# Fault Analysis of QCA Combinational Circuit at Layout & Logic Level

Vaishali Dhare[1], Usha Mehta[2]

[1] *Assistant Professor, Institute of Technology, Nirma University, Ahmedabad, Gujrat, India*
[2] *Senior Member, IEEE, Professor, Institute of Technology, Nirma University, Ahmedabad, Gujrat, India*
[1]vaishali.dhare@nirmauni.ac.in [2]usha.mehta@nirmauni.ac.in

***Abstract*—QCA (Quantum-dot Cellular Automata) is the most capable future nanotechnology for computing. Defects are most likely to occur in QCA devices due to the nanoscale. Faults caused by these defects must be analyzed. This paper implement the QCA combinational circuit, half adder for which fault analysis is carried out. This paper presents the fault analysis of QCA combinational circuit, half adder at layout level using QCADesigner tool and at logic level using Hardware description Language for QCA (HDLQ).**

***Index Terms*—quantum-dot, QCA, defect, HDLQ.**

## I. INTRODUCTION

Conventional Complementary Metal Oxide Semiconductor (CMOS) technology will reach to its limit in nearby future. Ultra-thin gate oxides, doping fluctuations and short channel effects are occurring in this technologies if it goes to nanoscale. Single Electron Transistor (SET) [1], Quantum-dot Cellular Automata (QCA) [2] and Resonant Tunneling Diodes (RTD) [3] are the alternate nanotechnologies to CMOS technology. Among all nanotechnologies, Quantum-dot Cellular Automata is most capable technology [2]. QCA is able to implement high density, low power and high speed circuits and system. Device density of $10^{12}$ devices/cm$^2$ is possible in QCA. Also it can operate at THZ frequencies. Transistor less computation is possible in QCA with high device density and speed [4]. QCA is the array of quantum cells in which information is hold in. Transfer of information is possible due to the Coulombic coupling and interaction between cells [5].

At nanoscale, there are chances of defects in QCA devices and systems. Various defect analysis and its effects on QCA devices are carried out in [6-8]. This paper presents the implementation of simple QCA combinational circuit, half adder using QCADesigner [9]. Further the defect and fault analysis are carried out at layout level using QCADesigner. Also fault analysis is carried out at logic level using Hardware Description Language for QCA (HDLQ) to match the results.

The contents of rest of the paper is as follows, section II describes the fundamental concepts of QCA. We discuss the classification of defects in section III. Section IV discusses the features of HDLQ. Implementation is presented in section V. Section VI and VII presents the fault analysis at

layout and logic level respectively with results. Paper is concluded in section VIII.

## II. FUNDAMENTALS OF QCA

As shown in Fig. 1. QCA cell consists of four quantum dots. Cell configuration with five and six quantum dots are also exists. All four dots are positioned at the corners of a square cell. Each cell have two extra mobile electrons. These electrons can tunnel between the quantum dots of the cell [10].

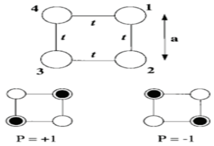


Fig. 1. A schematic of a four-dot QCA cell [10]

Electrons within the cells always occupy antipodal sites due to Coulombic repulsion. If electrons occupied the antipodal site as shown in Fig.1. left side cell polarizations is considered as P=+1. If electrons occupied the antipodal site as shown in Fig.1. right side cell polarizations is considered as P=-1. These polarizations P=+1 and P=-1 are encoded as binary ‘1’ and ‘0’ respectively.

The types of QCA implementations are magnetic, metal-islands, molecular and semiconductor [10] [11] [12] [13]. Magnetic and metal-island QCA can achieve the speed up to MHz while semiconductor and molecular QCA can achieve the speed up to THz. Lots of research in development and fabrication process have been done. Molecular QCA is the most promising type in which single molecular can act as a cell. Small size molecular QCA cells are fabricated using self-assembly process [14].

The basic building blocks of QCA are majority voter, inverter and binary wire [15]. Majority Voter (MV) consists of three inputs and one output. It is shown in Fig. 2. The Boolean function of the majority gate is given as F = AB + BC + AC. The output of the majority gate is determined by the majority of its three inputs. If two of the inputs are low, the output will be a low. If two of the inputs are high, then output will be a high. Here high refers to the polarization state P = +1, and low refers to the polarization state P = -1.

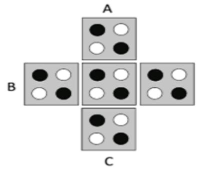


Fig. 2. QCA majority gate [15]

An AND gate and an OR gate can be easily built using a majority gate with one input fixed to either low or high respectively. Another fundamental logic gate in QCA with one input and one output is inverter shown in Fig. 3. It takes the input logic and produces its inverse logic on output. The signal from left splits into two binary wire and get inverted at the point of convergence. Many implementations are possible for inverter but the given one is robust implementation. Reliable inversion of logic ‘0’ and ‘1’ is possible in this implementation because of geometrical symmetry.

978-1-4673-8786-6/15/$31.00 ©2015 IEEE

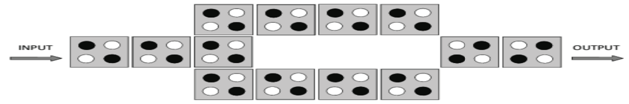


Fig. 3. QCA inverter [15]

A binary wire is used to transfer information from one part of the circuit to another. In a QCA circuit, a wire not only helps in information transfer, it actually can performs some computational operation on the information to be transferred. The two basic types of QCA wire namely normal wire and the inverter chain are shown in Fig. 4 and Fig. 5. respectively.

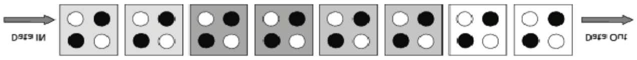


Fig. 4. Binary QCA wire [15]

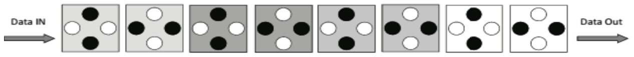


Fig. 5. An inverter 45$^o$ chain [15]

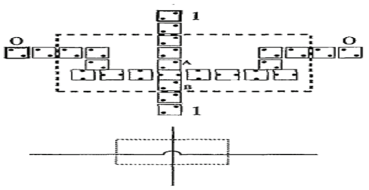


Fig. 6. Crossover wires [15]

Coplanar crossover wires are possible in QCA because of its cellular nature. Coplanar crossover is impossible in conventional circuits. Fig.6 shows the two coplanar crossing of two QCA binary wires [15].

Clocking is important in QCA circuit in which clock signal is given to each cell. CMOS wires buried under the QCA surface is the clocking source. Contrasting to conventional CMOS clock, QCA clock has four phases other than high and low values. Clock provides power gain to the QCA circuits [16]. Clocking is done by electrostatically switching the cell from a null state to the locked state through the switching state. In null state, cell doesn't hold any binary information. Cell state is determined by its neighbors in switching state. Finally it adopts the neighbor's cell polarization in locked state [17]. QCA clocking mechanism consists of four clock signals with equal frequencies. (Phase=0) is considered as the reference clock signal. Other clock signals are delayed one (phase = π/2), two (phase = π) and three (phase = 3π/2) quarters of a period as shown in Fig. 7.

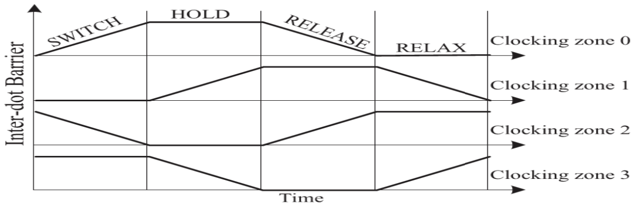


Fig. 7. Clocking scheme [16]

Each clock zone contains four phases namely switch, hold, release and relax. The tunneling barriers are raised in the switch phase. Now the cells become polarized according to the polarization of their driver in the switch phase. Actual computation takes place in switch phase. Further any tunneling is suppressed because of the high barriers at the end of switch phase. Now the cell polarization is fixed. The tunneling barriers are set at high value in the hold phase. In this hold phase the cell is acting as an input to the next stage. The barriers are lowered in the release phase. In this phase the cells are allowed to relax to an unpolarized state. The cell barriers are remain lower in the relaxed phase to keep the cells in an unpolarized state. The subsystem will again reach to the first clock phase after this phase for repolarization.

## III. DEFECT CLASSIFICATION

A defect is an error introduced into a device during the manufacturing process. There are possibility of defect during synthesis and deposition phase. In synthesis phase single cell is fabricated while in deposition phase the cells are placed on substrate at desired location [6]. The defect classification is shown in Fig. 8. Defects can cause extra dots or/and electrons in synthesis phase. It also happens that dots or/and electrons in cell can be missed in synthesis phase. In fabrication a missing or additional dot is rarely possible. It is because of ease of purification of small size inorganic molecules. There are more chances of occurrence of defects in the deposition phase as compared to the synthesis phase. Any cell of device may get misaligned in misalignment defects that is the direction of this cell is not perfectly aligned in such a type of defects. The defective cell is misplaced from its original position in cell misalignment defect. A cell of device or circuit is missing in a missing cell deposition defect. The cell without electron or missing electron defect can be model as missing cell deposition defect. Additional cell may get deposited on the substrate in additional cell defect. This additional or extra cell can be deposited along the defect free device's adjacency boundary.

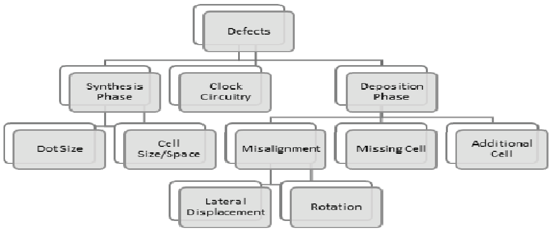


Fig. 8. Classification of defect characterization

Momenzadeh *et al.* have analyzed the deposition defects in molecular QCA devices and circuits [6]. Unlike metal-based QCA, defect can occur in molecular QCA implementation because of erroneous deposition of cells on a substrate. This deposition may cause the missing cell or an additional cell which are nearer to the layout or placed within the layout of a device. Extensive simulation and the impact of missing cell and additional cell defects were evaluated for in the devices like majority voter, inverter, different wire types like straight, L-shape, coplanar crossing and fanout.

Fault set is defined in [6] for all the QCA devices caused by defects. This paper uses these fault set to analyze the behavior of QCA circuit under the influence of defect fault. This analyses further helpful to find the test vector to detect the fault.

## IV. HDLQ

QCA Designer [9] is layout and device level simulator. QCA system's logic behavior can be described by HDLQ [18]. HDLQ libraries are designed using Verilog HDL for basic QCA building blocks majority voter, inverter, wire and crossover. Manually fault can be injected at logic level in the basic blocks of QCA circuits and systems. Designer can analyze the logic behavior of the QCA circuits and systems in the presence of faults using HDLQ. It further helpful to find the test vector. It also supports the bidirectionality and timing/clocking partitioning.

a) Example of HDLQ module for Majority voter

As shown in Fig. 2. majority voter is basic building block of QCA. MV has a function F (A, B, C) = AB+BC+AC. As per the defect analysis carried out in [6] MV contents the faults stuck_ at_ B (s_a_B) and F (A', B, C'). So two inputs are given in the HDLQ MV, one is *fault*0 and another one is *fault*1. If *fault*0 = 0, *fault*1 = 0 then it is considered as the fault-free configuration. If *fault0 = 0*, *fault*1 = 1 then it is considered as the stuck-at-B fault. If *fault*0 = 1 and *fault1 = 0* then it is considered as the fault F (A', B, C'). In the simulation of MV, user can set the value of fault accordingly for which he or she wants to verify the behavior of QCA designer under the influence of fault.

HDLQ supports the fault injection capabilities for all basic building blocks. Using this feature of HDLQ effect of faults can be analyzed in QCA circuits and systems.

## V. IMPLEMENTATION

The block level QCA combinational circuit, half adder is shown in Fig.9. It is not optimized since we wanted to carry out the fault analysis on various basic blocks. It consists of 4 majority voters, 2 inverters, 6 L-shaped wires, 2 fanouts and 1 crossover wire as discussed in section II.

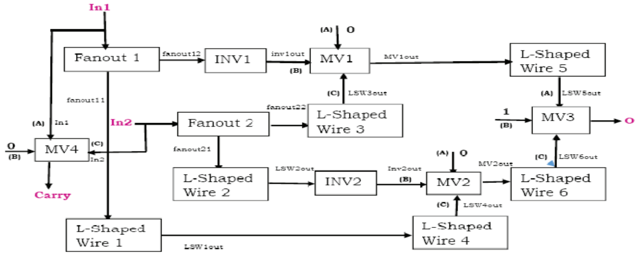


Fig. 9. Block diagram of half adder with QCA devices

In half adder sum is implemented by exoring two inputs. Three majority voters are required namely MV1, MV2 and MV3 to implement the exor gate. As shown in block diagram of half adder in Fig.9, one of the inputs of MV1 and MV2 is connected to the logic 0 to get AND gate and one input of MV3 is connected to logic 1 to get the OR gate. Inversion of inputs namely In1 and In2 is taken by Inverter INV1 and inverter INV2 respectively. The output of MV3 is "out" which is "sum" of the half adder.

Boolean equation is given below

MV1out = In1'In2

MV2out = In2'In1

MV3out = out = In1'In2+In2'In1

We know that the carry output of the half adder can simply achieve by performing AND operation on two inputs. So it is achieved by making MV4 as AND gate by keeping one of its inputs as logic 0. Inputs to MV4 are In1, In2 and logic 0. So output of MV4 is, MV4out=carry=In1In2. This shown in shown in Fig.9.

The block diagram of half adder shown in Fig.9. is implemented using QCADesigner to verify the functionality of it. The layout of half adder is shown in Fig.10. Four MVs are shown by black rectangles and two inverters are shown by red rectangles. It also uses different wire types L-Shaped wire, fanout and crossover. The various colors in layout represents the clock zone 0, clock zone 1, clock zone 2, and clock zone 3 as discussed in section II. This design can be optimized for no. of cells. Looking into the aim to carry out fault analysis, maximum no. of QCA devices are included in the design. So this design contents all basic QCA building blocks namely majority voter, inverter, fanout, L-shaped wire and crossover.

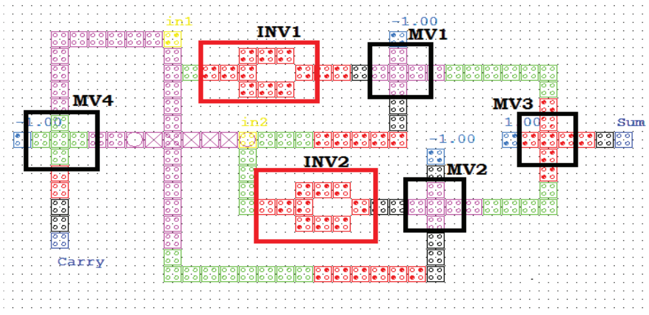


Fig. 10. Layout of half adder

The half adder is simulated using QCADesigner V 2.3.0. [19] Bistable approximation simulation engine is used with cell size 18nmX18nm, dot diameter 5nm, cell to cell distance of 2.5nm and radius of effect 50nm. Other simulation parameter are kept as default. The simulation waveform is shown in Fig.11. Output "sum" is appearing after delay of two clock cycles and output "carry" is appearing after delay of one clock cycle due the clock zone assignment. This is highlighted in Fig. 11 by red rectangle for "sum" output and green rectangle for "carry" output.

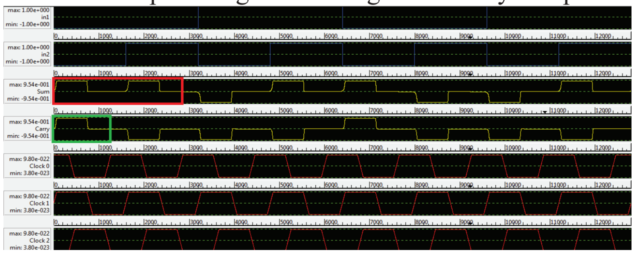


Fig. 11. Simulation results of half adder

Also single cell omission or missing cell deposition defect is observed for this circuit. Simulation is carried out by omitting one by one each cell of each device in the design. This results are matching with the analysis carried by M. Momenzadeh *et al.* [6]. This missing cell deposition defect was used to model the fault list. This fault list and its effect in our design at layout level is discussed in the next section.

## VI. FAULT ANALYSIS AT LAYOUT LEVEL

In this paper only missing cell defects are considered. A cell of device or circuit is missing in a missing cell deposition defect. No. of simulations are carried out one by one by omitting the single cell of devices in half adder.

This results are matching with the previous work carried out in [6]. Table I shows the fault list for all devices discussed in section II which occurs due to missing cell deposition defects. Defects in input, output and Crossover are not considered. Single missing cell in MV, inverter changes their functionality.

TABLE I Fault set caused by missing cell defect [6]

| Device | Fault Set |
|---|---|
| Majority Voter (MV) | *S_a_B*<br>*Maj (A',B,C')* |
| Inverter (INV) | *S_a_A* |
| L-Shaped Wire | *S_a_A'* |
| Fanout | *S_a_A'* |

a) Example of defect in majority voter (MV1)

Majority voter shown in Fig. 2. is considered. As per the missing cell defect analysis carried out in [6], If cell A or cell C is missing then output of MV1 will be B. stuck at B (S_a_B) fault is observed when cell A or C is missing. If middle cell, device cell D is missing, output of MV1 will give function M (A', B, C'). Missing cell defect causes the fault in QCA device.

At layout level to analyze the fault caused by the missing cell defect, we omitted one by one cell in each device of design. Simulations are done using QCADesigner V 2.3.0. [19] Bistable approximation simulation engine with cell size 18nmX18nm, dot diameter 5nm, cell to cell distance of 2.5nm and radius of effect 50nm. Other simulation parameter are kept as default.

These faults and its effect on primary output (PO) namely "sum" and "carry" for half adder is summarized in Table II.

For example, in MV1 two faults are possible namely S_a_B due to the missing of cell A or B and F (A', B, C') due to missing of cell D. So simulation is carried out by omitting cell A or cell B to analyze this S_a_B fault. If In1 and In2 are same i.e. In1=0 and In2=0 or In1=1 and In2=1 then the fault free primary output (PO) "sum" must be logic 0. Due to the S_a_B fault in MV1 output of MV1 becomes In1'. Now inputs to the MV3 are In1' and In2'In1. MV3 is acting as OR gate, so output of it which is PO "sum" is logic 1 instead of logic 0. This is shown in Fig. 12. The faulty and fault free value of "sum" is depicted in Table II for all the faults and devices.

Faults in MV4 affects only "carry" output, "sum" output remains unaffected whereas faults in other devices MV1, MV2, MV3, LShaped wire, fanout and inverter causes the change in output "sum" , "carry" output remains unaffected.

## VII. FAULT ANALYSIS AT LOGIC LEVEL

This work was intended to carry out the mentioned fault analysis at logic level using HDLQ. We have seen that HDLQ support the injection of faults caused by the missing cell defects. The module of half adder is created by instantiating the HDLQ device modules.

Proper care has taken to interconnect the modules to get the functionality as half adder. Xillinx 13.1 ISE tool is used for synthesis and simulation. The synthesis and simulation results are shown in Fig. 13. and Fig.14. respectively. Synthesis results shows the basic blocks or devices of QCA used in the design of half adder.

TABLE II Summary of fault effects on PO

| Device | Fault | PO Value<br>Faulty (fault free) |
|---|---|---|
| MV1 | *S_a_B* | *0 (1)* |
| | *Maj(A', B, C')* | *0 (1)* |
| MV2 | *S_a_B* | *0 (1)* |
| | *Maj(A', B, C')* | *0 (1)* |
| MV3 | *S_a_B* | *0 (1)* |
| | *Maj(A', B, C')* | *0 (1)* |
| MV4 | *S_a_B* | In1(In1In2)[for carry] |
| | *Maj(A', B, C')* | In1+In2(In1In2)[for carry] |
| INV1 | *S_a_A* | *In1'(In1)* |
| INV2 | *S_a_A* | *In2'(In2)* |
| Fanout1 | *S_a_A' for f1* | *In1'(In1)* |
| Fanout2 | *S_a_A' for f1* | *In2'(In2)* |
| L-Shaped Wire1 | *S_a_A'* | *In1(In1')* |
| L-Shaped Wire2 | *S_a_A'* | *In2'(In2)* |
| L-Shaped Wire3 | *S_a_A'* | *In2(In2')* |
| L-Shaped Wire4 | *S_a_A'* | *In1(In1')* |
| L-Shaped Wire5 | *S_a_A'* | *In1'(In1), 0(1)* |
| L-Shaped Wire6 | *S_a_A'* | *In2'(In2), 0(1)* |

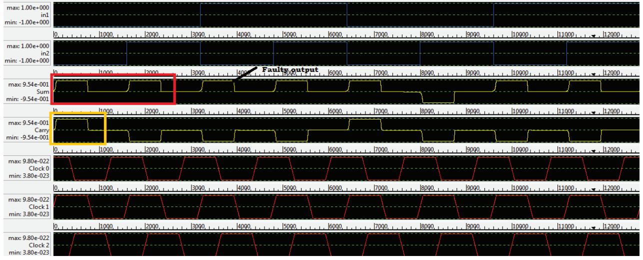


Fig. 12. Simulation for MV1 at s_a_B fault

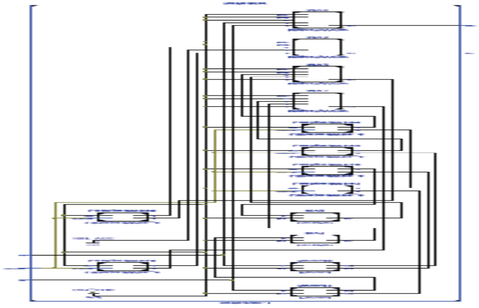
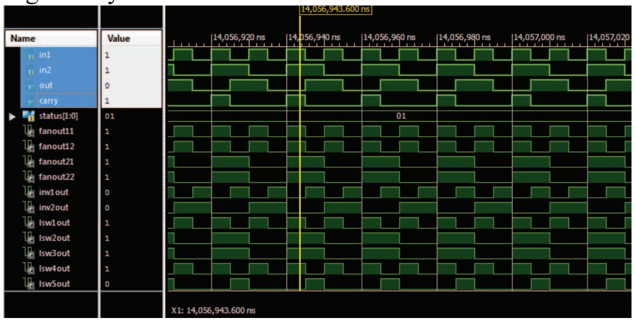


Fig. 13. Synthesis Result



| Name | Value |
|---|---|
| in1 | 1 |
| in2 | 1 |
| out | 0 |
| carry | 1 |
| status[1:0] | 01 |
| fanout11 | 1 |
| fanout12 | 1 |
| fanout21 | 1 |
| fanout22 | 1 |
| inv1out | 0 |
| inv2out | 0 |
| lsw1out | 1 |
| lsw2out | 1 |
| lsw3out | 1 |
| lsw4out | 1 |
| lsw5out | 0 |

Fig. 14. Simulation result

Auxiliary inputs are given in HDLQ to inject the faults in the design. These are summarized in Table III.

TABLE III Fault status in HDLQ

| **Device** | **Auxiliary Input** | **Fault** | **Condition** |
|---|---|---|---|
| Majority Voter MV | *fault0, fault1* | *Fault free* | *fault0=0, fault1=0* |
| | | *Maj(A',B,C')* | *fault0,=1,fault1=1* |
| | | *S_a_B* | *fault0,=0,fault1=1* |
| Inverter INV | *fault* | *Fault Free* | *fault=0* |
| | | *S_a_A* | *fault=1* |
| Fanout | *fault* | *Fault Free* | *fault=0* |
| | | *S_a_A'* | *fault=1* |
| L-Shaped Wire | *fault* | *Fault Free* | *fault=0* |
| | | *S_a_A'* | *fault=1* |

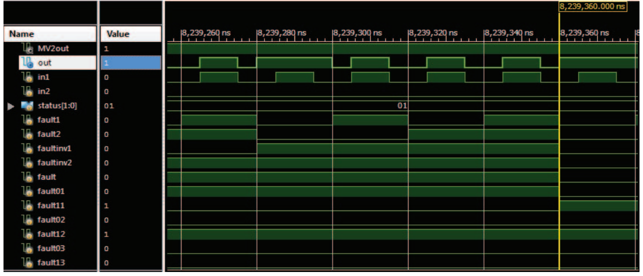
As per the Table III, depending upon the value of auxiliary input for respective QCA device the fault can be injected.

One by one the fault is injected and is simulated to observe its effects on PO. Test bench is generated for various fault values and input stimuli. The responses are matching with the the fault analysis carried out using QCADesigner as discussed in section V.

a) Example of fault injection in majority voter (MV1)

For MV1 fault0 and fault1 are set to logic 0 and logic 1 respectively for S_a_B fault. In this case for input In1 and In2 both equal to logic 0, output (PO) "out" which is sum must be 0. Since s_a_B fault is injected in MV1, its output will be B which is In1' means logic 1. Output of MV2, In2'In1 is logic 0. Output of MV3 which is oring of MV1 And MV2 outputs is now 1. So fault free and faulty values are logic 0 and logic 1 respectively.

The simulation result after this fault injection is shown in Fig. 15.



| Name | Value |
|---|---|
| MV2out | 1 |
| out | 1 |
| in1 | 0 |
| in2 | 0 |
| status[1:0] | 01 |
| fault1 | 0 |
| fault2 | 0 |
| faultinv1 | 0 |
| faultinv2 | 0 |
| fault | 0 |
| fault01 | 0 |
| fault11 | 1 |
| fault02 | 0 |
| fault12 | 1 |
| fault03 | 0 |
| fault13 | 0 |

Fig. 15. Simulation result for MV1 at S_at_B

No. of simulations are carried out to analyze the faults effects in various modules of QCA half adder using HDLQ. This results are similar to the fault analyze carried out at layout level. We have got the same results as layout level in the logic level. These are mentioned in the Table II.

## VIII. CONCLUSION

QCA Half adder is implemented for fault analysis. Faults caused by single missing cell deposition defects are analyzed at layout level using QCADesigner tool. These faults are also analyzed at logic level using the one of the features of HDLQ. Similar results are obtained using these two methods. These analysis will helpful to the generation of test vector for the testing of QCA circuits and system.

## REFERENCES

[1] K.Likharev, "Single electron devices and their applications", *Proc. of the IEEE*, 87(4), pp. 606–632, April 1999.

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

[3] K. Chen, K. Maezawa, and M. Yamamoto, "Inp-based high-performance monostable-bistable transition logic elements (MOBILE's) using integrated multiple input resonant-tunneling devices", *IEEE Electron Device Letters,* 17(3), pp. 127–129, March 1996.

[4] G. L. Snider, A. O. Orlov, I. Amlani, X. Zuo, G. H. Bernstein, C. S. Lent, J. L. Merz, "Quantum-dot cellular automata: Review and recent experiments", *Journal of Applied Physics*, vol 85, no.8, pp. 4283-4285, April 1999.

[5] Craig S. Lent and Gregory L. Snider, "The development of quantum-dot cellular automata", Field Coupled Nanocomputing, LNCS, Springer, pp.3-20, 2014.

[6] M. Momenzadeh, M. Ottavi and F. Lombardi, "Modeling qca defects at molecular-level in combinational circuits," *in Proc. of the 20th IEEE DFT*, pp. 208–216, 2005.

[7] M. Tahoori, M. Momenzadeh, J. Huang, and F. Lombardi,"Testing of quantum cellular automata," *IEEE Trans. on Nanotechnology*, vol. 3, no. 4, pp. 432–442, 2004.

[8] M. B. Tahoori, M. Momenzadeh, J. Huang, and F. Lombardi, "Defects and faults in quantum cellular automata at nano scale,*" in Proc. of the 22nd IEEE VTS,* pp. 291–296, 2004.

[9] 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. Nanotechnology,* vol. 3, no. 1, pp. 26–31, Mar. 2004.

[10] C.S. Lent and P.D, Tougaw, "A device architecture for computing with quantum dots", *in proc. of the IEEE*, vol.85, no.4, pp. 541-557, 1997.

[11] R.P. Cowburn and M.E. Welland, "Room temperature magnetic quantum cellular automata", *Science*, 2000.

[12] A. O. Orlov, Ravi K. Kummamuru, Rajagopal Ramasubramaniam, Geza Toth, Craig S. Lent, Gary H. Bernstein, and Gregory L. Snider, "Experimental demonstration of a latch in clocked quantum-dot cellular automata", *Applied Physics Letters,* 78(11):1625 –1627, March 2001.

[13] Craig S. Lent, "Bypassing the transistor paradigm", *Science*, 288(5471):1597–1599, 2000.

[14] Qi, H., Sharma, S., Li, Z., Snider, G. L., Orlov, A. O., Lent,C.S.,Andfehiner, T. P., " Molecular quantum cellular automata cells: Electric field driven switching of a silicon surface bound array of vertically oriented two-dot molecular QCA", *Journal of American. Chem. Society*, pp.15250–15259, 2003.

[15] P. Tougaw and C. Lent, "Logical devices implemented using quantum cellular automata," *Journal of Applied Physics,* vol. 75, no. 3, pp. 1818–1825, 1994.

[16] Geza Toth and Craig S. Lent, "Quasiadiabatic switching for metal-island quantum-dot cellular automata", *Applied Physics Letters*, 85(5):2977–2984, March 1999.

[17] K. Hennessy and C. Lent, "Clocking of molecular quantum-dot cellular automata," *Journal of Vaccum Science and Technology*, vol. 19, no. 5, pp. 1752–1755, 2001.

[18] M. Ottavi, L. Schiano, and F. Lombardi, "HDLQ: A HDL environment for QCA design," *ACM J. Emerging Technol. Comput. Syst.,* vol. 2, no. 4, pp. 243–261, Oct. 2006.

[19] http://www.mina.ubc.ca/qcadesigner