Int. j. inf. tecnol. (February 2022) 14(1):41–47
https://doi.org/10.1007/s41870-021-00836-2

ORIGINAL RESEARCH

A modular approach to design ternary content addressable memory architecture in quantum dot cellular automata

Mrinal Goswami[1] · Jayanta Pal[2] · Rohit Tanwar[1] · Prashant Rawat[1]

Received: 7 June 2021 / Accepted: 7 November 2021 / Published online: 14 January 2022

Abstract Quantum-dot cellular automata (QCA) is evolving as a prominent field coupled nano-electronic technology to fulfill the demand for low power high-speed compact devices. The features like high device density, enviable low power dissipation, and higher switching-speed make QCA more attractive for nano circuit realization. In QCA, the priority has been given to design random access memory (RAM) over content addressable memory (CAM) due to its various applications in digital circuits. However, CAM is equally important as it also has applications in the network layer. CAM has the potential to find memory content very rapidly over the network. This paper presents a cost-efficient, single-layer design of CAM and its ternary form (TCAM) in QCA. The proposed 1-bit CAM cell consists of a D flip flop and an XNOR gate. To test the flexibility of the proposed TCAM, a 4-bit TCAM is also realized. The comparison results signify a considerable reduction in terms of cost (83%), power dissipation (34%), the area covered (20%), total cell count (22%), and latency (40%). In this work, the QCADesigner simulator is utilized to design all the QCA circuits. Also, the energy estimation of the proposed TCAM cell is done by the QCAPro simulator.

Keywords Content-addressable memory (CAM) · Nano communication · Quantum dot cellular automata (QCA) · Ternary content addressable memory (TCAM) · Cellular automata · Random access memory (RAM)

✉ Mrinal Goswami
mgoswami@ddn.upes.ac.in

[1] Department of Systemics, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India

[2] Department of Information Technology, Tripura University, Agartala, India

1 Introduction

CMOS based digital industry possess the benefit of the fast switching speed of transistor reducing the feature size for the last six decades. However, reducing feature size beyond a certain limit creates serious problems (like lithography, quantum effects, power consumption, inter-connection) at the nanoscale level. These challenges open up the search for a viable alternative technology that can fulfill the increasing demands of chip design beyond the CMOS level. Quantum-dot cellular automata (QCA), a field coupled nanotechnology, has the potential to scale down beyond the range of CMOS [8]. QCA designs can be practically implemented using four technology: metal island, semiconductor, molecular, and magnetic. In this brief, the QCA designs are proposed based on semiconductor QCA, which can be operated at room temperature [6].

Data processing application heavily uses scanning of items in memory. The scan procedure includes choosing a sequence of addresses, reading the content in memory, and comparing the information with the item selected. The time required for this type of memory scan can be reduced considerably if stored data can be chosen by providing data itself in the input rather than by an address. This is possible using content addressable memory (CAM) instead of random access memory (RAM). Unlike RAM, CAM can scan the entire memory content in parallel, and if the match is found, it returns the address where the requested input is present. However, CAM requires exact data inputs ('0' and '1') to scan the memory content. To address this issue, ternary content-addressable memory (TCAM) is used, which supports partial scan (‘0’, ‘1’, and X(don’t care)), that reduces scan time as compared to CAM. TCAM is an updated version of CAM. TCAM can scan entire contents in a single clock cycle. Various networking equipment, such as routers, utilizes the benefits of TCAM to increase the speed of route look-up in the network layer [16].

Several designs for network equipment in QCA have been proposed previously, such as circuit switching [4], butterfly switching network [3], nano-router [17], reversible crossbar switch [5]. On the other hand, a single attempt is made to realize a TCAM memory structure in QCA [16]. However, the design covers a considerable area, dissipates more power, and the design cost is very high.

The whole manuscript is divided into the following sections: Sect. 2 introduced an overview of quantum-dot cellular automata (QCA). Section 3 will discuss existing work. The proposed designs are presented in Sect. 4. Section 6 will analyze the power consumption of the presented designs. The concluding remark of the brief is presented in Sect. 7.

2 Basic foundation

QCA logic gates are the core entity found in any QCA circuit. The two basic logic gates in QCA are the majority voter and the inverter. All the complex QCA circuits can be easily implemented using combinations of majority voters and inverters. The following section will discuss the various QCA logic gates.

2.1 Quantum dot cellular automata (QCA)

QCA is a field-coupled technology that does not utilize transistors, but instead of transistors, QCA uses charge configuration within a QCA cell. A standard QCA cell can occupy four quantum dots placed at four corners (Fig. 1a). The free movement of electrons within the dots is possible via tunneling, which creates a bi-state device with logic 0 (P = – 1) and logic 1 (P = + 1) as shown in Fig. 1b. The basic building block which is used in almost all the QCA circuits is 3-input majority voter (Fig. 1c) represented as M(K, L, M) = KL + LM + KM where K, L, M are its three inputs [12]. As the name suggests, the majority voter will vote the majority of its inputs and pass it to the output. If any two inputs are set at ‘1’, the majority voter will produce ‘1’ as an output. On the other hand, if any two inputs are set at ‘0’, then the majority voter will have ‘0’ as an output. The eight input combinations, along with their corresponding outputs, are shown in Table 2.1. By presetting the polarization of any one of its inputs as P = –1 and P = +1, it can be configured as an AND/OR gate, respectively. Two possible QCA inverter implementations exist, as shown in (Fig. 1d). There are two different types of coplanar wire-crossing found in QCA, as shown in Fig. 2. The first structure (from left) is the most conventional coplanar cross-over found in QCA. It utilizes two different types of QCA cells to transfer two different signals on the same plane. The rest two structures represent clock zone-based cross-over, which uses the same kinds of cells for cross-over of two signals with a clock phase difference of 180^0 [1, 13]. The clock zone-based cross-over is considered the most stable cross-over in QCA. Apart from these two coplanar cross-over, multi-layer cross-over is also possible in QCA. However, the complexity of multi-layer cross-over is much higher than the coplanar cross-over [7, 10, 14].

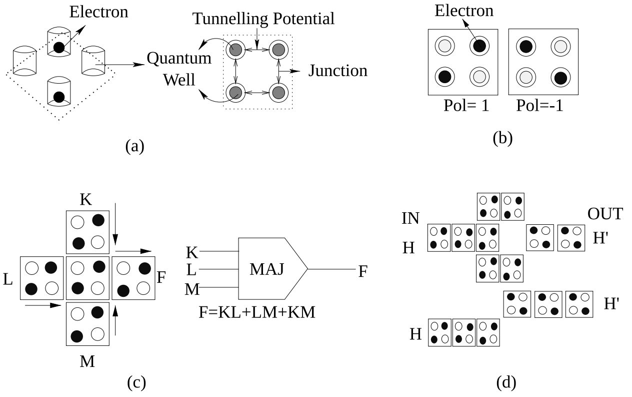


Fig. 1 QCA Fundamentals **a** Quantum dots **b** QCA cells **c** Majority voter **d** Inverter

2.2 QCA clocking

QCA systems utilized four stages clocking scheme (switch, hold, release, and relax), as shown in Fig. 3. In stage 1, the inter-dot barriers between QCA cells are upraised, which polarizes the QCA cells. At the end of stage 1, QCA cells achieve the most extreme potential vitality. In the hold stage, QCA cells retain their state. The inter-dot barriers reach the highest point, due to which it is not possible to change the state in the hold stage. In the release stage, sinking inter-dot barriers provides unpolarization of these

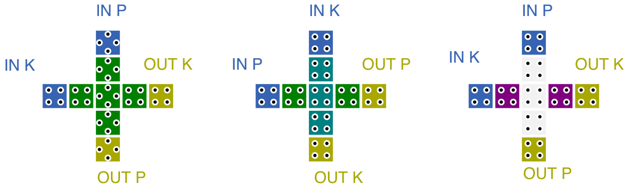


Fig. 2 Coplanar wire-crossing in QCA

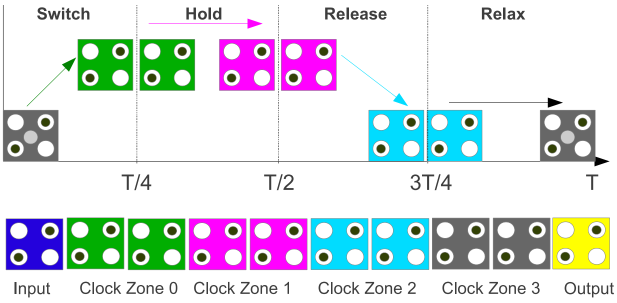


Fig. 3 QCA clocking

cells. Finally, in the relax stage, it retains an unpolarized state.

2.3 Basics of TCAM

The common software methods for table lookup, such as hash function, radix trees, are slow [11]. Various table lookup methods found in the network layer, originally developed in the software, are now being replaced by hardware to speed up the table lookup performance. Content addressable memory (CAM) can be a good solution to meet the required performance. CAM can be considered as an advanced version of random access memory (RAM). Unlike RAM, CAM supports parallel search by the content itself. Two types of CAM can be found: Binary CAM and Ternary CAM or TCAM. Binary CAM or simply CAM is capable of storing and searching in binary values, i.e., in '0' and '1'. Therefore, CAM is suitable for exact table-lookup operations. TCAM enhances the power of CAM by combining a don't care (X) option providing partial matching. Figure 4 shows an TCAM based routing. The input (01101) of TCAM matches with two entries (0110X and 011XX); however, the priority encoder chooses that entry, which matches with the longest prefix (0110X) with the input data word and forwards the address (01) of that input data word.

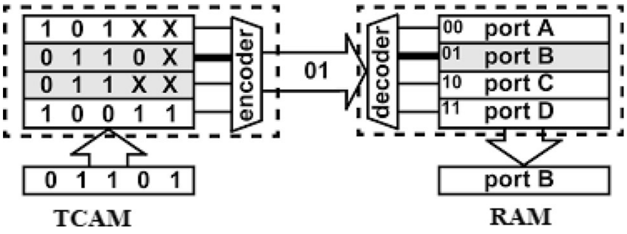


Fig. 4 TCAM based routing

3 Related work

Several designs for random access memory (RAM) in QCA have been proposed previously [9, 18]. On the other hand, TCAM in QCA has not been appropriately studied until now. The first and lone attempt to design a TCAM in QCA can be found in [16]. The authors have first developed a CAM cell which is used further to realize the TCAM cell. The CAM cell design utilizes a flip flop for single-bit memory and an XNOR logic to compare the input with the stored input of the flip-flop. In addition to this, TCAM cell is proposed utilizing two single-bit memory cells and one comparator. The authors have suggested that writing in TCAM should take 2 inputs for specifying the input values, '0', '1', or 'X,' i.e., to write any value TCAM read/write line needs to set at '1' followed by '0'. But writing any value utilizing two inputs substantially slows the writing time of TCAM. Moreover, the multi-layer cross-over is used in the TCAM design, which itself has four times higher complexity in comparison with the coplanar cross-over [10]. The multi-layer cross-over of TCAM can be modified to the coplanar cross-over, but it will increase area consumption and latency.

4 Proposed designs

This section will highlight the QCA designs of proposed CAM and TCAM cells. All the QCA designs proposed in this section are verified with QCADesigner 2.0.3 [15, 20].

4.1 Proposed QCA 1-bit CAM

The proposed CAM cell comprises one D flip-flop and one XNOR gate (works as a comparator). Figure 5 depicts the proposed CAM cell. It has two inputs *R/W*, *I*, and one output *OUT*. *R/W* input is used for changing the mode from read to write or write to read. Input *I* is used to read the input values from memory and store input value into the

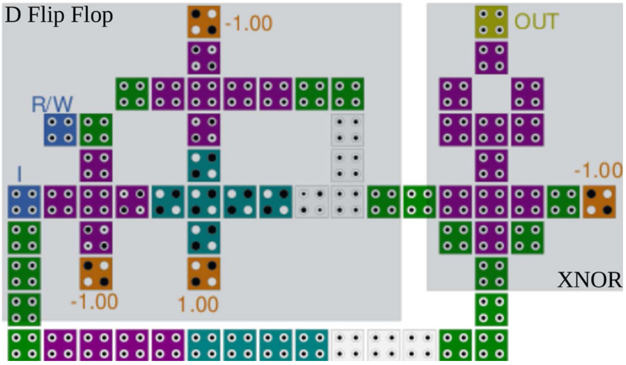


Fig. 5 Proposed QCA CAM

memory. If R/W is SET, then the CAM cell will write the input value in the flip flop provided by I; else, the CAM cell will read the stored value and compare the read value with input I. The comparison is made with the XNOR gate [2].

The correct operation of the proposed CAM cell can be observed in Table 1. In the write mode, for every input I, the CAM cell will trigger one in the output irrespective of its last stored value. On the other hand, in read mode, the input value I is compared with the previously stored value and will trigger in the output only if both the input match. Figure 6 depicts the simulation report of the proposed QCA CAM. It is apparent that in the write operation (R/W = 1), CAM cell triggers one in the output for input $I = 1$ as well as input $I = 0$. In the read mode, the input value (I = 1) is compared with the previously stored value (zero in this case) of the flip flop and found a mismatch.

4.2 Proposed QCA 1-bit TCAM

The CAM is suitable for such applications where the input is matched with the exact content available in the memory. CAM can search and store binary words made of ‘0’s and ‘1’s. TCAM is more powerful than CAM because TCAM can search and store ternary values made of ‘0’s, ‘1’s, and ‘X’s. The state ‘X’ represents don’t care or ‘mask,’ which is useful for partial matching. This means we need to implement a third state (‘X’) in CAM along with ‘0’ and ‘1’. This is done by adding an extra flip-flop (secondary flip flop) to the CAM cell, as shown in Fig. 7a. The corresponding QCA TCAM representation is shown in Fig. 7b. In [16], authors have suggested that writing in TCAM should take 2 input cycles for specifying the input values, ‘0’,’1’, or ‘X,’ i.e., R/W = ‘1’ followed by ‘0’ for writing any value. But writing any value in two inputs substantially slows the writing time of TCAM. This bottleneck is removed by having separate input lines (D_0 and D_1) for which the proposed TCAM is equally fast for writing.

Table 1 Truth table of CAM cell operation

Operation	R/W	I	Previous stored value	OUT
Write	1	0	X	1
Write	1	1	X	1
Read	0	0	0	1
Read	0	0	1	0
Read	0	1	0	0
Read	0	1	1	1

X = Don’t Care

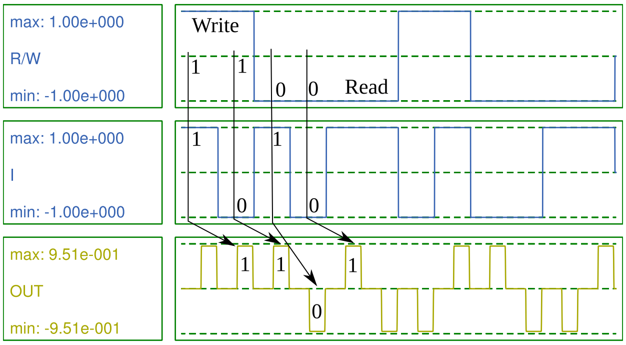
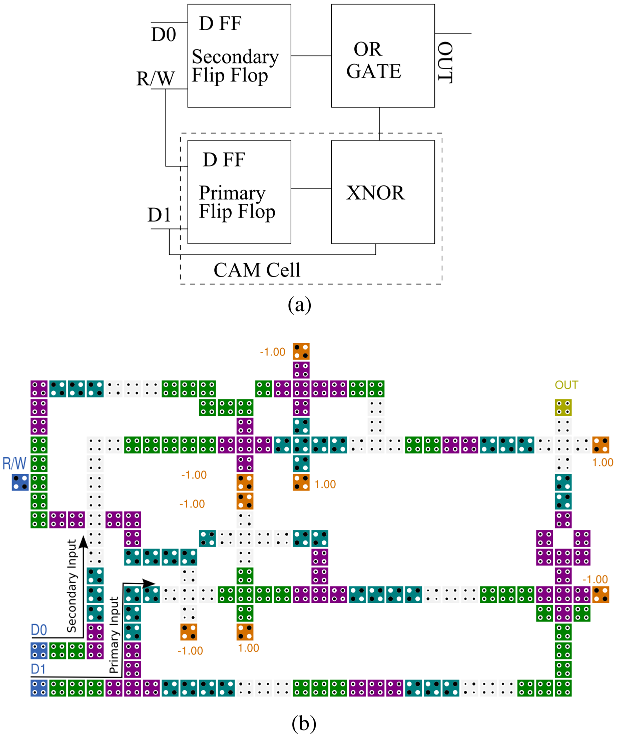


Fig. 6 Simulation report of proposed QCA CAM



(a)

(b)

Fig. 7 The proposed QCA TCAM. **a** Block diagram. **b** QCA layout

The truth table of TCAM cell operation is shown in Table 2. TCAM cell reading operation is similar to the CAM cell, i.e., feed the search value to primary input (D_1) and turn R/W input to zero. In the read mode of TCAM, it is apparent that the secondary input (D_0) has no involvement. However, the previously stored input of the secondary cell comes into play if the primary input and the stored input of the primary cell mismatch. If both (D_1) and the stored input of the primary cell do not match with each other, then TCAM will trigger one in the output provided stored input of secondary cell is one; otherwise, TCAM will print zero in the output. QCA TCAM will always trigger one in the output in the write mode, indicating that the write operation is completed successfully. But the point is, what value is written in both the flip-flops? The answer

Table 2 Truth table of TCAM cell operation

Operation	R/W	D_0	D_1	Stored input of primary cell	Stored input of secondary cell	OUT
Write	1	X	X	X	X	1
Read	0	X	0	0	X	1
Read	0	X	0	1	0	0
Read	0	X	0	1	1	1
Read	0	X	1	0	0	0
Read	0	X	1	0	1	1
Read	0	X	1	1	X	1

can be found in Table 3. In the write mode, the secondary input (D_0) plays a significant role. The don't care (X) is written in the primary cell for the input combinations (D_1 D_0) 10 and 11 in a single clock cycle.

The simulation result of TCAM established the correctness of the proposed TCAM cell, which is depicted in Fig. 8. In the first write operation (R/W = 1), both the inputs ($D_0 = D_1 = 0$) are fixed at zero and the final output of TCAM is one. This write operation will store zero at both the flip flops (Table 3). The next input combination ($D_0 = 0$ and $D_1 = 1$) will store an don't care in the primary flip flop. In the third iteration, a read operation is carried out where both the primary and secondary inputs are zero ($D_0 = D_1 = 0$). The final output for the third iteration is zero because, in this iteration, a zero value is trying to read; however, the previously stored value of the primary cell is one. Therefore, there is a mismatch between the stored and read values; hence, TCAM produces zero in the final output. The next iteration will be a hit because the primary input scanning for 1 and the store value is also 1.

The proper realization of the content addressable memory demands a modular approach to expand its word size. Therefore, four TCAM modules have been serially arranged to make a TCAM storage cell of a word size of four bits, as shown in Fig. 9. Moreover, it can be extended to N, such serially arranged TCAM module for making word storage of N bits. Similarly, it is also possible to extend M, such memory storage word arranged in parallel to form $N \times M$ TCAM. The output of each of these lines can be fed to a priority encoder to design the complete $N \times M$ modular TCAM architecture.

Table 3 Write table of TCAM cell

R/W	D_0	D_1	Value written in primary cell	OUT
1	0	0	0	1
1	0	1	X	1
1	1	0	1	1
1	1	1	X	1

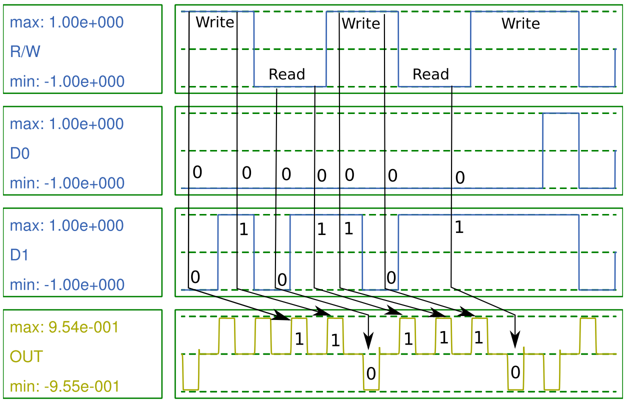


Fig. 8 Simulation report of TCAM Cell

5 Performance analysis

The performance of the proposed CAM and TCAM cell is compared with the previous design found in [16], and it is reported in Table 4. The proposed CAM cell achieves 61% and 45% reduction in terms of area and clock zones, respectively. Also, TCAM supports 20% improvement in the area and 40% improvement in clock zones. Moreover, the TCAM design proposed here generates a 40% faster output than the model discussed in [16]. The proposed TCAM design is a single-layer design that utilizes clock zone-based cross-over.

The proposed TCAM also establishes a cost-efficient design as it shows 83% improvement over its counterparts. The cost of both the proposed QCA design is calculated as per the cost equation (Cost = $[MV^{l} + I + C^{m}] * T^{n}$ where l,m,n $\geq$ 1) found in [10]. Here, MV represents a three-input majority gate; I represents the inverter gate, C represents cross-over, and T represents the delay of the QCA

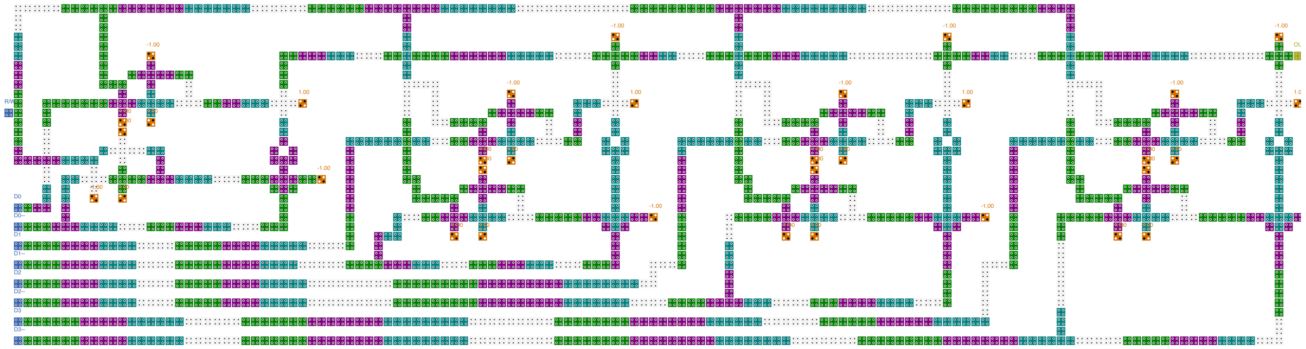


Fig. 9 Proposed 4-bit TCAM

Table 4 Design performance of proposed CAM Cell

Design	Area in μm^2	Clock Zone	No. of Cells	Cost	Type of Cross Over
CAM Cell					
In [16]	0.18	11	141	623	–
Proposed	0.07	6	67	60	–
Improvement	61%	45%	52%	90%	–
TCAM Cell					
In [16]	0.30	20	238	11250	ML
Proposed	0.24	12	185	1836	CZB
Improvement	20%	40%	22%	83%	–

ML = Multi-layer; CZB = Clock Zone Based

Table 5 Energy dissipation of QCA TCAM

Design	Avg. leakage energy dissipation (eV)			Avg. switching energy dissipation (eV)		
	0.5 E_k	1.0 E_k	1.5 E_k	0.5 E_k	1.0 E_k	1.5 E_k
In [16]	0.08635	0.25886	0.45713	0.35397	0.30036	0.25144
Proposed	0.06181	0.18342	0.32359	0.19457	0.16520	0.13837

circuit. To get the same reference for comparison of the proposed design with the previously discussed design, the values of l=2, m=2, and n=3 are taken [10].

6 Power analysis

The well-known QCAPro, a power assessment simulator for QCA circuits, was explored in [19]. This simulator is utilized to examine the power requirement of TCAM cell under 3 distinct energy levels (0.5 E_k, 1.0 E_k, 1.5 E_k). The thermal hotspot shows the power consumption of different TCAM cells at 0.5 E_k energy level. The power consumption differences between the proposed TCAM and the TCAM discussed in [16] is shown in Table 5. The total power consumption of the previous TCAM design is 2.48524 eV, whereas the proposed TCAM consumes only 1.62325 eV. The proposed TCAM is 34% more power-efficient than its counterparts.

7 Conclusion

This paper investigates a cost-efficient content addressable memory (CAM) and its ternary version (TCAM) design. It is a single-layer design that uses the most robust clock zone-based cross-over for wire-crossing. It is seen that the

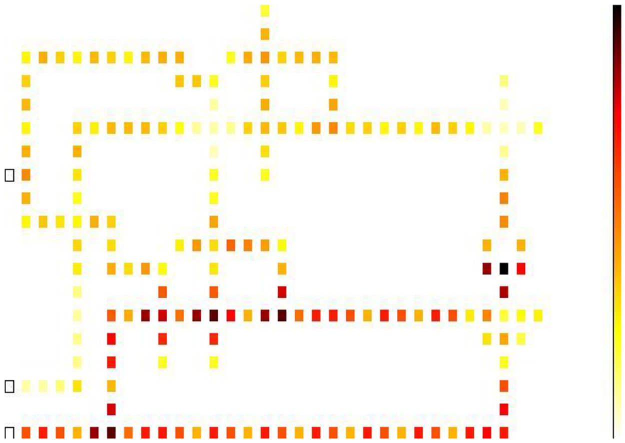


Fig. 10 QCAPro report for TCAM at 0.5 E_k energy level

proposed TCAM design has the least cell count, least area, least clock zone, which in turn leads to the least reported cost so far. Also, the energy dissipated by the proposed TCAM design is almost 34% less than the previously reported design. In addition to this, the flexibility of the proposed TCAM design has also been tested by designing a modular 4-bit TCAM structure.

References

1. Abedi D, Jaberipur G, Sangsefidi M (2015) Coplanar full adder in quantum-dot cellular automata via clock-zone-based crossover. IEEE Trans Nanotechnol 14(3):497–504
2. Ahmad F, Bhat GM, Khademolhosseini H, Azimi S, Angizi S, Navi K (2016) Towards single layer quantum-dot cellular automata adders based on explicit interaction of cells. J Comput Sci 16:8–15. https://doi.org/10.1016/j.jocs.2016.02.005. http://www.sciencedirect.com/science/article/pii/S1877750316300126
3. Bahar AN, Wahid KA (2020) Design of an efficient n x n butterfly switching network in quantum-dot cellular automata (qca). IEEE Trans Nanotechnol 19:147–155. https://doi.org/10.1109/TNANO.2020.2969166
4. Das JC, De D (2017) Circuit switching with quantum-dot cellular automata. Nano Commun Netw 14:16–28. https://doi.org/10.1016/j.nancom.2017.09.002
5. Das JC, De D (2017) Nanocommunication network design using qca reversible crossbar switch. Nano Commun Netw 13:20–33. https://doi.org/10.1016/j.nancom.2017.06.003
6. DiLabio G, Wolkow R, Pitters J, Piva P (2014) Atomistic quantum dot. http://www.google.co.in/patents/US8816479. US Patent 8,816,479
7. Fahimeh DSA, MBKKN, Bagherzadeh N (2021) A combined three and five inputs majority gate-based high performance coplanar full adder in quantum-dot cellular automata. Int J Inf Technol 13:1165–1177. https://doi.org/10.1007/s41870-019-00365-z
8. Lent C, Tougaw PD (1997) A device architecture for computing with quantum dots. Proc IEEE 85(4):541–557. https://doi.org/10.1109/5.573740
9. Liolis O, Mardiris VA, Sirakoulis GC, Karafyllidis IG (2018) Qantum-dot cellular automata ram design using crossbar architecture. In: 2018 IEEE/ACM international symposium on nanoscale architectures (NANOARCH), pp 1–5
10. Liu W, Lu L, O'Neill M, Swartzlander EE (2014) A first step toward cost functions for quantum-dot cellular automata designs. IEEE Trans Nanotechnol 13(3):476–487. https://doi.org/10.1109/TNANO.2014.2306754
11. McAuley AJ, Francis P (1993) Fast routing table lookup using cams. In: IEEE INFOCOM '93 the conference on computer communications, proceedings, vol 3, pp 1382–1391
12. Goswami M, Tanwar R, PR, Sen B (2021) Configurable memory designs in quantum-dot cellular automata. Int J Inf Technol 13:1381–1393. https://doi.org/10.1007/s41870-021-00687-x
13. Safoev N, Ahmed S, KT, Naz SF (2021) Design of fault tolerant bifunctional parity generator and scalable code converters based on QCA technology. Int J Inf Technol. https://doi.org/10.1007/s41870-021-00730-x
14. Raj M, Ahmed S, Lakshminarayanan G (2020) Subtractor circuits using different wire crossing techniques in quantum-dot cellular automata. J Nanophoton 14(2):1–11. https://doi.org/10.1117/1.JNP.14.026007
15. Retallick J, Walus K (2020) Limits of adiabatic clocking in quantum-dot cellular automata. J Appl Phys 127(5):054502. https://doi.org/10.1063/1.5135308
16. Sardinha LH, Silva DS, Vieira MA, Vieira LF, Neto OPV (2015) Tcam/cam-qca: (ternary) content addressable memory using quantum-dot cellular automata. Microelectron J 46(7):563–571. https://doi.org/10.1016/j.mejo.2015.03.020
17. Sardinha LHB, Costa AMM, Neto OPV, Vieira LFM, Vieira MAM (2013) Nanorouter: a quantum-dot cellular automata design. IEEE J Sel Areas Commun 31(12):825–834. https://doi.org/10.1109/JSAC.2013.SUP2.12130015
18. Song Z, Xie G, Cheng X, Wang L, Zhang Y (2020) An ultra-low cost multilayer ram in quantum-dot cellular automata. IEEE Trans Circ Syst II: Express Briefs pp 1–1
19. Srivastava S, Asthana A, Bhanja S, Sarkar S (2011) Qcapro - an error-power estimation tool for qca circuit design. In: ISCAS'11, pp 2377–2380
20. Walus K, Dysart T, Jullien GA, Budiman R (2004) QCADesigner: a rapid design and simulation tool for quantum-dot cellular automata. Trans Nanotechnol 3(1):26–29