

A Universal Gate for Combinational Design of QCA Circuits

Young Bok Kim, Yong-Bin Kim, and F. Lombardi

Dept of Electrical and Computer Engineering

Northeastern University

Boston, USA

{youngbok,ybk,lombardi}@ece.neu.edu

Abstract— As an emerging technology, Quantum-dot Cellular Automata (QCA) has been extensively analyzed in the technical literature for implementing logic circuits; in this technology, the basic combinational circuit is the majority voter with three inputs. In this paper, the design of a logic gate which implements any combinational function of at most three input variables is presented. This type of gate is generally referred to as a universal gate and is often used as logic resource in array structures such as FPGAs. This paper investigates a logic characterization of the universal gate using two different techniques found in the technical literature: AND/OR-based synthesis and MV-based synthesis. Important design features of the gate (such as the utilization of the majority voting function or MV and inverters) and design issues revolving around signal routing of the input signals are presented along with the introduction of non-blocking interconnect circuit in this paper to extend the logic capabilities.

Keywords—component QCA, emerging technologies, quantum computing, universal gate.

I. INTRODUCTION

At nano regimes, computation is substantially different from conventional VLSI; so-called emerging technologies have been investigated to take advantage of extremely small feature size and high device density for implementing new computational paradigms [12]. Among emerging technologies, Quantum-dot Cellular Automata (QCA) [1][4] relies on novel design concepts to exploit new physical phenomena (such as Coulombic interactions), and implement unique paradigms (such as memory-in motion and processing-by-wire [3][2]). Recent developments in QCA manufacturing (involving molecular implementations) have substantially changed the nature of processing; at very small features sizes it is anticipated that either self-assembly, or large scale cell deposition on insulated substrates will be utilized. In these implementations, QCA cells (each made of two dipoles) will be deposited on parallel V-shaped tracks; in this case, the QCA cells will be arranged in a dense layout and computation will occur among adjacent cells.

QCA design was initially proposed in [5] by the so-called SQUARES methodology; SQUARES utilizes an 5×5 square grid for constructing specific QCA devices (such as the MV and the INV) and for establishing signal connections in larger circuits.

In [10] the computational capabilities of fully (uncustomized) and partially (customized) square grids have been investigated; it has been shown [10] by analysis and then simulation that uncustomized QCA grids have similar

properties for generating combinational logic functions. Different so-called blocks can be designed by adding input and output cells to a grid. These blocks generate combinational functions under various latching arrangements. Unfortunately, no INV function can be generated in a fully populated grid. Many combinational functions can be generated using not fully populated grids. However, it has been shown [9] [10] that for QCA the functions with at most 3 variables (such as the MV) provide the basis for efficient combinational design. This paper addresses the QCA design of a so-called universal gate for three inputs, i.e. a QCA circuit which can generate any combinational function using three input signals (in both true and complemented forms). The universal gate represents the basic logic block by which programmable circuits could be implemented in a fashion reminiscent of logic resources in Field Programmable Gate Arrays (FPGAs) [15] for VLSI. In this paper, logic design for a universal gate with 3 input signals is pursued using different synthesis techniques which are tailored to QCA, namely the AND/OR and MV-based approaches [14]. To extend the logic capabilities of a universal gate to an array (such as a FPGA), routing circuitry must be also considered in the design; these circuits are required for the distribution of the true and complemented signals as well as constant polarization signals to the universal gate. A non-blocking interconnect circuit is introduced for full connectivity of the signals.

II. 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. 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 1. 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).

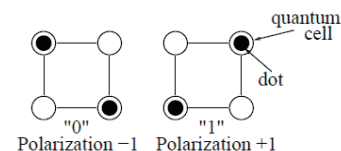


Figure 1 QCA cell and polarization states

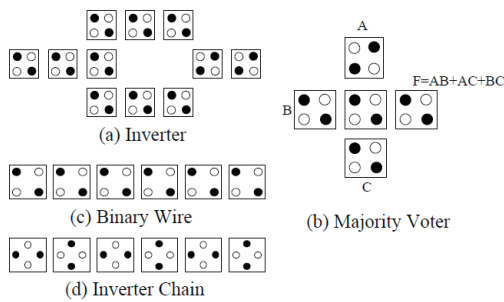


Figure 2 Basic QCA devices

III. STRUCTURED QCA DESIGN

Nano-based circuits and systems such as QCA are confronted with new challenging problems. For example, manufacturing of QCA must take into consideration the very large density (especially for molecular technologies) of these implementations. Moreover, circuits will be assembled from bottom-up to account for the extensive spatial distribution of the cells deposited (or grown) on the substrate. Therefore, it is anticipated that the building blocks of QCA as well as other emerging technologies should be modular to allow versatility in assembly and design. In [5], the so-called SQUARES methodology has been proposed for QCA; the basic building block in SQUARES is a 5×5 QCA cell grid. [5] embeds different circuits (such as the MV and INV) in the 5×5 grid. However, this methodology results in considerable overhead with respect to area utilization and more importantly the number of required clocking zones. Moreover, 15 clocking zones are required due to the complex arrangement for circuit-level timing among the SQUARES blocks.

The logic capabilities of structured design have been addressed in [9]. Prior to deposition, a non fully populated grid and an assignment of input/output cells to each side of the grid can be utilized to significantly change the logic behavior, thus significantly affecting the layout of the design. 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).

Lemma 1: An output function with $V = 2$ can not be generated by a QCA circuit of $I = 2$.

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

Simulation results in [9] have shown that Theorem (1) can be extended to the general case of V and I , i.e. the generation of an output function with $V = k$ requires a QCA circuit with $I = k+1$. Even if currently, 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, it is likely that problem will much fall in the NP domain.

IV. QCA SYNTHESIS

At logic level, two synthesis approaches are applicable to QCA-based design. The first approach referred to as AND/OR

based synthesis has been recently proposed specifically for QCA combinational circuits [7]. This approach reduces the number of MV gates required for computing three variable Boolean functions to facilitate the conversion of SOP expressions into QCA majority logic. Thirteen standard functions (exhaustively found and proposed in [8]) are utilized to completely represent all three-variable Boolean functions. Using a three cube representation, an iterative procedure is proposed [7] to generate a reduced majority gate expression which is amenable to QCA. The generated expression however is not always optimal, because QCA based designs of three levels of MVs are in some cases found (it is well know that any combinational function can be implemented with any gate by a two-level circuit network).

The second synthesis approach which can be applied to QCA, is the so-called MV-based synthesis [14]. This approach relies on the logic analysis of the majority function as an instance of threshold logic. Threshold logic has been extensively analyzed in the past and the majority threshold function of three variables (i.e. corresponding to the MV) is equivalent to a logic representation which can be easily implemented in QCA. Synthesis under the technique of [14] is based on identifying negated or permuted variables of a function such that restrictions can be generated to comply with the voting nature (such as agreement or disagreement) of this threshold function. An iterative process which can be extended to the general case of n variable voting functions, is required to establish the function restrictions for the specified and unspecified minterms in the SOP representation.

V. UNIVERSAL GATE

A universal gate with n inputs is formally defined [6] as a combinational circuit that can implement all possible functions; this circuit can generate at the single output any and all 2^n product terms and the exhaustive combinations of the product terms [6] as in a SOP representation. To utilize a universal gate in larger designs, routing must be considered. While full connectivity of the signals is often not required, the non-blocking nature of this arrangement establishes the worst case complexity of this type of circuit. Therefore, a QCA universal gate consists of two basic types of resource:

- The interconnect resource consists of three fabrics: (a) a distribution network (denoted by DN); (b) K parallel 8-to-1 multiplexers (denoted by MUX); (c) a line interconnect (denoted by LI). The distribution network has 8 input lines denoted as $i_h, h=1, \dots, 8$ corresponding to the three input signals (a, b, c) their complements (a', b', c') as well as the two fixed polarity values for the control cells ("0", "1"). An i_h is connected to every h th input of a 8-to-1 multiplexer, i.e. each i_h has a fanout of K . The single output line of each j th multiplexer ($j=1, \dots, K$) is connected to the j th input line of the logic resource. The interconnect resource provides full connectivity between its inputs and outputs (no blocking among signals).
- The logic resource may receive as inputs the outputs of the interconnection fabric (if present); the logic fabric has K input lines (denoted as $I_j, j=1, \dots, K$). K is

Identify applicable sponsor/s here. If no sponsors, delete this text box. (sponsors)

dependent on the structure of the logic resource as generated through a logic synthesis process. Each I_j can be connected (through the interconnection fabric) to any of the i_h ; so I_j can take as value any of the three literals (A, B, C), their complements (A', B', C') and two fixed value signals ("0", "1") corresponding to the fixed polarity values for the control cells in the MVs. The output is a single signal given by the output function F ; F can be any combinational function of 3 variables, i.e. $F = \sum \text{from } i M_j$ where M_j is the j th product minterm ($j=0, \dots, 7$) and $\sum \text{from } i$ denotes the ORing of the minterms (as SOP representation).

If the universal gate consists only of the logic resource, then this gate is referred to as an unrouted universal gate (denoted by U_U). If both resources are present, then it is referred to as the routed universal gate (denoted by U_R).

The process, by which the logic resource of a universal gate is generated, can be thought as an iterative procedure by which circuits implementing each and every specified combinational function are combined into a single circuit. A further restriction is given by the labeling of the inputs to the circuits for logic compatibility. The procedure employed in this paper is given by the following process:

- 1) The circuit implementing each of the 13 standard functions f_i is generated using the selected synthesis algorithm. Let G be an empty graph, $i=1$.
- 2) Each circuit implementing f_i is described by a directed labelled graph $g_i=(E_i, V_i)$ where the vertex V_i consists of MVs as vertices and the set E_i consists of the directed edges connecting the vertices as well as the primary input edges. Each primary input edge is labelled by the corresponding signal (either true literal, or complemented literal or control cell binary value).
- 3) The isomorphism of each g_i with G is then established, i.e. whether the circuit represented by G can also implement g_i or vice versa. Modify G appropriately. Increment i .
- 4) If i is less than 14 go back to (2). Else continue.
- 5) The resulting G is the unrouted universal gate.

As graph isomorphism is NP complete [13], the above procedure has an exponential complexity; however for this application, the graph representation of the circuit is very simple, therefore its execution is not excessive. Moreover, for QCA graph isomorphism implies not only the compatibility of the logic input signals, but also the possible presence of constant values (0 and 1) as fixed polarity control inputs to the MV

VI. UNIVERSAL GATE DESIGNS

A. AND/OR-based Synthesis

The unrouted universal gate as well as the implementations of all 13 standard functions using this universal gate is shown in Figure 5(a). 3 clocking zones and 6 MVs are needed in this implementation; three of the MVs are programmed to

implement the AND function by setting one of the input permanently to 0 (fixed polarization control input), so $K=10$. This universal gate is not optimal as a three-level MV network implementation is required. The routed universal gate is shown in Figure 6(a). Let T denote the number of tiles and N denote the number of cells ($C=9T$). C and Z denote the number of control cells and clocking zones respectively. These figures of merit are given in Table I.

B. MV-based Synthesis

The universal gate generated using the MV-based synthesis approach is shown in Figure 5(b); the implementations of the 13 standard functions of [7] are also shown. In this case, U_U consists of 4 MVs (arranged in a two-level configuration), one of the MV is always programmed to implement the AND function by setting one of the control inputs permanently to "0". The universal gate is shown in Figure 6(b); the hardware and timing requirements are shown in Table II.

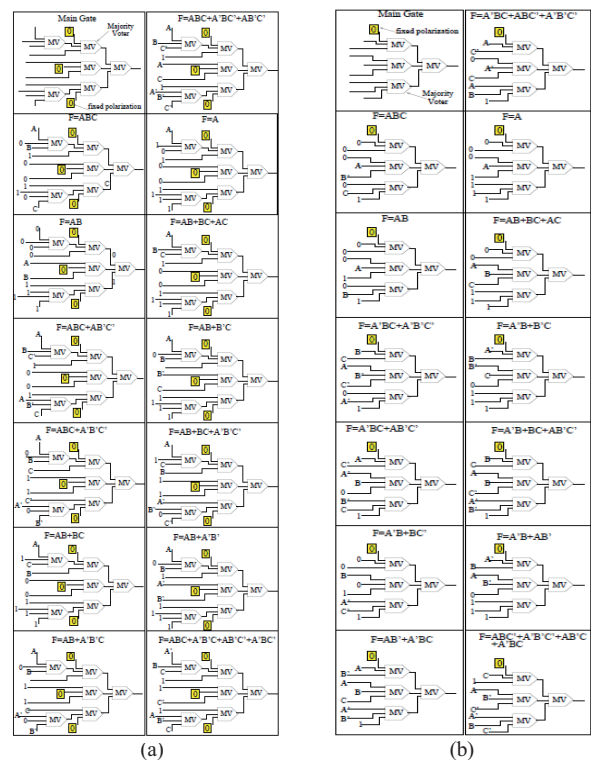


Figure 3 Unrouted Universal Gate and 13 standard function implementations, (a) AND/OR-based (b) MV-based

TABLE I UNIVERSAL GATE, AND/OR-BASED SYNTHESIS

	T	N	C	Z
DN	$251 \times 10 = 2510$	$2501 \times 9 = 22509$	0	8
MUX	$209 \times 10 = 2090$	$2090 \times 9 = 18810$	$31 \times 10 = 310$	9
LI	685	$685 \times 9 = 6165$	0	1
U_U	29	$29 \times 9 = 261$	3	3
U_R	5314	$5314 \times 9 = 47826$	313	21

TABLE II UNIVERSAL GATE, MV-BASED SYNTHESIS

	T	N	C	Z
DN	$251 \times 8 = 2008$	$2008 \times 9 = 18072$	0	8
MUX	$210 \times 8 = 1680$	$1680 \times 9 = 15120$	$31 \times 8 = 248$	9
LI	440	$440 \times 9 = 3960$	0	1
U_U	21	$21 \times 9 = 189$	1	2
U_R	4149	$4149 \times 9 = 37341$	249	20

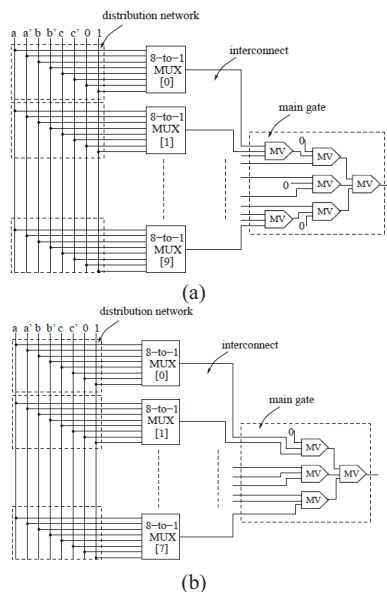


Figure 4 Routed Universal Gate, (a) AND/OR-based (b) MV-based

The unrouted universal gate (generated by the MV-based synthesis and the graph isomorphism procedure presented previously) is optimal in many respects:

- 1) The 4 MVs represent the minimum number of devices in QCA for a full two-level network implementation; moreover, the two-level implementation also meets the criterion of minimality for combinational function for logic design.
- 2) The two-level network requires only 2 clocking/timing zones which is optimal for a two-level MV implementation.
- 3) The number of inputs to the unrouted universal gate is the same as the number of possible values in a QCA circuits of 3 input signals, i.e. the 3 true variables, the 3 complemented variables and the 2 control values to the MVs ("0" and "1"), so $K=8$. This is a necessary and sufficient condition of optimality (the 3 MVs at the first level of the circuit implementation have 9 inputs, of which one input is fixed to the fixed polarity value of 0).

VII. DISCUSSION AND CONCLUSION

Using the designs presented in previous sections, the following features can be observed. As an optimal technique, MV-based synthesis [14] results in an unrouted universal gate with lower hardware and timing requirements than the AND/OR based synthesis. The arrangement by which the 13 standard combinational functions can be embedded into a universal gate has been resolved using a graph isomorphic model which has resulted in an universal gate of optimal configuration (a 2-level network arrangement made of 4 MVs and with only 8 inputs corresponding to the minimum possible number of different values in QCA). By comparison, the AND/OR-based synthesis generates a universal gate of 3-levels, requiring 3 clocking/timing zones and 13 inputs (of which 3 are fixed polarity). As established in previous sections, routing

requirements for an universal gate are very cumbersome and account for a large hardware overhead. This implies that communication networks with efficient routing capabilities (probably non fully connected as in the proposed designs) must be investigated prior to a possible FPGA implementation in QCA. This result supplements and confirms the finding of [11] on the urgent need for these circuits in QCA. Moreover, this paper has shown that the QCA design of logic circuits is challenged with unique features at logic level; in this respect, it has been proved that universal gate design offers the advantage of versatility and ease of implementation using the basic QCA device primitives of MV and INV. However, due to the high complexity unrestricted and non-blocking routing to the universal gate poses severe limitations for its applicability to programmable QCA architectures such as FPGA. Current investigation is pursued to establish the viability of the universal gate as standard cell in a QCA design library for combinational logic design.

REFERENCES

- [1] C.S. Lent , P.D. Tougaw and W. Porod, "Quantum Cellular Automata: The Physics of Computing with Arrays of Quantum Dot Molecules", PhysComp '94: Proceedings of the Workshop on Physics and Computing, IEEE Computer Society Press, pp. 5-13, 1994.
- [2] M.T. Niemier and P.M. Kogge, "Problems in designing with QCAs: layout=timing", International Journal of Circuit Theory and Applications, vol.29(1), pp. 49-62, 2001.
- [3] S.E. Frost, A.F. Rodrigues, A.W. Janiszewski, R.T. Rausch and P.M. Kogge, "Memory in Motion: A Study of Storage Structures in QCA", 1st Workshop on Non-Silicon Computation, 2002.
- [4] C.G.Smith, "Computation Without Current", Science, vol. 284(2), p. 274, 1999.
- [5] D. Berzon and T.J. Fountain, "A memory design in QCAs using the SQUARES formalism," Proceedings Ninth Great Lakes Symposium on VLSI, pp. 166-169, 1999.
- [6] N. N. Biswas, "Logic Design Theory," Prentice Hall, Englewood Cliffs, 1993.
- [7] R. Zhang, K. Walus, W. Wang and G.A. Jullien, "A Method of Majority Logic Reduction for Quantum Cellular Automata," IEEE Transactions on Nanotechnology, vol 3, no. 4, pp. 443-450, 2004.
- [8] J.R. Koza, "Genetic Programming: on the Programming of Computer by Natural selection," MIT Press, Cambridge, 1992.
- [9] J. Huang, M. Momenzadeh, M. Ottavi, L. Schiano and F. Lombardi, "A Pre-deposition Methodology for Tile-Based Design of QCA Combinational Circuits, internal report, 2004. K. Elissa, "Title of paper if known," unpublished.
- [10] M. Momenzadeh, J. Huang, M. Ottavi, N. Park and F. Lombardi, "Computing with Grids of QCA Cells," Internal report, 2004
- [11] M.T. Niemier, A.F. Rodrigues and P.M. Kogge, "A Potentially Implementable FPGA for Quantum Dot Cellular Automata," 1st Workshop on Non-Silicon Computation (NSC-1), held in conjunction with 8th Int. Symp. on High Performance Computer Architecture (HPCA-8), Boston, MS., 2002.
- [12] R.Compano, L.Molenkamp, D.J.Paul, "Technology Roadmap for Nanoelectronics", European Commission IST programme, Future and Emerging Technologies
- [13] J. D. Ullman, "Computational Aspects of VLSI," Computer Science Press, Rockville, Md., 1984.
- [14] S. Muroga, "Threshold Logic and its Applications," Wiley Interscience, New York, 1971.
- [15] J.V. Oldfield R.C. Dorf, "Field-Programmable Gate Arrays," Wiley Interscience, New York, 1995.