Field-coupled Nanocomputing Energy Analysis Tool

Marco Antonio Ribeiro*, Jeferson Figueiredo Chaves*†, Omar Paranaiba Vilela Neto*
*Department of Computer Science
Federal University of Minas Gerais, Belo Horizonte, Minas Gerais
Email: [marcoantonio, jefchaves, omar]@dcc.ufmg.br
†Department of Computer Engineering
Federal Center of Technological Education of Minas Gerais, Belo Horizonte, Minas Gerais

***Abstract*—Field-coupled nanocomputing (FCN) devices are expected to reach ultralow power consumption, thus the energetic losses emerging from the irreversible erasure of information becomes significant. In this case, the development of tools for determining the lower bounds related to those losses are quite important. In this paper we present our tool and its algorithms that quantify these information losses. Our software reads a FCN layout design, recognize the logic gates based on a standard cell library, builds a graph representing its netlist and then calculate the energy losses accordingly to two different methods found in the literature. To the best of our knowledge, this work presents the first attempt to automate the aforementioned calculations.**

I. Introduction

Energy efficiency is an important issue to consider when it comes to digital systems development. Despite all the accomplishments in the CMOS evolution, there has been an active search for its replacement [1]. Despite the fact that there is no mature technology available yet, many new devices are being considered as replacement for CMOS transistors.

Field-coupled nanocomputing (FCN) devices are one of the research topics likely to be used in the design of new hardware components. Information transfer and computation are achieved in FCN via local field interactions between nanoscale building blocks that are organized in patterned arrays. Several FCN paradigms are currently under active investigation, including quantum-dot cellular automata (QCA), molecular quantum cellular automata (MQCA) and nanomagnetic logic (NML) [2].

Although most of these emerging technologies present very low power consumption, they are energetically bounded by a rigid thermodynamic limit. In 1961, Rolf Landauer argued that any irreversible computational process, i.e., those where information is lost, results in the loss of $kTln(2)$ joules per bit erased, where k is the Boltzmann constant and T is the temperature [3]. The validity of this thermodynamic limit in computation, also know as Landauer's principle, has been disputed since its proposal. Only recently, in 2012, Landauer's principle was experimentally verified, confirming that there is a physical limit in irreversible computation [4].

Field-coupled nanocomputing devices are expected to reach ultralow power consumption, thus the energetic losses emerging from the irreversible erasure of information becomes significant. In this case, the development of tools for determining the lower bounds related to those losses are quite important. In this paper we present our tool and its algorithms that quantify these information losses. The software operates on a circuit layout design in order to discover its logic and then do its calculations. Quantum-dot cellular automata (QCA) has the most mature CAD tools when compared against other FCN technologies. That is the reason why we choose to support first QCA circuits, despite our tool ability to operate on any other FCN circuit with small changes that will be clear in section IV.

The remaining of this paper is organized as follows: In section II we present an introduction to QCA. The related work is presented in section III. Then our tool and its algorithms are presented and discussed in section IV. Finally, section V concludes the paper and presents some relevant extensions of this work.

II. Quantum-dot Cellular Automata

The QCA technology basic cells 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, a back plane voltage controls the cell occupancy. 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 two equivalent energy states. 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.

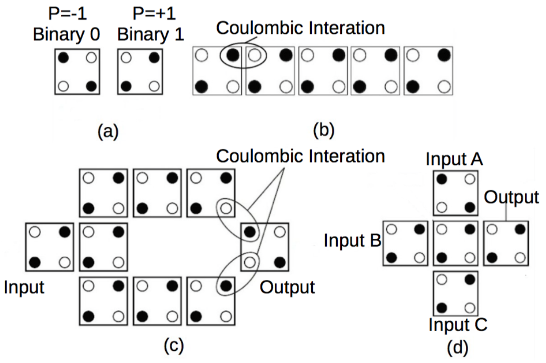

Fig. 1: (a) Possible polarizations of QCA cells with four quantum dots. Black dots represent the electrons positions. (b) A QCA wire. (c) A QCA inverter. (d) A QCA Majority Gate.

When two cells are nearby, the polarization of one cell will influence the polarization of the other cell. In this case, the two possible polarization states of the second cell will not be equivalent. For example, consider that a cell (cell 1) has its polarization fixed at P1 = +1 and it is placed next to a second cell (cell 2). The distribution of charges of cell 2 is influenced by the distribution of charges in cell 1, which is then responsible for the polarization of cell 2 (P2). So, cell 2 tends to have the same polarization as cell 1, reducing the Coulomb interaction between all the electrons involved. Following the same rule, a wire can be built by placing several QCA cells in a row, as shown in Fig. 1 (b).

The two fundamental QCA gates, inverter and majority gates, are presented and explained in details. When cells are placed diagonally to each other, they tend to have reverse polarizations due to the repulsion between electrons. This feature can be explored to build an inverter, such as the one shown in Fig. 1 (c).

The majority gate (Fig. 1 (d)) is the most important basic QCA logic device as it can be used to build AND and OR gates, besides being used to build more complex devices. The device cell at the center of the gate has its lowest energy when it assumes the polarization of the majority of the three input cells because this is the configuration where the repulsion between the electrons in the three inputs cells and the electrons in the device cell is the smallest. Observe in Fig. 1 (d) that, even though input cell A has the polarization that represents binary 0, the output cell has the same polarization as cells B and C, which are the majority in this case. Also, if input cell A is always fixed at binary 0, an AND gate with two inputs (B and C) is defined. In the same way, if the same cell A is always fixed at binary 1, an OR gate is formed. With ANDs, ORs, and inverters, any logic function can be implemented. So, any computational circuit can be fulfilled with QCA paradigm.

In order to build more complex QCA devices, one not only needs to select carefully the placement of QCA cells but also needs to synchronize the information to avoid 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 [5]. 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 its states are frozen and its outputs can be used as inputs to the next clock zone.

III. Related Work

QCA technology is expected to reach ultralow power consumption, thus the energetic losses emerging from the irreversible erasure of information becomes significant. In this case, the development of methods determining the lower bounds related to those losses are quite important.

One solution to this problem was proposed by Hanninem [6]. In his research he analyzed the information losses in adders based on Shannon's information theory framework. This work shows that the information loss depends on the level of abstraction in which the system is considered. In the monolithic case, i.e., the one analyzed only for its input-output relations without defining the subsystems that compose them, only a theoretical lower limit for the lost energy is provided. As the level of abstraction lowers, one can observe that energetic losses rises. Not only the change of levels but also the type of logic element used as the basic module of the system determine and influence more realistically the lower bound for the energetic consumption.

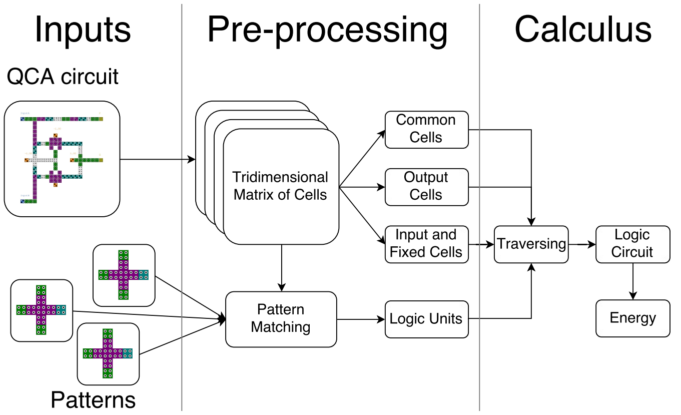

Fig. 2: Algorithm overview.

Ercan took step further towards the development of a framework with the same objective [7]. Besides the elements of Shannon's information theory, their method contemplates a thermodynamics framework which provides a better understanding of the link between the logic and the physical aspect of the devices.

Srivastava and collaborators proposed a probabilistic modeling tool to estimate polarization error and an upper bound of non-adiabatic switching power loss in QCA circuits [8]. Although the concerns about energy, their work focus on a rare and specific case, the non-adiabatic switching.

However, to the best of our knowledge, this work presents the first attempt to create a tool that automatically calculate a lower bound of energy losses related to information erasure in FCN circuits.

IV. The Tool

The tool calculates the energy based in the information losses accordingly two methods [6], [7]. In order to do that, it starts reading a layout circuit design made by QCADesigner [9] and files with patterns of logic gates. Then it recognizes the logic gates via a template matching, marking the found logic gates inputs with its provable values (considering all possible input combinations in all system's variable inputs) and finally follows the information flow in order to maintain only the possible values (we called this process ”traversing”). The main structure of the implementation is showed in Figure 2 and it is described in the following subsections.

A. Inputs

The QCADesigner uses a semi-structured file to store the projects developed in that tool, dividing the circuit in layers.

The Substract layer contains the information of the size of the cells used by the file and other information of the subtract (position, size). The Level layers contain the Cells Layers by circuit level from the highest to the lowest and, finally, the Cells layers contain the cells blocks. Each cell block contains the physical information of the cell in the circuit and the type of the cell.

Other inputs are the *.lunit* files which describe the Logic Unit by showing its logic expression and patterns containing cell offset and type. These files behave like a standard cell library and can be improved by the user with new patterns.

B. Pre-processing

The algorithm begins by converting the QCA circuit to a tridimensional matrix. Initially the default cell size is obtained from the substract layer. Then for every level layer the cells are extracted with their clock zone number, position, polarization and type. These types can vary from Input cells, Fixed polarized cells (treated as Input), Common cells (used as propagation medium for the information) and Output cells. These cells are inserted in the matrix according to their position discretized in a way that adjacent cell's index differs only in one unit. This part is the only one that must be replaced in order support other FCN technology. The parser that reads the circuit design and extract the information needed to create the tridimensional matrix of cells depends on the specification format of the layout circuit file.

Each unit is scanned within the matrix by pattern matching. During this process, superposition of two found logic units can happen when a portion of their pattern matches the other. In this case, the first unit found in the traversing is enabled and the other is disabled.

Now that all the cells have been scanned, the information is propagated as flows from the inputs (varying and fixed) to create a circuit from the matrix. An important aspect in this approach is to maintain a consistent logic propagation, i.e., when two flows meet, they should merge into one with information of both.

For this to work, each flow has a list of input states containing a N conditions, where each position relates to the starting state of the N^{th} varying input, as fixed inputs cannot have more than one state. Each state consists of a set of possible values. The inputs being binary, the states can be:

Started as $false$ = $\{ \ 0 \ \}$
Started as $true$ = $\{ \ 1 \ \}$
May be $true$ or $false$ = $\{ \ 0 \ , \ 1 \ \}$

The third state occurs when an input was not found yet and, consequently, not merged with this flow. The empty set is an error state which exclude the resulting flow.

For each varying inputs we start two flows, one representing when it starts on $false$ where his position on the list is $\{ \ 0 \ \}$ and another representing when it starts on $true$ where his position on the list is $\{ \ 1 \ \}$. In both flows, all the other inputs are $\{ \ 0 \ , \ 1 \ \}$. As for the fixed inputs, they begin with a flow with a list of sets $\{ \ 0 \ , \ 1 \ \}$.

Figure 3 shows an example. The varying inputs are colored in blue (inputs 1 and 2), the fixed inputs (3 and 4) are orange and the only output is yellow. The logic units (majority gates) are inside the blue squares. The flows initialized before the traversing are also indicated in Table I and in the tables in Figure 3 (a).

TABLE I: Starting flows for the circuit in Figure 3 (a)

Origin	Input$_1$ state	Input$_2$ state	Value
Input$_1$	$\{ \ 0 \ \}$	$\{ \ 0 \ , \ 1 \ \}$	$false$
	$\{ \ 1 \ \}$	$\{ \ 0 \ , \ 1 \ \}$	$true$
Input$_2$	$\{ \ 0 \ , \ 1 \ \}$	$\{ \ 0 \ \}$	$false$
	$\{ \ 0 \ , \ 1 \ \}$	$\{ \ 1 \ \}$	$true$
Input$_3$	$\{ \ 0 \ , \ 1 \ \}$	$\{ \ 0 \ , \ 1 \ \}$	$false$
Input$_4$	$\{ \ 0 \ , \ 1 \ \}$	$\{ \ 0 \ , \ 1 \ \}$	$true$

C. Calculus

The merging occurs when two or more flows arrive at a logic unit and is done by applying the intersection of sets in each of the positions of both lists. If any of these intersections results in an empty set, the merging is ignored and these flows cannot merge. This operation is applied for every combination of flows coming from different inputs.

In Figure 3 (a) the flows from Input$_1$, Input$_2$ and Input$_3$ propagate through the cells until they reach the Majority Gate$_1$. This process results in the data state combinations presented in table $Majority\ Gate_1\ Flows$ in Figure 3 (b). $Majority\ Gate_1$ is marked as used and the flows propagate to $Majority\ Gate_2$.

In Figure 3 (b) the flows from Majority Gate$_1$ output, Input$_1$ and Input$_4$ have reached Majority Gate$_2$. Majority Gate$_2$ is marked as used and the flows propagate to the output as showed in Figure 3 (c).

After the traversing process, we will have a graph connecting all the reachable cells from an input cell, representing the circuit. Then the user can choose from two energy analysis options:

1) **By section of clock:** all pieces inside a clock zone are considered as one logic unit. Hanninem's analysis is carried out [6]. Here the information losses occurs on the purple area (purple Majority Gate and purple wire) and on the white area (white Majority Gate).
2) **By logic unit:** the analysis is applied in all valid logic unit taken in the traversing algorithm. Ercan's method is carried out [7]. Here the information losses occurs on the purple Majority Gate and on the white Majority Gate. Each one of these cells is an output of a section. The inputs are selected according to the cell before the output.
   - If the output is also an output of a valid and used logic unit, then all the inputs of the logic unit are considered inputs of this section
   - Else the input is the cell before the output (wire)

   After that, every section has its own set of inputs and outputs and is considered a logic unit.

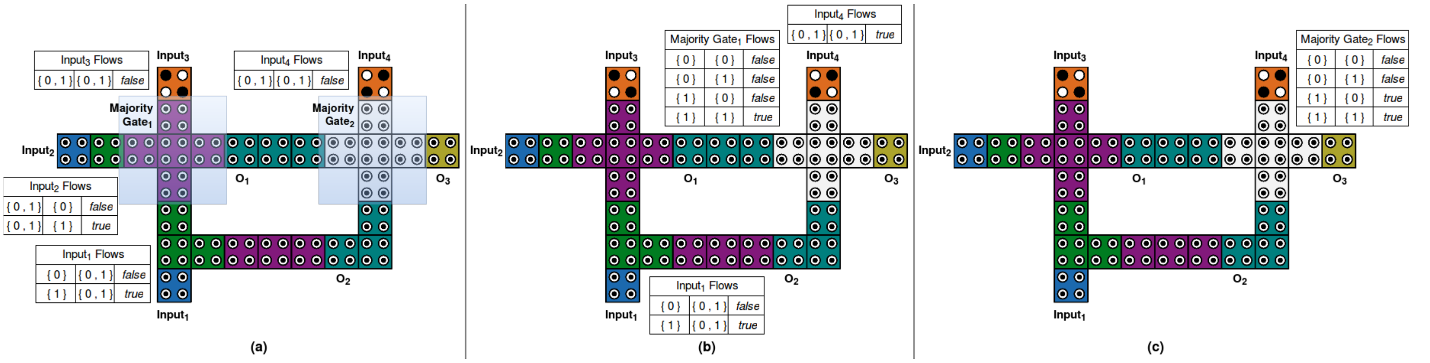

Fig. 3: Data flow in the traversing algorithm.

TABLE II: Truth-table of circuit in Figure 3

Input$_1$	Input$_2$	O$_1$	O$_2$	O$_3$
$false$	$false$	$false$	$false$	$false$
$false$	$true$	$false$	$false$	$false$
$true$	$false$	$false$	$true$	$true$
$true$	$true$	$true$	$true$	$true$

The main calculation is done from the set of all possible binary inputs and outputs of each logic unit. The energy analysis by section of clock in the purple area, for example, in Figure 3 circuit has [Input$_1$Input$_2$] as inputs and [O$_1$O$_2$] as outputs. As we can see in Table II, this section has four different input combinations with equal probabilities and three different output combinations with 2, 1 and 1 appearances respectively. In the other energy analysis, by logic unit, the information loss in the Majority Gate$_1$ results as a relation between [Input$_1$Input$_2$] as inputs and O$_1$ as output. In this case we have the same condition with the input but only two combinations in the output with 3 and 1 appearances respectively.

For every possible input i, its probability of appearance considering all the others is given by:

$$prob(i) = \frac{appearances(i)}{\sum_{n=0}^{n=total_i-1} appearances(n)} \tag{1}$$

Then, we calculate its self information

$$I(i) = -log_2(prob(i)) \tag{2}$$

And lastly its information entropy

$$H(i) = prob(i) * I(i) \tag{3}$$

The same is done for the outputs o. The dissipated energy by the logic unit is calculated from

$$\left(\sum_{n=0}^{n=total_i-1} H(i)\right) - \left(\sum_{n=0}^{n=total_o-1} H(o)\right) \tag{4}$$

To obtain the dissipated energy for the circuit itself we have to sum all dissipations in all sections or logic units. These energy calculations in our example results in 0.5 $K_BTln(2)$ J for section of clock and 1.19 $K_BTln(2)$ J for logic unit.

V. Conclusion

In this work, we proposed and implemented a tool that obtains the energy low bound for FCN circuits. Despite that our current version only works on QCA circuits, only one change is required to support any other FCN device: the parser that reads the circuit design and extract the information needed to create the tridimensional matrix of cells.

For future work, we intend to extend the support to others FCN devices and embedded these energy calculations into QCADesigner software in order to guide the design of energy efficient QCA circuits. We also envision to build a tool for Nanomagnet Logic designs incorporating the same energy calculations.

References

[1] R. K. Cavin, P. Lugli, and V. V. Zhirnov, “Science and engineering beyond moore’s law,” *Proceedings of the IEEE*, vol. 100, no. 13, pp. 1720–1749, 2012.

[2] N. G. Anderson and S. Bhanja, *Field-Coupled Nanocomputing*. Springer, 2014.

[3] R. Landauer, “Irreversibility and heat generation in the computing process,” *IBM journal of research and development*, vol. 5, no. 3, pp. 183–191, 1961.

[4] A. Bérut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, and E. Lutz, “Experimental verification of landauer/’s principle linking information and thermodynamics,” *Nature*, vol. 483, no. 7388, pp. 187–189, 2012.

[5] C. S. Lent and P. D. Tougaw, “A device architecture for computing with quantum dots,” *Proceedings of the IEEE*, vol. 85, no. 4, pp. 541–557, 1997.

[6] I. K. Hänninen, C. S. Lent, and G. L. Snider, “Quantifying irreversible information loss in digital circuits,” *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 11, no. 2, p. 10, 2014.

[7] I. Ercan and N. G. Anderson, “Heat dissipation in nanocomputing: lower bounds from physical information theory,” *Nanotechnology, IEEE Transactions on*, vol. 12, no. 6, pp. 1047–1060, 2013.

[8] S. Srivastava, A. Asthana, S. Bhanja, and S. Sarkar, “Qcapro-an error-power estimation tool for qca circuit design,” in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 2377–2380.

[9] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman, “Qcadesigner: a rapid design and simulation tool for quantum-dot cellular automata,” *Nanotechnology, IEEE Transactions on*, vol. 3, no. 1, pp. 26 – 31, march 2004.