Novel optimized tree-based stack-type architecture for 2*n*-bit comparator at nanoscale with energy dissipation analysis

A. Arunkumar Gudivada[1] · Gnanou Florence Sudha[1]

Abstract

Comparator is an essential building block in many digital circuits such as biometric authentication, data sorting, and exponents comparison in floating-point architectures among others. Quantum-dot Cellular Automata (QCA) is a latest nanotechnology that overcomes the drawbacks of Complementary Metal Oxide Semiconductor (CMOS) technology. In this paper, novel area optimized 2*n*-bit comparator architecture is proposed. To achieve the objective, 1-bit stack-type and 4-bit tree-based stack-type (TB-ST) comparators are proposed using QCA. Then, two tree-based architectures of 4-bit comparators are arranged in two layers to optimize the number of quantum cells and area of an 8-bit comparator. Thus, this design can be extended to any 2*n*-bit comparator. Simulation results of 4-bit and 8-bit comparators using QCADesigner 2.0.3 show that there is a significant improvement in the number of quantum cells and area occupancy. The proposed TB-ST 8-bit comparator uses 2.5 clock cycles and 622 quantum cells with area occupancy of 0.49 μm^2 which is an improvement by 10.5% and 38%, respectively, compared to existing designs. Scaling it to a 32-bit comparator, the proposed architecture requires only 2675 quantum cells in an area of 2.05 μm^2 with a delay of 3.5 clock cycles, indicating 9.35% and 28.8% improvements, respectively, demonstrating the merit of the proposed architecture. Besides, energy dissipation analysis of the proposed TB-ST 8-bit comparator is simulated on QCADesigner-E tool, indicating average energy dissipation reduction of 17.3% compared to existing works.

Keywords Quantum-dot Cellular Automata (QCA) · Comparator · Tree-based stack-type architecture · Energy dissipation analysis

---

✉ A. Arunkumar Gudivada
aarunkumarg@gmail.com

Gnanou Florence Sudha
gfsudha@pec.edu

1   Electronics and Communication Engineering, Pondicherry Engineering College, Puducherry 605014, India

1 Introduction

Demand for devices with power consumption less than a few milliwatts is on the rise due to developments in related technologies. Due to problems in channel length and threshold leakage current at nanoscale, the Complementary Metal Oxide Semiconductor (CMOS) technology may not work efficiently [1]. According to Moore's law, the number of transistors should be doubled for every two years. However, because of the limitation of Complementary Metal Oxide Semiconductor technology at nanodimensions, International Technology Roadmap for Semiconductor (ITRS) is on the lookout for a better low power technology to cater to the consumer's ever-demanding needs. Quantum-dot Cellular Automata (QCA) is a high-speed novel nanotechnology that works at terahertz speed and is based on the principle of columbic interaction of electrons in quantum cells.

While performing signal processing applications, like Fourier transforms and fast Fourier transforms, operations on fixed points can be easily and effectively executed. However, operations on floating-point numbers play a crucial role in such applications. For these types of computations, basic CMOS implementations are not well suited. To perform addition, subtraction, multiplication, or division on floating-point numbers at fast rates, low power architectures are preferred. In such operations, the comparator plays an important role. Such a comparator with very low power at nanodimensions is designed in this paper using Quantum-dot Cellular Automata.

Quantum-dot Cellular Automata is an emerging nanotechnology used to design circuits at nanoscales [2]. The process of transferring the data among the cells is known as Cellular Automata which was discovered by Von Neumann. Similarly, Quantum wells can be created using a simple nanostructure of silicon to store the electron. Thus, Quantum-dot Cellular Automata is a mix of these two technologies. There are four different types of fabrication methods available for QCA. They are metal QCA, semiconductor QCA, molecular QCA, and magnetic QCA. Among these, metal QCA is commonly used but works best at low temperatures. However, molecular QCA works appropriately, even at room temperature [3]. In order to fabricate nanostructured molecular QCA, several studies have been done. The nanostructured molecular QCA fabricated using Fe and Ru in [4] showed better capacitance property for storage of information. Besides, several studies have been done on nanostructures [5–7], which also give better characteristics on data storage capacity about 2899 mAh/g through electrochemical process and hydrogen energy.

A quantum cell is a square-shaped cell consisting of four dots at its corners and two electrons. It has a dimension of $18 \text{ nm} \times 18 \text{ nm}$ with each dot of 5 nm diameter. The cells are spaced 20 nm apart. Information is passed from one cell to another by electrostatic interactions. Initially, the two electrons are in the rest position. When polarization of -1 or $+1$ is applied, the electrons are positioned in the dots to represent either logic '0' or logic '1' as shown in Fig. 1.

In CMOS technology, along with transistors, wires are also used. However, in QCA, the quantum cell itself acts as wires. There are two different types of wiring

Fig. 1 Basic quantum cell: **a** rest condition, **b** logic0 state, **c** logic1 state [1]

(a) **(b)** **(c)**
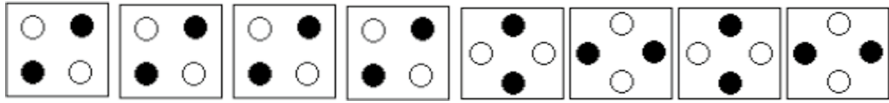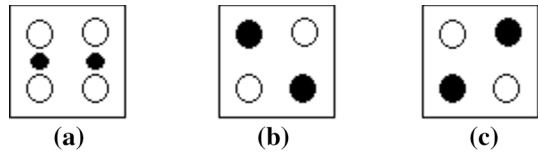
Fig. 2 90° wire (left) and 45° wire (right) [2]

Input-2

Input-1 Output

Input-3

Input

Output
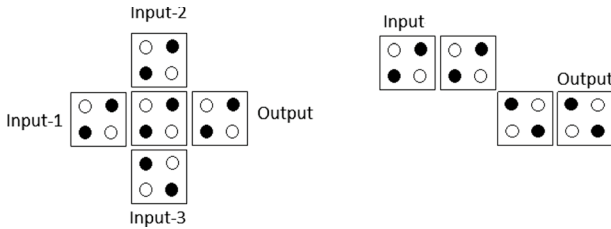
Fig. 3 QCA gates: **a** MVG and **b** INV gate [2]

techniques available in QCA. They are 90° wire and 45° wire, as shown in Fig. 2. 90° wire is used to pass the information in true form, whereas 45° wire is used to pass in complemented form. Mostly, 45° wires are used in wire crossings.

Two major gates used in QCA are majority voter gate (MVG) and inverter (INV) gate. The MVG indicated in Fig. 3a has five cells in which the center cell is known as a device cell. The device cell will experience the data that has the majority of the inputs. The INV gate, as shown in Fig. 3b, consists of four quantum cells in series which complement the input due to the repulsion of the electrons.

2 Clocking scheme in QCA

In QCA, clocking determines the direction of the data to be propagated. There are four phases in the clocking system of QCA. They are switch, hold, release, and relax. During the switching period, the cell tries to acquire the data available in the neighboring cell. At hold, the cell completely receives the data of the neighboring cell. In the release phase, the cell tries to lose its data and transfers it to the next succeeding cell. In the last phase, that is, relax phase, the cell is completely free from data (Fig. 4).

The color of the cell identifies the application of the clock on the cell. Green-colored cell indicates the clock 0 phase, the pink-colored cell means the clock 1 phase, the blue-colored cell indicates the clock 2 phase, and the white-colored cell indicates the clock 3 phase. No two different clocked cells will be in the same state.
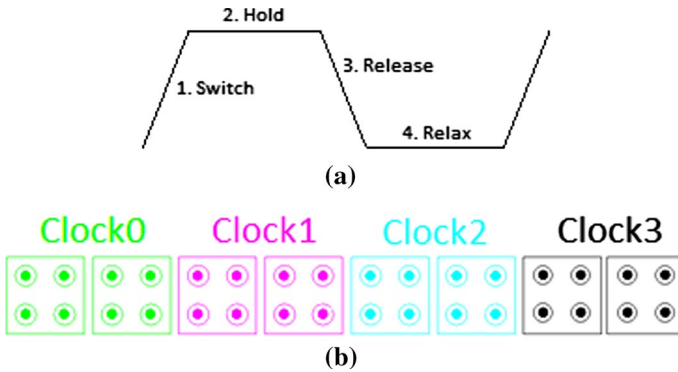
Fig. 4 Clocking in QCA: **a** clock phases, **b** color representation of clocks [2]
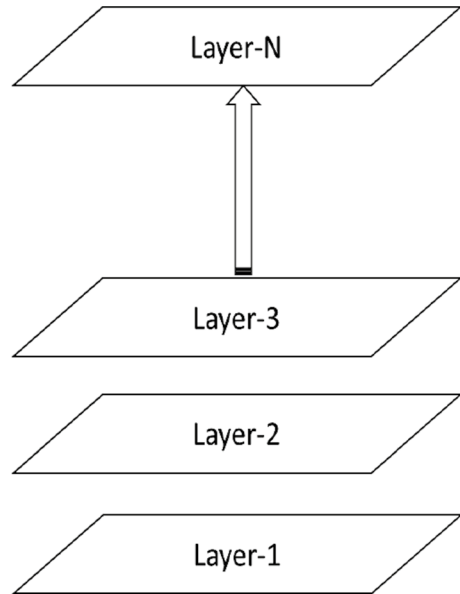
3 QCA architectures

In Cellular Automata, data in one cell can be transmitted to neighboring cells. As QCA is a combination of quantum-dot and Cellular Automata, it exhibits the properties of both quantum technology and Cellular Automata. Hence, the data in a quantum cell are transmitted to all the neighboring cells within its radius. A neighboring cell may be an adjacent cell or a diagonal cell or a vertical cell. A vertical cell is a cell that is present in an upper or lower layer. In QCA design, layouts can be designed not only in one layer but also in multiple layers. When two layers are constructed and, if a cell in layer-1 comes exactly below the cell in layer-2, the data can be propagated in a vertical direction. As minimizing area is paramount in QCA designs, the multilayer approach is preferred. In earlier works, the multilayer approach was utilized for wire crossing, but now QCA designs are constructed in multiple layers.

As the multilayer approach promises a revolutionary improvement in area minimization, this paper deals with comparators design in multilayers. In this paper, the multilayer structure is called a stack-type approach. As shown in Fig. 5, in the stack-type approach, the entire logical circuit is divided into three or smaller circuits depending on a particular logic function and these are constructed in different layers which are interconnected vertically.

Another architecture used by many authors is referred to as tree-based approach. Tree-based architecture involves constructing the QCA in a hierarchical manner. This type of architecture includes any logical circuit using large number of majority voter gates in a tree structure.

Among the many digital circuit implementations, the comparator plays a very crucial role in several applications. Generally, a comparator needs two inputs of n-bits each. In this paper, novel 1-bit stack-type (ST) comparator and 4-bit and 8-bit tree-based stack-type (TB-ST) comparator designs are proposed using QCA. In [8, 9], the logical circuit of the 1-bit comparator is partitioned into two sub-circuits, and each circuit is designed in distinct layers. The authors show that stack type or multi-layer is the better approach in minimizing the QCA area; however, both the designs

do not provide an optimized QCA layout. In [10], the tree-based approach was used for the design of higher-order comparators.

The objective of this paper is to propose a novel area optimized 2n-bit comparator architecture with lower energy dissipation. In this work, the tree-based method of [10] is modified by replacing the majority voter gates with EXOR gates to minimize the number of cells. The stack-type approach is combined with the modified tree-based architecture model to decrease the quantum cells used and area further. The inputs are applied to the first stage consisting of majority voter gates connected in a tree-based manner. As the data move from stage to stage, the number of majority voter gates gets reduced. Finally, the three output functions for 'less than,' 'greater than,' and 'equal to' conditions are obtained.

This paper is further structured as follows: Sect. 4 reviews the previous works of comparators designed using QCA. Section 5 explains the design of the novel ST 1-bit comparator and its simulation results. Section 6 discusses the proposed TB-ST 4-bit and 8-bit architectures and layouts. Section 7 describes the simulation results of the 8-bit comparator. Energy dissipation analysis is studied in Sect. 8, and the conclusion of the proposed designs is described in Sect. 9.

4 Related works on QCA comparators

Comparator is a digital circuit which is used to compare two input binary numbers A, B and produce any of the three outputs: A greater than B (Abig$B$), A equal to B (Aequal$B$), and A less than B (Bbig$A$), as shown in Fig. 6.

The output logic functions for the 1-bit comparator are given by

Fig. 6 Basic comparator
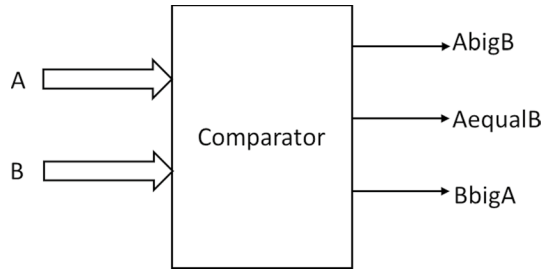
$$AbigB = A\overline{B} \tag{1}$$

$$AequalB = \overline{(A \oplus B)} \tag{2}$$

$$BbigA = \overline{A}B \tag{3}$$

Several designs for 1-bit binary comparators have been proposed using QCA from [8] to [27]. All the designs have aimed at reducing the quantum cell count, area occupied, and delay. In [8], Hayati et al. proposed a 1-bit comparator in a multilayered approach with a cell count of 79 in an area of 0.038 μm^2 and having a latency of one clock cycle. In [9], Roohi et al. proposed a multilayered 1-bit comparator with 54 cells in 0.04 μm^2 area and one clock cycle delay. In [10], Babu et al. proposed a C-Gate for comparison function for five bits. The authors implemented the tree-based structure consisting of all majority voter gates. In [11], Blaz Lampreht et al. proposed a 1-bit comparator with 221 quantum cells in 0.07 μm^2 area with nine clock cycles delay. In [12], Meghanad et al. reduced the number of clock zones, so that delay proportionately reduced to 2 clock cycles, and designed a tree-based 1-bit comparator with 135 quantum cells in an area of 0.2 μm^2. Later in [13], Abdullah-Al Shafi et al. reduced the quantum cells further to 117 with occupancy of 0.171 μm^2. The delay was also reduced to one clock cycle. Qiu Ke-ming et al. in [14] designed a 1-bit binary comparator with 100 cells in an area of 0.127 μm^2 with the same delay [13]. Further, Jun-wen et al. [15] suggested a 1-bit comparator with the same number of cells in an area of 0.11 μm^2 with a reduced delay of 0.75 clock cycles. In [16], Xia et al. proposed 1-bit, 4-bit, and 8-bit comparators with a cell count of 97, 722, and 1620, respectively. The area (μm^2)–delay pair was in the order of 0.032–1, 1.63–3.5, and 4.3–4.5, respectively. The authors came up with a new design model called a tree-based architecture, which has *n* input lines to three output lines. Though this approach reduced the area of the layout, the entire design consisted of only majority voter gates.

Furthermore, in [17], Ajitha et al. proposed a single-bit binary comparator with 81 cells in an area of 0.05 μm^2 with one clock cycle latency. Subsequently, in [18] and [19], the authors designed 1-bit binary comparators with a consistent delay of one clock cycle using 73 and 67 quantum cells and area, in the order of 0.064 μm^2 and 0.08 μm^2, respectively. In [20], Umira et al. proposed a 1-bit comparator with 58

cells in 0.07 μm^2 area, having a delay of nine clock cycles. Later in [21], Bahar et al. suggested a 1-bit comparator using 47 cells in an area of 0.042 μm^2 with a great reduction in delay of 0.5 clock cycles. In [22], Behrepour et al. and in [23] Roy et al. introduced a novel approach for one-bit comparators with cell counts of 43, 40, and 37 in an area of 0.071 μm^2, 0.032 μm^2, and 0.028 μm^2, respectively, with the delay of one clock cycle.

Shiri et al. [24] proposed a 1-bit comparator with 38 cells in 0.03 μm^2 area with a delay of 0.5 clock cycles. In [25], Das et al. proposed a 1-bit comparator with 37 cells in 0.023 μm^2 area with a latency of 0.75 clock cycles. Further, in [26], Marshal Raj et al. designed an optimized 1-bit binary comparator with only 34 quantum cells within an area of 0.02 μm^2 with a latency of one clock cycle. Later in [27], Stefania Perri et al. recommended the tree-based architecture and cascaded architectures. Here, the authors proposed tree-based QCA architectures for 16-bit and 32-bit comparators with 35 and 83 majority voter gates.

5 Proposed stack-type 1-bit binary comparator

In this paper, an optimized stack-type (ST) architecture is introduced to construct a 1-bit comparator. In this novel stack-type approach, layouts are created in layers keeping the radius of the cell unchanged. The basic concept of the proposed 1-bit comparator design is shown in Fig. 7a. The XNOR operation for the equality function and one AND operation for less than function are set in the main layer, i.e., layer-1, whereas the AND gate for the greater than operation is set in the top layer-3. Figure 7b shows the complete architecture of the proposed 1-bit stack-type comparator.

Figure 8a depicts the formation of the three layers in a three-dimensional view, and Fig. 8b shows the three individual layers of the proposed design. The inputs are commonly applied to the main layer, which is transferred to the top layer via the middle layer-2. The inputs are shown in blue color with vertical cell types as they propagate data in the vertical direction.

For the proposed comparator design, the three-input XOR gate given by Newaz [28] was used along with an INV gate to obtain the XNOR function. As this XOR gate requires only ten quantum cells, this gate was used in this paper. For the AND and NOT operation, the MVG gate and INV gates were employed for the design. Thus, using this optimized stack-type design model, the area can be significantly reduced for higher-order comparator designs.

5.1 Simulation results of proposed stack-type 1-bit comparator

QCADesigner 2.0.3 was used for the simulation of the proposed ST 1-bit comparator. The simulation results are shown in Fig. 9 for three cases. For the first case, the input vectors are taken as $A = 1$ and $B = 0$. For this, the output, $A < B$, is 1, while the other outputs $A > B$ and $A = B$ are 0 s. Similar results are observed for the input vectors $A = 0$ and $B = 1$; then, the result $A > B$ is 1, and for $A = B = 0$ or

Fig. 7 1-bit ST comparator: **a** logic circuit, **b** proposed layout


(a)

(b)
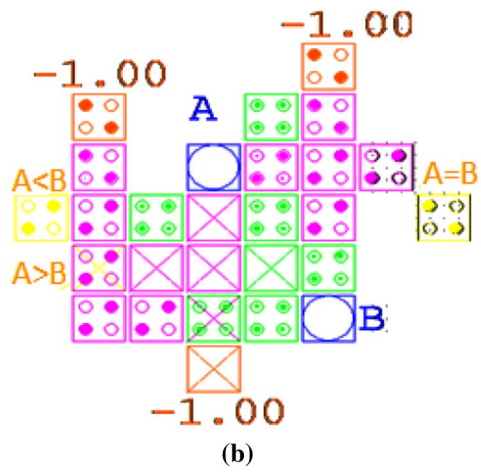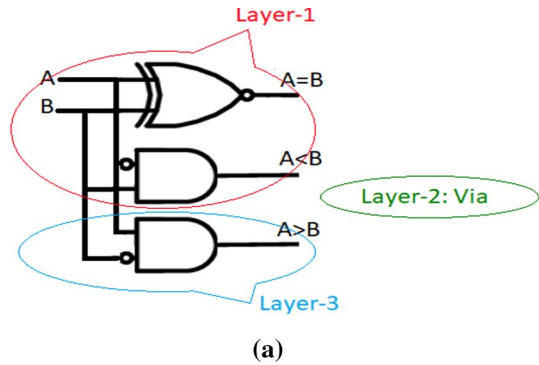
1, the output $A = B$ asserts one while the other outputs are 0 s. As shown in Fig. 9, the output occurs after 0.5 clock period indicated by the red-colored arrow. This clearly demonstrates that the proposed design functions as a 1-bit comparator.

Table 1 compares different parameters like quantum cells used, area, latency, and cost for the proposed and existing 1-bit comparator designs. The cost of the layout is determined by the product of area and latency as given in [29]:

$$\text{Cost} = \text{Area}\left(\mu m^2\right) \times \text{Latency(number of clock cycles)} \tag{4}$$

The percentage improvement of the proposed design is also given in Table 1. The proposed 1-bit comparator is designed with only 33 quantum cells, in an optimized area of 0.02 μm^2 with a delay of two clock phases or 0.5 clock cycles. The performance of the proposed optimized 1-bit binary comparator is better with respect to number of cells and delay compared to the existing designs [8–26]. It is observed that an improvement of 3% in cell number and 50% in both latency and cost of the design is achieved compared to [26].
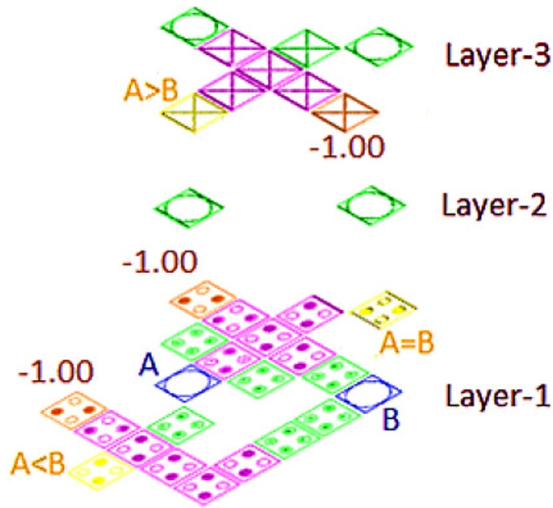
(a)
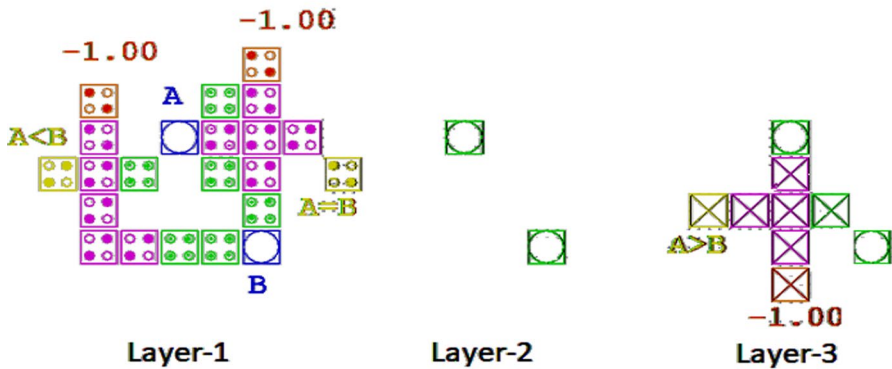
(b)

Fig. 8 1-bit ST comparator: **a** 3D view, **b** individual layers

6 Proposed tree-based stack-type architecture for 4-bit and 8-bit comparators

The tree-based architecture proposed in this paper is a modified and optimized version of the architecture in [10]. The entire comparator circuit was realized using only majority voter gates. However, using the majority voter gates alone increases the circuit complexity. The 4-bit comparator implemented in [10] used a tree-based architecture with 22 majority voter gates and occupied more area. Besides, as the size of input bits is increased in the comparator, the complexity of the layout is also increased.
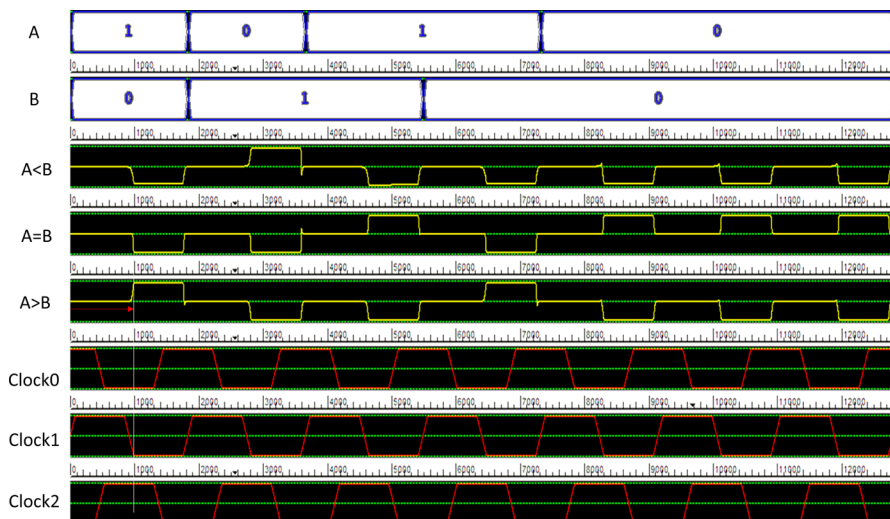
Fig. 9 Simulation results of proposed 1-bit stack-type comparator

Table 1 Comparison of proposed and existing 1-bit comparators

Comparator design	Quantum cells	Area (μm^2)	Latency (clock cycles)	Cost
Hayati [8]	79	0.038	1	0.038
Roohi [9]	54	0.04	1	0.04
BlazLampreht [11]	221	0.07	9	0.63
Meghanad [12]	135	0.2	2	0.4
Abdullah-Al Shafi [13]	117	0.171	1	0.171
QiuKe-Ming [14]	100	0.127	1	0.127
Jun-Wen [15]	100	0.11	0.75	0.0825
Xia [16]	97	0.032	1	0.032
Ajitha [17]	81	0.05	1	0.05
Ghosh [18]	73	0.064	1	0.064
Janulis [19]	67	0.08	1	0.08
Umira [20]	58	0.055	0.75	0.04125
Bahar [21]	47	0.042	0.5	0.021
Bahrepour [22]	43	0.071	1	0.071
Roy design-I [23]	40	0.032	1	0.032
Roy design-II [23]	37	0.028	1	0.028
Shiri [24]	38	0.03	0.5	0.015
Das [25]	37	0.023	0.75	0.01725
Marshal Raj [26]	**34**	**0.02**	**1**	**0.02**
Proposed stack-type design	**33**	**0.02**	**0.5**	**0.01**
Improvement	**3%**	**–**	**50%**	**50%**

Hence, in this paper, a modified tree-based design is proposed to reduce the quantum cells and occupancy area. Three majority voter gates and two inverters of the logic circuit in [5] are replaced by a single EXOR gate, as shown in Fig. 10 without violating the basic function $\overline{A_n}B_n + A_n\overline{B_n} \Rightarrow A_n \oplus B_n$. If each majority voter gate takes five quantum cells and each inverter takes four cells, then the number of cells required to implement the EXOR function in [10] is 24 cells, including one cell to propagate the data to the next cell. However, in the proposed design, only 11 cells are necessary, leading to the minimization of the area in the overall comparator circuit realization.

In general, an n-bit comparator has two n-bit inputs represented as $A_{n-1}A_{n-2}\cdots A_1A_0$ and $B_{n-1}B_{n-2}\ldots B_1B_0$. Considering a 4-bit comparator, the two inputs of the comparator are $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$, as shown in Fig. 11. This 4-bit comparator circuit has a significant reduction in the number of cells as the proposed design replaces three majority voter gates and two inverter gates at each level with an equivalent EXOR gate as follows:

$$\overline{A_3}B_3 + A_3\overline{B_3} \Rightarrow A_3 \oplus B_3 \ :: 3 \, \text{majority gates} \ + \ 2 \, \text{inverter} \rightarrow 1 \, \text{EXOR gate}$$

$$\overline{A_2}B_2 + A_2\overline{B_2} \Rightarrow A_2 \oplus B_2 \ :: 3 \, \text{majority gates} \ + \ 2 \, \text{inverter} \rightarrow 1 \, \text{EXOR gate}$$

$$\overline{A_1}B_1 + A_1\overline{B_1} \Rightarrow A_1 \oplus B_1 \ :: 3 \, \text{majority gates} \ + \ 2 \, \text{inverter} \rightarrow 1 \, \text{EXOR gate}$$

$$\overline{A_0}B_0 + A_0\overline{B_0} \Rightarrow A_0 \oplus B_0 \ :: 3 \, \text{majority gates} \ + \ 2 \, \text{inverter} \rightarrow 1 \, \text{EXOR gate}$$

This modified tree is combined with the optimized stack-type design model to achieve better performance in the area and number of cells. The proposed tree-based stack-type (TB-ST) 4-bit comparator is designed by the following steps:
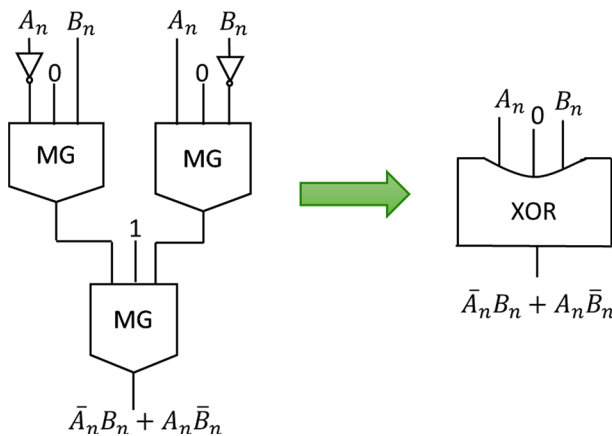

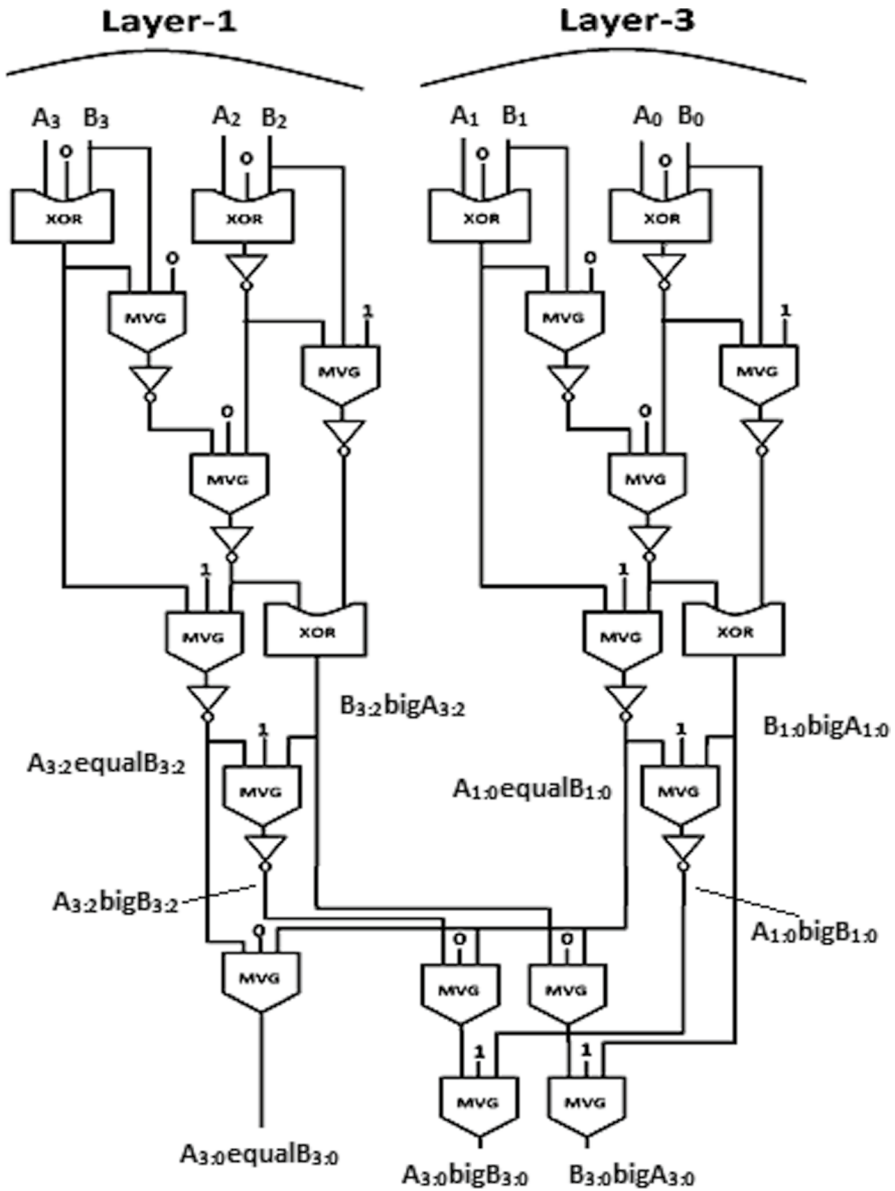
Fig. 10 Majority logic reduction

Fig. 11 QCA schematic of 4-bit ST-TB comparator

1 Two TB 2-bit comparators in layer-1 and another two TB 2-bit comparator in layer-3 are first taken as shown in Fig. 11. In this way, the number of cells as well as the area of the 4-bit comparator is reduced.

2 The outputs of the 2-bit comparator in layer-3 are brought down to the layer-1 via layer-2 to perform the AND/OR operations with the outputs of 2-bit compara-

tor in layer-1. The outputs of the 2-bit comparators in layer-3 are $A_{3:2}$**equal**$B_{3:2}$, $A_{3:2}$**big**$B_{3:2}$, $B_{3:2}$**big**$A_{3:2}$ and in layer-1 are $A_{1:0}$**equal**$B_{1:0}$, $A_{1:0}$**big**$B_{1:0}$, $B_{1:0}$**big**$A_{1:0}$.

3   Next, the two output sections are ANDed and ORed using the majority voter gate (MVG) indicated as $M(X, Y, 0)$ and $M(X, Y, 1)$, respectively, and given in Eqs. (5)–(7), to obtain the final comparator output.

The following equations represent the 'equal to,' 'greater than,' and 'less than' outputs of the proposed TB-ST 4-bit comparator:

$$A_{3:0}\text{equal}B_{3:0} = M((A_{3:2}\text{equal}B_{3:2}), (A_{1:0}\text{equal}B_{1:0}), 0) \tag{5}$$

$$A_{3:0}\text{big}B_{3:0} = M(M((A_{3:2}\text{big}B_{3:2}), (A_{1:0}\text{equal}B_{1:0}), 0), A_{1:0}\text{big}B_{1:0}, 1) \tag{6}$$

$$B_{3:0}\text{big}A_{3:0} = M(M((B_{3:2}\text{big}A_{3:2}), (A_{1:0}\text{equal}B_{1:0}), 0), B_{1:0}\text{big}A_{1:0}, 1) \tag{7}$$

where $M(a, b, c) = ab + bc + ca$ corresponds to the majority voter gate's function. If any one of the inputs is 0, it gives AND function, while if any one of the inputs is 1, it acts as OR.

With these steps, the architecture of 4-bit TB-ST comparator is designed and layout is shown in Fig. 12. The cells in the layer-1 are indicated as normal cells, and the cells in the layer-3 are indicated as crossbar cells. For the stack-type design, layer-1 lies beneath the layer-3 via layer-2 as shown in Fig. 12.

Furthermore, the layout of 8-bit comparator is designed using the proposed TB-ST 4-bit comparator. For this, the following steps are to be carried out:
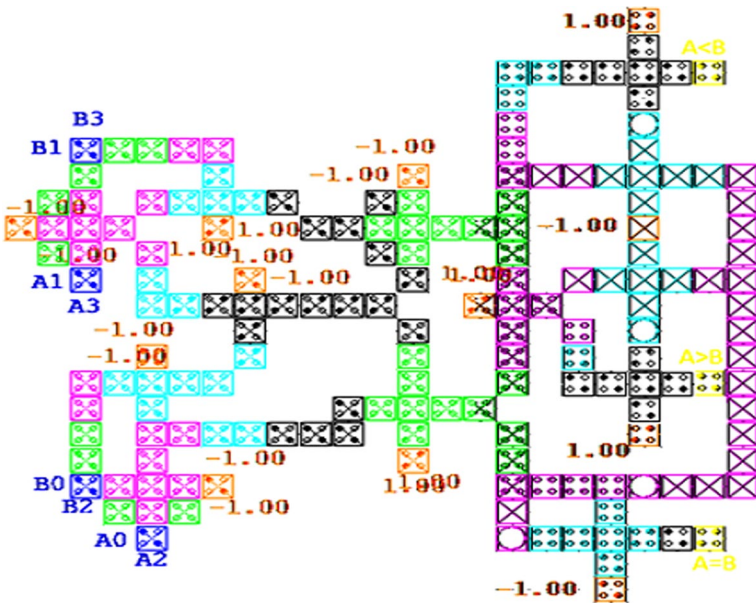

Fig. 12 Proposed TB-ST 4-bit comparator design

1 Two TB-ST 4-bit comparators are required.
2 The three outputs of each TB-ST 4-bit comparator are combined using AND/OR gates to obtain the final output of the 8-bit comparator.

For the QCA design, the AND and OR functions are realized using the majority voter gates. The logic circuit of the proposed TB-ST 8-bit comparator is shown in Fig. 13a.

The QCA layout architecture of the proposed TB-ST 8-bit comparator is shown in Fig. 13b. In the QCA layout, the two TB-ST 4-bit comparators are placed on either side for the compactness of the design. The outputs of each 4-bit comparator are applied to majority voter gates to perform the AND/OR functions to obtain the final 8-bit comparator output.
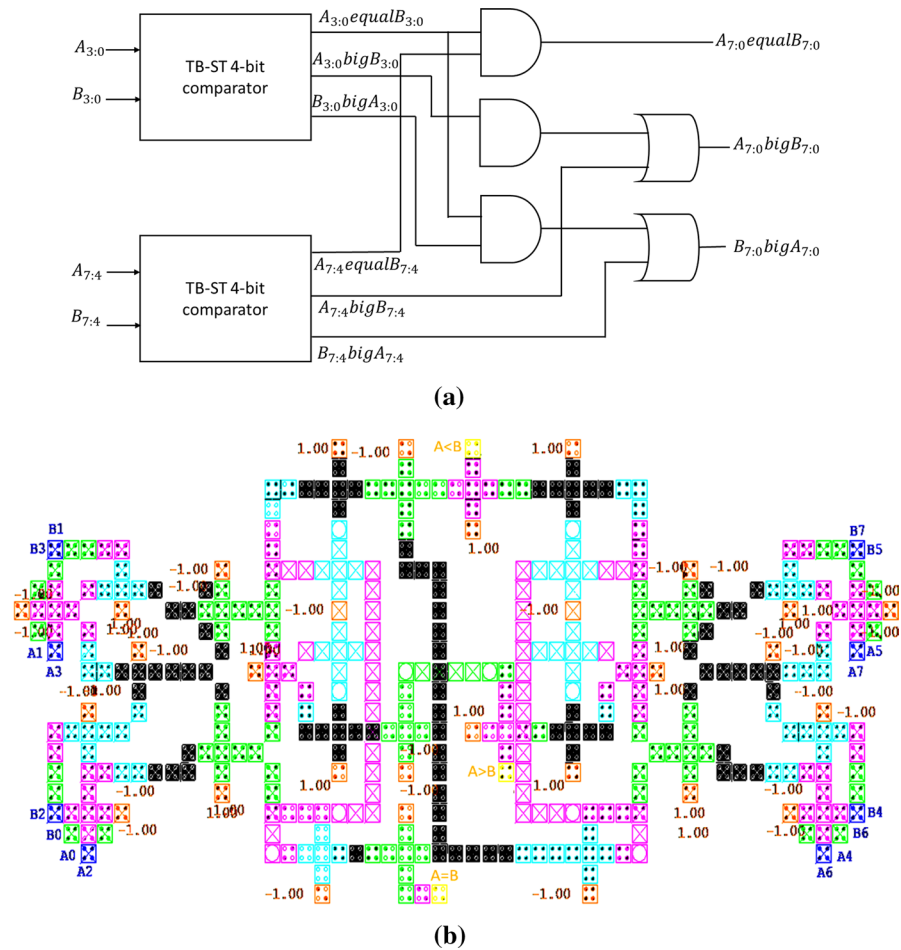

Fig. 13 Proposed TB-ST 8-bit comparator: **a** logic diagram, **b** QCA design

7 Simulation results and discussion

The proposed TB-ST 8-bit comparator is simulated on QCADesigner 2.0.3 using the coherence vector engine simulation method. The specifications of this simulation method are as shown in Fig. 14a. The simulation specifications indicate the transient analysis of the design. Some of the important specifications are temperature, clock period, incremental step size, high and low levels of clock signal, layer separation, and radius of effect. The coherence vector engine uses the Euler method to compute the comparison function. As the Euler method is fast and efficient, it was utilized to perform the mathematical operations. Temperature is another most important parameter of QCA circuits. In this paper, the proposed TB-ST 8-bit comparator can be made to operate at different temperatures, including room temperature 300 K.

The simulation results are shown in Fig. 14b–d; the input conditions and output values are marked with different colors for 'greater than,' 'less than,' and 'equal to' operations. The red color oval highlights the 'less than' operation (in this case $A < B$ is asserted to 1); the yellow color oval highlights the 'greater than' operation (in this case $A > B$ is asserted to 1); and blue color highlights the 'equality' operation between inputs (in this case $A = B$ is asserted to 1). It can be observed that the delay of the proposed design is 2.5 clock cycles as each of comparison results occurs only after 2.5 clock cycles.

As shown in Fig. 14b, when $A = 63$ and $B = 248$, the 8-bit comparator output is $A < B$, which is indicated in a red oval shape. When $A = 255$ and $B = 12$, then the output of the comparator is A > B, which is indicated in a yellow oval shape. Similarly, when $A = 96$ and $B = 96$, then the output of the comparator is $A = B$, which is highlighted in blue oval shape after 2.5 cycles of delay.

The simulation was carried out for repeatability and for several more cases of inputs as shown in Fig. 14c and d. For $A = 6$ and $B = 8$, the 8-bit comparator output is $A < B$, which is indicated in a red oval shape. When input was changed to $A = 88$ and $B = 34$, then the output of the comparator is $A > B$, which is displayed in a yellow oval shape. Similarly, when $A = 32$ and $B = 32$, then the output of the comparator is $A = B$, which is highlighted in blue oval shape after 2.5 cycles of delay. In Fig. 14d, for $A = 4$ and $B = 8$, the 8-bit comparator output is $A < B$, while with $A = 14$ and $B = 3$, the output of the comparator is $A > B$. Similarly, when input is made equal, $A = 6$ and $B = 6$, then the output of the comparator is $A = B$, which is highlighted in a blue oval shape. For the state of $A = 0$ and $B = 7$, the output is asserted for $A < B$, which is indicated with a red oval shape.

The comparison results of some of the existing and proposed 4-bit comparator designs are tabulated in Table 2. The results show that the proposed TB-ST 4-bit comparator provides better performance in terms of the number of cells and area. There is a significant improvement in both these parameters of 6.7% and 48%, respectively, compared to the design [27]. Due to this novel design, the size of the comparator is approximately reduced to half that of the previous designs. However, the delay of the proposed design is slightly increased to achieve the full output voltage swing. The comparison charts are depicted in Fig. 15a.
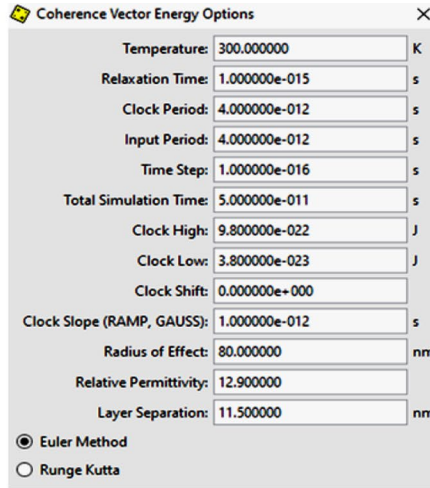
(a)
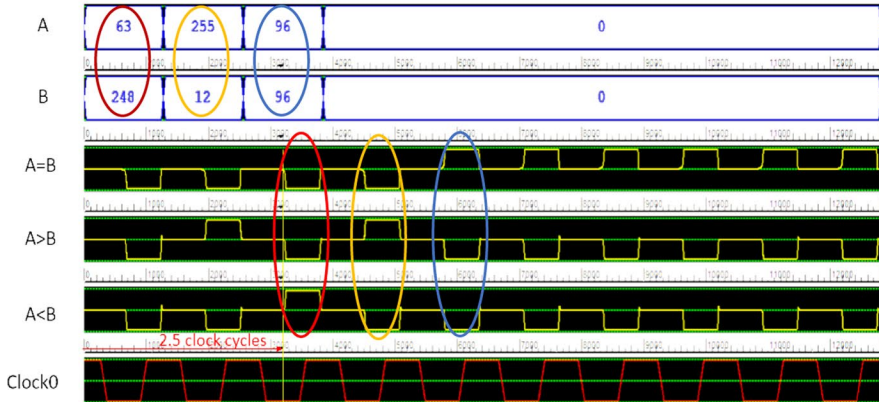
(b)
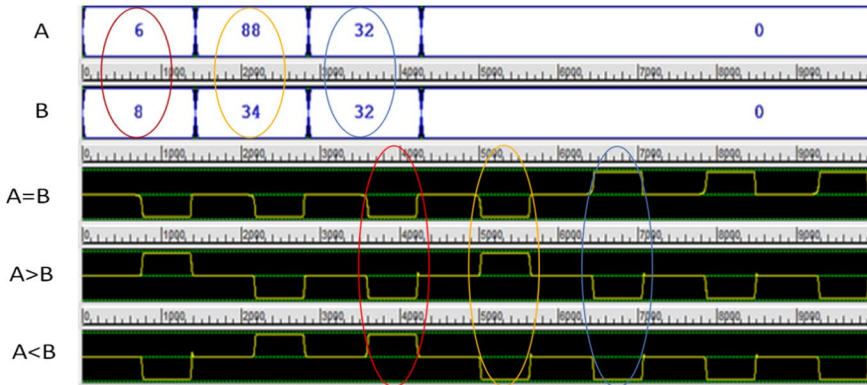
(c)

Fig. 14 Simulation of proposed TB-ST 8-bit comparator: **a** simulation engine specifications, **b–d** simulation results for different cases of input
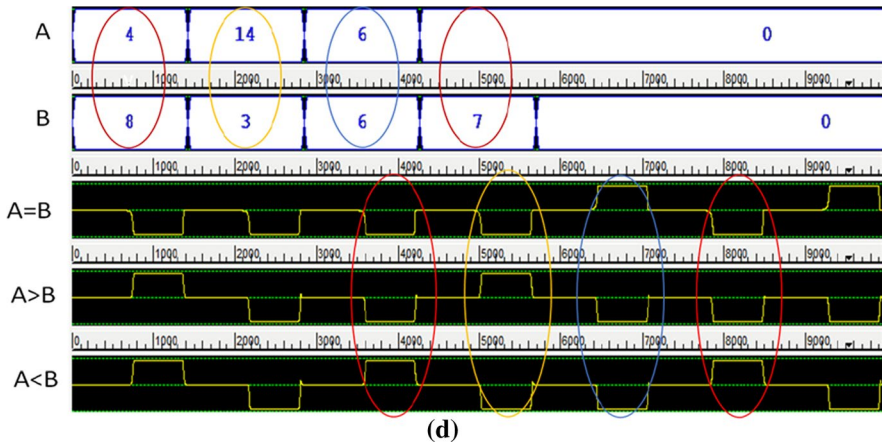
(d)

Fig. 14 (continued)

Table 2 Comparison of existing and proposed 4-bit comparators

Comparator design	No. of cells	Area (μm^2)	Latency (clock cycles)	Cost
Xia [16]	722	1.63	3.5	5.705
Jun-wen [15]	853	1.06	1.75	1.855
Perri-CB [27]	297	0.41	1.25	0.5125
Perri-TB [27]*	**295**	**0.41**	**1.25**	**0.5125**
Proposed TB-ST layout	**275**	**0.21**	**2**	**0.42**
Improvement	**6.7%**	**48%**	**–**	**18%**

The comparison results of the proposed TB-ST 8-bit comparator are tabulated in Table 3, and the corresponding chart is demonstrated in Fig. 15b. The results indicate that the proposed TB-ST 8-bit comparator is better with respect to area and number of quantum cells. The quantum cell count and area are improved by 10.5% and 38%, respectively, compared to the design in [27].

The simulation results have been demonstrated for 1-bit, 4-bit, and 8-bit comparators. The performance of the proposed architectures is proved in terms of quantum cells, area, and delay and compared with the existing methods. Nevertheless, even if the number of bits in the inputs is increased, there is only an incremental increase in these parameters; however, this increase is not greater than the parameters of existing works. On simulating a 16-bit comparator using the proposed TB-ST layout, the number of quantum cells required is 1252 in an occupied area of 1.25 μm^2, which is an improvement by 5% and 25.6%, respectively, while the delay is three clock cycles. Similarly, a 32-bit comparator using the proposed architecture requires only 2675 quantum cells in an occupied area of 2.05 μm^2 with a delay of 3.5 clock cycles. This is an improvement by 9.35% and 28.8%, respectively. Hence, it can be summarized that the average improvement

(a)

(b)
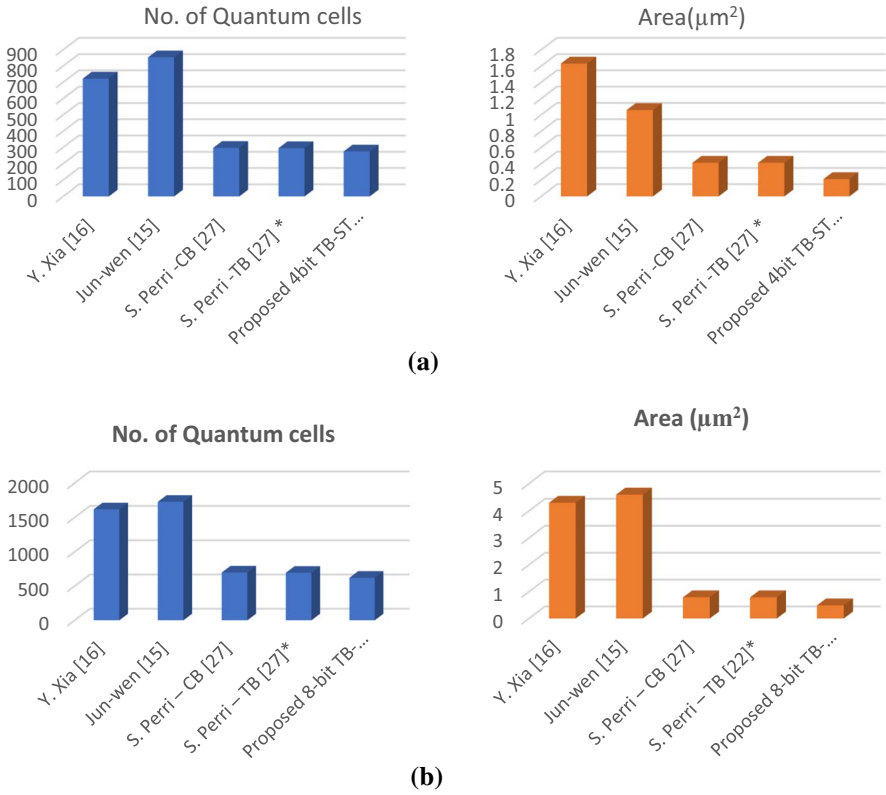
Fig. 15 Quantum cell and area comparison chart of **a** 4-bit comparators, **b** 8-bit comparators

Table 3 Comparison of 8-bit comparators

Comparator design	No. of cells	Area (μm^2)	Latency (clock cycles)	Cost
Y. Xia [16]	1620	4.3	4.5	19.35
Jun-wen [15]	1730	4.6	4	18.4
Perri-CB [27]	699	0.79	1.75	1.385
Perri-TB [27]*	**695**	**0.79**	**1.75**	**1.385**
Proposed TB-ST layout	**622**	**0.49**	**2.5**	**1.715**
Improvement	**10.5%**	**38%**	**–**	**–**

Table 4 Energy dissipation comparison

Design	Sum_Ebath(eV)	Avg_Ebath(eV)	Temperature(K)
8-bit comparator in [27]	4.56e–001	3.35e–002	1
Proposed 8-bit TB-ST comparator	3.05e–001	2.77e–002	1
Improvement (%)	32.7	17.3	–

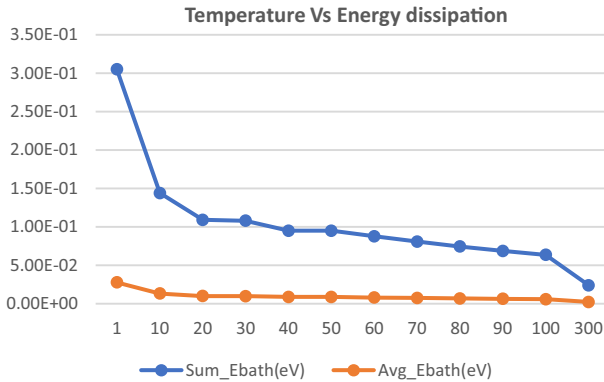**Table 5** Energy dissipation characteristics of the proposed design

Design	Sum_Ebath (eV)	Avg_Ebath (eV)	Temperature (K)
Proposed 8-bit TB-ST comparator	3.05e–001	2.77e–002	1
	1.44e–001	1.31e–002	10
	1.09e–001	9.90e–003	20
	1.08e–001	9.77e–003	30
	9.50e–002	8.63e–003	40
	9.50e–002	8.63e–003	50
	8.77e–002	7.97e–003	60
	8.07e–002	7.34e–003	70
	7.43e–002	6.76e–003	80
	6.86e–002	6.23e–003	90
	6.35e–002	5.77e–003	100
	2.38e–002	**2.17e–003**	**300**

factors for the number of quantum cells and area are 8% around 30%, respectively, whereas delay constantly increases in the order of 0.5 clock cycles.

8 Energy dissipation analysis of TB-ST 8-bit comparator

In addition to reducing the number of quantum cells and area, the proposed 8-bit TB-ST comparator is also proved to have better energy dissipation. In QCA, each quantum cell in the circuit dissipates some amount of energy. This energy dissipation varies from cell to cell. This variation in energy among cells is due to the difference in data traffic among the cells. The energy dissipation analysis and comparison of proposed architecture are performed with the 8-bit comparator designed in [27] and shown in Table 4. The comparative analysis proves that the proposed 8-bit comparator provides significant improvement in energy dissipation. The Sum_Ebath is improved by 32.7%, and Avg_Ebath is enhanced by 17.3%.

In addition to this, the energy dissipation analysis is studied at various temperatures using QCADesigner-E tool. Two parameters are available in this tool to deduce the energy consumption of every cell. They are Sum_Ebath and Avg_Ebath [30]. The Sum_Ebath provides the total energy of all the quantum cells used in the bath, and Avg_Ebath describes the average energy of all the quantum cells in the bath. The term 'bath' refers to the group of quantum cells whose energy dissipation is

Fig. 16 Characteristics of energy dissipation of proposed design

more in the entire layout environment. The values of Sum_Ebath and Avg_Ebath in eV are tabulated at different temperatures in Table 5.

It is clear that the proposed 8-bit TB-ST comparator works well at room temperature 300 K with a lower value of Sum_Ebath and Avg_Ebath. The Sum_Ebath and Avg_Ebath of the proposed TB-ST 8-bit comparator are 2.38e–002 eV and 2.17e–003 eV, respectively, at ambient temperature. The characteristic curves of energy dissipation (eV) with respect to temperature (Kelvin) are shown in Fig. 16.

9 Conclusion

In this paper, an innovative architecture named TB-ST is suggested to design a $2n$-bit comparator. At first, an optimized 1-bit ST comparator is designed and implemented in Quantum-dot Cellular Automata in three layers to optimize the area. Later, an area optimized 4-bit TB-ST comparator is designed, and using this, an 8-bit TB-ST comparator is developed and results were studied. The designed comparators are simulated with test input vectors using QCADesigner 2.0.3 tool. The proposed ST 1-bit comparator shows better performance with 33 quantum cells and 0.5 clock cycles of delay, with an improvement of 3% and 50% in quantum cells and area occupancy, respectively. Besides, the designed TB-ST 4-bit comparator also shows better performance with 275 quantum cells in an optimized area of 0.21 μm^2, having an improvement of 6.7% and 48%, respectively. Similarly, the extended TB-ST 8-bit comparator has been developed with 622 quantum cells in an area of 0.49 μm^2 with an enhancement of 10.5% and 38%, respectively. Further extensions to 16-bit and 32-bit comparators also demonstrate improved performances.

Lower energy dissipation of the design compared to previous architectures is also verified. Results have proved an improvement of 32.7% in energy dissipation at 1 K. In addition, the characteristics between energy dissipation and temperature depict that the proposed comparator has a stable response at room temperature.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Porod W, Lent CS, Bernstein GH (1994) Quantum cellular automata. Nanotechnology 4:49–57
2. Tougraw PD, Lent CS (1994) Logical devices implemented using quantum cellular automata. J Appl Phys 46556:1818–1825
3. Qi H, Sharma S, Li Z, Snider GL, Orlov AO, Lent CS, Fehlner TP (2003) Molecular quantum cellular automata cells. Electric field driven switching of a silicon surface bound array of vertically oriented two-dot molecular quantum cellular automata. J Am Chem Soc 125:15250–15259
4. Tokunaga K (2010) Metal dependence of signal transmission through molecular quantum-dot cellular automata (QCA): a theoretical study on Fe Ru, and Os mixed-valence complexes. Materials 3:4277–4290. https://doi.org/10.3390/ma3084277
5. Ashrafi S et al (2020) Novel sonochemical synthesis of $Zn_2V_2O_7$ nanostructures for electrochemical hydrogen storage, International Journal of Hydrogen energy. Int J Hydrog Energy 45(41):21611–21624. https://doi.org/10.1016/j.ijhydene.2020.05.166
6. Zinatloo-Ajabshir S, Morassaei MS, Amiri O, Salavati-Niasari M, Foong LK (2020) $Nd_2Sn_2O_7$ nanostructures: green synthesis and characterization using date palm extract, a potential electrochemical hydrogen storage material. Ceram Int 46:17186–17196. https://doi.org/10.1016/j.ceramint.2020.03.014
7. Zinatloo-Ajabshir S, Salehi Z, Amiri O, Salavati-Niasari M (2019) Simple fabrication of $Pr_2Ce_2O_7$ nanostructures via a new and ecofriendly route; a potential electrochemical hydrogen storage material. J Alloy Compd 791:792–799
8. Mohsen H, Rezaei A (2012) Design and optimization of full comparator based on quantum-dot cellular automata. ETRI J 34:284–287
9. Roohi A, Thapliyal H, DeMara RF (2015) Wire crossing constrained QCA circuit design using bilayer logic decomposition. Electron Lett 51:1677–1679
10. Babu BM, Bharadwaja PVSR and Neelima K (2015) Stubby delay comparator in QCA: the C-gate. In: Proceedings of International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], Nagercoil, IEEE, pp 1–5
11. Lampreht B et al (2008) Quantum-dot cellular automata serial comparator. In: Proceedings of EUROMICRO Conference on Digital System Design Architectures, Methods and Tools, Parma, IEEE, pp 447–452
12. Wagh MD, Sun Y, Viswanath A (2008) Implementation of comparison function using quantum-dot cellular automata. NSTI-Nanotech 3:76–79
13. Abdullah-Al-Shafi Md., Bahar AN (2016) Optimized design and performance analysis of novel comparator and full adder in nanoscale. Cogent Eng 3:1237864. https://doi.org/10.1080/23311916.2016.1237864
14. Ke-ming Q and Yin-shui X (2007) Quantum-dots cellular automata comparator. In: Proceedings of International Conference on ASIC, Guilin, IEEE, pp 1297–1300
15. Jun-wen L and Yin-shui X (2008) A novel design of quantum-dots cellular automata comparator using five-input majority gate. In: Proceedings of International Conference on Solid-State and Integrated Circuit Technology (ICSICT), Qingdao, IEEE, pp 1–3
16. Xia YS and Ke-Ming Q (2007) Comparator design based on quantum-dot cellular automata. In: Proceedings of International Conference on ASIC, IEEE, pp 1297–1300
17. Ajitha D, Ramanaiah KV and Sumalatha V (2014) A novel design of cascading serial bit-stream magnitude comparator using QCA. In: Proceedings of International Conference on Advances in Electronics Computers and Communications, Bangalore, IEEE, pp 1–6
18. Ghosh B, Gupta S and Kumari S (2012) Quantum dot cellular automata magnitude comparators. In: Proceedings of International Conference on Electron Devices and Solid-State Circuit (EDSSC), Bangkok, IEEE, pp 1–2
19. Janulis JR, Tougaw PD, Henderson SC, Johnson EW (2004) Serial bit-stream analysis using quantum-dot cellular automata. IEEE Trans Nanotechnol 3:158–164

20. Umira S, Qadri R, Bangi ZA and Banday MT (2018) A novel comparator-a cryptographic design in quantum dot cellular automata. In: Proceedings of International Conference on Sustainable Energy, Electronics, and Computing Systems (SEEMS), Greater Noida, India, IEEE, pp 1–10
21. Bahar AN, Roy K, Asaduzzaman Md, Rahman Bhuiyan MM (2017) Design and Implementation of 1-bit comparator in quantum-dot cellular automata. Cumhur Univ Fac Sci J (CSJ) 38:146. https://doi.org/10.17776/csj.72358
22. Davoud B (2015) A novel full comparator design based on quantum-dot cellular automata. Int J Inf Electron Eng 5:406–410
23. Roy SS, Mukherjee C, Panda S, Mukhopadhyay AK and Maji B (2017) Layered T comparator design using quantum-dot cellular automata. In: Proceedings of devices for integrated circuit (DevIC), Kalyani, IEEE, pp 90–94
24. Shiri A, Rezai A, Mahmoodian H (2019) Design of efficient coplanar 1-bit comparator circuit in QCA technology. Facta Univ Ser Electron Energ 32:119–128
25. Das JC, De D (2016) Reversible comparator design using quantum dot-cellular automata. IETE J Res 62(3):323–330
26. Raj M and Gopalakrishnan L (2019) Efficient quantum-dot cellular automata comparator for nanoscale communication. In: Proceedings of International Conference on Microwave Integrated Circuits, Photonics and Wireless Networks (IMICPW), Tiruchirappalli, India, IEEE pp 307–311
27. Stefania P, Pasquale C, Giuseppe C (2019) Design of efficient binary comparators in quantum-dot cellular automata. IEEE Trans Nanotechnol 13:192–202
28. Newaz A, Waheed S, Hossain N (2017) A novel 3-input XOR function implementation in quantum dot-cellular automata with energy dissipation analysis. Alex Eng J 57(2):729–738
29. Adelnia Y, Abdalhossein R (2019) A novel adder circuit design in quantum-dot cellular automata technology. Int J Theor Phys 58:184–200
30. Gudivada AA, Sudha GF (2020) Design of Baugh-Wooley multiplier in quantum-dot cellular automata using a novel 1-bit full adder with power dissipation analysis. SN Appl Sci 2:813. https://doi.org/10.1007/s42452-020-2595-5

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.