This document is an author-formatted work. The definitive version for citation appears as:

A. Roohi, R. Zand, S. Angizi and R. F. Demara, "A Parity-Preserving Reversible QCA Gate with Self-Checking Cascadable Resiliency," in IEEE Transactions on Emerging Topics in Computing, vol. PP, no. 99, pp. 1-1. doi: 10.1109/TETC.2016.2593634

http://ieeexplore.ieee.org/abstract/document/7518653/

A Parity-Preserving Reversible QCA Gate with Self-Checking Cascadable Resiliency

Arman Roohi, *Student Member, IEEE,* Ramtin Zand, *Student Member, IEEE,* Shaahin Angizi, *Student Member, IEEE,* and Ronald F. DeMara, *Senior Member, IEEE*

Abstract

A novel Parity-Preserving Reversible Gate (PPRG) is developed using Quantum-dot Cellular Automata (QCA) technology. PPRG enables rich fault-tolerance features, as well as reversibility attributes sought for energy-neutral computation. Performance of the PPRG design is validated through implementing thirteen standard combinational Boolean functions of three variables, which demonstrate from 10.7% to 41.9% improvement over the previous gate counts obtained with other reversible and/or preserving gate designs. Switching and leakage energy dissipation as low as 0.141 eV and 0.294 eV, for 1.5 E_k energy level are achieved using PPRG, respectively. The utility of PPRG is leveraged to design a 1-bit full adder with 171 cells occupying only 0.19 μm^2 area. Finally, fault detection and isolation properties are formalized into a concise procedure. PPRG-based circuits capable of self-configuring active recovery for selected three-variable standard functions are realized using a memoryless method irrespective of garbage outputs.

Index Terms

Quantum-dot cellular automata, reversible logic, parity-preserving, self-checking fault detection methodology, full adder.

I. Introduction

QUANTUM-DOT CELLULAR AUTOMATA (QCA) has been widely investigated for potential improvements over transistor-based technologies like Complementary Metal-Oxide-Semiconductor (CMOS). QCA offers high device density, low power consumption and high switching speed features [1, 2]. Due to these attractive attributes, quantum gates and circuits have been targeted for their enabling roles toward computational reversibility. Reversibility in a quantum computing structure is defined as a one-to-one mapping of inputs to outputs, which leads to low power consumption and error detection ability [3]. It is worth mentioning that the loss of one bit of information expends greater than or equal to *kTln2* joules of energy, where *k* is Boltzman's constant and *T* the absolute temperature that the computation is operated [4]. As an example of the considerable previous research undertaken to construct reversible components based on nanotechnologies including QCA, use of the *Bennett* clock mechanism achieves less than *kTln2* switching energy dissipation [5-9].

In this paper, we first present a novel parity-preserving reversible gate which is then used to design a fault tolerant reversible full adder. In our proposed *Parity-Preserving Reversible Gate (PPRG)*, besides the one-to-one mapping feature of reversibility, the number of 1's in the input is equal to that of the output. Therefore, any fault which causes mismatch in parity can be detected.

This paper is organized as follows. Section II presents review of the QCA and the reversible gates. Section III develops the functionality of PPRG and then realizes a fault tolerant full adder which is based on it. Section IV describes a fault detection and isolation methodology for PPRG-based designs. Finally, conclusions are provided in Section V.

II. Background

A. QCA Review

QCA cells have been realized through several physical implementation methods such as semiconductor [1-2,10], metal island (metal-dot) [5,10-11], molecular [2,5,43], and nanomagnetic dot [5,10,12]. The first two are the most promising implementations, due to their high operation speed and integration density. Specifically, semiconductor QCA cells can be fabricated using existing industrial frameworks. The main difference between the metal dot and semiconductor QCA implementations is that metal dots are capacitively-coupled, while the quantum dots are coulombically-coupled in the semiconductor realization. The major challenge of these technologies is their low operating temperature, while molecular and nanomagnetic implementations can operate in room temperature. However, molecular integration is relatively complicated, and also it is not compatible with existing technologies for read and write operations. In contrast, nanomagnetic QCAs have been fabricated and shown proper functionality, although their performance are significantly lower than semiconductor and metal QCAs.

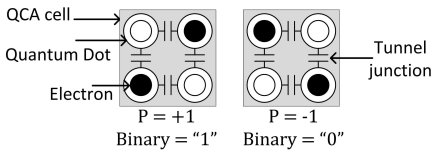

Fig. 1: QCA cell with two possible polarizations.

The basic element in QCA is a square archetype QCA cell, which is comprised of four quantum dots and two identical charges. Each cell has two certain polarization states which is obtained owing to the Coulomb interaction between the two free electrons in order to reach the maximum distance [1]. Unlike CMOS, QCA encodes binary data using inter-cell Coulomb repulsion between electrons that forms two stable diagonal positions , i.e. P=-1 and P=+1, which are assigned to denote a logic "0" and logic "1", respectively, as shown in Fig. 1 [2].

Additionally, due to the Coulombic effects, binary data passes through a row of cascaded QCA cells which constitutes a QCA wire. In order for a QCA signal to propagate properly, synchronization and energy restoration are essential which are obtained by applying a specific four-phased clocking scheme. There are two types of QCA clocking schemes commonly utilized, the Landauer [4] (high speed) and the Bennett [3] (low power consumption). Although the Launder clocking suffers from abrupt switching, it can provide the system with two benefits, which are fine-grained pipelining and high throughput. These have been emphasized in designing QCA circuits. The Landauer clocking scheme, which is generally used in QCA circuitry includes four distinct zones, *Switch*, *Hold*, *Release*, and *Relax*, where each zone is shifted by 90° compared to the previous zone, as shown in Fig. 2(a).

Considering clocking strategies in conjunction with circuit design strategies can lead to the concept of physical reversibility. The connection between logical reversibility and this concept is comprehensively discussed in [41]. If a logically-reversible QCA system can be implemented by employing physically reversible devices, associated power dissipation caused by information erasure can become zero. Although, the Launder clocking strategy used in this paper lends itself to circuits with high throughput and fine-grained pipelining, which are essential for the proposed highly cascadable gate, it still requires the circuit to be reversible at very fine levels. Adopting a more realistic and physical clocking strategy, as referred to uni-directional Bennett scheme [41], provides more flexibility in designing the reversible gates, since only the gate needs to be reversible as opposed to each segment of the gate. However, this strategy has lower throughput as compared to Launder clocking strategy and incurs a time overhead to the system. In order to combine the low power advantage of reversible computing with the high throughput advantage of pipelining, a combination of Bennett clocking and memory storage was proposed as an innovative approach which has been presented in [42].

The required adiabatic clocking mechanism is produced by CMOS circuits including a metal conductor, and wires such as nanowire [5] or CNTs [13]. The wires are placed under the QCA structure in the y-plane to generate the electric field whereas the conductor above the QCA cells spreads in the xy-plane to guide the produced electric field in z-direction, as shown in Fig. 2(b). The produced clock is then applied to the cells to control the height of the interdot barrier, either raising or reducing it, between dots within a QCA cell. A QCA cell is polarized due to the effects of adjacent cells during the Switch phase, and then in the Hold phase, interdot barriers are raised to the maximum height to maintain its polarization. In the Release phase, the cell moves to unpolarized state by losing its polarity, due to the reduction in the interdot barrier. Finally, in the Relax phase, the cell is unpolarized with negligible barrier. Our work is based on the aforementioned clocking mechanism [14]. Due to the QCA synchronization issues, specific implementation rules such as number of cells in a clock zone or synchronize flows arriving to a gate, must be considered during the design of a QCA-based circuit. Moreover, each QCA clock zone consist of at least two cells to preserve its influence on the subsequent clock zone.

Almost all QCA circuits are built based on the fundamental elements, the inverter gate and the Majority Gate (MG) [1], [2]. Figure 3 depicts widely used QCA designs for the inverter gate and the MGs. For MG designs, existence of electrostatic repulsion among the electrons of the input cells forces the central cell(s), i.e. device cell, to the stable polarization and propagate this state to the output cell as an output which is polarized to the majority polarization of inputs. The 3-input MG, as shown in Fig. 3(b) functions $AB + BC + AC$ (A, B and C are the three inputs). MG could be also programmed such that it functions as an AND or an OR logic function, if one of the three inputs is set to -1 or +1, respectively.

Recently, several designs for 5-input MG have been proposed [16-19] as shown in Fig. 3(c). The 5-input MG outputs "1" if and only if at least three input cells are set to "1". It functions according to the following equation:

$$\begin{aligned} M(A,B,C,D,E) = {} & ABC + ABD + ABE + ACD + \\ & ACE + ADE + BCD + BCE + BDE + CDE \end{aligned} \tag{1}$$

In similar manner to the 3-input MG, by fixing the polarization of two of the five inputs in the 5-input MG as logic "0" or logic "1", a 3-input AND gate or a 3-input OR gate can be achieved, respectively.

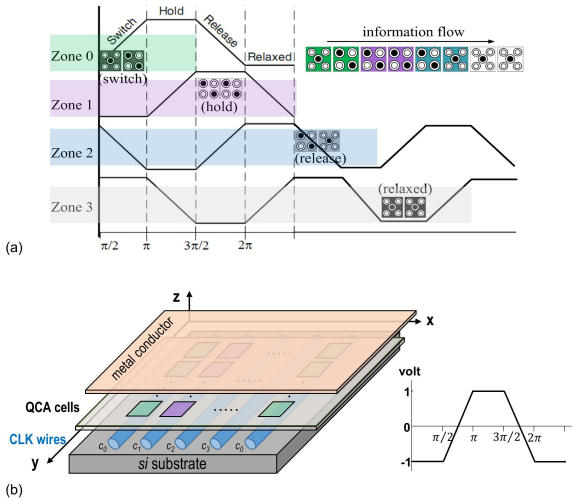

Fig. 2: QCA clocking scheme (a) four-phased clock of the QCA zones, and (b) cross-sectional view.

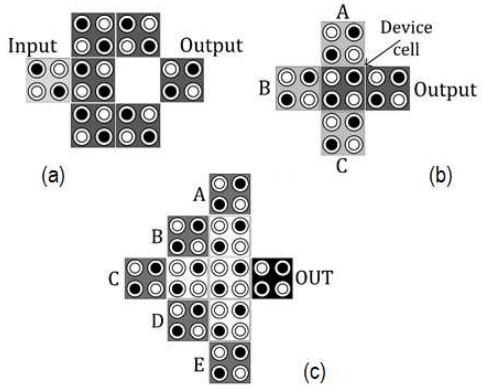

Fig. 3: Layouts of (a) an inverter, (b) a 3-input majority gate, and (c) a 5-input majority gates.

B. Reversible Gates Overview

Due to the inability of conventional logic gates to restore the inputs from the produced outputs, referred to as *reversibility*, and leveraging reversible computing in nanotechnology, i.e. quantum computing, extensive researches have been studied. The main concept of reversible computing is that during the computation no information is lost and the total operation is reversible, i.e. the number of inputs is same as the number of outputs which have mathematically one-to-one mapping.

Commonly-cited reversible gates include Fredkin and Toffoli gates. The Toffoli gate [20] has two control inputs which are copied the first two inputs to the first two outputs and the third output is controlled by the first two inputs. The Fredkin gate [21], also known as CSWAP gate, is a universal reversible gate that controlled by the first input. Here if A=1, the second and third input lines are swapped, otherwise if A=0, then the input lines propagates to the outputs.

Although reversible logic circuits target energy conservation, the leveraging of conventional methods for detecting errors using these circuits faces some challenges. Since one of the most widely used method for error detection is parity checking, parity-preserving reversible gates were sought and introduced [22]. Figure 4 shows the several schematic of 3-input reversible and preserving gates. As shown in Fig. 4(a)-(f) are reversible gates and Fig. 4(g)-(h) are preserving fault-tolerant gates. Only Fig. 4(b) illustrates a preserving reversible gate as identified below.

A reversible gate with parity-preserving capability satisfies the following conditions:

(1) Mapping of its inputs to outputs is bijective,

(2) XOR of the inputs produces a result equal to the XOR of the outputs ($I_1 \oplus I_2 \oplus ... \oplus I_n = O_1 \oplus O_2 \oplus ... \oplus O_n$).

III. Proposed Parity-Preserving Reversible Gate

In this Section, a QCA-based 3-input, 3-output Parity-Preserving Reversible Gate (PPRG) is proposed using one 2-input XOR gate, two invertors, and two 2-input multiplexers (MUXs). PPRG provides a one-to-one input-output mapping, i.e. each set of the input signals is exclusively mapped to a unique set of output signals. Thus, the input

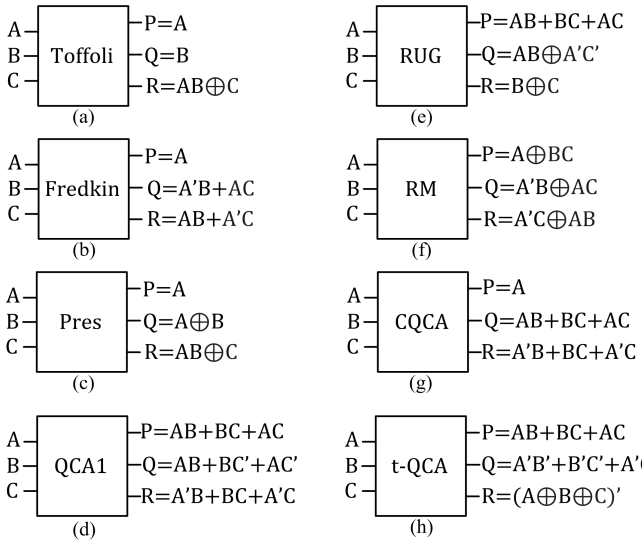

Fig. 4: Block diagram of (a) Toffoli [20], (b) Fredkin [21], (c) Pres [23], (d) QCA1 [6], (e) RUG [24], (f) RM [25], (g) CQCA [26], and (h) t-QCA [27]

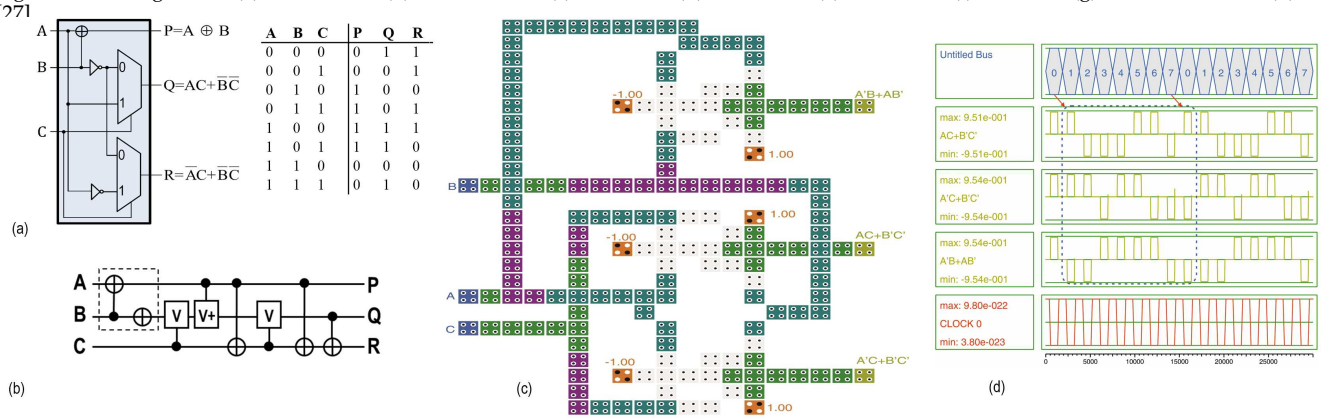

Fig. 5: Proposed Parity-Preserving Reversible Gate (PPRG) (a) schematic and its truth table, (b) quantum representation, (c) layout , and (d) simulation results.

TABLE I: Gate counts comparison for implementation of 13 standard functions.

Standard functions	CQCA [26]	t-QCA [27]	Toffoli [6]	Peres [23]	QCA1 [6]	RUG [24]	RM [25]	Fredkin [26]	PPRG (developed herein)
Reversible/Conservative	No/Yes	No/Yes	Yes/No	Yes/No	Yes/No	Yes/No	Yes/No	Yes/Yes	Yes/Yes
1. F = A.B.C	2	2	2	2	2	3	2	2	2
2. F =A.B	1	1	1	1	1	1	1	1	1
3. F = A.B.C + $\overline{A}$.B.$\overline{C}$	3	8	3	3	2	2	2	3	2
4. F = A.B.C + A.$\overline{B}$.$\overline{C}$	6	2	5	4	3	3	3	4	3
5. F =A.B + B.C	2	2	2	2	2	3	2	2	2
6. F = AB + $\overline{A}$.B.C	5	6	5	3	3	3	2	5	2
7. F = $\overline{A}$.B.C + $\overline{A}$.B.$\overline{C}$ + A.B.$\overline{C}$	6	6	6	4	3	3	3	6	3
8. F= A	1	1	1	1	1	1	1	1	1
9. F= A.B + B.C + A.C	1	1	5	4	1	1	5	5	2
10. F=A.B+$\overline{B}$.C	3	4	3	3	3	3	1	1	1
11. F= A.B+B.C+$\overline{A}$.B.$\overline{C}$	6	2	5	1	4	3	2	6	2
12. F= A.B+ $\overline{A}$.$\overline{B}$	4	1	2	1	2	1	2	2	2
13. F=A.B.C+$\overline{A}$.$\overline{B}$.C+ A.$\overline{B}$.$\overline{C}$ +$\overline{A}$.B.C	3	2	2	3	2	3	2	3	2
Total	43	38	42	32	29	30	28	41	25
Improvement (%)	41.9%	34.2%	40.5%	21.9%	13.8%	16.7%	10.7%	39%	-

signal could be extracted by knowing only the output of the gate, which validates the PPRG reversible property. Figure 5(a) shows the schematic and its related truth table of the PPRG, in which $P = A \oplus B$, $Q = A.C + \overline{B}.\overline{C}$, and $R = \overline{A}.C + \overline{B}.\overline{C}$, where A, B, and C, are inputs and P, Q, and R are outputs of the gate.

To overcome the difficulty of realizing a fault-tolerant reversible gate, our proposed gate is designed in a manner such that it provides equal parity for each input set and its corresponding output, if parity is the number of ones in each set. The truth table, shown in Fig. 5(a) demonstrates the parity-preserving feature of the PPRG. Figure 5(b)

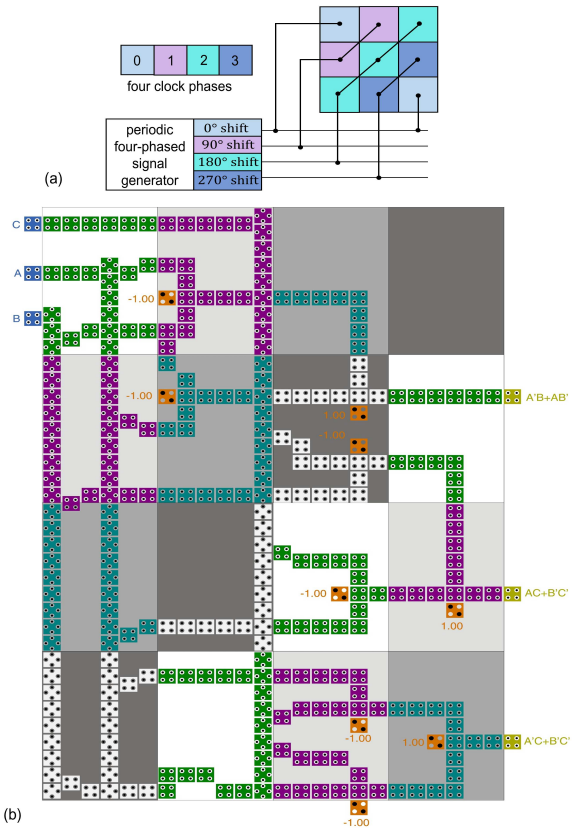

Fig. 6: (a) 2-DW Clocking scheme for the QCA designs [31], and (b) PPRG layout under the 2-DW clocking scheme (PPRG-2DW).

depicts the quantum representation of the PPRG with a quantum cost of seven. The quantum cost of each reversible gate represents its computational complexity, which is calculated by counting the numbers of primitive gates, such as NOT, Controlled-V and Controlled-V+ gates [28].

Herein, a robust single-layered wire-crossing approach, proposed in [29], is leveraged to implement the QCA layout of the PPRG, as shown in Fig. 5(c). Both QCADesigner engines, the bistable approximation and the coherence vector [30] are utilized to validate functionality of the PPRG. The simulation waveform of proposed PPRG is shown in Fig. 5(d). It can be seen that the output is generated after a delay of 1 clock cycle (4 clock phases). The inputs are applied to the PPRG gate at clock phase 0 and the outputs are available at clock phase 4. To consider realistic clock distribution, the Two-Dimensional Wave (2-DW) clocking scheme proposed in [31] is utilized, as shown in Fig. 6(a). The 2-DW approach comprises an array of identical square-shaped zones, which provides low computation time while maintaining acceptable implementation complexity. Figure 6(b) depicts the implementation of our PPRG using 2-DW clocking mechanism (PPRG-2DW) along with the coplanar crossover for wire crossing, which provides outputs after a delay of 1.75 clock cycles (7 clock phases).

In order to evaluate the PPRG performance in comparison with previously proposed reversible gates, it is utilized to implement the thirteen standard 3-variable Boolean functions, which can represent all the possible 256 combinational forms of 3-input functions [32]. The number of gates required for different designs to implement the noted standard functions is listed in Table I. The last row of the table shows the ability of PPRG to synthesis various logical functions with lower average gate count, which verifies the PPRG better performance in term of gate count. The same evaluation procedure can be readily achieved for 2-variable Boolean functions, in which the obtained results show better performance for PPRG-based designs. Since, numbers two and three are prime, we can implement all n-variable Boolean functions using PPRG by partitioning it into m and p, 2-variable and 3-variable functions, respectively, where $n = 2 \times m + 3 \times p$. Accordingly, a reduced partitioning level is projected for the large scale QCA circuits after cascading.

It is worth noting that only the Fredkin gate and our proposed PPRG have both reversibility and parity-preserving features among all the gates listed in Table I. Hence, a comparison between the Fredkin and PPRG is summarized in Table II, which shows the superiority of our design in terms of cell count and area, in addition to the average number of required gates for implementing standard gates which is mentioned beforehand.

TABLE II: Cost comparison of parity-preserving reversible gates.

Reversible Gate	# of gates for Standard Functions	Cell count	Area $(\mu m)^2$	Quantum cost
Fredkin [26]	41	191	0.37	5
PPRG	25	171	0.19	7
Improvement	+39%	+10.5%	+48.64%	-40%

TABLE III: Leakage and switching energy dissipation results.

	leakage energy dissipation (eV)			switching energy dissipation (eV)		
Circuits	**0.5**E_k	**1.0**E_k	**1.5**E_k	**0.5**E_k	**1.0**E_k	**1.5**E_k
RM [25]	0.117	0.318	0.534	0.309	0.251	0.205
t-QCA-II [27]	0.067	0.186	0.316	0.227	0.191	0.161
Fredkin [21]	0.101	0.283	0.482	0.213	0.175	0.143
PPRG	0.058	0.168	0.294	0.193	0.166	0.141
PPRG-2DW	0.079	0.251	0.463	0.577	0.512	0.443

A. PPRG Power Analysis

A QCA power dissipation model has been proposed in [33] whereby total power dissipation of a QCA circuit is divided into two principle components, *leakage* and *switching*. Leakage power is the power losses throughout the clock alternations, from low-to-high or high-to-low, and power losses during the switching period is considered as switching power [34].

In order to expose the power efficiency of PPRG besides its beneficial structure, a comprehensive power analysis and comparison among the previously published Fredkin gate [21], RM [25], t-QCA-II [27] and PPRG is performed. QCAPro tool [33] is utilized as an adequate power estimator tool for estimating the power consumption of the structures at 2K temperature under three different tunneling energy levels (0.5 E_k, 1.0 E_k, and 1.5 E_k). The power dissipation map for both PPRG implementations shown in Fig. 5(c) and Fig. 6(b), are illustrated in Fig. 7 and Fig. 8, respectively. Cells with higher power dissipation are distinguished using thermal hotspots with darker colors.

Average leakage and switching energy dissipation of the two different PPRG implementations as well as the previously proposed gates are listed in Table III. The power analysis results elaborates our PPRG design significant efficiency in both switching and leakage energy dissipation criteria. As expected, the average energy dissipation of the PPRG-2DW implementation is higher than PPRG, due to the realistic clocking considerations.

B. PPRG Fault Analysis

The parity-preserving feature of the PPRG could be utilized along with a fault mapping look-up table, demonstrated in Table IV, to detect the faults occurred in the gate input and output pins. Table IV lists different test inputs required for fault detection, which can be summarized in a unique test vector, called *defect isolation test vector* = {000, 001, 100, 111}.

Herein, the fault isolation process of the PPRG is described through an example. Consider that the input is equal to "111" and the propagated faulty output is "110" instead of being "010". In this case, error could be quickly observed by detecting the parity mismatch between the input parity (P_{in} = "1") and the output parity (P_{out} = "0"). Then, the fault mapping look-up table will be utilized to distinguish both faulty pin and fault type, based on the

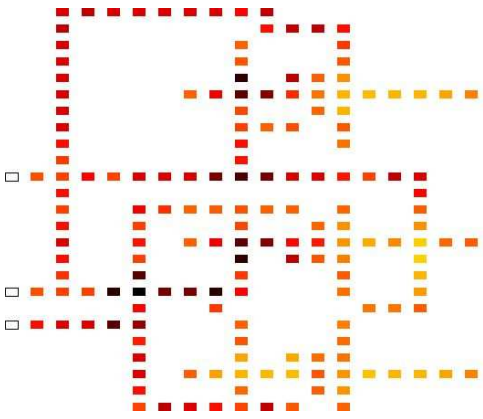

Fig. 7: Power dissipation map for the PPRG in Fig. 5(c).

mapping between test inputs and faulty outputs. As in this example, the look-up table exhibits a stuck at zero fault in the input pin B.

C. Fault-Tolerant Full Adder based on the Proposed PPRG

The Full Adder (FA) is the basic building block of many arithmetic components. Additionally, complex quantum computing requires availability of a compatible conservative reversible adder. Thus, several fault-tolerant FAs based on the reversible gates have been proposed in the literature [35-37].

Figure 9 depicts the schematic and cell layout of our proposed fault tolerant PPRG-based FA. The design leverages three PPRGs, first gate produces A⊕B, and second and third ones produce C_{out} and SUM outputs, respectively. Although conventionally in all reversible gates fan-out is not allowed, due to inherent quasi adiabatic clocking, in QCA logic gates fan-out is permitted [38]. The following equations express the functionality of proposed PPRG-based FA using the following gates depicted in Fig. 9:

$$
\begin{aligned}
&\mathbf{PPRG_1}: \\
&input_A = A,\ input_B = B,\ input_C = C \\
&output_P = P_1 = A \oplus B \\
&\mathbf{PPRG_2}: \\
&input_A = C,\ input_B = \overline{\mathrm{A}},\ input_C = P_1 \\
&output_Q = \mathbf{Q_2} = C.P_1 + \overline{\overline{\mathrm{A}}}.\overline{P_1} = C.(A \oplus B) + A.\overline{(\mathrm{A} \oplus \mathrm{B})} \\
&= A.\overline{\mathrm{B}}.C + \overline{\mathrm{A}}.B.C + A.B = A.B + B.C + A.C = \mathbf{C_{out}} \\
&\mathbf{PPRG_3}: \\
&input_A = P_1, input_B = C, input_C = not\ used \\
&\mathbf{P_3} = P_1 \oplus C = A \oplus B \oplus C = \mathbf{SUM}
\end{aligned}
\tag{2}
$$

A comparative performance analysis between our proposed 1-bit QCA FA and existing structures are given in Table V. Herein, the cost function proposed in [39] is utilized to calculate the overall cost, which is a function of delay (t), number of QCA gates (G), and number of crossovers (C). Overall cost function is expressed as below:

$$
\begin{aligned}
&Cost = (N\_gate + C^m) \times t \\
&N_{gate} = N_{MG}^n + N_{inv},\ t = 4 \times N_{clk}, \\
&C = l \times coplanar\ crossing\ cost
\end{aligned}
\tag{3}
$$

where N_{MG}, N_{inv}, and N_{clk} are number of MGs, inverters, and clocks, respectively. The parameter l denotes the number of layers, i.e. one for single layer and three for multilayer designs. Herein, since all the compared designs are single layer, l equals 1. It is worth noting that N_{MG} and C have greater effect on complexity, power consumption, and fabrication difficulty. Therefore, *Cost* has a quadratic relation with N_{MG} and C, i.e. n=m=2. Thus, the governing equation for cost calculation of 1-bit FA is $FA_{cost} = (N_{MG}^2 + N_{inv} + C^2) \times t \times N_{RG}$, where N_{RG} is number of utilized reversible gates.

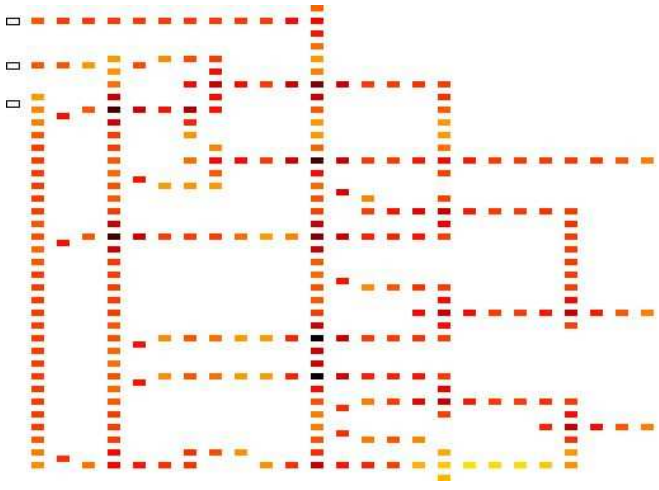

Fig. 8: Power dissipation map for the PPRG-2DW in Fig. 6(b).

Results provided in Table V implies that the overall cost is significantly reduced for our proposed FA. It surpasses the other investigated designs with the exception of QCA1 [6], which is not a parity-preserving design. Thus, proposed PPRG-based FA has the lowest overall cost comparing to the most cost-efficient designs using parity-preserving and reversible gates in their structure.

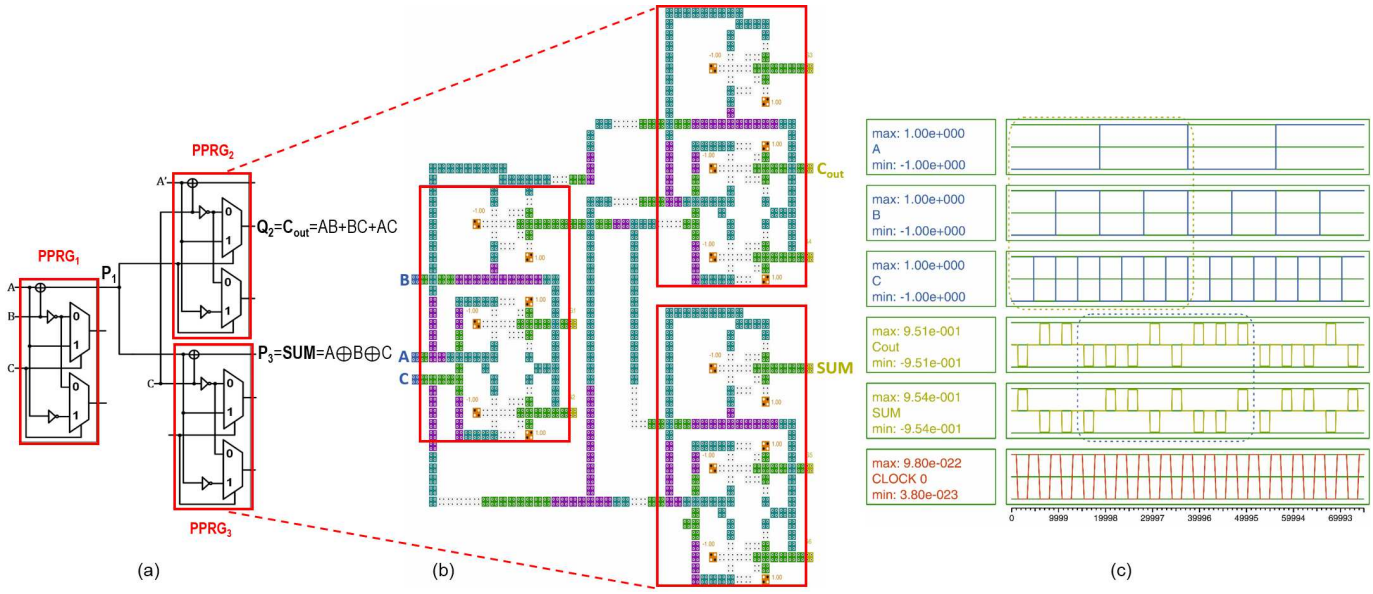

Fig. 9: The proposed reversible full adder (a) schematic, (b) the related cell layout, and (c) simulation results.

IV. Fault Analysis in PPRG-based Logic Designs

In this Section, inherent characteristics of the proposed PPRG are utilized to introduce an effective fault detection and isolation approach for PPRG-based logic circuit designs. Herein, two different PPRG-based circuit topologies are investigated: 1) completely-cascaded arrangement, and 2) partially-cascaded arrangement.

A. Completely-Cascaded Arrangement

Figure 10 shows the structure of the completely-cascaded PPRG-based designs, in which all the three output terminals of each PPRG is directly connected to the inputs of the subsequent gate. Table VI lists the output states in a completely-cascaded design according to the eight possible inputs. For instance, the first row of the table describes that applying a_0 as the primary input of the design, will result in obtaining a_0 as the level 4 output, i.e. the output of the fourth cascaded PPRG. Therefore, the period of the primary input propagation in the different output levels of a completely-cascaded PPRG-based design depends on the primary input, and can be obtained from the last column of the table. Herein, Table VI is defined as a *select matrix* to be utilized in the following pseudocode which describes the algorithm proposed in this paper for fault detection and isolation of the PPRG-based circuits:

TABLE IV: Stuck-at fault mapping Look-Up Table for the PPRG gate.

Fault suspicious PIN	Test Input "ABC"	Expected Output "PQR"	Faulty Output "PQR"	Stuck At Fault
A	100	111	011	Zero
	000	011	111	One
B	111	010	110	Zero
	001	001	101	One
C	111	010	000	Zero
	000	011	001	One
P	100	111	011	Zero
	111	010	110	One
Q	111	010	000	Zero
	001	001	011	One
R	100	111	110	Zero
	111	010	001	One

TABLE V: Comparison of fault tolerant full adder deigns.

Reversible Gate	N_{MG}	N_{inv}	C	t	N_{RG}	***Overall Cost***	***Cell count***	***Area*** $(\mu m)^2$
Fredkin [21]	6	8	5	4	5	*1380*	955	1.85
Toffoli [20]	4	5	3	4	4	*480*	672	1.48
QCA1 [6]	3	7	4	2	3	*192*	438	0.48
RUG [24]	7	11	10	8	2	*2560*	594	0.92
PPRG	6	6	3	4	3	***522***	***513***	***0.57***

Procedure: ERROR DETECTION

1: $Main_{input} \leftarrow$ Convert $Primary_{input}$ to a_0,...,a_7 type
2: ▷ based on the $Main_{input}$ select proper row of selection_matrix and copy into Test_Array
3: $Test\_Array \leftarrow$ selection_matrix[$Main_{input}$]
4: $Design\_Length \leftarrow Length$ (PPRG-based design)
5: ▷ Number of cascaded PPRGs in design
6: $F_1 \leftarrow Design\_Length/Length$ (Test_Array)
7: $F_2 \leftarrow Design\_Length$ mod $Length$ (Test_Array)
8: **if** $Main_{output} == Test\_Array[F_2]$ **then**
9: Print "Fault free!"
10: **Break**
11: **else**
12: Print "Fault detected!"
13: ▷ Run Error Isolation Procedure.
14: **end if**
15: **end procedure**

Procedure: ERROR ISOLATION

1: **for** $i \leftarrow 0; \quad i < F_1; \quad i \leftarrow i+1$ **do**
2: ▷ repeat this function F_1 times
3: **if** $Output(section[i]) \neq Main_{input}$ **then**
4: Print "Fault detected in section[i]!"
5: **goto** *28*.
6: **end if**
7: **end for** ▷ No error detected in F_1 sections.
8: Print "Fault detected in section[F_1]!"
9: ▷ Section F_1 is the last section in the design.
10: **goto** *35*.
11: Check Section[i] for Fault Isolation operation:

TABLE VI: Stuck-at fault mapping for the PPRG gate.

Primary Input	**Level 1 Output**	**Level 2 Output**	**Level 3 Output**	**Level 4 Output**	**Max Level**
000 (a_0)	a_3	a_5	a_6	a_0	4
001 (a_1)	a_1				1
002 (a_2)	a_4	a_7	a_2		3
003 (a_3)	a_5	a_6	a_0	a_3	4
004 (a_4)	a_7	a_2	a_4		3
005 (a_5)	a_6	a_0	a_3	a_5	4
006 (a_6)	a_0	a_3	a_5	a_6	4
007 (a_7)	a_2	a_4	a_7		3

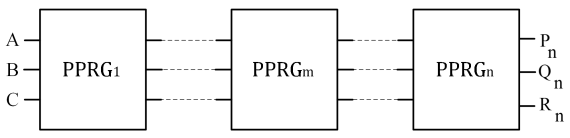

Fig. 10: Completely-cascaded PPRG-based design arrangement

for $k \leftarrow 0$; $k < Length(Test\_Array)$; $k \leftarrow k+1$ **do**
    if $Parity(PPRG[k]) \neq Parity(Primary_{input})$ **then**
        Print "Faulty PPRG detected!"
        Break
    end if
end for
Check Section[F_1] for Fault Isolation operation:
for $k \leftarrow 0$; $k < F_2$; $k \leftarrow k+1$ **do**
    if $Parity(PPRG[k]) \neq Parity(Primary_{input})$ **then**
        Print "Faulty PPRG detected!"
        Break
    end if
end for
end procedure

Figure 11 describes the above algorithm through an example. Consider the cascaded PPRG-based design in Fig 11(a). If the design is fault free then applying "000" (a_0) as the primary input produces a_3 as a desired output. However, if the design is considered as fault suspicious, the Error Detection procedure will be performed. According to above mentioned procedure, the Design_Length, F_1 and F_2 are equal to five, one and one, respectively. First, output of the design is compared with the desired output which can be obtained using *Test_Array* which is a $1\times$ *Design_Length* matrix that is equal to one of the rows of the select matrix. If they are equal then the design is fault free; otherwise, a fault is detected and Error Isolation procedure will be run. In this procedure, all sections are searched in order to find the faulty section by using the reversibility attribute of PPRGs. Then, the faulty PPRG can be detected in the isolated section via utilizing the parity-preserving feature of PPRG. Figure 11(a) and 11(b) show the fault free and faulty circuits, respectively. The required error detection and isolation steps are readily demonstrated in Fig. 11(c).

B. Partially-cascaded Arrangement

In this PPRG-based circuit design, there are various connections between input and output terminals of the PPRGs, which might result in a design that is not completely-cascaded. Additionally, some of the outputs of the PPRGs might be unused in the structure of the design, which are called Garbage (G) outputs.

In order to make the fault detection and isolation algorithm proposed for the completely-cascaded arrangement applicable for partially-cascaded PPRG-based designs, MUXs, and interconnections are added to the structure of the design. The primary design with n PPRGs is combined with $3 \times (n-1)$ MUXs which are placed at the input terminals of the PPRGs, as shown in Fig. 12. Moreover, a fault flag is provided as the MUX selector control signal to determine the operating mode of the design. If a fault occurs, the fault flag will be set to "1" and MUXs will convert the partially-cascaded PPRG design to the completely-cascaded arrangement. Then the error detection and isolation procedure will be run. Otherwise, fault flag is equal to "0" and the design operates in regular mode. This topology is affected by overhead from MUXs which can be alleviated by using an ultra-dense MUX design which is proposed in [40].

Example: Consider Fig. 13(a) realizing a partially-cascaded PPRG-based design which functions $F_7 = \overline{A}BC + \overline{ABC} + AB\overline{C}$. Assigning $Primary_{in}$ = "011" then the desirable out is "011". The fault-free state is shown in Fig. 13(b) in which `fault_flag`="0". The blue-colored lines shows the interconnections which provides the desired output. Now assume that the circuit is faulty and the obtained output is equal to "111". Therefore, `fault_flag`="1" will be equal to "1" as a result of which the design will be converted to a completely-cascaded arrangement. Interconnections in fault detection operating mode are shown by red lines in Fig. 13(c). Finally, the proposed fault detection and isolation algorithm will be executed to distinguish the faulty PPRG.

V. Conclusion

Energy-sparing gate designs using emerging technologies allow extensions in beneficial directions by embracing reversibility and the inclusion of inherent fault observability. In the case of QCA, the availability of appropriate clocking paradigms enables the utilization of parity-preserving gates. The feasibility of such features lies not only in the design of the gate itself, but also an efficient corresponding bit-level self-checking methodology. Herein, PRRG focuses first on achieving high Boolean expressiveness for compact logic realizations. This results in significant improvements in the design of standard logic functions having three variables reducing average gate count by at least 10.7% compared to previous designs. The utility of PPRG is further enhanced by its resilient cascadability. This is evident in the realization of a fault-tolerant reversible 1-bit full adder using 171 QCA cells occupying only

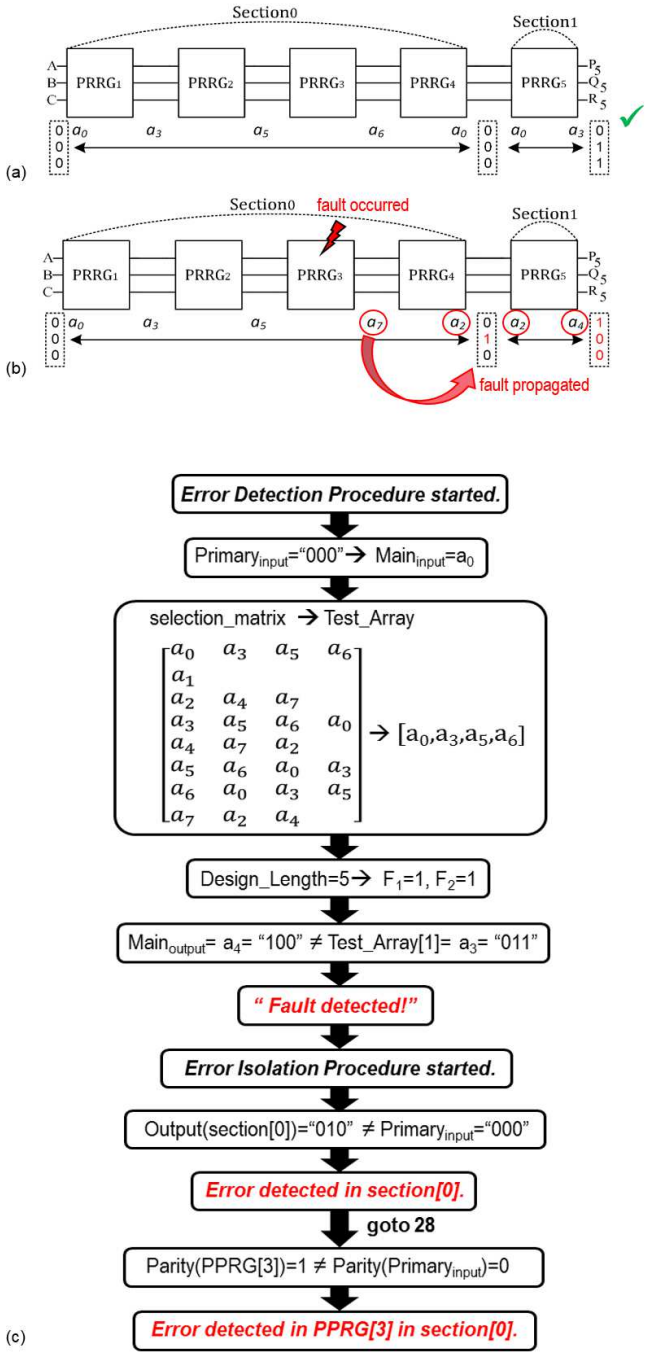

Fig. 11: Cascaded PPRG-based design, (a) fault-free design, (b) faulty design, and (c) error detection and isolation steps.

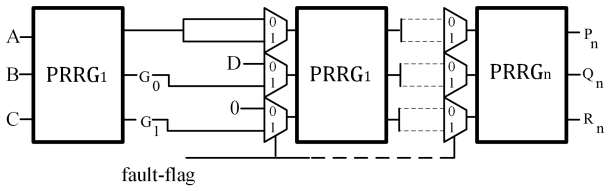

Fig. 12: Partially-cascaded PPRG-based design arrangement.

0.19 μm^2 area. Beyond previous works, we also emphasize the beneficial role of utilizing PPRGs to yield larger QCA robust reversible circuits. While the fault detection and isolation methodology based on progressive suspect reduction as developed herein, provides a concise solution for configurable resiliency, other future extensions such as syndrome-based error correction could be investigated.

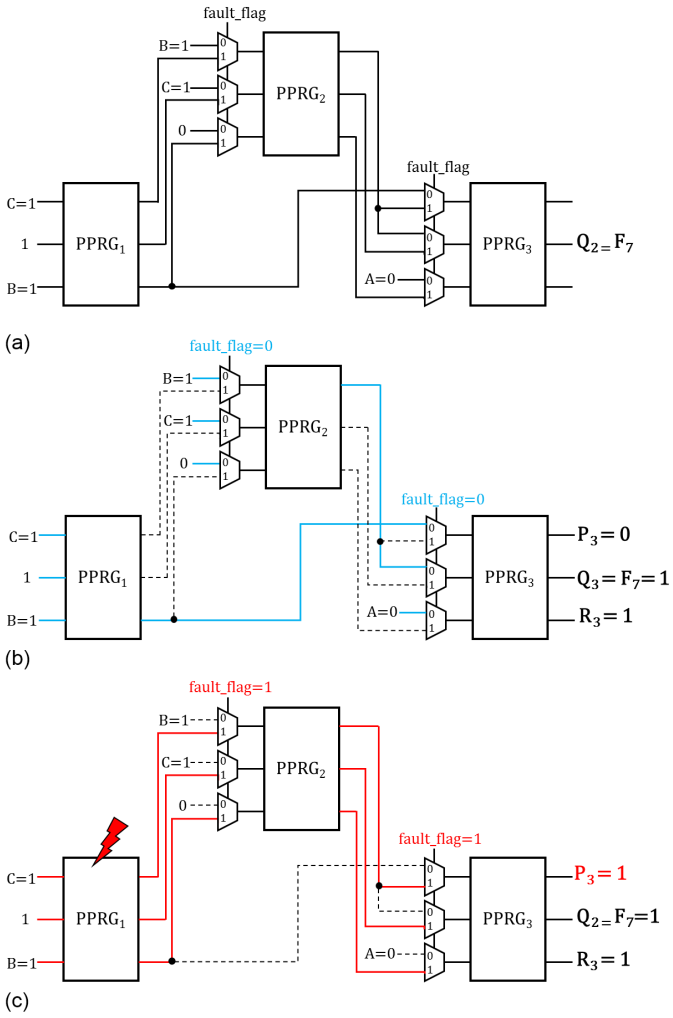

Fig. 13: (a) Partially-cascaded PPRG-based design, (b) fault-free operating mode which set `fault_flag` to 0, and (c) faulty operating mode which means `fault_flag` equals 1, so the circuit converted to a completely-cascaded arrangement.

References

[1] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, *Quantum cellular automata,* Nanotechnology, vol. 4 (1993) p. 49.

[2] C. S. Lent, and P. D. Tougaw, *A device architecture for computing with quantum dots,* Proceedings of the IEEE, vol. 85 (1997) pp. 541-557.

[3] C. H. Bennett, *Logical reversibility of computation,* IBM journal of Research and Development. 17(6), (1973) pp.525-532..

[4] R. Landauer, *Irreversibility and heat generation in the computing process,* IBM journal of research and development, 5(3), (1961) 183-191.

[5] C. S. Lent, M Liu, and Y. Lu, *Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling,*Nanotechnology. 17(16),(2006) 4240.

[6] X. Ma, J. Huang, C. Metra, and F. Lombardi, *Reversible gates and testability of one dimensional arrays of molecular QCA,* Journal of Electronic Testing. 24(1-3), (2008) 297-311.

[7] A. Roohi, H. Khademolhosseini, S. Sayedsalehi, and K. Navi, *Implementation of reversible logic design in nanoelectronics on basis of majority gates,* Computer Architecture and Digital Systems (CADS), 2012 16th CSI International Symposium on, 2012, IEEE.

[8] H. Thapliyal H, N. Ranganathan, *Reversible logic-based concurrently testable latches for molecular qca,* Nanotechnology, IEEE Transactions on. 9(1), (2010) 62-9.

[9] B. Sen, M. Dutta, S. Some, and BK. Sikdar, *Realizing Reversible Computing in QCA Framework Resulting in Efficient Design of Testable ALU,* ACM Journal on Emerging Technologies in Computing Systems (JETC). 11(3), (2014) 30.

[10] W. Porod, C. S. Lent, G. H. Bernstein, A. O. Orlov, I. Amlani, G. L. Snider, and J. L. Merz, *Quantum-dot cellular automata: computing with coupled quantum dots,* Int. J. Electron. 86, 549 (1999).

[11] G. Toth and C. S. Lent, *Quasiadiabatic switching for metal-island quantum-dot cellular automata*, J. Appl. Phys., vol. 85, pp. 29772984, 1999.

[12] A. Orlov, A. Imre, G. Csaba, L. Ji, W. Porod, and G. H. Bernstein, *Magnetic quantum-dot cellular automata: recent developments and prospects,* Journal of Nanoelectronics and Optoelectronics 3, no. 1 (2008): 55-68.

[13] S.E. Frost, T.J. Dysart, P.M. Kogge, and C.S. Lent, *Carbon nanotubes for quantum-dot cellular automata clocking*, in Proc. IEEE Conf. Nanotechnology, 2004, pp. 171173.

[14] K. Hennessy, and C.S. Lent, *Clocking of molecular quantum-dot cellular automata,* Journal of Vacuum Science & Technology B, 19(5), (2001) 1752-1755.

[15] K. Walus, G.A. Jullien, and V.S. Dimitrov *Computer arithmetic structures for quantum cellular automata,* Signals, Systems and Computers, 2004 Conference Record of the Thirty-Seventh Asilomar Conference on, 2003, IEEE.

[16] K. Navi, S. Sayedsalehi, R. Farazkish, and M. R. Azghadi, *Five-input majority gate, a new device for quantum-dot cellular automata,* Journal of Computational and Theoretical Nanoscience 7, no. 8 (2010) pp. 1546-1553.

[17] K. Navi, R. Farazkish, S. Sayedsalehi, and M. R. Azghadi, *A new quantum-dot cellular automata full-adder,* Microelectronics Journal. 41(12), (2010) 820-6.

[18] A. Roohi, H. Khademolhosseini, S. Sayedsalehi, and K. Navi *A symmetric quantum-dot cellular automata design for 5-input majority gate,* Journal of Computational Electronics. 13(3), (2014) 701-8.

[19] R. Akeela, and M.D. Wagh*A five-input majority gate in quantum-dot cellular automata,* In NSTI Nanotech, vol. 2, (2011) pp. 978-981.

[20] T. Toffoli, *Reversible Computing,* Tech memo MIT/LCS/TM-151, MIT Lab for Computer Science, 1980..

[21] F. Fredkin, and T. Toffoli, *Conservative logic,* Springer; 2002.

[22] B. Parhmai, *Fault-tolerant reversible circuits,* Signals, Systems and Computers, 2006 ACSSC'06 Fortieth Asilomar Conference on; 2006: IEEE.

[23] A. Peres, *Reversible logic and quantum computers,* Physical review A. 32(6), (1985) 3266.

[24] B. Sen, D. Saran, M. Saha, and B. K. Sikdar *Synthesis Of Reversible Universal Logic Around QCA With Online Testability,* Electronic System Design (ISED), 2011 International Symposium on; 2011: IEEE.

[25] B. Sen, M. Dutta, M. Goswami, and B. K. Sikdar, *Modular Design of testable reversible ALU by QCA multiplexer with increase in programmability,* Microelectronics Journal. 45(11), (2014) 1522-32.

[26] H. Thapliyal H, N. Ranganathan, *Conservative qca gate (cqca) for designing concurrently testable molecular qca circuits,* VLSI Design, 2009 22nd International Conference on; 2009: IEEE.

[27] B. Sen, M. Dutta, and B. K. Sikdar, *Efficient design of parity preserving logic in quantum-dot cellular automata targeting enhanced scalability in testing,* Microelectronics Journal. 45(2), (2014) 239-48.

[28] D. M. Miller, D. Maslov, and G. W. Dueck, *A transformation based algorithm for reversible logic synthesis*, in Proc. Design Automation Conf. , Jun. 2003, pp. 318323.

[29] S. Angizi, E. Alkaldy, N. Bagherzadeh, and K. Navi, *Novel robust single layer wire crossing approach for exclusive or sum of products logic design with quantum-dot cellular automata,* Journal of Low Power Electronics. 10(2), (2014) 259-71.

[30] K. Walus, T. J. Dysart, G. Jullien, and A. R. Budiman, *QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata,* Nanotechnology, IEEE Transactions on. 3(1), (2004) 26-31.

[31] V. Vankamamidi, M. Ottavi, F. Lombardi, *Two-dimensional schemes for clockingtiming of QCA circuits*, IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems 2008; 27(1)3444.

[32] R. Zhang, K. Walus, W. Wang, and G. Jullien, *A method of majority logic reduction for quantum cellular automata,* Nanotechnology, IEEE Transactions on. 3(4), (2004) 443-50.

[33] S. Srivastava, A. Asthana, S. Bhanja, and S. Sarkar, *QCAPro-an error-power estimation tool for QCA circuit design,* Circuits and Systems (ISCAS), 2011 IEEE International Symposium on; 2011: IEEE.

[34] S. Sheikhfaal, S. Angizi, S. Sarmadi, M. H. Moaiyeri, and S. Sayedsalehi, *Designing efficient QCA logical circuits with power dissipation analysis,* Microelectronics Journal. 46(6), (2015) 462-71

[35] J. W. Bruce, M. A. Thornton, L. Shivakumaraiah, P. Kokate, and X. Li, *Efficient adder circuits based on a conservative reversible logic gate,* VLSI, 2002 Proceedings IEEE Computer Society Annual Symposium on; 2002: IEEE.

[36] S. K. Mitra, and A. R. Chowdhury, *Minimum cost fault tolerant adder circuits in reversible logic synthesis,* VLSI Design (VLSID), 2012 25th International Conference on; 2012: IEEE.

[37] B. Sen, T. Adak, A. S. Anand, and B. K. Sikdar*Synthesis of reversible universal QCA gate structure for energy efficient digital design,*TENCON 2011-2011 IEEE Region 10 Conference; 2011: IEEE.

[38] K. K. Yadavalli, A. O. Orlov, J. P. Timler, C. S. Lent, and G. Snider, *Fanout gate in quantum-dot cellular automata,* Nanotechnology. 18(37), (2007) 375401.

[39] W. Liu, L. Lu, M. O'Neill, and E. E. Swartzlander, *A first step toward cost functions for quantum-dot cellular automata designs*, IEEE Transactions on Nanotechnology, Vol. 13, No. 3, pp. 476-487, 2014.

[40] A. M. Chabi, A. Roohi, H. Khademolhosseini, S. Angizi, R. F. Demara, and K. Navi, *Cost-Efficient QCA Reversible Combinational Circuits Based on a New Reversible Gate,* Computer Architecture and Digital Systems (CADS), 2015 18th CSI International Symposium on, 2015, IEEE.

[41] S. E. Frost-Murphy, M. Ottavi, M. P. Frank, and E. P. DeBenedictis, *On the Design of Reversible QDCA Systems,* Sandia National Laboratories, Albuquerque, NM, Tech. Rep. SAND2006-5990, 2006.

[42] A. M. Ottavi, S. Pontarelli, E. P. DeBenedictis, A. Salsano, S. Frost-Murphy, P. M. Kogge, F. Lombardi, *Partially reversible pipelined QCA circuits: combining low power with high throughput,* Nanotechnology, IEEE Transactions on. 10(6), (2011) 1383-1393.

[43] R. Wang, A. Pulimeno, M. R. Roch, G. Turvani, G. Piccinini and M. Graziano, *Effect of a Clock System on Bis-Ferrocene Molecular QCA,* Nanotechnology, IEEE Transactions on. 15(4), (2016) 574-582.