Testing of Quantum Dot Cellular Automata Based Designs

Mehdi Baradaran Tahoori and Fabrizio Lombardi
Northeastern University
Boston, MA 02115
{mtahoori,Lombardi}@ece.neu.edu

Abstract

There has been considerable research on quantum dots cellular automata as a new computing scheme in the nano-scale regimes. The basic logic element of this technology is a majority voter. In this paper, testing of these devices is investigated and compared with conventional CMOS-based designs. A testing technique is presented; it requires only a constant number of test vectors to achieve 100% fault coverage with respect to the fault list of the original design. A design-for-test scheme is also presented which results in the generation of a reduced test set.

1. Introduction

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.

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.

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. As shown in this paper, a 100% single stuck-at fault test set for designs mapped into QCAs doesn't necessarily detect all stuck-at faults in majority voters. However, the unique features of designs implemented by majority voters enable to use a reduced test set for 100% stuck-at coverage.

In this paper, testing of QCA based designs is studied and unique testing properties of this technology have been identified. An efficient test generation approach has been proposed. Also, a design-for-test scheme is introduced to improve testability of these designs.

2. Review

QCA is a novel 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.a.

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, as shown in Fig. 1.b. Logic AND and logic OR functions can be implemented from a majority voter by setting one input permanently to 0 and 1, respectively.

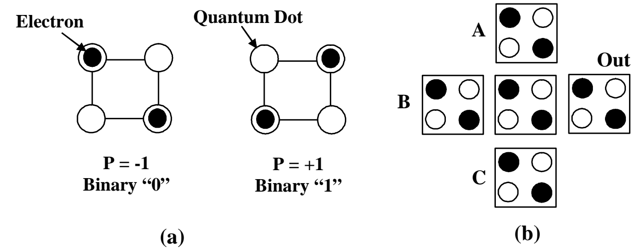

Figure 1 (a) Binary representation in QCA cells (b) Majority Voter

There has been a study of the fault tolerant properties of the majority voter under some manufacturing misalignments [2][3]. A misalignment (at least equal to half a cell width in the vertical direction) causes the MV to malfunction. This shows that a complete test of designs based on MVs is extremely crucial.

3. Test Set for Majority Voters

Consider a simple AND-OR structure shown in Fig.2a and a possible implementation using MVs in Fig. 2b. Note that there is no built-in VDD or ground lines in quantum dots based designs. There are two extra inputs connected to logic "1" and logic "0" in order to connect some selected inputs of MV to implement AND and OR logic functions. We call these inputs as *control line*. The input line of MV, which is connected to a control line, is called *control input* (the control line is the fanout stem and the control inputs of the MVs are fanout branches connected to a control line). The other inputs are called *non-control inputs*.

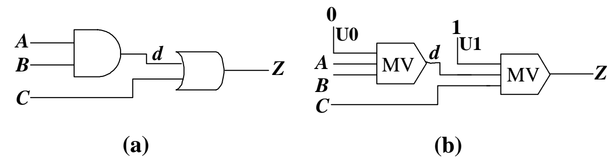

Figure 2 (a) an AND-OR circuit (b) implementation by MVs

The exhaustive testing of the circuit in Fig. 2.a needs all 8 combinations of the three inputs. The minimum test set with 100% single stuck-at fault coverage for this circuit contains four vectors. These vectors are ABC= (010, 100, 101, 110). However, 100% stuck-at coverage for the same fault list contains only two vectors for the implementation using MVs, shown in Fig. 2b. These vectors are ($ABCU_0U_1$) = (11100, 00011). In the first test vector, both control inputs, U_0 and U_1, are connected to 0. This vector detects A/0, B/0, C/0, d/0, and Z/0. The second input connects all control inputs to 1 and sets the primary inputs, A, B, and C, to 0. As a result, the MVs implement OR functions. This vector detects all stuck-at-1 faults.

Note that any 100% stuck-at coverage test set for the original design, Fig. 2.a, detects no stuck-at faults on the control lines of MVs, namely $U_0/1$, $U_0/0$, $U_1/1$, $U_1/0$.

Testing of a control line of MV for stuck-at faults requires that the two other inputs of MV have opposite values. By applying 1 and 0 on the control line, stuck-at-0 and stuck-at-1 faults on the control line will be detected, respectively.

If for a particular vector at the primary inputs, the two non-control inputs of each MV have different values, all stuck-at faults on control lines can be detected by only two test vectors. In the above example, the following two vectors must be added to detect control line faults: $(ABCU_0U_1) = (10011, 01100)$.

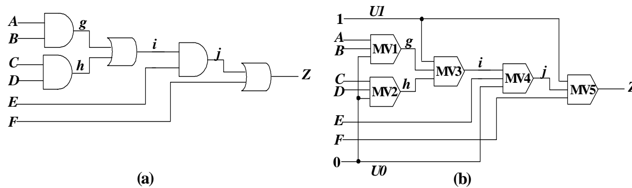

Figure 3 (a) network of AND-OR (b) MV implementation

However, testing for stuck-at faults on control lines cannot be always done in two test vectors. Consider the circuit shown in Fig. 3.a and the corresponding MV-based implementation in Fig. 3.b. There is no way to set the non-control inputs of MV1, MV2 and MV3 to opposite values at the same time (i.e. $A \neq B$, $C \neq D$, and $g \neq h$) because the control inputs of MV1 and MV2 are connected to the same control line. This results in more than two test vectors for stuck-at faults on all control inputs and lines.

4. General Test Set for a Network of MVs

Consider a logic network, composed only of AND and OR gates, which is implemented using QCA MVs. The general case in which NOT gates are also used (universal set) will be covered in the next section. The QCA network has two extra control inputs other than the primary inputs. To detect all single stuck-at faults with respect to the fault list in the original design, only two test vectors are needed for the MV implementation. The first test vector sets all primary inputs to "0" and two control inputs to "1". The second test vector sets all primary inputs to "1" and the control inputs to "0".

Lemma. The first test vector, as described above, detects all stuck-at-1 faults while the second vector detects all stuck-at-0 faults, with respect to the fault list in the original design.

Proof. Since in the first vector, the control inputs are set to 1, all MVs act as OR functions. As a result, by applying 0 on all primary inputs, any stuck-at-1 fault on any node produces an incorrect 1 on the primary output(s) and the fault will be detected. The second vector is the dual of the first one, in which all MVs behave as AND functions and by applying 1 on all primary inputs, any incorrect 0 on any node will produce an incorrect 0 at the primary output and the fault will be detected.

Note that these two test vectors guarantee 100% stuck-at fault coverage with respect to the fault list of the original design and do not detect stuck-at faults on the control inputs. In order to detect stuck-at faults on the control lines and control inputs, conventional (combinational) ATPG tools can be exploited to generate test for these fault. The network of MVs 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 controls of MVs.

This fault list is much smaller than the complete fault list for 100% stuck-at coverage (stuck-at fault on all nodes), since all nodes in the original design are covered by those two test vectors and removed from the fault list in this phase. This results in a reduction in the test generation time and the number of test vectors.

5. Design-for-Testability of QCA

Universal logic is considered here. In order to use the test vector pair presented in Sec. 4, DeMorgan's law can be exploited to change the structure of the design such that all inversions are pushed back to the primary input level. It is always possible to transform a general logic network of AND, OR, and NOT functions (universal logic) into an equivalent network consisting only of AND and OR functions which take literals (variables and their complements) as inputs. Therefore, the network of AND and OR functions can be implemented only by MVs and the same test generation approach presented in the previous section can be still exploited. The structure of these designs is shown in Fig. 4. The design is partitioned into two blocks: the first block is the inverting block which generates the literals, which is implemented by QCA inverter chains, followed by a block of MVs which implements the AND-OR network. Note that QCA inverter chains used at the input level take the same area as binary wire, while the area of an inverter inside the network is twice as an MV.

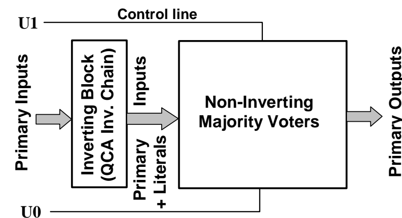

Figure 4 Architecture with separate blocks for inverters and MVs

6. 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.

It has been shown that only two test vectors on a QCA-based implementation detect all stuck-at faults with respect to the fault list on the original design. A technique is also presented to detect the remaining faults, such as the faults in the control lines of voters. A design-for-test scheme is presented based on a change in the structure of the design; the design is partitioned into a block of inverters and a block of majority voters. This structure results in a reduced test generation effort and test length.

References

[1] Tougaw, P. Douglas, Lent, Craig S , *Logical devices implemented using quantum cellular automata,* Journal of Applied Physics 75(3): 1818-25 FEB 1 1994.

[2] A. Fijany, B. N. Toomarian, *New Design for Quantum Dots Cellular Automata to Obtain Fault Tolerant Logic Gates*, Journal of nanoparticle Research, 3: pp. 27-37, 2001.

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