

A Shift-Register-Based QCA Memory Architecture

BARIS TASKIN, ANDY CHIU, JONATHAN SALKIND, and DANIEL VENUTOLO
Drexel University

A quantum-dot cellular automata (QCA) design of an $n \times m$ -bit, shift-register-based memory architecture is presented. The architecture maintains data at a stable conformation, which is contrary to traditional data in-motion concept for QCA architectures. The memory architecture is based on an existing dual-phase-synchronized, line-based, one-bit QCA memory cell building block that provides size and latency improvements over other known one-bit memory cells through its novel clocking scheme. Read/write latencies up to $\sim 2X$ lower than the existing tile-based architecture with three-phase, line-based memory cells are obtained. Simulations with QCADesigner and HDLQ are performed on a sample 4×8 bit memory architecture implementation.

Categories and Subject Descriptors: B.6.1 [**Logic Design**]: Design styles—*Cellular array and automata*

General Terms: Algorithms, Experimentation, Theory

Additional Key Words and Phrases: Quantum-dot cellular automata, clocking, memory design

ACM Reference Format:

Taskin, B., Chiu, A., Salkind, J., and Venutolo, D. 2009. A shift-register-based QCA memory architecture. ACM J. Emerg. Technol. Comput. Syst. 5, 1, Article 4 (January 2009), 18 pages. DOI = 10.1145/1482613.1482617 <http://doi.acm.org/10.1145/1482613.1482617>

1. INTRODUCTION

Quantum-dot cellular automata (QCA) is a potentially promising technology as an alternative to complementary-metal-oxide semiconductor (CMOS) technology for nanoscale device implementations. The implementation of the QCA technology has been demonstrated with metal-dot QCA devices at very low (e.g., cryogenic) temperatures [Snider et al. 1999]. Circuit structures such as the majority gate, binary wires and fanouts have been fabricated with metal-QCA dots [Orlov et al. 1999; Amlani 1999; Kummaruru et al. 2002; Yadavalli

This article originally appeared in the *Proceedings of the 2007 IEEE/ACM International Symposium on Nanoscale Architectures*.

Author's address: B. Taskin, Electrical and Computer Engineering Department, Drexel University, Philadelphia, PA 19104.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org. © 2009 ACM 1550-4832/2009/01-ART4 \$5.00. DOI 10.1145/1482613.1482617 <http://doi.acm.org/10.1145/1482613.1482617>

ACM Journal on Emerging Technologies in Computing Systems, Vol. 5, No. 1, Article 4, Pub. date: January 2009.

et al. 2005]. Recently, research in molecular QCA implementations have been emerging through advances in DNA tiling and molecular self-assembly [Hu et al. 2005]. Molecular QCA technology is projected to operate at the room temperature with much higher functional densities [Hennessy and Lent 2001; Lent and Isaksen 2003; Amlani et al. 2003; Lent et al. 2003; Huang et al. 2005].

The manufacturing and assembly of quantum-dot devices from a physics and chemistry point of view is the pillar to the emergence of QCA technology as a viable alternative to CMOS nanoscale device implementations. While some QCA research has to be performed specific to the metallic or molecular QCA manufacturing data (such as power dissipation research), general *nanoarchitecture design* studies can be performed by modeling QCA-dots at a higher level of abstraction, relatively independent of the manufacturing technology. In the presented work, one such study is performed, which involves the *automata* design of a novel QCA-based memory architecture using the logical representation of quantum dots. Such logical representation of quantum dots is advantageous in terms of characterizing QCA nanoarchitectures independent of the manufacturing technology, which enables a systematic analysis.

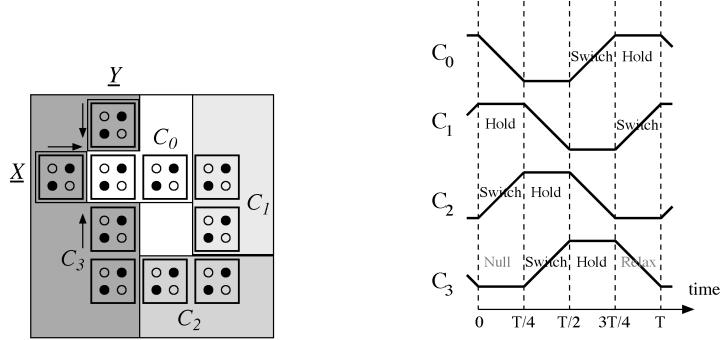
Memory design in QCA logic has been a challenge¹ due to the locality of computation and constant propagation of data in cellular automata. Alternative memory cell and architecture implementations have been proposed (reviewed in Section 2), which provide different performance as well as operational characteristics. The research presented in this article is performed on the following structures:

- Two types of line-topology-based memory cells: the three-phase memory cell presented in Vankamamidi et al. [2005a] and the dual-phase memory cell presented in Taskin and Hong [2006],
- Two memory architectures: the tile-based architecture presented in Vankamamidi et al. [2005b] and Ottavi et al. [2005] and the novel shift-register-based architecture presented in this article.

The two types of line-based memory cells are similar in topology and design, and differ from other memory and logic cells in promoting the use of alternative clocking schemes in addition to the four-phase clocking scheme conventionally used in QCA operation. Thanks to the alternative clocking schemes, significant performance improvements in read/write latency (in the number of clock cycles) and cell area (in the number of clocking zones) are achievable. For instance, a tile-based $n \times m$ bit memory architecture design is presented in Vankamamidi et al. [2005b] and Ottavi et al. [2005] that uses the three-phase line-based memory cell presented in Vankamamidi et al. [2005a] with the unconventional three phase clocking scheme. This tile-based memory architecture provides conventional memory operation with a denser bit-storage area than previously offered QCA memory architectures.

The novelty of this paper is a shift-register-based, $n \times m$ memory architecture that utilizes dual-phase, line-based memory cell presented in Taskin and Hong

¹This challenge is pronounced from the suggested *automata design* point of view. Memory design is currently also a challenge from a manufacturing point of view, as is any other complex QCA system.



(a) QCA implementation where X and Y are control/data signals and the data is stored traveling around the feedback loop. Clock zones are color coded.

(b) Traditional four-phase clocking strategy.

Fig. 1. Four-phase, feedback-topology memory cell.

[2006] with the unconventional dual-phase clocking scheme. Similar to the tile-based architecture, the design density is much improved over previous QCA memory architectures. Furthermore, read/write latency of the proposed shift-register architecture is up to $\sim 2X$ lower than the previously proposed tile-based QCA memory architecture.

The rest of this article is organized as follows. In Section 2, a brief survey of previous research on QCA memory design is presented. In Section 3, the three-phase and dual-phase line based memory cells and the tile-based architecture for the three-phase memory cells are reviewed. In Section 4, the proposed shift-register based memory architecture for the dual-phase line-based memory is presented. In Section 5, simulation results using the popular QCA design simulator QCADesigner and the design language HDLQ are presented. In Section 6, performance characteristics of the presented memory architecture are analyzed. Finally, conclusions are offered in Section 7.

2. QCA MEMORY DESIGN OVERVIEW

Previously offered QCA memory architectures can be categorized as serial-access [Berzon and Fountain 1999; Frost et al. 2002, 2003] and parallel-access memories [Walus et al. 2003; Ottavi et al. 2005; Vankamamidi et al. 2005a]. A serial-access QCA memory has been presented as part of the SQUARES formalism in Berzon and Fountain [1999]. In Berzon and Fountain [1999], data storage is performed through a feedback topology cell, which is illustrated in Figure 1. The feedback provides the typical *unidirectional* path for propagation of the stored data value over the four consecutive clocking zones of the traditional QCA clocking strategy.

The read/write circuitry can be quite complex for serial access memories. A particular type of serial-access memory, the H-memory presented in Frost et al. [2002], incorporates thread bouncing computation method [Frost et al. 2003] in order to reduce this complexity. H-memory provides unique operation and

computation practices, within which packets of instructions and data are sent through the H-tree architecture for local computations.

The parallel access QCA memory architecture presented in Walus et al. [2003] emulates the CMOS RAM design by using the QCA feedback topology memory cell (Figure 1) for data storage. This simple design operates with the four-phase clocking scheme, not requiring any additional CMOS clock generators. However, complex clocking zones (not row or column stackable) are necessary due to the clock zone requirements of the feedback-topology memory cell. Additionally, synchronization with the traditional four-phase clocking scheme mandates a *unidirectional* data flow, limiting the exploration of performance through the *bidirectional* data flow advocated in the design of the line-based memory cells.

3. LINE-BASED MEMORY CELLS

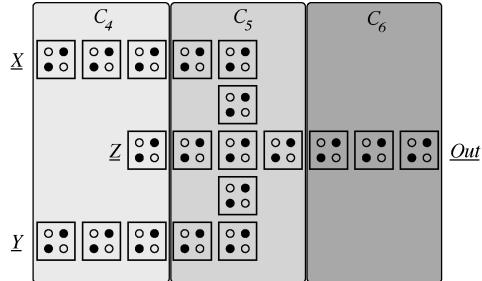
Line-based memory cells are built on the premise of bidirectional data flow. Note that, bidirectional data flow is not possible with the conventional four-phase clocking scheme. By defining additional clock phases, however, some combination (two or more) of clock zones can create flows in one direction at certain durations of their periodic clocking cycle whereas the direction can be reversed at other durations. In other words, two adjacent clock zones that are in hold and switch phases (respectively) at one time can later be observed in the switch and hold phases, respectively.

The two types of line-based memory, the three-phase and the dual-phase memory, are reviewed in Sections 3.1 and 3.2, respectively.

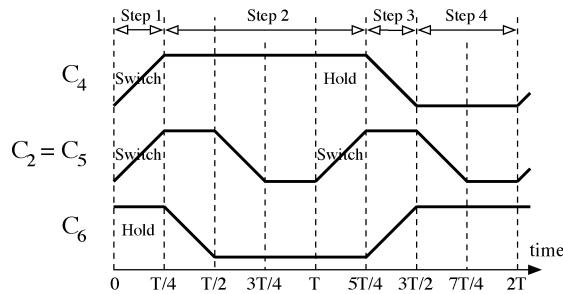
3.1 Three-Phase Line-Based Memory

The three-phase, line-based memory cell [Vankamamidi et al. 2005a], shown in Figure 2, is the first study to exploit non-standard clocking strategies in QCA operation. This cell consists of three clock zones C_4, C_5, C_6 , where the four conventional clocking zones in the *rest* of the circuit are labeled C_0, C_1, C_2, C_3 . Clock phase C_5 is identical to one of the clock phases of the four-phase clocking scheme, namely C_2 in Figure 1(b). Thus, unlike C_4 and C_6 , the clock phase C_5 does not require an additional clock source. Consequently, the three-phase clocking scheme requires two additional clock generator circuits. Clock signals to zones 4, 5, and 6 (synchronized by phases $C_4, C_2 = C_5$, and C_6 , respectively) all follow the hold-release-null-switch cycle. The clock phase of zone 5 is specially designed such that when either zones 4 or 6 is in the hold phase, zone 5 is in the switch phase, enabling bidirectional data flow—and storage—on the memory *line* between Z and Out. The read/write latency of the three-phase cell is equal to the period $2T$ of the synchronization scheme (where T is the period of the conventional four-phase clocking scheme).

Tile-based memory architecture, shown in Figure 3, is proposed in Vankamamidi et al. [2005b] and Ottavi et al. [2005] as an architecture that employs the three-phase, line-based memory cell. This architecture is composed of one input tile (on the left), three internal tiles (in the middle), and one output tile (on the right). When tiles are placed adjacently, a major loop for data



(a) QCA implementation where \underline{X} and \underline{Y} are control/data signals and data is stored traveling along the line \underline{Z} -Out.



(b) Three-phase clocking scheme for the line memory.

Fig. 2. Line-based memory with three-phase clocking.

propagation is constructed. Two bits of data are stored in each tile, one on the upper, one on the lower propagation path. In each internal tile, the three-phase, line-based memory cell shown in Vankamamidi et al. [2005a] is used as the bit storage mechanism.

In an m -bit row of the tile-based architecture memory, data bits are stored circulating on a closed feedback loop as shown for $m = 3$ in Figure 3(b). Thus, when data needs to be read or written, an external counter of $\log m$ bits is used to synchronize the writing/reading head with the requested data bit location (discussed but not implemented in Vankamamidi et al. [2005b] and Ottavi et al. [2005]). Such operation leads to variable read/write latencies, as initiating a memory access operation may take anywhere between 0 to $m - 1$ cycles, followed by m cycles for the read/write operation totaling up to a latency between m and $2m - 1$.

3.2 Dual-Phase Line-Based Memory

An alternative implementation of the line-based memory with dual-phase synchronization, shown in Figure 4, is presented in Taskin and Hong [2006]. This cell consists of two clocking zones C_6 , C_7 and consequently requires two clock phases (in addition to the four clock phases C_0 , C_1 , C_2 , C_3 in the rest of the circuit). The two clock phases, C_7 and C_8 , are generated by a single clock generator and a phase shifter. In operation, Zone 7 switches when zone 8 is holding (step 1)

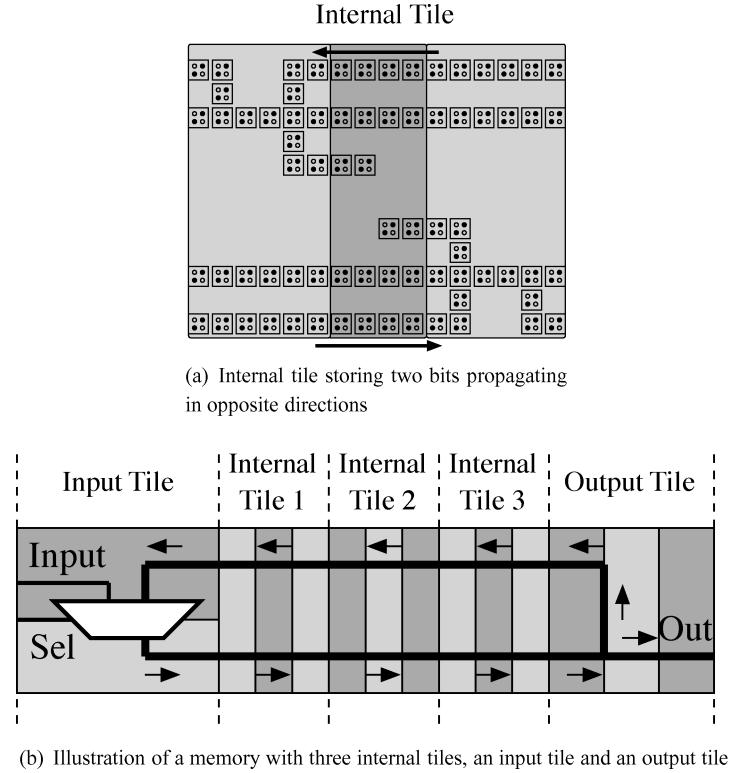


Fig. 3. Tile-based memory architecture.

and Zone 8 switches when zone 7 is holding (step 2), creating the bidirectional data flow. Consequently, data written to the cell moves back and forward on the memory *line* (between Z and Out, bidirectionally), constituting the storage mechanism. The read/write latency of the three-phase cell is equal to the period $3T/2$ of the synchronization scheme (where T is the period of the conventional four-phase clocking scheme). In order to ensure proper synchronization with the surrounding circuitry (which is typically synchronized with the traditional four-phase clocking scheme with a clock period of T), the $3T/2$ period of two-phase memory is extended to $2T$. Under such a synchronization scheme, the write latency is extended to $2T$ (to be identical to three-phase cell), whereas the read latency is improved to T , leading to 2X faster read operations than the three-phase cell. The differences between the dual and three-phase line-based memory cells are categorized in Table I.

In order to observe the operational characteristics of the dual-phase line-based memory cell in a *multirow* (stacked) architecture, a shift-register-based memory architecture is proposed.

4. SHIFT-REGISTER-BASED MEMORY ARCHITECTURE

The building block of the presented shift-register-based memory architecture is shown in Figure 5. The data storage in the building block is established with

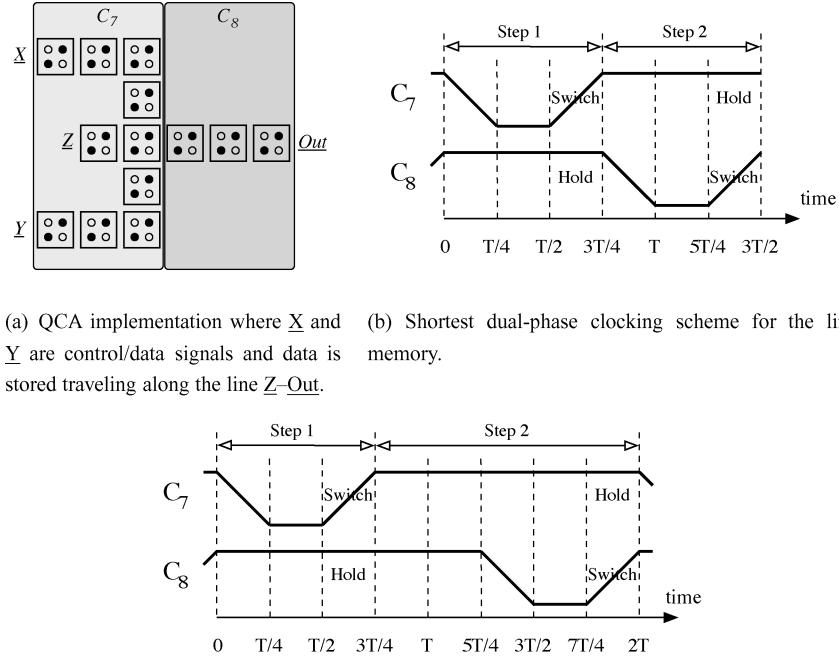


Fig. 4. Dual-phase line-based memory cell.

Table I. Design and Performance Comparison of the Three-Phase and Dual-Phase Line-Based Memory Cells

	Three-phase	Dual-phase
Additional clock phases	3	2
CMOS clock generators	2	1
Clock zones per memory cell	3	2
Read latency	$2T$	T
Write latency	$2T$	$2T$

the dual-phase memory cell and the read/write control is maintained through the AND and NOR gates. The memory architecture is built similar to a shift-register, where data is stored within the building blocks (e.g., register equivalents) and the blocks are replicated for a multibit storage mechanism (not shown in Figure 5).

The CMOS equivalent of building block for the shift-register-based memory architecture is shown in Figure 6. In brief, the dual-phase, line-based memory cell maintains data by applying opposite values to the two inputs ($\underline{X} \neq \underline{Y}$, also see Figure 4). In order to write data D into the cell, identical values are applied to the two inputs ($\underline{X} = \underline{Y} = D$). A read/write signal R/W is defined for each row that permits the read/write operations on the m -bit-long, shift-register structure (of an $n \times m$ memory). The read/write control circuitry (AND and NOR gates) drives

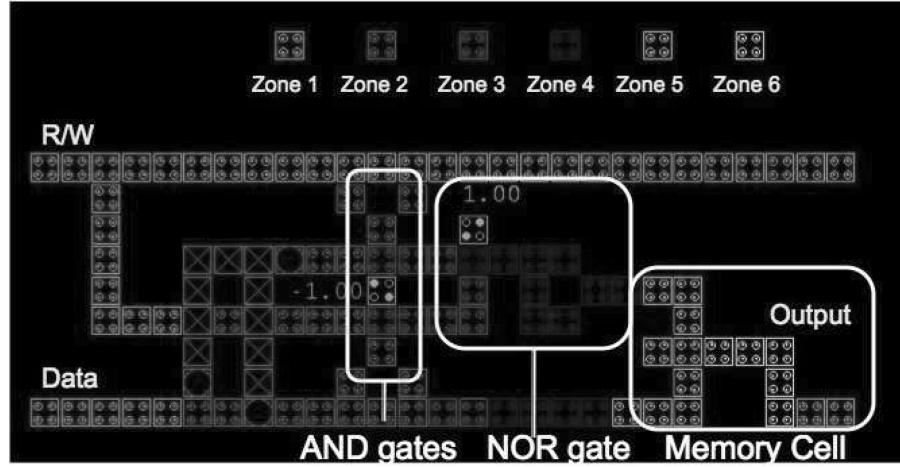


Fig. 5. Template building block for the shift-register-based memory architecture: the line-based memory cell with control circuitry.

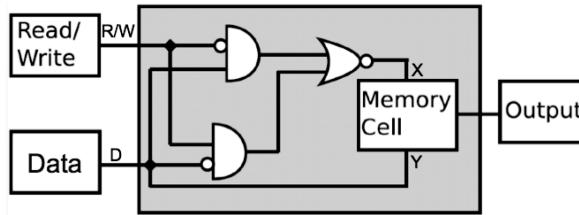


Fig. 6. CMOS equivalent of the building block.

the X and Y inputs, based on the data signal D and a read/write signal R/W :

$$X = \overline{(R/W)D + (R/W)\bar{D}} = (R/W) \odot D, \quad (1)$$

$$Y = D. \quad (2)$$

When the Read/Write (R/W) signal is asserted, new data is written in, while the previously stored data is read:

$$X = 1 \odot D = D = Y \Rightarrow (X = Y = D, \text{ write operation}). \quad (3)$$

The rest of the time (when $R/W = 0$), the data remains stored:

$$X = 0 \odot D = \bar{D} \neq Y \Rightarrow (X \neq Y = D, \text{ maintain operation}). \quad (4)$$

4.1 Architecture Design

Within the shift-register-based architecture, data is maintained in a static arrangement with serial input and output (due to the shift-register structure). Shift-registers are most suitable for first-in-first-out (FIFO) type operations but also can be used as a typical memory with a feedback wire between the serial output and the serial input. In the proposed QCA implementation, such a feedback QCA wire is used to complete the architecture design for a typical

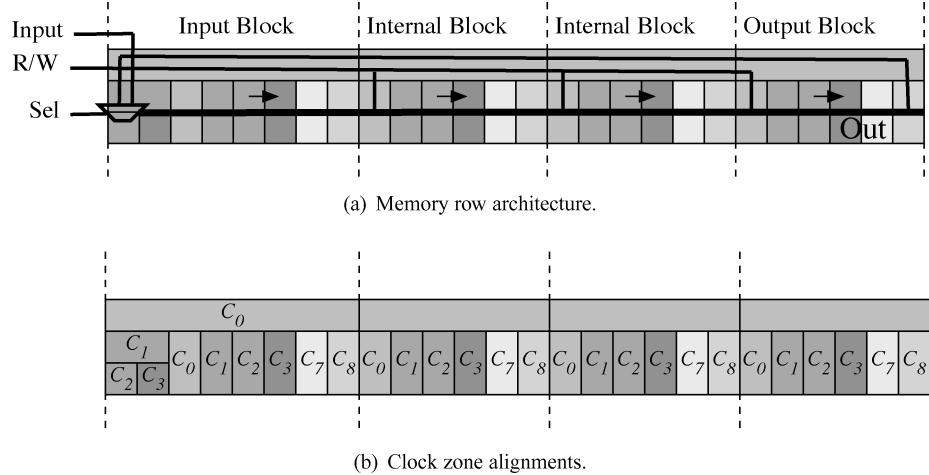


Fig. 7. Illustration of a shift-register-based memory with two internal tiles, an input tile and an output tile.

memory. To this end, a QCA routing line is drawn from the output of the wordline to the input of the wordline as shown in Figure 7. The template building block shown in Figure 5 is modified into input, internal and output blocks (discussed next in Section 4.2), permitting the shift-register architecture depicted in Figure 7. The data value that is stored in the shift register architecture is serially read from the `Out` terminal, while simultaneously being fed back into the multiplexer for data input to the shift register. If the shift register data needs to be stored back into the memory after a read operation, the control signal `Sel` is asserted to multiplex the feedback data signal into the shift register memory. For a destructive read or for a typical FIFO operation, the `Sel` signal is not asserted, multiplexing the external input data signal `Input` into the shift register memory.

The data feedback line is drawn along the length of the memory wordline in the same clock zone as the read/write enable signal. The area impact of this feedback line is minimal as the clock zone is shared with the read/write line. Also of importance is the area and latency impact of the three additional clock zones C_1 , C_2 , and C_3 defined on the input side of the shift-register architecture. These clock zones are stacked together, each with half the height of a typical clock zone. The multiplexer gate to select between the previously stored data and external data input is implemented within these three (additional) conventional clock zones.

As illustrated in Figure 7, the proposed shift-register QCA memory features parallel, regular layout of clock zones. The orientation and type of clock zones are clearly marked in Figure 7(b). It should be noted, that, the regularity of the clock zones can further be improved by sacrificing the latency of operation, where necessary. For instance, the multiplexer on the input side can be implemented in perfectly parallel clocking zones [Vankamamidi et al. 2006], as opposed to the relatively regular and compact clock zones used in this work.

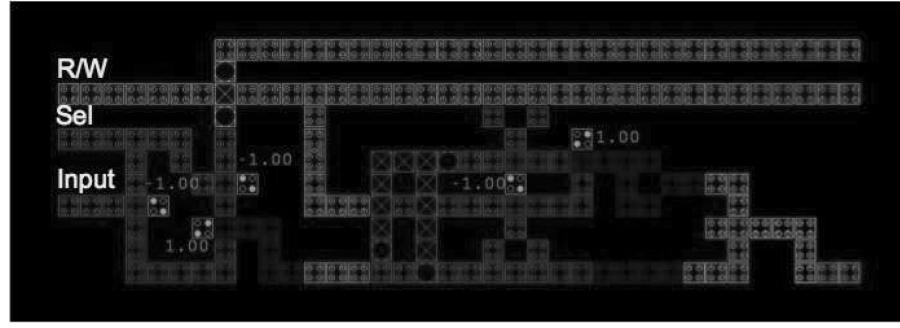


Fig. 8. Input block for the shift-register-based memory architecture.

A sample QCA layout of the proposed architecture, slightly modified to adhere to the simulation limitations of the experimental verification tool, is presented next.

4.2 QCA Architecture Layout

As mentioned above, the building block of the shift-register-based QCA memory shown in Figure 5 is redesigned into three specialized categories: the *input* block (Figure 8), the *internal* block (Figure 9) and the *output* block (Figure 10). This specialization is necessary to draw the feedback line connecting the Out terminal of the n -bit shift register to the multiplexed Input terminal at the input block as shown in Figure 7. The internal block is similar to the building block in Figure 5 except for the QCA line in the same zone and placed parallel to the read/write signal for output feedback. The output block has a short read/write signal and the output terminal fanouts to the feedback line propagating backwards along the direction of the shift-register architecture. The input block has the most significant changes to encapsulate the control features for the rewriting operation. In particular, the input block has three additional clock zones and three majority gates formed into a multiplexer. As illustrated in Figure 7 and visible in Figure 8, the three additional clock zones have approximately half the height of original clock zones and have comparable widths. The rewrite operation is performed with the area overheads of three clock zones for the control circuitry (e.g., for any n -bit shift register) and the latency overhead of one clock cycle (T) induced by the propagation of the feedback data.

The QCA layout of a demonstrative 4×8 shift-register-based memory architecture (using the building blocks in Figures 8, 9, and 10) is shown in Figure 11. Note that the building blocks are stacked in a row and the clocking zones between building blocks are meshed seamlessly for uninterrupted operation and dense implementation.² The regularity of the clock zones in the overall architecture is visible from the color coding in Figure 11, which adheres to the

²Note that in Figure 11, additional spaces are inserted between building blocks as well as between the read/write signal line and the rest of the building block for better visibility; denser implementations are possible.

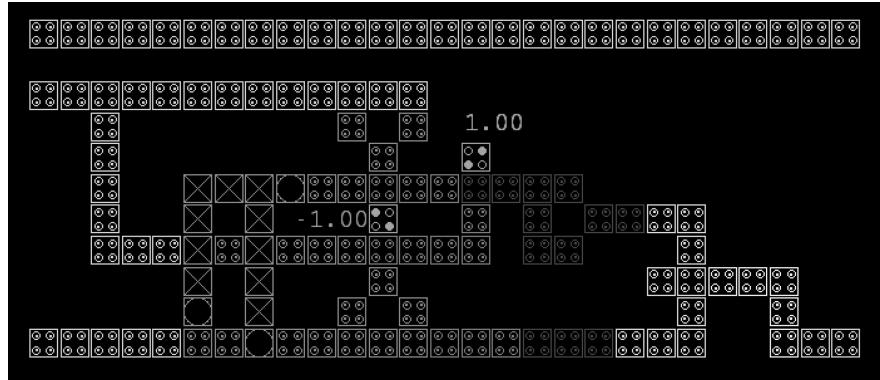


Fig. 9. Internal block for the shift-register-based memory architecture.

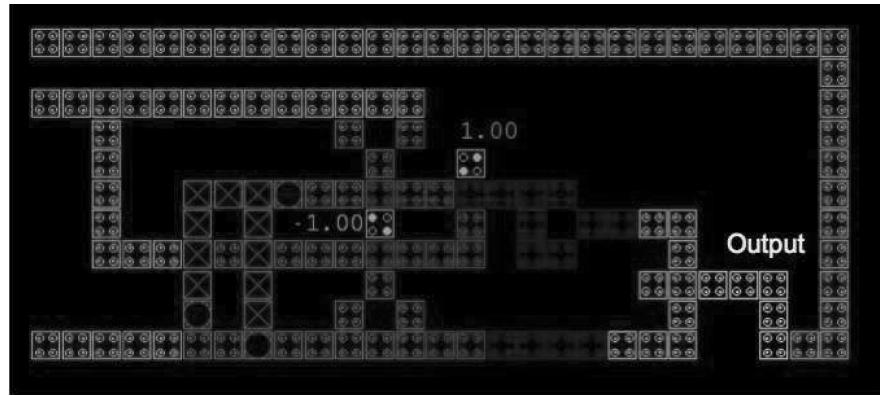
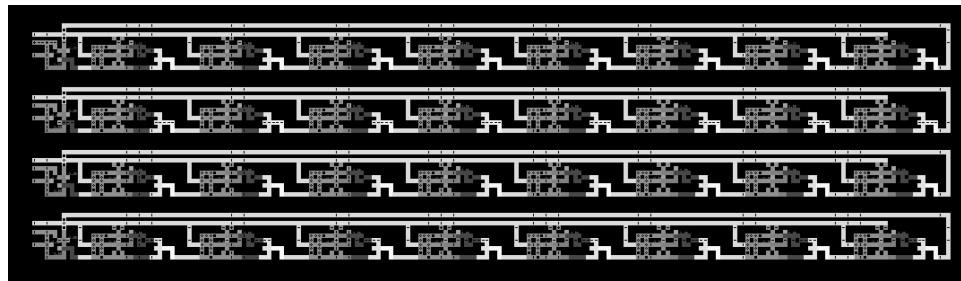


Fig. 10. Output block for the shift-register-based memory architecture.

Fig. 11. QCA diagram of a sample $n \times m$ memory architecture, where $n = 4$ and $m = 8$.

configuration depicted in Figure 7. Such regularity of clock zones is important for CMOS clock wiring layout within the QCA manufacturing process.

In Figure 11, the R/W signal (and data feedback signal) is routed in an exclusive horizontal clocking zone for simultaneous delivery to building blocks. Thus, the number of QCA cells in a line (wire) that can be placed in a single clock zone dictates the physical limitation of operation for the displayed memory

architecture. Alternative read/write (or row select) signal delivery methods, such as those analogous to traditional CMOS methods of metal bypass, multiple drivers, and hierarchical word-line selection schemes [Rabaey et al. 2003] can easily be adopted for longer wordsize implementations using this architecture.

5. SIMULATIONS AND ANALYSIS

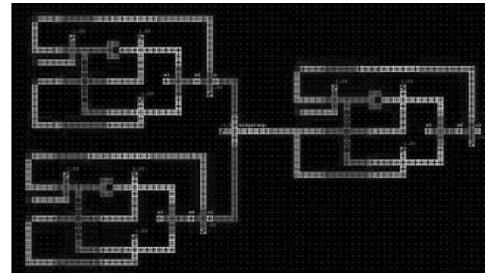
Two different simulators, QCADesigner³ [Walus et al. 2004] and HDLQ [Ottavi et al. 2006] with Modelsim, are used to verify the functionality of the presented shift-register-based memory architecture. QCADesigner is used to simulate small, simple building blocks with higher accuracy, while HDLQ is used for the verification of the larger-scale system avoiding the slow execution speeds of QCADesigner-only simulation.

QCADesigner is an open source software which has been widely used by QCA researchers. However, QCADesigner simulator (v2.0.3) only supports the conventional four-phase clocking scheme for QCA circuits. In order to simulate QCA architectures with alternative clocking schemes, such as the three-phase, line-based memory in Vankamamidi et al. [2005a] or the dual-phase, line-based memory architectures in Taskin and Hong [2006], modifications must be performed to the design or to the simulator. The former method is followed in Vankamamidi et al. [2005a], where a time-to-space transformation procedure is devised. The latter method is followed in Taskin and Hong [2006], where the software is modified to accommodate advanced clocking schemes. In this article, the modified software in Taskin and Hong [2006] is used.

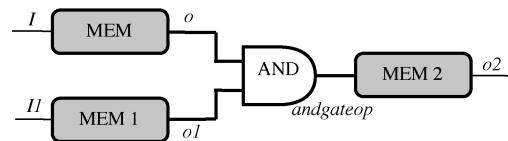
HDLQ [Ottavi et al. 2006] is a design language tool for the simulation of QCA architectures written in Verilog Hardware Description Language (HDL) style. Instead of modeling QCA at the physical level with quantum-mechanical computations, it models QCA circuits by assuming an ideal logical model of QCA operation. Such a treatment is especially valuable for fast prototyping of complex QCA circuitry. In HDLQ, a circuit is described as a series of interconnected modules, where each module is a common element of a QCA circuit. For instance, elements modeling a QCA wire, fanout, inverter, and majority voter are provided in the library. Adjacent elements of a QCA circuit have the inputs and outputs wired in Verilog representation. HDLQ also allows for bidirectionality of QCA devices, which is essential to the presented shift register design.

In the first step of design verification, the line-based memory cell is simulated using QCADesigner. For demonstrative purposes, the simple circuit shown in Figure 12 is constructed. In this circuit, data is read from two QCA memory cells (*MEM* and *MEM1*), logical AND operation is performed on this data, and the result is stored in the QCA memory cell *MEM2*. In Figure 12(a), three instances of the proposed memory cell are shown, *MEM* and *MEM1* on the left and *MEM2* on the right. The simulations results, including the simulation waveforms for the nodes labeled in Figure 12(b), are shown in Figure 13. Inputs 1110 and 1011 are applied to data input terminals *I* and *I1*, respectively. Simulation depicts the values of internal nodes and the output node *o2* of the AND operation on these values. For instance, the output of 1 is observed for the

³<http://www.qcadesigner.ca>.



(a) Layout view.



(b) Schematic view.

Fig. 12. Experimental circuit for the dual phase line-based memory cell.

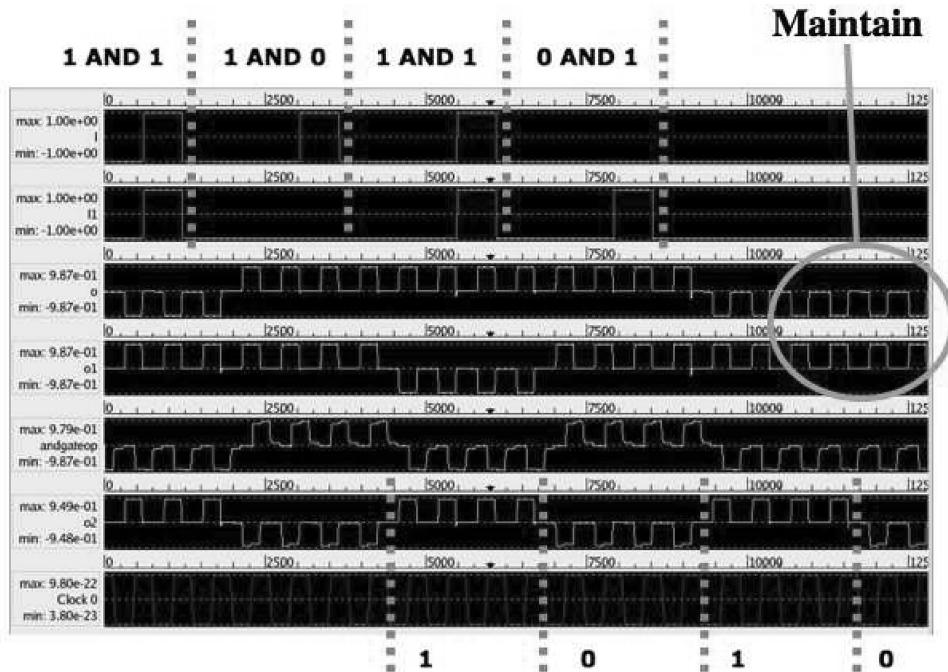


Fig. 13. Simulation waveforms obtained with QCADesigner (modified to accommodate advanced clocking) for the circuit shown in Figure 12.

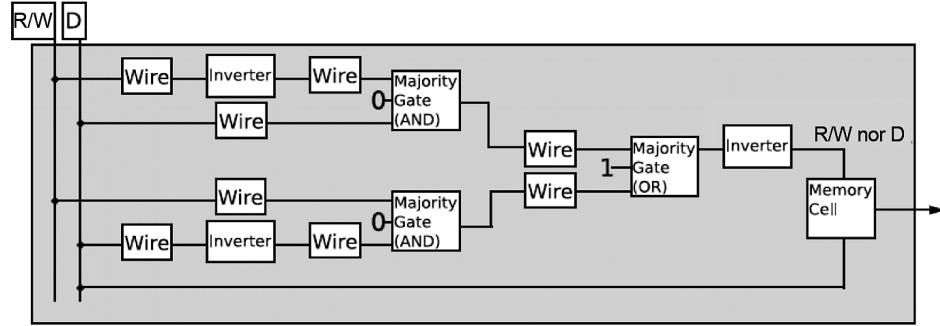


Fig. 14. HDLQ block diagram of presented memory cell shown in Figure 5.

AND operation of values 1 and 1 stored in *MEM* and *MEM1* cells, respectively ($1 \text{ AND } 1 = 1$). The simulations waveforms *o* and *o1* depict the stored values in *MEM* and *MEM1* cells, respectively. For the first set of data, *o1* and *o2* are 1 and 1, respectively. The *and gateop* internal node depicts the computed AND value, for example, 1 for the first set of data ($1 \text{ AND } 1 = 1$). This value is written into *MEM3* cell and shown at the output terminal *o2* after the write latency of that memory cell. Similar operations are observed for input sets of 1/0 ($1 \text{ AND } 0 = 0$), 1/1 and 0/1 ($0 \text{ AND } 1 = 0$). On the simulation waveform in Figure 13, read and write operations from and to memory cells *MEM*, *MEM1*, and *MEM2* are illustrated multiple times. Correct outputs are observed on all input patterns. In the simulation, maintain operation is depicted on internal outputs *o* (*o1*) for memory cell *MEM* (*MEM1*) on input value 0 (1) during the last $T/4$ of simulation waveform. In conclusion, these QCADesigner simulations confirm the proposed operation of the dual-phase, line-based memory cells.

Next, an HDLQ module is developed for the building block, as shown in Figure 14. This model emulates the functionality of the single-bit memory cell with Verilog code in the same style as the preexisting HDLQ modules. Using this macromodel for the line-based memory cell (whose operation is proved with the QCADesigner simulations mentioned above), experiments at the system level are performed in order to verify the functionality and characterize the performance of the proposed shift-register based memory *architecture*. The schematic for the *nxm* architecture is shown in Figure 15. Note that this HDLQ module excludes the feedback wire from the last bit to the first bit. Nonetheless, the model captures the important shifting and storage capabilities of the architecture and an extension to model the feedback wire is trivial. HDLQ simulation results, shown in Figure 16, consist of four rows of the shift-register architecture. The signals *A1* and *A2* serve as address signals. The testbench demonstrates the read, maintain, and write operations. Initially, the memory is reset by writing value 0 to each bit (serially). Then each of the four memory rows has a different pattern written to it. The staircase-pattern-waveforms demonstrating the shift operation are visible upon close inspection in Figure 16. Because each row is essentially a shift register, new data must be supplied as the current data is output. In experiments, zeros are written to the memory for this purpose. The memory holds its value for a time, then each row's data is output.

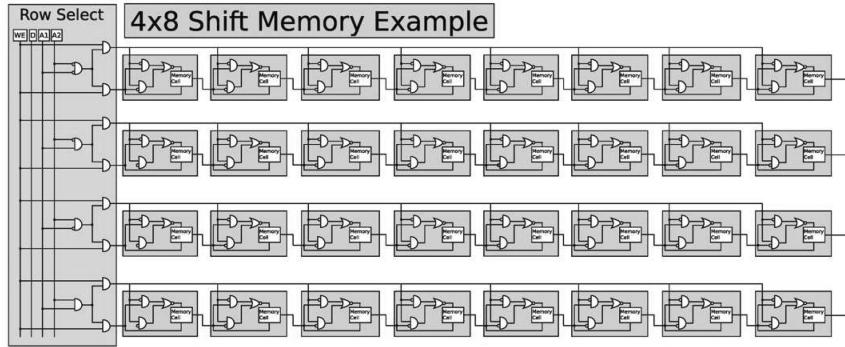
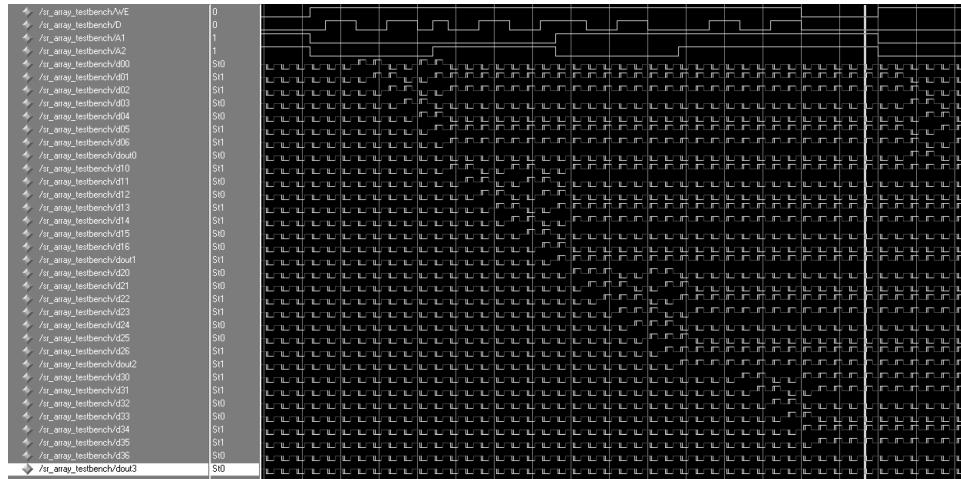
Fig. 15. Block diagram of a sample $n \times m$ memory architecture, where $n = 4$ and $m = 8$.

Fig. 16. ModelSim waveform from HDLQ description of the 4x8 memory. All bits are initialized to value 0 and then overwritten.

For better visualization of the write process, memory operation on row 3 is magnified in Figure 17, onto which a data pattern 11001100 is written. In Figure 17, the top four signals are the control signals. The signal R/W is the read/write signal, D is the data being written, and the signals $A1$ and $A2$ are the row address bits. The rest of the signals, from $d30$ to $dout3$ show the value stored at each memory cell. The cell labeled $d30$ is the leftmost cell and the $dout3$ is the rightmost cell. The staircase pattern, due to shifting, is also visible upon close inspection.

For clarity, it is important to emphasize that the simulation times shown HDLQ simulations do not represent the projected implementation performance of the QCA system; it is used to count the number of cycles in operation. Actual operation times will depend on QCA manufacturing technique and the clock speed.

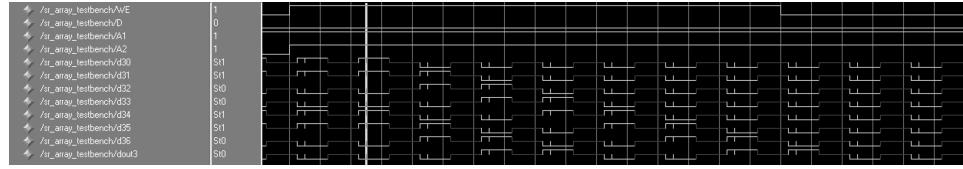


Fig. 17. Data pattern 11001100 from row 3 of the four memory rows read in parallel in consequent cycles based on shift-register operation.

6. COMPARISONS

The read and write latencies of the shift register architecture are m cycles for an m -bit row, as bits are read and written serially (and sequentially). As detailed in Section 3.1, the read/write latency of the tile-based architecture is a variable between m and $2m - 1$ cycles due to the circulation of data. Variable read/write latencies might cause hardship in system operation, so a constant read/write latency of $2m - 1$ cycles can be selected, which is the latency for the worst-case operation (first-bit-to-read is at the internal tile adjacent to the output tile). In comparison, the read/write latency of the QCA RAM architecture is at least proportional to wordsize (latency changes based on the address decoder implementations) and the read/write latency of H-memory is a constant dependent on the size of the binary tree (H-tree), thus cannot be directly compared.

Compared with the H-memory proposed in Frost et al. [2002] and RAM-based QCA memory proposed in Walus et al. [2003], in which conventional four-phase clocking and the feedback-topology memory cell are used, the proposed shift-register memory architecture permits denser implementations. For instance, the number of clocking zones in the H-memory depends on the wordsize and the number of words, while the number of clocking zones in line-based memories depend only on wordsize. For tile-based and shift-register-based architectures (with line-based memory cells), QCA implementations are much denser. Amongst these two architectures, the tile-based architecture is denser than the proposed shift-register-based architecture. An internal tile occupies three clocking zones and holds two bits of data, while a shift register internal block shown in Figure 9 occupies six clocking zones and holds a single bit of data. The low utilization of area in the tiles [empty space between top and bottom rows, see Figure 3(a)] slightly degrades the density advantage. Overall, considering such $\sim 2X$ area disadvantage and the $2X$ read/write latency advantage, the proposed shift-register architecture provides for a less dense but faster memory implementation compared to the tile-based memory architecture. It is important to note that the tile-based architecture requires an additional circuitry (a counter) as part of the read/write control mechanism. This external circuitry is not included in Vankamamidi et al. [2005b] and Ottavi et al. [2005], thus excluded from the comparisons presented in this paper as noted in Table II.

A categorized comparison of the two QCA memory architectures is presented in Table II. In brief, the advantages of the *dual-phase* line-based memory cells (integral part of the building blocks) presented in Table I are observed for the presented shift-register based architecture, as well, except for the area advantage. The area disadvantage is due to the inclusion of clock zones for the

Table II. Comparison of the Tile-Based Architecture and the Shift-Register-Based Architecture

	Tile-based[Vankamamidi et al. 2005b]	Shift-register-based
Building blocks	Tiles	Shift register cell
Clock zones per cell	3	6
Area utilization	Low	High
Bits stored per cell	2	1
Read/write latency	m to $2m - 1$	$m + 1$
Required external circuitry	Counter	<i>None</i>

read/write control circuitry (AND and NOR gates), which, otherwise, does not significantly impact the nanoarchitecture performance.

7. CONCLUSIONS

This article presents a novel, shift-register-based QCA memory architecture. Benefits of the dual-phase, line-based memory cell are observed in the proposed shift-register-based architecture in a dense implementation with desired regularity in clock zone alignments. The read/write latency is linear in the number of bits stored per row, which records up to $\sim 2X$ improvements in read/write latency over of the tile-based memory architecture (with three-phase, line-based memory cells). Furthermore, unlike the tile-based memory architecture, shift-register-based architecture provides a constant read/write latency.

REFERENCES

- AMLANI, I. 1999. Digital logic using quantum-dot cellular automata. *Science* 284, 289–292.
- AMLANI, I., ZHANG, R., TRESEK, J., NAGAHARA, L., AND TSUI, R. K. 2003. Manipulation and characterization of molecular scale components. In *Proceedings of the IEEE Design Automation Conference*. 276–277.
- BERZON, D. AND FOUNTAIN, T. J. 1999. A memory design in QCAs using the SQUARES formalism. In *Proceedings of the Great Lakes Symposium on Very Large Scale Integration (VLSI)*. 166–169.
- FROST, S., RODRIGUES, A., JANISZEWSKI, A., RAUSCH, R., AND KOGGE, P. 2002. Memory in motion: A study of storage structures in QCA. In *Proceedings of the Workshop on Non-Silicon Computing*.
- FROST, S. E., RODRIGUES, A. F., GIEFER, C. A., AND KOGGE, P. M. 2003. Bouncing threads: Merging a new execution model into a nanotechnology memory. In *Proceedings of the IEEE Symposium on VLSI*. 19–25.
- HENNESSY, K. AND LENT, C. S. 2001. Clocking of molecular quantum-dot cellular automata. *J. Vacuum Sci. Tech. B* 19, 5, 1752–1755.
- HU, W., SARVESWARAN, K., LIEBERMAN, M., AND BERNSTEIN, G. H. 2005. High resolution electron beam lithography and DNA nano-patterning for molecular QCA. *IEEE Trans. Nanotech.* 4, 3, 312–316.
- HUANG, J., MOMENZADEH, M., SCHIANO, L., AND LOMBARDI, F. 2005. Simulation-based design of modular QCA circuits. In *Proceedings of the IEEE Conference on Nanotechnology*. 533–536.
- KUMMAMURU, R. K., TIMLER, J., TOOTH, G., LENT, C. S., RAMASUBRAMANIAM, R., ORLOV, A. O., BERNSTEIN, G. H., AND SNIDER, G. L. 2002. Power gain in a quantum-dot cellular automata latch. *Appl. Phys. Lett.* 81, 7, 1332–1334.
- LENT, C., ISAKSEN, B., AND LIEBERMAN, M. 2003. Molecular quantum-dot cellular automata. *J. Am. Chem. Soc.* 125, 4, 1056–1063.
- LENT, C. S. AND ISAKSEN, B. 2003. Clocked molecular quantum-dot cellular automata. *IEEE Trans. Electron Dev.* 50, 9, 1890–1896.
- ORLOV, A. O., AMLANI, I., TOOTH, G., LENT, C. S., BERNSTEIN, G. H., AND SNIDER, G. L. 1999. Experimental demonstration of a binary wire for quantum-dot cellular automata. *J. Appl. Phys. Lett.* 74, 19, 2875–2877.

- OTTAVI, M., PONTARELLI, S., VANKAMAMIDI, V., SALISANO, A., AND LOMBARDI, F. 2005. Design of a QCA memory with parallel read/serial write. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*. 292–294.
- OTTAVI, M., SCHIANO, L., PONTARELLI, S., VANKAMAMIDI, V., AND LOMBARDI, F. 2006. Timing verification of QCA memory architectures. In *Proceedings of the IEEE Conference on Nanotechnology*. Vol. 1. 391–394.
- OTTAVI, M., VANKAMAMIDI, V., LOMBARDI, F., PONTARELLI, S., AND SALISANO, A. 2005. Design of a QCA memory with parallel read/serial write. In *Proceedings of the IEEE Symposium on Very Large Scale Integration (VLSI)*. 292–294.
- RABAEY, J. M., CHANDRAKASAN, A., AND NIKOLIC, B. 2003. *Digital Integrated Circuits: A Design Perspective*, 2 ed. Prentice Hall.
- SNIDER, G. L., ORLOV, A. O., AMLANI, I., BERNSTEIN, G., LENT, C. S., MERZ, J. L., AND POROD, W. 1999. Quantum-dot cellular automata: Line and majority logic gate. *Journal of Applied Physics* 38, 7227–7229.
- TASKIN, B. AND HONG, B. 2006. Dual-phase line-based QCA memory design. In *Proceedings of the IEEE Conference on Nanotechnology*. 302–305.
- VANKAMAMIDI, V., OTTAVI, M., AND LOMBARDI, F. 2005a. A line-based parallel memory for QCA implementation. *IEEE Trans. Nanotech.* 4, 6, 690–698.
- VANKAMAMIDI, V., OTTAVI, M., AND LOMBARDI, F. 2005b. Tile-based design of a serial memory in QCA. In *Proceedings of the ACM Great Lakes Symposium on VLSI*. 201–206.
- VANKAMAMIDI, V., OTTAVI, M., AND LOMBARDI, F. 2006. Clocking and cell placement for QCA. In *Proceedings of the IEEE Conference on Nanotechnology*. 343–346.
- WALUS, K., DYSART, T. J., JULIEN, G. A., AND BUDIMAN, R. A. 2004. Qcadesigner: a rapid design and simulation tool for quantum-dot cellular automata. *IEEE Trans. Nanotech.* 3, 1, 26–31.
- WALUS, K., VETTETH, A., JULIEN, G., AND DIMITROV, V. 2003. Ram design using quantum-dot cellular automata. In *Proceedings of the Nanotechnology Conference and Trade Show*. Vol. 2. 160–163.
- YADAVALLI, K. K., ORLOV, A. O., KUMMAMURU, R. K., LENT, C. S., BERNSTEIN, G. H., AND SNIDER, G. L. 2005. Fanout in quantum-dot cellular automata. In *Proceedings of the Device Research Conference*. 121–122.

Received October 2008; revised ; accepted October 2008