IET Nanobiotechnology

Research Article

Router design for nano-communication using actin quantum cellular automata

ISSN 1751-8741
Received on 19th May 2020
Revised 21st June 2020
Accepted on 10th July 2020
E-First on 1st September 2020
doi: 10.1049/iet-nbt.2020.0186
www.ietdl.org

Biplab Das[1] ✉, *Debashis De*[1]

[1]*Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, West Bengal, BF-142, Sector-1, Saltlake, Kolkata-700064, India*

✉ *E-mail: biplab.das52@gmail.com*

Abstract: Logic expressions can be designed from actin filaments. It is a protein that makes the cellular structure and plays an important role in intracellular communication. Nano communication technique has been established using actin cellular automata. Among several rules, (1, 30) and (4, 27) rules have been used to design 2 to 1 multiplexer, 4 to 1 multiplexer, 1 to 2 demultiplexer and 1 to 4 demultiplexer. Router or data selector has been made of using multiplexer and demultiplexer. Three novel circuits such as multiplexer, demultiplexer and nano-router have been designed using the projected mechanism. The primary focus of this proposed technique is on different designs of the multiplexer, demultiplexer and minimum cell count with minimum time steps. The different router circuits have been simulated with the help of Simulink by which output has been verified for different circuits. Stuck at fault analysis is also done in this study. Device density and power consumption have also been included in this study. A comparative analysis of the different designs of the router provides a better concept of circuit optimisation. Furthermore, this study analyses convenient forthcoming applications in nano-technology and nano-bio-molecular systems involving the proposed parameters.

1 Introduction

Actin is depicted to produce consistent progress between monomeric globular actin (G-actin) shape and filamentous actin (F-actin) shape. Actin is a stand out amongst the most bounteous organic proteins obtained from the cytoplasm of most eukaryotic cells. Actin is more often than not considered as a muscle-fibre and furthermore known as thin fibres and really an essential piece of the cytoskeleton [1]. Cytoskeleton just not keeps up the cell structure, yet in addition, it goes about as the processor of cell data. Actin fibre is comprised of little globular actin (G-actin) monomers contained inside a double helical structure. A modern form logic system can be introduced by adopting collision-based computing (CBC) [2], a ubiquitous logic design thought upheld over F-actin [3]. The automata are ascertained with the notion of partitioned one-dimensional quantum cellular automata (P-1D-QCA) and are shown in Fig. 1*a*. Since 90's cytoskeletal microfilaments filaments and microtubules filaments have led to design logic systems [4]. Comparative impacts of the cytoskeleton with F-actin like structure are talked over in other literature [5, 6]. Fig. 2*a* depicts the structure of an F-actin and its analogous cell mapping. After exploring the natural behaviour of F-actin, researchers have proposed to design molecular networks using it [7–9]. In light of the inferences of the actin quantum automata [4], this paper presents the design of a data line selector cum router considered as the element of any communication network. The diameter of an F-actin is 7 nm [2]. This model is expected to shorten the successful circuit size and include effectiveness to any current CMOS innovation-based circuit. Actin inspired QCA generated algorithms are very useful for cryptographic techniques and thereby for security purposes, owing to the algorithms' methodical, logical and simple attributes [10]. Various types of actin networks give mechanical support to cells, and provide trafficking routes through the cytoplasm to aid signal transduction [4].

2 Motivation and contribution

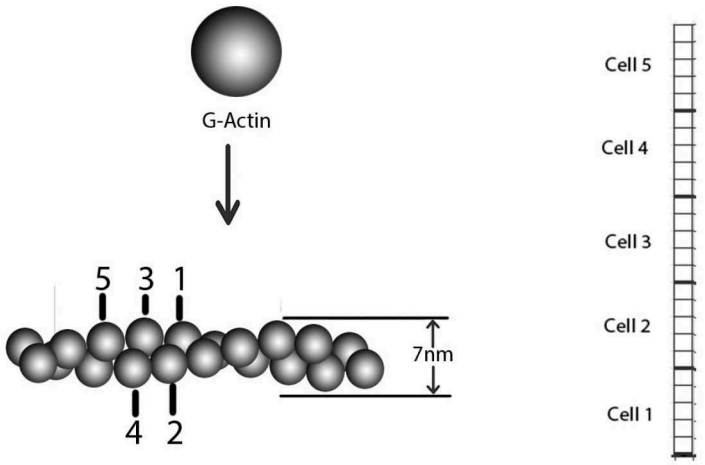
The semiconductor device innovation has been amplified quickly in the previous decades; however the capacity of high density is required. This data line selector model of F-actin quantum cellular automata (QCA) comprehends the potential outcomes of efficacious and tiny designs of unconventional logic circuits. Double helical-shaped filamentous actin having width 7 nm can be produced in room temperature. Conventional semiconductor technology has issues with size and the conceivable number of chips in the devices, whereas in unconventional semiconductor technology one can overcome the limitation of the traditional technology by using actin QCA (AQCA) concept. We have discussed elaborately this effective design. So AQCA concept of circuit configuration is uncovered in this study. The contributions of the paper are as follows:

- Defining the state transition table for a rule.
- Designing 2 to 2 and 4 to 4 data line selector cum router.
- Device density.
- Power consumption.
- Fault analysis.

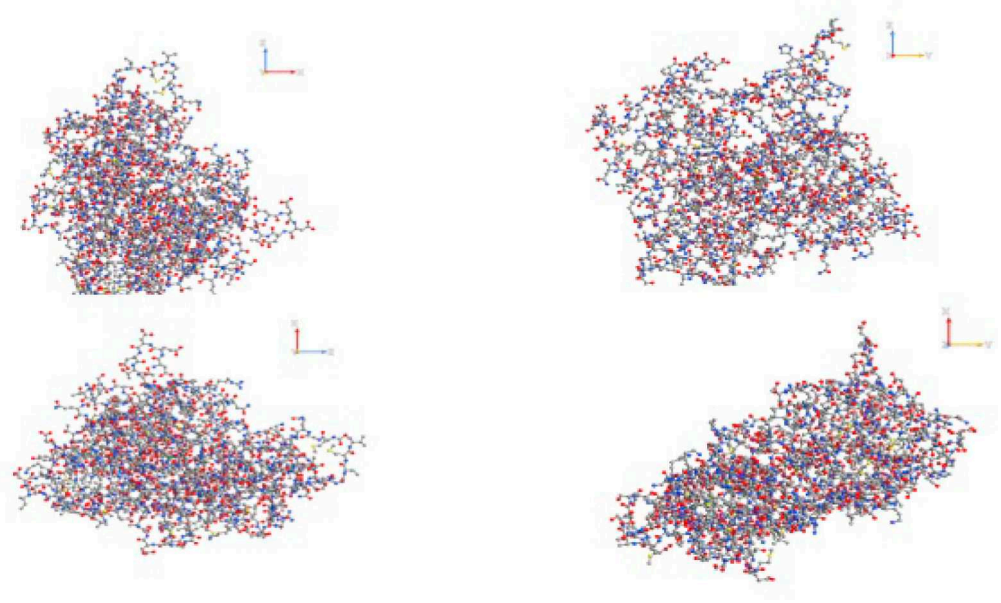
3 Related work

A cellular automaton is a well-defined computational theory. Actin-based QCA concept is beneficial for bio-inspired computing [11]. Without using the classical bits, only qubits are used in this article to develop the logical activities of QCA [12]. Nowadays, molecular-based QCA cells are being designed with the concept of two-dot QCA cell and four-dot [13–18] QCA cell. These concepts are also used to design semiconductor-based on various materials [19–21]. Establishing digital logic in CBC is an idea that came from a special feature of it, i.e. occurring collision of the particles with each other. Protein channel systems (P-systems) are realised accountable for transmitting between the cells of QCA methods [16, 22]. Actin like structure concept is used to develop unconventional computing [23]. Further study ascertains that memory and impulses possessed by cells of it can be controlled by putting in electrical signals in the p-system [24]. Actin filaments were observed to have the capability of self-assembling. F-actin takes part in vesicle movement [25]. AQCA concept is also applicable for designing the concept of droplet machine and voltage circuit [26, 27]. The Boolean logic function can be developed by the intracellular connectivity characteristics of AQCA [28]. Actin filament can be used as an actin network that

© The Institution of Engineering and Technology 2020



a



b

Fig. 1 *Structure of F-actin and mapping of different cells and F-actin characteristics*
(a) Mapping of actin molecules into a one dimensional array of cells, ***(b)*** F-Actin representation by ATK VNL QuantumWise in various dimension

has two or three states [29]. Actin filament networks can act as a computational medium where different cell structures of actin are used for developing the logical concept [30]. AQCA concept is also capable of describing the effectiveness of molecules and the design of digital circuitry [31].

4 Basics of QCA

QCA is a formulation of quantum computation where Qubits represent information instead of classical bits [1, 32, 33]. The working principle of this type of systems the main concept in a grid of 'cells', which can be formed as binary or entangled. Some relevant and useful techniques are used to develop the concept throughout this paper.

4.1 Cell in QCA

In cellular automata (CA) cell is a unit of functioning. A cell can be in any state like an active or an inactive state as well as in a superposition state. A partitioned CA is put under thought in the following discussion where cells are partitioned into five substates, as stated in [32, 33].

4.2 State definition in QCA

Three states have been established in the concept of QCA, i.e. zero ('0'), one ('1') and superposition state. Superposition state is a combination of multiple states of '0' and '1' at the same time. For the simplification of designing, we have used only '0' and '1' state in this paper. The blank square boxes represent the substates that are not active, square boxes that are coloured with black colour represents the substates that are active.

4.3 Collision-based computing (CBC) on AQCA

In a P1-QCA [34], the system works with the core concept of CBC [2]. The outline design of this system is sketched below. Fig. 1*a* shows how the mapping of actin monomers belonging to an F-actin chain into a P1-QCA is done. Fig. 2*a* shows how a cell movement and collision between cells in a different timespace occur. Fig. 2*a* illustrates pulses coming from different adjacent cells influencing substates of a cell. The third substate of cell 3 will remain unchanged as it is cell 3 itself and 1st, 2nd, 4th and 5th substates of cell 3 will change depending on the active or inactive state of substate 3 of cell 3. Inside of an arbitrarily long actin chain similar movements of substates are applied to every cell. Such collisions happen when a cell experiences two or additional incoming pulses from its adjacent cells. A collision can be widely categorised as reflection, annihilation, crossover etc. [35]. In a collision of vesicles, microscopic structures of lipid-bound nutrients transported by actin filaments can also be spotted [36]. The rule matrix is formed to sort out collisions and to describe the state transformation of the neighbouring cell [10]. Collisions can be

© The Institution of Engineering and Technology 2020

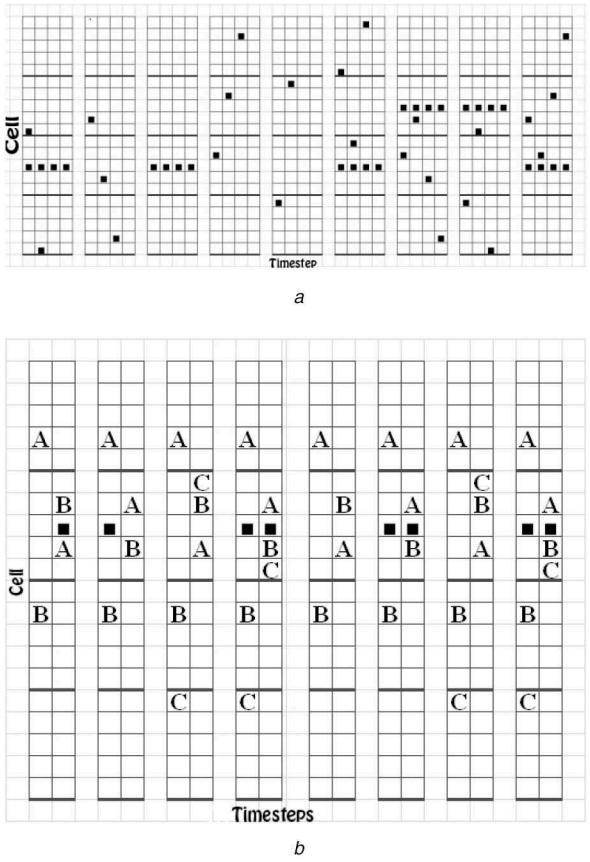


Fig. 2 *Collision between cells in different timesteps with different rules*
(a) Cell moves and collision between cells in different timesteps, ***(b)*** Collision with Rule (4, 27)

Table 1 Colour combination of F-actin images

Atoms	Colour coding
hydrogen (H)	white
carbon (C)	black
nitrogen (N)	dark blue
oxygen (O)	red
sulphur (S)	yellow
calcium (Ca)	dark green
phosphorus (P)	orange

determined and exercised accordingly, as in [37]. In this paper to design the proposed circuit rule (1, 30) and (4, 27) is randomly referenced from the set of 32 unitary rules described in [1, 10].

4.4 Details of actin automata

Actin is a contractile muscle protein that is found in nearly all eukaryotic cells. The protein size and diameter of actin is 43 kDa [1] and 8 nm, respectively. The structure of actin is extremely conserved and it participates in numerous protein–protein interactions, including the formation of F-actin [38]. The key to actin function in many cell movement processes is its dynamic polymerisation and depolymerisation activity [39]. Actin is mainly found in two distinct conformations and they are globular (G) actin and fibrous (F) actin [40].

The transition from monomeric G-actin to polymerised F-actin has gained a lot of attention as this process regulates many physiological processes by interacting with other actin-binding proteins and thereby causing cell motility. The Protein Data Bank (PDB) [5] is an online repository of many characterised proteins, nucleic acids and other large molecules from various biological organisms, whose crystal structures are deposited in PDB. The three-dimensional (3D) chemical structures of all the available macromolecules in the database could be found and other known molecular structure formats for the purpose of downloading, visualisation and use in further computational studies. Previously, F-actin was experimentally isolated from the skeletal muscle of a European rabbit (*Oryctolagus cuniculus*) for the purpose of modelling the protein using X-ray fibre diffraction [41] technique, implemented at various intensities with a net resolution of 3.3 Å° in the radial direction and 5.6 Å° along the equatorial plane. In this work, from the PDB (database) we can get the 3D crystal formation of the characterised F-actin [42] and can also get the simulation result with the help of ATK-VNL QuantumWise simulation package which is shown in Fig. 1*b*. Details of the protein (molecule) model like the total no. of atoms present, names of constituting atoms, individual bond distances between atoms in Angstrom, relative atomic positions in 3D space and dimensions of the molecule could be acquired from QuantumWise. Employing QuantumWise, the PDB file for the F-actin model is launched and on a white background, the 3D structure of the same is aligned in different geometric orientations and respective screenshots of the alignments are captured thereafter. The heterogeneity in the atomic population is exhibited by different colours that are indexed in Table 1. QuantumWise generated screenshots of the F-actin model specific to the European Rabbit, existing at different spatial orientations are compiled in Fig. 1*b*.

4.5 Rule (1, 30)

In rule (1, 30), 1 is represented in five-bit binary ‘00001’ and 30 as ‘11110’. The matrix representation of rule (1, 30) is (‘00001’, ’11110’). It shows that how the excitation state of a cell can take values like ‘1’, i.e. excited state or ‘0’, i.e. the idle state. The rows represent the current excitation of corresponding cells. The columns represent the number of incoming pulses from the corresponding neighbourhood of four cells [32]. The ‘0’ as the first element in the matrix represents the current excitation of a cell stays at ‘0’ if no pulse arrives from its neighbours. Similarly, the ‘1’ as the 5th element of the first row represents an inversion of excitation from 0 to 1 for four incoming pulses.

© The Institution of Engineering and Technology 2020

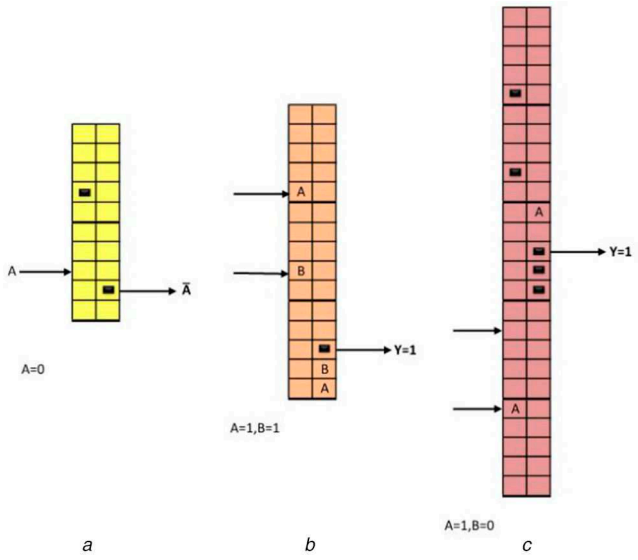


Fig. 3 *Realisation of*
(a) NOT gate, ***(b)*** AND gate, ***(c)*** OR gate

4.6 Rule (4, 27)

The matrix representation of rule (4, 27) is ('00100', '11011') [10]. From the matrix form, it can be observed that this rule inverts the central substate of a collided particle if exactly two other substates are active in the target cell. In other words, the activating part of this rule is '2'. That is when exactly two neighbours are active; then the central substate turns on if it is currently in the resting state. However, if the central substate is currently inactive state, it will deactivate for the same condition. Fig. 2*b* can be consulted for an overview of collisions with this rule. It can be observed that the first rule does not affect on the collision of two particles as a member of 'rules of the first group' as the activation of any cell changes only if four incoming pulses arrive from the neighbours' collision of up to three elements does not effect on individual propagation.

5 Proposed work

The design of the data line selector cum router is proposed using different methods in this segment. This data line selector cum router, which is considered as radical of any communication system is comprised of a mux and a demux [43]. Appropriate input data line depending on the selected signal input(s) is selected and sent by the mux as data bit to the demux. At the receiving end, according to the select signal input(s) of the demux, the output data line is selected. At first, a NOT gate, an AND gate and an OR gate are realised by the QCA-based design embedded with the rule (4, 27) and (3, 28). Finally, the design of these gates guides to the proposed design of the data line selector cum router. Collisions between the incoming pluses or signals in the substates of a single cell rule to build the logic gates and circuits [36]. In Fig. 2*b*, a horizontal axis is assumed to denote the timestep and the series of cells are vertically placed. The left-most cell series is considered at timestep 1.

5.1 Realisation of NOT gate, AND gate and OR gate

A NOT gate can be achieved by only two cells, as shown in Fig. 3.

Cell 2 is introduced by [00010] and gate input 'A' is given at the substate no. 3 of cell 1 (c1s3). Substate no. 4 of cell 1 (c1s4) reflects the output of the gate. If the input 'A' is an inactive state, c1s4 will active instead of output substate no. 2 of cell 1 (c1s2). This phenomenon leads to achieving optimised NOT gate. The design of an AND gate is presented in Fig. 3, where input 'A' and input 'B' are placed at substate no. 1 of cell 3 (c3s1) and substate no. 2 of cell 2 (c2s2), respectively. Substate no. 3 of cell 1, i.e. the excitation state of cell 1 will hold the effect of a collision between two input signals according to rule (4, 27) [32] and decide the output ('Y'). To achieve an OR gate rule (3, 28) [33] is applied and introduced in Fig. 3. The collision between either three or four neighbouring signals leads to the resultant state of the output cell.

In this segment, the proposed design of 2-input to 2-output data line selector cum router is comprised of two dissimilar avenues.

5.2 Design of proposed 2-input to 2-output (2×2) *data line selector cum router*

In this segment, the proposed design of 2-input to 2-output data line selector cum router is comprised of two dissimilar avenues.

5.2.1 Method I (with NOT gate, AND gate and OR gate): This design, presented in Figs. 4*a* and *b*, the elementary one, has similarities with the traditional digital circuit design of mux and demux. In this method, initially, mux is implemented by a NOT, an AND and an OR gate to select the input data line 'I0' or 'I1'. A NOT gate associated with an AND gate set up demux which selects between the output data lines 'Y0' and 'Y1'.

5.2.2 Method II (optimised and reduced): This design is the optimised form of the design of method I. Two single automata are used to design mux and demux. The number of required cells and timesteps are minimised here. The design is demonstrated in Figs. 4*c* and *d*.

5.3 Design of proposed 4-input to 4-output (4×4) *data line selector cum router method*

Proposed 4-input to 4-output data line selector cum router is also constructed here in two unique ways.

5.3.1 Method I (using 2×2 *data line selector):* The basic design, as shown in Fig. 5*a*, is developed in a hierarchical approach, i.e. using three 2×2 data line selector. The detriment of this design is that multiple automata are introduced to reflect the output data line.

5.3.2 Method II (optimised and reduced): This method leads to realise the advantageous design with two single automata and the minimum number of cells. Figs. 5*b* and *c* represent the design of a 4×4 data line selector, where yellow-coloured automata is the 4×1 mux and green coloured stands for 1×4 demux.

5.4 Simulation technique

An approach to simulate the proposed design of the data line selector cum router is presented using Matlab 2016a. The proposed design, based on a state transition, is simulated through the 'Simulink' toolbox. 'Subsystem', a component of Simulink, is customised with five inputs and five outputs. Fig. 6*a* represents the subsystem which defines a cell with five substates of the first automata in any simulation. The columns of the subsystems are considered as timesteps. 'Switch' is configured in the subsystem to decide the excitation state of a cell, which depends on the rest four substates of the same cell [33]. Fig. 6*a* demonstrates the subsystems reflecting the cells of rest automata after the initial one. Fig. 6*b* shows the simulation result of the proposed design.

6 Device density

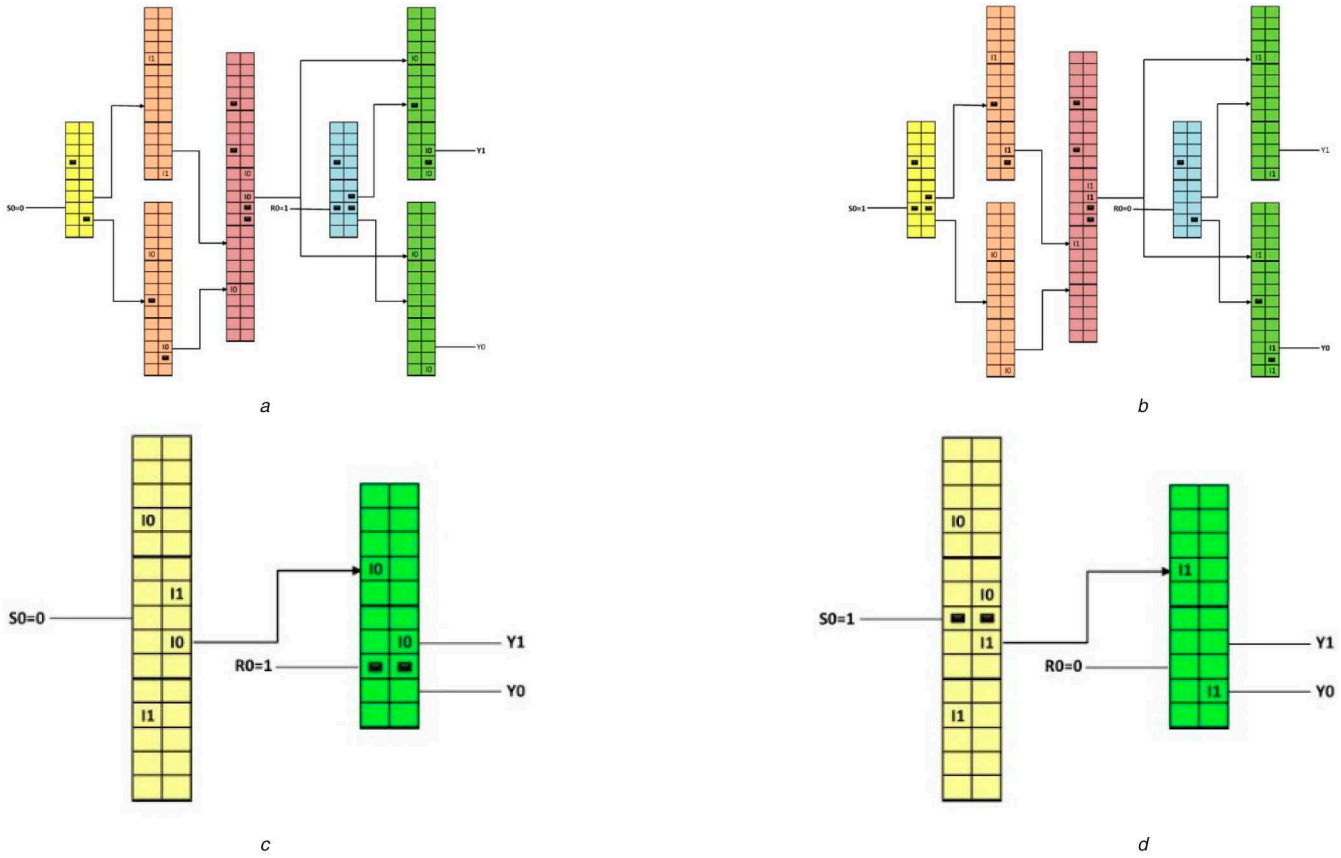
Near about 7 nm is the diameter of a helical actin filament and 5.4 nm [44] is the length or diameter of globular actin (G-actin) [10]. The proposed design of mux contains seven molecules or cells and the demux contains three in method II of 4×4 data line selector cum router.

Because of the helical structure of actin, the length of the mux and demux can be considered as 4×4 diameters of G-actin, i.e. 4×5.4 nm and 2×5.4 nm, respectively.

Now, the volume of the 4×4 data line selector cum router in method II can be calculated as

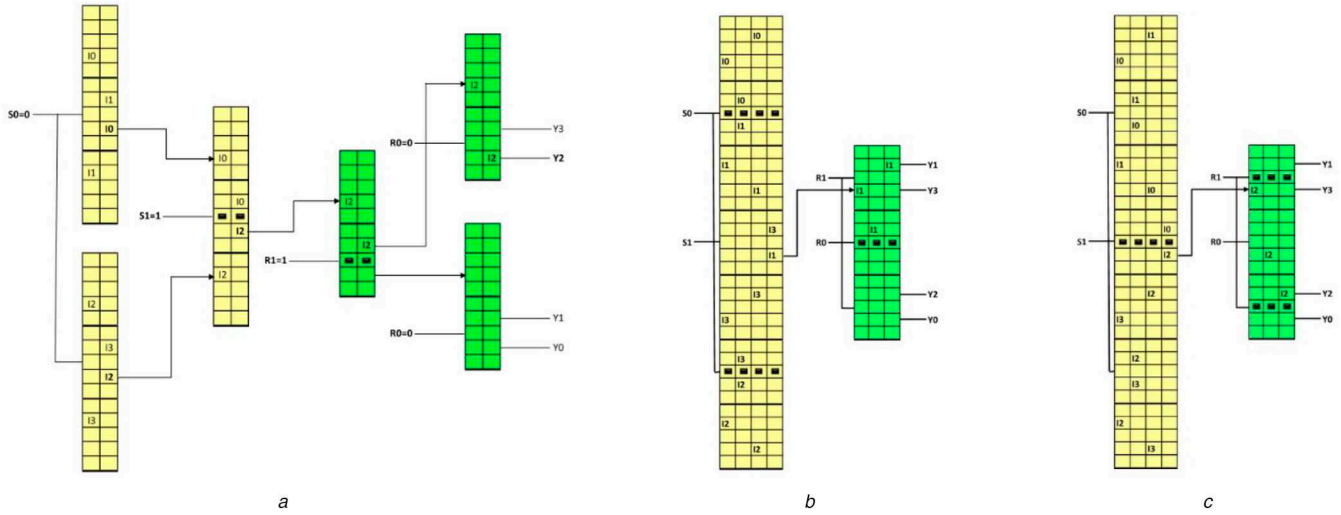
$$4 \times \pi \times (7/2)2 \times [(4+2) \times 5.4]\ \text{nm}^3 = 4987.59\,\text{nm}^3 \quad (1)$$

© The Institution of Engineering and Technology 2020



a b c d

Fig. 4 *Different design approach in method I and method II*
(a) When $S0 = 0$ for method I, ***(b)*** When $S0 = 1$ for method I, ***(c)*** When $S0 = 0$ for method II, ***(d)*** When $S0 = 1$ for method II



a b c

Fig. 5 *Different design approach 4 × 4 router in method I and method II*
(a) When $S0 = 0$ and $S1 = 1$ for method I, ***(b)*** When $S0 = 1$ and $S1 = 0$ for method II, ***(c)*** When $S0 = 0$ and $S1 = 1$ for method II

7 Power consumption

F-actin submerged in NaCl and KCl solution can act as an RLC circuit [10]. Direct current pulses are passed by the inductor is called short circuit and blocked by the capacitor is called an open circuit. The circuit is simulated in NI multisim which will hold only repellent element values figured in the following equation:

$$\boldsymbol{R}1 = 6.11\,\mathrm{M\Omega}, \quad \boldsymbol{R}2 = 0.9\,\mathrm{M\Omega} \tag{2}$$

The values are enumerated at 293 K atmospheric temperature and for 0.15 M K+ and 0.02 M Na+. Two units of G-actin were considered and simulated for calculating power consumption as output is obtained within one timestep and within that a pulse transfer from one G-actin, i.e. monomer to the next. In the simulation shown in Fig. 6*b*, the output voltage displayed by multimeter is 98.79 mV, whereas the input voltage is 100 mV. As inductor and capacitor do not effect in DC equivalent circuit, the voltage drop by the total equivalent resistance is

$$\delta V = V_o ut - V_i n = (100 - 98.79)\ \mathrm{mV} = 1.21\,\mathrm{mV} \tag{3}$$

Now, with the help of ohm's law, i.e. (4), the calculation of power consumption can be done

$$P = V^2/R \tag{4}$$

Thus, power consumption, calculated from (2) and (3), for each active substate is 119.8 fw. Now, for method II, 4-to-4 nano-router, at 0101 condition, the number of moves of input signals and select

© The Institution of Engineering and Technology 2020

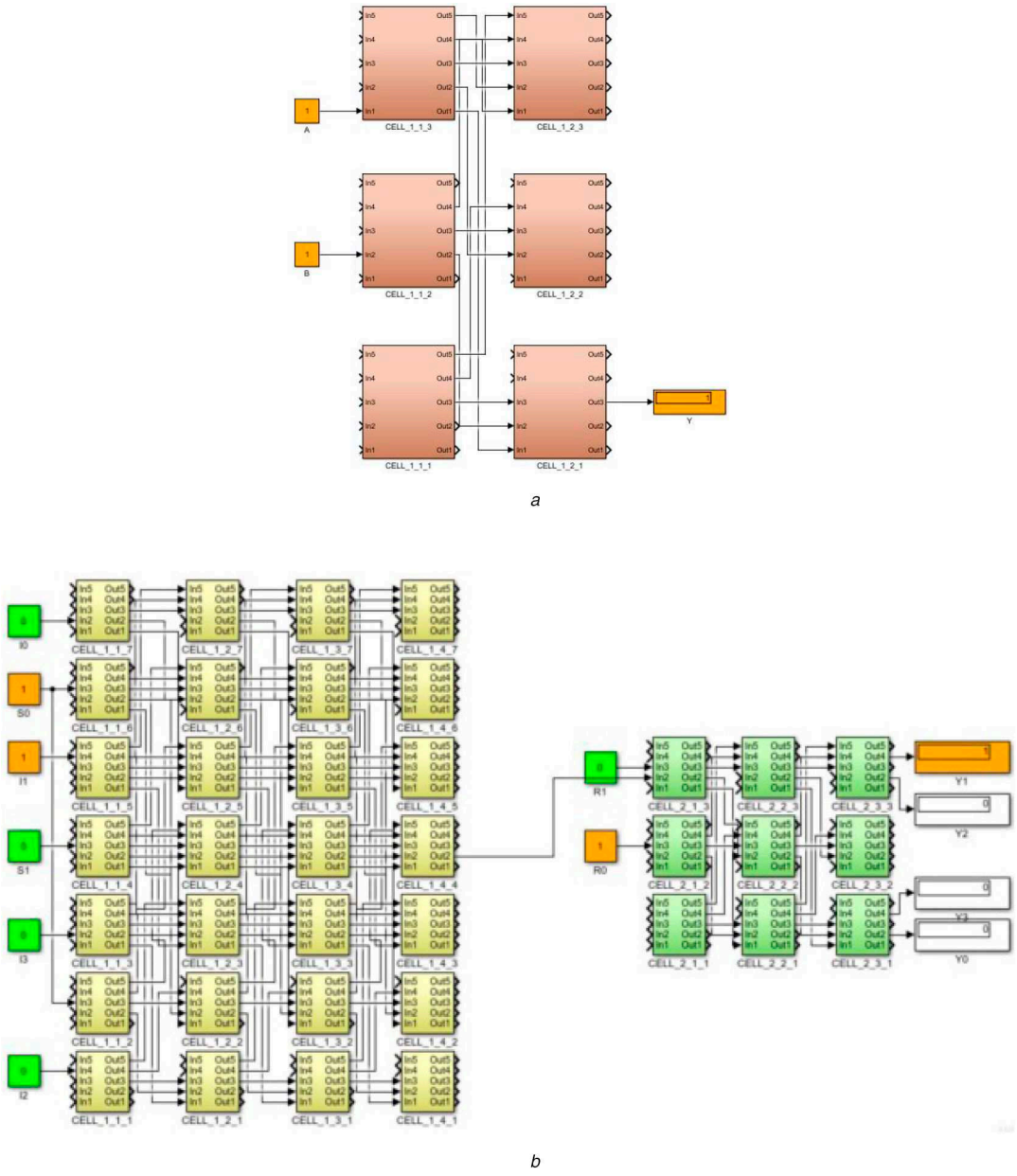


Fig. 6 *Simulation technique and simulation result of the proposed design*
(a) Subsystem which defines a cell with five Substates, ***(b)*** Simulation result of router when $S0 = 1$ and $S1 = 0$

Table 2 Missing cell-based fault analysis

Test vector (***A***, ***B***)	Expected output (***Y***)	Missing cell 1	Missing cell 2	Missing cell 3
(0, 0)	0	missing	0	0
(0, 1)	0	missing	0	0
(1, 0)	0	missing	0	0
(1, 1)	1	missing	0	0

line signals through substates. So, the highest power consumption at 0101 condition for this design will be 21×119.8 fw = 2515.8 fw.

8 Fault analysis

Due to short between signal paths or break-in signal path faults can occur in the CMOS logic circuit. The stuck-at fault is the most popular model of logical fault representation. Table 2 represents all possible stuck at fault in a two-input AND gate. Table 3 represents stuck at fault in CMOS AND gate design. In our proposed data line selector cum router design fault can arise due to missing of a cell or cells. Missing cell-based fault analysis for 2×2 router for Mux and Demux is shown in Tables 4 and 5, respectively, where ***I***1, ***I***0, ***S***0 and ***R***0 are the input vectors and ***Y***0 and ***Y***1 are the output vectors. For method II of 4×4 nano-router design, fault matrices are built and shown in Tables 6–8, respectively, where ***S***1, ***S***0, ***R***1 and ***R***0 are the input vectors and ***Y***0, ***Y***1, ***Y***2 and ***Y***3 are the output vectors. '***M***' denotes the missing vector in Tables 5 and 8, respectively. The analysis is done for single-cell missing in all cases.

9 Comparative analysis

The proposed data line selector cum router has been designed in two different methods, i.e. method I and method II. The method I is a combination of different logic gates like AND, NOT and OR gates and method II is an optimised and reduced method. The discrimination between these two methods is presented in Figs. 5*a* and *b* for a 4×4 data line selector cum router. A comparative analysis is shown in Fig. 7 for 4×4 data line selector cum router. This comparison is done with different cell count and time steps for

© The Institution of Engineering and Technology 2020

Table 3 Stuck at fault analysis in different input vectors

Test vector (***A***, ***B***)	Expected output (***Y***)	*a* stuck at 0	*a* stuck at 1	*b* stuck at 0	*b* stuck at 1
(0, 0)	0	0	0	0	0
(0, 1)	0	0	1	0	0
(1, 0)	0	0	0	0	1
(1, 1)	1	0	1	0	1

Table 4 Missing cell-based fault analysis for 2 × 2 router for MUX

Test vector (***I***1, ***I***0, ***S***0, ***R***0)	Expected output (***Y***0, ***Y***1)	Missing cell 1 (***Y***0, ***Y***1)	Missing cell 2 (***Y***0, ***Y***1)	Missing cell 3 (***Y***0, ***Y***1)
(0, 0, 0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
(0, 0, 0, 1)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
(0, 0, 1, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
(0, 0, 1, 1)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
(0, 1, 0, 0)	(1, 0)	(1, 0)	(0, 0)	(0, 0)
(0, 1, 0, 1)	(0, 1)	(0, 1)	(0, 0)	(0, 0)
(0, 1, 1, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
(0, 1, 1, 1)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
(1, 0, 0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
(1, 0, 0, 1)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
(1, 0, 1, 0)	(1, 0)	(0, 0)	(0, 0)	(1, 0)
(1, 0, 1, 1)	(0, 1)	(0, 0)	(0, 0)	(0, 1)
(1, 1, 0, 0)	(1, 0)	(1, 0)	(0, 0)	(0, 0)
(1, 1, 0, 1)	(0, 1)	(0, 1)	(0, 0)	(0, 0)
(1, 1, 1, 0)	(1, 0)	(0, 0)	(0, 0)	(1, 0)
(1, 1, 1, 1)	(0, 1)	(0, 0)	(0, 0)	(0, 1)

Table 5 Missing cell-based fault analysis for 2 × 2 router for DMUX

Test vector (***I***1, ***I***0, ***S***0, ***R***0)	Expected output (***Y***0, ***Y***1)	Missing cell 1 (***Y***0, ***Y***1)	Missing cell 2 (***Y***0, ***Y***1)
(0, 0, 0, 0)	(0, 0)	(M, M)	(0, 0)
(0, 0, 0, 1)	(0, 0)	(M, M)	(0, 0)
(0, 0, 1, 0)	(0, 0)	(M, M)	(0, 0)
(0, 0, 1, 1)	(0, 0)	(M, M)	(0, 0)
(0, 1, 0, 0)	(1, 0)	(M, M)	(0, 0)
(0, 1, 0, 1)	(0, 1)	(M, M)	(0, 0)
(0, 1, 1, 0)	(0, 0)	(M, M)	(0, 0)
(0, 1, 1, 1)	(0, 0)	(M, M)	(0, 0)
(1, 0, 0, 0)	(0, 0)	(M, M)	(0, 0)
(1, 0, 0, 1)	(0, 0)	(M, M)	(0, 0)
(1, 0, 1, 0)	(1, 0)	(M, M)	(0, 0)
(1, 0, 1, 1)	(0, 1)	(M, M)	(0, 0)
(1, 1, 0, 0)	(1, 0)	(M, M)	(0, 0)
(1, 1, 0, 1)	(0, 1)	(M, M)	(0, 0)
(1, 1, 1, 0)	(1, 0)	(M, M)	(0, 0)
(1, 1, 1, 1)	(0, 1)	(M, M)	(0, 0)

different methods. Fig. 7 proves that method II is an optimised and reduced method. Discussion on fault analysis of the proposed design has been done in Section 8. An analysis between CMOS AND gate and realised AND gate by actin cellular automata concerning all possible fault occurrences is given in Table 9 where the possible number of fault occurrences in CMOS-based design for an AND gate is 6 and actin-based design for an AND gate is 3.

10 Conclusion

The first method described in the present work is simple to understand and applicable to any complex circuit. The second method requires fewer time steps, but it still necessitates the need

Table 6 Missing cell-based fault analysis for 4 × 4 router for MUX

Test vector (***S***1, ***S***0, ***R***1, ***R***0)	Expected output (***Y***0, ***Y***1, ***Y***2, ***Y***3)	Missing cell 1	Missing cell 2	Missing cell 3
(0, 0, 0, 0)	(I0, 0, 0, 0)	(I0, 0, 0, 0)	(I0, 0, 0, 0)	(I0, 0, 0, 0)
(0, 0, 0, 1)	(0, I0, 0, 0)	(0, I0, 0, 0)	(0, I0, 0, 0)	(0, I0, 0, 0)
(0, 0, 1, 0)	(0, 0, I0, 0)	(0, 0, I0, 0)	(0, 0, I0, 0)	(0, 0, I0, 0)
(0, 0, 1, 1)	(0, 0, 0, I0)	(0, 0, 0, I0)	(0, 0, 0, I0)	(0, 0, 0, I0)
(0, 1, 0, 0)	(I1, 0, 0, 0)	(I1, 0, 0, 0)	(I1, 0, 0, 0)	(I1, 0, 0, 0)
(0, 1, 0, 1)	(0, I1, 0, 0)	(0, I1, 0, 0)	(0, I1, 0, 0)	(0, I1, 0, 0)
(0, 1, 1, 0)	(0, 0, I1, 0)	(0, 0, I1, 0)	(0, 0, I1, 0)	(0, 0, I1, 0)
(0, 1, 1, 1)	(0, 0, 0, I1)	(0, 0, 0, I1)	(0, 0, 0, I1)	(0, 0, 0, I1)
(1, 0, 0, 0)	(I2, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(1, 0, 0, 1)	(0, 12, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(1, 0, 1, 0)	(0, 0, 12, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(1, 0, 1, 1)	(0, 0, 0, I2)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(1, 1, 0, 0)	(I3, 0, 0, 0)	(I3, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(1, 1, 0, 1)	(0, I3, 0, 0)	(0, I3, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(1, 1, 1, 0)	(0, 0, I3, 0)	(0, 0, I3, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(1, 1, 1, 1)	(0, 0, 0, I3)	(0, 0, 0, I3)	(0, 0, 0, 0)	(0, 0, 0, 0)

Table 7 Missing cell-based fault analysis for 4 × 4 router for MUX

Test vector (***S***1, ***S***0, ***R***1, ***R***0)	Expected output (***Y***0, ***Y***1, ***Y***2, ***Y***3)	Missing cell 4	Missing cell 5	Missing cell 6
(0, 0, 0, 0)	(I0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(0, 0, 0, 1)	(0, I0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(0, 0, 1, 0)	(0, 0, I0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(0, 0, 1, 1)	(0, 0, 0, I0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(0, 1, 0, 0)	(I1, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(I1, 0, 0, 0)
(0, 1, 0, 1)	(0, I1, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, I1, 0, 0)
(0, 1, 1, 0)	(0, 0, I1, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, I1, 0)
(0, 1, 1, 1)	(0, 0, 0, I1)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, I1)
(1, 0, 0, 0)	(I2, 0, 0, 0)	(I2, 0, 0, 0)	(I2, 0, 0, 0)	(I2, 0, 0, 0)
(1, 0, 0, 1)	(0, 12, 0, 0)	(0, I2, 0, 0)	(0, I2, 0, 0)	(0, I2, 0, 0)
(1, 0, 1, 0)	(0, 0, 12, 0)	(0, 0, I2, 0)	(0, 0, I2, 0)	(0, 0, I2, 0)
(1, 0, 1, 1)	(0, 0, 0, I2)	(0, 0, 0, I2)	(0, 0, 0, I2)	(0, 0, 0, I2)
(1, 1, 0, 0)	(I3, 0, 0, 0)	(I3, 0, 0, 0)	(I3, 0, 0, 0)	(I3, 0, 0, 0)
(1, 1, 0, 1)	(0, I3, 0, 0)	(0, I3, 0, 0)	(0, I3, 0, 0)	(0, I3, 0, 0)
(1, 1, 1, 0)	(0, 0, I3, 0)	(0, 0, I3, 0)	(0, 0, I3, 0)	(0, 0, I3, 0)
(1, 1, 1, 1)	(0, 0, 0, I3)	(0, 0, 0, I3)	(0, 0, 0, I3)	(0, 0, 0, I3)

to synchronise two separate automata dedicated to specific tasks. All the discussed methods show how actin filaments are useful to realise logic circuits at the molecular level. From these implications, the second method can be considered to be the most suitable method in designing a multiplexer. More efficient design and analysis of the Boolean function of multiplexer exposed in this technology. From the proposed design and comparative analysis, the first method is chosen as a straightforward method but need to synchronise multiple automata. Method II design is ideal for both 2×2 and 4×4 nano-router as having a minimum number of cells and automata.

© The Institution of Engineering and Technology 2020

Table 8 Missing cell-based fault analysis for 4 × 4 router for DMUX

Test vector (***S***1, ***S***0, ***R***1, ***R***0)	Expected output (***Y***0, ***Y***1, *Y*2, ***Y***3)	Missing cell 1	Missing cell 2	Missing cell 3
(0, 0, 0, 0)	(I0, 0, 0, 0)	(M, 0, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(0, 0, 0, 1)	(0, I0, 0, 0)	(M, I0, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(0, 0, 1, 0)	(0, 0, I0, 0)	(M, 0, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(0, 0, 1, 1)	(0, 0, 0, I0)	(M, 0, M, I0)	(0, 0, 0, 0)	(0, M, 0, M)
(0, 1, 0, 0)	(I1, 0, 0, 0)	(M, 0, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(0, 1, 0, 1)	(0, I1, 0, 0)	(M, I1, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(0, 1, 1, 0)	(0, 0, I1, 0)	(M, 0, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(0, 1, 1, 1)	(0, 0, 0, I1)	(M, 0, M, I1)	(0, 0, 0, 0)	(0, M, 0, M)
(1, 0, 0, 0)	(I2, 0, 0, 0)	(M, 0, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(1, 0, 0, 1)	(0, 12, 0, 0)	(M, I2, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(1, 0, 1, 0)	(0, 0, 12, 0)	(M, 0, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(1, 0, 1, 1)	(0, 0, 0, I2)	(M, 0, M, I2)	(0, 0, 0, 0)	(0, M, 0, M)
(1, 1, 0, 0)	(I3, 0, 0, 0)	(M, 0, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(1, 1, 0, 1)	(0, I3, 0, 0)	(M, I3, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(1, 1, 1, 0)	(0, 0, I3, 0)	(M, 0, M, 0)	(0, 0, 0, 0)	(0, M, 0, M)
(1, 1, 1, 1)	(0, 0, 0, I3)	(M, 0, M, I3)	(0, 0, 0, 0)	(0, M, 0, M)

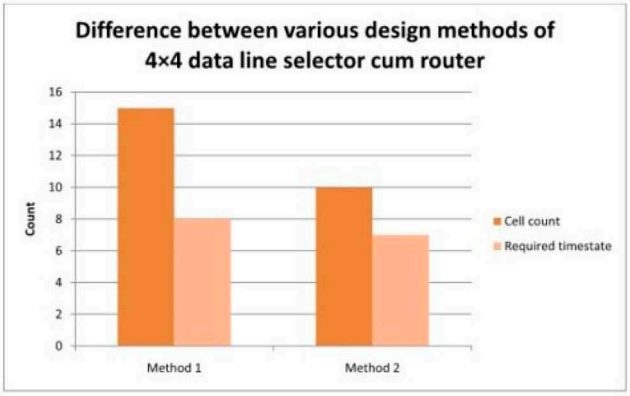


Fig. 7 *Comparison between different methods*

Table 9 Comparison with CMOS-based design and actin-based design

Methods	Possible number of fault occurrences in an AND gate
conventional CMOS design	6
actin-based design	3

11 References

[1] Siccardi, S., Adamatzky, A.: 'Actin quantum automata: communication and computation in molecular networks', *Nano. Commun. Netw.*, 2015, **6**, (1), pp. 15–27

[2] Adamatzky, A.: '*Collision-based computing*' (Springer, London, 2002)

[3] Tuszyński, J.A., Portet, S., Dixon, J.M., *et al.*: 'Ionic wave propagation along actin filaments', *Biophys. J.*, 2004, **86**, (4), pp. 1890–1903

[4] Lahoz Beltra, R., Hameroff, S.R., Dayhoff, J.E.: 'Cytoskeletal logic: a model for molecular computation via Boolean operations in microtubules and microtubule-associated proteins', *Biosystems*, 1993, **29**, (1), pp. 1–23

[5] Schwab, A., Fabian, A., Hanley, P.J., *et al.*: 'Role of ion channels and transporters in cell migration', *Physiol. Rev.*, 2012, **92**, (4), pp. 1865–1913

[6] Cantiello, H.F.: 'Role of actin filament organization in cell volume and ion channel regulation', *J. Exp. Zool.*, 1997, **279**, (5), pp. 425–435

[7] Lin, E.C., Cantiello, H.F.: 'A novel method to study the electrodynamic behavior of actin filaments. Evidence for cable-like properties of actin', *Biophys. J.*, 1993, **65**, (4), pp. 1371–1378

[8] De Loof, A., De Haes, W., Boerjan, B., *et al.*: 'The fading electricity theory of ageing: the missing biophysical principle?', *Ageing Res. Rev.*, 2013, **12**, (1), pp. 58–66

[9] Plankar, M., Brežan, S., Jerman, I.: 'The principle of coherence in multi-level brain information processing', *Prog. Biophys. Mol. Biol.*, 2013, **111**, (1), pp. 8–29

[10] Sadhu, T., Das, B., Ghosh, T., *et al.*: 'XOR transform with actin quantum automata for better security against cryptanalysis', *Microsyst. Technol.*, 2019, **25**, (5), pp. 1705–1717

[11] Capstick, M.H., Marnane, W.P.L., Pethig, R.: 'Biologic computational building blocks', *Computer. (Long. Beach. Calif)*, 1992, **25**, (11), pp. 22–29

[12] Mayne, R., Adamatzky, A., Jones, J.: 'On the role of the plasmodial cytoskeleton in facilitating intelligent behaviour in slime mould *Physarum polycephalum*', arXiv preprint arXiv:150303012, 2015

[13] Das, J.C., De, D.: 'Reversible comparator design using quantum dot-cellular automata', *IETE J. Res.*, 2016, **62**, (3), pp. 323–330

[14] Das, J.C., Debnath, B., De, D.: 'Image steganography using quantum dot-cellular automata', *Quantum Matter*, 2015, **4**, (5), pp. 504–517

[15] Das, J.C., De, D.: 'Reversible binary to grey and grey to binary code converter using QCA', *IETE J. Res.*, 2015, **61**, (3), pp. 223–229

[16] Das, J. C., De, D.: 'User authentication based on quantum-dot cellular automata using reversible logic for secure nanocommunication', *Arab. J. Sci. Eng.*, 2016, **41**, (3), pp. 773–784

[17] Das, K., De, D.: 'Characterization, test and logic synthesis of novel conservative and reversible logic gates for QCA', *Int. J. Nanosci.*, 2010, **9**, (3), pp. 201–214

[18] Das, K., De, D.: 'A novel approach of and-or-inverter (AOI) gate design for QCA'. 4th Int. Conf. on Computers and Devices for Communication, 2009, CODEC 2009, Kolkata, India, 2009, pp. 1–4

[19] Graziano, M., Pulimeno, A., Wang, R., *et al.*: 'Process variability and electrostatic analysis of molecular qca', *ACM J. Emerg. Technol. Comput. Syst. (JETC)*, 2015, **12**, (2), p. 18

[20] Kumar, A., Ghosh, B., Salimath, A.: 'Optimum sizing of magnetic cell in magnetic quantum dot cellular automata', *Quantum Matter*, 2015, **4**, (6), pp. 570–576

[21] Bhattacharjee, P., Das, K., De, M., *et al.*: 'Spice modeling and analysis for metal island ternary qca logic device', In: '*Information systems design and intelligent applications*' (Springer, India, 2015). pp. 33–41

[22] Tsukerblat, B., Palii, A., Clemente Juan, J.M.: 'Self-trapping of charge polarized states in four-dot molecular quantum cellular automata: bi-electronic tetrameric mixed-valence species', *Pure Appl. Chem.*, 2015, **87**, (3), pp. 217–282

[23] Adamatzky, A.: 'Advances in physarum machines gates, hulls, mazes and routing with slime mould.'. PARCO, Ghent, Belgium, 2011, pp. 41–54

[24] Partner, H.L., Nigmatullin, R., Burgermeister, T., *et al.*: 'Structural phase transitions and topological defects in ion Coulomb crystals', *Phys. B Condensed Matter*, 2015, **460**, pp. 114–118

[25] Mayne, R., Adamatzky, A.: 'On the computing potential of intracellular vesicles', *PLOS ONE*, 2015, **10**, (10), p. e0139617

[26] Adamatzky, A., Schnaub, J., Huber, F.: 'Actin droplet machine', *R. Soc. Open Sci.*, 2019, **6**, (12), p. 191135

[27] Siccardi, S., Adamatzky, A., Tuszyński, J., *et al.*: 'Actin networks voltage circuits', *Phys. Rev. E*, 2020, **101**, (5), p. 052314

[28] Mayne, R., Adamatkzy, A.: 'Cellular automata modelling of slime mould actin network signalling', *Nat. Comput.*, 2019, **18**, (1), pp. 5–12

[29] Adamatzky, A.: 'On discovering functions in actin filament automata', *R. Soc. Open Sci.*, 2019, **6**, (1), p. 181198

[30] Schumann, A.: '*Behaviourism in studying swarms: logical models of sensing and motoring*' (Springer, Switzerland, 2019)

[31] Ardesi, Y., Pulimeno, A., Graziano, M., *et al.*: 'Effectiveness of molecules for quantum cellular automata as computing devices', *J. Low Power Electron. Appl.*, 2018, **8**, (3), p. 24

[32] Das, B., Paul, A.K., De, D.: 'An unconventional arithmetic logic unit design and computing in actin quantum cellular automata', *Microsyst. Technol.*, 2019, pp. 1–14, doi: 10.1007/s00542-019-04590-1

[33] Sadhu, T., Das, B., De, D., *et al.*: 'Design of binary subtractor using actin quantum cellular automata', *IET Nanobiotechnol.*, 2017, **12**, (1), pp. 32–39

[34] Lloyd, S., Braunstein, S.L.: 'Quantum computation over continuous variables', *Phys. Rev. Lett.*, 1999, **82**, (8), p. 1784

[35] Mayne, R., Adamatzky, A.: 'The *Physarum polycephalum* actin network: formalisation, topology and morphological correlates with computational ability'. Proc. of the 8th Int. Conf. on Bioinspired Information and Communications Technologies (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Boston, MA, USA, 2014, pp. 87–94

[36] Siccardi, S., Tuszynski, J.A., Adamatzky, A.: 'Boolean gates on actin filaments', *Phys. Lett. A*, 2016, **380**, (1–2), pp. 88–97

[37] Adamatzky, A., Mayne, R.: 'Actin automata: phenomenology and localizations', *Int. J. Bifurc. Chaos*, 2015, **25**, (2), p. 1550030

[38] Schuh, M.: 'An actin-dependent mechanism for long-range vesicle transport', *Nat. Cell Biol.*, 2011, **13**, (12), pp. 1431–1436

[39] Inokuchi, S., Mizoguchi, Y.: 'Generalized partitioned quantum cellular automata and quantization of classical ca', arXiv preprint quant-ph/0312102, 2003

[40] Mayne, R., Adamatzky, A., Jones, J.: 'On the role of the plasmodial cytoskeleton in facilitating intelligent behavior in slime mold *Physarum polycephalum*', *Commun. Integrative Biol.*, 2015, **8**, (4), p. e1059007

[41] De, D., Sadhu, T., Das, J.C.: 'Bioprocess modeling and simulation of half subtractor using actin based quantum cellular automata', *Mater. Today: Proc.*, 2016, **3**, (10), pp. 3276–3284

[42] Bernstein, F.C., Koetzle, T.F., Williams, G.J., *et al.*: 'The protein data bank: a computer-based archival file for macromolecular structures', *Arch. Biochem. Biophys.*, 1978, **xtbf185**, (2), pp. 584–591

[43] De, D., Ghatak, K.P.: '*Basic electronics*' (Pearson Education, India, 2003)

[44] Povarova, O.I., Sulatskaya, A.I., Kuznetsova, I.M., *et al.*: '*Actin folding, structure and function: is it a globular or an intrinsically disordered protein?*' (INTECH Open Access Publisher, Croatia, 2012)

© The Institution of Engineering and Technology 2020