A simple Synthesis Process for Combinational QCA Circuits: QSynthesizer

Vaishali Dhare[1], Usha Mehta[2]
[1, 2] Institute of Technology, Nirma University, Ahmedabad, Gujarat, India
[1]vaishali.dhare@nirmauni.ac.in, [2]usha.mehta@nirmauni.ac.in

Abstract*— Quantum-dot-Cellular Automata (QCA) is coming up as the viable technology which may replace existing CMOS technology. In this context, Electronic Design Automation (EDA) tools for design and fabrication like simulation, synthesis, testing etc. are essential. After design entry and behavioral simulation, the synthesis tool is used to convert the behavioral level description into gate level description. The synthesized circuit can be later on used for functional and timing verification, test development and further layout level synthesis. Even though, few QCA synthesis methods are available in the literature, there is lack of simple QCA synthesis method which uses commonly available CMOS based EDA tools as far as possible and hence, less costly. Therefore, in this paper, a simple but novel synthesis method for combinational QCA circuit "QSynthesizer" is proposed. The authors do neither claim an optimized method of synthesis nor a very high level of research in the field of QCA synthesis tool development. Still, this paper definitely contains a frugal innovation for synthesizing QCA circuit in absence of rarely available, low cost QCA synthesizers. The proposed synthesis method comprises logic reduction algorithm implemented in C++, Perl scripts and widely used synthesis tools like Leonardo Spectrum for digital design. The results on* **Microelectronics Centre of North Carolina** *(MCNC) benchmark circuits using proposed QSynthesizer show the effectiveness of the proposed method.

Keywords—QCA, Synthesis method, Majority Voter, logic Reduction

I. INTRODUCTION

The CMOS technology ruling the semiconductor industry since last five decades but very soon the scaling of the transistor will reach its fundamental limit [1,2]. These limitations steered the development of new computational paradigms and devices [3]. As per ITRS 2015 [4], new device technology is going to be introduced in "Beyond CMOS" era. The QCA [5], Single Electron Transistor [6], Spintronic etc. are new viable candidates of this "Beyond CMOS" era. QCA is an array of cells, each cell carries the binary information and the transfer of information flow is possible by the Coulombic interaction between cells [5].

Generally, CMOS technology based combinational circuit consist of AND, OR, NAND, NOR and INV logic primitives while QCA circuit consists of main logic primitive Majority Voter (MV). Therefore, it is necessary to convert any combinational logic description available either in Boolean equation or data flow form into QCA primitives. In this paper, we have proposed the synthesis method, QSynthesizer which comprises logic reduction algorithm, necessary scripting and widely available synthesis tool for CMOS technology. In a way, this is a frugal innovation to help the researchers of this newly emerging field. MCNC benchmark circuits are synthesized using QSynthesizer and results are compared with the available synthesis method. As the researchers of testing field generally uses the International Symposium for Circuits and Systems (ISCAS) format of the synthesized circuit, the QSynthesizer also converts the synthesized netlist in to the compatible ISCAS format.

The content of the paper is as follows: Section II presents QCA background. Implementation steps and experimental results are presented in Section III. Paper concludes in Section IV.

II. QCA BACKGROUND

A. QCA Cell

The four dots QCA cell with quantum dot's number (sites) is shown in Fig. 1(a). As per the electrons configurations shown in Fig. 1 (b) and (c), cell is encoded as Logic "0" and "1" respectively [5].

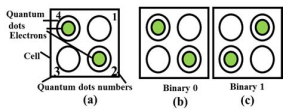

Fig. 1. QCA Cell (a) Schematic (b) with Binary 0 (c) with Binary 1 [5]

B. QCA Logic Primitives

QCA logic primitives, MV and inverter are shown in Fig.2 (a) and (b) respectively [7]. MV is 3-input basic primitive consist of 5 cell configuration, three input cells A, B and C, one middle cell and one output cell. Boolean function of MV is F = AB + BC + AC. 2-input AND and OR logic implementations are possible by keeping one of the inputs of MV at fixed polarization P=-1 i.e. logic "0" and P=+1 i.e. logic "1" respectively. MV as MV_AND and MV_OR gates are shown in Fig. 2(c) and (d) respectively. In this paper, QCA circuit with the basic primitive MV, MV as AND gate (MV_AND) and MV as OR gate (MV_OR) are considered.

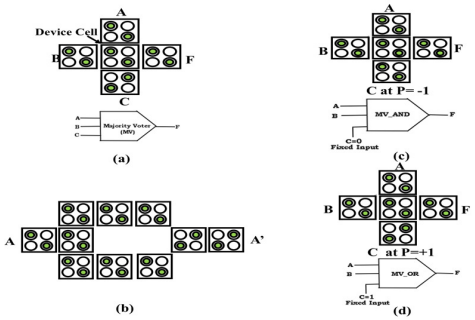

Fig. 2. QCA Basic Primitives (a) Majority Voter (MV) (b) Inverter (c) MV as MV_AND (d) MV as MV_OR

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00105

III. IMPLEMENTATION AND RESULTS

Synthesis methods for digital logics into QCA basic primitives are proposed in [8-13]. The basic goal of the proposed work is:

1. To convert available logically described i.e. Berkeley Logic Interchange Format ("blif") benchmark circuits into synthesized QCA circuit containing QCA primitives i.e. MVs, MV_AND, MV_OR and inverters.
2. To represent the synthesized circuit netlist in the ISCAS (International Symposium on Circuit and System) format.

A. *Implementation*

The proposed synthesis method consists of following steps:

1. Conversion of "blif" to Boolean Equations: Using Open source ABC tool, the netlist file given in BLIF format is converted into ".txt" file containing normal Boolean equation.
2. Logic Reduction: The Boolean equations obtained from the ABC tool are given to the logic reduction algorithm. This logic reduction technique has a major focus on:
   i. The search of implicit MV function in the individual function
   ii. The search of common subfunctions across all the functions of the given circuit.
3. Instantiation of MV, MV_AND and MV_OR in Verilog file: To generate gate level netlist consisting of QCA gates like MV, MV_AND, MV_OR and Inverter.
4. Synthesis using Leonardo Spectrum or Xilinx ISE: The "Verilog" file is synthesised and synthesised gate level netlist and Register Transfer Level (RTL) schematic are obtained.
5. Conversion of gate level netlist to ISCAS format: The synthesised gate level netlist of the given circuit is converted into test generation algorithm's compatible, ISCAS format using Perl script.

B. *Results and discussion*

The logic reduction algorithm is implemented in C++ and Perl scripts are developed for QSynthesizer. All the experiments conducted on the workstation of Core i3 processor with 2.4GHz and 4.00GB RAM. The results are shown on widely used combinational MCNC benchmark circuit set. The comparative analysis in the context of a number of MVs required for the given MCNC benchmark circuits is mentioned in Table I. The number of Primary Inputs (PIs) and Primary Outputs (POs) of benchmark circuit set are mentioned in III and IV columns of Table I respectively. Column V represents the total number of MVs required if the functions are directly implemented in the Sum of Product (SoP) form without any optimization and the number of MVs in case of MALS [9] is mentioned in the VI column. The results of proposed QSynthesizer is given in VII column.

It is observed from Table I that, the number of MVs required to implement given MCNC circuit is less compared to the SoP conversion and available synthesis tool. Also, in most of the cases, the count of MVs is less compared to the MALS synthesis method [9]. The MALS is optimizing individual Boolean Function of the circuit. But when the circuit is having more than one Boolean Functions of some common inputs, there are chances that the part of the functions is being common with other functions. For circuits where the proposed method gives more number of MVs, we could not get the exact reasons of this in the unavailability of synthesized MALS circuits for fair analysis.

TABLE I. EXPERIMENTAL RESULTS OF PROPOSED QSYNTHESIZER

Sr. No.	Circuit	# PI	# PO	# MV		
				AND/OR Gates	MALS [9]	QSynthesizer
1	C17	7	2	6	-	6
2	cm42a	4	10	22	21	18
3	cm82a	5	3	22	16	14
4	cm85a	11	3	44	34	43
5	cm138a	6	8	26	-	16
6	cm150a	21	1	76	46	61
7	cm151a	12	2	36	42	29
8	cm162a	14	5	55	46	36
9	cm163a	16	5	53	42	41
10	cmb	16	4	55	44	47
11	decod	5	16	50	28	26
12	tcon	17	16	40	24	32
13	majority	5	1	17	6	8
15	unreg	36	16	112	84	112
16	pcle	19	9	62	67	55

IV. CONCLUSION

In this paper, the synthesis method for QCA circuits: QSynthesizer is proposed. This synthesis method converter the circuits which are available in "blif" format into synthesized gate level netlist consists of QCA logic primitives Majority Voter (MV), MV_AND, MV_OR and inverters. It is shown that by using existing CMOS synthesis tool, logic reduction algorithm and necessary scripting, a low cost synthesis tool can be developed. This tool may be useful to the researchers with limited resources.

FUTURE WORK

This work can be the base of the further optimized open source synthesis techniques using low cost methods.

REFERENCES

[1] Peercy P.S, "The drive to miniaturization," Nature, vol. 406, 2000, pp. 1023–1026.

[2] Meindl J. D, "Beyond Moore's Law: the interconnect era," Comp. Sci. Eng., vol. 5, 2003, pp. 20–24.

[3] http://www.itrs.net/Links/2013ITRS/Home2013.htm

[4] "International Technology Roadmap for Semiconductors (ITRS)", 2015 Edition, https://www.semiconductors.org/clientuploads/Research_Technology/ITRS/2015/6_2015 ITRS 2.0 Beyond CMOS.pdf

[5] Lent, Craig S., P. Douglas Tougaw, Wolfgang Porod, and Gary H. Bernstein, "Quantum cellular automata," Nanotechnology, vol. 4, no. 1, 1993, pp. 49-57.

[6] K. Likharev, "Single electron devices and their applications," Proc. of IEEE, vol 87, no. 4, 1999, pp. 606–632.

[7] Tougaw, P. Douglas, and Craig S. Lent, "Logical devices implemented using quantum cellular automata," Journal of Applied Physics, vol. 75, no. 3, 1994, pp. 1818-1825.

[8] Zhang, R., et al., "A method of majority logic reduction for quantum cellular automata," IEEE Transactions on Nanotechnology, vol 3 (4), pp. 443-450, 2004.

[9] Zhang, Rui, Pallav Gupta, and Niraj K. Jha, "Majority and minority network synthesis with application to QCA-, SET-, and TPL-based nanotechnologies," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. (26), no. 7, 2007, pp. 1233-1245.

[10] Kong, Kun, Yun Shang, and Ruqian Lu., "An optimized majority logic synthesis methodology for quantum-dot cellular automata," IEEE Transactions on Nanotechnology, vol. 9, no. 2, 2010, pp.170-183.

[11] Wang, Peng, Mohammed Y. Niamat, Srinivasa R. Vemuru, Mansoor Alam, and Taylor Killian, "Synthesis of Majority/Minority Logic Networks," IEEE Transactions on Nanotechnology, vol.14, no. 3, 2015, pp. 473-483.

[12] Devadoss, Rajeswari, Kolin Paul, and M. Balakrishnan, "MajSynth: Ann-input Majority Algebra based Logic Synthesis Tool for Quantum-dot CellularAutomata" 2015.

[13] Mishra, Vipul Kumar, and Himanshu Thapliyal, "Heuristic Based Majority/Minority Logic Synthesis for Emerging Technologies." In VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID), 2017 30th International Conference on, pp. 295-300. IEEE, 2017.