Australian Journal of Basic and Applied Sciences, 5(11): 1817-1823, 2011
ISSN 1991-8178

Reconfigurable Logic Based On Quantum-dot Cellular Automata

Homayun Motameni and Behnam Montazeri

Department of Computer Engineering, Islamic Azad University, Arak Branch Arak, Iran.

Abstract: In some special applications, the type of operation that a digital system should do may not be obvious in advance and in different conditions we need different outputs from this system. In these types of applications we need reconfigurable hardwares. Nowadays it is common to use FPGA technology for designing of these programmable blocks. But we should consider novel technologies for increasing speed up and decreasing the scale. In this paper it is shown and considered designing reconfigurable blocks by Quantum-dot Cellular Automata and simulating them with QCADesigner tool.

Key words: reconfigurable hardware,QCA,quantum-dot cellular automata, programmable logic.

INTRODUCTION

Using hardwares with ability of producing new logical functions and configuring themselves with the new situation can increase efficiency of the system and decreases the expenses of designing and implementing of different hardwares that each one has its own specific duty.A reconfigurable hardware containing erasable parts which changes itself dynamically. This structure can increase compatibility of hardwares in different operating cases. These hardwares are made up of blocks and interconnecting buses within themselves which is shown in "Fig. 1".These blocks and interconnecting buses are programmable and able to change their logical structure. This capability is embedded in FPGA and FPGA is today's solution for designing these hardwares (Jim Torresen and Knut Arne Vinger, 2002) But FPGA is not efficient for obtaining propagation delay less than 0.001ns and we must replace it with the other newer technology (Scott Hauck and Andr´e DeHon, 2008).

Quantum-Dot Cellular Automata (QCA):

QCA is a bistable quantum cell. We can use each stable case of this structure for logical zero and one. This cell has four quantum dots which place in four angles of a square. Each QCA cell has two free electrons which can move between four quantum dots. Electron move out from one cell to other is not possible. Polarity P is obtained from electronic charge of each quantum dot that discussed in (Lent, C.S. *et al.*, 1994; Tougaw, P.D. and C.S. Lent, 1994). For Coulomb repulsion a polarized QCA can be stable in two polarities. These two stable cases are P=+1 and P=-1 that is shown in "Fig. 2".

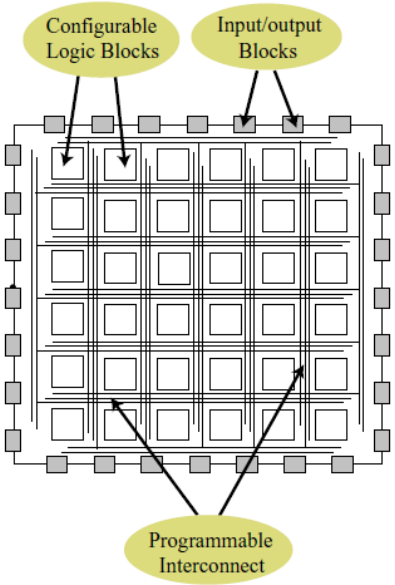

Fig. 1: Schematic of FPGA internal Input/Output and logical blocks and programmable interconnections (Jim Torresen and Knut Arne Vinger, 2002).

Corresponding Author: Homayun Motameni, Department of computer Engineerign, Islamic Azad University, Arak Branch Arak, Iran.

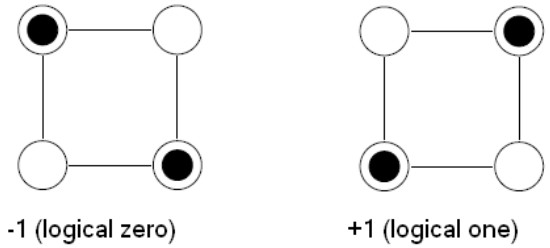

Fig. 2: Two possible charge configurations are used to represent logical zero and one.

Logical information can transfer by interactions among neighbor cells (Tougaw, P.D. and C.S. Lent, 1994; S. Karthigai lakshmi and G. Athisha, 2010). The cell-cell response curve can be computed by solving the two particle Schrodinger equation (Lent, C.S. *et al.*, 1994). It can been seen in "Fig. 3" that the cell-cell response is highly non-linear,which indicates signal restoration. Even a slightly polarized input cell induces an almost fully polarized output cell.

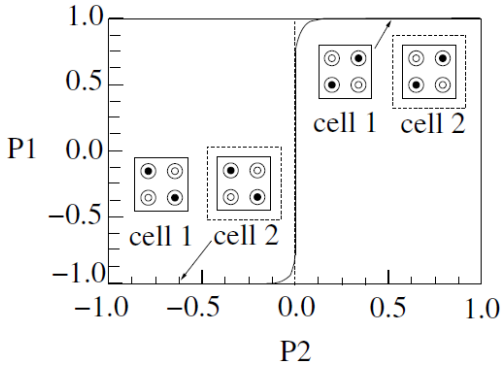

Fig. 3: The cell-cell response curve**.**

If polarity of input is changed, the polarity of closest neighbor will change, too. And the other neighbors follow them as well. "Fig. 4" shows a QCA wire input changes from P=-1 to P=+1, the rest of cells change after one another to P=+1. It is a simple sample of transferring data between linear series of QCA (K. Walus *et al.*, 2003).

Implemented Types of QCA:

Four types of implemented QCAs are:

a) *Metal-dot QCA.*
It was discussed in (I. Amlani *et al.*, 1999).

b) Semiconductor QCA.
It was discussed in (C. Smith *et al.*, 2003; A. Fujiwara *et al.*, 2003).

c) Magnetic QCA.
It was discussed in (R. Cowburn and M. Welland, 2000; G. Bernstein *et al.*, 2005).

d) Molecular QCA.
It was discussed in (H. Qi *et al.*, 2003; Yuhui L.U. *and* Criag S. Lent, 2005).

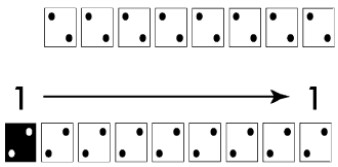

Fig. 4: a QCA wire and its cells interaction.

Designing Logical Gates with QCA:

As we know common logical circuits are designed by using Boolean algebra and we can also use this algebra for designing logical structures with QCA.We can assume P=+1 for logical one and P=-1 for logical zero. One of

the most basic gates in designing logical circuits with QCA is majority gate (Michael T. Niemier and Peter M. Kogge, 1999; Scott Hauck and Andr´e DeHon, 2008). In a 3-input majority gate, output is equal to one(zero), when at least two of three inputs become one(zero). It is shown in "Fig. 5a". A three input majority gate is formulated as Maj(A,B,C)=A.B+B.C+C.A.So in a QCA inverter gate, when input is one(zero) output will be zero(one). It is shown in "Fig. 5b".

Reconfigurable Structures Based on QCA:

In logical structures with QCA there is reconfigurable capability e.g. for a 3-input majority gate with fixing input c with P=-1(logical zero) we can obtain F=a.b and by putting input c with P=1(logical one), F=a+b is resulted (Ismo H. Anninen and Jarmo Takala, 2007). By combining AND and Inverter gates or OR and Inverter gates we can design any logical functions. There are several algorithms to transform conventional Boolean logical functions to majority gate logical functions (R. Zhang *et al.*, 2004; R. Zhang *et al.*, 2005). We can benefit from this privilege in designing bigger reconfigurable structures. For example by connecting three 3-input majority gates (Whitney J. Townsend and Jacob A. Abraham, 2004), which is shown in "Fig. 6" we can design a bigger reconfigurable gate for realization of several logical functions that are illustrated in table I.

Table 1: Control inputs and corresponding output functions.

F (Output Function)	S0	S1	S2
a.b.c.d	0	0	0
(a+b).c.d	1	0	0
a.b.(c+d)	0	1	0
(a+b).(c+d)	1	1	0
a.b+c.d	0	0	1
a+b+c.d	1	0	1
a.b+(c+d)	0	1	1
a+b+c+d	1	1	1

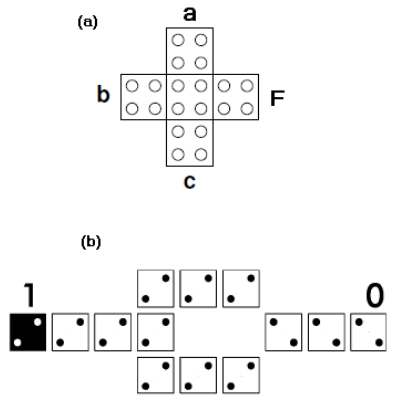

Fig. 5: QCA gates: (a) Majority gate, (b) Inverter gate.

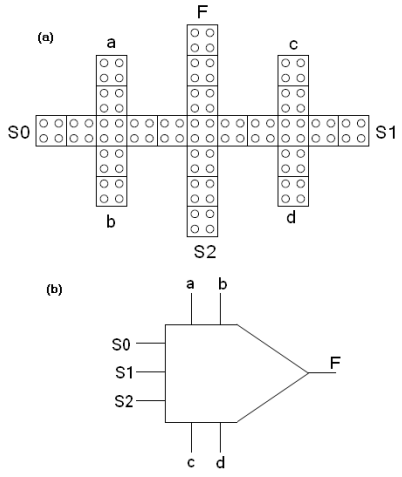

Fig. 6: Complex 4-input gate*:* (a) QCA structure, (b) Schematic diagram.

"Fig. 7" Shows the simulation results for this complex gate when S2=1,S1=0,S0=0 (F= a.b + c.d). We should consider the use of input variable complements in designing logical functions. Hence we should design reconfigurable structures for selecting inputs complements. In "Fig. 8" a sample of this type of circuit is shown and its operation is illustrated in Table II. In this structure with a select line we can decide for inputting primary variable or its complement. This selector is made with five 3-input majority gates and one inverter.It was simulated with QCADesigner and simulation results is shown in "Fig. 9". By combining circuits in "Fig. 6" and "Fig. 8" we can design a new circuit with more flexibility for reconfiguring.

Table 2: Operation of Input Selector.

S	out (output function)
0	A
1	Not A

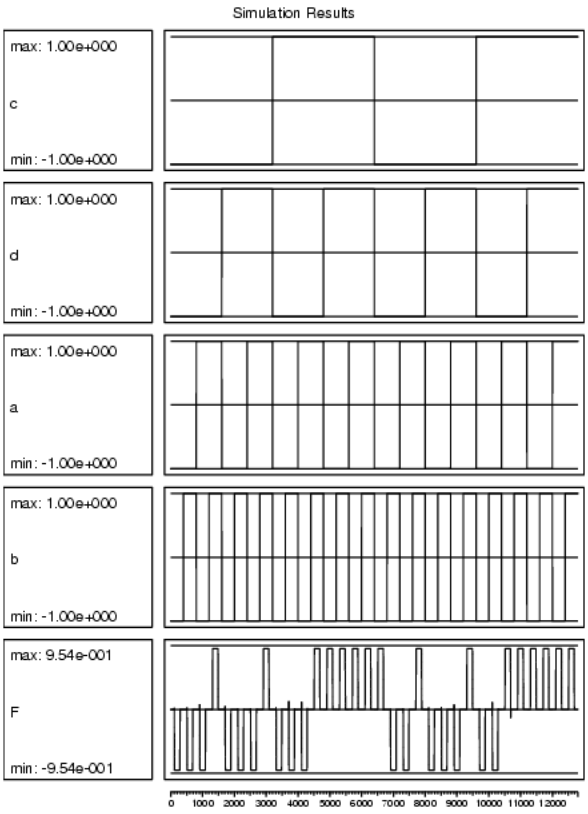

Fig. 7: Simulation results in QCADesigner for complex gate when F=a.b+c.d.

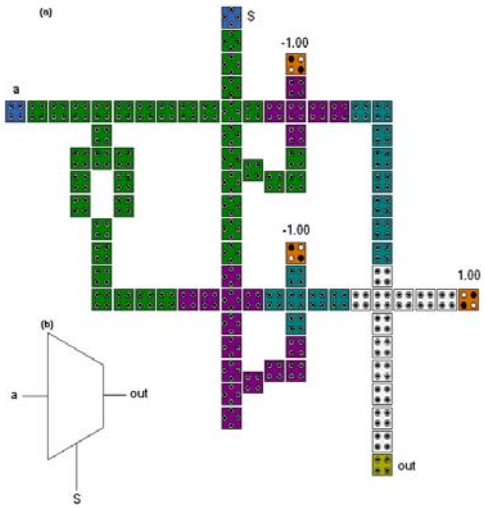

Fig. 8: Input Selector: (a) QCA structure, (b) Schematic diagram.

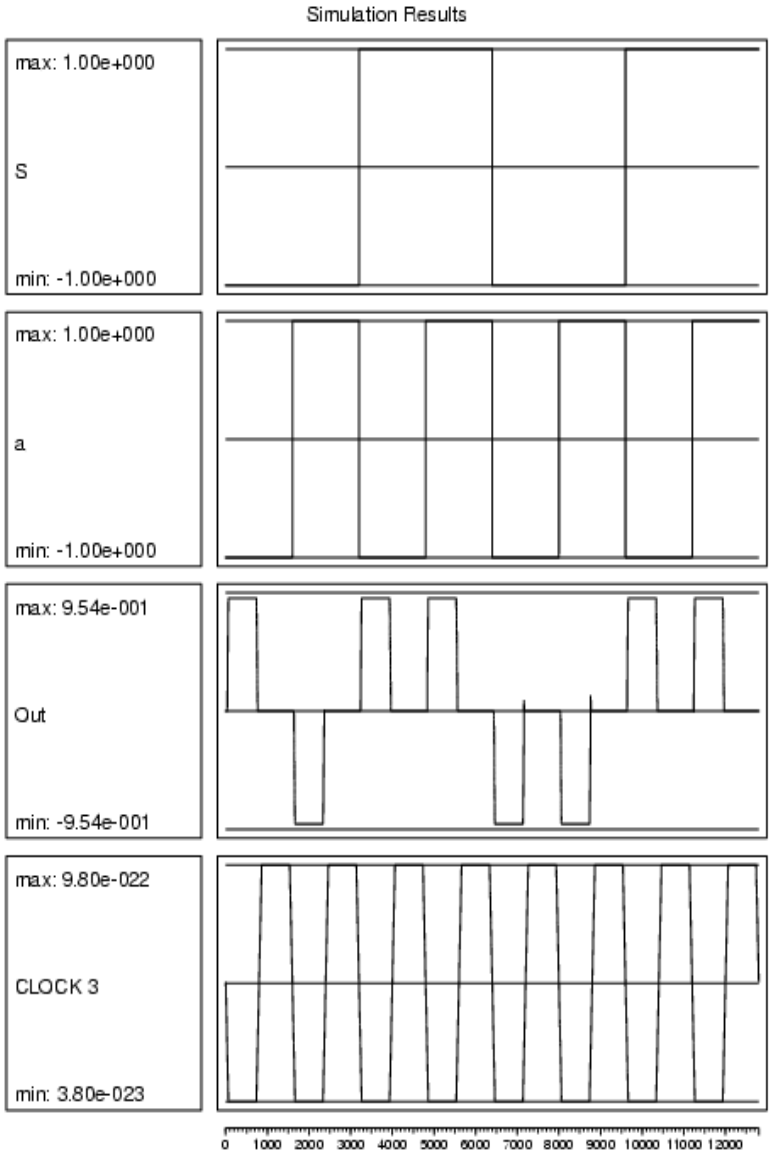

Fig. 9: Simulation Results in QCADesigner for Selector.

e.g. for four inputs such as a,b,c,d we can make a bigger reconfigurable block with 23 three- input- majority gates and four inverter gates which is shown in "Fig. 10".This reconfigurable block has seven s*e*lect lines so we can have a great number of logical functions in output. We can also use this block for making bigger reconfigurable blocks and produce more complex Boolean functions.

Conclusion:

In this paper QCA was considered as a novel structure for designing reconfigurable logics and two types of reconfigurable structures was designed and simulated.A bigger reconfigurable block was designed by combining those small reconfigurable blocks with each other. This reconfigurable block has seven select lines so we can have a great number of logical functions in output.

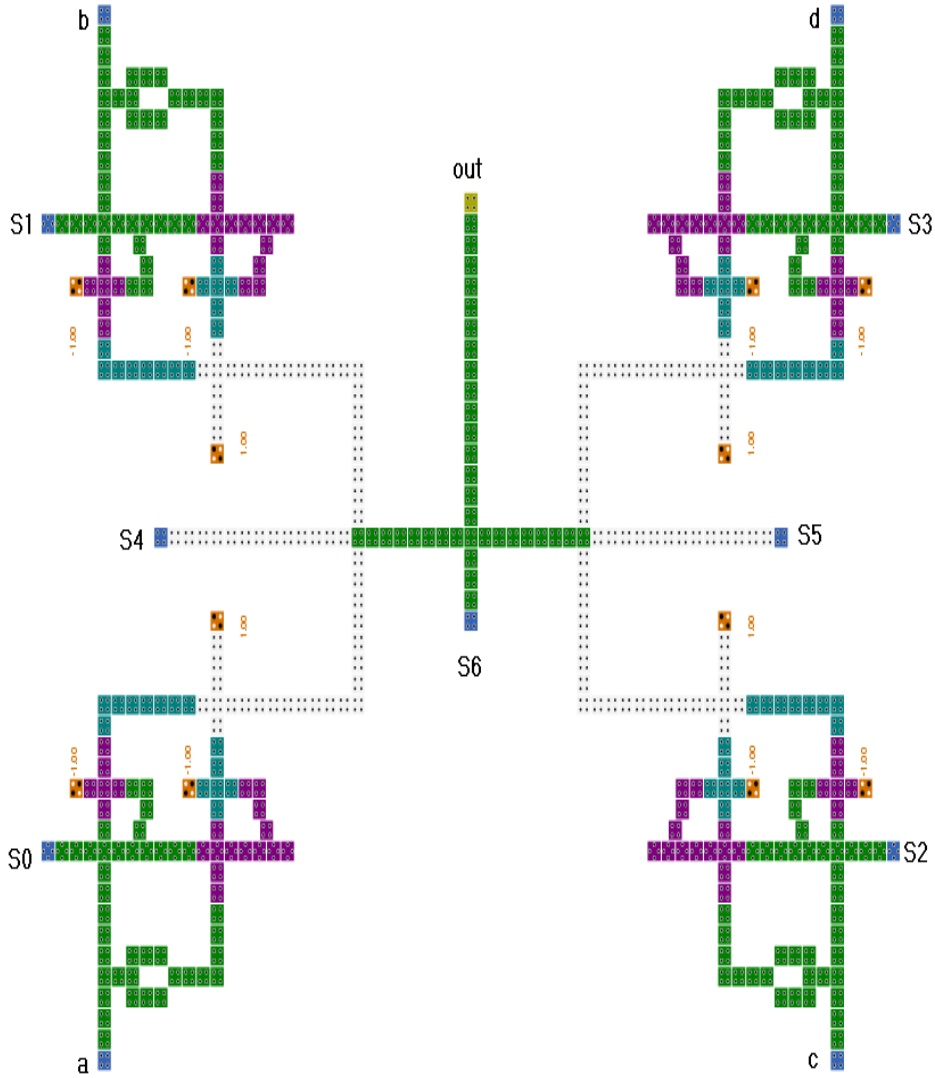

Fig. 10: Reconfigurable block by combining smaller reconfigurable structures.

REFERENCES

Amlani, I. A. Orlov, G. Toth, G. Bernstein, C. Lent and G. Snider, 1999. "Digital logic gate using quantum-dot cellular automata," Science, 284: 289-291.

Bernstein, G., A. Imre, V. Metlushko, A. Orlov, L. Zhou, G.C.L. Ji and W. Porod, 2005. "Magnetic QCA systems," Microelectronics Journal, 36: 619-624.

Cowburn, R. and M. Welland, 2000. "Room temperature magnetic quantum cellular automata," Science, 287(5457): 1466-1468.

Fujiwara, A., S. Horiguchi, M. Nagase and Y. Takahashi, 2003. "Threshold voltage of si single-electron transistor," Japanese Journal of Applied Physics Part 1-Regular Papers Short Notes and Review Papers, 42: 2429-33.

Ismo, H. Anninen and Jarmo Takala, 2007. "Binary Multipliers on Quantum-Dot Cellular Automata," ELEC. ENERG., 20(3): 541-560.

Jim Torresen *and* Knut Arne Vinger, 2002. "High Performance Computing By Context Switching Reconfigurable Logic, "In proc. of 16th European Simulation Multiconference (ESM-2002), pp: 207-210.

Karthigai lakshmi, S. and G. Athisha, 2010. "Efficient Design of Logical Structures and Functions using Nanotechnology Based Quantum Dot Cellular Automata Design," International Journal of Computer Applications (0975-8887), 3(5).

Lent, C.S., P.D. Tougaw and W. Porod, 1994. “Quantum Cellular Automata: The Physics of Computing with Arrays of Quantum Dot Molecules,” Phys. Comp. 94: Proceedings of the Workshop on Physics and Computing, IEEE Computer Society Press, pp: 5-13.

Michael, T. Niemier and Peter M. Kogge, 1999. “Logic in Wire: Using Quantum Dots to Implement a Microprocessor,” GLS '99 Proceedings of the Ninth Great Lakes Symposium on VLSI,IEEE Computer Society Washington, DC, USA.

Qi, H., S. Sharma, Z. Li, G. Snider, A. Orlov, C. Lent and T. Fehlner, 2003. “Molecular quantum cellular automata cells. electric field driven switching of a silicon surface bound array of vertically oriented two-dot molecular quantum cellular automata,” J. Am. Chem. Soc., 125: 15250-15259.

Scott Hauck and Andr´e DeHon, 2008. Reconfigurable computing: the theory and practice of FPGA-based computation, Morgan Kaufmann Publishers.

Smith, C., S. Gardelis, A. Rushforth, R. Crook, J. Cooper, D. Ritchie, E. Linfield, Y. Jin and M. Pepper, 2003. “Realization of quantum-dot cellular automata using semiconductor quantum dots,” Superlattices and Microstructures, 34: 195-203.

Tougaw, P.D. and C.S. Lent, 1994. “Logical Devices Implemented Using Quantum Cellular Automata, “ Journal of Applied Physics, 75(3): 1818-1825.

Walus, K., G.A. Jullien and V.S. Dimitrov, 2003. “Computer Arithmetic Structures for Quantum Cellular Automata,” 37th Asilomar Conf. Signals, Systems and Computers.

Whitney, J. Townsend and Jacob A. Abraham, 2004. “Complex Gate Implementations for Quantum Dot Cellular Automata,” 4th IEEE Conference on Nanotechnology, pp: 625-627.

Yuhui, L.U. and Criag S. Lent, 2005. “Theoretical Study of Molecular Quantum-Dot Cellular Automata,” Journal of Computational Electronics, 4: 115-118.

Zhang, R., P. Gupta and N. Jha, 2005. “Synthesis of majority networks for qca-based logical devices,” International Conference on VLSI Design, pp: 229-234.

Zhang, R., P. Gupta, L. Zhong and N. Jha, 2004. “Synthesis and optimization of threshold logic networks with application to nanotechnologies,” in Design, Automation and Test in Europe Conference and Exhibition, 2: 904-909.