Synthesis Of Reversible Universal Logic Around QCA With Online Testability

Bibhash Sen*, Divyam Saran*, Mousumi Saha†
*Department of Computer Science and Engineering
†Department of Computer Application
National Institute Of Technology, Durgapur
West Bengal-713209, India
bibhash.sen@gmail.com, divyamsaran@gmail.com,
msaha.nitd@gmail.com

Biplab K Sikdar
Department of Computer Science and Technology
Bengal Engineering and Science University, Shibpur,
West Bengal 711103, India
biplab@cs.becs.ac.in

***Abstract*—Quantum-dot Cellular Automata (QCA) can be a viable technology for CMPs (chip multi-processors) with thousands of processors. The QCA based reversible logic promises energy efficient design of the digital circuits. However, the requirement of excessive logic gates as well as its high defect rate limit the performance of a QCA based design. This work proposes a new approach to synthesize the reversible universal QCA logic gate (RUG) with the target to reduce the garbage outputs as well as the number of logic gates to realise a design simultaneously ensuring the defect tolerance. A concurrent error detection methodology is introduced to support the online testing of a circuit designed around the RUG. The experimental designs establish that the RUG can ensure an energy saving cost effective realization of testable QCA circuits.**

Keywords: *Universal QCA, Reversible gate, Online Testing, Symmetric Functions and Energy Efficient Design.*

I. Introduction

The current digital design techniques target energy efficient realization of complex logic circuits with high device density. The QCA (*Quantum-dot Cellular Automata*) is considered to be a promising technology to meet such a design target (device density of $10^{12} device/cm^2$). It achieves switching speed of $10ps$ and a power dissipation of the order of $100W/cm^2$ [1].

The fundamental unit of QCA based design is the 3-input $majority\ gate$ (majority voter). Since the majority gate is not functionally complete, the majority gate with inverter, called MI, are used to realize the QCA designs. However, the number of logic gates required to realize a QCA design as well as the resulted garbage outputs limit the performance of such a design in nano-scale.

The universal gate structures such as AOI (and-or-inverter) [2] have been proposed to achieve the cost effective digital design. However, these gates realize the irreversible logic and, therefore, can't be the right choice for energy efficient design.

Landauer [3] proved that for irreversible logic computations, each bit of information loss generates $k_B Tln2$ joules of heat energy as the energy required for a binary transition E_{bit} is given by the Shannon-von Neumann- Landauer (SNL) expression [3]

$$E_{bit} \geq E_{SNL} = k_B Tln2 = 0.017eV, \tag{1}$$

where k_B is the Boltzmann constant, T=300K. This is the minimum energy to process a bit. However, the reversible computation allows computing beyond the SNL limit. Bennett [4] showed that a zero power dissipation in logic circuits is possible only if a circuit is composed of reversible logic gates.

The reversible logic conserves energy using a charge recovery process in CMOS. On the other hand, the QCA circuit is a clocked information preserving system [5]. The energy dissipation of a QCA circuit can be significantly lower than $k_B Tln2$. This energy saving feature can be the additional point in favour of the introduction of QCA technology in CMPs. However, the hurdle in introducing QCA technology is its proneness to high defect rate [2].

The above scenario motivates us to investigate a new gate structure around QCA that is reversible and also realizes the universal logic function. This satisfies the requirement of optimum logic gates as well as minimum number of garbage outputs in an energy efficient design simultaneously ensuring the defect tolerance. We propose a reversible universal logic gate (RUG) that reduces the number of logic gates and garbage outputs in a digital design around it.

It is established that the RUG leads to an area saving implementation of complex logic simultaneously ensuring energy loss close to zero. The effectiveness of RUG is evaluated through the realization of standard and symmetric functions. A concurrent error detection scheme for RUG based design is also proposed that avoids the use of comparators in conventional reversible circuit testing. The next section provides the basics of QCA and the fundamentals of reversible gates.

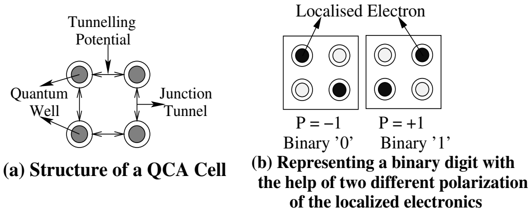
II. The QCA Basics

A quantum dot is a region where an electron is quantum-mechanically confined as shown in Fig.1(a). A quantum cell consists of four such quantum dots at each corner of

978-0-7695-4570-7/11 $26.00 © 2011 IEEE
DOI 10.1109/ISED.2011.53

IEEE computer society

a square and contains two free electrons [1]. The electrons can quantum-mechanically tunnel among the dots and settle either in polarization P = -1 (logic 0) or in P = +1 (logic 1) as noted in Fig.1(b). The timing/synchronization in QCA is accomplished by the cascaded clocking of four distinct and periodic phases [1].



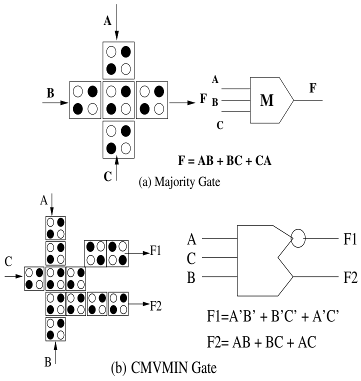
(a) Structure of a QCA Cell

(b) Representing a binary digit with the help of two different polarization of the localized electronics

Figure 1. A QCA cell

A. QCA logic gate

The basic structure realized with QCA is the 3-input *majority gate*, i.e. MV(A,B,C) = maj(A,B,C) = AB+BC+CA (Fig.2(a)). To realize universal function, a CMVMIN gate structure that simultaneously realises 3-input minority logic (MIN) and majority voter (MV) in its 2 outputs F1 and F2 (Fig.2(b)) is also proposed in [6].



(a) Majority Gate

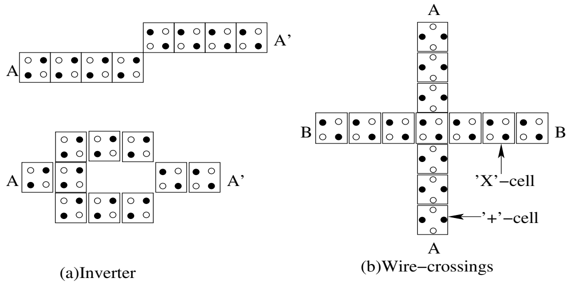
(b) CMVMIN Gate

Figure 2. A Majority gate and a CMVMIN gate

Inverter realized in two different orientations is shown in Fig.3(a). In QCA based logic implementation, two kinds of QCA wires are possible. Fig.3(b) describes the only allowable wire crossing in QCA based design. It requires two different orientations, a 90 degree (×-cell) and a 45 degree (+ -cell) structure. However, manufacturing nano-scale cells with two different orientations is a challenging task in QCA.

B. Reversible logic gate

A logic gate is reversible if the mapping of its inputs to outputs is bijective -that is, every distinct input (I_v) yields a distinct output (O_v) and the number of inputs is equal to the number of outputs. An NxN reversible gate is having



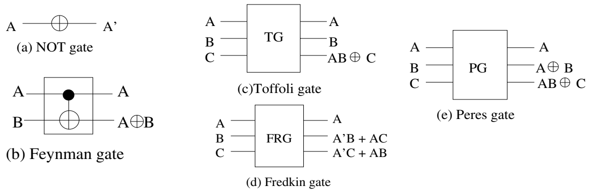
(a)Inverter

(b)Wire−crossings

Figure 3. Wire-crossing and inverter

I_v = (I_1, I_2, I_3,................................, I_N),
O_v= (O_1, O_2, O_3,.........................., O_N).

A detailed list of reversible gates is reported in [7]. The most important among those are the NOT, CNOT or Feynman gate, Toffoli gate, Fredkin gate, Peres gate etc [9] (shown in Fig.4).



(a) NOT gate

(b) Feynman gate

(c)Toffoli gate

(d) Fredkin gate

(e) Peres gate

Figure 4. Reversible gates

The synthesis of reversible logic differs significantly from traditional irreversible logic synthesis approaches as fan-outs and feedback paths are not permitted in reversible logic [9].

III. The Reversible Universal Gate (RUG)

Characterization of thirteen 3 variable standard functions [2] and symmetric functions [11] raises the following design issues:

- What are the best possible logic function(s) that can implement boolean logic functions efficiently?
- How can the number of logic gates be minimised in a design to make it more cost-effective?
- How to ensure energy efficient computing avoiding the information loss in a circuit realizing complex logic?

In this section, we explore the properties of reversible universal logic gate (RUG) and the designs around it. To design the logic gate, we consider a universal function. An n-input universal logic gate is considered for realization of any n-variable function and referred to as the n-ULG. In the current work, the following universal function, described in [12],

$$\boldsymbol{f}\textbf{(A,B,C)= AB + A'C'} \qquad (1)$$

is considered for the proposed RUG structure. Since the use of this universal function helps easy realization of

XNOR/XOR, the RUG can enable low cost realization of many other complex boolean functions.

The block diagram of RUG is shown in Fig.5. It also realizes, in its output o1, a majority function. The output o3 (XOR) of RUG enables energy saving realization of logic circuit with the RUG. The QCA cell layouts of RUG is shown in Fig.6 .

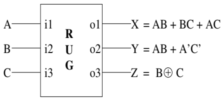


Figure 5. Block diagram of reversible universal gate (RUG)

The RUG(A,B,C) is, therefore, a 3x3 gate with the output mapping functions o1 = MF(A,B,C) = AB+BC+CA, o2 = UF(A,B,C) = AB+A'C' and o3 = B⊕C. The truth table of RUG is shown in Table I. From the table, it can be verified that the input pattern corresponding to a particular output pattern can be uniquely determined -that is, RUG is reversible.

Table I
TRUTH TABLE OF RUG GATE

A	B	C		X	Y	Z
0	0	0		0	1	0
0	0	1		0	0	1
0	1	0		0	1	1
0	1	1		1	0	0
1	0	0		0	0	0
1	0	1		1	0	1
1	1	0		1	1	1
1	1	1		1	1	0

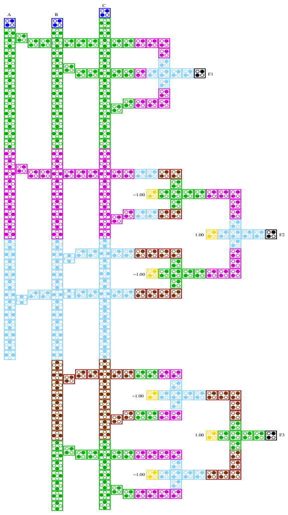


Figure 6. QCA cell layouts of RUG

The effectiveness of RUG in digital design is evaluated in realizing the 13 standard functions, symmetric functions and the reversible benchmark circuits.

A. Realizing standard functions with RUG

The thirteen standard functions (shown in Table II) represent all 256 (2^{2^3}) 3-variable boolean functions since any 3-variable boolean function can be converted to one of the thirteen standard functions [2].

A comparative study on the performance of conventional reversible gates and the RUG in implementing standard functions, is also reported in Table II. It can be observed from Table II that RUG provides best design option.

B. Realizing symmetric functions

A switching function f(x_1,x_2,....x_n) is called symmetric (or totally symmetric) with respect to the variables x_1,x_2,....x_n, if it is invariant under any permutation of its variables. The symmetric functions are receiving considerable attention from the researchers working in the field of VLSI design [11]. A number of synthesis techniques for boolean symmetric functions are reported [11] to suit different applications. A comparative performance analysis is given in implementing 2 and 3 variable symmetric functions in Fig.7. It is evident from Fig.7 that the implementation cost of symmetric functions is drastically reduced in a RUG based realization than that of other realizations.

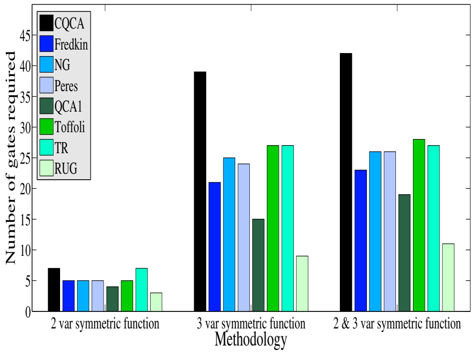


Figure 7. Performance of reversible gates in realizing symmetric functions

A generalised synthesis schemes for symmetric functions is explored in [13]. For n variables, there are a total of $2^{n+1}-2$ symmetric functions. For synthesis of all symmetric functions a generalised σ-functions [13] $\sigma_n^k(x_1, x_2, ..., x_n)$ are defined as $\oplus_{\{i_1,i_2,...,i_K\}} x_{i_1}, x_{i_2}, ..., x_{i_k}$ for k=0, 1,...., n. For example, Over the set of 3 variables $\{x_1, x_2, x_3\}$, the σ-functions are

- σ_3^0=1,
- σ_3^1= $x_1 \oplus x_2 \oplus x_3$,
- σ_3^2=$x_1x_2 \oplus x_1x_3 \oplus x_2x_3$,
- σ_3^3=$x_1x_2x_3$

Table II
REALIZATION OF STANDARD FUNCTIONS WITH REVERSIBLE GATE

13 standard functions	FREDKIN	PERES	QCA1 [14]	TOFFOLI	TR [10]	RUG
01. F=AB'C	2	2	2	3	2	3
02. F=AB	1	1	1	1	3	1
03. F=A'BC+A'B'C'	3	4	3	4	3	2
04. F=A'BC+AB'C'	4	3	6	#	4	3
4.8 05. F=A'B+BC'	5	5	2	4	4	3
06. F=AB'+A'BC	3	8	6	#	5	3
07. F=A'BC+ABC'+A'B'C'	5	5	6	#	7	3
08. F=A	1	1	1	1	1	1
09. F=AB+BC+CA	8	11	1	#	6	1
10. F=A'B+B'C	1	4	3	4	3	3
11. F=A'B+BC+AB'C'	4	2	6	2	2	3
12. F=AB+A'B'	2	3	3	2	2	1
13. F=ABC'+A'B'C'+AB'C+A'BC	11	4	7	2	3	3
Average Number of gates	3.8462	4.0769	3.6154	2.5556	3.4615	2.3077

'#': Difficult to implement

Every symmetric function can be written as a linear combination (with respect to the XOR operation) of at most (n+1) different σ-functions [13].

If we represent the number of RUGs needed for i^{th} σ-function by $T_{\sigma_n^i}$, then the following proposition points to the sufficient condition for the effectiveness of RUG in implementing σ-functions, that leads to the realization of different symmetric functions of n-variable.

Proposition: *Each σ-functions of n variable can be realized with at most:*

- $T_{\sigma_n^i} = 1$ *when* $i = 0$*, since* $\sigma_n^0 = 1$ *irrespective of n.*
- $T_{\sigma_n^i} = n - 1$ *when* $i = 1$ *or n.*
- $T_{\sigma_n^i} = T_{\sigma_{n-1}^i} + T_{\sigma_{n-1}^{i-1}} + 2$ *when* $i = 2$ *to* $n - 1$ *and n>3 (where* $T_{\sigma_3^2} = 1$*).*

Example: A 3 variable symmetric function $x_1x_2x_3 \oplus x_1'x_1'x_3'$ is equivalent to the linear combination of $\sigma_3^0 \oplus \sigma_3^1 \oplus \sigma_3^2$. In three variable (n=3), σ_3^0 function requires one RUG, σ_3^1 function requires two RUGs, σ_3^2 requires one RUG, and two RUGs for σ_3^3 function. So, at most six RUGs are needed to implement that three variable σ functions. That is, to implement $x_1x_2x_3 \oplus x_1'x_1'x_3'$, we need at most six RUGs *Fig.8*.

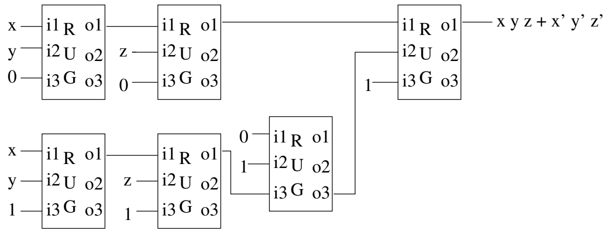


Figure 8. Implementation of $xyz \oplus x'y'z'$ with RUGs

C. Realization of benchmarks with RUG

Table III reports the implementation cost of reversible benchmark circuits [15] with RUG. It is evident from Table III that the reversible benchmark circuits can be better realized with the RUGs.

Table III
PERFORMANCE OF RUG IN REALIZING REVERSIBLE BENCHMARK

Benchmark	♯input	♯output	♯ Gates	♯ Garbage
RD32	3	2	3	4
Ham3	3	3	8	12
4mod5	4	1	11	19
Xor5	5	1	4	8
2of5	5	1	20	30
RD53	5	3	28	23

IV. CONCURRENT ERROR DETECTION

The last two rows in Table I represent that for any value of C and A=B=1, the output X is always 1. This is described in the first row of Table IV. All such combination of A, B, C that generate the constant values at X are summarised in Table IV.

Table IV
TRUTH TABLE SHOWING CONSTANT VALUES AT X

A	B	C		X
1	1	ϕ		1
ϕ	1	1		1
1	ϕ	1		1
0	0	ϕ		0
ϕ	0	0		0
0	ϕ	0		0

ϕ =Don't care value

Now, if input A = D, is faulty, then the minimal input condition to propagate this error (D) to X can be done by setting the other inputs to 0 or 1. For example, the value of A (=1) can be propagated to X (=1) for B=1 and C=ϕ (first row of Table IV) or B=ϕ and C=1 (third row of Table IV). That is,⟨ D10|D⟩ and ⟨D01|D⟩ are two schemae (⟨**input-values**|**output**⟩) for propagating the value of A to output X. These two are noted in the first two rows of Table V. Table V is the fault/error propagation table of X.

Table V
TRUTH TABLE FOR ERROR PROPAGATION AT X

A	B	C		X
D	1	0		D
D	0	1		D
0	D	1		D
1	D	0		D
0	1	D		D
1	0	D		D

D = Faulty value

The fault propagation in other outputs Y and Z are shown in Table VI) and Table VII respectively. These tables are utilised to generate test patterns for a RUG based design.

Table VI
TRUTH TABLE FOR ERROR PROPAGATION AT Y

A	B	C		Y
D	1	1		D
D	0	0		D'
1	D	ϕ		D'
0	ϕ	D		D'

D = Faulty value, D'= complement of D

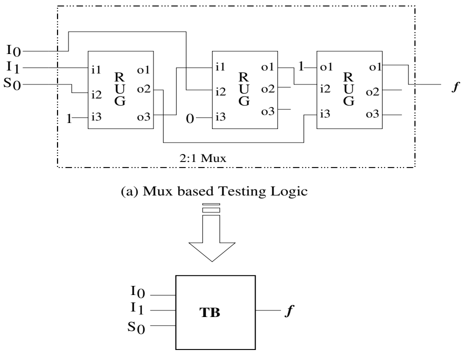
Table VII
TRUTH TABLE FOR ERROR PROPAGATION AT Z

A	B	C		Z
ϕ	D	0		D
ϕ	D	1		D
ϕ	0	D		D
ϕ	1	D		D'

D = Faulty value, D'= complement of D

For testing the RUG based circuit, a 2:1 Mux is constructed using three RUG as shown in Fig.9(a).

A. Test pattern generation for RUG based Mux



(a) Mux based Testing Logic

(b) Mux based Testing Block

Figure 9. Mux based test logic with RUG

Let us consider the fault s-a-0 at the input(I_0) of RUG2 of 9(a). It can be propagated to the o1 output of RUG2 either ⟨0D1|D⟩ or ⟨1D0|D⟩. As the i3 input of RUG2 is 0 which is mandatory for the construction of 2:1 Mux, the correct test pattern will be ⟨1D0|D⟩. It fixes S_0 = 1. Now this defines RUG1 schema as ⟨ ϕ 01|1⟩ or ⟨ ϕ 10|1⟩ (first two rows of Table VII). However i3 input of RUG1 is 1. Therefore, the correct test pattern at RUG1 is ⟨ ϕ 01 | 1 ⟩. That is, test pattern of $\langle i_0 i_1 s_0 \rangle$ is ⟨ D ϕ 0 ⟩. All possible value of $I_0 I_1 S_0$ to generate the test vector for RUG based 2:1 Mux are shown in Table VIII.

Table VIII
TEST VECTOR OF 2:1 MUX

I_0	I_1	S_0		f
D	ϕ	0		D
ϕ	D	1		D
0	1	D		D
1	0	D		D'

D = Faulty value, D'= complement of D, ϕ =Don't care

B. Online testing of RUG based circuit

Fig.10 shows the online test structure for a RUG. It exploits the following rule to ensure runtime fault detection of RUG.

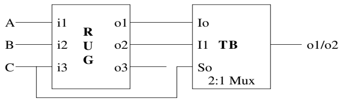


Figure 10. Online testing with 2:1 Mux based TB

- If $S_0 = 0$, TB (Fig.9(b)) propagates I_0 (here, o1) to f, i.e. detects fault at I_0 (here, o1).
- If $S_0 = 1$, TB propagates I_1 (here, o2) to f, i.e. detects fault at I_1 (here, o2).

Let assume a_i represents a 3-bit pattern with decimal value i, for example, a_0 = 000, a_5 = 101. In Fig.10, o1 and o2 are fed to I_0 and I_1 respectively in TB. Now a_0 is fed to RUG and the output is a_2 (Table I). In this output a_2, fault-free value of o1 is 0. For the fault s-a-1 at the first output o1 of RUG, faulty value is 1. So, following D-algorithm [16], we can state that the vector $a_0 = 000$ is the required test vector for s-a-1 fault at o1. Similarly, for s-a-0 at o1, the test vector ($\langle a_6 \rangle$) can detect both the faults through I_0. It effectively produces complete test vector set $\langle a_0, a_4, a_5, a_7 \rangle$ for the stuck-at faults of the structure.

To detect all the stuck at faults at three outputs of RUG, simultaneously, two TB are cascaded as shown in Fig.11. Since, fanout is not allowed in reversible logic, it is avoided by using the reversible Feynman Gate (FG) as shown in Fig.12(b). However, feynman gate is not required in QCA (Fig.13).

The output pattern of ***f1*** and ***f2*** (Fig.13) can detect the faults of the RUG based design that generate output X, Y, and Z concurrently. If the ⟨f1 f2 ⟩ on application of $\langle a_0, a_4, a_5, a_7 \rangle$ is ⟨ 01,11,10 ⟩, the circuit is fault-free. In

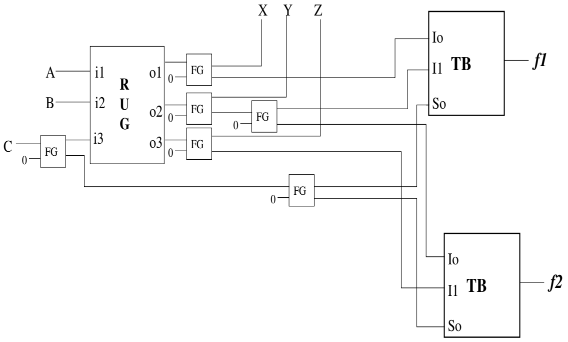
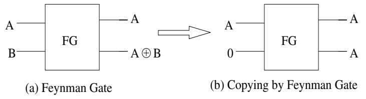


Figure 11. Online testing



(a) Feynman Gate (b) Copying by Feynman Gate

Figure 12. Copying by Faynman gate

other cases, the ⟨f1 f2 ⟩ pattern indicates fault. Thus the reversible gate along with this TB implemented in QCA technology can be used for concurrent detection of faults in a QCA circuit. It avoids the use of comparator in reversible circuit testing.

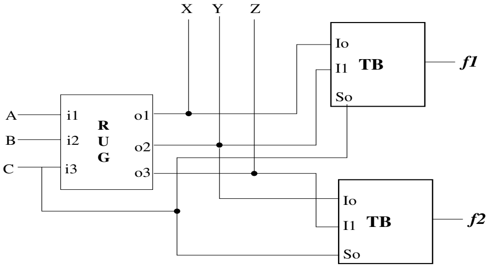


Figure 13. Online testing with respect to QCA

V. Conclusions

This work introduces a reversible QCA logic gate structure referred to as the RUG which saisfies the role of universal reversible gate in an energy efficient logic design. It is established that the RUG based QCA circuits are cost-effective in terms of number of logic gates, garbage outputs as compared to the designs based on existing reversible gates. This cost effective feature of RUG, with high device density, will play important role in CMPs with multi-processors. Realization of symmetric functions and benchmarks using RUG has been reported. It establishes the fact that any reversible gate with universal, majority and XOR function can enable low power QCA designs. A new methodology of concurrent error detection in reversible logic circuit is also proposed. It denies the use of comparators for online testing of reversible circuits.

References

[1] C.S Lent and P.D.Taugaw,*'A device architecture for computing with Quantum dots'*,in Proceedings of IEEE, vol.85, no.4, pp.541-557, April 1997.

[2] Momenzadeh, M, Jing Huang, Tahoori, M.B., Lombardi, F.,*'Characterization, test, and logic synthesis of and-or-inverter (AOI) gate design for QCA implementation'*, in IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, pp-1881 - 1893 No. 12, December,2005.

[3] R. Landauer, *'Irreversibility and Heat Generation in the Computational Process'* , IBM Journal of Research and Development, 5, pp. 183-191, 1961.

[4] C.H. Bennett, *'Logical Reversibility of Computation'* , IBM J.Research and Development, pp. 525-532, November 1973.

[5] J. Timler and C. Lent, *'Maxwells demon and quantum dot cellular automata'*, Journal of Applied Physics, vol.94, no. 2, pp. 1050, 2003.

[6] S. Ditti, P. K. Bhattacharya, P. Mitra and B. K. Sikdar,*'Logic Realization with Coupled QCA Majority-Minority Gate'*, VDAT, 2008.

[7] P. Kerntopf, *'Synthesis of multipurpose reversible logic gates'*, in Proc. Euromicro Symp. DSD, 2002, pp 259-267.

[8] T. Toffoli, *'Reversible computing'*,MIT, Cambridge, MA, 1980. Tech. Rep. MIT/LCS/TM-151.

[9] E. Fredkin and T. Toffoli, *'Conservative logic'*, Int. J. Theor. Phys.,vol. 21, no. 3/4, pp. 219253, 1982.

[10] Himanshu Thapliyal and Nagarajan Ranganathan,*'Design of Efficient Reversible Binary Subtractors Based on A New Reversible Gate'*, Proceedings of IEEE Computer Society Annual Symposium on VLSI, 2009.

[11] H. Rahman. D.K. Das and B.B. Bhattacharya,*'Mapping symmetric functions to hierarchical modules for path-delay fault testability'*, Proceedings, Asian Test Symposium (ATS), IEEE CS Press, USA, pp. 284-289, Nov. 2003.

[12] Yinshui Xia and Keming Qiu *'Design and application of Universal logic gate based on Quantum-Dot Cellular Automata'*, in Proceedings of 11th IEEE conference on Communication Technology, 2008.

[13] D. Maslov,*'Efficient reversible and quantum implementations of symmetric Boolean functions '*, IEEE Proc,-Circuits Devices Syst., vol.153, No.5, October 2006.

[14] X. Ma, J Huang, C Metra and F Lombardi,*'Testing Reversible 1D Arrays for molecular QCA'*, in Proceedings of 21st IEEE internation symposium on Defect and Fault Tolerence in VLSI(DFT'06) 2006.

[15] http://www.iqc.cs/ dmaslov/research.html

[16] M. Abramovici, M. A. Breuer, and A Friedman, *Digital System Testing and Testable Design.* New York:IEEE Press,1995.