operation is 34 clock phases (8.5 clock cycles). We simulated the design by using the coherence vector model, and the results are shown in Figure 14. The first two waveforms are the inputs to the logic unit and the

third waveform is the output of the logic unit which performs OR and AND operations. Following the three inputs to the adder unit, the sum and carry out outputs are shown. The truth tables for the logic and add operations are also shown to be compared with the waveforms. The waveform intervals that correspond to the truth tables are highlighted by rectangles. The functional correctness of the design can be easily identified. An extensive simulation using the non-linear approximation model which is about 100 times faster than the coherence vector model showed similar results although the results are not shown here due to the limited space.

4. Conclusions

Most of QCA designs from previous work cannot function properly. In this paper we have identified several primitive design patterns that will fail due to noises of multiple inputs. We analyze such failures and conclude that most of failures are due to the ignorance of the sneak noise paths. A set of disciplinary rules that can effectively suppress noises is presented for making robust QCA designs. The correctness of designs which are compliant with the rules can be verified by using the time-dependent simulation model such as Coherence Vector, as well as time-independent simulation models such as bistable and non-linear approximation.

References

[1] International Technology Roadmap for Semiconductors: http://public.itrs.net/Files/2003ITRS/Home2003.htm

[2] C.S. Lent, P.D. Tougaw, W. Porod, and G.H. Bernstein, "Quantum cellular automata," Nanotechnology, vol. 4, pp. 49-57, 1993.

[3] P.D. Tougaw and C.S. Lent, "Logical devices implemented using quantum cellular automata", Journal of Applied physics, vol. 75(3) pp. 1818-1825, February 1, 1994.

[4] W. Wang, K. Walus, G.A. Jullien, "Quantum-Dot Cellular Automata Adders", IEEE Nano 2003 Conference, San Francisco, 2003.

[5] A.Fijany, N. Toomarian, K. Modarress, M. Spotnitz, "Bit-serial Adder Based on Quantum Dots", NASA technical report, Jan. 2003.

[6] A. Vetteth, K. Walus, V.S. Dimitrov, G.A. Jullien, "Quantum-dot cellular automata carry-look-ahead adder and barrel shifter", IEEE Emerging Telecommunications Technologies Conference, Dallas, Sept. 2002.

[7] M.T. Niemier, M.J. Kontz, P.M. Kogge, "A Design of and Design Tools for a Novel Quantum Dot Based Microprocessor", Proceedings of the 27th Design Automation Conference, p. 227-232, June 2000.

[8] J. Timler and C. S. Lent, "Power gain and dissipation in quantum-dot cellular automata", J. Appl. Phys., vol 91, no 2, pp. 823-831, 2002

[9] QCA Designer, http://www.qcadesigner.ca

[10] W. Wang, K. Walus, G.A. Jullien, "Quantum-Dot Cellular Automata Adders", IEEE Nano 2003 Conference, San Francisco, CA 2003.

[11] P.D. Tougaw and C.S. Lent, "Dynamic Behavior of Quantum Cellular Automata," Journal of Applied Physics, vol.80, no.8, pp.4722-4736, October 15, 1996.
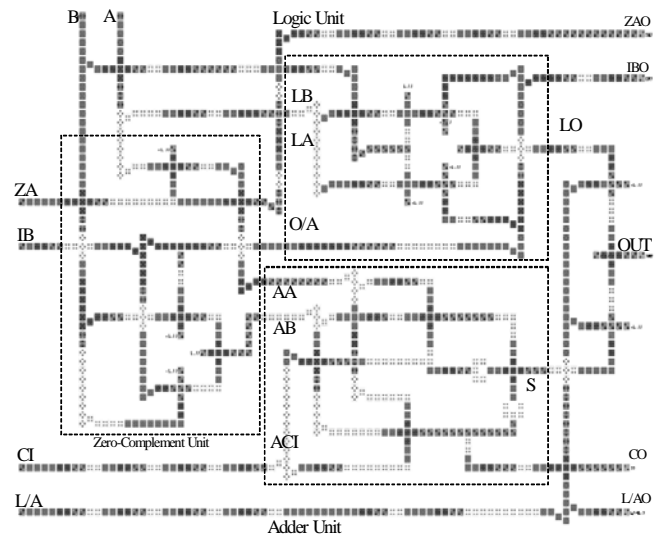
Figure 13: Schematic of the Simple 12 ALU bit slice

LA	LB	OR	AND
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

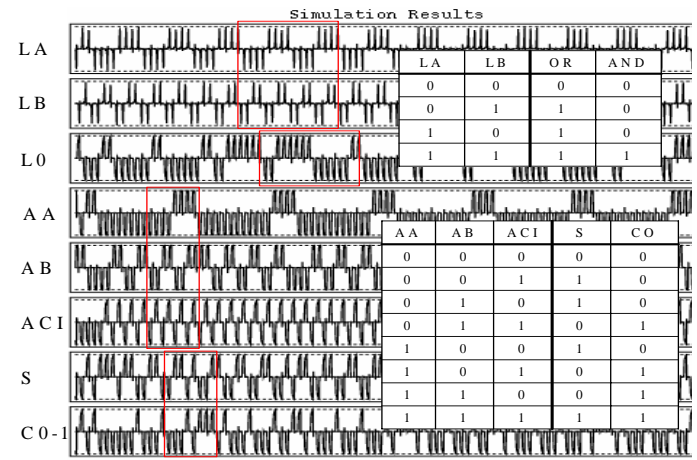| AA	AB	ACI	S	CO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 14: Simulation of the Simple 12 ALU bit slice