Defects and Faults in Quantum Cellular Automata at Nano Scale

Mehdi Baradaran Tahoori, Mariam Momenzadeh, Jing Huang, Fabrizio Lombardi
Department of Electrical and Computer Engineering,
Northeastern University, Boston, MA, 02115
{mtahoori,mmomenza,hjing,lombardi@ece.neu.edu}

***Abstract*— There has been considerable research on quantum dot cellular automata (QCA) as a new computing scheme in the nano-scale regimes. The basic logic element of this technology is majority voter. In this paper, a detailed simulation-based characterization of QCA defects and study of their effects at logic-level are presented. Testing of these devices at logic-level is investigated and compared with conventional CMOS-based designs. Unique testing features of designs based on this technology are presented and interesting properties have been identified.**

I. Introduction

The exponential scaling in feature sizes and the increase in processing power have been successfully achieved by conventional lithography based VLSI technology over the last few decades. However, this faces serious challenges, due to the fundamental physical limits of CMOS technology such as ultra-thin gate oxides, short channel effects, doping fluctuations and the increasingly difficult and expensive lithography in nano-scale regimes.

There has been extensive research in recent years at nano scale to supersede conventional CMOS technology. It is anticipated that these technologies can achieve a density of 10^{12} devices/cm^2 and operate at THZ frequencies.

One of the fundamental issues in the testing community is the radical shift in computation and fabrication technology and its effect on the test flow. Do test generation and design-for-test become even intractable? Since the manufacturing process for nano devices is ill-defined, it is extremely difficult to address manufacturing testing problems. However, it would be inappropriate to ignore testing of these device till manufacturing state.This paper tries to address this issue for one of the proposed trends in nanometer era.

Among these new devices, *quantum dot cellular automata* (QCA) not only gives a solution at nano scale, but also it offers a new method of computation and information transformation [1]. In terms of feature size, it is projected that a QCA cell of few nanometer size can be fabricated through molecular implementation by a self-assembly process.

The unique feature of QCA based designs is that logic states are not stored in voltage levels as in conventional electronics, but they are represented by the position of individual electrons.

For QCA, the cells must be aligned precisely at nano scales to provide correct functionality, so proper testing of these devices for manufacturing defects and misalignment plays a major role for quality of QCA based circuits.

The basic logic element in this technology is the majority voter. Since the basic logic elements of QCA-based designs are different from conventional CMOS designs, they need different testing schemes.

One of the interesting features of this technology is that it is possible to investigate some of the manufacturing issues (especially defects at nano scale) by quantum-mechanics simulation of these devices.

In this paper, the defect characterization of these devices has been extensively studied; effects of defects are investigated at logic-level . Also, testing of QCA is compared with testing of conventional CMOS implementations of these logic devices.

The approach proposed in this work is based on simulating different manufacturing misalignments, investigating their effects at logic level and identifying the test vectors for detection of all faults. Different fabrication schemes of a majority voter at cell level are performed; these different implementations are compared in terms of defect tolerance and testability.

As test sets generated based on the stuck-at fault model are quite acceptable for testing conventional CMOS-based designs, this model doesn't capture behavior of most of prevalent defects in CMOS fabrication process. So it is possible to investigate effectiveness of stuck-at test sets for QCA defects even though QCA defect mechanisms cannot be modeled by the stuck-at model. This is investigated in this paper.

Also, defect injection is exploited to study the behavior of QCA-based circuits in the presence of defects and to measure the effectiveness of different test sets in detecting them.

The rest of this paper is organized as follows. In Sec. II, a review of QCA is presented. In Sec. III, testing of QCA-based design at logic level is discussed. In Sec. IV, the defect characterization of QCA is presented. In Sec. V, test set, defect and fault coverage are discussed. Finally, Sec. VI concludes the paper.

II. review

QCA is a novel nano device that stores logic states not as voltage levels but rather based on the position of individual electrons. A quantum cell can be viewed as a set of four charge containers or dots, positioned at the corners of a square. The cell contains two extra mobile electrons which can quantum mechanically tunnel between dots, but not cells. The electrons are forced to the corner positions by Coulomb repulsion. The two possible polarization states represent logic 0 and logic 1, as shown in Fig 1.

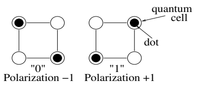


Fig. 1. QCA cell polarization

Unlike conventional logic in which information is transferred from one device to another by electrical current, QCA does so by Coulomb interaction which connects the state of one cell to the state of its neighbors. This results in a technology in which information transfer (interconnection) is the same as information transformation (logic manipulation). Power dissipation in QCA circuits is ultra low compared with conventional CMOS circuits [12][13].

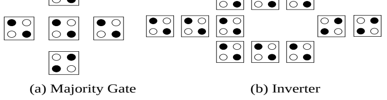


Fig. 2. QCA devices

The basic logic gate in QCA is the *majority voter*. The majority voter with logic function $MV(A, B, C) = AB + AC + BC$, can be realized by only 5 QCA cells (compared to a CMOS implementation which requires 16 transistors), as shown in Fig. 2 . Logic AND and logic OR functions can be implemented from a majority voter by setting one input permanently to 0 and 1, respectively.

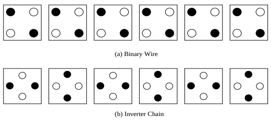


Fig. 3. Binary Wire

The binary wire (as interconnect) and the inverter are shown in Fig. 3. Unlike conventional CMOS in which it is the simplest block, it consumes considerable area in QCA. The ratio of the number of transistors for AND to NOT in CMOS technology is 6/2, while the ratio of the number of cells in QCA technology is 5/12.

The concept of clocking for QCAs has been introduced in [14]; this consists of four clock zones. QCA memories based on recursive H structures have been proposed in [18]. So sequential as well as combinational designs can be realized using QCAs. Some designs based on QCA (including a carry look ahead adder, barrel shifter, microprocessors, FPGA, and neural networks) have been proposed [4][5][6][7][8][9][10].

Currently, micro-sized QCA devices have been fabricated with metal cells which operates at 50mk [12][1]. In [12], an experimental demonstration of a basic QCA cell is presented. The device under study is composed of four metal dots, connected with tunnel junctions and capacitors. The experiments confirmed that switching of a single electron in the double dot cell can control the position of a single electron in another double dot cell. In [2], building of basic logic with these cells is demonstrated. It is predicted that molecular scale ($\sim$2nm) yields operation of QCA at room-temperature. It is also stated in [9] that room-temperature operation requires that QCA cells to be fabricated in the range of 1nm - 5nm size. [9] proposes some possible realizations of molecular QCA. It describes the progress toward making QCA molecules and working out surface attachment chemistry compatible with QCA.

There has been a study of the fault tolerant properties of the majority voter under some manufacturing misalignments [11][16][17]. Based on this simulation-based study, a fault tolerant MV block has been proposed. It has been shown that MV is more vulnerable to misalignment in the vertical direction than in the horizontal direction. A misalignment (at least equal to half a cell width in the vertical direction) causes the MV to malfunction. This confirms that a complete test of designs based on MVs is extremely needed.

III. Logic-Level Testing

The overall structure of the QCA implementation of (combinational) logic designs is shown in Fig. 4. The block consists of an interconnection of majority voters and inverters. There are two system-level control lines, U_0 and U_1, which are connected to majority voters. U_0 is connected to logic ”‘0”’ and sets some majority voters as the AND function, whereas U_1 is connected to logic ”‘1”’ and sets other majority voters as the OR function. A simple example is shown in Fig. 5. These control lines can provide more controllability since these lines can be seen as extra input lines during test time. This unique feature of QCA can be exploited to achieve higher test coverage and quality.

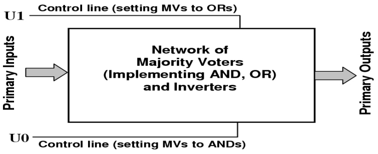


Fig. 4. The QCA implementation of logic networks using majority voters (implementing AND and OR) and Inverters

Since logic designs are implemented as a network of majority voters and inverters (as the universal logic set) in QCA technology, it is important to investigate the properties of these network, especially for test execution.As shown through the following statements, these networks have unique and interesting testing features which cannot be achieved in conventional CMOS implementations.

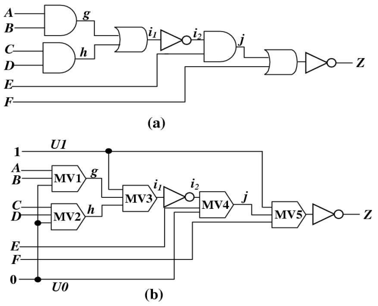


Fig. 5. (a) a simple AND-OR logic (b) MV-based implementation

Consider a majority voter with input lines A, B, and C, and the output line Z ($Z = AB + AC + BC$).

Property 1. *Consider a majority voter with input values a, b, and c, (for lines A, B, and C, respectively) and output z. If the all inputs are flipped, $abc \rightarrow a'b'c'$, the output will be also flipped, $z \rightarrow z'$.*

Note that this is not the case for other logic functions such as AND, NOR, etc. For example, consider a three input AND gate with inputs 100 and output 0. If the inputs are flipped to 011, the output will remain 0.

Property 2. *If there is inversion at any input and/or the output of the majority voter, property 1 still holds.*

Property 3. *Consider a majority voter with input pattern abc (for lines A, B, and C, respectively). The stuck-at-v fault on any input or output line of the voter is detectable (fault effect appears at output line) by abc if and only if the stuck-at-v' fault on that line is detectable by $a'b'c'$.*

Proof. Consider l stuck-at-v fault. If l is an input line, consider the l is A, without loss of generality. The fault is detected if and only if the value of a is v' and the other inputs, b and c, have opposite values. As a result, a' is v and b' and c', have opposite values. Hence, $a'b'c'$ detects the l stuck-at-v'.

Again, this property doesn't hold for other logic functions. As an example, consider an AND gate with test vector 11 which detects stuck-at-0 at the top input (and the bottom input too). The complement of this vector, 00, doesn't detect any single stuck-at-1 on the inputs.

Property 4. *If there are some inversions at any inputs and/or the output of the majority voter, property 3 still holds.*

The interesting property of majority voters is that the above properties hold for any arbitrary network of majority voters (including inverters).

Property 5. *Consider an arbitrary network of majority voters (and inverters) with primary input vector V. If all bits of V are flipped, $V \rightarrow V'$, all nodes in the network will be flipped.*

Proof. The proof is based on induction on the level (distance) of each majority voter in the network from the primary inputs, by forming a topological order of the majority voters in the network. The step of induction is property 2.

Property 6. *Consider an arbitrary network of majority voters (and inverters) with primary input vector V. For any node n in the network, n stuck-at-u is detected by V, if and only if n stuck-at-u' is detected by V'.*

Proof. The proof is similar to the proof of property 5. The step of induction is property 4.

Property 5 and 6 are very interesting and proved unique features of a network of MVs (and inverters). Based on property 5, the test vector pair (V, V'), where V is any arbitrary vector, causes a transition on all nodes of the network. Also, the three vectors (V, V', V) cause both fall and rise transitions on all nodes in the network. Hence, a 100% toggle fault coverage test set is applicable.

Based on property 6, the fault list for any network of majority voters (and inverters) can be divided into two parts: just one fault per each node, because if a vector V detects one stuck-at fault on that node, V' will detect the other stuck-at fault on that node. As a corollary, this feature can be exploited to reduce the size of the fault list, and hence ATPG execution, for the control inputs (to be generated by ATPG) into half.

To generate tests for stuck-at faults in a network of MVs and inverters, conventional (combinational) ATPG tools can be exploited. The network of MVs and inverters is first transformed into a hierarchical gate-level netlist. Each MV is replaced by a hierarchical cell implementing the majority function. We only consider pin faults on the inputs of these hierarchical cells which correspond to the inputs of MVs. As explained above, only half of the pin faults must be considered for the test generation.

IV. Defect Characterization

In this section, the robustness of QCA Majority Gates and Binary Wires, as well as some QCA circuits is investigated. The basic functionality of a QCA device is based on the Coulombic interaction among neighboring QCA cells (depending on the accuracy and geometry of its implementation). Various configurations of QCA devices have been studied using the QCADesigner[1] v1.20 simulation tool. For accuracy, the bistable model is employed. This is a quantum mechanical engine using the Jacobi algorithm to calculate the eigenvalues/vectors of the Hamiltonian matrix.

A. Defect and Failure Modes

To perform a defect characterization of QCA devices and circuits and study their effects at logic-level, appropriate defect mechanisms and modes must be considered which 1) can be simulated using the available simulation tool 2)be realistic for manufacturing and fabrication defects.

A *cell displacement* is a defect in which the defective cell is misplaced from its original direction. Several cell displacement defects are shown in Fig. 6. In a *cell misalignment* defect, the direction of the defective cell is misplaced. Some examples of cell misalignments are shown in Fig.7. In a *cell omission* defect, a particular cell is missing as compared to the original (defect-free) arrangement.

In this work, the following defects are considered and simulated for QCA devices: all possible combinations of displacement of cells with respect to the central cell for different distances, misalignment of cells in different directions, and rotation. For QCA circuits, cell omission defects are also simulated.

B. Majority Voter Defect Analysis

Consider a defect free majority voter has dot size $5nm$, cell size $20nm \times 20nm$, cell distance $5nm$, as shown in Fig. 6(1).

Different defects in the majority voter, including cell displacement and misalignment have been considered and simulated. The results for cell displacement and misalignment are shown in Table I and Table II, respectively. Only faulty entries are shown in the tables, in the form of (fault-free/faulty) values.

The data shows that in most cases the horizontal input cell (i.e. cell B) is dominant; this cell seems to have a bigger impact on the center cell than A and C. For misalignment, any single cell misalignment greater or equal to half a cell causes malfunction (fault at logic-level). In some cases the fault margin is smaller.

C. Rotated Majority Voter Defect Analysis

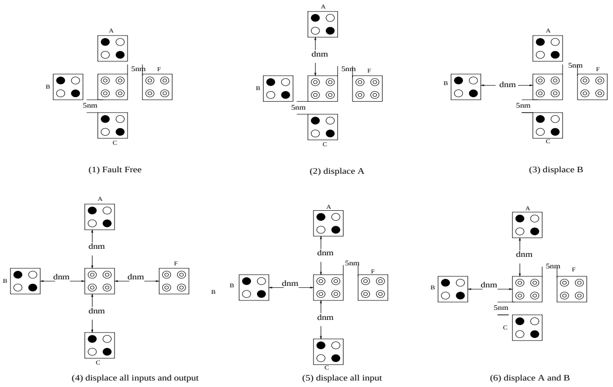
The simulation results show that the majority voter is robust with respect to rotation of all input and output cells around the center cell, i.e. the logic-level behavior of rotated majority voter is the same as the original one. Based on this observation, some simulations are performed to investigate the robustness of simple majority voters when rotated (RMG). The basic functionality of majority voters is based on the Coulombic interaction among four neighboring QCA cells, strongly depending on the precision and geometry of its implementation. We focus on validating different configurations of a majority voter in a 45^o rotation, as shown in Fig. 8.

The simulation results show that the rotated majority voterfunctions normally except when the following moves occur:

TABLE I

Results for displacement in majority voter

<table>
<tr><td colspan="4">displace cell A: fig 6(2)</td></tr>
<tr><td colspan="2">$d \leq 15nm$ Normal Operation</td><td colspan="2">$d \geq 20nm$, F=B</td></tr>
<tr><td colspan="4">displace cell B: fig 6(3)</td></tr>
<tr><td colspan="2" rowspan="6">$d \leq 40nm$ Normal Operation</td><td colspan="2">$d \geq 45nm$</td></tr>
<tr><td>A B C</td><td>F</td></tr>
<tr><td>001</td><td>Z(no polarization)</td></tr>
<tr><td>011</td><td>Z(no polarization)</td></tr>
<tr><td>100</td><td>Z(no polarization)</td></tr>
<tr><td>110</td><td>Z(no polarization)</td></tr>
<tr><td colspan="4">displace all input/output cells: fig 6(4)</td></tr>
<tr><td colspan="2" rowspan="2">$d \leq 10$ or $30 \leq d \leq 40nm$
Normal Operation</td><td colspan="2">$15 \leq d \leq 25nm$</td></tr>
<tr><td>A B C</td><td>F</td></tr>
<tr><td colspan="2" rowspan="2">$d \geq 45nm$
F=Z(no polarization)</td><td>010</td><td>0/1</td></tr>
<tr><td>101</td><td>1/0</td></tr>
<tr><td colspan="4">displace all input cells: fig 6(5)</td></tr>
<tr><td colspan="2" rowspan="4">$d \leq 15$ or $d = 40nm$
Normal Operation</td><td colspan="2">$20 \leq d \leq 25$ or $d = 35nm$</td></tr>
<tr><td>A B C</td><td>F</td></tr>
<tr><td>010</td><td>0/1</td></tr>
<tr><td>101</td><td>1/0</td></tr>
<tr><td colspan="2">$d = 30nm$</td><td colspan="2" rowspan="6">$d \geq 45nm$
F=Z(no polarization)</td></tr>
<tr><td>A B C</td><td>F</td></tr>
<tr><td>000</td><td>0/1</td></tr>
<tr><td>010</td><td>0/1</td></tr>
<tr><td>101</td><td>1/0</td></tr>
<tr><td>111</td><td>1/0</td></tr>
<tr><td colspan="4">displace cells A and B: fig 6(6)</td></tr>
<tr><td colspan="2">$d \leq 5nm$ Normal Operation</td><td colspan="2">$d \geq 10nm$, F=C</td></tr>
</table>



(1) Fault Free

(2) displace A

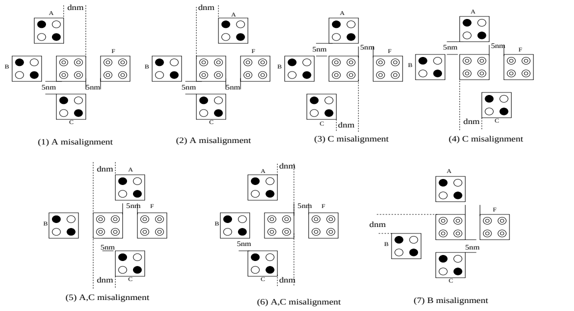
(3) displace B

(4) displace all inputs and output

(5) displace all input

(6) displace A and B

Fig. 6. Displacement in Majority Voter



(1) A misalignment

(2) A misalignment

(3) C misalignment

(4) C misalignment

(5) A,C misalignment

(6) A,C misalignment

(7) B misalignment

Fig. 7. Misalignment in Majority Voter

[1] QCADesigner is the product of an ongoing collaboration between the University of Calgary ATIPS Laboratory and the University of Notre Dame.

TABLE II

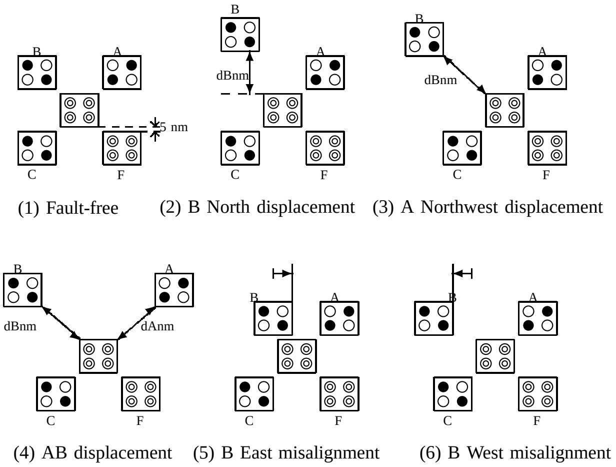
RESULTS FOR MISALIGNMENT IN MAJORITY VOTER

move A towards west: fig 7(1)			
$d \leq 5nm$ Normal Operation		$d \geq 10nm$, F=B	
move A towards east: fig 7(2)			
$5 \leq d \leq 15nm$		$d = 20$ or $d = 30nm$	
A B C	F	Normal Operation	
001	0/1		
010	0/1	$d = 25nm$	
101	1/0	F=A	
110	1/0		
move C towards west: fig 7(3)			
$d \leq 5nm$		$d \geq 10nm$	
Normal Operation		F=B	
move C towards east: fig 7(4)			
$5 \leq d \leq 15nm$		$d = 20$ or $d = 30nm$	
A B C	F	Normal Operation	
010	0/1		
011	1/0	$d = 25nm$	
100	0/1	F=C	
101	1/0		
move A,C towards west: fig 7(5)			
$d \geq 5nm$ F=B			
move A,C towards east: fig 7(6)			
$d = 5, 20$, $d \geq 30nm$		$10nm \leq d \leq 15nm$	
F=B		A B C	F
$d = 25nm$		000	0/1
Normal Operation		010	0/1
		101	1/0
		111	1/0
move B towards south/north: fig 7(7)			
$d \leq 5nm$		$d \geq 45nm$	
Normal Operation		A B C	F
		001	0/1
		011	1/0
		100	0/1
		110	1/0

- A input north, with dA$\geq$ $10nm$ for ABC $= 001, 110$ (output follows C input). A similar output appears when moving A to northeast with dB$\geq$ $10\sqrt{2}nm$.
- B input north, with dB$\geq$ $40nm$. The output is unknown (unpolarized) for ABC $= 001, 011, 100, 110$. A similar output appears when moving B to the northwest with dB$\geq$ $30\sqrt{2}nm$.
- C input south, with dC$\geq$ $15nm$ for ABC $= 011, 100$ (output follows A input). A similar output appears when moving C to the southwest with dC$\geq$ $10\sqrt{2}nm$.
- A, B, C or A, B, C, F away for d$\geq$ $30\sqrt{2}nm$. The output is unknown (Z) for all input combination with such a distance.
- A and B inputs away with d$\geq$ $10\sqrt{2}nm$ for ABC $= 001, 110$ (the output follows the C input).
- A and C inputs away with d$\geq$ $10\sqrt{2}nm$ for ABC $= 010, 101$ (the output follows the B input).
- B and C inputs away with d$\geq$ $10\sqrt{2}nm$ for ABC $= 011, 100$ (the output follows the A input).

Cell misalignment defects for rotated majority voterhave also been considered. Two of these misalignment examples (for B input) are illustrated in Fig. 8. The following shows the results for these misalignments:

- Shifting input A west (half/full cell size), leads the output F to follow input A, while shifting A east effects the output such that it follows the input C values.
- The majority voter functions normally when input B is shifted west for half or full cell size. However, the output becomes unknown for inputs ABC $= 001, 011, 100, 110$ when dB$\geq$ $40nm$.
- The output follows input B when B is shifted east for half or full cell size.
- A similar trend is seen when input C is shifted to west or east: output follows A when C is shifted west, and it follows C when C is shifted east.



(1) Fault-free (2) B North displacement (3) A Northwest displacement

(4) AB displacement (5) B East misalignment (6) B West misalignment

Fig. 8. Rotated Majority Voter: Fault-free, with displacement or misalignment.

1) OMG and RMG comparison: The results for different configurations of the Original majority voter and the Rotated majority voter are illustrated and compared in Table III.The majority voter is completely robust with respect to rotation of all inputs and output cells around the central cell. This gives a significant degree of freedom for synthesizing designs based on QCA as rotated majority voter can be used as the original majority voter block. However, the original block is more dependent on the middle input (B) than the other inputs (A and C), both in terms of displacement and misalignment. In the rotated version, this dependency can be completely changed based on the degree of rotation. An overall comparison in the table asserts that the rotated majority voter is more fault-tolerant than the original majority voter. Note that only half and full misalignmnets are considered.

TABLE III

ORIGINAL MAJORITY VOTER VS. ROTATED MAJORITY VOTER

Config.	Faults	OMV	RMV
A move	distance	d$\geq$ $20nm$	d$\geq$ 10(N) or $10\sqrt{2}nm$ (NE)
	# of faults	2	2
B move	distance	d$\geq$ $45nm$	d$\geq$ 40(W) or $30\sqrt{2}nm$ (NW)
	# of faults	4	4
C move	distance	d$\geq$ $20nm$	d$\geq$ 10(S) or $10\sqrt{2}nm$ (SW)
	# of faults	2	2
ABC move	distance	20 $\leq$d$\leq$ 35 or d$\geq$ $45nm$	d$\geq$ $30\sqrt{2}nm$
	# of faults	2/4/8	8
ABCF move	distance	15 $\leq$d$\leq$ 25 or d$\geq$ $45nm$	d$\geq$ $30\sqrt{2}nm$
	# of faults	2/8	8
AB move	distance	d$\geq$ $7.5nm$	d$\geq$ $10\sqrt{2}nm$
	# of faults	2	2
AC move	distance	d$\geq$ $7.5nm$	d$\geq$ $10\sqrt{2}nm$
	# of faults	2	2
Z move	distance	d$\geq$ $45nm$	d$\geq$ $30\sqrt{2}nm$
	# of faults	8	8
A/C misalignment	# of faults	4	4
B misalign. West	# of faults	4	0
B misalign. East	# of faults	4	2

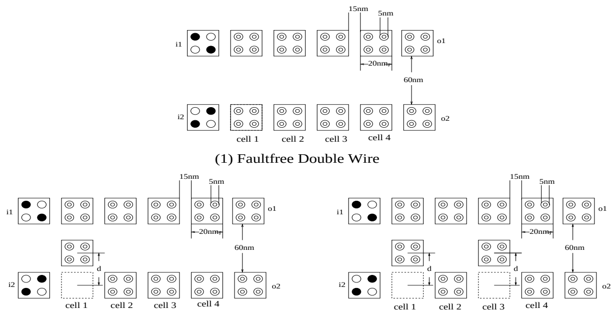
D. Binary Wires and Inverter Chains

The effect of cell displacement defects on two parallel binary wires as well as two parallel inverter chains have been investigated.

1) Double Binary Wire: Two defect-free binary wires are shown in Figure 9(a);these wires are denoted as the upper wire ($i1$ to $o1$) and the lower wire ($i2$ to $o2$). The cells used in this simulation have size $20nm \times 20nm$, the dot diameter is $5nm$. In the defect-free case, the cells in the same wire are separated by $15nm$. The distance between the wires is $60nm$.

The displacement defects are simulated by moving one or two cells in the lower wire towards the upper wire, by a displacement d, as shown in Figure 9(b).

The simulation results are shown in Table IV. These results show that in most cases the lower wire is dominated by the upper wire. $o1$ and $o2$ are either equal to $i1$ or $\overline{i1}$, depending on which cell(s) are displaced and the value of the displacement, d. In most cases the upper wire functions normally, i.e. $i1 = o1$. However, it can be observed that in some cases the upper wire behaves as an inverter. Clearly, unlike CMOS designs, the coupling defects at QCA device-level do not behave as *wired bridging fault* model. However, these defects manifest themselves as a dominant model at logic level, in which the output of a wire is determined by the value of the coupled wire.

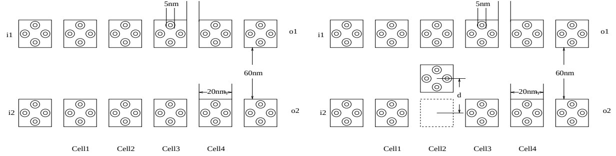


(1) Faultfree Double Wire

(2) Defects in Double Wire

Fig. 9. Displacment in Binary Double Wire

2) Double Inverter Chains: The double inverter chain is shown in Figure 10(a). The cells used in this simulation have size $20nm \times 20nm$, the dot diameter is $5nm$. In the fault free case, each cell in the same wire is separated by $15nm$, the distance between the wires is $60nm$ (the same as the simulation for the double binary wires). The simulation results for moving one of the cells in the bottom wire towards the upper wire, with displacement d, (as shown in Figure 10(b)) are presented in Table V. It can be concluded that the displacement defects behave as according to the *dominating bridging fault* model at logic level. Moreover, by comparing the these results with those for binary wires, binary wires are more defect tolerant than the inverter chains in the case of displacement coupling defects.



(a) Fault Free Inverter Chain

(b) Single Cell Displacement Defect

Fig. 10. Displacment in Double Inverter Chains

E. Defects and Faults in a Full-Adder

A QCA implementation of a full adder using three majority voters and two inverters is shown in Fig.11. The corresponding QCA layout is shown in Fig.12 which contains 145 cells. The cells are 18nm × 18nm with dot size of 5nm. 40 different single cell omission defects have been simulated in this circuit.

TABLE IV

RESULTS FOR DOUBLE BINARY WIRE

moving cell1 OR cell2			
$d \leq 40nm$ Normal	$d = 45 - 50nm$ $o1 = i1$ $o2 = i1$	$d \geq 55nm$ $o1 = i1$ $o2 = Z$	
moving cell3 OR cell4			
$d \leq 35nm$ Normal	$d = 40 - 50nm$ $o1 = i1$ $o2 = \overline{i1}$	$d \geq 55nm$ $o1 = i1$ $o2 = Z$	
moving cell1 AND cell2			
$d \leq 35nm$ Normal	$d = 40 - 50nm$ $o1 = i1$ $o2 = i1$	$d \geq 55nm$ $o1 = i1$ $o2 = Z$	
moving cell1 AND cell3			
$d \leq 35nm$ Normal	$d = 40 - 50nm$ $o1 = i1$ $o2 = i1$	$d = 45nm$ $o1 = i1$ $o2 = \overline{i1}$	$d \geq 55nm$ $o1 = i1$ $o2 = Z$
moving cell1 AND cell4; OR moving cell 2 AND cell 3; OR moving cell3 AND cell4			
$d \leq 35nm$ Normal	$d = 40 - 50nm$ $o1 = i1$ $o2 = \overline{i1}$	$d \geq 55nm$ $o1 = i1$ $o2 = Z$	
moving cell2 AND cell4			
$d \leq 15nm$	$d = 20 - 25nm$ $d = 40 - 45nm$	$d = 30 - 35nm$	$d = 50nm$
Normal	$o1 = i1$ $o2 = \overline{i1}$	$o1 = i1$ $o2 = i1$	$o1 = \overline{i1}, o2 = i1$ $o2 = i1$
$d \geq 55nm$ $o1 = i1$ $o2 = Z$			

TABLE V

RESULTS FOR DOUBLE INVERTER CHAINS

fault free		
$o1 = \overline{i1}; o2 = \overline{i2}$		
moving cell1 OR cell2 OR cell3		
$d \leq 35nm$ Normal	$d = 40nm - 50nm$ $o1 = \overline{i1}$ $o2 = \overline{i1}$	$d \geq 55nm$ $o1 = \overline{i1}$ $o2 = Z$
moving cell4		
$d \leq 30nm$ Normal	$d = 35nm - 50nm$ $o1 = \overline{i1}$ $o2 = \overline{i1}$	$d \geq 55nm$ $o1 = \overline{i1}$ $o2 = Z$

1) Defects in Wires and Inverter Chain: Removing a single cell from a binary wire doesn't affect its functionality at logic-level although it may result in some delay faults. However, a single cell omission in a wire implemented as an inverter chain results in an unwanted complementation at the output of the chain.

Those binary wires which change direction in the layout (e.g. L shape) are very sensitive to the defects on the corner cells. Cell omission defect at the corner cell is equivalent to unwanted complementation fault at logic-level.

2) Defects in Wire Crossing: In QCA implementation, two different wires (horizontal and vertical) can cross each other in the same layer (*co-planar wire crossing*). In this case, one of them is implemented as a binary wire, while the other one is implemented as an inverter chain (i.e. the cells in the other wire are rotated). In the fault-free case, the wires are unaffected by each other and can carry different signal values.

However, this structure is very vulnerable to cell omission defects at or near to the crossing point. The cell omission defect at the cross point results in an unwanted complementation on the inverter chain and the binary wire is *dominated* by the faulty value of the inverter chain (dominating bridging fault). Cell omission defects for the cells adjacent to the crossing point have similar effects, i.e. the value of the binary wire

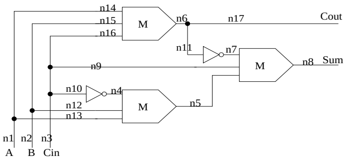


Fig. 11. One-bit QCA full adder

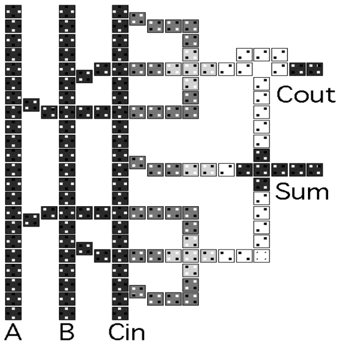


Fig. 12. one-bit QCA full adder layout

is dominated by the faulty value of the inverter chain.

3) Defects in the Majority Voter: The results of defects in a majority voter of the full-adder is consistent with the defect characterization results for a single majority voter: the horizontal input has more impact on the output than the vertical inputs. Cell omission defect on the horizontal input cell doesn't affect the functionality. However, a cell omission defect on any of vertical inputs causes the output to be dominated only by the horizontal input, i.e. the output is shorted to the horizontal input.

The cell omission defect on the center cell of a majority voter with vertical input values a and b, and horizontal input c changes the function to be the majority of a', b', and c. This can be interpreted as unwanted complementation faults on both vertical inputs.

V. Test Sets Coverage and Fault Model

The effectiveness of different stuck-at test sets have been evaluated for the simulated defects on a single majority voter. The following are the main results of this evaluation:

- In all simulations, *super exhaustive* input patterns (i.e. all possible input transitions) are used. Our data show that there is no sequence dependent behavior at logic level; i.e. none of the manufacturing misalignments introduce a state dependency at logic level.
- Except for a single case (i.e. the displacement of all inputs and output cells) faults are detected using a subset of some 100% stuck-at fault test sets. Note that not all of these 100% stuck-at test sets are equal.
- A particular 100% 2-detect stuck-at test set (each fault is detected by two vectors) can detect all manufacturing defects, except for one case, i.e. the simultaneous displacement of the top and left inputs.
- Moreover, a particular 100% single stuck-at test set (001,010,011,101) can detect all simulated defects.

The results for the full-adder circuit shows that none of the defects behave as stuck-at faults at logic-level. However, cell omission defects in wires implemented as inverter chains mainly result in *unwanted complementation* faults in which at extra inverter is present in the faulty wire. Cell omission defects at corner cells in the binary wires also behave this way.

We also considered stuck-at test sets for the full-adder (Fig.11) and computed the corresponding defect coverage with respect to cell omission defects. Note that for a full-adder, any two vectors $\{(a,b,c),(a',b',c')\}$ will result in 100% PIN stuck-at fault coverage. For example, $\{(010),(101)\}$ is a 100% PIN stuck-at coverage. However, this test set can detect only 17 out of 28 cell omission defects (Note that 12 of 40 simulated cell omission defects do not affect its functionality). By considering all internal nodes (n1 to n17 in Fig.11), $\{000,001,011,100,101\}$ is a 100% single stuck-at test set. This test set can detect all $40-12=28$ detectable defects. This shows that the specific QCA implementation must be considered for test generation to achieve a high defect coverage.

VI. Conclusion

Quantum dots cellular automata (QCA) are novel devices which are promising in the era of nano scale computing. In this paper, testing of QCA based designs has been investigated. A detailed defect characterization for QCA basic logic devices and some representative circuits has been presented. As shown in this paper, the coupling mechanisms and behavior of defects at logic-level (i.e. faults) are not similar to those in a conventional CMOS fabrication process. For example, an *Unwanted complementation* fault at logic-level has been observed for a considerable number of cases of *cell omission* defects. Hence, appropriate fault models for QCA must be developed and used for test generation.

The effectiveness of different stuck-at test sets in detecting QCA defects has been studied. Our results show that to achieve high defect coverage, the specific QCA implementations of each function must be considered for test generation. Some interesting and unique properties of QCA implementation of logic networks have been investigated. As shown in this paper, a network of majority voters (and inverters) has unique testing properties: Any (V, V', V) test set achieves 100% toggle fault coverage. And V detects n stuck-at-u if and only if V' detects n stuck-at-u' .

References

[1] I.Amlani, A.O.Orlov, G.L.Snider, C.S.Lent, "Demonstration of a Six-dot Quantum Cellular Automata System", *Applied Physics Letters*, vol 72, pp. 2179-2181, 1998.

[2] I. Amlani, A.O. Orlov, G. Toth, C.S. Lent, G.H. Bernstein, and G.L. Snider ,"Digital Logic Gate Using Quantum-Dot Cellular Automat", *Science* 284 (5412): 289-291

[3] C.D.Armstrong, W.M.Humphreys, A.Fijany, " The Design of Fault Tolerant Quantum Dot Cellular Automata Based Logic ", *11th NASA Symposium on VLSI Design*, 2003

[4] C.D.Armstrong, W.M.Humphreys, "The Development of Design Tools for Fault Tolerant Quantum Dot Cellular Automata Based Logic", *2nd International Workshop on Quantum Dots for Quantum Computing and Classical Size Effect Circuits*, 2003.

[5] A. Fijany and B. N. Toomarian, "New design for quantum dots cellular automata to obtain fault tolerant logic gates", *Journal of Nanoparticle Research*, vol. 3, pp. 27-37, 2001.

[6] S.E.Frost, A.F.Rodrigues,A.W.Janiszewski,R.T. Rausch,P.M.Kogge, "Memory in Motion: A Study of Storage Structures in QCA", *1st Workshop on Non-Silicon Computation*, 2002.

[7] M.Governale, M.Macucci, G.Iannaccone, C.Ungarelli, "Modeling and Manufacturability Assessment of Bistable Quantum-Dot Cells", *Journal of Applied Physics* vol 85, pp. 2962 ,1999.

[8] K.Hennessy, C.S.Lent, "Clocking of Molecular Quantum-Dot Cellular Automata", *Journal of Vaccum Science and Technology*, vol 19(5), pp. 1752-1755, 2001.

[9] M.Lieberman, S.Chellamma, B.Varughese, Y.Wang, C.S.Lent, G.H.Bernstein, G.Snider, F.Peiris, "Quantum-Dot Cellular Automata at a Molecular Scale", *Annals of the New York Academy of Sciences*, vol 960, pp. 225-239, 2002.

[10] M.T.Niemier, A.F.Rodrigues, P.M. Kogge, "A Potentially Implementable FPGA for Quantum Dot Cellular Automata", *1st Workshop on Non-Silicon Computation*, 2002.

[11] M.T. Niemier, P.M.Kogge, "Logic-in-Wire: Using Quantum Dots to Implement a Microprocessor", *International Conference on Electronics, Circuits, and Systems (ICECS '99)*,1999.

[12] A.O.Orlov, I.Amlani, G.H.Bernstein, C.S.Lent, G.L.Snider, "Realization of a Functional Cell for Quantum-Dot Cellular Automata", *Science*, vol 277, pp 928-930, 1997.

[13] QCADesigner Home Page: www.atips.ca/projects/qcadesigner/

[14] R.Compano, L.Molenkamp, D.J.Paul, "Technology Roadmap for Nanoelectroincs", *European Commission IST programme, Future and Emerging Technologies*.

[15] C.G.Smith, "Computation Without Current", *Science*, vol 284, pp. 274, 1999.

[16] P.D. Tougaw and C.S. Lent, "Logical Devices Implemented Using Quantum Cellular Automata", *Journal of Applied Physics*, Vol 75(3), pp. 1818-1825, 1994.

[17] P.D.Tougaw, C.S.Lent, "Dynamic Behavior of Quantum Cellular Automata", *Journal of Applied Physics*, vol 80(8), pp. 4722-4736, 1996.

[18] V. S. Dimitrov, G. A. Jullien, K. Walus, "Quantum-Dot Cellular Automata Carry-Look-Ahead Adder and Barrel Shifter", *IEEE Emerging Telecommunications Technologies Conference*, 2002.

[19] K. Walus, R. A. Budiman, and G. A. Jullien, "Effects of morphological variations of self-assembled nanostructures on quantum-dot cellular automata (QCA) circuits", *Frontiers of Integration, An International Workshop on Integrating Nanotechnologies*, 2002.

[20] K.Walus, A.Vetteth, G.A.Jullien, V.S.Dimitrov, "RAM Design Using Quantum-Dot Cellular Automata", *NanoTechnology Conference*,vol 2, pp. 160-163, 2003.