**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**

***IJCSMC, Vol. 3, Issue. 3, March 2014, pg.316 – 325***

**RESEARCH ARTICLE**

# Logical Fault Detection Based on Conservative QCA for Ultra Low Power Devices

**Yamuna S**
M.E Scholar
Sri Shakthi Institute of Engineering and Technology, Coimbatore
yamuna114@gmail

## Abstract

***As transistors decrease in size to accommodate in a single die, it increases chip computational capabilities. However, transistors cannot get much smaller than their current size. The quantum-dot cellular automata (QCA) approach represents one of the possible solutions in overcoming this physical limit. It can be implemented by using reversible gates which will not dissipate power during computation. In general, the testing of sequential circuits is difficult as compared to combinational circuits since it need to test along with the previous state. In this paper, the sequential circuit based on reversible logic can be tested using only two test vector, thereby eliminating the need for any type of scan-path access to internal memory cells. The two test vectors are all 1s and all 0s. The designs of two vectors testable latches, master-slave flip-flops and double edge triggered (DET) flip-flops are presented. Along with that a new design for multiplexer is implemented based on conservative QCA logic, that is not reversible in nature but has similar properties as the Fredkin gate of working as 2:1 multiplexer. The proposed MX-cqca gate surpasses the Fredkin gate in terms of complexity (the number of majority voters), speed, and area. The importance of the proposed work lies in the fact that it provides the design of reversible sequential circuits completely testable for any stuck-at fault by only two test vectors.***

***Keywords: quantum-dot cellular automata, Fredkin gate, reversible logic, double edge triggered flipflop, mx-cqca***

## 1. Introduction

Personal mobile communications and portable computing systems are the fastest growing sectors of the consumer electronics market. The electronic devices at the heart of such products need to dissipate low power, in order to conserve battery life and meet packaging reliability constraints. Although low power design methodologies will solve the problem of designing complex, low power digital VLSI circuits will still be subject to manufacturing defects. It was implicitly assumed that traditional Design for Testability methodologies is suitable for digital circuits designed using low power methods. A VLSI chip can dissipate up to three times higher power during testing when compared to normal (functional) operation due to higher switching activity. Power loss in the electronic system is a very crucial limiting factor that can be reduced or minimized with the help of using the reversible logics “a concept came from Thermodynamics” [17].

Reversible circuits are those circuits that do not lose information during computation and reversible computation in a system can be performed only when the system comprises of reversible gates. As a fundamental contribution, Landauer has shown that during irreversible computation 1 bit of information lost results in KTln2 Joules of energy dissipation[2]. Bennett in another seminal contribution proved that this KTln2 joule of energy dissipation will not occur if computation is performed in a reversible manner[4]. An N × N (N inputs and N outputs) reversible gate can be represented as

$$I_v = I_1, I_2, I_3, I_4, ........., I_N \quad (1.1)$$

$$O_v = O_1,O_2,O_3,O_4, .........,O_N \quad (1.2)$$

where Iv and Ov represent input and output vectors, respectively. Classical logic gates are irreversible since input vector states cannot be uniquely reconstructed from the output vector states[5]. Conservative logic is a logic family that exhibits the property that there are an equal number of 1s in the outputs as there are in the inputs. Conservative logic can be reversible in nature or may not be reversible in nature. Reversibility is the property of circuits in which there is one to-one mapping between the inputs and the output vectors, that is for each input vector there is a unique output vector and vice-versa[10]. Conservative logic is called reversible conservative logic when there is a one-to-one mapping between the inputs and the outputs vectors along with the property that there is equal number of 1s in the outputs as in the inputs. Conservative logic circuits are not reversible, if one-to-one mapping between the inputs and the outputs vectors is not preserved[7]. Several important metrics need to be considered in the design of reversible circuits the importance of which needs to be discussed. The constant input in the reversible quantum circuit is called the ancilla input qubit (ancilla input bit), while the garbage output refers to the output which exists in the circuit just to maintain one-to-one mapping but is not a primary or a useful output. Quantum computers of many qubits are extremely difficult to realize thus the number of qubits in the quantum circuits needs to be minimized[15]. There is a need of technology independent design and synthesis of reversible logic circuits that are applicable to quantum computing. The reversible circuit has other important parameters of quantum cost and delay which need to be optimized. The quantum cost of a design is the number of 1x1 and 2x2 reversible gates used in its design, thus can be considered equivalent to number of transistors needed in a conventional CMOS design[12]. The delay of a reversible circuit can be computed by calculating its logical depth when it is designed from smaller 1x1 and 2x2 reversible gates. An important metric for evaluating reversible circuits is the Garbage count. Garbage is defined as the number of outputs added to make an n-input k-output Boolean function ((n, k) function) reversible. Hence, one of the major issues in designing a reversible circuit is garbage minimization. Reversible logic is very essential for the construction of low power, low loss computational structures which are very essential for the construction of arithmetic circuits used in quantum computation, nano technology and other low power digital circuits. The important reversible gates[10] used for reversible logic synthesis are,fredkin gate, toffoli gate, feynmann gate, peres gate and so on. According to Moore's law, transistor size will double every 18 months but CMOS technology is reaching closer to the limits beyond which the feature size cannot be downscaled further without compromising the proper functioning of the device. CMOS devices suffer from thermal effects, as they have to discharge all the stored energy when flipping from 1 to 0[14].QCA are one of the emerging nanotechnologies which make it possible to achieve circuit densities and clock frequencies beyond the limits of existing CMOS technology. QCA has significant advantage in terms of power dissipation[2], as it does not have to dissipate all its signal energy during transition. Hence, QCA is considered as one of the promising technologies to achieve the thermodynamic limit of computation[6]. A QCA cell is a coupled dot system in which four dots are at the vertices of a square. Fredkin gate is a (3x3) conservative reversible gate, having the mapping (A, B, C) to (P=A, Q=A'B+AC, R=AB+A'C), where A, B, C are the inputs and P, Q, R are the outputs, respectively. It is called a 3x3 gate because it has three inputs and three outputs. Figure 1 shows the Fredkin gate. The cell has two extra electrons that occupy the diagonals within the cell due to electrostatic repulsion. The cell polarization P measures the charge distribution along diagonal axes and is given by equation 1 (here Pi denotes the electronic charge at dot i). When electrons are in dots 1 and 3, P = -1 (Logic '0') and when electrons in dots 2 and 4, P = +1 (Logic '1'). Figures 1.4(a) and 1.4(b) show the 4 quantum dots in a QCA cell[19], and the implementation of logic '0' and logic '1' in a QCA cell, respectively.

$$\mathbf{P = \frac{(P2 + P4) - (P1 + P3)}{P1+P2+ P3 + P4}} \quad (1.3)$$

In this paper, we propose the design of testable sequential circuits based on conservative logic gates. The proposed technique will take care of the fan-out (FO) at the output of the reversible latches and can also disrupt the feedback to make them suitable for testing by only two test vectors, all 0s and all 1s. In other words, circuits will have feedback while executing in the normal mode[11]. However, in order to detect faults in the test mode[18], our proposed technique will disrupt feedback to make conservative reversible latches testable as combinational circuits. The proposed technique is extended toward the design of two vectors testable master-slave flipflops and double edge triggered (DET) flip-flops. Thus, our work is significant because we are providing the design of reversible sequential circuits completely testable for any unidirectional stuck-at faults by only two test vectors. The reversible design of the DET flip-flop is proposed for the first time in the literature. Further, while designing a QCA sequential circuit, the designer may sometimes prefer to sacrifice the reversibility to save the number of QCA cells while keeping the test strategy to be the same that is the design can still be tested by two test vectors. Thus, we also propose a new conservative logic gate called multiplexer conservative QCA gate (MX-cqca) that is not reversible in nature but has similar properties as the Fredkin gate of working as 2:1 multiplexer. The proposed MX-cqca gate surpasses the Fredkin gate in terms of complexity [the number of majority voters (MV)], speed, and area.

## 2. Conservative Reversible Fredkin Gate

The Fredkin gate is a popularly used reversible conservative logic gate, first proposed by Fredkin and Toffoli in [8]. The Fredkin gate shown in Fig. 1 can be described as a mapping (A, B, C) to (P = A, Q = A'B + AC, R = AB + A'C), where A, B, C are the inputs and P, Q, R are the outputs, respectively.

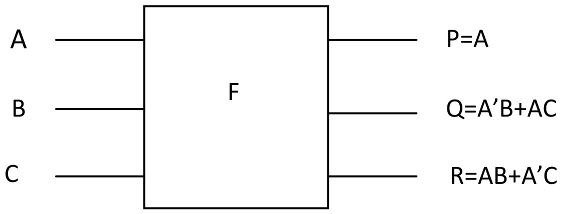
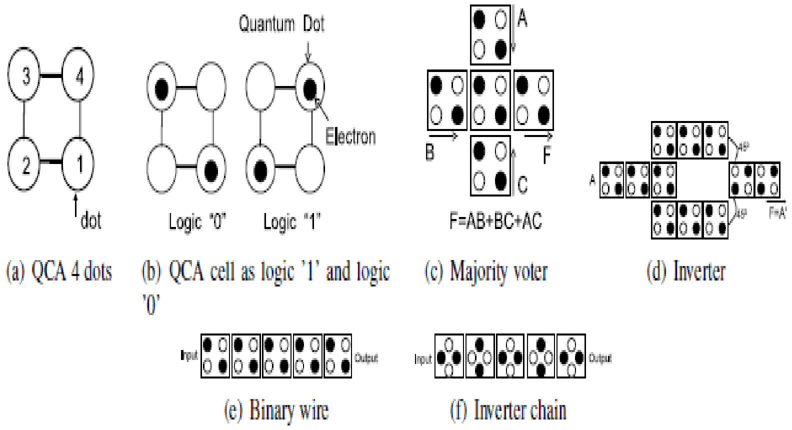


Fig.1 Fredkin gate

The truth table for the Fredkin gate is illustrated in [3], which demonstrates that Fredkin gate is reversible and conservative in nature, that is, it has unique input and output mapping and also has the same number of 1s in the outputs.

### 2.1 Basics Of QCA Computing

The basic QCA device is the majority voter or majority gate, and has the output function as F=AB+BC+AC, where F is the majority of the inputs A, B and C. The majority voter can be made to work as an AND gate or as an OR gate, by setting one of the inputs as '0' and '1', respectively [16]. (For example, if C=0 we will get F=A.B. Similarly if C=1, we will get F=A+B). Another important gate in QCA is the inverter, which is formed when a QCA cell, say cell-1 is placed 45 degrees to another QCA cell, for example cell-0, cell-1 gets the inverse value of cell-0. There can be many ways of designing the QCA inverter, one of which is shown in Figure 2. In QCA computing, signal transfer is made through wires that are of two types, Binary wire and inverter chain.



(a) QCA 4 dots (b) QCA cell as logic '1' and logic '0' (c) Majority voter (d) Inverter

(e) Binary wire (f) Inverter chain

Fig.2 Basic QCA cell

### 2.2 Design Of Testable Reversible Latches

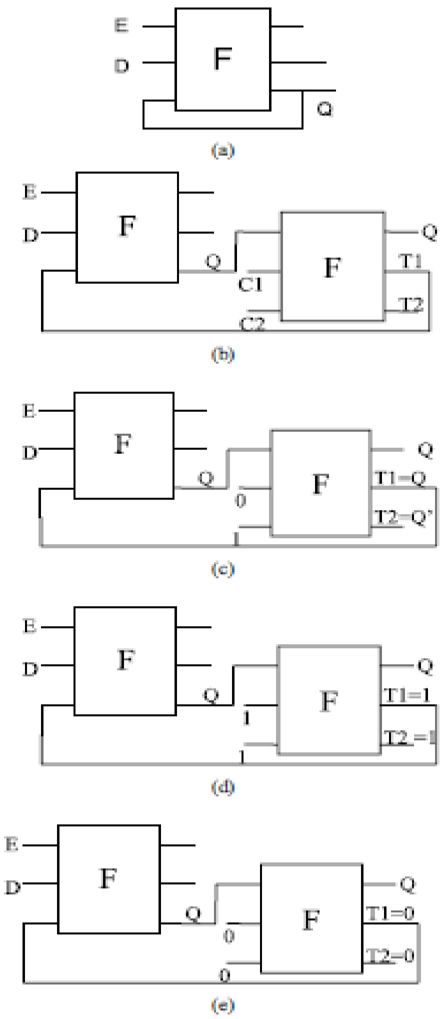
The characteristic equation of the D latch can be written as Q+ = D · E + .E · Q. In the proposed work, enable (E) refers to the clock and is used interchangeably in place of clock. When the enable signal (clock) is 1, the value of the input D is reflected at the output that is Q+ = D. While, when E = 0 the latch maintains its previous state, that is Q+ = Q. Fig. 3(a) shows the realization of the reversible D latch using the Fredkin gate. But FO is not allowed in conservative reversible logic. Moreover, the design cannot be tested by two input vectors all 0s and all 1s because of feedback, as the output Q would latch 1 when the inputs are toggled from all 1s to all 0s and could be misinterpreted as stuck-at-1 fault. In this paper, we propose to cascade another Fredkin gate to output Q as shown in Fig. 3(b).

The design has two control signals, C1 and C2. The design can work in two modes: 1) normal mode and 2) test mode.

1) Normal Mode: The normal mode is shown in Fig. 3(c) in which we will have C1C2 = 01 and we will have the design working as a D latch without any fan-out problem.

2) Test Mode (Disrupt the Feedback): In test mode, when C1C2 = 00 as shown in Fig. 3(d) it will make the design testable with all 0s input vectors as output T1 will become 0 resulting in making it testable with all 0s input vectors. Thus, any stuck-at-1 fault can be detected. When C1C2 = 11 as shown in Fig. 3(e), will become 1 and the design will become testable with all 1s input vectors for any stuck-at-0 fault. It can seen from above that C1 and C2 will disrupt the feedback in test mode, and in normal mode will take care of the fanout. Thus, our proposed design works as a reversible D latch and can be tested with only two test vectors, all 0s and all 1s, for any stuck-at fault by utilizing the inherent property of conservative reversible logic.

The reversible Fredkin gate has two of its outputs working as 2:1 Muxes, thus the characteristic equation of the D latch can be mapped to the Fredkin gate (F).



(a)

(b)

(c)

(d)

(e)

Fig. 3. Design of testable reversible D latch using conservative Fredkin gate. (a) Fredkin gate based D latch. (b) Fredkin gate based D latch with control signals C1 and C2. (c) Fredkin gate based D Latch in normal mode: C1 = 0 and C2 = 1. (d) Fredkin gate based D latch in test mode for stuck-at-0 fault: C1 = 1 and C2 = 1. (e) Fredkin gate based D latch in test mode for stuck-at-1 fault: C1 = 0 and C2 = 0.

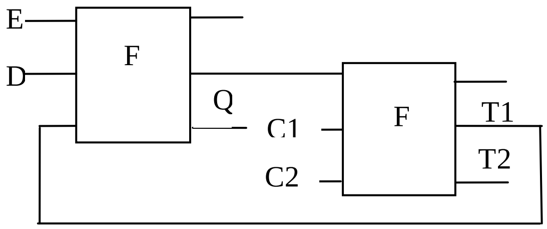


Fig.4 Fredkin gate based negative enabled D latch

A negative enable reversible D latch will pass the input D to the output Q when E = 0; otherwise maintains the same state. The characteristic equation of the negative enable D latch is Q+ = D·  .E +E· Q. This characteristic equation of the negative enable reversible D latch can be mapped on the second output of the Fredkin gate as shown in Fig. 4. The second Fredkin gate in the design take cares of the FO. The second Fredkin gate in the design also helps in making the design testable by two test vectors, all 0s and all 1s, by breaking the feedback based on control signals C1 and C2 as illustrated above for positive enable reversible D latch. The negative enable D latch is helpful in the design of testable reversible master-slave flipflops.

This is because as it can work as a slave latch in the testable reversible master-slave flip-flops in which no clock inversion is required. The details of which are discussed in the section describing reversible master-slave flip-flops.

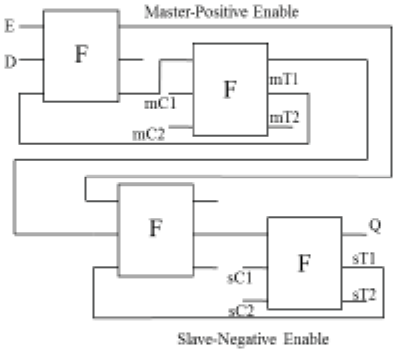


Fig.5 Design Of Testable Master Slave Flipflop

### 2.3 Design Of Testable Master Slave Flipflop

In this paper, we have proposed the design of testable flip-flops using the master slave strategy that can be tested for any stuck-at faults using only two test vectors, all 0s and all 1s. Fig. 5 shows the design of the master-slave D flip-flop in which we have used positive enable Fredkin gate-based testable D latch shown in Fig. 3(b) as the master latch, while the slave latch is designed from the negative enable Fredkin gate-based testable D latch shown earlier in Fig. 4. The testable reversible D flip-flops has four control signals mC1,mC2, sC1, and sC2. mC1 and mC2 control the modes for the master latch, while sC1 and sC2 control the modes for the slave latch. In the normal mode, when the design is working as a master-slave flip-flop the values of the controls signals will be mC1 = 0 and mC2 = 1,  sC1 = 0 and sC2 = 1 ( as similar to values of the control signals C1 and C2 earlier described for the testable D latches).

In the test mode.

1) To make the design testable with all 0s input vectors for any stuck-at-1 fault, the values of the controls signals will be mC1 = 0 and mC2 = 0, sC1 = 0 and sC2 = 0. This will make the outputs mT1 and sT1 as 0, which results in breaking the feedback and the design becomes testable with all 0s input vectors for any stuck-at-1 fault.

2) To make the design testable with all 1s input vectors for any stuck-at-0 fault, the values of the control signals will be mC1 = 1 mC2 = 1, sC1 = 1, and sC2 = 1. This will result in outputs mT1 and sT 1 having a value of 1, breaking the feedback and resulting in the design testable with all 1s input vectors for any stuck-at-0 fault. The other type of master-slave flip-flops, such as the testable master-slave T flip-flop, testable master-slave JK flip-flop, and testable master-slave SR flip-flop can be designed similarly in which master is designed using the positive enable corresponding latch, while the slave is designed using the negative enable Fredkin gate-based D latch. For example, in the design of master-slave T flip-flop, the master is designed using the positive enable T latch, while the slave is designed with the negative enable D latch.

### 2.4 Design Of Testable Reversible DET Flipflop

The double edge triggered flip-flop is a computing circuit that sample and store the input data at both the edges, that is at both the rising and the falling edge of the clock. The master-slave strategy is the most popular way of designing the flip flop. In the proposed work E (Enable) refers to the clock and are used interchangeably in place of clock. In the negative edge triggered master-slave flip-flop when E=1 (the clock is high), the master latch passes the input data while the slave latch maintains the previous state. When E=0 (the clock is low), the master latch is in the storage state while the slave latch passes the output of the master latch to its output. Thus, the flip-flop does not sample the data at both the clock levels and waits for the next rising edge of the clock to latch the data at the master latch[13].

In order to overcome the above problem, researchers have introduced the concept of double edge triggered (DET) flip-flops which sample the data at both the edges. Thus DET flip-flops can receive and sample two data values in a clock period thus frequency of the clock can be reduced to half of the master-slave flip flop while maintaining the same data rate. The half frequency operations make the DET flip flops very much beneficial for low power computing as frequency is proportional to power consumption in a circuit. The DET flip-flop is designed by connecting the two latches, viz., the positive enable and the negative enable in parallel rather than in series. The 2:1 MUX at the output transfer the output from one of these latches which is in the storage state (is holding its previous state).

In the proposed design of testable reversible DET flip-flop, the positive enable testable reversible D latch and the negative enable testable reversible D latch are arranged in parallel. The Fredkin gates labeled as 1 and 2 forms the positive enable testable D latch while the Fredkin gates labeled as 3 and 4 forms the negative enable testable D latch. In reversible logic fanout is not allowed so the Fredkin gate labeled as 1 is used to copy the input signal D. The Fredkin gate labeled as 6 works as the 2:1 MUX and transfer the output from one of these testable latches (negative enable D latch or the positive enable D latch) that is in the storage state (is holding its previous state) to the output Q.

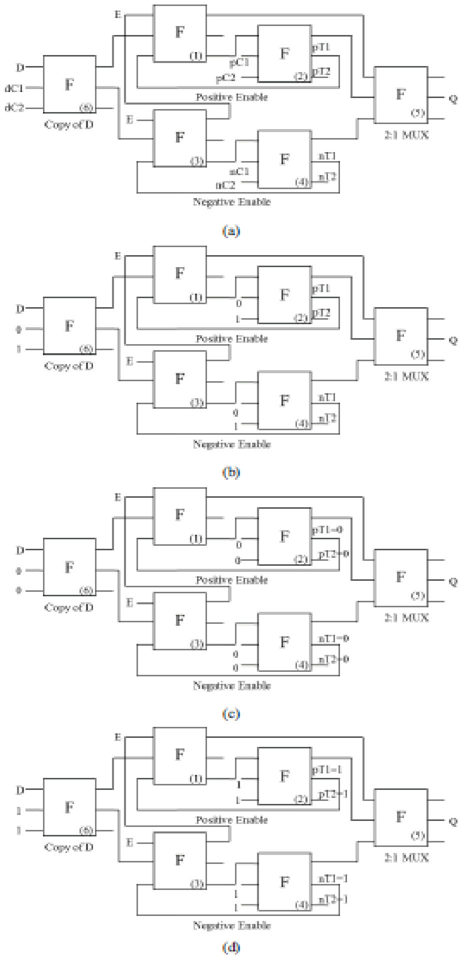


Fig.6 Fredkin gate-based DET flip-flop. (a) Fredkin gate based DET flipflop. (b) Normal mode. (c) Test mode for stuck-at-1 fault. (d) Test mode for stuck-at-0 fault.

In the proposed design of testable reversible DET flip-flop pC1 and pC2 are the controls signals of the testable positive enable D latch, while nC1 and nC2 are the control signals of the testable negative enable D latch. Depending on the values of the pC1, pC2, nC1 and nC2 the testable DET flip-flops works either in normal mode or in the testing mode.

- **Normal Mode:** The normal mode of the DET flip-flop is illustrated in Figure 6(a) in which the pC1=0, pC2=1, nC1=0 and nC2=1. The pC1=0, pC2=1 helps in copying the output of the positive enable D latch thus avoiding the fanout while the nC1=0 and nC2=1 helps in copying the output of the negative enable D latch thus avoiding the fanout.
- **Test Mode:** There will be two test modes :

All 1s Test Vector: This mode is illustrated in Figure 6(c) in which control signals will have value as pC1=1, pC2=1, nC1=1 and nC2=1. The pC1=1 and pC2=1 help in breaking the feedback of the positive enable D latch, while the nC1=1 and nC2=1 help in breaking the feedback of the negative enable D latch. This makes the design testable by all 1s test vector for any stuck-at-0 fault.

All 0s Test Vector: This mode is illustrated in Figure 6(b) in which the control signals will have value as pC1=0, pC2=0, nC1=0 and nC2=0. The pC1=0 and pC2=0 help in breaking the feedback of the positive enable D latch while the nC1=0 and nC2=0 help in breaking the feedback of the negative enable D latch. This makes the design testable by all 0s test vector for any stuck-at-1 fault.

### 2.5 Proposed Multiplexer Conservative QCA

For many of the designs, the designer could potentially be interested in using the testing advantages of conservative logic but saving the number of QCA cells. Thus, in this work we propose a new conservative logic gate that is conservative in nature but is not reversible.

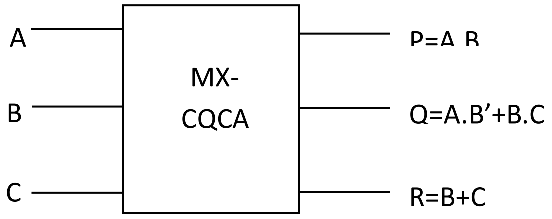


Fig.7 proposed MX-CQCA gate

The proposed conservative logic gate is called Multiplexer Conservative QCA gate (MX-CQCA) and has 3 inputs and 3 outputs. MX-CQCA has one of its outputs working as a multiplexer that will help in mapping the sequential circuits based on it, while the other two outputs work as AND and OR gates, respectively.

The mapping of the inputs to outputs of the MX-CQCA is: P= AB; Q=AB'+BC; R=B+C, where A, B, C are the inputs and P, Q, R are the outputs, respectively. Figure 3.4 shows the block diagram of the MX-CQCA gate. Table 3.1 shows the truth table of the MX-CQCA gate.

Table 1 Truth Table of MX-CQCA gate

| A | B | C | P | Q | R |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

The table verifies the gate's conservative logic nature, i.e., that the numbers of 1s in the inputs is equal to the number of 1s in the outputs.

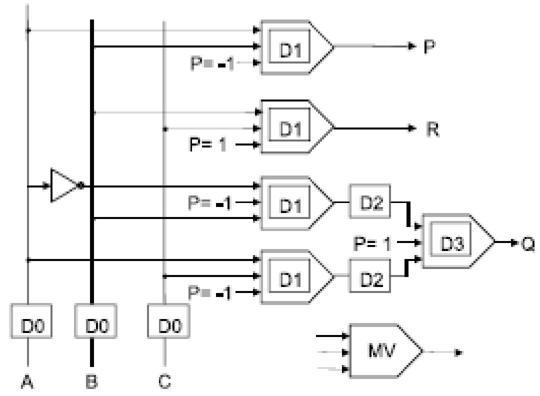


Fig.8 QCA model of MX-CQCA

From the QCA design Figure 8, we can see that the proposed MX-CQCA gate requires four clocking zones and five majority gates for its QCA implementation. The basic element of QCA circuits is majority gate and invertors. Hence, efficiently constructing QCA circuits using majority gates has attracted a lot of attention[9].QCA computing is one the promising technology to implement reversible logic gates. The QCA design of Fredkin gate has four-phase clocking scheme, in which the clocking zone is denoted by the number next to D (D0 means clock 0 zone, D1 means clock 1 zone and so on). The Fredkin gate has two level majority voter (MV) implementation, and it requires 6 MVs and 4 clocking zones for implementation. The number of clocking zones in a QCA circuit represents the delay of the circuit[1] (delay between the inputs and the outputs). Higher the number of clocking zones, lower the operating speed of the circuit.

### 2.6 Design Methodology For Nonreversible Testable Design Based On Mx-Cqca Gate

The proposed conservative logic gate MX-CQCA is useful to design any majority logic and multiplexer logic-based testable nonreversible circuits. In the existing literature, 13 standard functions are proposed to represent all three-variable Boolean functions[8]. These thirteen functions are widely used in QCA and majority logic-based synthesis. In order to design any complex function based on MX-CQCA, the proposed design methodology can be summarized in the following three steps.

1) Step 1: The input function is decomposed into the Boolean network in which every node has almost three variables. This step is similar to the design methodology proposed.

2) Step 2: The three variable function generated at every

node in Step 1 is mapped to its MX-CQCA-based implementation. The mapping is based on the library of 13 standard functions implemented using the MX-CQCA.

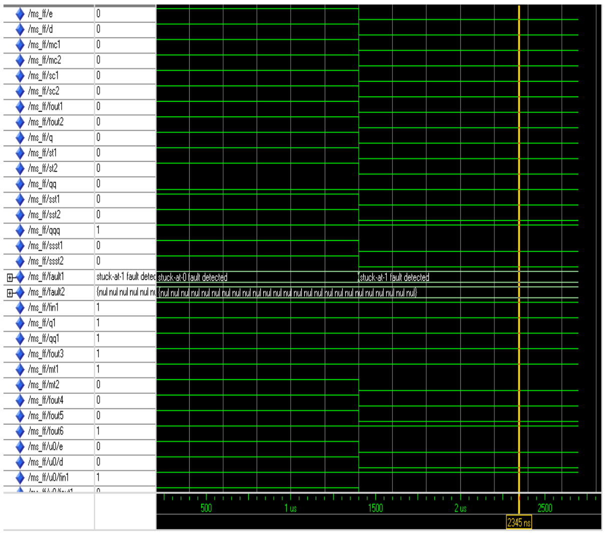
3) Step 3: The nodes which have fan-out of more than one are identified, and MX-CQCA gates are used to form the copy of the signals, which have fan-out of more than one.

## 3. Results and Discussions

Figure 9 shows the simulation of the master-slave D flip-flop in which we have used positive enable Fredkin gate based testable D latch shown in Figure 3.2 as the master latch, while the slave latch is designed from the negative enable Fredkin gate based testable D latch. The testable reversible D flip-flops has four control signals mC1, mC2, sC1 and sC2. mC1 and mC2 control the modes for the master latch, while sC1 and sC2 control the modes for the slave latch. When mC1=0, mC2=0 at master side and sC1=0 and sC2=0 at slave side is forced then stuck at 1 fault is detected at their test output and similarly for stuck at 0 fault is detected.

## 4. Conclusion

This paper proposed reversible sequential circuits based on conservative logic that is testable for any unidirectional stuck-at faults using only two test vectors, all 0s and all 1s. The proposed sequential circuits based on conservative logic gates outperform the sequential circuit implemented in classical gates in terms of testability. The sequential circuits implemented using conventional classic gates do not provide inherited support for testability. Hence, a conventional sequential circuit needs modification in the original circuitry to provide the testing capability. Also as the complexity of a sequential circuit increases the number of test vector required to test the sequential circuit also increases. For example, to test a complex sequential circuit thousand of test vectors are required to test all stuck-at-faults, while if the same sequential circuit is build using proposed reversible sequential building blocks it can be tested by only two test vectors, all 0s and all 1s. Thus, the main advantage of the proposed conservative reversible sequential circuits compared to the conventional sequential circuit is the need of only two test vectors to test any sequential circuit irrespective of its complexity. The reduction in number of test vectors minimizes the overhead of test time for a reversible sequential circuit.



| | |
|---|---|
| /ms_ff/e | 0 |
| /ms_ff/d | 0 |
| /ms_ff/mc1 | 0 |
| /ms_ff/mc2 | 0 |
| /ms_ff/sc1 | 0 |
| /ms_ff/sc2 | 0 |
| /ms_ff/fout1 | 0 |
| /ms_ff/fout2 | 0 |
| /ms_ff/q | 0 |
| /ms_ff/st1 | 0 |
| /ms_ff/st2 | 0 |
| /ms_ff/qq | 0 |
| /ms_ff/sst1 | 0 |
| /ms_ff/sst2 | 0 |
| /ms_ff/qqq | 1 |
| /ms_ff/ssst1 | 0 |
| /ms_ff/ssst2 | 0 |
| /ms_ff/fault1 | stuck-at-1 fault detec |
| /ms_ff/fault2 | {nul nul nul nul nul nu |
| /ms_ff/fin1 | 1 |
| /ms_ff/q1 | 1 |
| /ms_ff/qq1 | 1 |
| /ms_ff/fout3 | 1 |
| /ms_ff/mt1 | 1 |
| /ms_ff/mt2 | 0 |
| /ms_ff/fout4 | 0 |
| /ms_ff/fout5 | 0 |
| /ms_ff/fout6 | 1 |
| /ms_ff/u0/e | 0 |
| /ms_ff/u0/d | 0 |
| /ms_ff/u0/fin1 | 1 |

stuck-at-0 fault detected
stuck-at-1 fault detected
{nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul}
500
1 us
1500
2 us
2500
2345 ns

Fig. 9 simulation of master slave flipflop

The proposed work has the limitation that it cannot detect multiple stuck-at-faults as well as multiple missing/additional cell defects. In conclusion, this paper advances the state-of-the-art by minimizing the number of test vectors needed to detect stuck-at-faults as well as single missing/additional cell defects.

## References

[1] Dalui M, Sen B, and Sikdar B. K, (2010) 'Fault tolerant QCA logic design with coupled majority-minority gate,' Int. J. Comput. Appl., vol. 1, no. 29, pp. 81–87.

[2] A. Fijany and B. Toomarian, 2001 "New design for quantum dots cellular automata to obtain fault tolerant logic gates," *J. Nanoparticle Res.*, vol. 3, no. 1, pp. 27–37.

[3] E. Fredkin and T. Toffoli, 1982 "Conservative logic," *Int. J. Theor. Phys.*, vol. 21, nos. 3–4, pp. 219–253.

[4] Himanshu Thapliyal,Nagarajan Ranganathan and Saurabh Kotiyal,(2013) 'Design of testable reversible sequential circuits', ieee transactions on very large scale integration (vlsi) systems, vol. 21, no. 7.

[5] P. Kartschoke, 1992 "Implementation issues in conservative logic networks," M.S. thesis, Dept. Electr. Eng., Univ. Virginia, Charlottesville.

[6] Kong K,Shang Y, and Lu R,(2010) 'An optimized majority logic synthesis methodology for quantum-dot cellular automata,' IEEE Trans. Nanotechnol., vol. 9, no. 2, pp. 170–183.

[7] Mahammad S and Veezhinathan K,(2010) 'Constructing online testable circuits using reversible logic', IEEE Trans. Instrum. Meas., vol. 59, no. 1, pp. 101–109.

[8] S. F. Murphy, M. Ottavi, M. Frank, and E. DeBenedictis, 2006 "On the design of reversible QDCA systems," Sandia National Laboratories, Albuquerque, NM, Tech. Rep. SAND2006-5990.

[9] Ravi1 T and Kannan V,'A Novel Design of D-Flip Flop Using New RR Fault Tolerant Reversible Logic Gate'.

[10] J. Ren and V. K. Semenov, "Progress with physically and logically reversible superconducting digital circuits," *IEEE Trans. Appl. Superconduct.*, vol. 21, no. 3, pp. 780–786, Jun. 2011.

[11] G. Swaminathan, J. Aylor, and B. Johnson, "Concurrent testing of VLSI circuits using conservative logic," in *Proc. Int. Conf. Comput. Design*,Cambridge, MA, Sep. 1990, pp. 60–65.

[12] G. Swaminathan, 1989 "Concurrent error detection techniques using parity," M.S. thesis, Dept. Electr. Eng., Univ. Virginia, Charlottesville.

[13] Thapliyal H and Ranganathan N (2010), 'Design of reversible sequential circuits optimizing quantum cost, delay and garbage outputs',ACM J. Emerg. Technol. Comput. Syst., vol. 6, no. 4, pp. 14:1–14:35.

[14] Thapliyal H, Srinivas M.V, and Zwolinski M, (2005), 'A beginning in the reversible logic synthesis of sequential circuits,' in Proc. Int. Conf. Military Aerosp. Program. Logic Devices, Washington, DC pp. 1–5.

[15] Thapliyal H, (2011) ‘Design, synthesis and test of reversible logic circuits for emerging nanotechnologies,’ Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. South Florida, Tampa.

[16] H. Thapliyal and N. Ranganathan, 2010 “Reversible logic-based concurrently testable latches for molecular QCA,” *IEEE Trans. Nanotechnol.*, vol. 9,no. 1, pp. 62–69,.

[17] P. Tougaw and C. Lent, 1996 “Dynamic behavior of quantum cellular automata,” *J. Appl. Phys.*, vol. 80, no. 8, pp. 4722–4736.

[18] Zhang R, Walus K, Wang W, and Jullien G.A, 2004 ‘A method of majority logic reduction for quantum cellular automata,’ IEEE Trans. Nanotechnol., vol. 3, no. 4, pp. 443–450.