Tile-Based QCA Design Using Majority-Like Logic Primitives

J. HUANG, M. MOMENZADEH, L. SCHIANO, M. OTTAVI, and F. LOMBARDI
Northeastern University

The design of circuits and systems in Quantum-dot Cellular Automata (QCA) is still in infancy. The basic logic primitive in QCA is the majority voter (MV), that is not a universal function; so, inverters (INV) are also required. Blocks (referred to as tiles) are utilized in this article. A tile with a combined logic function of MV and INV (MV-like function) is proposed. It is shown that the MV-like tile can be effectively used in logic design as basic primitive. Tiles based on both the fully populated (FP) and non-fully populated (NFP) grids are investigated in detail. Various arrangements in inputs and outputs are also possible among the 4 sides of a grid, thus defining different tiles. Using a coherence vector simulation engine, it is shown that the 3 × 3 grid offers versatile logic operation. Different combinational functions such as majority-like and wire crossing are obtained using these tiles. Tile-based design of different circuits is compared to gate-based and SQUARES designs.

Categories and Subject Descriptors: B.6.0 [**Logic Design**]: General

General Terms: Design

Additional Key Words and Phrases: QCA, emerging technologies, processing-by-wire

1. INTRODUCTION

Nanotechnology provides new possibilities for computing due to the unique properties that arise at such reduced feature sizes. Consider the processing features of CMOS systems: some circuits perform computation, while others are used for signal/data transfer and communication. In FPGAs, for example, computation is performed by the logic resources or PEs (processing elements), while communication is accomplished by the interconnect fabric (consisting of wires and switches in the channels separating the PEs). *Quantum-dot Cellular Automata* (QCA) [Lent et al. 1994] is an *emerging technology* that does not fully rely on this separation of roles. In QCA, *computation and communication* occur simultaneously [Amlani et al. 1999; Frost et al. 2002; Niemier et al.

This article is an extended version of the paper that has appeared in *Proceedings of the IEEE Conference on Nanotechnology*, IEEE Computer Society Press, Los Alamitos, CA, 2005.

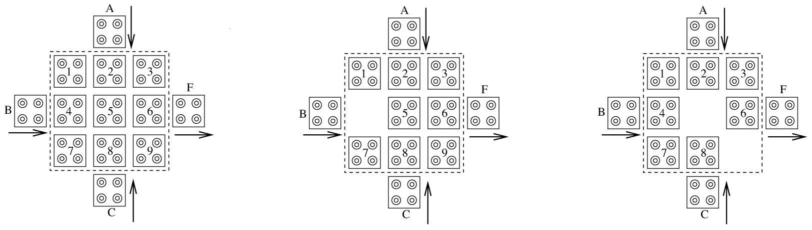
Authors' address: Department of Electrical and Computer Engineering, Northeastern University, 263 Eagen Research Center, Huntington Ave., Boston, MA 02115; E-mail: {hjing,mmomenza, lschiano,mottavi,lombardi}@ece.neu.edu.

1999]. Logic elements are based on two gate primitives (inverter, INV and majority voter, MV) to implement combinational circuits; as for interconnect, two arrangements referred to as the binary wire and the inverter chain, are utilized [Tougaw and Lent 1994]. QCA is very promising [Smith 1999] because computational paradigms, which radically depart from traditional CMOS, can be implemented with this technology [Frost et al. 2002; Niemier et al. 2002; Dimitrov et al. 2002, Walus et al. 2002, 2003]. QCA design involves diverse and new paradigms such as *memory-in-motion* and *processing-by-wire* [Frost et al. 2002; Niemier and Kogge 2001]. For memory-in-motion [Frost et al. 2002], storage and sequential circuits can be assembled using a loop in which binary information is circulated. Memory-in-motion is an instance of the more general paradigm of processing-by-wire. Processing-by-wire (PBW) [Niemier and Kogge 2001] is the QCA capability by which information manipulation can be accomplished, while transmission and communication of signals take place. PBW capabilities can be observed in the so-called inverter chain as well as in the arrangement of the cells in a majority voter.

The existing literature on QCA design mostly uses a gate-based methodology [Walus et al. 2003; Niemier and Kogge 2001]. In a gate-based design, following logic synthesis, individual gates (MV and INV) are connected to form the desired circuit. The majority function (MV) is not universal, so inversion (INV) is also required. Inversion can be achieved in QCA using a 45 degrees cell orientation. However, it has been shown that this arrangement is not defect-tolerant [Tahoori et al. 2004]. An inverter chain (Figure 3) can be used. An issue associated with using the inverter chain is that rotated cells (cells rotated by 45 degrees) are employed; these cells are difficult to manufacture. Inversion can also be achieved using the INV gate (Figure 3). In CMOS, the INV is the simplest gate, however in QCA the INV gate is at least as large as the MV.

Recent developments in QCA manufacturing involve molecular implementations. It is expected that homogeneous cell arrangements will be constructed by either self-assembly, or large scale cell deposition on insulated substrates [Bernstein et al. 2004]. These manufacturing techniques are well suited to modularization. QCA design can be implemented by modularization through a simple, Manhattan-style interconnect; however, this design is expected to generate an area overhead compared to a gate-based design. This has also been encountered in CMOS: a design using a full-custom layout is usually smaller than a design using standard-cells. In the technical literature, QCA design at the modular level has not been treated in depth. A methodology known as SQUARES has been proposed [Berzon and Fountain 1999], in which the basic building block is a 5 × 5 QCA cell grid. Logic functions are determined based on the direct embedding of the MV and INV circuits into this grid, rather than on analyzing the different interactions and possible circuit configurations of the QCA cells. Furthermore, as it will be shown in Section 5, SQUARES results in significant area and delay overheads.

In this article, a design based on elementary building blocks referred to as tiles, is proposed for QCA. Two types of tiles, namely the *active tiles* and the *passive tiles* can be used [Huang et al. 2005a]. A tile is defined as active if it implements a combinational logic function with minterm(s) (of at least two



(a) Fully Populated (b) Non Fully Populated (c) Non Fully Populated

Fig. 1. Examples of Fully and Non Fully Populated QCA grids.

literals). For example, a tile that performs the majority function is an active tile. A tile is said to be passive if it implements logic functions of only one literal, that is, wire and INV functions. The basic logic primitive in the proposed design is the MV-like tile. This tile performs a majority function with selective inversion (if required) at its inputs. The MV-like function is universal and area efficient compared with using separate MV and INV. In addition to the MV-like tile, passive tiles are used for interconnect [Huang et al. 2005a].

A tile can be built from an $n \times n$ square grid of QCA cells. Both fully populated (FP) and non fully populated (NFP) grids can be used as basic logic blocks. Figure 1(a) illustrates a 3×3 FP grid (made of 9 cells) with three inputs (A, B and C) and one output (F), as mostly used in this paper. An NFP grid is obtained from the FP grid by selectively undepositing QCA cells. Figures 1(b) and (c) illustrate two instances of a 3×3 NFP grid when one and two cells are undeposited respectively. Each numerated box is a QCA cell that may be deposited (i.e., included in the final QCA layout). This process is therefore applicable in the design phase prior to cell deposition [Huang et al. 2005a]. In a circuit, isolation among tiles is enforced through the input/output cells and spacing between tiles to limit unwanted interactions [Huang et al. 2005a]. As shown in later sections, this results in a clocking arrangement which is simpler than the one required by SQUARES.

Among $n \times n$ grids, the 3×3 grid has computational properties which makes it very attractive for designing larger circuits [Huang et al. 2005b]. Tiles which utilize the 3×3 grid, are therefore used as examples. Using different input and output cells, five tiles are analyzed as providing a high degree of flexibility in logic operation. Different logic functions can be generated by using less than n^2 cells in a grid of dimension n (NFP grid). The functional behaviors of the tiles are reported using simulation and analyzed in detail. Tile-based circuits are compared with SQUARES and a gate-based design. It will be shown that clocking using the proposed tiles is simpler than the one required by SQUARES.

The rest of the article is organized as follows: Section 2 introduces the preliminaries inclusive of a brief review of QCA. QCA tiling is described in Section 3. Section 4 extends the general principles of tiling to the 3×3 grid and to five different tiles; logic behaviors based on input and output cells and NFP grids, are presented and discussed. A discussion and analysis of the simulation results are also provided. Section 5 presents various designs of QCA circuits as examples of the applicability of the proposed tile-based design.

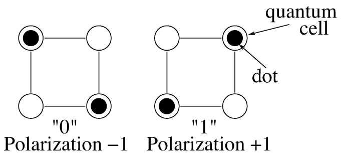
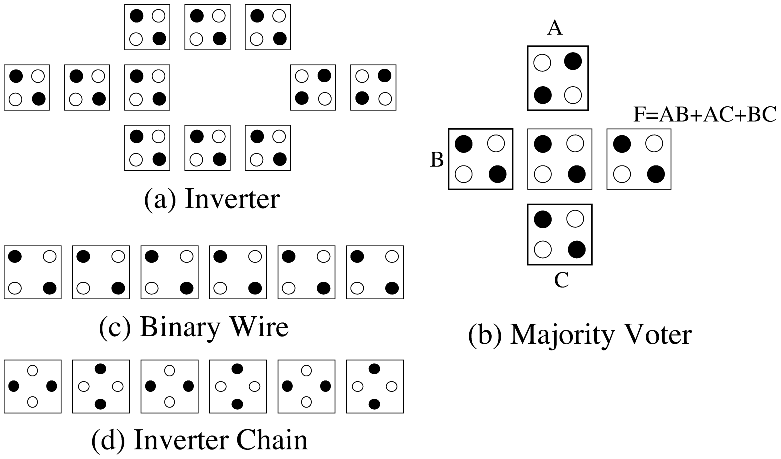


Fig. 2. QCA cell and polarization states.



(a) Inverter

(b) Majority Voter

(c) Binary Wire

(d) Inverter Chain

Fig. 3. Basic QCA devices.

2. REVIEW OF QCA

A QCA cell can be viewed as a set of four charge containers or “dots”, positioned at the corners of a square [Tougaw and Lent 1994]. The cell contains two extra mobile electrons which can quantum mechanically tunnel between dots, but not cells. The electrons are forced to the corner positions by Coulombic repulsion. The two possible polarization states represent logic “0” and logic “1”, as shown in Figure 2.

Unlike conventional logic circuits in which information is transferred by electrical current, QCA operates by the Coulombic interaction that connects the state of one cell to the state of its neighbors. This results in a technology in which information transfer (interconnection) is the same as information transformation (logic manipulation).

One of the primitive logic gates in QCA is the *majority voter* [Tougaw and Lent 1994]. The majority voter with logic function $\text{Maj}(A, B, C) = AB + AC + BC$, can be realized by only five QCA cells, as shown in Figure 3(b). Logic AND and OR functions can be implemented from the majority voter by setting an input (the so-called programming or control input) permanently to a 0 or 1 value. The inverter is the other basic gate in QCA and is shown in Figure 3(a). The binary wire and inverter chain (as interconnect fabric) are shown in Figure 3(c)(d).

In QCA, timing is accomplished by clocking in four distinct and periodic phases (as shown in Figure 4). A QCA circuit is partitioned into a number of clocking zones and all cells in the same zone are controlled by the common clock signal. The use of a quasi-adiabatic switching technique for QCA circuits

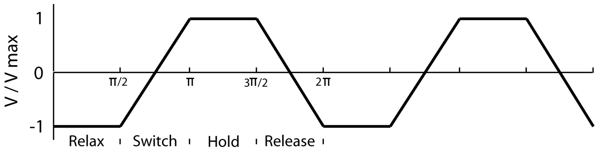


Fig. 4. Four-phased signal for clocking zones in QCA, adiabatic switching.

requires a four-phased clocking signal (which is commonly supplied by CMOS wires buried under the QCA circuitry) for modulating the inter-dot tunneling barrier. The four phases are Relax, Switch, Hold and Release. During the Relax phase, there is no interdot barrier and a cell remains unpolarized. During the Switch phase, the interdot barrier is slowly raised and a cell attains a definitive polarity under the influence of its neighbors. In the Hold phase, barriers are high and a cell retains its polarity. Finally, in the Release phase, barriers are lowered and a cell loses its polarity. Timing zones of a QCA circuit or system are arranged in this periodic fashion such that zones in the Hold phase are followed by zones in the Switch, Release and Relax phases. A signal is effectively "latched" when one clocking zone goes into the Hold phase and acts as input to the subsequent zone. This clocking mechanism provides inherent pipelining [Antonelli et al. 2004] and allows multibit information transfer for QCA through signal latching.

The software simulation tool QCADesigner[1] v1.4.0 (Unix version) is used for evaluation [QCADesigner]. QCADesigner v1.4.0 features different simulation engines. Throughout this paper, the coherence vector engine is used due to its accurate and detailed evaluation of QCA. The coherence vector engine is based on the density matrix approach [QCADesigner], which models the power dissipative effects of QCA. This engine performs a time-dependent simulation of the QCA design. Each cell is modeled as a two-state system which is represented by a Hamiltonian.

3. TILING FOR QCA

The design of nano circuits and systems requires a substantially different approach than for CMOS-based VLSI. The large density expected for QCA (especially for molecular implementations) [Qi et al. 2003; Jiao et al. 2003] represents one of the significant features in the manufacturing of these systems. As a technology in infancy, QCA still requires versatile building blocks for deposition and circuit assembly. The SQUARES methodology has been proposed [Berzon and Fountain 1999], in which the basic building block is a 5 × 5 QCA cell grid. However, as it will be shown in Section 5, SQUARES results in a high area overhead. Furthermore, timing is also complex in SQUARES. A QCA simulation methodology based on tiling has been proposed in [Huang et al. 2005a]. It partially relies on the early work of [Berzon and Fountain 1999], but it provides a more complete characterization of the design process at logic level. In this article, a design is proposed based on the so-called MV-like tile as logic primitive. The MV-like tile is area efficient, because it combines the logic functions

[1] QCADesigner is developed by the ATIPS lab at the University of Calgary in Canada.

of MV and INV. The role of a grid with respect to the generation of new logic functions (inclusive of the wire crossing and MV-like) is also analyzed and assessed with respect to the input and output cells (and corresponding signals) prior to cell deposition on a substrate.

Tiling is very relevant to an emerging technology such as QCA. However, the following features must be properly addressed:

—Tiles must be flexible in the generation of logic functions at high polarization levels. As discussed in Section 2, the two fully polarized states in a QCA cell represent the two boolean logic states in QCA. Moreover, tiles should also be robust to limit interactions from unwanted cells.

—Between and among tiles, signals should be routed at ease (such as with a Manhattan strategy).

—The tiles should have stable signals, that is, no undetermined value (due to lack of polarization, or the presence of a glitch) should be present at an output.

Tiling is suitable prior to cell deposition and can be described as follows: Let a *tile* be defined as a square grid of cells (FP or NFP) with the addition of input and output cells. An input or output cell can be placed only in the middle of each side of the grid to ease routing.

Within a two-dimensional layout, active as well as passive tiles can be used [Huang et al. 2005a]. Active tiles perform logic computation, while passive tiles perform interconnection and limited computation. Passive tiles are mostly used for signal transfer/routing as well as providing separation among active tiles in the layout [Huang et al. 2005a]. Unwanted interactions among cells in different grids are very limited as no immediate adjacency between tiles is allowed; isolation is enforced through spacing (as provided by an area with no cells between tiles). Following logic synthesis and technology mapping (by which combinational functions of a QCA circuit S are mapped to specific QCA gates, such as MV), customization of the layout for implementing S is needed. In this process, clocking issues must also be included, that is, S is partitioned into zones to preserve the correct flow of data [Antonelli et al. 2004]. Without loss of correctness or generality, S is assumed to be expressed in SOP (Sum Of Products) form with minterms of possibly multiple literals; many iterations may be required for successful routing using passive tiles [Huang et al. 2005a].

It has been shown in [Huang et al. 2005a, 2005b] that the 3 × 3 grid is most attractive for tile-based design because it provides versatile logic functions. The 2 × 2 and the 4 × 4 grids cannot be utilized due to lack of polarization (in the latter) and processing (in the former) [Huang et al. 2005b]. As for the 5 × 5 and 6 × 6 grids, the output functions are rather complex and do not map efficiently to a logic synthesis process due to the irregular nature of the SOP and the minterms (with different numbers of literals) [Huang et al. 2005b]. Moreover, it has been shown that for logic design, functions of more than four literals in a minterm seldom occur in practice [McCluskey 1986]. So in this article, tiles are built based on the 3 × 3 grid. This is consistent with the condition that the side of the grid should have an odd number of cells to allow the placement of an input or output cell at its center for ease of routing [Huang et al. 2005a].

The selection of a grid also affects the area that a circuit occupies in the layout: a 3 × 3 grid can embed an MV or an INV (i.e., with no input/output cell, the MV and INV are isomorphic to the 3 × 3 grid). So while the number of cells may be larger in a tile-based design, the layout area is not increased compared to a gate-based QCA design which utilizes MVs and INVs.

Hereafter, the following assumptions are made:

(1) As applicable to a design process prior to cell deposition, the proposed design will retain only those cells which are required for deposition to implement S in the final layout.

(2) The one-dimensional clocking scheme is assumed [Orlov et al. 2000]; clocking is from left (inputs) to right (outputs). So, all cells in a tile (grid and input/output cells) are assumed to be within a single timing zone and all tiles in the same column are in the same zone.

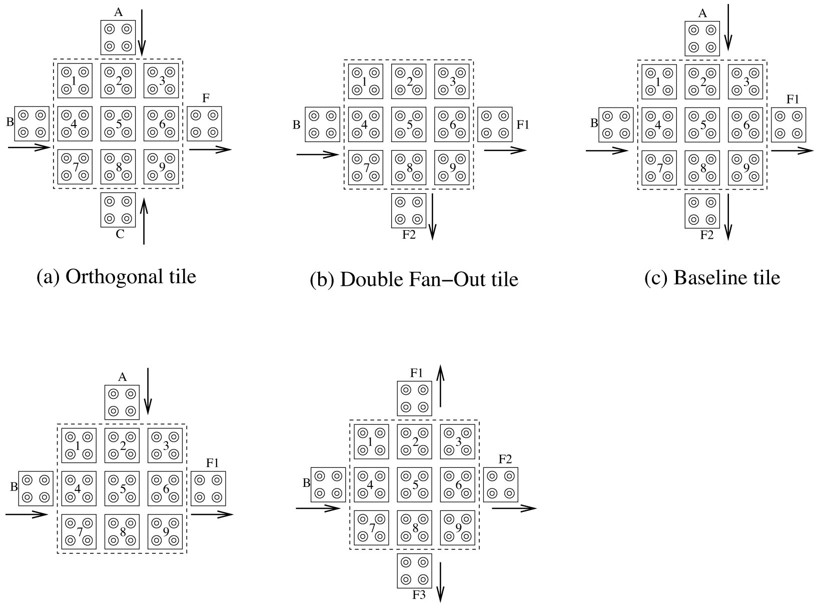
(3) The no logic state (referred to as the *undetermined state*) may occur for some patterns due to lack of definitive polarization at the output. When the polarization level of a cell is too weak, a no definite (or undetermined) logic state is said to be encountered. The value of the undetermined state is denoted by ”-”.

(4) In all simulations, the following parameters as representative of QCA with metal dots are used: cell size is 10 nm × 10 nm, the cell-to-cell distance is 2.5 nm. The dot has 2.5 nm diameter. Unless otherwise noted, a combinatorially exhaustive evaluation of the tiles and grids is pursued.

4. TILES OF A 3 × 3 GRID

An NFP grid is generated from an FP grid by selectively undepositing cells. This process changes the logic behavior of a QCA circuit; moreover, only the cells which are kept in the final layout are deposited on the substrate. So, it is interesting to compare the characteristics of tiles with different input/output cells. For the 3 × 3 grid, the five tiles (shown in Figure 5) are possible. Tiles with one input and one output are not considered due to the obvious wire function; they are referred to as interconnection (passive) tiles. An exhaustive simulation has been pursued for these tiles (with NP and NFP grids), that is, the absence of i cells $i = 0, \ldots, 8$ from each tile's layout has been investigated. Note that for all tiles, the absence of all cells results in an undetermined value at all outputs. The results of these simulations for each tile are as follows:

—The orthogonal tile has three inputs (the horizontal input cell B and the vertical input cells A and C) and one output (the horizontal output cell F), as shown in Figure 5(a). New MV-like functions (majority function with at least one input inversion) and NXOR are possible at the output by selectively undepositing cells in this tile. The MV-like function occurs due to the interaction of the cells at the corners of the tile with the center cell of the MV (i.e., cell 6). Table I shows the simulation results when at most one cell is undeposited from the orthogonal tile. Let U denote the set of undeposited cells (as labeled



(a) Orthogonal tile (b) Double Fan−Out tile (c) Baseline tile

(d) Fan−In tile (e) Triple Fan−Out tile

Fig. 5. 3 × 3 tiles with an FP grid.

Table I. Generation of Output Function by Undepositing at Most One Cell in the Orthogonal Tile

U	F	U	F
none	Maj(A, B, C)	5	B
1	Maj(A, B, C)	6	Maj(A', B, C')
2	Maj(A', B, C)	7	Maj(A, B, C)
3	Maj(A, B, C)	8	Maj(A, B, C')
4	Maj(A, B, C)	9	Maj(A, B, C)

in Figure 5) and F denote the generated output function. For example, in Table I, Maj(A', B, C) is achieved as output MV-like function provided cell 2 is not deposited from an orthogonal FP tile.

—The double and triple fan-out tiles are shown in Figure 5(b) and 5(e); these tiles have one input (given by the horizontal cell B) and two (or three) outputs (given by the output cells $F1$, $F2$ (and F)). Simulation results show that both original and complemented signals can be observed at the outputs by undepositing cells in these tiles.

—The baseline tile has two inputs (one vertical input cell A and one horizontal input cell B) and two outputs (the horizontal output cell $F1$ and the vertical output cell $F2$), as shown in Figure 5(c). This tile accomplishes wire crossing and signal complementation by selectively undepositing cells. In the FP case (as shown in Table II), this tile operates as a switch, that is, the two input signals cross each other. Signal complementation can be obtained at the

Table II. Generation of Output Function by Undepositing at Most a Cell in the Baseline Tile

U	$F1$	$F2$	U	$F1$	$F2$
none	B	A	5	A'	B'
1	B	A	6	A'	A
2	B	B	7	A	A
3	B	B	8	B	B'
4	A	A	9	B	A

Table III. Logic Functions Obtained for FP and NFP (3 × 3 Tiles)

		Triple	Double	
Baseline	Orthogonal	Fan-Out	Fan-Out	Fan-In
F_1F_2	F	$F_1F_2F_3$	F_1F_2	F_1
BA	A	BBB'	BB	A
$A'B'$	A'	$B'B'B'$	$B'B'$	B
BB	B	$BB'B'$	BB'	A'
BB'	C	BBB	$B'B$	B'
$A'A$	C'	$B'BB$		
AA	Maj(A, B, C)	$B'B'B$		
$A'A'$	Maj(A', B, C)	$BB'B$		
$B'B'$	Maj(A, B, C')	$B'BB'$		
AB	Maj(A', B, C')			
AA'	NXOR			
AB'				
$A'B$				
$B'B$				

outputs by undepositing cells; however, no inversion is generated in the crossing property.

—The fan-in tile is shown in Figure 5(d); this tile has two inputs (one vertical input cell A and one horizontal input cell B) and one output (given by the horizontal output cell F). Simulation results show that the four logic output functions (original and complement of each input signal) and the undetermined function are possible, namely A, B, A', B' and $-$.

Table III summarizes the different logic functions obtained at the output of each tile when $i = 0, \ldots, 8$ cells are undeposited. As an example, Figure 6 presents a comparison between the tile-based design of the function Maj(A', B, C) with designs obtained by SQUARES and using QCA devices in a gate-based method. The tile-based design of this function requires an area (with no input/output cell) of 9 cells, while the SQUARES-based and gate-based designs require areas of $6 \times 25 = 150$ and $7 \times 8 = 56$ cells, respectively. Additionally, a tile-based design has a smaller delay compared to the SQUARES-based and gate-based designs. One clocking zone is needed in the tile-based design, while 2 and 3 clocking zones are used in the SQUARES-based and gate-based designs, respectively. Other circuits will be discussed in Section 5.

Prior to deposition, the arrangement by which an NFP grid and an assignment of input/output cells are utilized, can significantly change the logic behavior of a tile, thus significantly affecting the layout of the final design. However,

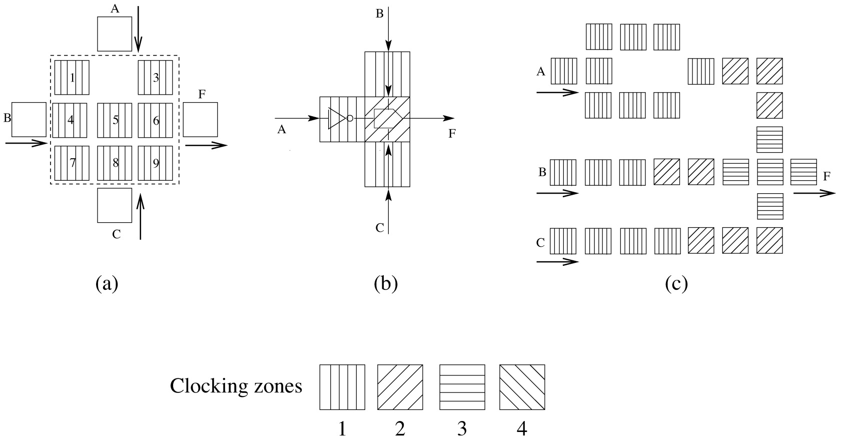


Fig. 6. Design of the Maj(A', B, C) function by (a) Tile-based, (b) SQUARES, (c) Gate-based.

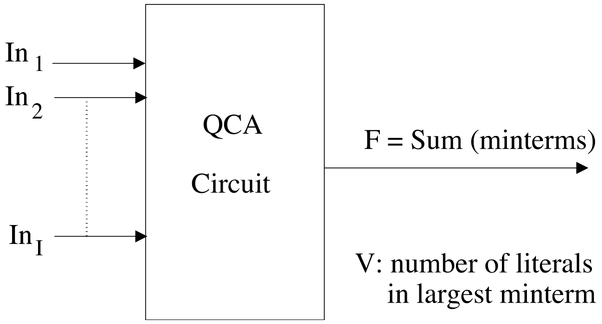


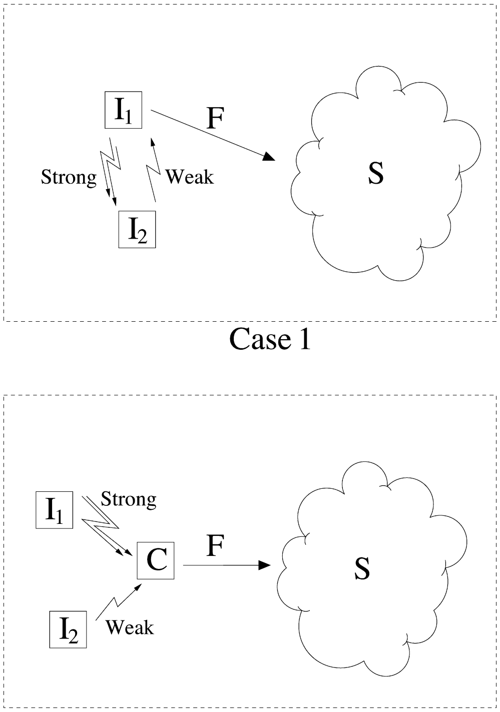
Fig. 7. QCA combinational circuit (single output) with I inputs.

as reported in Table III, simulation results have shown that only the orthogonal tile is active, while the remaining four tiles (as well as the interconnection tiles) are passive. Let I denote the number of input cells and V the number of literals found in the largest minterm of the SOP representation of the output function of a QCA circuit exclusive of the undetermined literal (Figure 7). For example, consider $F = \text{Maj}(A,B,C) = AB+AC+BC$, so $I = 3$ and $V = 2$. In the analysis below, it is assumed that the input signals are not generated through fixed-polarity cells. The following lemma characterizes the logic behavior of QCA combinational circuits.

LEMMA 1. *An output combinational function with $V = 2$ cannot be generated by a QCA circuit of $I = 2$.*

PROOF. Consider initially the scenario for $I = 1$ when generating an output combinational function with $V = 2$. For $I = 1$, the circuit can be either a wire, or an inverter (this corresponds to a passive tile in the proposed design); for both types of circuit, $V = 1$ by definition.

Consider next the scenario for $I = 2$ and a QCA circuit S (as shown in Figure 8 with two input cells I_1 and I_2 and an output function F, hence $F = f(I_1,I_2)$.



Case 1

Case 2

Fig. 8. Cases in Proof of Lemma 1.

Two distinct cases can be distinguished:

(1) *Limited Cell Interaction.* There is no additional cell in the QCA circuit except the two cells with inputs I_1 and I_2. Then, the output is determined by the Coulombic interactions among these cells as determined by the switching induced by a cell on the other cell; in this case, the solution of the thermodynamic equation (e.g., the Hamiltonian of [QCADesigner] for this circuit ensures that in steady state, the lowest energy state of the output corresponds to the polarization of the stronger cell, that is, $F = I_1$ (provided I_1 is the stronger cell) and $V = 1$.

(2) *Additional Cell Interaction.* There is at least an additional cell (denoted by C) with no input other than I_1 and I_2 (by assumption of $I = 2$); C basically acts as a center cell to the two inputs, while interacting with the remaining cells in S. C is affected by the input cells, however its polarity due to adiabatic switching [Tougaw and Lent 1994] will be determined by the stronger input cell (say I_1 with no loss of generality). Hence, depending on the position of I_1 with respect to C, C will have a polarization equal to I_1 or the complement of I_1, that is, I_1' if for example, the two cells are at 45 degree angle. As C provides the only input to the remaining cells in S (so $I = 1$), then all remaining cells cannot generate a minterm of two literals. In all cases, F has a minterm of only one literal (i.e., $F = f(I_1)$). Note that

the case in which more than single cell interactions are possible from C, does not change the function F (consisting of only one literal minterm, i.e., $V = 1$), because there is no other input signal to the remaining cells in S.

All above cases correspond to circuits that have the limited computational capability to only propagate (as fan-in or fan-out wires), or complement one of the input signals (such as the 45 degree and the INV gates), that is, $V = 1$. So by construction, the condition of $V = 2$ can not be met by a QCA circuit with $I = 2$. This concludes the proof of the lemma.

The following theorem directly follows from Lemma 1 and the basic operation of an MV.

THEOREM 1. *The generation of an output combinational function with $V = 2$ requires a QCA circuit with at least $I = 3$.*

Simulation results (such as presented previously in Section 4) have shown that Theorem 1 can be extended to the general case of V and I, that is, the generation of an output function with $V = k$ requires a QCA circuit with $I = k + 1$. At this moment, the proof of this statement remains open due to the inability to find a formal characterization to the problem due to the exponential number of combinations in the arrangements of the input cells. The authors believe that this problem will much likely fall in the NP hard domain.

Among the five considered tiles, the orthogonal tile presents unique processing features, because it can implement different MV-like functions. The following features are observed for the orthogonal tile:

- —There is no MV-like function with inversion at B. This is caused by the strong polarization of the cell aligned with the center cell of the MV.
- —As at least a single inversion can occur internally to an orthogonal tile by selectively undepositing cell(s), then B can be used as a control input and different functions can be generated. For $B = 0$, $F = \text{Maj}(A', B, C) = A'C$ and $F = \text{Maj}(A', B, C') = A'B'$. This last case corresponds to a 2-input NAND gate. Equivalently, a 2-input NOR gate can be generated by using $B = 1$ and $\text{Maj}(A', B, C')$.

The above considerations show the flexibility of MV-like logic functions which can be generated by the orthogonal tile as an active tile. As for generating logic functions, it is obvious that for an NFP grid with a small number of deposited cells, some outputs may have an undetermined value, thus making the tile unusable for design. However, in many cases, an NFP grid results in a configuration of high polarity.

5. EXAMPLES OF QCA CIRCUITS

Different circuit designs are evaluated in this section. Two figures of merit are reported: number of required clocking zones and number of cells that must be deposited on a substrate. As noted earlier, the area occupied in the layout by the 3 × 3 grid (either FP or NFP) is the same as for an INV or MV (with no input/output cells). It is assumed that the design is partitioned into columns of clocking zones and the signals flow from left to right [Antonelli et al. 2004]. The

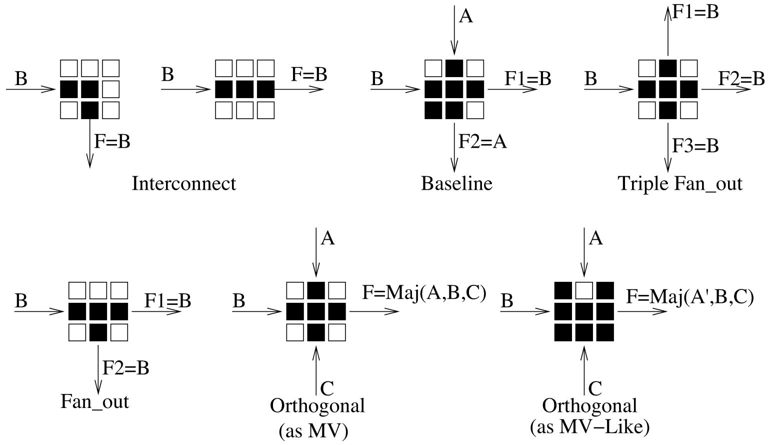


Fig. 9. Tiles used in the design of the full adder.

following restrictions are applicable to all tiles (the orthogonal tile, the double fan-out tile, the triple fan-out tile and the baseline tile) for timing purposes: (1) signal propagation is from left to right and (2) the outputs of the tile can be only used in a clocking zone to the right of the clocking zone in which the tile is located. When two tiles are placed adjacent to each other, in some cases, additional spacing for isolation may be required.

As shown next for few sample circuits, the proposed tile-based design achieves significant improvements in both area and delay compared with the SQUARES based design. The area (cell count) reduction is the result of using a smaller grid (3 × 3 instead of 5 × 5 used by the SQUARES) and the MV-like tiles that are very efficient for logic implementation. The one-dimensional clocking scheme for adiabatic switching used in the tile-based design results in a reduced delay (as measured by the number of clocking zones) compared to the SQUARES design. However, in general a modular design (such as the tile-base design proposed in this paper) will require more area than a gate-based design, this has been also encountered in CMOS (i.e., standard-cell versus full-custom layouts). The proposed tile-based design will result in a reduction in area and number of clocking zones when implementing logic functions that require selective input inversion (such as the MV-like primitives of the orthogonal tile).

5.1 One-Bit Full Adder

The one-bit full adder is analyzed first. In this design, different tiles (such as the baseline, double and triple fan-out and the orthogonal tiles) are utilized. The configurations used for these tiles are shown in Figure 9. The undeposited cells are denoted by white squares. The deposited cells are denoted by black squares. The baseline tile is used to achieve wire crossing; the double and triple fan-out tiles are used for signal routing; MV as well as MV-like functions are employed using the orthogonal tile. It is interesting that although the MV and the triple fan-out are similar in Figure 9, the arrangements in input/output cells cause the

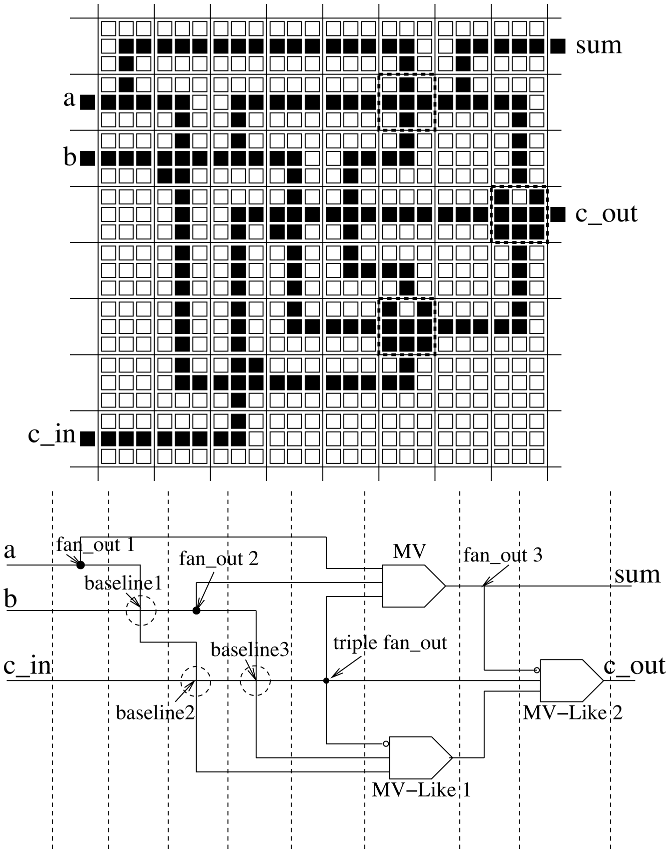


Fig. 10. Full adder using proposed tile-based design.

two tiles to function differently. Also the flow of signals is enforced by arranging the clocking zones.

The one-bit full adder is built using one MV and two MV-like (with inversion at one of the inputs) gates. The QCA layout as well as the corresponding circuit schematic are shown in Figure 10. Three baseline tiles (as wire crossing), two double fan-out tiles, one triple fan-out tile, three orthogonal tiles (as MV and MV-like) are used in this design. These tiles are connected using passive tiles, which function as wires. The MV gate and the MV-like gates are highlighted by dotted squares. In the design of a full adder, no additional isolation is needed between tiles.

The tile-based design uses the same logic schematic as the gate-based design of [Wang et al. 2003]. The QCA layout of the gate-based design [Wang et al. 2003] is shown in Figure 11. Three MVs and one INV gate are used in this design; it occupies an area of $18 \times 22 = 396$ cells. In the proposed tile-based design, since inversion can be realized using MV-like tiles, no INV is used. Therefore, it requires $8 \times 8 = 64$ tiles (an area corresponding to $64 \times 9 = 576$ cells).

The tile-based design can also be compared with the design obtained by SQUARES (shown in Figure 12). The tile-based design saves considerable cell

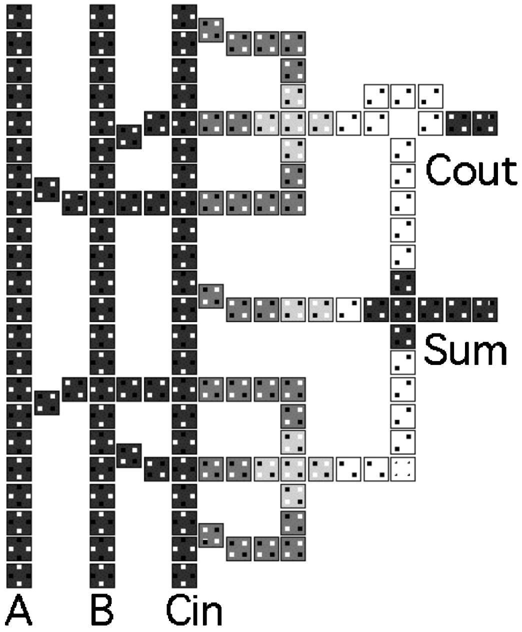


Fig. 11. Gate-based design of the full adder.

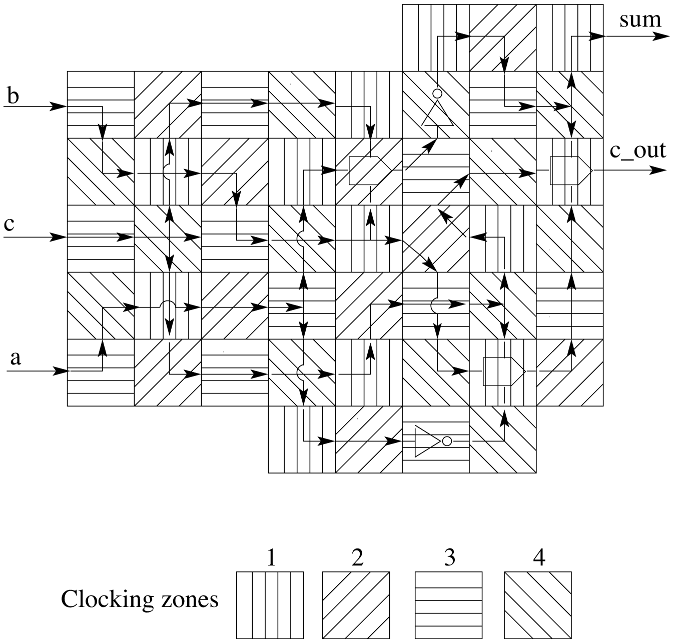


Fig. 12. Full adder using SQUARES-based design.

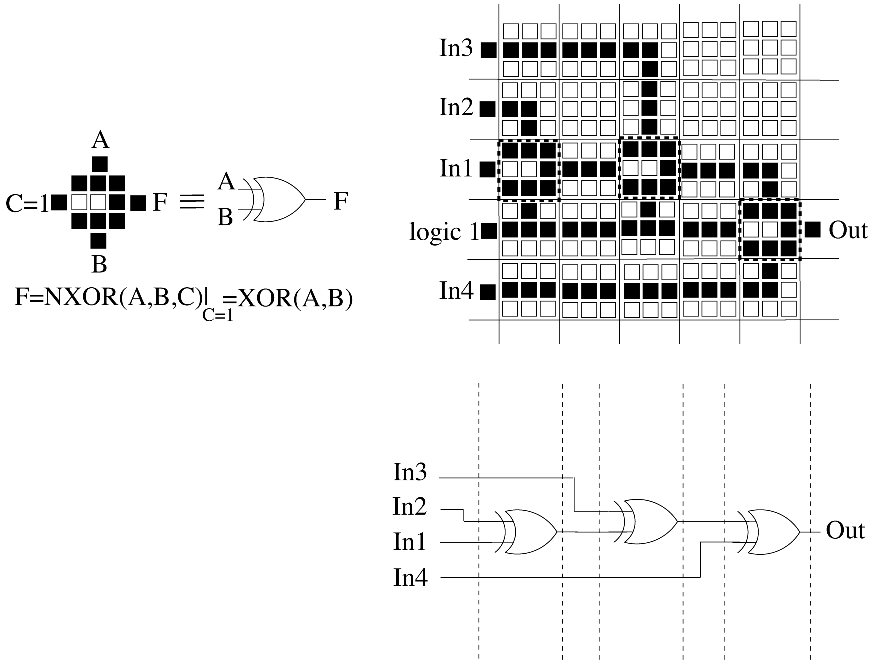


Fig. 13. 4-bit Parity Checker using proposed tile-based design.

area at a significantly reduced latency (in terms of clocking zones). Specifically, the full adder implemented by SQUARES requires $8 \times 7 = 56$ tiles with $56 \times 25 = 1400$ cells. Hence, the full adder using the 3×3 grid as part of the tile achieves a 58% area reduction. Additionally, the proposed method has a smaller delay compared to the SQUARES-based design. Only 8 clocking zones are needed in the tiled-based design, while 15 clocking zones are used in the SQUARES-based design, which corresponds to a 45% reduction in input–output latency.

5.2 Parity Checker

A 4-bit parity checker is considered as a second example; this circuit is constructed by using three NXOR gates (with logic “1” at one of the inputs). The QCA layout as well as the corresponding circuit schematic are shown in Figure 13. Two double fan-out and three orthogonal tiles (as NXOR) are used in this design. These tiles are connected using interconnection tiles, which function as wires. The NXOR gates are highlighted by dotted squares. As in the case of the full adder, no additional isolation is needed.

As NXOR can be realized using orthogonal tiles, no AND, OR and INV gates are used. Therefore the design using the proposed tile-based approach requires $5 \times 5 = 25$ tiles (corresponds to an area of $25 \times 9 = 225$ cells). A gate-based design requires an area of $51 \times 29 = 1479$ cells (Figure 14); therefore, a tile-based design results in a significant area reduction (68%).

The proposed tile-based design can be compared with the design obtained by the SQUARES (as shown in Figure 15). The parity checker implemented by

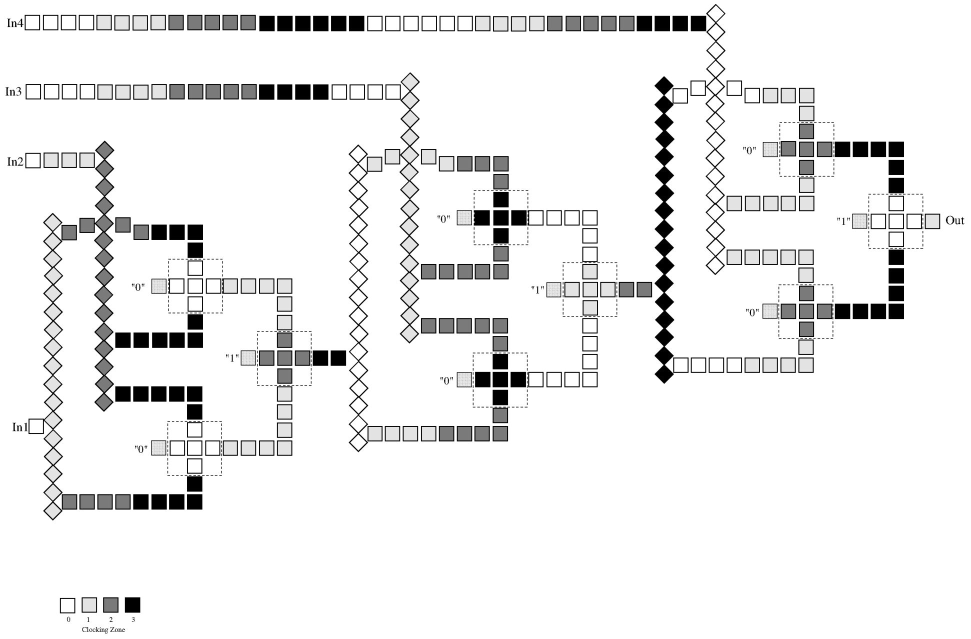


Fig. 14. Gate-based 4-bit Parity Checker.

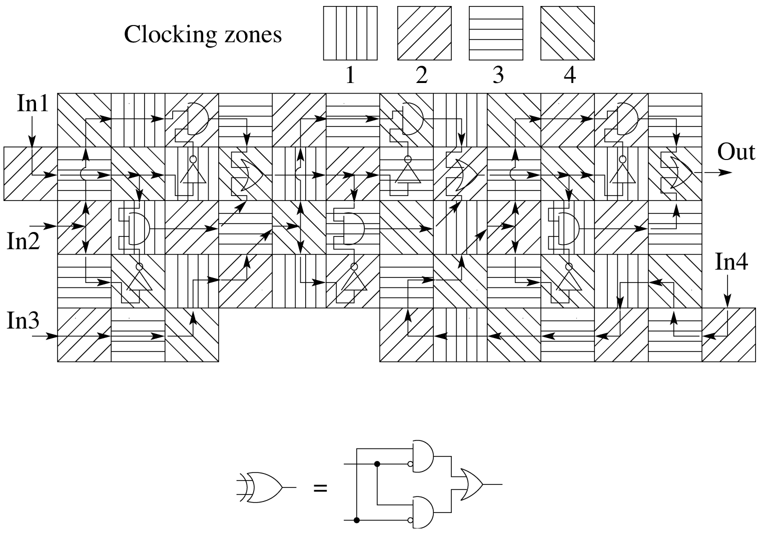


Fig. 15. 4-bit Parity checker design using SQUARES-based approach.

SQUARES requires 8 × 7 = 56 tiles (corresponds to an area of 56 × 25 = 1400 cells). Hence, the parity checker using the 3 × 3 grid and corresponding tiles achieves a 58% area reduction. Again the tile-based design has proven to have a reduced delay. A 74% reduction in the number of clocking zones (5 versus 19) is achieved compared with the SQUARES-based design.

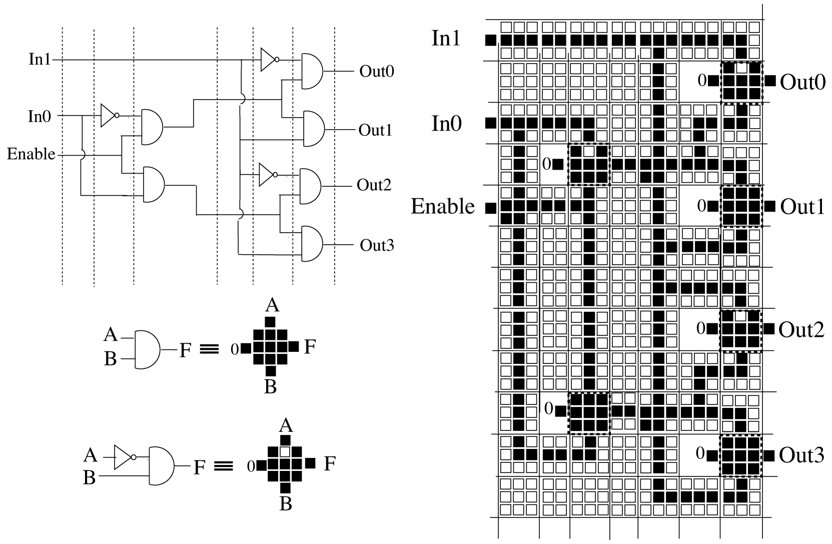


Fig. 16. Tile-based 2-to-4 Decoder.

5.3 2-to-4 Decoder

A 2-to-4 decoder is built using three MVs and three MV-like (with inversion at one of the inputs) gates. The QCA layout as well as the corresponding circuit schematic are shown in Figure 16. Three baseline tiles (as wire crossing), seven double fan-out tiles, six orthogonal tiles (as MV and MV-like) are used in this design. These tiles are connected using interconnection tiles, which function as wires. The MV gate and the MV-like gates are highlighted by dotted squares. Additional spacing for isolating orthogonal tiles from baseline tiles, and the fixed inputs of the orthogonal tiles from wires, are required.

This design requires $12 \times 5 = 60$ tiles and $12 \times 2 \times 2 = 48$ isolation cells (588 cell area). Compared with a gate-based design (Figure 17), which occupies an area of 400 cells, a tile-based design results in a 47% overhead. The 2-to-4 decoder implemented by SQUARES (as shown in Figure 18) requires $8 \times 6 = 48$ squares (corresponds to an area of $48 \times 25 = 1200$ cells). Hence, the 2-to-4 decoder using the 3×3 grid and its tiles achieves a 51% area reduction and a 45% reduction in the number of clocking zones (6 versus 11 for the SQUARES-based design).

5.4 2-to-1 MUX

The 2-to-1 Multiplexer (MUX) is built with two MVs and one MV-like (with inversion at one of the inputs) gate. The tile-based QCA layout as well as the circuit schematic are shown in Figure 19. This design requires 17 tiles, a total of $17 \times 9 = 153$ cells in area. The gate-based design is shown in Figure 20 and the SQUARES design is shown in Figure 21. The MUX implemented by a gate-based design consists of one INV and three MV gates and it occupies an

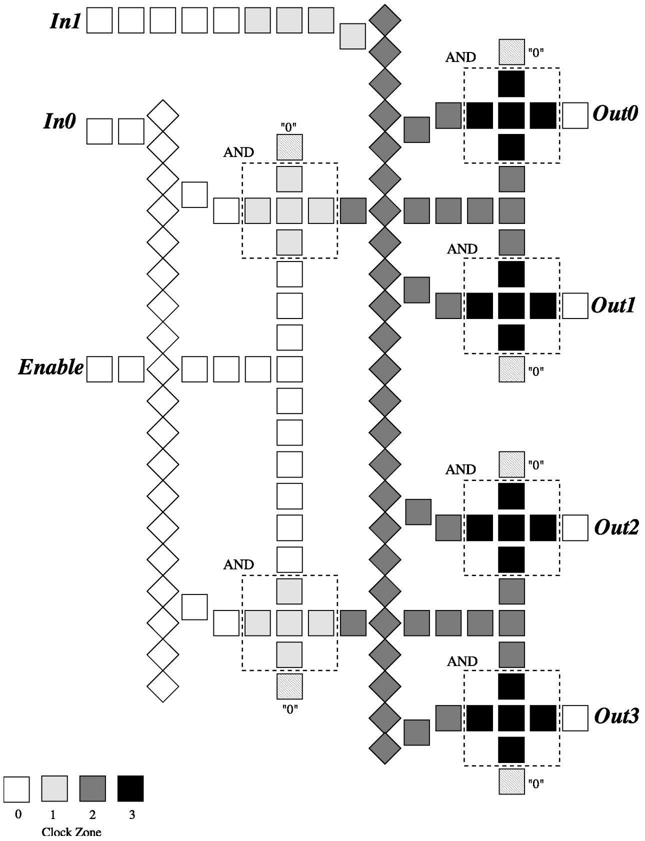


Fig. 17. Gate-based 2-to-4 decoder.

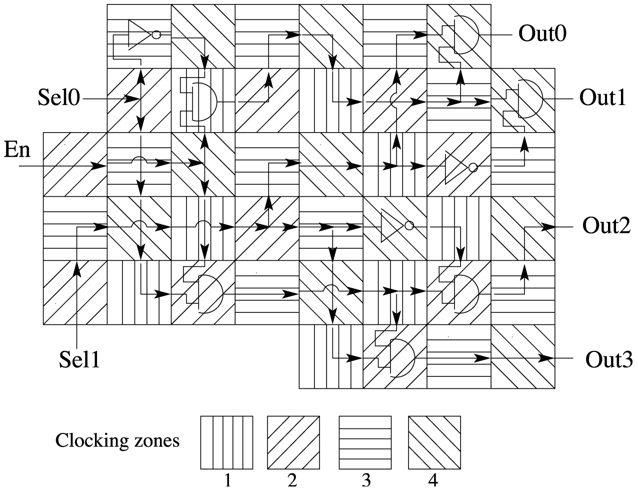


Fig. 18. 2-to-4 Decoder using SQUARES-based approach.

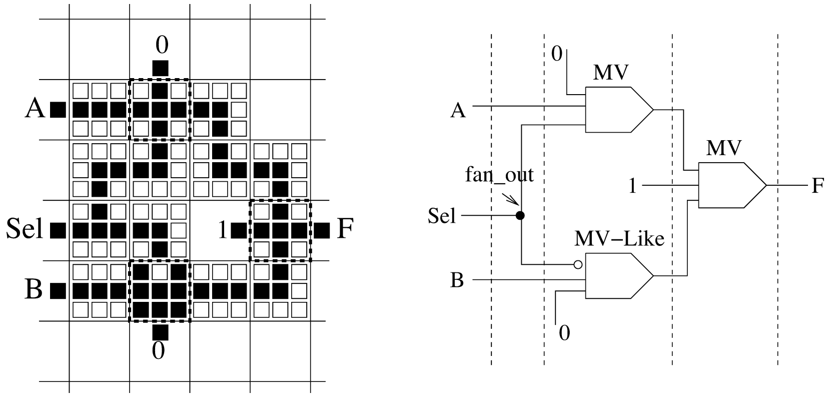


Fig. 19. Tile-based 2-to-1 MUX.

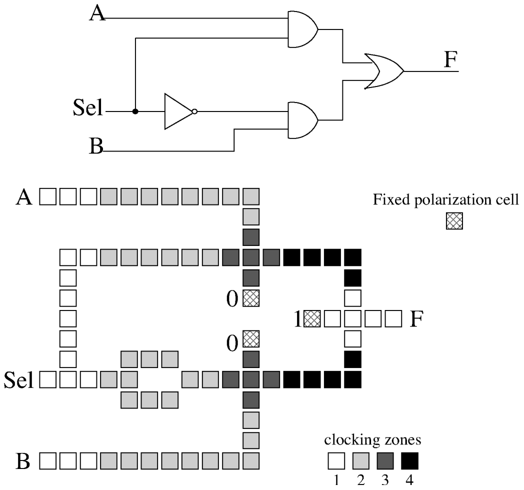


Fig. 20. Gate-based 2-to-1 MUX.

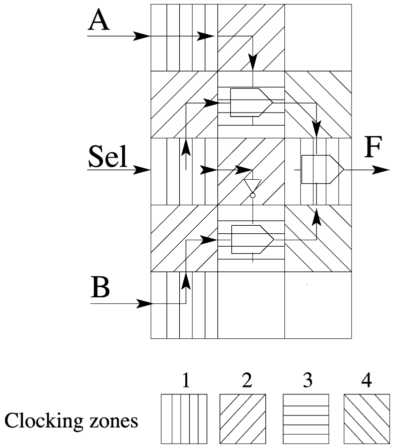


Fig. 21. SQUARES-based 2-to-1 MUX.

Table IV. Circuits Using Tiles, SQUARES and Gates

Circuit		2-to-4 Decoder	One Bit Adder	Parity Checker	2-to-1 MUX
Tile-based	# of Tiles	60	64	25	17
	Total # of cells	588	576	225	153
	# Clk zones	6	8	5	4
SQUARES	# of SQUARES	48	56	56	12
	Total # of cells	1200	1400	1400	625
	# Clk zones	11	15	19	5
Gate-based	Total # of cells	400	396	1479	234
	# Clk zones	5	5	22	5

area of 13 × 18 = 234. The tile-based design achieves a 34.6% area reduction. The SQUARES design needs 12 squares, therefore a total of 12 × 25 = 625 cells in area. SQUARES has a significantly higher area overhead compared to the tile-based design. The delay for both the gate-base design and SQUARES is 5, while for the tile-based design it is 4.

Table IV summarizes the results implementing the analyzed circuits using the proposed tile-based design, SQUARES and the gate-based design (using MV and INV gates). Note that in some cases (such as for the parity checker) a tile-based design requires a small number of deposited cells than a gate-based design.

6. CONCLUSION

This article has presented a novel design of combinational circuits by employing basic blocks (referred to as tiles) for assembling QCA circuits prior to cell deposition on a substrate. In this paper, a tile is a square grid of cells with input/output cells. Grids can be fully populated (FP) or non fully populated (NFP). The 3 × 3 grid provides very versatile operation. With an assignment of input/output cells, different tiles can be utilized for generating a variety of combinational functions. As proposed in this article, the basic logic primitive is the MV-like tile; this tile performs the majority function with selective inversion at the input. By combining the functions of MV and INV, the MV-like tile offers an advantage in terms of area efficiency. These tiles have been extensively simulated and analyzed; the presented analysis has confirmed that NFP grids can be efficiently used in designing QCA circuits. Different circuit designs have been presented and compared with SQUARES as well as a traditional QCA gate-based design. It has been shown that a tile-based design achieves considerable area as well as delay (the number of clocking zones between inputs and outputs) reduction compared with SQUARES, (and in some cases also compared with a traditional gate-based design). The generation of new combinational functions (such as MV-like functions) and the simple arrangement in the clocking zones make tiles a viable design technique for QCA.

Current research involves the investigation of defect tolerance of tile-based QCA circuits for metal and molecular implementations.

REFERENCES

AMLANI, I. ORLOV, A. O., TOTH, G., LENT, C. S., BERNSTEIN, G. H., AND SNIDER, G. L. 1999. Digital logic gate using quantum-dot cellular automata. *Science 284*, 5412, 289–291.

ANTONELLI, D. A., CHEN, D. Z., DYSART, T. J., HU, X. S., KAHNG, A. B., KOGGE, P. M., MURPHY, R. C., AND NIEMIER, M. T. 2004. Quantum-dot cellular automata (QCA) circuit partitioning: Problem modeling and solutions. In *Proceedings of the Design Automation Conference*, 363–368.

BERNSTEIN, G. H., HU, W., HANG, Q., SARVESWARAN, K., AND LIEBERMAN, M. 2004. Electron beam lithography and liftoff of molecules and DNA rafts. In *Proceedings of the IEEE Conference on Nanotechnology*. IEEE Computer Society Press, Los Alamitos, CA, 201–203.

BERZON D. AND FOUNTAIN, T. J. 1999. A memory design in QCAs using the SQUARES formalism. In *Proceedings of the 9th Great Lakes Symposium on VLSI*. 166–169.

DIMITROV, V. S., JULLIEN, G. A., AND WALUS, K. 2002. Quantum-dot cellular automata carry-look-ahead adder and barrel shifter. In *Proceedings of the IEEE Emerging Telecommunications Technologies Conference*. IEEE Computer Society Press, Los Alamitos, CA, pp. 2/1–2/4.

FROST, S. E., RODRIGUES, A. F., JANISZEWSKI, A. W., RAUSCH R. T., AND KOGGE, P. M. 2002. Memory in motion: A study of storage structures in QCA. In *Proceedings of the 1st Workshop on Non-Silicon Computation*.

HUANG, J., MOMENZADEH, M. SCHIANO, L., AND LOMBARDI, F. 2005a. Simulation-based design of modular QCA circuits. In *Proceedings of the IEEE Conference on Nanotechnology* (Nagoya, Japan). IEEE Computer Society Press, Los Alamitos, CA (Paper WE-P7-1, IEEE CD-ROM 05TH8816C).

HUANG, J. MOMENZADEH, M., SCHIANO, L., OTTAVI, M., AND LOMBARDI, F. 2005b. A methodology for tile-based Design of QCA combinational circuits. Internal Report, ECE Department, Northeastern Univ., Boston, MA, available on request.

JIAO, J., LONG, G. L., GRANDJEAN, F., BEATTY, A. M., AND FEHINER, T. P. 2003. Building blocking for the molecular expression of QCA, isolation and characterization of a covalently bounded square array of two ferrocenium and two ferrocene complexes. *J. Amer. Chem. Soc. 125*, 25, 7522–7523.

LENT, C. S., TOUGAW, P. D., AND POROD, W. 1994. Quantum cellular automata: The physics of computing with arrays of quantum dot molecules. In *PhysComp '94: Proceedings of the Workshop on Physics and Computing*. IEEE Computer Society Press, Los Alamitos, CA, 5–13.

MCCLUSKEY, E. 1986. *Logic Design Principles*, Prentice-Hall, Englewood Cliffs, NJ.

NIEMIER, M. T. AND KOGGE, P. M. 1999. Logic-in-wire: Using quantum dots to implement a microprocessor. In *Proceedings of the International Conference on Electronics, Circuits, and Systems (ICECS'99)*. 1211–1215.

NIEMIER, M. T. AND KOGGE, P. M. 2001. Problems in designing with QCAs: Layout=timing. *Int. J. Circ. Theory Appl. 29*, 1, 49–62.

NIEMIER, M. T., RODRIGUES, A. F., AND KOGGE, P. M. 2002. A potentially implementable FPGA for quantum dot cellular automata. In *Proceedings of the 1st Workshop on Non-Silicon Computation (NSC-1)*, (held in conjunction with 8th International Symposium on High Performance Computer Architecture (HPCA-8)).

ORLOV, A. O., AMLANI, I., KUMMAMURU, R., RAJAGOPAL, R., TOTH, G., LENT, C. S., BERNSTEIN, G. H., AND SNIDER, G. L. 2000. Experimental demonstration of clocked single-electron switching in quantum-dot cellular automata. *Appl. Phys. Lett.*, *77*, 2, 295–297.

QCADESIGNER HOME PAGE: www.atips.ca/projects/qcadesigner/.

QI, H., SHARMA, S., LI, Z., SNIDER, G. L., ORLOV, A. O., LENT C. S., AND FEHINER, T. P. 2003. Molecular quantum cellular automata cells: Electric field driven switching of a silicon surface bound array of vertically oriented two-dot molecular QCA. *J. Amer. Chem. Soc.*, *125*, 49, 15250–15259.

TAHOORI, M., MOMENZADEH, M., HUANG, J., AND LOMBARDI, F. 2004. Defects and faults in quantum cellular automata at nano scale. In *Proceedings of VLSI Test Symposium*. 291–296.

TOUGAW, P. D. AND LENT, C. S. 1994. Logical devices implemented using quantum cellular automata. *J. Appl. Phys. 75*, 3, 1818–1825.

TOUGAW, P. D. AND LENT, C. S. 1996. Dynamic behavior of quantum cellular automata. *J. Appl. Phys.*, *80*, 8, 4722–4736.

WALUS, K., BUDIMAN, R. A., AND JULLIEN, G. A. 2002. Effects of morphological variations of self-assembled nanostructures on quantum-dot cellular automata (QCA) circuits. In *Proceedings of Frontiers of Integration: An International Workshop on Integrating Nanotechnologies*.

WALUS, K., VETTETH, A., JULLIEN, G. A., AND DIMITROV, V. S. 2003. RAM design using quantum-dot cellular automata. In *Proceedings of the NanoTechnology Conference*, Vol 2. pp. 160–163.

Wang, W., Walus, K., and Jullien, G. A. 2003. Quantum-dot cellular automata adders. In *Proceedings of the IEEE Conference on Nanotechnology*, IEEE Computer Society Press, Los Alamitos, CA, 461–464.

Received January 2005; revised July 2005 and December 2005; accepted December 2005