

Design Tools for an Emerging SoC Technology: Quantum-Dot Cellular Automata

Circuit and system design tools can be designed for devices in which digital states would be expressed by electron and hole positioning in arrays of fabricated quantum dots and individual molecules.

By KONRAD WALUS, *Member IEEE*, AND GRAHAM A. JULLIEN, *Fellow IEEE*

ABSTRACT | The future of system-on-chip (SoC) technologies, based on the scaling of current FET-based integrated circuitry, is being predicted to reach fabrication limits by the year 2015. Economic limits may be reached before that time. Continued scaling of electronic devices to molecular scales will undoubtedly require a paradigm shift from the FET-based switch to an alternative mechanism of information representation and processing. This paradigm shift will also have to encompass the tools and design culture that have made the current SoC technology possible—the ability to design monolithic integrated circuits with many hundreds of millions of transistors. In this paper, we examine the initial development of a tool to automate the design of one of the promising emerging nanoelectronic technologies, quantum-dot cellular automata, which has been proposed as a computing paradigm based on single electron effects within quantum dots and molecules.

KEYWORDS | Design tools; computing; nanoelectronics; quantum-dot cellular automata (QCA); scaling

I. INTRODUCTION

The invention of the transistor, the integrated circuit (IC), and the laser have fuelled a revolution in electronics and communications that has made possible many of the

technologies we have grown to depend on in our daily lives. The exponential growth of the number of transistors on a monolithic IC, the so-called Moore's Law, has also led to an exponential growth in computing power. There is wide agreement that this exponential growth is threatened due to problems in both the fabrication of an ever increasing density of circuits and to the operation of the devices themselves as they are scaled to increasingly smaller sizes. Several international semiconductor analysis groups have identified a "brick wall" to standard technology scaling beyond 2015. These predictions are documented in the International Roadmap for Semiconductors (ITRS) [1]. With the approach of limits to microelectronics scaling, it is becoming increasingly urgent to investigate alternatives both to devices and to computational paradigms that can enable the continuation of the exponential growth in computing power that we will increasingly rely on for advancements in networking, health care, industry, business, and entertainment.

The concept of a system-on-chip (SoC) technology encompasses the integration of the many components required for increasingly sophisticated information processing systems onto a single chip. These components need not be transistor based although they are required to be realized using a compatible set of materials, and appropriate technologies that enable their integration. The ability to place hundreds of millions of transistors onto a single chip has been a triumph for both the fabrication technology developers as well as the developers of the design tools that make this exercise a possibility within ever decreasing time to market limits imposed by the market conditions that drive the international semiconductor industry. The importance of design tools can be seen in the first book dedicated to SoC [2], where a paradigm shift, referred to as

Manuscript received September 18, 2005; revised February 3, 2006. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under a grant, in part by Micronet R&D (a former Canadian Network of Centres of Excellence) under a grant, in part by iCORE (Alberta) under a grant, and in part by CMC Microsystems.

K. Walus is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: konradw@ece.ubc.ca).

G. A. Jullien is with the Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: jullien@atips.ca).

Digital Object Identifier: 10.1109/JPROC.2006.875791

platform-based design, was encapsulated. The major feature of the paradigm shift (which transcends the platform-based design approach targeted in the book) is reflected in the use of increasingly sophisticated design tools that allow layers of abstraction to be used in which the higher layers essentially hide details of the fabrication technology from the designer. Changes in fabrication technology are reflected in the increasing sophistication of the lower level tools. The integration of future nanoelectronic technologies within the SoC paradigm will require a comprehensive set of tools; hopefully, the upper layers of abstraction will be an evolution, rather than revolution, from the current tools, thus offering a reasonably smooth design path to enable the continuation of SoC design. This will certainly represent a challenge, particularly with new nanoelectronic devices that depart from the current FET based devices, and fabrication techniques, such as directed self-assembly. As we develop these emerging technologies, we must keep in mind how they will ultimately fit into the SoC paradigm.

With this approach, we still have to look at new low-level design tools that allow the development of circuits and small systems with disruptive emerging nanoelectronic technologies. During the early stages of the invention and subsequent development of the transistor and IC, sophisticated computational tools were not available; in fact, future advances in IC technology were required to produce those tools. However, we now have the computational resources and sophisticated software techniques required, and so there is nothing to stop us from developing low-level design tools for promising emerging technologies, even though those technologies do not yet exist at a commercial level.

In order to introduce the concept of design tools for emerging technologies, we have two choices. The first is to provide an overview of different emerging technologies with limited descriptions of current approaches, if any, to the generation of design and simulation tools. The second approach is to select one of the promising emerging technologies and provide a reasonable level of detail relating to the development of a low-level but comprehensive design tool. We have elected to follow this second approach, and in this paper we discuss the development of such a low-level design tool for quantum-dot cellular automata (QCA).

Emerging technologies based on quantum dots have gained significant research interest over the past few years, and QCA has been identified as a solution for realizing computing circuits using quantum dots and interacting molecules. QCA is a novel computing paradigm which encodes information in the configuration of electrons within the QCA cell, and relies on charge interactions to enable the transmission and processing of information. We have chosen to concentrate on this technology in this paper because it represents a paradigm shift in device structure in that it is not based on a controlled conducting channel (unlike the ubiquitous FET device and its nanoelectronic counterparts). QCA technology will also require somewhat

of a design paradigm shift in that processing architectures will have to be based on very fine-grain pipelines. The design tool we will discuss, QCADesigner, contains a layout tool as well as simulators based on quantum mechanical principles, as opposed to the circuit solvers in SPICE and similar analog tools used in FET device simulators. One of the primary advantages of this computing paradigm is that it is, in principle, scalable to complexities which will be required to fit into the SoC framework.

The paper is organized as follows. In Section II we introduce QCA, including the standard cell and basic theory behind the intercell interactions. In Section III we present, in some detail, the theory behind information flow in QCA and the clocking and latency concepts that will be needed to explore the construction of basic logic blocks and architectures using these blocks. Section IV provides details for these basic logic blocks. In Section V we introduce a design tool, QCADesigner, that provides a low-level design and simulation environment for future QCA technology; we discuss both the layout tool and the simulation engines that drive the simulator. In Section VI we show some results obtained from the use of the tool including logic blocks and small architectures using these blocks; we also discuss the use of the tool in identifying problems and solutions with certain proposed QCA technologies. It is our view that the ability to use computer-aided design (CAD) tools to identify weaknesses in proposed emerging technologies, before they are capable of commercial-level implementation, is crucial in both the selection and subsequent development of such technologies. We finish the paper in Section VIII with some overall conclusions. Section VII briefly reviews various QCA implementations that have been explored over the past decade or so, all of which are amenable to the use of a tool such as QCADesigner for exploration in a design and simulation environment.

II. QCA

A. QCA Basic Cell

QCA and the QCA cell was first introduced by Prof. C. S. Lent at the University of Notre Dame [3]. QCA information processing is based on the coulombic interactions between many identical QCA cells, each constructed using four to six electronic sites coupled through quantum mechanical tunneling barriers. The electronic sites represent locations that an electron can occupy. In semiconductor implementations, these sites are realized using coupled quantum dots. Fig. 1 illustrates different potential cell geometries. The cells are designed to contain two mobile electrons which repel each other as a result of their mutual coulombic repulsion, and, in the ground state, tend to occupy the diagonal sites of the cell. Binary information can be encoded in the position of the electrons in the cell. This bistable property of QCA cells provides a natural bridging point to digital electronics.

The location of mobile electrons in a cell determines the cell's polarization. A completely isolated cell will have no external perturbations to force it into a particular polarization. As a result, the state of this isolated cell can form a quantum mechanical superposition of the two diagonal states; i.e., $P = 0$. Cells in this NULL state do not induce neighboring cells to take on any particular polarization and can be considered to be inactive.

In the four-dot cell shown in Fig. 1(b), such superimposed NULL states only exist if the cell is maintained in coherent isolation; real devices will naturally drop to a random diagonal state due to the loss of quantum mechanical coherence to the environment. In the four-dot cell, adjustable tunneling barriers are required between the dots of the cell to bring the cell in and out of this superimposed state. The other QCA cell geometries have extra configurations dedicated to realizing the NULL polarization and no quantum mechanical superposition is required. The potential energy of the sites that represent the NULL sites is controlled using coupled electrodes, and provides a direct mechanism for clocking the cells. With respect to circuit design and layout, the use of four-dot cells will suffice in demonstrating the required concepts presented in this paper. Within the present configuration of our design tool, the mechanism for clocking the cells is based on adjustable tunneling barriers, and, for consistency,

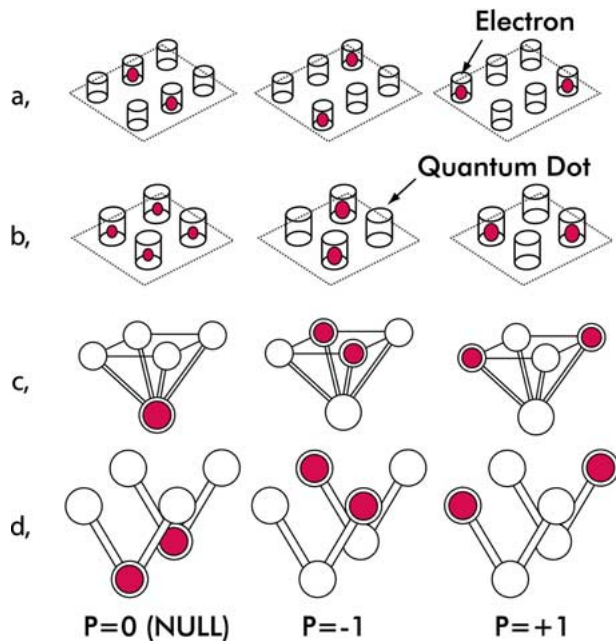


Fig. 1. Different geometries used to implement the QCA cell. Planar six-dot (a) and four-dot (b) cells are used in semiconductor and metal-island implementations. The NULL state in the four-dot cell in (b) is realized with the quantum mechanical superposition of the two polarization states. In this case the electrons are shown as being evenly distributed between the four sites of the cell. Molecular QCA cells are created using three-dimensional geometries shown in (c) and (d).

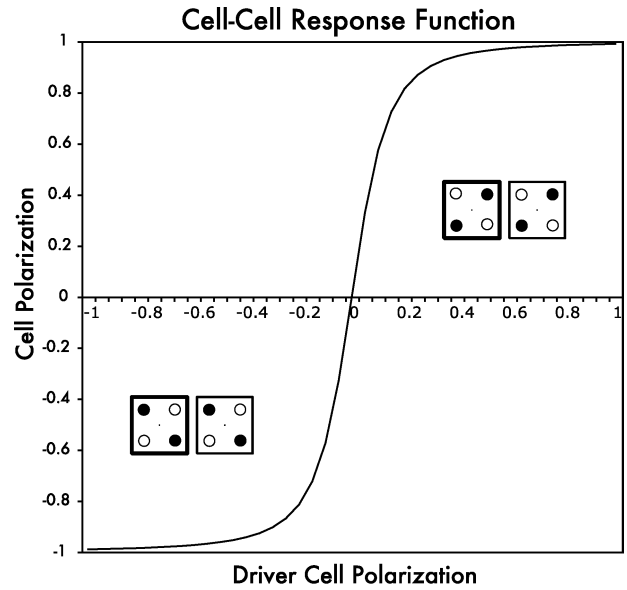


Fig. 2. Nonlinear cell-to-cell response function. The output cell is almost completely polarized for even a small polarization of the input cell (bold outline).

we will only present discussions based on these types of cells.

B. QCA Cell Interactions

Adjacent cells interact via electrostatic forces, where a quadrupole moment is induced in neighboring cells from the nonuniform distribution of cell charge. These perturbative fields create a dependency of each cell's state on the polarization of other cells. The resulting cell-to-cell interaction function is highly nonlinear, as can be seen in Fig. 2. This response function provides a noise margin and signal restoration along lengthy arrays.

Computation is accomplished by taking advantage of the fact that, provided a favorable environment, the system will tend to the ground state. As a result, QCA computing has often been referred to as ground state computation. If two cells are placed adjacent to each other, they will tend to align their polarization as a result of the electrostatic interaction of electrons between the cells. By taking advantage of this property, information can be transmitted along a wire of cells. At the leading edge of the information wave is a "kink"; i.e., two cells with opposing polarization. Unlike standard technologies, where metallic interconnects are used to connect transistors together, QCA cells act as both the switching device as well as the interconnects. This difference has a significant impact on optimizing QCA computing architectures and the latency of the circuits.

The coulombic interaction between two cells can be described by the kink energy, E_{kink} , associated with the energetic cost of two cells having opposite polarization [4].

Table 1 Kink Energy Between Adjacent Cells

Cell Type	Cell Size	Kink Energy
Molecular QCA ($\epsilon_r = 1$)	< 2nm	> 0.3eV
Self-Assembled	5nm	9.13 meV
Lithographically Defined	10nm	4.56 meV
Lithographically Defined	20nm	2.28 meV

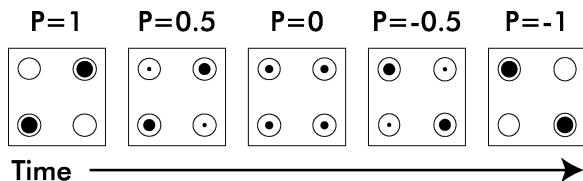


Fig. 3. Cell switching polarization from $P = 1$ to $P = -1$.

There are background charges of $+e/2$ in each dot, to ensure that the cell remains overall charge neutral. These positive charges must be included in the calculation of the kink energy. In general, the interaction between cells can be described by a quadrupole-quadrupole interaction and so we can show that the kink energy decays as the fifth power of the distance between cells. Table 1 lists the kink energy for various cell sizes. The dimensions listed in the table represent the width and height of the cell bounding box, the cell dots are located in the center of the four quadrants of this box.

III. QCA INFORMATION FLOW

Switching is driven by perturbations introduced by outside influences, such as neighboring cells, which cause the cell to switch from one polarization to another as illustrated in Fig. 3. This involves the transfer of electrons between the sites of the cell, which is made possible due to quantum mechanical tunneling. Quantum tunneling enables particles to be transmitted through potential barriers without having the required energy to overcome the barrier.

The design of QCA circuits involves finding a layout of cells, where the ground state of the layout for a particular set of boundary conditions provided by the inputs is the solution to the designed logical function. By providing a suitable environment, the cell will relax to the ground state. Changes in the boundary conditions (input values) cause the system to relax to a new ground state, and a new output.

Unfortunately, computing with the ground state implies that the system is sensitive to temperature effects. In order for the system to be thermodynamically robust, the kink energy, E_{kink} , and other relevant energies must be larger than the thermal ambient energy $k_B T$. At room temperature, $k_B T \approx 26$ meV, and it is apparent from Table 1 that only molecular scale devices have the potential for reliable room temperature operation.

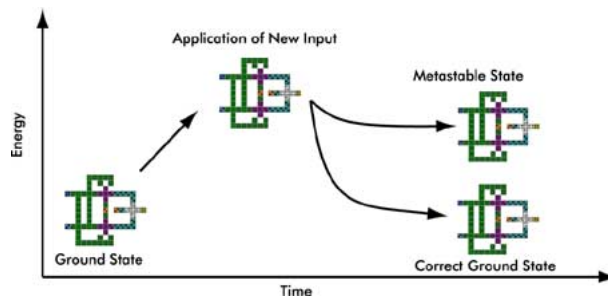


Fig. 4. Schematic representation of system energy versus time. The system starts at the ground state and is excited out of the ground state by the application of new inputs. The system then settles either to the correct ground state or a metastable state.

Switching without clocking involves the application of inputs which elevate the total energy of the system by introducing kinks. These kinks propagate through the circuit and are dissipated into the environment as the system settles to a new ground state. During the transition, it is possible that the system will settle to metastable state, as illustrated in Fig. 4. This problem can be reduced with the application of adiabatic switching.

The prediction of QCA switching speed depends highly on the particular implementation. Consider the switching of the standard semiconductor cell. In this case, switching speed depends on the relative strength of elastic and inelastic switching processes. If the system is strongly coupled to the substrate, the inelastic process will dominate and the switching speed will be determined by the strength of the coupling. The calculation of this coupling is an extremely difficult problem to solve, and can be determined using experimental techniques. However, if elastic processes dominate the switching, the switching speed can be determined using the Schrödinger equation and modeled on a computer. Previous work has shown that the standard cell has a switching time as low as 2 ps [5] when operating in this coherent regime. Table 2 lists the order of the theoretical cell switching frequencies for the various cell implementations.

A. Power Dissipation

Power dissipation can be estimated from established models under different assumptions of cell coupling to the environment [8], [9]. If cells are switched quasi-adiabatically, the total dissipation per cycle into the environment will be less than E_{kink} . However, if the cells

Table 2 Theoretic Orders of QCA Cell Switching Frequencies

Implementation	Frequency
Molecular [4]	THz
Semiconductor [5]	THz
Metallic Island [6]	GHz
Magnetic [7]	MHz

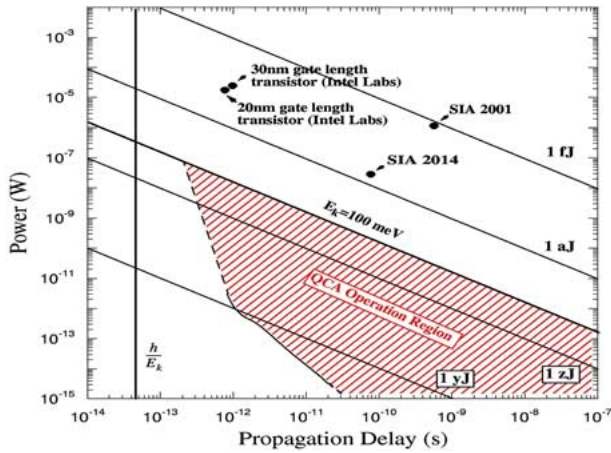


Fig. 5. Power dissipation in QCA and SIA predictions for CMOS [8]. Reused with permission from J. Timler, J. Appl. Phys., vol. 91, p. 823 (2002). Copyright 2002, American Institute of Physics.

are tightly coupled to the environment and irreversible operations are performed, each cell will dissipate E_{kink} for each cycle. Fig. 5 shows the calculated power dissipation per device versus the propagation delay. In the worst case, QCA is still expected to operate at power dissipations below the SIA roadmap for semiconductor predictions for 2014 CMOS technology.

B. Cell Clocking

As with standard technologies, QCA clocking provides a mechanism for synchronizing information flow through the circuit. However, unlike standard technologies, which have a built-in directionality for information flow, the clock also controls the direction of information flow in a QCA circuit. The QCA clock has also been shown to provide the power gain required for proper circuit operation [8]. Beyond this, the clock enables the quasi-adiabatic switching of cells which is required to avoid problematic metastable states. There are currently two proposed methods of clocking QCA cells.

The mechanism by which the clock signal changes the state of the cells from NULL to one of the two polarization states depends on the particular implementation. In the four-dot implementation, the clock signal is used to adjust the height of the tunneling barriers between the quantum dots. When the clock is low, the electrons are trapped in their associated positions and are unable to tunnel to other dots, effectively latching the cell. When the clock signal is high, the electronic wavefunction becomes delocalized and the cell is said to be in the NULL polarization state. In between, the cells are either latching or relaxing. Due primarily to the challenges in implementing such a system, most of the proposed QCA devices have extra sites which enable additional electronic configurations associated with the NULL state. In these implementations, the clock signal is

tied to the potential energy of these extra sites. When these sites have a lower potential compared to the active cell sites, the electrons will relax into the NULL state. Alternatively, if the potential of these sites is raised above the peripheral sites, the cell will relax into the active state. Within our simulation tool, clocking is implemented as a signal applied to adjustable tunneling barriers. Zone clocking and continuous clocking are two potential methods for clocking QCA cells.

1) *Zone Clocking*: With zone clocking all the cells in a design are grouped into one of four available clocking zones; each cell in a particular clocking zone is connected to one of the four available phases of the QCA clock shown in Fig. 6 [10]. Each cell in the zone is latched and unlatched synchronized with the changing clock signal.

The clock signals act to pump information throughout the circuit as a result of the successive latching and unlatching in cells connected to the different clock phases. For example, a wire, which is clocked from left to right with increasing clocking zones, will carry information in the same direction; i.e., from left to right. This acts to pipeline QCA circuits at a clocking zone level. QCA wires are unique in that more than one bit of information can be propagated along the same wire at any one time. Within the present tool, zone clocking is the only method that is accessible, and our following discussions will be based on results using this method.

2) *Continuous Clocking*: Continuous clocking involves generating a potential field by a system of submerged electrodes [11], [12]. The potential of the different sites of the cells depends on the total electric field generated at those sites by the electrodes. By applying phase-shifted sinusoids to each of the electrodes a forward moving wave can be generated at the level of the cells. Cells are latched at the wavefront of this forward moving wave as illustrated in Fig. 7.

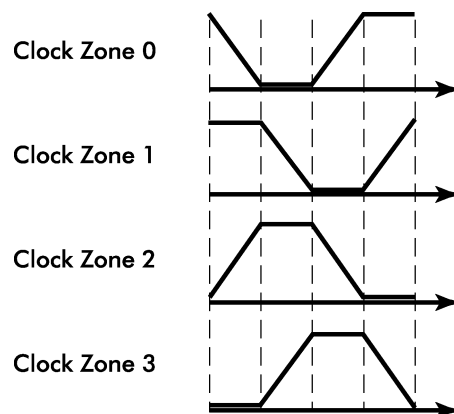


Fig. 6. The four phases of the QCA clock used to control information flow in the QCA circuit.

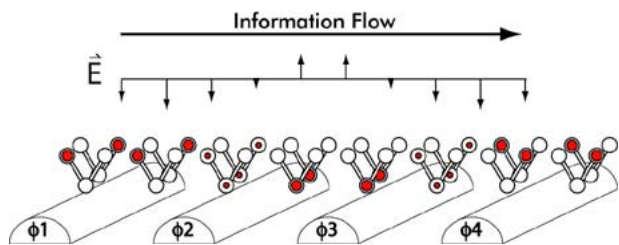


Fig. 7. Submerged electrodes can be used to clock QCA cells by applying a forward moving electric field at the level of the cells.

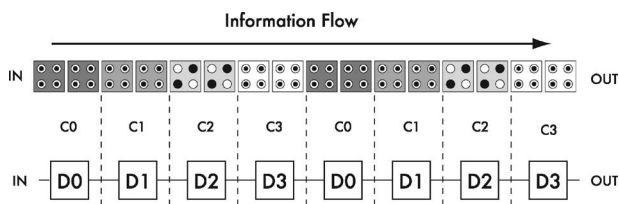


Fig. 8. QCA wire shown with cells and schematic representation. C0, C1, C2, and C3 are the four phases of the clock. Each of the clocking zones maps to a numbered D-latch in the circuit representation. Notice that only one clocking zone is latched.

C. Interconnect Latency

Within the zone clocking scheme, each group of cells connected to a particular phase of the clock can be considered as a D-latch [13]. As each group of cells in a particular clocking zone become latched they retain their information until the clock is relaxed, independent of changes in the polarization of neighboring cells. A length of QCA wire can be represented schematically as shown in Fig. 8. This inherent zone latching has a major effect on the design cost function. In all schematics presented in this paper, we use the numbered D-latch representation to incorporate the clock zone information.

In the QCA layout shown in Fig. 8, the different clocking zones are represented with different shades of gray. Simulation results for this clocked wire are shown in Fig. 9. The simulation results show that the first two latched periods of the output cell have values which are indeterminate, since the number of clock cycles simulated has been insufficient to clock the input all the way to the output.

The deep pipeline inherent in QCA circuits forces us to evaluate our designs differently than we would if we were designing with traditional technologies. Even in heavily pipelined transistor based logic architectures, there will normally be many gates in a combinational structure between each latch in the pipeline. In QCA, the latency is determined completely by the largest number of clocking zones between input and output, with each gate and wire connection being connected to a clocking zone. The original concept of QCA, as a quantum-level implementation

of classical cellular automata (CA) [3], is evident from the inherent clocked operation at the device level. Although CA architectures are not normally implemented in most synchronous digital systems, we can look at structures that have similar properties as being target candidates for QCA implementation. Examples are bit-serial architectures and systolic arrays. Interestingly, parallel designs, which require large fan-outs to the parallel logic components, will introduce more latency because of the very nature of QCA interconnects; thus, we will need to change our computing paradigms to match the different cost functions inherent with using future QCA technology.

D. Interfacing

Any nanoscale computing technology requires a robust method of interfacing with the macro world. Interfacing the QCA circuit requires a technology that can create and detect the configuration of single electrons within the cell. Researchers have verified that a single-electron transistor (SET) can be used to read the state of individual output cells [6], [14], [15]. Such SETs are highly sensitive to the

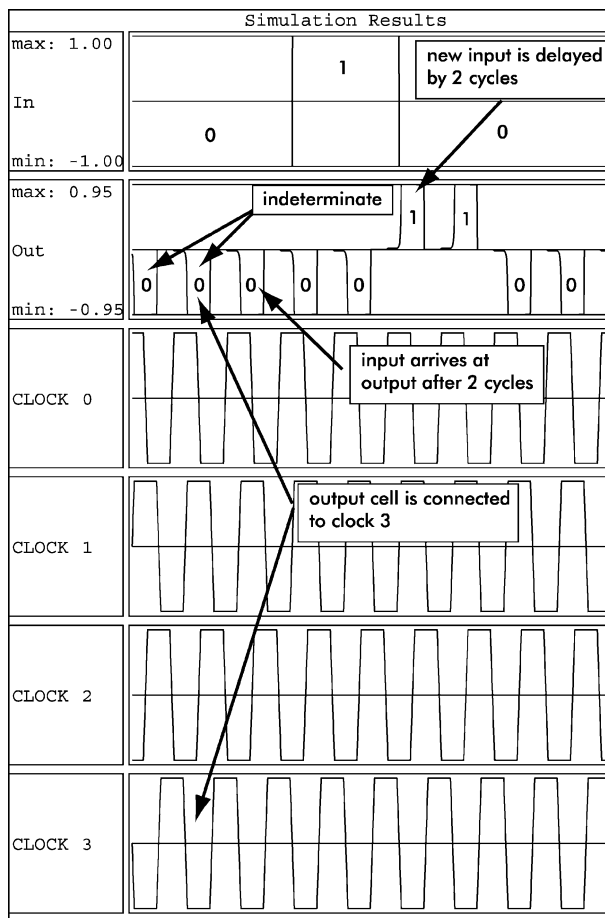


Fig. 9. Simulation output of the clocked wire. The first two outputs are indeterminate because they represent polarization in the wire prior to the complete propagation of the first input.

presence or absence of a single charge at the gate electrode of the SET. Significant research has been invested in developing techniques to realize room-temperature [16] and high-frequency [17] SETs. The state of an input cell can be set with the presence of a charged electrode located nearer to one of the four dots of the input cell. The induced potential from this electrode will be sufficient to alter the ground state configuration and therefore the polarization of the cell.

IV. QCA LOGIC

A. Inverter

The most common inverter design is shown in Fig. 10. This *fork inverter* has two legs of the input QCA wire which interact at a 45° angle with the first cell of the output wire. At this angle, the coupling between cells is negative and can be exploited to realize the complement function.

B. Majority Gate

The fundamental logic primitive available with QCA technology is the majority gate. This gate performs the following Boolean function:

$$\text{Maj}(A, B, C) = AB + AC + BC. \quad (1)$$

At least two of the inputs to the gate must be asserted before the output is asserted. This gate is a member of the higher class of threshold gates where the sum of weighted inputs must exceed the threshold before the output is asserted [18]. The three-input majority gate is implemented with the layout shown in Fig. 11.

The majority gate can be programmed to perform the standard AND and OR operations by fixing the polarization to one of the three available inputs as illustrated in Fig. 12.

C. Fixed Polarization Cells

As discussed in the previous section, fixed polarization cells are required to implement AND and OR functions.

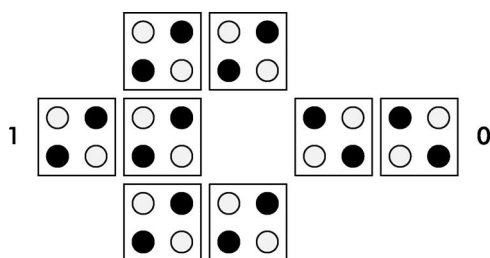


Fig. 10. Layout of a fork inverter. The two arms of the fork reinforce the inverter function by doubling the coupling to the output cells.

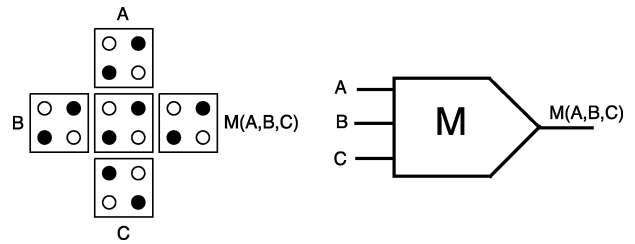


Fig. 11. The three-input majority gate is the fundamental logic primitive available with QCA. It is created from a cross pattern of five cells. At least two of the cells must be logic “1” before the output is logic “1.”

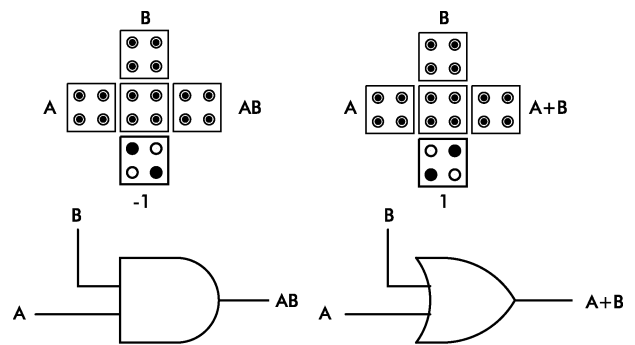


Fig. 12. The majority gate as a programmable AND and OR gate. Fixing one of the inputs to logic “0” creates a two-input AND gate. Alternatively, if we fix one of the inputs to logic “1,” we obtain a two-input OR gate.

There are two methods for realizing these, the first involves a QCA network that distributes a constant polarization to all the AND and OR gates, inverting where necessary. The other method involves the dot-level manipulation of cells within the circuit. These fixed polarization cells can be implemented by simply removing two quantum dots from the cell, leaving the two dots associated with the diagonal that gives the desired polarization. The use of fixed polarization cells results in circuits which consume far less area, and are also less likely to experience errors associated with the large network required by the first approach. One approach to minimizing the overall hardware requirements, as well as the number of fixed cells, is to find the optimal majority representation of a circuit [19].

V. QCADesigner

A. Rationale

When commercial fabrication processes have reached a stage of advancement where any emerging technology is an economic reality, the potential of the technology should already have been explored and uses for the technology

found. In fact, this *vertical* approach may even change the amount of effort spent on developing a technology based on, for example, the uses (or lack of them) found and the fabrication tolerances required. The best way to evaluate a technology is to provide tools for design and simulation and make these available to a large group of future designers within that technology. With sufficiently accurate simulation models and computational techniques, these tools are able to predict the level of success of nanocircuit designs, even though such nanotechnology is not available. Thus, the availability of a design and simulation tool for promising emerging nanotechnologies provides interested researchers with the ability to explore the design, and also some elements of the fabrication, space of the technology. This should lead to commercialization that can be accomplished in a much shorter time frame than would otherwise be possible. This approach also allows a large body of future designers in the technology to become very familiar with its design requirements even before it is commercialized. Such design tools should also provide the necessary infrastructure to evaluate the potential of the underlying technology and the extent to which it can be scaled to complexities that would require design paradigms such as those now developed for SoC.

Within the context of this paper, we describe one tool, QCADesigner [20]–[23], which facilitates the rapid design, layout, and simulation of complex QCA circuits by providing standard and easy-to-use CAD capabilities with an advanced back end to support quantum mechanical simulations of complex circuits [24]. QCADesigner is available to the QCA research community via the Internet.¹ We invite readers to explore the tool particularly in light of the potential of QCA in emerging SoC technologies.

B. Tool Requirements

A complete evaluation of a technology has to be made by considering many facets of design, including determining which circuits can be mapped directly to the new technology, which ones require a rethinking of the architectural choices, which layouts create robust circuits, etc. In most cases this will require a time investment by several groups and many people, all interacting with the design and simulation tools to perform their analysis. The design tool(s) should not require the users to possess detailed theoretical knowledge of the devices and technology in order to use the tool appropriately. Rather, the tool should allow the users to fully explore and extend the design space in order that as many interested designers as possible become involved in the effort. To maximize the number of users, developers of the tool(s) have to consider features that include graphical user interface design that does not obfuscate the features behind multilevel dialog

trees or command line parameters. With any simulation tool, and in particular with emerging technologies that are yet to be commercially realized, the simulation models will unavoidably change. Simulation tools should be developed to handle such changes by design.

C. QCADesigner Data Flow

Simulation engines within QCADesigner model the statics and dynamics of QCA operation. Although two simulation engines are currently included with the tool, it is relatively easy for external groups to develop and incorporate their own simulation engines, since the source code for the tool is made available. The QCA circuit is internally represented by a linked list of cell objects and their associated properties (position, size of the cell, size of the dots, charge distribution, etc). When the user starts a simulation, the circuit and the appropriate simulation engine options, as well as a vector table of inputs, are sent to the simulation engine. If the user does not provide a vector table then QCADesigner will perform an exhaustive simulation over all possible input vectors.

The simulation engine interprets the circuit and extracts the cell properties and the variables it requires to complete the simulation. Once completed, the simulation engine generates a simulation data structure, which consists of the traces for each designated output and input cell, as well as a unique trace for each of the four phases of the clock. This structure is sent to the graphing data interpreter, which constructs the digital representation of any designed data buses by considering where the traces exceed a predefined threshold. Once all structures have been built,

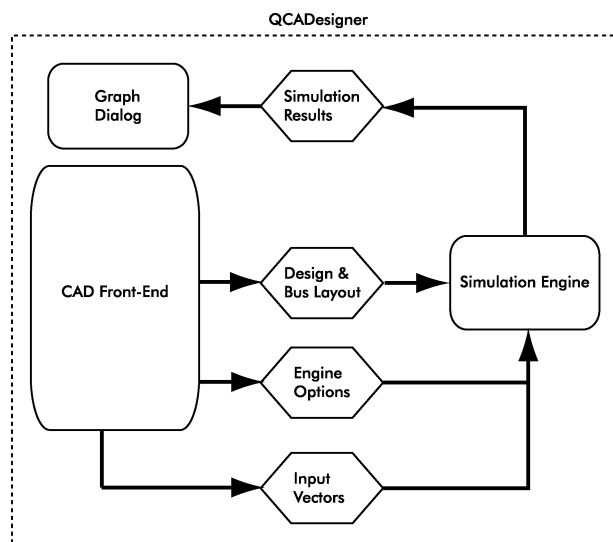


Fig. 13. The flow of data from the QCADesigner front end which sends a linked list of cells, simulation engine options, and a vector table to the simulation engine. Once the simulation is complete, the engine sends the simulation data back to the graph dialog which interprets and displays the results to the user.

¹[Online]. Available: <http://www.qcadesigner.ca>

the graph dialog displays the signal traces, as well as the digital representation of the buses. The flow of data through the simulation engine is illustrated in Fig. 13.

D. Simulation Engines

Included in the current version of QCADesigner are two different simulation engines: the bistable and coherence vector engines. The main challenge in implementing physically accurate simulations is the lack of experimental data for the various QCA implementations. Even though several small QCA systems have been developed as proof-of-concept experiments [10], [15], [25], [26], these devices do not necessarily represent scalable QCA technology. As a result, one of the main objectives of this effort is to provide motivation for further research into the implementation of such devices and developing a continued dialog between circuit designers and researchers investigating these implementations.

References [5], [8], [27]–[31] develop models which describe the static and dynamic behavior of electronic QCA. In general, quantum mechanical systems are not suitable for efficient simulation on a classical computer, and the models are approximations of the full quantum mechanical behavior of the system. It is possible to simulate the full quantum mechanical behavior of very small systems, but the problem grows exponentially with the size of the system and will quickly exceed the capability of any available computer. We are therefore required to make approximations about the quantum mechanical coupling between cells which enable the reduction of the problem to something that is manageable.

E. The Layout Tool

A screen shot of the QCADesigner design environment is shown in Fig. 14. QCADesigner has attracted some

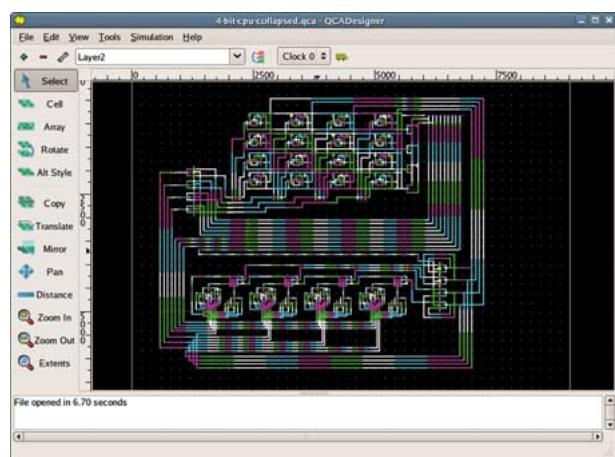


Fig. 14. Screen shot of the QCADesigner design environment. QCADesigner is shown here running under Linux but can also be compiled on Windows, Mac OS X, BSD, and Solaris.

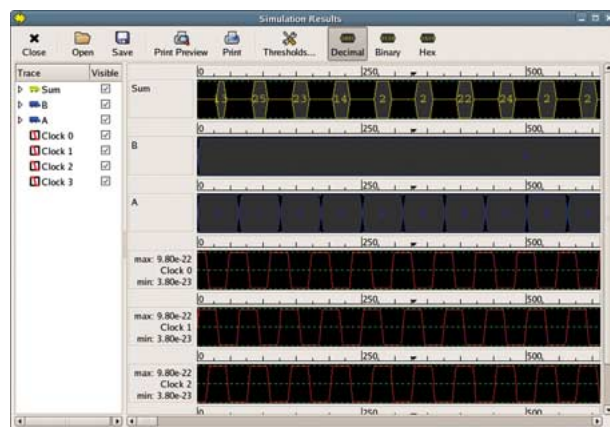


Fig. 15. QCADesigner waveform analysis dialog with digital bus display.

important new developers, and results from simulations using this tool have been published by several international groups [32]–[42]. QCADesigner is written in C and employs a wide range of open-source software such as the GTK graphics library, and is maintained under the GNU Public License (GPL) for open-source software. Developing the project in this manner enables it to be compiled and used on a wide range of systems including Windows, Linux, Mac OS X, BSD, and Solaris. The objective of the project is to create an easy to use simulation and layout tool available freely to the research community via the Internet [43]. One of the most important design specifications is that other developers should be able to easily integrate their own utilities and simulation engines into QCADesigner. This is accomplished by providing a standardized method of representing information within the software.

1) *CAD Capabilities:* Some of the important CAD features include:

- add/remove cells;
- move/scale/rotate cells;
- create input/output cells;
- create arrays of cells;
- group cells into digital buses;
- create/edit labels;
- PostScript printing;
- ability to merge designs.

A graphical results interpreter, shown in Fig. 15, has also been incorporated into the tool. The graphing window allows the user to analyze the simulation results as well as create detailed PostScript printouts for figures in publications. The tool also includes a digital interpreter, which can combine output traces into logical data buses and display the resulting digital representation of the combined trace data. This will facilitate future automatic

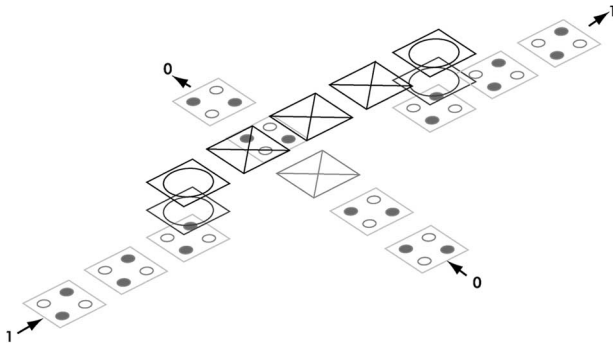


Fig. 16. Alternative cell views to represent cells on different layers of QCA.

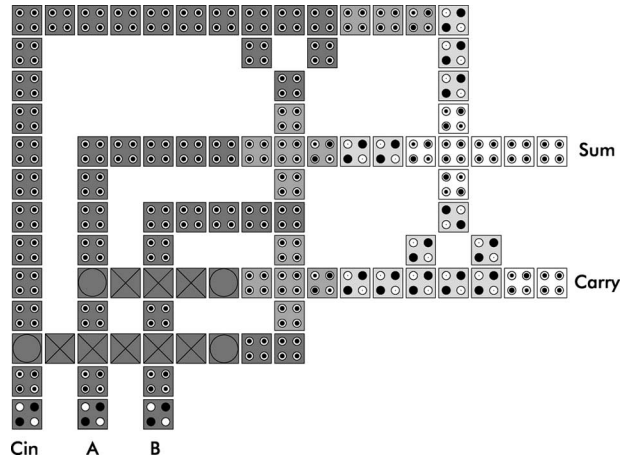


Fig. 18. QCA cell layout of adder.

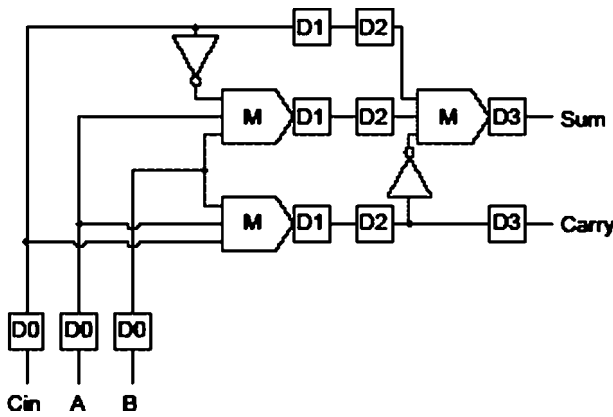


Fig. 17. Adder schematic using only majority gates.

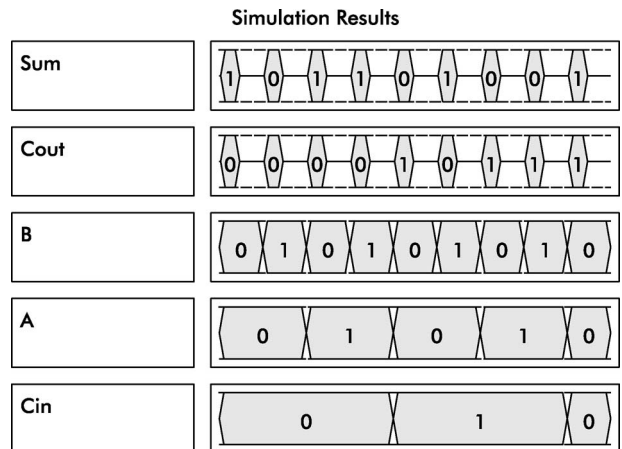


Fig. 19. Simulation results for the full adder.

testing and comparison of the simulation output against a predefined set of vectors.

VI. USING THE DESIGN TOOL

A complete evaluation of a technology cannot be made without the rigor of actually designing and simulating many of the fundamental circuits required by any computing technology. Using QCADesigner, many of these circuits have already been developed and simulated. This section presents circuits and results based on the present capabilities of the tool. There is ongoing research into clocking and cell geometries that are currently not implemented in the tool; results based on these developments will be presented in future versions of QCADesigner.

One of the main features that is currently unavailable in the tool is the continuous clocking scheme, which may enable the coplanar wire crossing as described in [44]. All of the available circuits have been designed within the zone clocking scheme currently available in the tool. With zone clocking, it has been found that multiple levels of

cells are required to implement the wire crossing as described in Section VI-B. Such a problem does not exist in the continuous clocking scheme and there is significant effort to try and implement models for those simulations within QCADesigner.

As a design convention, we have implemented alternative cell representations into QCADesigner so that the user can easily identify cells that exist on separate layers of a QCA circuit. The cells which make up a vertical interconnect are displayed with a circle centered in each cell. The cells which are part of the second circuit layer, and not part of the vertical interconnect, are represented with an “X” through the cell. The cells directly below cells on the top layer are also shown with an “X.” Fig. 16 illustrates this convention.

Because we ensure that all vertically stacked cells are connected to the same clocking zone, a new clocking

layout is not required for each layer of cells, simplifying the actual implementation.

A. Example Designs

1) *QCA Addition*: Of the variety of information processing tasks that computers perform, the most basic is certainly the addition of two single digit binary numbers. References [45] and [46] investigate the design and layout of a QCA full-adder using QCADesigner in more detail. The schematic for the full adder is shown in Fig. 17.

The layout for this adder is shown in Fig. 18. The apparent waste of space in this layout is a result of design requirements that attempt to minimize the crosstalk between different parts of the QCA circuit. The simulation results for this adder are shown in Fig. 19. The delay between the application of the input and the appearance of the associated output is due to the inherent delay in the circuit. Fig. 17 shows the four clocking zones (one clock cycle) between the input and output.

2) *n-bit QCA Adders*: A QCA ripple-carry adder is implemented by cascading n 1-bit QCA full adders. The schematic for a 4-bit adder is shown in Fig. 20. As the inputs increase in significance, a larger number of clock cycles are introduced in order to synchronize the arrival of the input with the carry of the preceding adder.

The layout of the 4-bit adder is shown in Fig. 21.

The overall latency can be obtained by considering the critical path from one of the inputs of the first adder to the carry output of the n th adder. Since the carry output of a QCA full adder requires one clock cycle to complete, the overall latency for the n -bit adder will be n clock cycles. The advantage of the QCA adder is that it is pipelined. As a result, we can feed new inputs into the adder every clock

cycle. When the pipeline is full a new output will be available every clock cycle independent of the depth of the pipeline.

An n -bit QCA carry-look-ahead adder can be expressed as

$$\begin{aligned}
 C_i &= G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \dots + P_i \dots P_1 C_0 \\
 &= G_i + P_i (G_{i-1} + P_{i-1} (\dots + (G_1 + P_1 C_0))) \\
 S_i &= C_{i-1} \oplus P_i = C_{i-1} \bar{P}_i + \bar{C}_{i-1} P_i
 \end{aligned}
 \tag{2}$$

where

$$\begin{aligned}
 G_i &= A_i B_i \quad \text{generate signal} \\
 P_i &= A_i \oplus B_i = A_i \bar{B}_i + \bar{A}_i B_i \quad \text{propagate signal.}
 \end{aligned}
 \tag{3}$$

Due to the limitation of QCA to two-input AND and OR gates, the critical path of a 4-bit carry-look-ahead adder consists of ten gates; i.e., ten majority gates, from the input of the first bit to the carry output C_4 or the sum output S_4 as shown in Fig. 22. Therefore, the best case latency of this architecture is estimated at five clock cycles.

In an actual layout, the latency may be larger because the long interconnects associated with the most significant carries would have to be divided into more than one clocking zone. The advantages of carry-look-ahead adders are dependent on the ability to realize AND and OR gates with more than two inputs. A ripple-carry adder is considered the slowest bit-parallel adder design in CMOS circuits. However, the QCA ripple-carry adder has better performance when compared to both the QCA carry-look-ahead and carry-select adder designs.

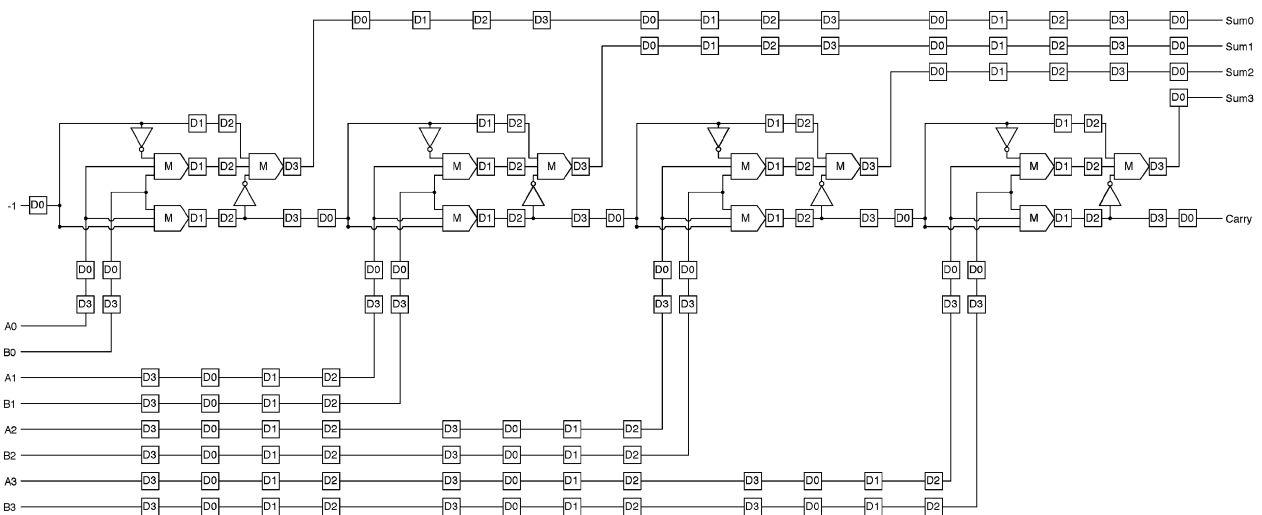


Fig. 20. Schematic of a 4-bit QCA ripple-carry adder.

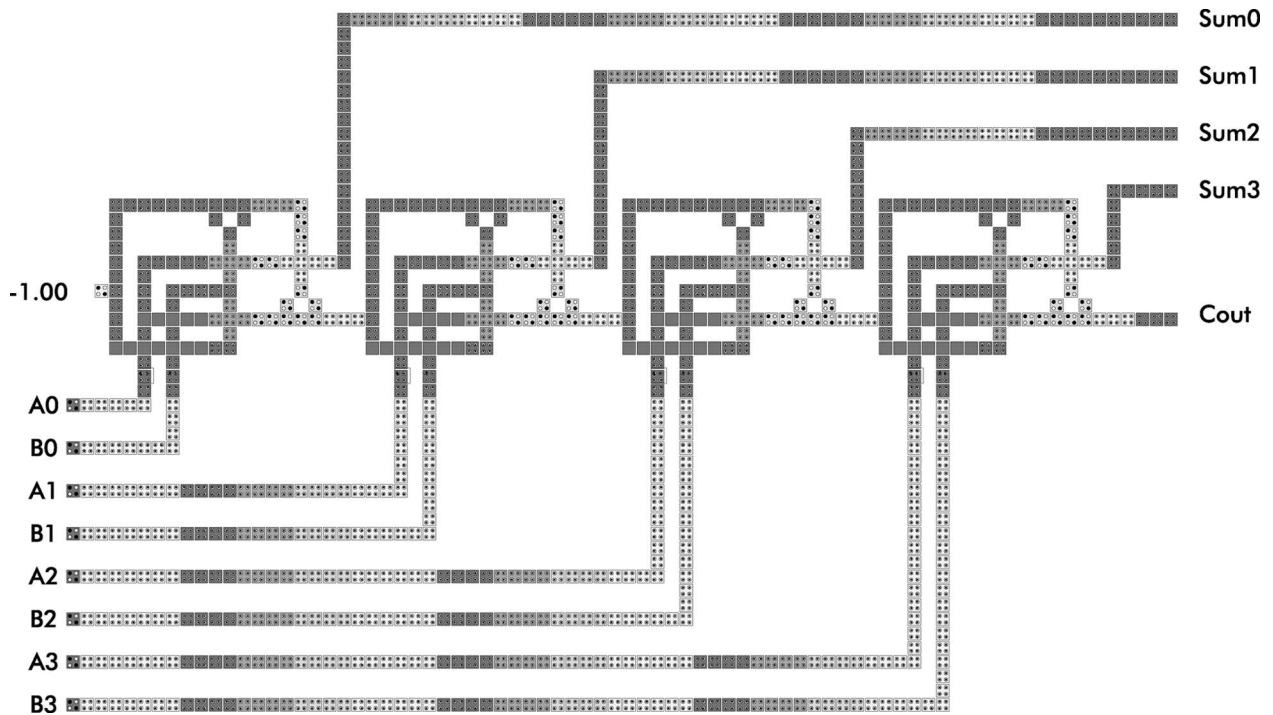


Fig. 21. Layout of a 4-bit QCA ripple-carry adder.

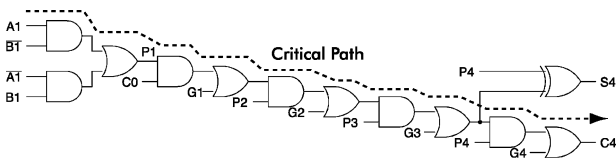


Fig. 22. Critical path of a 4-bit QCA carry-lookahead adder: from the input of the 1st bit to the carry output C_4 or the sum output S_4 .

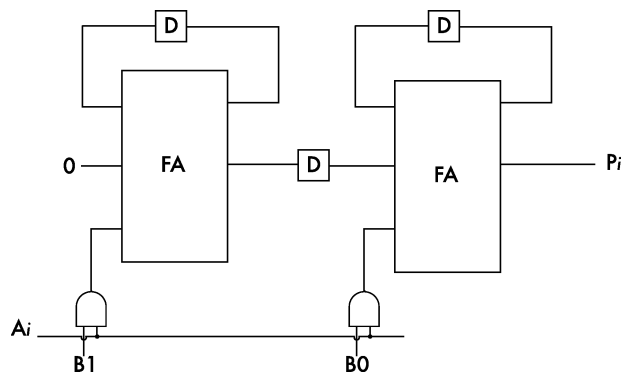


Fig. 23. Block schematic of bit-serial multiplier.

3) QCA Multiplication: A constant coefficient multiplier has been implemented using the adders described above [13]. The block schematic for a 2-bit multiplier is shown in Fig. 23. The D-latches in this schematic are required for the proper operation of the device, and are not QCA zone latches. In order to map this design into a QCA circuit, we have to realize that many more D-latches are introduced from the very nature of the QCA circuit. The D-latch between the two adder blocks in the original schematic (Fig. 23), is implemented in QCA by adding four additional clocking zones (one clock cycle) along that path.

One of the inputs to the multiplier is broadcast across the multiplier serially, the other is constant and implemented using fixed polarization cells. The schematic for this multiplier is shown in Fig. 24. The overall latency for the multiplier is three clock cycles; one clock cycle inherent to each adder, and one to implement the latch between them.

The QCA layout for the multiplier is shown in Fig. 25. The schematic is drawn to match the layout as much as possible. The multiplier can be easily scaled by adding full-adder blocks and partial product generators. We have experimented with designs as large as 32-bit using this layout. The size of the multiplier grows linearly with the number of bits, making it efficient in area. As the size of the multiplier is increased, the latency increases according to

$$L = 2n - 1 \tag{4}$$

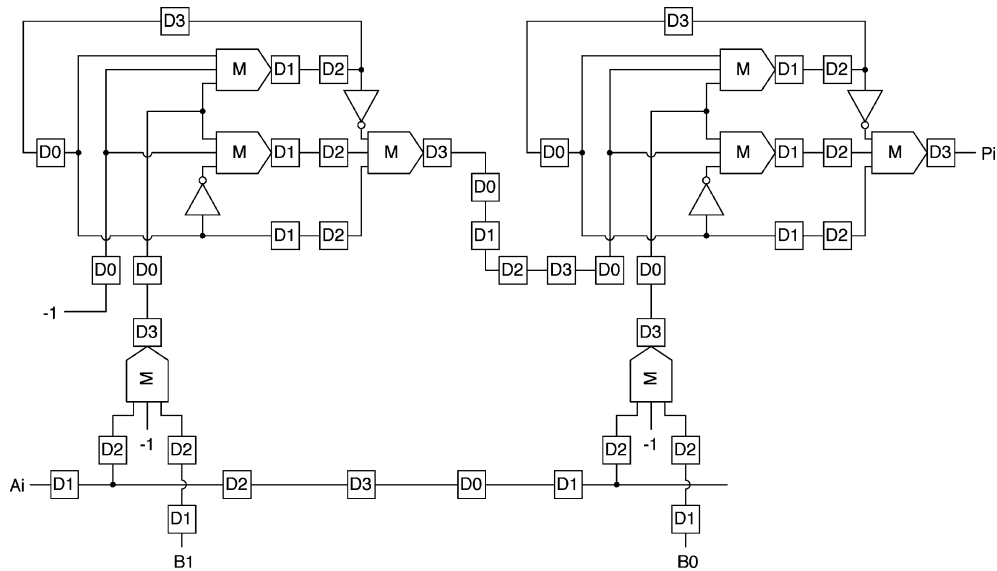


Fig. 24. Schematic of QCA bit-serial multiplier.

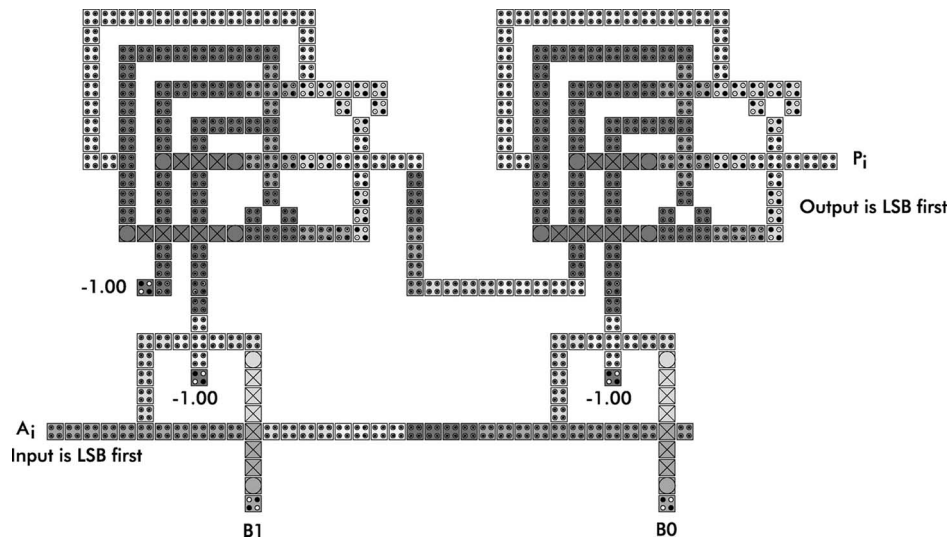


Fig. 25. Layout of QCA bit-serial multiplier.

where the latency L is measured in clock cycles, and n is the size of the multiplier in input bits.

4) *Memory*: Zone clocking creates a shift register with each wire of clocked cells. However, each cell is connected to a clock signal that will clear the cell contents once every clock cycle. To correct for this loss of data, small loops can be used to retain information. The simplest memory loop consists of all four clocking zones, enabling the information to continuously circulate in the loop. Fig. 26 shows a

QCA memory loop without any mechanism for reading or writing information.

5) *Memory Cell*: Extra control can be added to the memory loop to create a memory cell. The memory cell is a building block of RAM and implements a bit-level read/write function [47]. The QCA memory is volatile, and all stored information will be lost when the power to the circuit is disconnected. In order to control the function of this memory cell, we introduce two controls and one

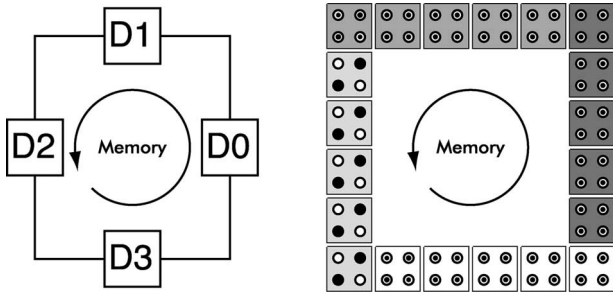


Fig. 26. The most basic memory element in QCA is the loop; as each of the clocks latch and unlatch, they circulate information around the loop.

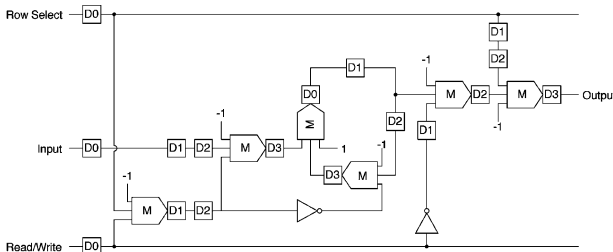


Fig. 27. Schematic of the 1-bit memory cell.

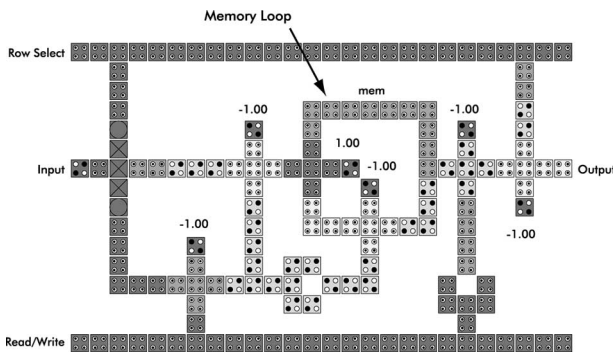


Fig. 28. QCA layout of the memory cell.

input. The first control wire, called the *Row Select*, serves to enable the memory cell to be used in a larger memory grid. The second control, called the *Read/Write*, selects the present operation to be performed on the memory. A schematic representation of the loop memory cell is shown in Fig. 27.

The associated QCA circuit layout is shown in Fig. 28. This layout attempts to maximize the application of this memory cell to larger memories, by running the *Row Select* and *Read/Write* control wires through the entire length of the memory cell. In this way, a row of

memory can be created by simply laying out these cells in a linear array, and extending the control wires to each of the cells.

The stored memory value is constantly circulated inside the memory loop until the *Read/Write* and *Row Select* wires are polarized to logic “1,” at which time the incoming input is fed into the memory loop and circulated. If the *Row Select* is polarized to logic “1,” and the *Write/Read* is polarized to “0,” the current memory value inside the loop is fed to the output. If the *Row Select* is polarized to logic “0,” then the memory cell will always polarize the output cell to logic “0.” Simulation results for the memory cells are shown in Fig. 29. Interestingly, the use of circulating memory storage is not a new concept and can be traced back at least to the mercury delay lines that were used to implement storage in the tube computers of the 1950s.

6) *4-bit Processor*: The most complex circuit designed so far using QCADesigner has been a simple 4-bit processor [48]. This processor was designed mainly as a proof-of-concept to demonstrate that reasonably complex architectures are possible to build using QCA technology, as well as to create a platform for investigating the inherent zone level pipelining. This circuit is intended to demonstrate the level of complexity that can be handled by QCADesigner and not as a demonstration of the ideal architecture for computing using QCA technology. Such architectures are still being developed.

The processor is limited to operate on instructions fed into the circuit directly at the inputs, no program memory is used. The design incorporates a 4×4 RAM to provide temporary storage. The design is based on a simple accumulator architecture shown in Fig. 30.

Basic arithmetic and logic operations are implemented in the arithmetic logic unit (ALU). This ALU is built by extending the 4-bit adder. The ALU layout can be seen in Fig. 31.

An input to the ALU requires 11 clock cycles (44 consecutive clock zones) to propagate through the entire unit. Due to the natural pipelining introduced by the clocking, a

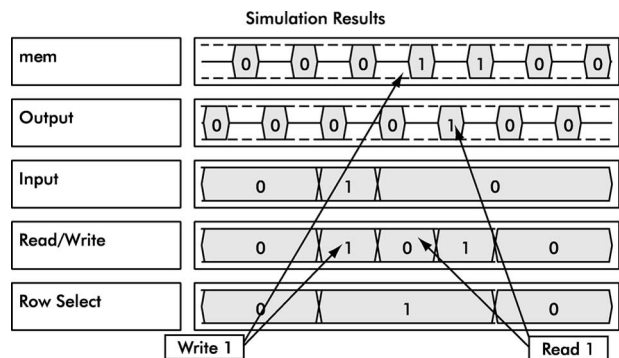


Fig. 29. Simulation results for the memory cell.

new input can be applied to the ALU in each consecutive clock cycle, allowing 11 operations to be in the pipeline at any one time.

In order to provide the processor with data loading and storing capabilities, four 4-bit registers act as physical memory. Instructions are available to store the current accumulator value into any register, and similarly load the contents of any register into the accumulator for computations using the ALU.

Fig. 32 shows the memory layout. It takes four clock cycles for data to be stored in a register, and five clock cycles for data to be available at the output. However, as discussed above, this reduces to one read or write operation per clock cycle in the steady state. The feed-through path, i.e., write, then immediately read, consumes only six clock cycles, since the read operation is not required to wait for the write to complete.

The accumulator is implemented using a 2 : 1 multiplexer with outputs fed back into one of the two input channels. The *Select* control enables the accumulation of the present ALU output. The propagation path in the feedback loop has a two clock cycle pipeline allowing for the accumulation of two values. These two values, however, are accessible every second clock cycle.

Fig. 33 shows the layout of the complete QCA processor.

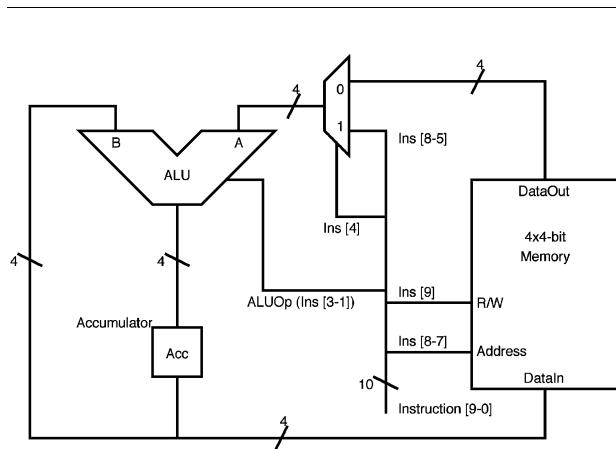


Fig. 30. Processor architecture.

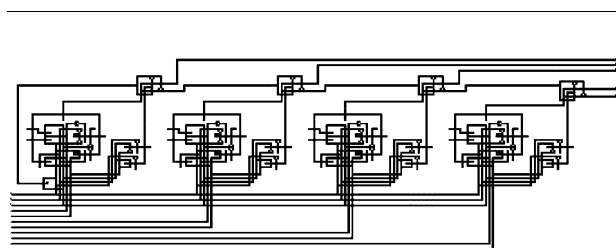


Fig. 31. The 4-bit QCA ALU. The inputs arrive at the lower left corner, and the outputs are read at the upper right.

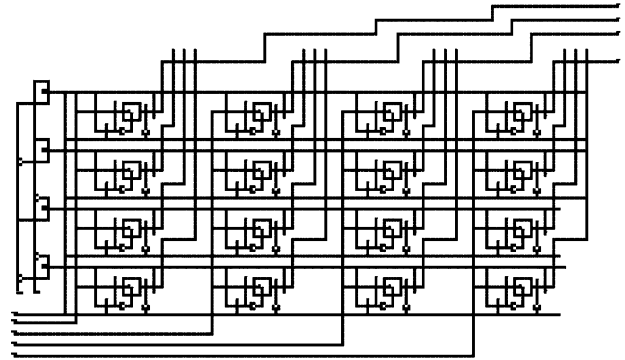


Fig. 32. The 4 × 4-bit memory block. Data enters the block at the lower left, along with the register address and read/write signal. The data is then made available at the upper right.

Each component of the processor, including the interconnect between components, has some latency associated with it, creating a system-wide pipeline. The number of pipeline stages for each component of the processor is shown in Fig. 34.

In order to eliminate the controller logic for this simple design, we have taken advantage of this inherent pipelining. Our instructions contain control bits that determine how each of the processor components handles the arguments associated with that instruction. The control data is distributed to each of the processor components through a set of interconnects, which are clocked such that the control signals arrive synchronously with the associated argument to each of the processor blocks.

This design was successfully simulated using QCADesigner on an Intel Pentium 4 computer system with 2 GB of RAM. The simulation performed the addition of two pieces of data. The result of the addition was then written back to the internal memory. The total simulation time to perform this operation was approximately 5 h on the described system using a time-independent simulation engine. In order to improve simulation performance, significant effort is being invested in optimizing the available simulation engines.

B. Discovering Problems

One of the advantages of creating CAD tools for technologies that have yet to be implemented is the ability to detect problems with theoretical implementation techniques, and to provide, where possible, solutions to these problems. In this subsection we provide an example that relates to the coplanar crossover feature that has been discussed in several earlier papers on QCA [49], [50]. Using the CAD layout and simulation tools in QCADesigner, problems with this building block were identified and a solution was provided and implemented in the tool.

QCA coplanar crossover has been previously proposed as a means of crossing signal wires on a single plane so that

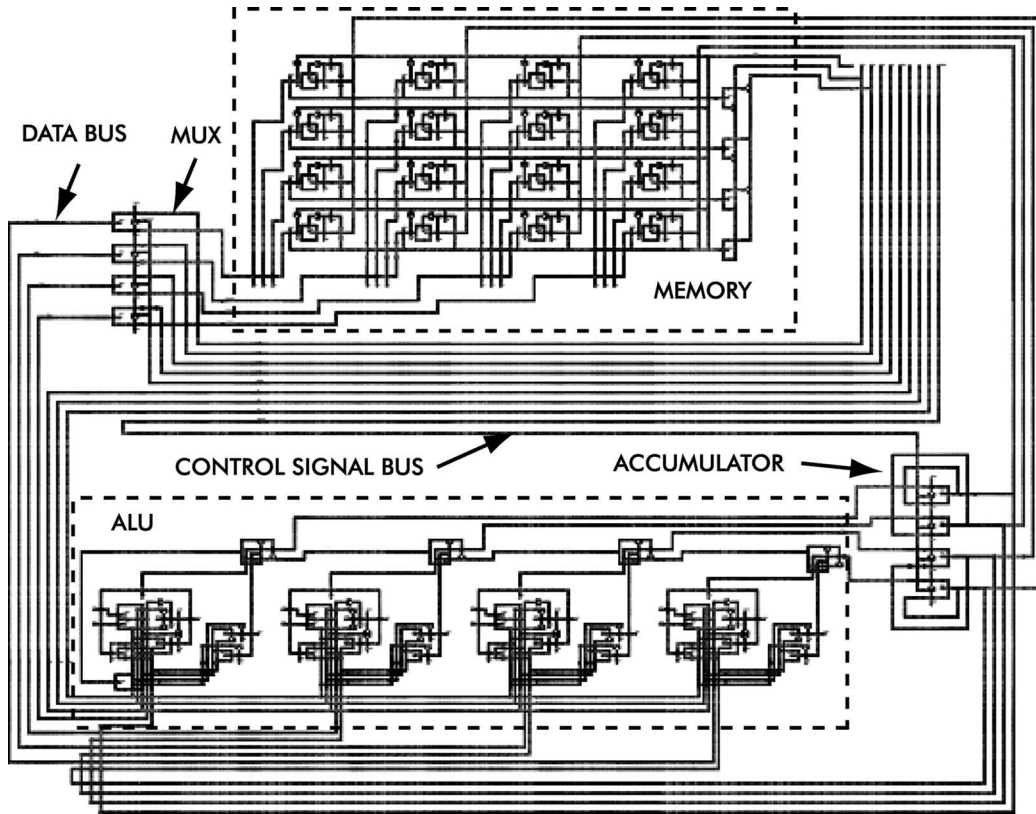


Fig. 33. QCA layout of the 4-bit processor.

completely planar circuits are able to be designed [49]. When placed adjacent to each other, a regular cell, and a cell with the orientation of the dots rotated by 45° (rotated cell) have very little mutual interaction. This property allows the coplanar crossover to transmit information independently along the two different cell wires: one wire composed of regular cells and the other composed of rotated cells, as shown in Fig. 35 [49].

If there is no crosstalk between the two interconnects, a signal is able to propagate across the gap. However, such a system appears to be very sensitive to fabrication variations that break the symmetry required for proper operation [51], [52]. Variations in the position of cells or dots in cells of this building block introduce crosstalk between the two interconnects, which quickly dominate its operation.

To demonstrate a further problem associated with the coplanar crossover, consider the simple circuit shown in Fig. 36. The horizontal wire is sectioned by the three vertical interconnects. Each of these sections are weakly coupled to the other sections of the horizontal wire and very sensitive to random perturbations. In this example, we have placed a fixed cell above the third section of horizontal wire. The third section, as well as the one before it, takes on the polarization of the fixed cell and not that of the input.

Since this type of circuit block appears everywhere in a complex design, it is unlikely that coplanar crossovers will be a building block of future QCA circuits using noncontinuous clocking. The ability to cross signals is essential for circuit design. If the coplanar crossover is not

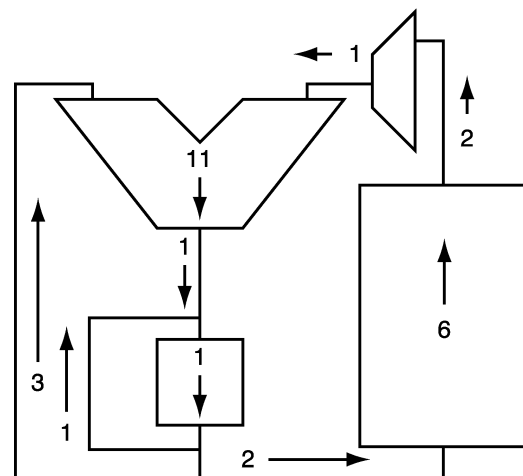


Fig. 34. Number of clock cycles through each stage of the processor.

a functional building block, then given the constraints of the technology, we must consider alternatives for crossing signals. Previous work has examined the possibility of multilayer QCA [53]. Using these multilayer QCA cells, we can effectively cross signals over on another layer as shown previously in Fig. 16. Unlike present CMOS ICs, where metal layers are used to connect discontinuous sections of a circuit and cannot perform any intelligent functions, the extra layers of QCA can be used as active components of the circuit. In this way, we believe that multilayer QCA circuits can potentially consume much less area as compared to planar circuits.

VII. POTENTIAL QCA TECHNOLOGIES

To date, QCA is a commercially unrealized technology. However, several experimental devices have been fabricated [6], [15], [25], [26]. The different proposed implementations attempt to realize the required bistable and locally interacting behavior required by the QCA paradigm. Of all the implementations, four distinct classes of QCA have clearly emerged. These are:

- metal-island [6], [15], [25], [26];
- semiconductor [54], [55];
- molecular [11], [56]–[60];
- magnetic [7], [61], [62].

Researchers investigating QCA implementations are currently focusing on what could be considered the “holy grail” of QCA implementations: molecular QCA. A

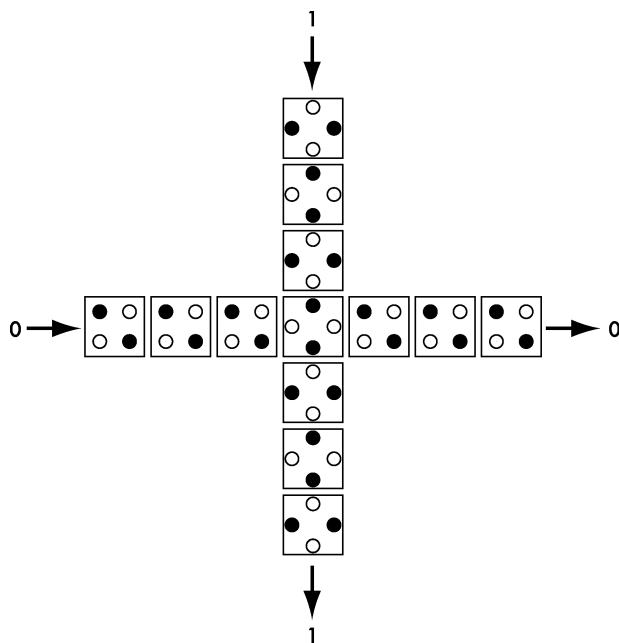


Fig. 35. A crossover assumes that there is very little interaction between the vertical and horizontal wires.

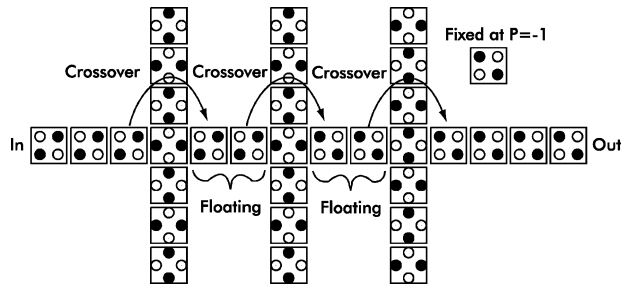


Fig. 36. A section of wire with three coplanar crossovers; the last two sections of horizontal wire take on the polarization of the fixed cell rather than the polarization at the input.

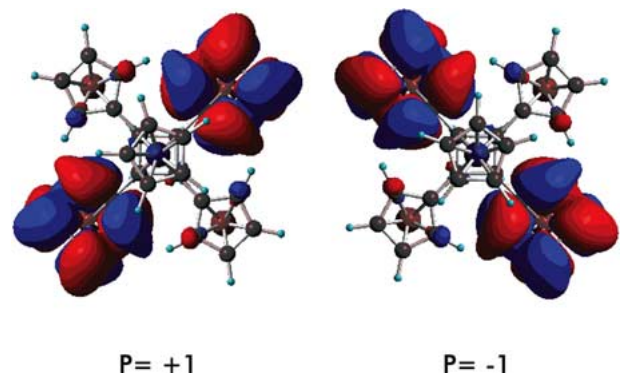


Fig. 37. Four-dot molecular QCA cell [64]. With kind permission of Springer Science and Business Media.

proposed molecular QCA cell with four electronic sites is shown in Fig. 37.

The molecules are bonded to the substrate surface at adjacent sites allowing for electrostatic interactions to induce the free electrons in the molecule to switch position between available redox sites. The application of the QCA paradigm to molecular electronics appears to be promising, as molecules are very good charge containers and the interactions have sufficiently high energy to ensure room temperature operation; undoubtedly one of the requirements of future nanotechnology SoC solutions.

VIII. DISCUSSION AND CONCLUSION

In this paper we have provided an initial exploration of a design tool for an emerging nanoelectronic technology: QCA. The premise of the paper is that it is beneficial to use design and simulation tools to explore disruptive technologies, before they are commercialized, in order to determine their efficacy as both enhancements and replacements for current SoC technologies. Our approach in this paper, has been to concentrate on one of the

promising disruptive technologies (QCA), to provide sufficient details of the technology and the design tool, and to extend to the reader the opportunity to evaluate a freely available design tool in order to demonstrate the advantages of this approach for the development of future SoC technologies.

Our exploration has covered the basic theory of QCA including a variety of cellular architectures, logic elements, and processor structures. We have also provided some detail about the CAD tool, QCADesigner, and the two simulation engines that were developed that allow a tradeoff between accuracy and computational complexity. The paper is illustrated with several examples of QCA logic and processor architecture designed using QCADesigner, including a demonstration of the ability of a good simulation tool in providing evidence of potential failure of the technology and solutions that can be explored to overcome these failures. We have also provided a roundup of the most promising QCA implementation technologies.

Emerging nanotechnologies that are being, and will be, explored as replacements for current FET-based technologies, may be our only hope of continuing along the Moore's Law exponential path and providing the antici-

pated increases in circuit density that define our concept of SoC. Several promising emerging technologies have already been identified and others will undoubtedly appear over the next decade or so. Whichever technologies prove to be successful replacements, and there may be several that are finally used in a mixed technology scenario, we will need to explore them thoroughly in a sophisticated design and simulation environment. Prior to commercial fabrication success of any new technology, we will need to evaluate, make robust, and be ready for the assimilation of that technology into the fabrication market place. This can only be accomplished through the use of design and simulation tools that accurately predict the complete nanoenvironment within which the new technology will work. ■

Acknowledgment

The authors would like to thank iCore, the Natural Sciences and Engineering Research Council of Canada, Micronet R&D (a former Canadian Network of Centres of Excellence), and CMC Microsystems for their support.

REFERENCES

- [1] International Technology Roadmap for Semiconductors. (ITRS). [Online]. Available: <http://public.itrs.net>, 2004.
- [2] H. Chang, L. R. Cooke, M. Hunt, G. Martin, A. McNelly, and L. Todd, *Surviving the SOC Revolution: A Guide to Platform-Based Design*. Berlin, Germany: Springer-Verlag, 2000.
- [3] C. S. Lent, "Quantum cellular automata," *Nanotechnology*, vol. 4, pp. 49–57, 1993.
- [4] C. S. Lent and P. D. Tougaw, "A device architecture for computing with quantum dots," *Proc. IEEE*, vol. 85, no. 4, pp. 541–557, Apr. 1997.
- [5] P. D. Tougaw and C. S. Lent, "Dynamic behavior of quantum cellular automata," *J. Appl. Phys.*, vol. 80, no. 8, pp. 4722–4735, Oct. 1996.
- [6] R. K. Kummamuru, A. O. Orlov, R. Ramasubramaniam, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Operation of a quantum-dot cellular automata (QCA) shift register and analysis of errors," *IEEE Trans. Electron Devices*, vol. 50, no. 9, pp. 1906–1913, Sep. 2003.
- [7] C. György et al. "Nanocomputing by field-coupled nanomagnets," *IEEE Trans. Nanotechnol.*, vol. 1, no. 4, pp. 209–213, Dec. 2002.
- [8] J. Timler and C. S. Lent, "Power gain and dissipation in quantum-dot cellular automata," *J. Appl. Phys.*, vol. 91, no. 2, pp. 823–831, Jan. 2002.
- [9] —, "Maxwell's demon and quantum-dot cellular automata," *J. Appl. Phys.*, vol. 94, no. 2, pp. 1050–1060, Jul. 2003.
- [10] G. Toth and C. S. Lent, "Quasiadiabatic switching of metal-island quantum-dot cellular automata," *J. Appl. Phys.*, vol. 85, no. 5, pp. 2977–2984, 1999.
- [11] K. Hennessy and C. S. Lent, "Clocking of molecular quantum-dot cellular automata," *J. Vac. Sci. Technol.*, vol. B19, no. 5, pp. 1752–1755, 2001.
- [12] E. P. Blair and C. S. Lent, "An architecture for molecular computing using quantum-dot cellular automata," in *Proc. 3rd IEEE Conf. Nanotechnology*, 2003, pp. 402–405.
- [13] K. Walus, G. A. Jullien, and V. Dimitrov, "Computer arithmetic structures for quantum cellular automata," in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, 2003, pp. 1435–1439.
- [14] A. O. Orlov, R. K. Kummamuru, R. Ramasubramaniam, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Clocked quantum-dot cellular automata shift register," *Surf. Sci.*, vol. 532–535, pp. 1193–1198, 2003.
- [15] G. H. Bernstein, I. Amlani, A. O. Orlov, C. S. Lent, and G. L. Snider, "Observation of switching in a quantum-dot cellular automata cell," *Nanotechnology*, vol. 10, pp. 166–173, 1999.
- [16] Y. Wan, K. Huang, S. F. Hu, C. L. Sung, and Y. C. Chou, "Coulomb blockade oscillations in ultrathin gate oxide silicon single-electron transistors," *J. Appl. Phys.*, vol. 97, no. 11, p. 116106, 2005.
- [17] T. M. Buehler, D. J. Reilly, R. P. Starrett, A. D. Greentree, A. R. Hamilton, A. S. Dzurak, and R. G. Clark. (2006, June). Single-shot readout with the radio frequency single electron transistor in the presence of charge noise ArXiv. [Online]. Available: <http://arxiv.org/abs/condat/0304384>
- [18] R. Zhang, P. Gupta, L. Zhong, and N. K. Jha, "Threshold network synthesis and optimization and its application to nanotechnologies," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 1, pp. 107–118, Jan. 2005.
- [19] K. Walus, G. Schulhof, R. Zhang, W. Wang, and G. A. Jullien, "Circuit design based on majority gates for applications with quantum-dot cellular automata," in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, 2004, pp. 1354–1357.
- [20] K. Walus, M. Mazur, G. Schulhof, G. A. Jullien, and V. Dimitrov, "Quantum cellular automata: using cad to investigate a future nano-electronic technology," presented at the *Micronet Annu. Workshop*, Ottawa, ON, Canada, 2005.
- [21] K. Walus, T. Dysart, G. A. Jullien, and R. A. Budiman, "QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 3, no. 1, pp. 26–31, Mar. 2004.
- [22] K. Walus, V. Dimitrov, G. A. Jullien, and W. C. Miller, "QCADesigner: A CAD tool for an emerging nano-technology," in *Micronet Annu. Workshop*, Toronto, ON, Canada, 2003. [Online]. Available: <http://www.qcadesigner.ca/papers/micronet2003.pdf>
- [23] K. Walus, T. Dysart, G. A. Jullien, and R. A. Budiman, "QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata," presented at the *2nd Int. Workshop Quantum Dots for Quantum Computing and Classical Size Effect Circuits*, Notre Dame, IN, 2003.
- [24] K. Walus, A. Vetteth, G. A. Jullien, and V. Dimitrov, "Design and simulation of quantum-dot cellular automata," presented at the *Symp. Microelectronics Research and Development in Canada*, Ottawa, ON, Canada, 2002.
- [25] I. Amlani, A. O. Orlov, R. K. Kummamuru, G. H. Bernstein, C. S. Lent, and G. L. Snider, "Experimental demonstration of leadless quantum-dot cellular automata cell," *Appl. Phys. Lett.*, vol. 77, no. 5, pp. 738–740, 2000.

- [26] A. O. Orlov, R. K. Kumamuru, R. Ramasubramaniam, G. Toth, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Experimental demonstration of a latch in clocked quantum-dot cellular automata," *Appl. Phys. Lett.*, vol. 78, no. 11, pp. 1625–1627, 2001.
- [27] G. Tóth, "Correlation and coherence in quantum-dot cellular automata," Ph.D. dissertation, Univ. Notre Dame, Notre Dame, IN, 2000.
- [28] G. Tóth and C. S. Lent, "Role of correlation in the operation of quantum-dot cellular automata," *J. Appl. Phys.*, vol. 89, no. 12, pp. 7943–7953, Jun. 2001.
- [29] C. S. Lent, P. D. Tougaw, and W. Porod, "Bistable saturation in coupled quantum dots for quantum cellular automata," *Appl. Phys. Lett.*, vol. 62, no. 7, pp. 714–716, Feb. 1993.
- [30] C. S. Lent and P. D. Tougaw, "Lines of interacting quantum-dot cells: a binary wire," *J. Appl. Phys.*, vol. 74, no. 10, pp. 6227–6233, Nov. 1993.
- [31] P. D. Tougaw, C. S. Lent, and W. Porod, "Bistable saturation in coupled quantum-dot cells," *J. Appl. Phys.*, vol. 74, no. 5, pp. 3558–3565, Sep. 1993.
- [32] S. Bhanja and S. Sarkar, "Graphical probabilistic inference for ground state and near-ground state computing in QCA circuits," in *Proc. IEEE Conf. Nanotechnology*, 2005, pp. 290–293.
- [33] M. B. Tahoori, M. Momenzadeh, J. Huang, and F. Lombardi, "Defects and faults in quantum cellular automata at nano scale," in *Proc. 22nd IEEE VLSI Test Symp.*, 2004, pp. 291–296.
- [34] T. Wei, K. Wu, R. Karri, and A. Orailoglu, "Fault tolerant quantum cellular array (QCA) design using triple modular redundancy with shifted operands," in *Proc. Asia and South Pacific Design Automation Conf.*, 2005, pp. 1192–1195.
- [35] W. J. Townsend and J. A. Abraham, "Complex gate implementations for quantum dot cellular automata," in *Proc. IEEE Conf. Nanotechnology*, 2004, pp. 625–627.
- [36] J. Huang, M. Momenzadeh, M. B. Tahoori, and F. Lombardi, "Defect characterization for scaling of QCA devices," in *Proc. 19th IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems*, 2004, pp. 30–38.
- [37] T. J. Dysart and P. M. Kogge, "Strategy and prototype tool for doing fault modeling in a nano-technology," in *Proc. IEEE Conf. Nanotechnology*, 2003, pp. 356–359.
- [38] J. Huang, M. Momenzadeh, M. B. Tahoori, and F. Lombardi, "Design and characterization of an and-or-inverter (AOI) gate for QCA implementation," in *Proc. ACM Great Lakes Symp. VLSI*, 2004, pp. 426–429.
- [39] M. B. Tahoori and F. Lombardi, "Testing of quantum dot cellular automata based designs," in *Proc. Design, Automation and Test in Europe Conf. and Exhibition*, 2004, pp. 1408–1409.
- [40] M. Momenzadeh, M. B. Tahoori, J. Huang, and F. Lombardi, "Quantum cellular automata: New defects and faults for new devices," in *Proc. Parallel and Distributed Processing Symp.*, 2004, p. 207a.
- [41] K. Kim, K. Wu, and R. Karri, "Towards designing robust QCA architectures in the presence of sneak noise paths," in *Proc. Design Automation and Test in Europe*, 2005, vol. 2, pp. 1214–1219.
- [42] S. K. Lim, R. Ravichandran, and M. Niemier, "Partitioning and placement for buildable QCA circuits," *J. Emerg. Technol. Comput. Syst.*, vol. 1, no. 1, pp. 50–72, 2005.
- [43] K. Walus and G. Schulhof, *QCADesigner homepage*. [Online]. Available: <http://www.qcadesigner.ca/>
- [44] E. P. Blair, "Tools for the design and simulation of clocked molecular quantum-dot cellular automata circuits," M.S. thesis, Univ. Notre Dame, Notre Dame, IN, 2003.
- [45] W. Wang, K. Walus, and G. A. Jullien, "Quantum-dot cellular automata adders," in *Proc. IEEE Conf. Nanotechnology*, 2003, pp. 461–464.
- [46] A. Vetteh, K. Walus, V. Dimitrov, and G. A. Jullien, "Quantum-dot cellular automata carry-look-ahead adder and barrel shifter," in *Proc. IEEE Emerging Telecommunications Technologies Conf.*, 2002, pp. 2/1–2/4.
- [47] A. Vetteh, K. Walus, G. A. Jullien, and V. Dimitrov, "RAM design using quantum-dot cellular automata," in *Proc. Nanotechnology Conf. and Tradeshow*, 2003, vol. 2, pp. 160–163.
- [48] K. Walus, M. Mazur, G. Schulhof, and G. A. Jullien, "Simple 4-bit processor based on quantum-dot cellular automata (QCA)," in *Proc. Application Specific Architectures, and Processors Conf.*, 2005, pp. 288–293.
- [49] P. D. Tougaw and C. S. Lent, "Logical devices implemented using quantum cellular automata," *J. Appl. Phys.*, vol. 75, no. 3, pp. 1818–1825, Feb. 1994.
- [50] J. R. Janulis, P. D. Tougaw, S. C. Henderson, and E. W. Johnson, "Serial bit-stream analysis using quantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 3, no. 1, pp. 158–164, Mar. 2004.
- [51] K. Walus, G. A. Jullien, and R. A. Budiman, "QCA co-planar wire-crossing and multi-layer networks," presented at the *iCore Banff Summit*, Banff, AB, Canada, 2004.
- [52] K. Walus, G. Schulhof, and G. A. Jullien, "High level exploration of quantum-dot cellular automata (QCA)," in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, 2004, vol. 1, pp. 30–33.
- [53] A. Gin, P. D. Tougaw, and S. Williams, "An alternative geometry for quantum-dot cellular automata," *J. Appl. Phys.*, vol. 85, no. 12, pp. 8281–8286, Jun. 1999.
- [54] C. Ungarelli, S. Francaviglia, M. Macucci, and G. Iannaccone, "Thermal behavior of quantum cellular automaton wires," *J. Appl. Phys.*, vol. 87, no. 10, pp. 7320–7325, 2000.
- [55] M. Macucci, G. Iannaccone, S. Francaviglia, and B. Pellegrini, "Semiclassical simulation of quantum cellular automaton circuits," *Int. J. Circuit Theory Appl.*, vol. 29, no. 1, pp. 37–47, Jan. 2001.
- [56] C. S. Lent, B. Isaksen, and M. Lieberman, "Molecular quantum-dot cellular automata," *J. Amer. Chem. Soc.* vol. 125, pp. 1056–1063, 2003.
- [57] C. S. Lent and B. Isaksen, "Clocked molecular quantum-dot cellular automata," *IEEE Trans. Electron Devices*, vol. 50, no. 9, pp. 1890–1896, Sep. 2003.
- [58] Z. Li and T. P. Fehlner, "Molecular qca cells. 2. characterization of an unsymmetrical dinuclear mixed-valence complex bound to a surface by an organic linker," *Inorg. Chem.*, vol. 42, no. 18, pp. 5715–5721, 2003.
- [59] J. Jiao, G. J. Long, F. Grandjean, A. M. Beatty, and T. P. Fehlner, "Building blocks for the molecular expression of quantum cellular automata. isolation and characterization of a covalently bonded square array of two ferrocenium and two ferrocene complexes," *J. Amer. Chem. Soc.*, vol. 125, no. 25, pp. 7522–7523, 2003.
- [60] Y. Wang and M. Lieberman, "Thermodynamic behavior of molecular-scale quantum-dot cellular automata (QCA) wires and logic devices," *IEEE Trans. Nanotechnol.*, vol. 3, no. 3, pp. 368–376, Sep. 2004.
- [61] A. Imre, G. Csaba, L. Ji, A. Orlov, G. H. Bernstein, and W. Porod, "Majority logic gate for magnetic quantum-dot cellular automata," *Science*, vol. 311, no. 5758, pp. 205–208, 2006.
- [62] M. E. Welland and R. P. Cowburn, "Room temperature magnetic quantum cellular automata," *Science*, vol. 287, pp. 1466–1468, 2000.
- [63] M. C. B. Parish, "Modeling of physical constraints on bistable magnetic quantum cellular automata," Ph.D. dissertation, Univ. London, London, U.K., 2003.
- [64] Y. Lu and C. S. Lent, "Theoretical study of molecular quantum-dot cellular automata," *J. Comput. Electron.*, vol. 4, pp. 115–118, 2005.

ABOUT THE AUTHORS

Konrad Walus (Member, IEEE) received the degree in electrical engineering from the University of Windsor, Windsor, ON, Canada, in 2001 and the Ph.D. degree in electrical engineering from the University of Calgary, Calgary, AB, Canada, in 2005.

He was a Student Systems Engineer at AgentWare Systems Incorporated Canada, from 2000 to 2001. Konrad started as an Assistant Professor in the Department of Electrical and Computer Engi-



neering, University of British Columbia, Vancouver, BC, Canada, in September 2005. His primary research focus is on modeling and computer-aided design of emerging nanotechnologies with a primary focus on quantum-dot cellular automata (QCA).

Dr. Walus has received several awards including the National Science and Engineering Council Scholarship, iCORE scholarship, University of Calgary Dean's Research award, and the Micralyne Microsystems Design Award. He has recently been named the recipient of the Alberta Science and Technology Leaders of Tomorrow award.

Graham A. Jullien (Fellow, IEEE) receiving the B.Tech. degree in electrical engineering, from the University of Loughborough, Loughborough, U.K., in 1965, the M.Sc. degree from the University of Birmingham, Birmingham, U.K., in 1967, and the Ph.D. degree from the Aston University, Birmingham, in 1969.



From 1961 to 1966, he was a Student Engineer and Data Processing Engineer at English Electric Computers, Kidsgrove, U.K. From 1975 to 1976, he was a Visiting Senior Research Engineer at the Central Research Laboratories of EMI Ltd., Hayes, U.K. From 1969 to 2000, he was with the Department of Electrical and Computer Engineering at the University of Windsor, Windsor, ON, Canada, where he held the rank of University Professor and was the Director of the VLSI Research Group. Since January 2001, he has been with the Department of Electrical and Computer

Engineering at the University of Calgary, Calgary, AB, Canada, where he holds the iCORE Research Chair in Advanced Technology Information Processing Systems. He has published widely in the fields of digital signal processing, computer arithmetic, neural networks, and VLSI systems, and teaches courses in related areas. He has served on the technical committees of many international conferences, and he currently serves on the Editorial Board of the *Journal of VLSI Signal Processing*.

Dr. Jullien is a member of the Board of Directors of CMC Microsystems, DALSA Inc., and Micronet R&D (a Network of Centres of Excellence). He is a past Associate Editor of the IEEE TRANSACTIONS ON COMPUTERS. He hosted and was program cochair of the 11th IEEE Symposium on Computer Arithmetic, was program chair for the 8th Great Lakes Symposium on VLSI, technical and general program chair for the Asilomar Conference on Signals, Systems and Computers in 1999 and 2003, respectively. He was general cochair of the International Workshop on System-on-Chip for Real-Time Systems, Calgary, 2003-2005.