Microelectronics Journal 46 (2015) 563–571

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

TCAM/CAM-QCA: (Ternary) Content Addressable Memory using Quantum-dot Cellular Automata

Luiz H.B. Sardinha, Douglas S. Silva, Marcos A.M. Vieira, Luiz F.M. Vieira, Omar P. Vilela Neto *

Computer Science Department, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

ARTICLE INFO

Article history:
Received 18 June 2014
Received in revised form
3 November 2014
Accepted 20 March 2015

Keywords:
Quantum-dot Cellular Automata
(Ternary) Content Addressable Memory
Nanocomputing
Memory

ABSTRACT

This paper describes a Content Addressable Memory (CAM) architecture and its ternary variant called Ternary Content Addressable Memory (TCAM) using the Quantum-dot Cellular Automata (QCA). QCA is an alternative to the current integrated circuit (CMOS) paradigm based on the characteristics of confinement and mutual repulsion between electrons. It is expected to run with clocks in high frequency (in THz order), in nanometers scale and with very low energy consumption. First, this work presents the basic building blocks (1-bit memory cell, array of memory cells, ternary memory line and encoder). Then, we describe the complete TCAM and CAM architectures. Finally, the proposed architectures are tested and validated using QCADesigner simulator, attesting their functionalities. If QCA consolidates as a possible CMOS substitute, this study can impact the design of future components that uses TCAM and CAM such as routers and switches respectively.

1. Introduction

With new advances in technology, new nanoscale computational designs are emerging such as Quantum-dot Cellular Automata (QCA). QCA is based on the confinement and mutual repulsion of electrons. QCA is a promising alternative to the current most used silicon-based paradigm since QCA might be executed in high frequency, with low power consumption, and in really small size (in the nm range) [1–3].

Content Addressable Memory (CAM) is a type of memory where the input is a data word instead of a memory address such as Random Access Memory (RAM). CAM searches for the data word to check if it is stored anywhere in it. CAM provides the position where a given word can be found. CAM is often used in network switches.

Ternary CAM (TCAM) is a variant of CAM which allows a third matching state of “don't care” for one or more bits in the stored data word. This complies with some applications such as hierarchical networks. Ternary CAMs are often used in network routers.

In this paper, we describe an architecture for the TCAM and also for CAM using the Quantum-dot Cellular Automata (QCA) paradigm.

The main contributions of our work are as follows. First, we propose and implement CAM. Second, we project and implement TCAM. Third, we demonstrate the functionality and validate the proposed architectures using QCADesigner [4].

As QCA has been recognized as one of the most promising emerging technologies, the construction of the presented architectures can be really useful for the development of network switches and routers in this paradigm.

Efficient QCA memory design is a problem that has been brought the attention of the research community. Many works such as SQUARES [5], H-Memory [6], Line-Based [7], and Tiles [8] focus on this problem. But these works apply the RAM model, where the memory device searches addresses. Our work, on the other hand, describes CAM and TCAM, which are different from RAM model since CAM and TCAM searches are based on content instead of addresses. To the best of our knowledge, we are the first to design CAM and TCAM architectures and to implement them using QCA cells.

In this work, we start by providing a background on QCA and CAM/TCAM (Section 2). We discuss the related work in Section 3. We describe our memory architecture from a bottom-up approach, by first describing the basic components, which are the memory cell, memory cell line, ternary memory cell line, and encoder (Section 4). Then, we discuss our memory architecture and its QCA design (Section 5). In Section 6 we present our simulation results based on QCADesigner simulator and we show that the two architectures work as intended. Finally, we conclude (Section 7).

2. Background

In this section we present an overview about QCA, CAM, and TCAM.

* Corresponding author.
E-mail addresses: luizhenrique@dcc.ufmg.br (L.H.B. Sardinha), douglas.sales@dcc.ufmg.br (D.S. Silva), mmvieira@dcc.ufmg.br (M.A.M. Vieira), lfvieira@dcc.ufmg.br (L.F.M. Vieira), omar@dcc.ufmg.br (O.P. Vilela Neto).

http://dx.doi.org/10.1016/j.mejo.2015.03.020
0026-2692/

2.1. QCA

QCA cells are the basic units of QCA circuit and they are typically composed of four quantum dots located at the corners of a square. A dot, in this context, is just a region where an electric charge can be located or not. Each cell has two free and mobile electrons which are able to tunnel between adjacent dots. Also, the cell occupancy is controlled by a back plane voltage. Tunneling to the outside of the cell is not allowed due to a high potential barrier. The Coulomb interaction between the electrons tends to locate them at opposing diagonals, as shown in Fig. 1(a). An isolated cell may be in one of the two equivalent energy states. These states are called cell polarizations $P=+1$ and $P=-1$. So, it is possible to codify binary information by considering that $P=+1$ represents the value 1 and that $P=-1$ represents the value 0.

When two cells are placed near each other, the polarization of one cell will influence the polarization of the other cell. This feature is shown in Fig. 1(b). Following the same rule, a wire can be built by placing several QCA cells in a row, as shown in Fig. 1(b).

QCA logic devices are designed by selecting the placement of QCA cells in a way that leverages the interaction between them. Fig. 2 depicts two fundamental QCA gates, inverter and majority gates.

In order to build more complex QCA devices, one not only needs to carefully select the placement of QCA cells but also needs to synchronize the information, avoiding having a signal reaching a logic gate and propagating before the other inputs reach the gate. This characteristic is extremely important in QCA circuits, guaranteeing its correct operation. This feature is achieved by QCA clock. The clock is an electrical field which controls the tunneling barriers within a cell, thus keeping control when a cell might or might not be polarized [3]. The clock can be applied to groups of cells (clock zones). In each zone, a single potential can modulate the barriers between the dots. The scheme of clock zones permits a cluster of QCA cells to make a certain calculation and then have its states frozen, and, finally, have its outputs used as inputs to the next clock zone.

In QCA, the clock has four different phases [3,9]. QCA clocking can also be used to ensure proper switching of QCA arrays. Moreover, the duplex nature (symmetric behavior) of QCA is avoided, ensuring that the signal does not go back to the input during its propagation along the wires and across the logic gates. This characteristic is extremely important in QCA circuits, guaranteeing its correct operation.

2.2. CAM and TCAM

CAM is a type of memory that, different from the Random Access Memory (*RAM*), it searches based on content instead of addresses [10]. RAM returns the data which is stored in the given address. CAM, however, receives the data as input and returns where the data could be found within.

CAM is useful for many applications which need speedy searches such as Hought transformations, Huffman codification, Lempel–Ziv compression, and network switches to map MAC address into IP address and vice versa. CAM is most useful for building tables that search on exact matches such as MAC address tables.

TCAM is a special type of CAM which implements the ternary state (also known as "*don't care*"). TCAM can match a third state, which is any value. This feature could cause redundancy and/or multiple correspondences. TCAM is most useful for building tables for searching on longest matches such as IP routing tables organized by IP prefixes. To reduce latency and to make the communication faster, routers use TCAM.

Fig. 3 shows an example of searching a TCAM. The input is 01011. TCAM stores values with bits 0, 1, and "*don't care*" (represented as X). In the example, the input matches two memory lines (colored with gray background). In case of more than one matching, the line with higher number should be outputted. The encoder outputs 10 (third line), which is the line that has the content of the input and also has higher priority (line number).

3. Related work

The objective of the computational memory is to store data, allowing a subsequent recovery. As a nanotechnology and due to its main features, QCA is an attractive technology for the development of high density and low power memories. Moreover, QCA has been used to design different circuits and systems, making the

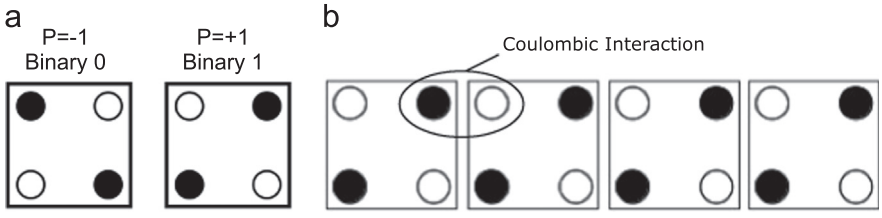

Fig. 1. (a) Possible polarizations of QCA cells with four quantum dots. Black dots represent the electrons positions. (b) Coulomb interaction between two QCA cells and a QCA wire.

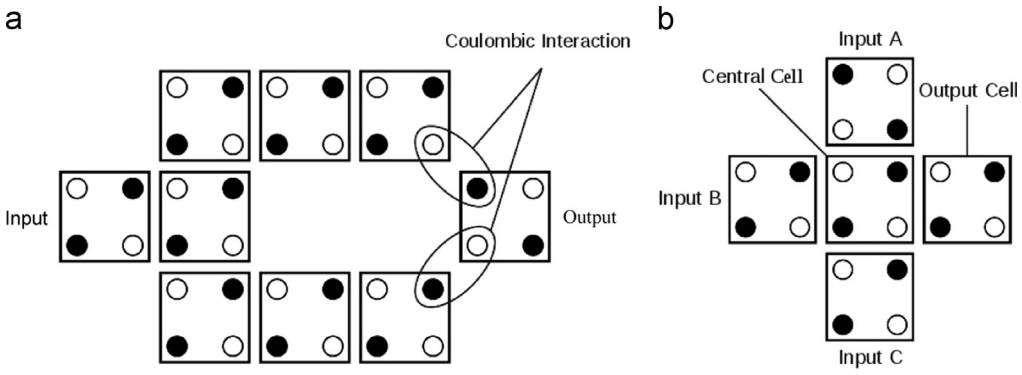

Fig. 2. Two QCA gates. (a) QCA inverter. (b) QCA majority gate.

design of memory an important part in the development of a complete computer system based in QCA.

As mentioned earlier, CAM and TCAM memories are used in important communication systems such as routers and switches, respectively. The design of a Nanorouter implemented by QCA had been recently presented [11]. This fact reinforces the importance of the development of these kinds of memories in QCA.

According to [12], sequential circuits and memory elements cannot be directly mapped into QCA using the same way of traditional CMOS technology. They also classify the memory architectures in two distinguished types, serial and parallel architectures, depending on how to access the memory.

The authors in [5] first presented an addressable S-RAM cell and later a memory system based on addressable shift registers using the SQUARES formalism. SQUARES is an initial attempt at standardizing the process of QCA circuit design and simplifying the engineering layout rules using pre-determined sized blocks (5×5 in that case). It is possible to build complex circuits sticking many of these blocks together. However, the use of SQUARES for big circuits is not recommended as it presents an extra cost both in terms of access time and spatial redundancy.

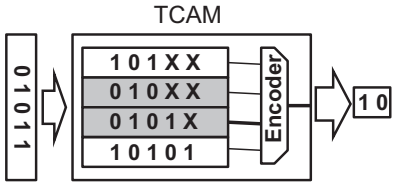

Fig. 3. TCAM example.

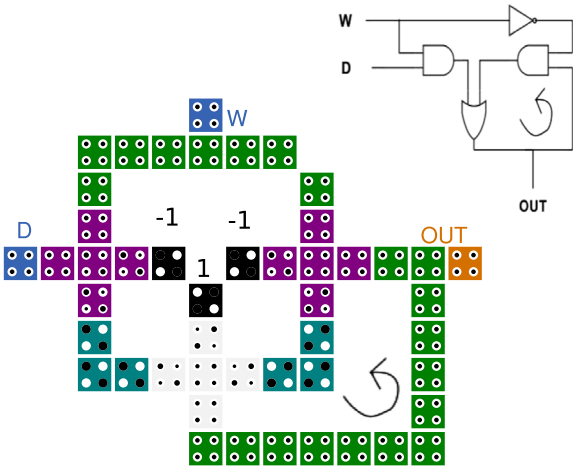

Fig. 4. 1-Bit memory cell.

Frost et al. [6] proposed a new QCA memory architecture, called H-Memory, made of many small spirals, each containing a word, and arranged in a recursive structure. This memory is a realization of a complete binary tree, where each leaf contains a memory cell and each node has a routing circuit. Read and write memory accesses are serial, taking constant time to reach any word. The nature of H-Memory design takes full advantage of the processing-in-wire aspect of QCA, allowing new possibilities for optimization, use as a cache, and integration with logic elements.

Walus et al. [13] presented a layout of a conventional Random Access Memory (RAM) architecture using QCA, showing the layout of individual memory cells, as well as, a suggested layout for a 1×4 RAM. The parallel memory architecture is based on a simple 2D grid layout of memory cells. The row of memory cells is addressed using a QCA decoder.

A Line-Based parallel memory was proposed later for QCA implementation [7,12]. This architecture is based on storing information on a QCA line by changing the direction of signal

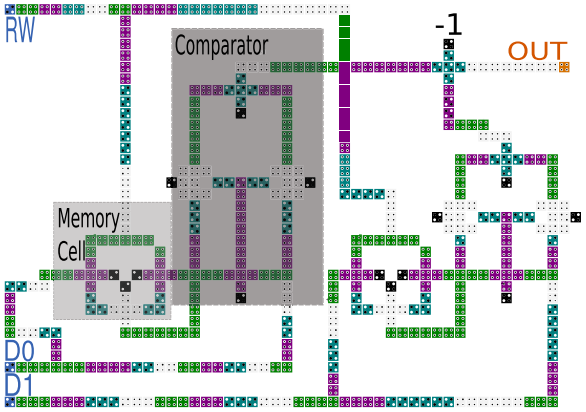

Fig. 5. 2-Bit QCA memory line layout.

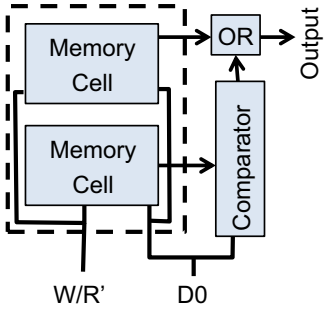

Fig. 6. Ternary memory line schematic.

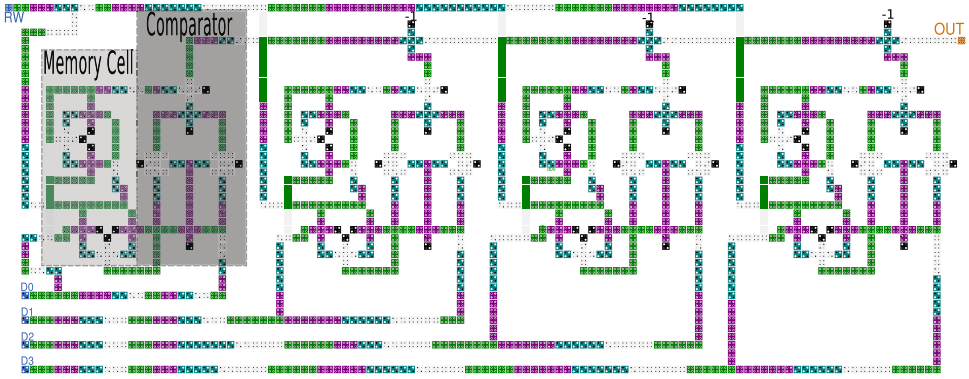

Fig. 7. 4-Bit QCA ternary memory line layout.

Table 1
Encoder truth table.

Input				Output		
O0	O1	O2	O3	FLAG	BIT1	BIT0
0	0	0	0	0	X	X
0	0	0	1	1	1	1
0	0	1	0	1	1	0
0	0	1	1	1	1	1
0	1	0	0	1	0	1
0	1	0	1	1	1	1
0	1	1	0	1	1	0
0	1	1	1	1	1	1
1	0	0	0	1	0	0
1	0	0	1	1	1	1
1	0	1	0	1	1	0
1	0	1	1	1	1	1
1	1	0	0	1	0	1
1	1	0	1	1	1	1
1	1	1	0	1	1	0
1	1	1	1	1	1	1

flow among three clocking zones. This line-based arrangement results in substantial savings in the number of zones and underlying circuitry's complexity for clocking the QCA memory. Still, this new architecture requires two additional clocking signals as it needs three zones and a four-step process whose timing is different from conventional adiabatic switching. The same group has proposed a new serial memory architecture for QCA, using basic building blocks known as tiles [8]. These tiles are used in the memory cell to construct a loop, as well as input/output capabilities for the Read/Write operations. The proposed serial memory uses different tiles to allow bidirectional signal propagation.

An improved line-based memory was proposed in [14]. This new architecture of the memory cell requires only two new clocking zones and utilizes two parallel clock zones, allowing less CMOS circuity for clock design and denser QCA implementations. Moreover, read throughput is improved to one operation per clock cycle.

In [15], a serial write and parallel read memory with few QCA cells was used as a Configurable Logic Block (CLB), an integral part of SRAM-based FPGA architecture.

Dehkordi et al. have proposed two improved structures for a loop-based RAM cell [16]. The first alternative presents smaller number of cells and wasted area has been reduced compared to traditional loop-based RAM cells. The second has duplicated the memory access time in the presence of smaller number of cells. The proposed method was validated and compared to the traditional loop-based cell in terms of speed and complexity, showing improvements.

Authors in [17] presented a novel 16-bit RAM architecture implemented with QCA. Finally, [18] presented a QCA memory cell based on two new clocking signal schemes. Thus the memory cell presents the four conventional clocking zones plus two new zones used to process writing and reading.

The paper [19] presented an efficient D Flip Flop and memory designs with QCA by applying a robust 2:1 multiplexer. In a recent paper [20], the authors presented new designs of Flip Flops and RAM with minimum number of cells by using a new wiring approach.

Unlike previous described works which utilize the RAM model, where memories search addresses, our work describes CAM and TCAM, which are different from RAM since they search based on content instead of addresses.

4. Basic components

In this section we describe the design of the basic components needed to build the full TCAM and CAM using QCA cells.

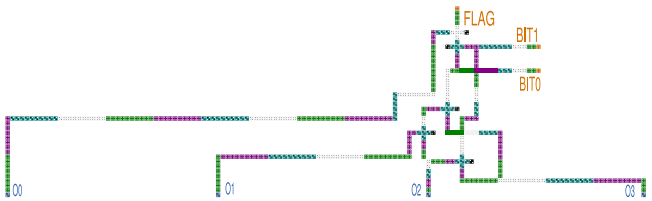

Fig. 8. QCA encoder layout.

Fig. 9. TCAM/CAM architecture schematic.

In all implementations, we have decided to apply the multi-layers approach in wire crossing, but the circuits can be easily modified to the coplanar wire crossing technique if desired. In both cases, some care must be taken with the clock zones at intersections to prevent the propagation of noise.

4.1. 1-Bit memory cell

We propose a cell architecture based on data loop (similar to the idea proposed by [13]) to build a 1-bit cell.

The cell has two inputs and one output. The input is the write/read ($W/\overline{R}$) signal and indicates if the request is to write or to read from memory. The input D is the bit to be stored in case of a write operation. The content of a given cell can be read at any moment, which is provided at the output O. Fig. 4 shows the circuit for the 1-bit memory cell using QCA cells.

4.2. Memory line

A memory line is a set of 1-bit memory cells connected serially. The memory line is also responsible for comparing the stored data with the input. It has two sets of inputs. The first input is the write/read ($W/\overline{R}$) signal that indicates if the request is to write or to read from memory. The second input is a set of input data σ. The memory line has one output signal O.

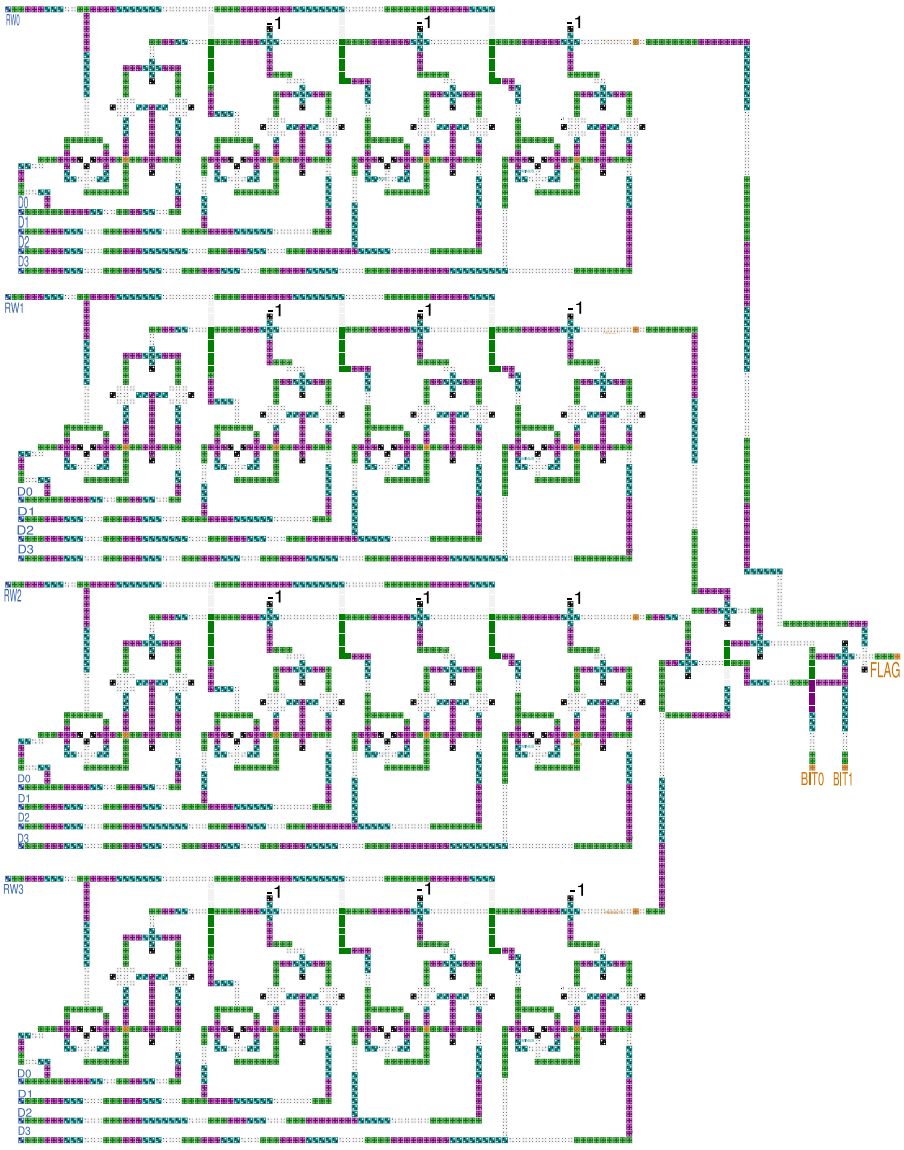

Fig. 10. QCA CAM architecture.

For a write command, the $(W/\overline{R})$ signal is set. The input data σ is stored in the memory cell.

For a search command, the $(W/\overline{R})$ signal is not set. The circuit receives as input n-bit data σ and compares σ with the stored data in its internal n-bit memory cells. The output O signal indicates if the input σ matches exactly the memory content.

To compare the input with the stored data, the n-bit memory line has n 1-bit comparators. Each comparator is implemented as the negation of the exclusive or (XNOR) gate. The comparator results are aggregated using AND gates implemented using majority gates. The result is forwarded to the output O.

Fig. 5 shows the proposed architecture of a 2-bit memory line using QCA cells. It has two 1-bit memory cells and two comparators. One memory cell and one comparator are indicated with different backgrounds. The two inputs are labeled D0–D1.

4.3. Ternary memory line

The ternary state is implemented by adding another 1-bit memory cell for each 1-bit memory cell in the memory line. The new added 1-bit memory cell stores the relevancy of previous 1-bit memory cell. So, the data stored on these new cells indicate if the content of other cells should be used for a match. A bit stored as 1 indicates that it should not be used for matching since it is a "*don't care*".

Fig. 6 depicts a ternary memory line schematic. Fig. 7 shows the layout for a 4-bit ternary memory line using QCA cells. This circuit is very similar to the CAM (shown in Fig. 5). The difference is that each 1-bit memory cell has another 1-bit memory cell. The search command operates in the same way as the CAM circuit. The write command has one difference to the CAM circuit. It needs to record the ternary state. So, the write command requires two consecutive data to the input. The first input bit is the value to be stored and the second input bit indicates whether the first bit is relevant (ternary state) to the comparison or not.

4.4. Encoder

After performing the comparisons, the circuit needs to map the results into a row number. The encoder does this mapping. It has two output signals: *FLAG*, which indicates if the search was successful, and *BIT* which indicates, in binary notation, the row where the data was found.

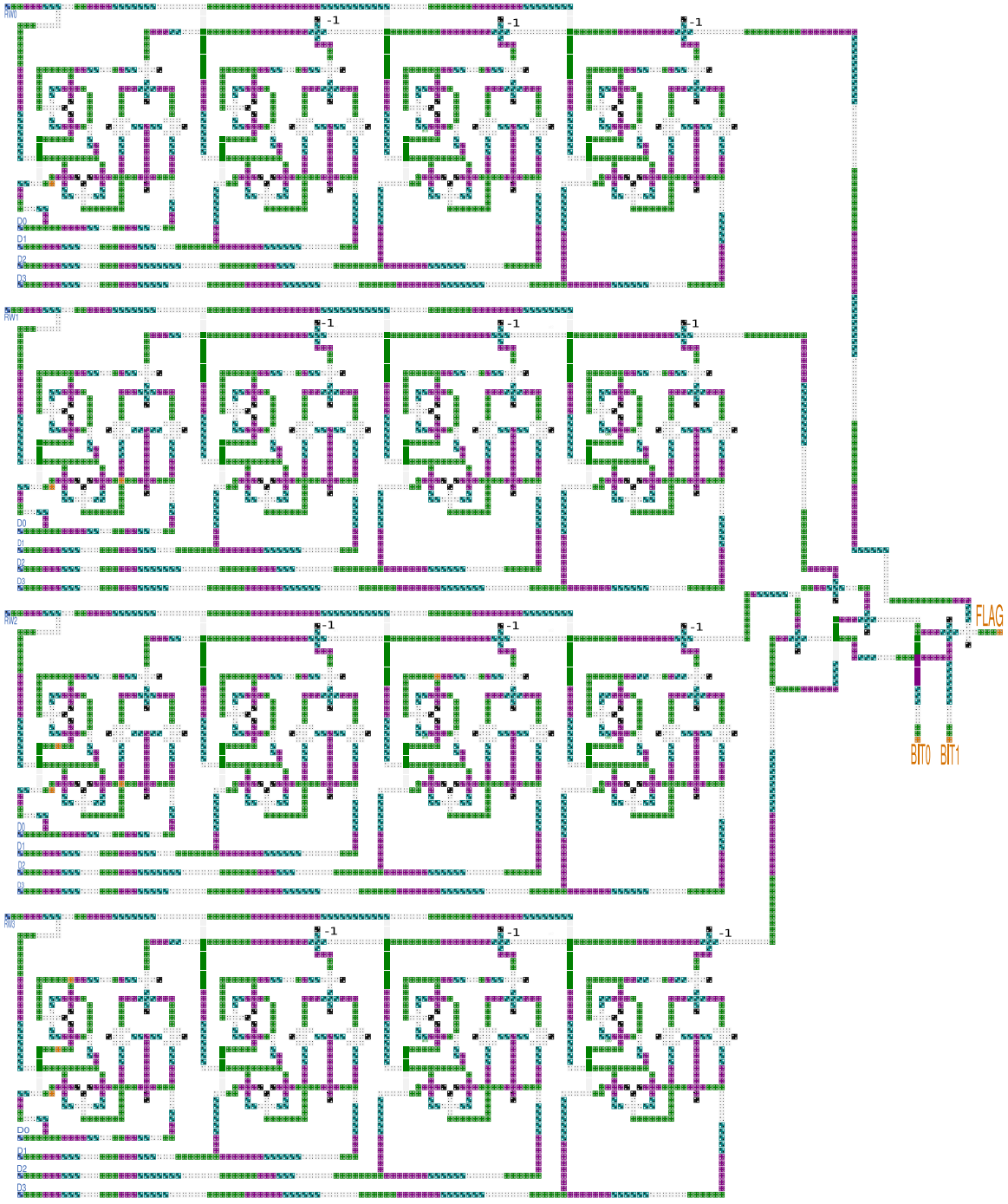

Fig. 11. QCA TCAM architecture.

Table 1 shows the truth table for the encoder. Note that *FLAG* is active if any of the inputs is also active. Fig. 8 depicts the circuit implementation.

The encoder computes the outputs using the following equations:

$$Bit_1 = O_3 \vee O_2$$

$$Bit_0 = O_3 \vee (O_1 \wedge \overline{O_2})$$

$$Flag = Bit_0 \vee Bit_1 \vee O_0$$

When more than one line matches the input data, it is the encoder that provides the priority to break ties. In our implementation, the line with larger number has higher priority. This information is encoded in Table 1.

5. Architecture

This section describes the proposed architecture of the CAM and TCAM. First we describe the circuit schematic and then the architecture using QCA cells.

5.1. Circuit schematic

As we described earlier, the CAM circuit receives the read/write signal (marked as R'/W) and the data input (marked as σ). The circuit gives as output (marked as *OUT*) the representation of the line index where the data have been found and/or the feedback signal for writing. Fig. 9 shows the schematic of the proposed

Table 2
QCADesigner simulation settings.

Parameter	Description	Value
Cell width	Width of each QCA square	18 nm
Cell height	Height of each QCA square	18 nm
Dot diameter	Diameter of each dot in a QCA cell	5 nm
Number of samples	Number of tested data during the simulation. Accuracy depends on this parameter	12,800 and 50,000
Convergence tolerance	Simulation of each sample iterates until the new value of polarization deviates from the old value by more than this predefined error limit	0.001
Radius of effect	Radius of effect of a cell is the radius at which it will interact with other cells	80 nm
Relative permittivity	Relation of the permittivity of fabrication material (GaAs/AlGaAs) to the vacuum permittivity	12.9
Clock high	Saturation of energy of clock signal when it is high	9.8E−22 J
Clock low	Saturation of energy of clock signal when it is low	3.8E−23 J
Clock amplitude factor	To make and effective clock, top 25% and bottom 25% of a single signal is dismissed	2
Layer separation	Distance between two layers	11.5 nm
Maximum iterations per sample	When the simulation of each state is not convergence based on this parameter, it automatically goes to the next state	100

architecture. It shows a four line 4-bit TCAM/CAM architecture for simplicity.

5.2. QCA architecture

In this section we describe the full architectures of the Content Addressable memory (CAM) and its ternary variant (TCAM) using QCA.

5.2.1. CAM

Fig. 10 shows the full proposed architecture of the CAM using QCA cells.

This architecture implements two commands:

1. *Search operation*: The searched data must be in the data inputs and the output will be the binary representation of the largest row index number where the data were found.
2. *Write operation*: The data which will be written should be in the data inputs and the write flag for a given row must be active. The feedback for the completion of this operation is equivalent to a match in that row.

5.2.2. TCAM

Fig. 11 shows the full proposed architecture of the TCAM using QCA cells. These architecture operations are similar to the CAM. Searches occur in the same way but write operations are different, as they need to record the ternary state. In this implementation there are four memory lines and the encoder is at the joint of them, on the right of Fig. 11.

6. Simulation results

In this section we present and discuss the results obtained simulating the proposed architectures using QCADesigner using the Coherence Vector simulation engine [4]. Table 2 brings the simulation parameters values used in the referred simulator. The results are presented after testing the memory architectures exhaustively. We describe a few test examples and their results showing the correct behavior.

6.1. Blocks simulation

The purpose of such simulations is to prove the correct functionality of the proposed blocks presented earlier. We have conducted exhaustive simulations for the memory cell, memory line with comparator and encoder and all the results were as expected. In order to save space these results are not presented here but they are essential to the correct operation of the proposed memories, whose results are presented ahead.

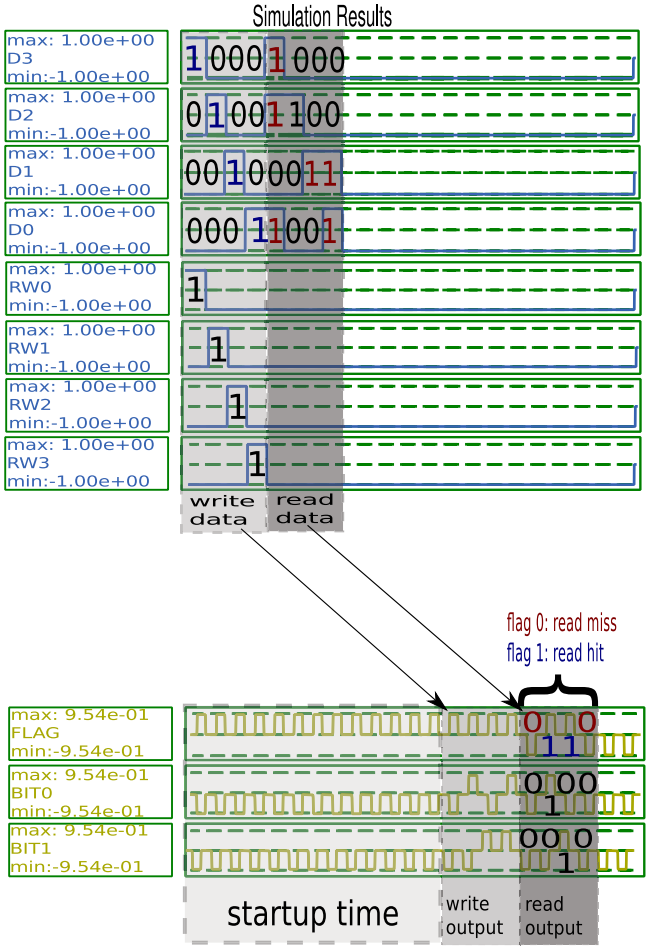

Fig. 12. Simulation results for the CAM example running.

6.2. Full architectures simulation

Here we present one example result for each developed full architecture (CAM and TCAM).

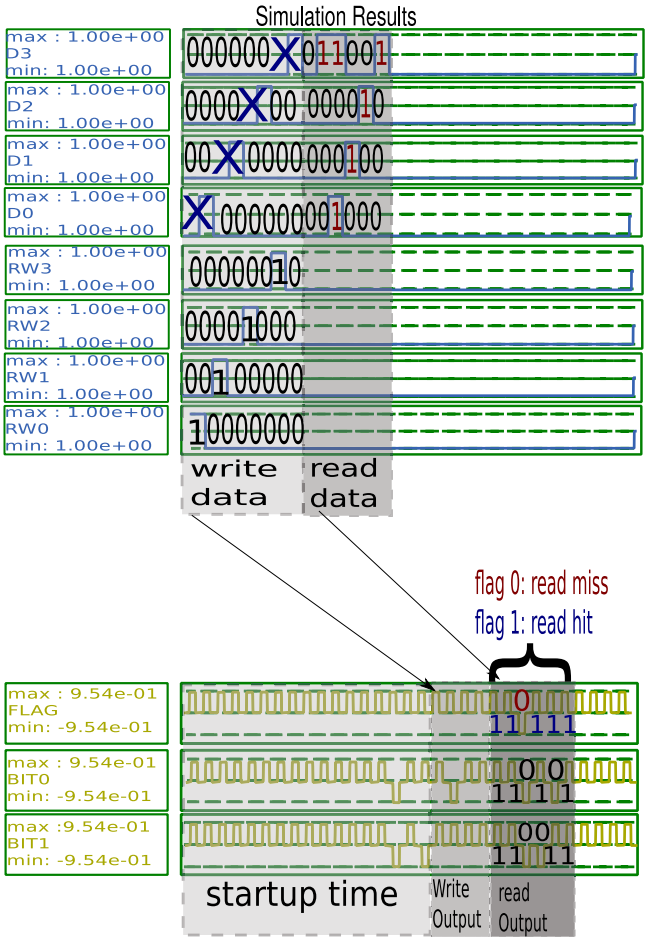

Fig. 13. Simulation results for the TCAM example running.

6.2.1. CAM

The example consists of initially writing data in the memory and then making searches on it. The simulation results are shown in Fig. 12. The results were as expected.

First, memory lines 0, 1, 2, and 3 receive respectively 0001, 0010, 0100, and 1000. Then, the system searches for 1011 (miss), 0010 (hit at line 1), 0100 (hit at line 2), and 1100 (miss). Observe that the flag is equal to zero when no data is found (miss).

6.2.2. TCAM

This example also consists of initially writing data in the memory and then making searches on it. The difference on this test is that the written data also contains the *don't care* possibility. The simulation results are shown in Fig. 13. The results were as expected and do respect the preferential rule for multiple matches (in that case, the bigger number is outputted). The writes and matches are again in the same color for best visualization.

First, memory line 0 receives X000 as input. As explained before, the write process in the TCAM takes two steps. In the first step, the memory line receives 0000. In the second step, the secondary memory cells of the same line receive 1000, indicating that the first bit is a "don't care". Then, memory lines 1, 2 and 3 receive respectively 0X00, 00X0, and 000X. Later, the system searches for 0000 (hit at line 3), 0001 (hit at line 3), 1001 (miss), 0100 (hit at line 1), 0010 (hit at line 2), and 0001 (hit at line 3). Observe that when no data is found (miss) the flag is equal to zero.

Table 3 summarizes the implementation results for the CAM and TCAM architectures (Figs. 10 and 11 respectively). To complete the data flow between the data input and output, the circuits require 17 and 21 complete clock cycles to the CAM and TCAM respectively.

Table 3
CAM and TCAM QCADesigner implementation results.

Circuitry Features	CAM	TCAM
Number of cells	4592	6419
Number of data input lines	4	4
Clock cycles	17	21

7. Conclusion

In this work, we proposed and implemented a Content Addressable Memory (CAM) and its Ternary variant (TCAM) using the Quantum-dot Cellular Automata (QCA). Due to the potential of QCA reaching high clocking rate, this novel architecture allows for higher speed memory searches. In a bottom-up approach, we first presented the basic components of the CAM and TCAM. We implemented these two circuits, demonstrated their functionalities and validated the architectures using the QCADesigner.

For future work, we intend to incorporate CAM and TCAM into other computational components. It also important to investigate how to construct these QCA circuits with physical devices.

Acknowledgments

We would like to thank INCT-DISSE MCTI/CNPq, CNPq, PRPq-UFMG and Fapemig.

References

[1] C. Lent, P. Tougaw, W. Porod, G. Bernstein, Quantum cellular automata, Nanotechnology 4 (1993) 49–57.
[2] P. Tougaw, C. Lent, Logical devices implemented using quantum cellular automata, J. Appl. Phys. 75 (3) (1994) 1818–1825.
[3] C. Lent, P. Tougaw, A device architecture for computing with quantum dots, Proc. IEEE 85 (4) (1997) 541–557.
[4] K. Walus, T. Dysart, G. Jullien, R. Budiman, Qcadesigner: a rapid design and simulation tool for quantum-dot cellular automata, IEEE Trans. Nanotechnol. 3 (1) (2004) 26–31, URL ⟨http://www.mina.ubc.ca/qcadesigner⟩.
[5] D. Berzon, T.J. Fountain, A memory design in QCAs using the squares formalism, in: Proceedings of the Ninth Great Lakes Symposium on VLSI, 1999, IEEE, Ann Arbor, Michigan, 1999, pp. 166–169.
[6] S.E. Frost, A.F. Rodrigues, A.W. Janiszewski, R.T. Rausch, P.M. Kogge, Memory in motion: a study of storage structures in QCA, in: The First Workshop on Non-Silicon Computing, vol. 2, 2002.
[7] M. Ottavi, V. Vankamamidi, F. Lombardi, S. Pontarelli, Novel memory designs for QCA implementation, in: The Fifth IEEE Conference on Nanotechnology, 2005, IEEE, Nagoya, Japan, 2005, pp. 545–548.
[8] V. Vankamamidi, M. Ottavi, F. Lombardi, A serial memory by quantum-dot cellular automata (QCA), IEEE Trans. Comput. 57 (5) (2008) 606–618.
[9] K. Walus, G. Jullien, Design tools for an emerging SOC technology: quantum-dot cellular automata, Proc. IEEE 94 (6) (2006) 1225–1244.
[10] K. Pagiamtzis, A. Sheikholeslami, Content-addressable memory (cam) circuits and architectures: a tutorial and survey, IEEE J. Solid-State Circuits 41 (3) (2006) 712–727.
[11] L.H.B. Sardinha, A.M.M. Costa, O.P. Vilela Neto, L.F.M. Vieira, M.A.M. Vieira, Nanorouter: a quantum-dot cellular automata design, IEEE J. Sel. Areas Commun. 31 (12) (2013) 825–834.
[12] V. Vankamamidi, M. Ottavi, F. Lombardi, A line-based parallel memory for QCA implementation, IEEE Trans. Nanotechnol. 4 (6) (2005) 690–698.
[13] K. Walus, A. Vetteth, G. Jullien, V. Dimitrov, Ram design using quantum-dot cellular automata, in: NanoTechnology Conference, vol. 2, 2003, pp. 160–163.
[14] B. Taskin, B. Hong, Improving line-based QCA memory cell design through dual phase clocking, IEEE Trans. Very Large Scale Integr. Syst. 16 (12) (2008) 1648–1656.
[15] M. Niamat, S. Panuganti, T. Raviraj, QCA design and implementation of sram based FPGA configurable logic block, in: 2010 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE, Seattle, Washington, 2010, pp. 837–840.

[16] M.A. Dehkordi, A.S. Shamsabadi, B.S. Ghahfarokhi, A. Vafaei, Novel ram cell designs based on inherent capabilities of quantum-dot cellular automata, Microelectron. J. 42 (5) (2011) 701–708.
[17] M. Kianpour, R. Sabbaghi-Nadooshan, A novel design and simulation of 16 bits ram implementation in quantum-dot cellular automata (QCA), in: 2012 16th IEEE Mediterranean Electrotechnical Conference (MELECON), IEEE, Medina Yasmine Hammamet, Tunisia, 2012, pp. 637–640.
[18] X. Yang, L. Cai, H. Huang, X. Zhao, A comparative analysis and design of quantum-dot cellular automata memory cell architecture, Int. J. Circuit Theory Appl. 40 (1) (2012) 93–103.
[19] S. Hashemi, K. Navi, New robust QCA D flip flop and memory structures, Microelectron. J. 43 (12) (2012) 929–940.
[20] S. Angizi, K. Navi, S. Sayedsalehi, A.H. Navin, Efficient quantum dot cellular automata memory architectures based on the new wiring approach, J. Comput. Theor. Nanosci. 11 (11) (2014) 2318–2328.