NanoRouter: A Quantum-dot Cellular Automata Design

5 authors, including:

Luiz H B Sardinha
Federal University of Minas Gerais
6 PUBLICATIONS   **122** CITATIONS

SEE PROFILE

Artur Melo Mota Costa
Federal University of Minas Gerais
6 PUBLICATIONS   **58** CITATIONS

SEE PROFILE

Omar Vilela Neto
Federal University of Minas Gerais
86 PUBLICATIONS   **724** CITATIONS

SEE PROFILE

Luiz Vieira
Federal University of Minas Gerais
149 PUBLICATIONS   **3,192** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Internet of Mobile Things View project

Project MORELIT - Remote Monitoring of High Power Transmission lines View project

NanoRouter: A Quantum-dot Cellular Automata Design

Luiz H. B. Sardinha, Artur M. M. Costa, Omar P. Vilela Neto, Luiz F. M. Vieira, and Marcos A. M. Vieira

Abstract—We present NanoRouter, a new router architecture implemented as a quantum-dot cellular automata (QCA). A router is a key component in the Internet core. It allows packets to be transferred in the Internet. QCA is a promising nanoscale technology where components have nano size, ultra-low power consumption and could have a clock rate on the terahertz range. In a bottom-up approach, we first describe the building blocks that compose NanoRouter such as crossbar, demux and parallel-to-serial converter and then describe the full architecture. We demonstrate the functionality, test and validate the proposed architecture and provided performance evaluations of NanoRouter. This new router architecture can increase the speed of the Internet core.

Index Terms—Packet Switch, Router, Communication System, Quantum-dot Cellular Automata, Nanocomputation, Nanocommunication.

I. Introduction

With the advance of technology at the nanoscale, new applications for communication are emerging. Researching on novel nanotechnology brings the potential for the design of new components. In this work, we focus on Quantum-dot Cellular Automata (QCA). QCA is a promising nanoscale technology where a cell has nano size and ultra-low power consumption. QCA is a possible alternative to replace transistor-based technology.

A router is a key component in the Internet core. It transfers packets from incoming links to outgoing links. The most popular router architecture requires that the access rate of the memories is at least as fast as the line rate [1]. This impose a hard restriction for the design of routers to work with fast lines.

A QCA design provides advantages such as ultra-small factor, low power consumption and high-speed clock when projecting new router architectures. According to [2], QCA clock rate could be in the range of 1-2 THz. In this paper, a novel router architecture based on QCA is proposed and also implemented. In an effort to bridge the gap between QCA technology and router architecture, we present the design and implementation of a router architecture based on QCA. We envision that this new router will be part of the Future Internet since it has the potential to increase the speed of the Internet core.

The main contributions of our work are as follows. First, we proposed and implemented a new router architecture that takes advantage of the ultra-small feature size of QCA, its

ultra-low power consumption and promising high clock rate (order of terahertz). In our proposed architecture, the router core (crossbar) does not need to be configured at the speed of the data link. Second, to the best of our knowledge, we are the first to also design and implement a demultiplexer component. Third, we designed and implemented parallel-to-serial converter components that probably will be incorporated into other QCA devices. Fourth, we demonstrate the functionality and validate the proposed architecture using QCADesigner simulator [3]. Finally, we analyze and provide performance evaluations of our NanoRouter.

Many variations of routers architectures already exists, including Clos [4], Batcher-banyan [5], combined input-output queued (CIOQ) [1]. However, they do not take advantage of QCA nanotechnology. To the best of our knowledge, there are few works in the literature relating QCA within networking and communication systems, so we are the first to propose and design a router based on QCA, which could lead to a faster and less energy consumption Internet.

In this work, we start by providing an introduction to QCA and Router Architecture (Section II). We discuss the related work, comparing our approach with the state-of-the-art (Section III). We describe our router architecture from a bottom-up approach, by first describing the basic components that compose our router. We also present the parallel-to-serial component that might be important to be incorporated for other QCA project (Section IV). Then, we discuss our router architecture and its QCA design (Section V). We present our simulation results based on QCADesigner simulator (Section VI). Finally, we present our concluding thoughts (Section VII).

II. Background

A. QCA

A possible alternative to the current CMOS/VLSI circuits is a computing paradigm known as Quantum-dot Cellular Automata - QCA [6]. QCA technology consists of a group of cells which, when combined and arranged in a particular way, are able to perform computational functions. QCA technology transfers information by means of the polarization state of various cells in contrast to traditional computers, which use the flow of electrical current to transfer information [6], [7], [8]. QCA is expected to create high clock frequency (in order of THz) in addition to low-power consumption in ultra small area.

The basic units of QCA circuits are cells made of quantum dots. A dot, in this context, is just a region where an electric charge can be located or not. A QCA cell has four quantum dots located at the corners. Each cell has two free and mobile

electrons which are able to tunnel between the quantum dots. It is assumed that 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. An isolated cell may be in one of 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.

P=-1   P=+1
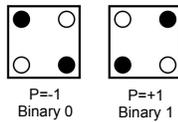Binary 0   Binary 1

Fig. 1.   Possible polarizations of QCA cells with four quantum dots. Black dots represent the electrons positions.

If a cell with fixed polarization is placed near another cell, the second cell will be influenced by the first cell and will change its polarization. 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 in cell 1 influences the distribution of charges in cell 2, 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. This feature is shown in figure 2 (a). In this way, it is possible to note that QCA cells placed in a row acts like a wire, as shown in figure 2 (b).
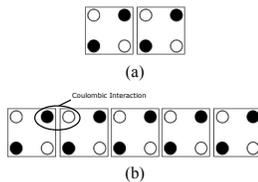

(a)

Coulombic Interaction

(b)

Fig. 2.   (a) Coulomb interaction between two QCA cells and (b) A QCA wire.

If one place the QCA cells in a way that leverages the interaction between them, one can create a QCA device with the desired logic. The two fundamental QCA gates, inverter and majority gates, are presented and explained in details following. Inverters can be implemented exploring the characteristic that when cells are placed diagonally to each other, they tend to have reverse polarizations due to the repulsion between electrons, such as the one shown in Figure 3 (a).

The basic QCA logic device is called majority gate (Fig. 3 (b)). 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. 3 (b) 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.

(a)

Input A

Central Cell   Output Cell
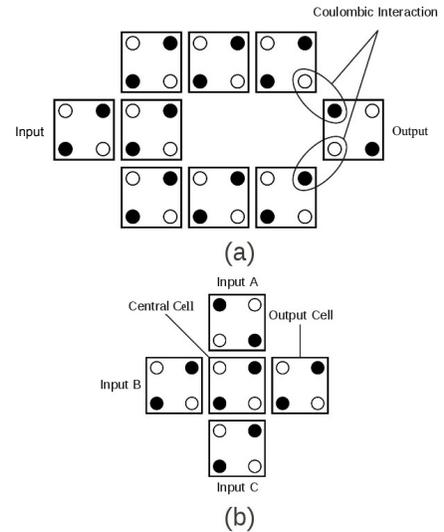
Input B

Input C

(b)

Fig. 3.   (a) A QCA inverter and (b) A QCA Majority Gate.

In QCA circuits, 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 [8]. Thus, in this case, the clock is used to synchronize the information, avoiding having a signal reaching a logic gate and propagate before the other inputs reach the gate. This characteristic is extremely important in QCA circuits, guaranteeing its correct operation. 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.

The QCA clock has four different phases [8] [9]. In the first phase, know as Switch, the QCA cells start depolarized with the tunneling potential barriers low. During this phase, the barriers between the dots are progressively increased and the cells start to polarize according to the state of their drivers (that is, their input or neighbor cells). The actual computation is made exactly at this phase. At the end of the first phase, the barriers are high enough to avoid the tunneling of any electron, so the states of the cells are fixed. During the second clock phase, known as Hold, the cells have fixed states as the barriers are kept high. So they can be used as inputs to the next stage. In the third clock phase, called Release, the barriers are lowered and the cells are allowed to relax to a depolarized state. Finally, the last clock phase is called Relax and the barriers are kept low and the cells remain depolarized.

Figure 4 shows an example of a wire with four clocking zones. In the first line, the cells in clock zone 0 is in Switch phase and the cells are allowed to polarize according to the

polarization of the input cell (Black cell). At the same time, the cells in clock zone 3 are in Hold phase due to an older signal that was transmitted by the wire. In the next clock time, clock 0 is in Hold phase and the polarizations of its cells is used as input for the cells of clock zone 1, in Switch phase. Next, the cells in clock zone 0 are depolarized (clock phase Release) and the cells of clock zone 1 are used to polarize cells of clock zone 2 and so on.
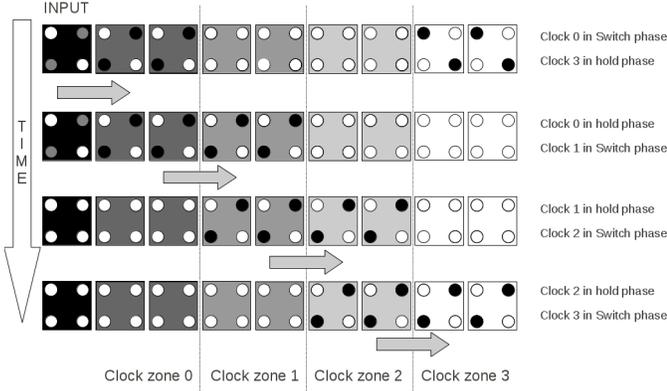

Fig. 4. An example of clocking zones controlling the polarization of cells in a wire.

In this section we have focused mainly in the logic aspects of QCA circuits needed to implement the proposed architecture. The reader interested in the quantum aspects can find more details in [8].

Despite the advancement of QCA logical circuits, the practical implementation is still a challenge. Some possible alternatives to the development of practical QCA circuits have been proposed at the literature. One of the first attempts to develop a physical implementation of QCA circuits was presented in [10]. In this case, the logic gate consists of a cell, composed of four dots connected in ring by tunnel junctions, and two single-dot electrometers. The logic AND and OR operations were verified. Other work described and demonstrated logic functionality using nanometer-scale magnets (MQCA) [11]. A wire and a majority gate were demonstrated and operate at room temperature. Recently, Haider et. al. [12] demonstrated the controlled coupling and occupation of silicon atomic quantum dots at room temperature. This controlled coupling of the electronic state of quantum dots can be used to the development of future QCA circuits. Finally, the development of molecular QCA has also been proposed [13]. This alternative could allow the creation of even smaller circuits working at room temperature. Other attempts to develop experimental devices for semiconductor, molecular and magnetic QCA can be found in [14]–[19].

B. Router Architecture

Here, we provide a router architecture overview. A router can be functionally divided into the control plane and data plane. The control plane is concerned with the network map. The control plane runs the routing protocols and sets the routing table. The data plane forwards the data packets with a switch and is concerned with the per-packet processing. In this work, we focus on the data plane components.

The router's data plane goal is to forward packets. We consider a router that transfer packets from N incoming links to N outgoing links. We denote R to be the data-rate, the rate at which packets arrive at every input and depart at every output.

Figure 5 presents a high-level view of a generic data-plane router architecture. There are four major components of a router:

- Input ports: represented by line cards in Figure 5 perform the physical and data-link functions. They also perform a lookup and forwarding function, so that a packet forwarded into the switching fabric of the router emerges at the correct output port.
- Switching fabric: it connects the router's input ports to its output ports.
- Output ports: an output port stores the packets that have been forwarded to it through the switching fabric and then transmits the packets on the outgoing link. The output port performs the reverse data-link and physical-layer functions.
- Routing processor: maintains the routing information and forwarding tables, performs network management functions and set the switch configuration.
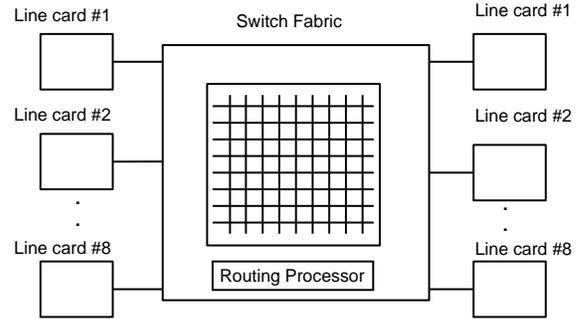

Fig. 5. Basic Data Plane Router Components.

Here, we describe how a packet is forwarded through the router. A packet arrives at the input port after the physical and data link functions are performed on the incoming link. Then, the lookup module extracts the packet destination IP and searches for the longest prefix matching with the forwarding table to determinate which output port to forward the packet via the switching fabric. After the output port of the packet has been determined, the packet can be forwarded into the switch fabric.

It is through the switching fabric that the packets are forwarded from an input port to an output port. A crossbar switch is an interconnection network consisting of $2n$ buses that connect n input ports to n output ports. It is possible to configure which horizontal bus connects to which vertical bus. After running a matching algorithm in the routing processor, the crossbar is configure to connect the input port to the output port accordingly to the matching algorithm output. Given the not practical high speed requirements of scheduling

algorithms, most routers use a heuristic scheduling algorithm such as iSLIP [20]. A packet arriving at an input port travels through the horizontal bus until it reaches the vertical bus, leading to the output port. If two or more packets are destined for the same output port, then only one packet will be transfer to the output port and all the other packets will be blocked and must wait at the input port.

Finally, the packet is buffered at the output port, where the data-link and physical-layer functions are executed.

III. RELATED WORK

Routers that have arriving packet buffered in the arriving line cards are called Input Queued (IQ) routers. If they are only buffered in the output line cards, they are denoted Output Queued (OQ) routers. There is also the combined input-output queued (CIOQ) routers. All these three types of routers need an interconnect bandwidth proportional to NR and time to set up the switch fabric is at least $\Theta(2R)$, where N is the number of links and R is the data-rate.

Some variations of routers architectures already exists, such as Clos [4], Batcher-banyan [5], combined input-output queued (CIOQ) [1]. However, they do not benefit from QCA nanotechnology. Our proposed router architecture differs from previous work because the router core (crossbar) does not need to be configured at the speed of the data link and there is no need to execute a matching algorithm to configure the switch fabric.

There are few works in the literature relating QCA within networking and communication systems. However, the recent advances in nanoscale devices may allow the development of enhanced and innovative communication systems. For example, the work by Ermolov et. al. [21] speculates that nanotechnology can be able to provide solutions for sensing, actuation, radio, embedding intelligence into the environment, power efficient computing, memory, energy sources, human-machine interation, materials mechanics, manufacturing, and environmental issues which could be crucial to the development of new communication systems, especially wireless devices. In the same work, Quantum-Dot Cellular Automata is quoted as one of the possible technologies that could enable the development of faster and less power consumer devices.

In a recent work [22], the authors presented a QCA strategy to construct a generic Delta Multistage Interconnection Networks (MINs) architecture. MINs are widely used in parallel multiprocessors systems to connect processor to processor and/or to memory modules. Also, MINs are frequently used to connect the nodes of IBMSP [23] and CRAY X-MP series [24]. Besides, MINs are applied for Networks-on-Chips (NoCs) to connect processors to memory modules on MPSoCs [25]. They have shown that these networks implemented using QCA can outperform the other nanotechnology-based implementations such as 16 nm CMOS and 16 nm carbon nanotube field effect transistors (CNFET) in therms of speed. Like them we used QCA nanotechnology but, instead of focusing solely on the interconnections (which can be used in the crossbar design), we implemented the entire router.

Graunke et. al. presented a crossbar network implemented using QCA cells [26]. This crossbar is made possible due to

the application of parallel-to-serial converters, shift registers and time-dependent latching devices. The circuit can be reprogrammed without making any physical alterations and can be extended to more complex devices by adding inputs, serial lines and majority gates. Although it has not been developed to be applied to communication systems, this circuit can be modified to serve as a communication device.

Although we do not find many papers on QCA being applied to network devices, QCA has been widely applied to the development of other logic circuits. This demonstrates the importance of research about this technology for future computers, as demonstrated in some recent studies described below.

Cho and Swartzlander proposed and simulated the design of three kinds of adders (ripple carry adder, carry lookahead adder and conditional sum adder) [27]. Gladshtein presented a serial decimal adder designed in QCA [28]. The author suggests that one of the advantage of the proposed adder is its error checking/correcting capability. Also, he indicates that the proposed solution allows the implementation of decimal nanocomputers design, avoiding both base-conversion errors and machine time losses due to this conversion. In other work, Pudi and Sridharan show a solution for the Ladner-Fischer prefix adder and a hybrid of the Ladner-Fischer and ripple carry adder, both implemented with QCA technology [29]. The authors show that the proposed hybrid adder is well-suited to the QCA model, presenting lower delay, smaller area and a smaller area-delay product than existing adder design in QCA. Recently, Dehkordi et. al. proposed two improved QCA structures for a loop-based Random Access Memory (RAM) cell [30]. The proprietary features of QCA circuits was applied with the aim of reducing the circuit area and improving the speed. In [31] a set of small functional blocks (called tiles) is proposed and applied to the design of bigger logic circuits. The proposed majority gate with an inverter input has been applied in the development of the demux presented in this work. Also, a tiled programmable fabric architecture using molecular QCA is proposed in [32]. The tiles are used to design more complex circuits. The design and simulation of modular 2^n to 1 QCA multiplexers are presented in [33]. Even reversible computing with QCA technology has been proposed. The irreversible circuits, such as the traditional CMOS circuits, lose information in most of logic gates. Landauer has shown that for irreversible logic computation, each bit of information lost generates $kTln2$ joules of heat energy, where k is Boltzmann's constant and T the absolute temperature at which the computation is performed [34]. Bennet showed that $kTln2$ energy dissipation would not occur if a computation is carried out in a reversible way [35]. Reversible circuits has the same number of inputs and outputs, avoiding the loss of energy. Thapliyal and Ranganathan proposed novel design for the implementation of concurrently testable sequential QCA circuits, based on reversible Fredkin gate [36].

The surprising lack of studies relating QCA with communication systems allows the exploration of a new line of research, such communication systems are very important components in nowadays computer systems and will be even more important in future computers.

IV. BASIC COMPONENTS

A. Crossbar

In QCA technology, information transport becomes a critical issue as cells are able to interact with all their neighbors. The current literature presents two alternatives for implementing this functionality in QCA circuits: rotated cells and multilayer cells.

The first alternative is the application of rotated cells. In this case, the quantum dots are rotated by 45 degrees inside the cell. When rotated and conventional cells are placed side by side there is no interference between them. Figure 6 shows two wires crossing in a plane. The horizontal wire is made of conventional cells while the vertical line is made of rotated cells.
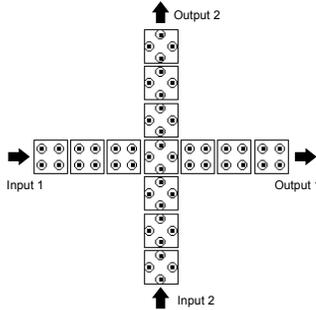

Fig. 6. Two wires crossing in a plane.

The main problem with this technique is that rotated cells are crossing the conventional cells wire, so two regular cells are separated by a greater distance. As a result, their effect on other regular cells diminishes significantly, causing the robustness of the circuit to diminish as well.

In order to bypass this problem, it was proposed that circuits could be constructed by stacking the cells in layers one above the other, that is, in multiple layers. According to this technique, the information is transported by the upper layers and, as a result, interlayer interference effects are suppressed and interaction between regular cells is strengthened. Figure 7 presents a circuit with two wires crossing informations, but in this case, the vertical wire applies the multiple layers approach. The cells marked with a circle mean that at this position there are three cells, each one in a different layer. So, the information from the first layer is propagated to the second layer and then to the third layer. Also the opposite can happen, the information can go from the third layer to the first layer. The cells marked with a cross mean that we have cells in the third layer. At this case, one can have a cell in the first layer that belongs to the horizontal wire and the cell in the third layer belongs to the vertical wire and works as a bridge.

The disadvantage that arises from this technique is that it results in an increase in the number of cells and in the number of layers as well. These drawbacks increase the circuit manufacturing costs and the risk of error occurrences. On the other hand, once the technological problems associated with the construction of multilayer circuits are solved, this technology will allow the creation of more robust circuits.
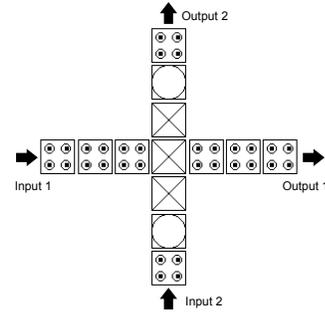

Fig. 7. Two wires crossing using multiple layers.

In this work we have decided to apply the multi-layers approach, but the circuits can be easily modified to the rotated cells design.

B. Demux

A demultiplexer (shorten as demux) is an electronic device that receives an input signal and selects one of many data output lines to send the input signal. The selection is done by signaling at the selection lines the output line.

In our case, all demuxes have one input line and four output lines (hereafter referenced as 1-to-4), using two selection lines. We can use demuxes 1-to-4 to build demux with more input and output lines.

The demux is an important part of our router circuit. It receives an input signal and directs it to the desired output line using the selection line information. Using the demux, we can decide which path, inside the circuit, the information will flow.

To the best of our knowledge, this is the first time a demux is proposed using QCA. To implement this circuit we follow the same logic used in CMOS circuits.

Figure 8 illustrates our QCA demux 1-to-4. It has one input line (IN), two selection lines (SEL_0 and SEL_1) and four output lines (OUT_0, OUT_1, OUT_2, OUT_3). We implemented it using majority gates. To solve the problem of lines crossing, without causing cells to interfere with each other, we used an architecture with multiple layers. The final result is a demux 1-to-4 that uses 149 cells and 8 clock cycles.

C. Parallel-to-Serial Converter

In a parallel-to-serial converter, a set of inputs data are carried by different wires and arrive in the device at the same time. Follow, the input values are carried over an output wire at different instants of time. A parallel-to-serial converter based on QCA has already been proposed [26]. Nevertheless, the simulation of the device was not provided.

Figure 9 shows the schematic of a parallel-to-serial converter with four inputs, such as previously proposed by [26]. The parallel inputs (input 0 to input 3) arrive in vertical wires and are serially transmitted by the horizontal line. Three QCA cells are used in the interface between each vertical wire and the horizontal wire in order to maximize the signal while avoiding interference. As one can note, the clocking zones of the horizontal wire provides the output in serial.
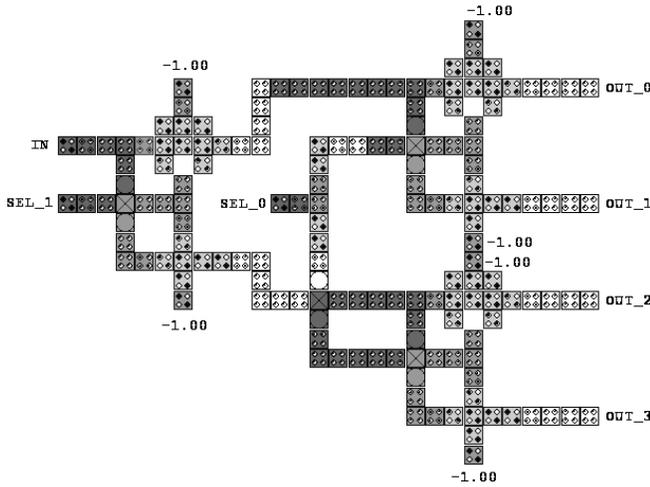
Fig. 8. Demux 1-to-4 implementation using QCA cells.

Fig. 9. Parallel-to-Serial based on QCA.

The first output signal is the input 0, followed by input 1, input 2 and, finally, input 3. After the arrival of the input data in the horizontal wire, the vertical cells have to stay in the Relax clocking phase until the input 3 signal gets the output cell. So, the vertical input cells belong to a especial clocking zone that remains in the Relax phase for a longer time. This is similar to that proposed by [26]. This feature was not implemented before in the QCADesigner simulator [3]. However, as a open source simulator, we were able to implement the necessary modification. An example of the parallel-to-serial converter is presented and discussed in section VI.

Although we have shown a converter with four inputs, the circuit can be easily modified to work with a greater number of inputs.

V. ROUTER ARCHITECTURE

In this section, we present the Router Architecture and its implementation with QCA cells.

By combining the demux, crossbar and parallel-to-serial components, which were described in the previous section, we design a packet router circuit.

Figure 10 presents our nano-router architecture based on QCA.

Initially, the incoming flow goes through the demultiplexer (demux) component. The demultiplexer determines which path the packet should go through the switch fabric. This implies what output port the flow will go.

The switch fabric is a crossbar that connects all the inputs to all the outputs. In our architecture, we do not need a controller to set up the switch configuration. We allow all possible n^2 input connections to all possible n^2 output connections. This architecture has the advantage that we do not need to set up the crossbar at the speed of the data link rate and there is no need

to execute a matching algorithm to configure the switch fabric. Realize that there is no packet collision inside the crossbar since each packet goes through a different path. Although we need a quadratic crossbar, this is not an issue due to the QCA nano-scale nature.

Multiple packets are transferred in parallel inside the crossbar. In the worst case, it is possible that all packets should be routed to the same output port. The parallel-to-serial converter handles this by directing the packets to the same output port. In our example, the parallel-to-serial component needs to run at a clock N times faster than the input data-link rate. If this is a concern, we could replace this component with a leaky bucket queue, where we have N input queues and one output queue.
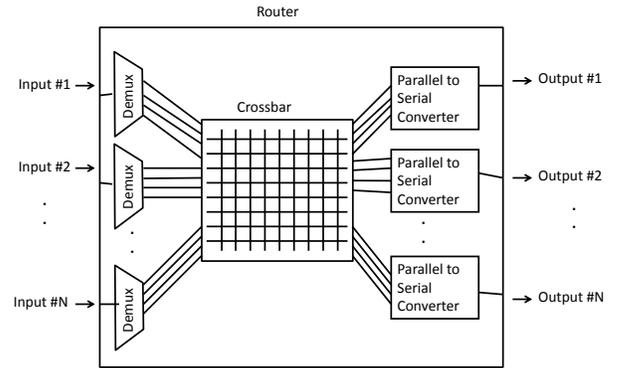

Fig. 10. Proposed Router Architecture.

Figure 11 shows the implementation of a 4x4 router architecture with QCA cells using the QCADesigner. For clarity purpose, we mark one instance of the demux component and one instance of the parallel-to-serial converter component. In the implementation, there are 4 demuxes and 4 parallel-to-serial converter components. Each different gray tone represents a different clock zone.

Despite we have presented a 4x4 router, it is also important to note that the proposed architecture is scalable and a larger NxN router can be implemented. At this case, one will have N demuxes and N parallel-to-serial converters. All components will grow in size in order to meet the demand.

As we previously described, the parallel-to-serial converter component needs to run for a period of four clock cycles to guarantee no lost of information. Therefore, for each four clock cycles, new input packets arriving at the input port can be router to the proper output port.

VI. SIMULATION RESULTS

In this section we present the simulation results. We implemented the router in QCADesigner simulator. Table I brings the simulation parameters values used in QCADesigner.

We present the QCADesigner results, which shows the polarization ($+1$ or -1) of the cells over time. A polarization of $+1$ represents a 1 and a polarization of -1 represents a 0.

First we show simulation results of the basic components. We performed many tests, here we present at least one simulation result to illustrate the correct behavior of the main
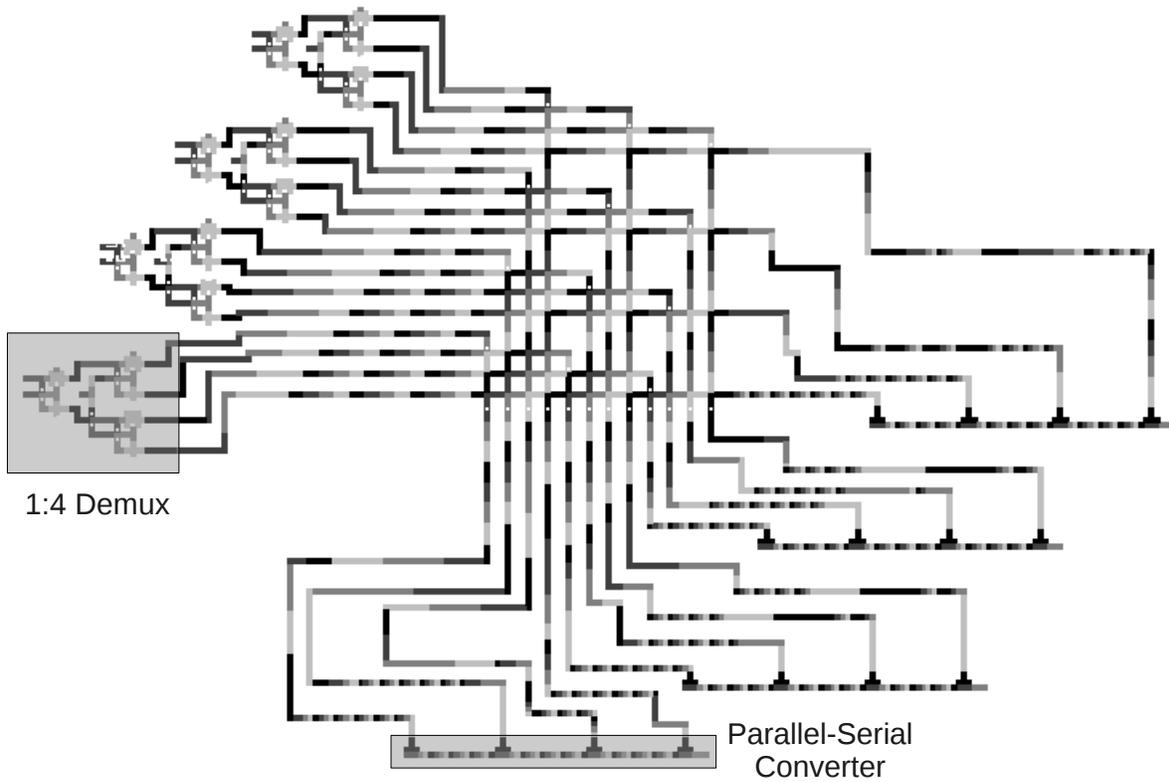
Fig. 11. Implementation of proposed router architecture with QCA cells. For clarity purpose, we mark one instance of the demux component and one instance of the parallel-to-serial converter component. Each different gray tone represents a different clock zone.

components. Next we present results for the entire router architecture.

A. Demux

Figure 12 illustrates the result for testing one demux 1-to-4. There are two selection lines, which varies to select output ports 0 (00 in binary) to 3 (11 in binary). The input line consists of a sequence of four zero´s and four one´s. The output shows the values for each output line. The arrows point from the input line values to the corresponding values in the output line. The Figure shows that each one of the four zero´s goes into each one of the output lines, as selected by the selection lines (varying from 0 to 3). Later, each one of the one´s goes into each one of the output lines as the selection lines varies from 0 to 3 again.

B. Parallel-to-Serial Converter

Table II shows an illustrative input test for the parallel-to-serial converter. Each input line receives a sequence of four bits. Each bit of the first input line propagates to the output. Next, the four bits of the second input line reaches the output. This happens for all bits of each input line. All input lines transmit in parallel and their output is serialized.

Figure 13 shows the result for testing the parallel-to-serial converter using the input from Table II. The arrows point from the input line values to the corresponding values in the output line. The output from QCADesigner simulator is the expected output.
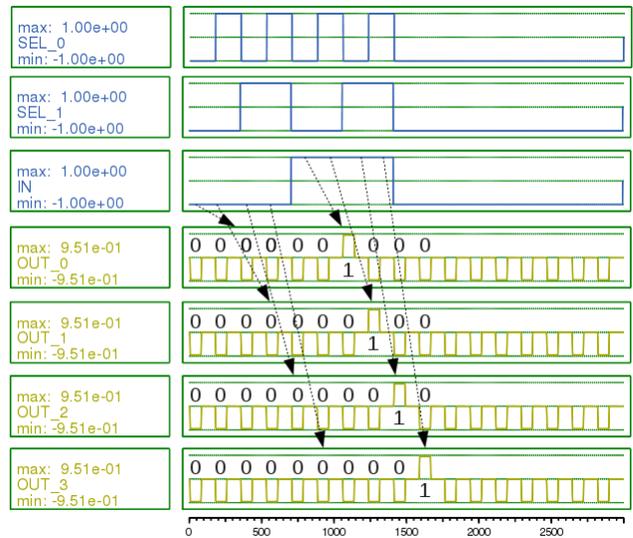

Fig. 12. Demux results.

C. Router

We implemented a 4x4 router (see Figure 11). Table III summarizes the implementation results using QCADesigner. The circuit has 4 input lines, each one with two lines indicating the destination, and 4 output lines. The values shown in Table III are related to the QCA cells size presented in

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	5nm
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-22J
Clock low	Saturation of energy of clock signal when it is low	3.8E-23J
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

TABLE I
QCA DESIGN SIMULATION SETTINGS.

	Packet
Input 0	1111
Input 1	1100
Input 2	1010
Input 3	1001

TABLE II
INPUT FOR TESTING THE PARALLEL-TO-SERIAL CONVERTER.

Table I, which are the default values used by QCADesigner. The molecular QCA cells are predicted to be $2nm$ width, drastically reducing the size of circuits [37] [13].

Number of Cells	4026
Number of Data Input Lines	4
Clock cycles	48
Area	13.81 μm^2

TABLE III
QCADESIGNER IMPLEMENTATION RESULTS.

In QCA we do not have a high impedance state. This would be useful to indicate that no communication is happening in a communication line. To indicate that a value is being used we implemented a scheme where first we send a signal 1 to indicate the start of a packet. To test the router, we defined packets with size 4, where the first bit is the start signal 1. In our architecture, packets could have any specific size.

To validate the circuit, we executed exhaustive tests. One such test result is shown in Figure 14. In this test case, all input lines have the same output (OUT2). The arrows show the packets from input lines, after the setup time, arriving at
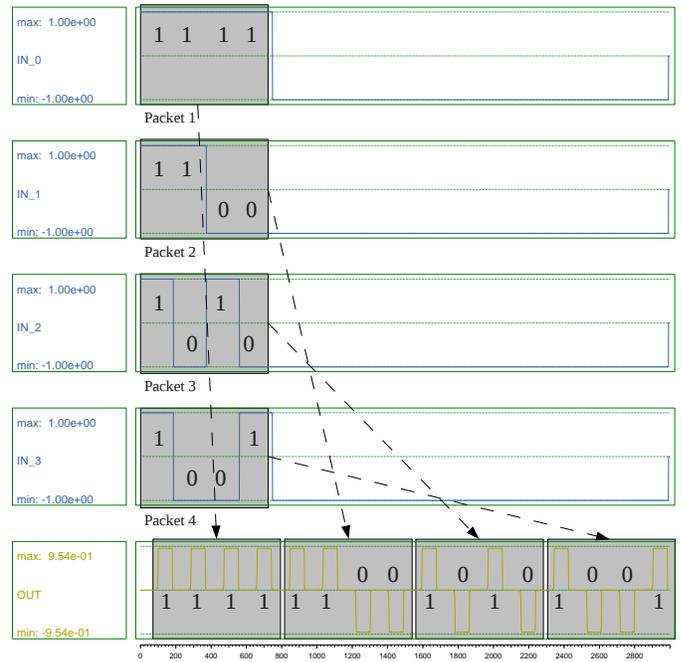

Fig. 13. Parallel-to-serial converter results.

the selected output line.

To illustrate more examples, here we show a second test, where now we use different output lines. Table IV shows the input which consists of four packets and their respective outputs for each input line. Column D refers to the destiny, the output line number for that packet. The output from QCADesigner simulator, which is correctly the expected output, is shown in Figure 15.

	Packet	D
Input 1	1111	0
Input 2	1000	0
Input 3	1111	1
Input 4	1101	3

TABLE IV
INPUT TEST CASE FOR TESTING PACKETS SENT TO DIFFERENT OUTPUTS.

VII. CONCLUSION

In this work, we proposed a new router architecture that benefits from the QCA nanotechnology. QCA is a promising nanoscale technology where components have nano size, ultra-low power consumption and might have a clock rate on terahertz range. In our novel architecture, there is no need to execute a matching algorithm to configure the switch fabric since the crossbar already connects the inputs to the outputs. This novel architecture allows for higher speed communication links, having the potential to speed up the Future Internet.

In a bottom-up approach, we first described the building blocks that compose our router such as crossbar, demux and parallel-to-serial converter and then described the full architecture. We demonstrated the functionality, validated the
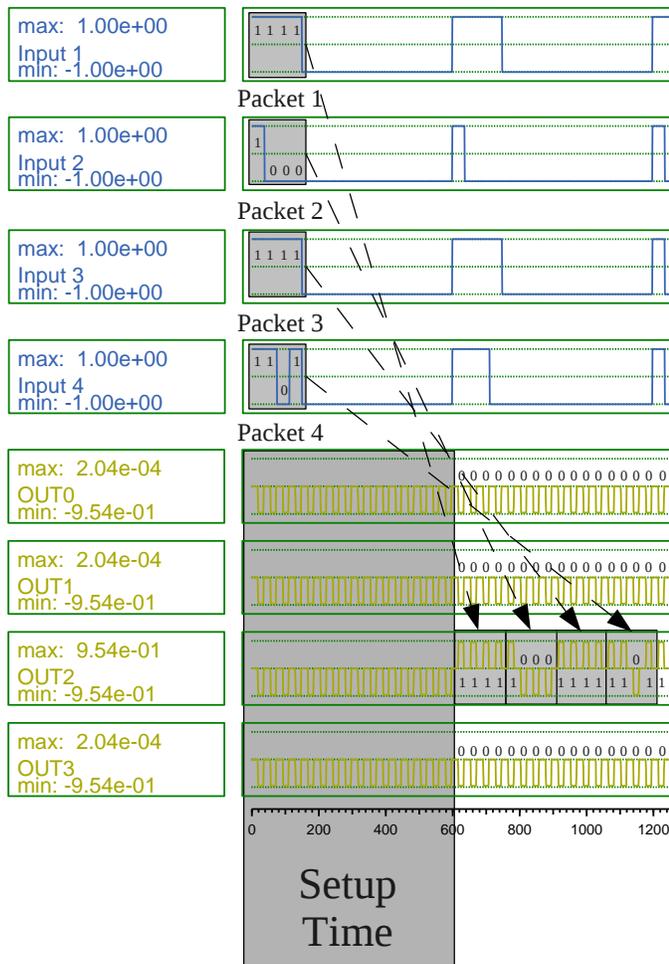
Fig. 14. Router results for first test where all packets are sent to output line 2.
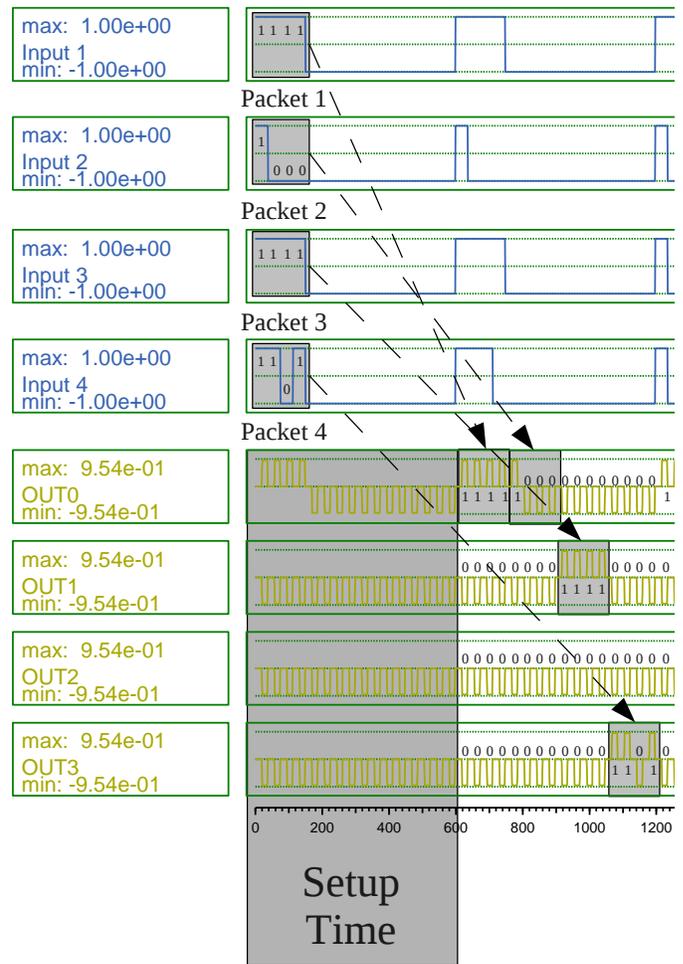

Fig. 15. Router results for the second test. Packets 1 and 2 sent to first output line, packet 3 sent to second output line, packet 4 sent to last output line.

proposed architecture and provided performance evaluations of our NanoRouter.

For future work, we intend to build the Nanorouter in the physical world. We envision future communication systems will benefit from QCA nanotechnology.

ACKNOWLEDGMENT

REFERENCES

[1] S. Iyer and N. W. McKeown, "Analysis of the parallel packet switch architecture," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 314–324, Apr. 2003.

[2] K. Kim, K. Wu, and R. Karri, "Quantum-dot cellular automata design guideline," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E89-A, no. 6, pp. 1607–1614, Jun. 2006.

[3] K. Walus, T. Dysart, G. Jullien, and R. 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. [Online]. Available: http://www.mina.ubc.ca/qcadesigner

[4] S. C. Liew, M.-H. Ng, and C. W. Chan, "Blocking and nonblocking multirate clos switching networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 307–318, Jun. 1998.

[5] M. Narasimha, "The batcher-banyan self-routing network: universality and simplification," *Communications, IEEE Transactions on*, vol. 36, no. 10, pp. 1175 –1178, oct 1988.

[6] C. Lent, P. Tougaw, W. Porod, and G. Bernstein, "Quantum cellular automata," *Nanotechnology*, vol. 4, pp. 49–57, 1993.

[7] P. Tougaw and C. Lent, "Logical devices implemented using quantum cellular automata," *Journal of Applied physics*, vol. 75, no. 3, pp. 1818–1825, 1994.

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

[9] K. Walus and G. Jullien, "Design tools for an emerging soc technology: quantum-dot cellular automata," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1225–1244, 2006.

[10] I. Amlani, A. Orlov, G. Toth, G. Bernstein, C. Lent, and G. Snider, "Digital logic gate using quantum-dot cellular automata," *Science*, vol. 284, no. 5412, pp. 289–291, 1999.

[11] A. Imre, G. Csaba, L. Ji, A. Orlov, G. Bernstein, and W. Porod, "Majority logic gate for magnetic quantum-dot cellular automata," *Science*, vol. 311, no. 5758, pp. 205–208, 2006.

[12] M. Haider, J. Pitters, G. DiLabio, L. Livadaru, J. Mutus, and R. Wolkow, "Controlled coupling and occupation of silicon atomic quantum dots at room temperature," *Physical review letters*, vol. 102, no. 4, p. 46805, 2009.

[13] C. Lent, B. Isaksen, and M. Lieberman, "Molecular quantum-dot cellular automata," *Journal of the American Chemical Society*, vol. 125, no. 4, pp. 1056–1063, 2003.

[14] A. Orlov, I. Amlani, G. Toth, C. Lent, G. Bernstein, and G. Snider, "Experimental demonstration of a binary wire for quantum-dot cellular automata," *Applied physics letters*, vol. 74, no. 19, pp. 2875–2877, 1999.

[15] I. Amlani, A. Orlov, R. Kummamuru, G. Bernstein, C. Lent, and G. Snider, "Experimental demonstration of a leadless quantum-dot

cellular automata cell," *Applied Physics Letters*, vol. 77, no. 5, pp. 738–740, 2000.

[16] A. Orlov, I. Amlani, R. Kummamuru, R. Ramasubramaniam, G. Toth, C. Lent, G. Bernstein, and G. Snider, "Experimental demonstration of clocked single-electron switching in quantum-dot cellular automata," *Applied Physics Letters*, vol. 77, no. 2, pp. 295–297, 2000.

[17] R. Cowburn and M. Welland, "Room temperature magnetic quantum cellular automata," *Science*, vol. 287, no. 5457, pp. 1466–1468, 2000.

[18] H. Qi, S. Sharma, Z. Li, G. Snider, A. Orlov, C. Lent, and T. Fehlner, "Molecular quantum cellular automata cells. electric field driven switching of a silicon surface bound array of vertically oriented two-dot molecular quantum cellular automata," *Journal of the American Chemical Society*, vol. 125, no. 49, pp. 15 250–15 259, 2003.

[19] R. Kummamuru, A. Orlov, R. Ramasubramaniam, C. Lent, G. Bernstein, and G. Snider, "Operation of a quantum-dot cellular automata (qca) shift register and analysis of errors," *Electron Devices, IEEE Transactions on*, vol. 50, no. 9, pp. 1906–1913, 2003.

[20] N. McKeown, "The islip scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, Apr. 1999.

[21] V. Ermolov, M. Heino, A. Kärkkäinen, R. Lehtiniemi, N. Nefedov, P. Pasanen, Z. Radivojevic, M. Rouvala, T. Ryhänen, E. Seppälä *et al.*, "Significance of nanotechnology for future wireless devices and communications," in *Proceedings of the 18th Annual IEEE Internal Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC-07)*, 2007, pp. 3–7.

[22] M. A. Tehrani, F. Safaei, M. H. Moaiyeri, and K. Navi, "Design and implementation of multistage interconnection networks using quantum-dot cellular automata," *Microelectronics Journal*, vol. 42, no. 6, pp. 913–922, Jun. 2011.

[23] R. Lauwereins, "Creating a world of smart re-configurable devices," *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream*, pp. 263–301, 2002.

[24] T. Cheung and J. Smith, "A simulation study of the cray x-mp memory system," *Computers, IEEE Transactions on*, vol. 100, no. 7, pp. 613–622, 1986.

[25] S. Duquennoy, S. Le Beux, P. Marquet, S. Meftali, and J. Dekeyser, "Mpnoc design: Modeling and simulation," in *15th IP Based SoC Design Conference (IP-SoC 2006), Grenoble, France*. Citeseer, 2006.

[26] C. Graunke, D. Wheeler, D. Tougaw, and J. Will, "Implementation of a crossbar network using quantum-dot cellular automata," *Nanotechnology, IEEE Transactions on*, vol. 4, no. 4, pp. 435–440, 2005.

[27] H. Cho and E. Swartzlander, "Adder designs and analyses for quantum-dot cellular automata," *Nanotechnology, IEEE Transactions on*, vol. 6, no. 3, pp. 374–383, 2007.

[28] M. Gladshtein, "Quantum-dot cellular automata serial decimal adder," *Nanotechnology, IEEE Transactions on*, vol. 10, no. 6, pp. 1377–1382, 2011.

[29] V. Pudi and K. Sridharan, "Efficient design of a hybrid adder in quantum-dot cellular automata," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 19, no. 9, pp. 1535–1548, 2011.

[30] M. Dehkordi, A. Shamsabadi, B. Ghahfarokhi, and A. Vafaei, "Novel ram cell designs based on inherent capabilities of quantum-dot cellular automata," *Microelectronics Journal*, vol. 42, no. 5, pp. 701–708, 2011.

[31] J. Huang, M. Momenzadeh, L. Schiano, M. Ottavi, and F. Lombardi, "Tile-based qca design using majority-like logic primitives," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 1, no. 3, pp. 163–185, 2005.

[32] R. Devadoss, K. Paul, and M. Balakrishnan, "p-qca: A tiled programmable fabric architecture using molecular quantum-dot cellular automata," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 7, no. 3, p. 13, 2011.

[33] V. Mardiris and I. Karafyllidis, "Design and simulation of modular 2n to 1 quantum-dot cellular automata (qca) multiplexers," *International Journal of Circuit Theory and Applications*, vol. 38, no. 8, pp. 771–785, 2010.

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

[35] C. Bennett, "Logical reversibility of computation," *IBM journal of Research and Development*, vol. 17, no. 6, pp. 525–532, 1973.

[36] H. Thapliyal and N. Ranganathan, "Reversible logic-based concurrently testable latches for molecular qca," *Nanotechnology, IEEE Transactions on*, vol. 9, no. 1, pp. 62–69, 2010.

[37] M. Lieberman, S. Chellamma, B. Varughese, Y. Wang, C. Lent, G. Bernstein, G. Snider, and F. Peiris, "Quantum-dot cellular automata at a molecular scale," *Annals of the New York Academy of Sciences*, vol. 960, no. 1, pp. 225–239, 2002.

Luiz H. B. Sardinha is a senior undergrad at the Universidade Federal de Minas Gerais (UFMG). His research interests are in Computer Networking and Nanocomputation.

Artur M. M. Costa is a senior undergrad at the Universidade Federal de Minas Gerais (UFMG). His research interests are in Computer Networking and Nanocomputation.

Omar P. Vilela Neto is an Assistant Professor of Computer Science at the Universidade Federal de Minas Gerais (UFMG). He received his undergraduate in Computer Engineering, M.S. and Ph. D. degrees in Electrical Engineering from the Pontifícia Universidade Católica do Rio de Janeiro. His research interest are in Computational nanotechnology and Nanocomputation.

Luiz F. M. Vieira is an Assistant Professor of Computer Science at the Universidade Federal de Minas Gerais (UFMG). He received his undergraduate and M.S. at the Universidade Federal de Minas Gerais in Belo Horizonte, and M.S. and Ph. D. degrees in Computer Science from the University of California Los Angeles (UCLA). His research interest is in Computer Networking.

Marcos A. M. Vieira is an Assistant Professor of Computer Science at the Universidade Federal de Minas Gerais (UFMG). He received his undergraduate and M.S. at the Universidade Federal de Minas Gerais in Belo Horizonte, and M.S. and Ph. D. degrees in Computer Science from the University of Southern California (USC). His research interest is in Computer Networking.