Studies in Computational Intelligence 879

Trailokya Nath Sasamal
Ashutosh Kumar Singh
Anand Mohan

Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective

Springer

Studies in Computational Intelligence

Volume 879

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

The books of this series are submitted to indexing to Web of Science, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink.

More information about this series at http://www.springer.com/series/7092

Trailokya Nath Sasamal · Ashutosh Kumar Singh ·
Anand Mohan

Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective

Springer

Trailokya Nath Sasamal
Department of Electronics
and Communication Engineering
National Institute of Technology
Kurukshetra, Haryana, India

Ashutosh Kumar Singh
Department of Computer Applications
National Institute of Technology
Kurukshetra, Haryana, India

Anand Mohan
Department of Electronics Engineering
Indian Institute of Technology
Varanasi, Uttar Pradesh, India

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-981-15-1822-5 ISBN 978-981-15-1823-2 (eBook)
https://doi.org/10.1007/978-981-15-1823-2

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Dedicated to
My wife: Smruti Rekha, son: Ridhaan,
and Parents.
—Trailokya Nath Sasamal

Anushka, Aakash, Akankshya, and Parents.
—Ashutosh Kumar Singh

My wife: Sudha Mohan, son: Asish Mohan,
daughter: Amrita Mohan, and Late Parents.
—Anand Mohan

Preface

Integrated circuits have become smaller, cheaper, and more reliable and certainly have revolutionized the world of electronics and computer architecture. They are essential components of almost all electronic devices and systems, many of which, such as the Internet, computers, and mobile phones, have become integral parts of modern life and have changed the way we live. The big question for current nanotechnology is, what kind of technology needs to be explored that could replace the conventional CMOS transistors? It is also predicted that CMOS scaling will be ending by 2019. This encourages researchers to come up with some alternative technologies like Quantum-dot Cellular Automata (QCA), carbon nanotube transistors (CNT), silicon nanowires (SiNWs), spin transistors, superconducting electronics, molecular electronics (ME), single electron transistors (SET), resonant tunneling devices (RTD), and tunneling phase logic (TPL). It involves considerable research in new materials, processes, and structures at the nanoscale to continue Moore's law. Each of these paradigms has a number of unique features that make it attractive as a candidate for post-CMOS nanocomputing, and each faces critical challenges to realization. These emerging technologies are not considered as a direct replacement for CMOS technology and may require a completely new architecture to achieve their functionality. *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective* brings all of these issues together in one place for readers and researchers who are interested in this rapidly changing field.

There is another motivation for considering Quantum-dot Cellular Automata, and especially, the design of a QCA circuit is radically different from the conventional digital design due to its unique characteristics at both physical and logic levels. High-level designs focus on logic and algorithmic design in addition to the physical design. In fact, there is a need of research toward both circuit architecture and device design which is required for a profound understanding of QCA nanotechnologies. There are few books available in this area which is based on a specific topic. We provide a composite solution for optimal logic designs for Quantum-dot Cellular Automata-based circuit and reversible logic circuit. The objective behind the proposed design methodologies is to obtain an optimal layout for some of the basic logic circuits considering key metrics such as wire delays, cell counts, and circuit area that help in improving the logic computation and information flow at physical implementation level. The work will help the researchers as well as the manufacturing companies in the development of low-power quantum computers.

This text is intended for senior undergraduate and graduate levels in the areas of nanoelectronics, computer arithmetic, and embedded systems. The book is also suitable for researchers in the areas of emerging nanotechnologies and its architecture, low-power digital design. Knowledge of digital logic system (combinational and sequential) is adequate to follow the material presented in this book.

This book is summarized as follows: First, we examine the QCA implementation of primitive reversible gates. Second, we have presented a compact 5-input majority gate using single-layer QCA technology. We have put forth this gate to design few efficient QCA circuits. Further, we consider a new design for the implementation of 3-input XOR that uses explicit interactions between QCA cells to produce the expected results. In order to show the efficacy of this XOR gate, adder and divider are designed based on it. Next, we discuss the architecture of an efficient 1-bit reversible ALU using an existing reversible gate in QCA. New QCA structures for D flip-flops, shift register, and memory cell are proposed, simulated, and evaluated. Finally, we have discussed briefly different clocking schemes.

Kurukshetra, India
October 2019

Trailokya Nath Sasamal
Ashutosh Kumar Singh
Anand Mohan

Contents

About the Authors

Trailokya Nath Sasamal is currently working as Assistant Professor in the Department of Electronics & Communication Engineering at National Institute of Technology, Kurukshetra, India since August 2013. He has more than 7 years research and teaching experience in various University systems of the India. He has received his Ph.D. degree from the Department of Electronics & Communication Engineering, NIT Kurukshetra, Haryana in 2018. He obtained his M. Tech degree in Electronics Engineering from Indian Institute of Technology, Banaras Hindu University, Varanasi, India. He has presented and published over 40 research papers in reputed journals and various national and international conferences. His research interests include Quantum-dot Cellular Automata, Reversible logic, and new architectures for emerging nano-devices.

Ashutosh Kumar Singh is working as a Professor; Department of Computer Applications; National Institute of Technology; Kurukshetra, India. He has more than 20 years research and teaching experience in various University systems of the India, UK, Australia and Malaysia. Prior to this appointment he has worked as an Associate Professor and Head of Department Electrical and Computer Engineering in School of Engineering Curtin University Australia offshore Campus Malaysia, Sr. Lecturer and Deputy Dean (Research and Graduate Studies) in Faculty of Information Technology, University Tun Abdul Razak Kuala Lumpur Malaysia, Post Doc RA in the Department of Computer Science, University of Bristol, Faculty of Information Science and Technology, Multimedia University Malaysia and Sr. Lecturer in Electronics and Communication Department at NIST, India. Dr Singh has obtained Ph. D. degree in Electronics Engineering from Indian Institute of Technology, BHU, India, Post Doc from Department of Computer Science, University of Bristol, UK and Charted Engineer from UK. He had been entitled for the awards such as Merit Award-03 (Institute of Engineers), Best Poster Presenter-99 in 86th Indian Science Congress held in Chennai, INDIA, Best Paper Presenter of NSC'99 INDIA and Bintulu Development Authority Best Postgraduate Research Paper Award for 2010, 2011, 2012. He is the co-author of six books which includes "Web Spam Detection Application using Neural Network", "Digital

Systems Fundamentals" and "Computer System Organization & Architecture". His research area includes Verification, Synthesis, Design and Testing of Digital Circuits. He has published more than 200 research papers till now in peer reviewed journals, conferences and news magazines and in these areas.

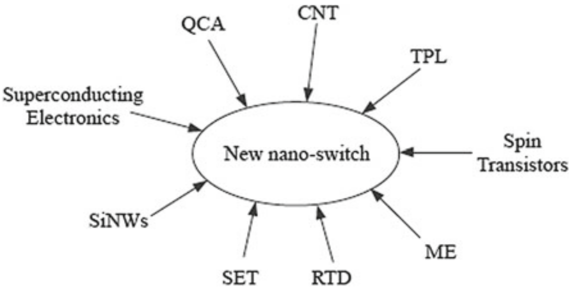
Anand Mohan, former Director of National Institute of Technology (NIT), Kurukshetra, Haryana has 41 years of rich experience in teaching, research, industrial R & D. He is currently working as Professor (HAG) in the Department of Electronics Engineering, IIT(BHU), Varanasi. He has made notable research contributions in the areas of robust watermarking, telemedicine, and fault tolerant digital system design. He has active research collaboration with Louisiana State University, USA and Multimedia University, Malaysia. Eleven students have been awarded Ph.D. degree under his supervision. Prof. Mohan has successfully completed eight sponsored projects funded by MHRD, AICTE and Ministry of Communication & Information Technology, Govt. of India, New Delhi. He has published 160 research papers in reputed international / national journals and conference proceedings and made two chapter contributions in books. Prof. Mohan has provided national level leadership to the defence related R & D activities as Chairman of Armament Sensors & Electronics (ASE) Panel under ARMREB, DRDO, Ministry of Defence, Govt. of India. As part of a doctoral research, he has developed a potential technology for application in 'Neutron Detection Reader' system for defence applications. He is recipient of 'Life Time Achievement Award' conferred by Kamla Nehru Institute of Technology (KNIT), Sultanpur (2016). His co-authored research papers have merited Best Postgraduate Research Paper Award of Bintulu Development Authority, Malaysia (2012), awards of International Union of Radio Science, Belgium (2005), Institution of Engineers, India (2003), and Indian Science Congress (1999). Prof. Mohan is Fellow of Institution of Electronics and Telecommunication Engineers (IETE), Fellow of Institution of Engineers (I), Member, IEEE, USA life member of Project Management Associates (PMA), New Delhi, and Life member of Indian Society of Technical Education (ISTE), New Delhi.

Chapter 1
Introduction

The exponential decrease in feature size causes serious challenges in CMOS technology due to oxide thickness, diffusion barriers, power dissipation, and leakage currents, etc. These challenges have reduced further scaling possibilities of current CMOS devices [1, 2]. The big question for nanotechnology is, what kind of technology needs to be explored that could replace the conventional CMOS transistors? It is also predicted that CMOS scaling will be ended by 2019 [3]. This encourages researchers to come up with some alternative technologies like Quantum-dot Cellular Automata (QCA) [4], carbon nanotube transistors (CNT) [5, 6], silicon nanowires (SiNWs) [7], spin transistors [8], superconducting electronics [9, 10], molecular electronics (ME) [11, 12], single electron transistors (SET) [13, 14], resonant tunneling devices (RTD) [15], and tunneling phase logic (TPL) [16]. It involves considerable research in new materials, processes, and structures at the nanoscale to continue Moore's law [17]. Moreover, the International Technology Roadmap for Semiconductors (ITRS) outlines some of these devices which are depicted in Fig. 1.1 [3].

It is addressed that these devices provide new possibilities for future computing paradigms. Some devices offer high-speed computation with high-density integration, while others promise extremely low-power dissipation. Currently, there is a

Fig. 1.1 Emerging research logic devices



T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_1

need of continuous research for devices that enable small sizes, high packing densities, and low-power dissipation. Most of the above-mentioned emerging technologies have feature sizes close to atomic and molecular scale where quantum effects are dominating, but induce uncertainty in device performance. Quantum-dot Cellular Automata (QCA) is one such promising nanotechnology that offers a new platform for information computation at nanolevel, which is able to provide extremely low-power consumption with small dimension and high-speed operation [18–21]. In the traditional technologies, the device-to-device interaction at nanoscale is one of the major limitations in further scaling of the devices. QCA technology aims to mitigate the unavoidable nanolevel issues, like device-to-device interaction to perform efficient computing. The information is transferred by employing propagation of polarization between two cells due to the Coulombic interaction of electrons. As a result, basic QCA cells are considered for constructing the two major components in QCA circuits, i.e., the computation and communication.

Energy efficiency has also been the most important metric of advancement in modern computer design [22–28]. One of the well-known theories regarding the fundamental energy limits in computation is Landauer's principle. Work by Landauer [29] showed that, regardless of the underlying technology, conventional logic circuits dissipate heat in an order of $k_\mathrm{B}T\ln 2$ joules for every bit of information that is lost, where k_B is the Boltzmann constant, and T is the operating temperature. Since a QCA device does not involve the transfer of electrons, it has the potential advantages for ultra-low-power computing, even below the traditional $k_\mathrm{B}T$ [30]. These attractive attributes enable quantum gates and circuit role toward computational reversibility.

After long discussions and numerical analyses of $k_\mathrm{B}T\ln 2$ energy bound [31–33], some recent experimental demonstrations confirmed its validity [34, 35]. These results indicate that the Landauer bound limits the minimum energy dissipation in modern CMOS-based computers [36–38], which perform irreversible logic operations. Since today's computing devices are usually built of elementary gates like AND, OR, NAND, etc., they are subjected to this principle hence, dissipate this amount of power in each computational step. In order to go beyond this limit, Edward Fredkin established a theory of reversible computing [39], where the entropy of information is conserved during computation to prevent the heat generation resulting from the entropy reduction. Reversible computing offers an alternative, where a logical operation does not yield information loss. According to [40], zero energy dissipation would be possible only if the network consists of reversible gates. Reversible logic realizes $n \times n$ functions where a bijective relation exists between input and output vector. In a reversible logic, every input pattern can be uniquely recovered from its output pattern, so no information is lost during computation. However, the experimental demonstration of reversible logic operations is in the initial stage of its development, and reversible computing will be achievable as practical logic devices in near future. The physical implementations of QCA devices are also limited, and they are not commercially fabricated. But QCA experimental devices using semiconductor, molecular, and magnetic materials have been investigated [41–44].

1.1 Motivation

The design of a QCA circuit is radically different from conventional digital design due to its unique characteristics at both physical and logic level. High-level designs focus on logic and algorithmic design in addition to the physical design. Thus, there is a need of research toward both circuit architecture and device design for a profound understanding of QCA nanotechnologies. In QCA, if the complexity increases, the delay may increase because of the increased cell counts and wire connections. In addition, the wiring channels for the input/output synchronization should be minimized as they add significant contribution to the circuit area. So, for a fast design in QCA, it is necessary to minimize the complexity with new optimized layouts. Hence, several implementations for QCA-based logic circuits have been made aiming to reduce circuit complexity and latency using diverse cell configuration and wire crossing methods. Thus, there is a need of advancement of novel structures which have potential benefits from a circuit design perspective. Although QCA logic components can be designed with QCA gates, extra delays will be introduced, which can lead to incorrect timing relationships. These timing issues present difficulties for interconnection and feedback which can affect the performance of QCA circuits. Therefore, assigning correct and efficient clocking zones to circuits is a major challenge in QCA circuit design.

The novel design paradigms are being proposed to keep up with the ever-growing need for computation power and speed. The information throughput of digital system is $B = n/t$ (maximum number of binary transition per unit time) where each binary transition requires energy E_{b}, and total power dissipation growth is in proportion to the information throughput: $P = B \times E_{\mathrm{b}}$. A bit transition requires energy E_{b} to process it and the 'Shannon-von Neumann-Landauer' (SNL) expression gives the lower limit of E_{b} as $k_{\mathrm{B}}T\ln 2 = 0.017$ eV (i.e., $2{:}72 \times 10^{-21}$ J) at room temperature [45], since the amount of energy dissipated in a system bears a direct relationship to the number of bits erased during computation. This amount may not seem to be significant, but it will become relevant in the future.

Let C is the typical capacitance of a node in a logic circuit, and V is the typical voltage swing between logic levels. This is because voltage-coded logic signals have energy of $E_{\mathrm{sig}} = 1/2(CV)^2$ and this energy gets dissipated whenever the node voltage is changed, it leads to dissipation of this energy and is orders of magnitude higher than the $k_{\mathrm{B}}T\ln 2$ factor as shown in Eq. 1.1.

$$\text{For } C = 5fF;\ \ V = 3v$$

$$E_{\mathrm{sig}} = 1/2(5fF)\left(3v^2\right) = 2 \times 10^{-14}\ \text{J/bit} \tag{1.1}$$

Heat from thermodynamical and logical entropy associated with a bit $E_{\mathrm{b}} = 2.72 \times 10^{-21}$ J. For a processor of 2×10^7 transistor dissipates heat at a rate of the processor frequency, for instance, 2 GHz, maximum electrostatic erasure energy (total energy lost as heat) $1/2(CV^2 \times 4{\times}10^7 \times 10^9/\mathrm{s} = 80$ W (All transistor switch at each cycle, i.e., worst case). For a computer operating at SNL limit at 300 K, the minimum size and switching time of binary switches can be estimated based on the Heisenberg uncertainty relations [45]. Minimum size of element comes out 1.5 nm, maximum integration density of elements $n_{\max} = 4.7 \times 10^{13}$ devices/cm^2, minimum switching time $t_{\min} = 0.04$ ps, and power dissipation per unit area $P = (n_{\text{max}} \times E_{\text{b}})/t_{\text{min}} = 3.7 \times 10^6\ \text{W/cm}^2$.

Above calculation shows that today's technology is still a factor of 1000 away from the Landauer limit [34], i.e., further performance improvements can be done up to this factor. Although the theoretical lower bound on power dissipation still does not constitute a significant fraction of the power consumption of current devices, it cannot be neglected [46]. So, one of the important aspects for adopting reversible logic is that it can offer a logic design methodology for designing ultra-low-power circuits beyond $k_{\text{B}}T\ln 2$ limit for those emerging nanotechnologies where the heat generated due to information loss will be a major factor of the overall heat dissipation. The synthesis of reversible logic has a close relation with the quantum logic synthesis, and the synthesis methods of reversible logic can be used to implement the quantum logic synthesis [47]. Thus, the study of reversible logic synthesis will contribute to the progresses of the ultra-low-power IC's design and the quantum computing. Quantum computers are realized in critical fields such as biotechnology, nanomedicine, and secure computing. Thus, the feasibility of reversible logic circuits could critically impact the realization of quantum computing. Synthesis of reversible logic circuits differs from the conventional one in many ways. Firstly, in reversible circuit there should be no fan-out, that is, each output will be used only once. Secondly, for each input pattern there should be a unique output pattern. Finally, the resulting circuit must be acyclic. Any reversible gate performs the permutation of its input patterns only and realizes the functions that are reversible. Traditional Boolean logic synthesis approaches like Karnaugh Map, Quine–McCluskey, etc. cannot be directly applied to synthesize a reversible logic circuits because of constraints like fan-outs are not allowed, and feedback is not permitted from gate outputs to inputs, equal numbers of I/O, presence of ancilla and garbage output, etc. So implementation only could be possible in the form of cascading of reversible gates. So, there is a need of research toward the design of reversible logic circuits, and it is important to minimize parameters such as ancilla and garbage bits, quantum cost, and delay in the design of reversible circuits. Recently, both reversible and fault tolerant helped to broaden the horizon of fields like low-power design, quantum computation, nanotechnology, DNA computing, optical computing cryptography, and informatics [48–52]. Thus, the reversible fault tolerant circuit is the most suitable and promising candidate with respect to conventional technologies. Numerous works have been published, showcasing advantages of fault tolerant reversible logic circuits. There are many parity preserving reversible gates proposed in the literature. Among them Double Feynman gate (F2G) depicted in [53] and Fredkin gate (FRG) depicted in [39]. Meanwhile, reversible gates such as TSG (Thapliyal Srinivas gate), MKG (Majid Keivan gate), HNG (Haghparast-Navi gate), PFAG (Peres Full Adder gate), and IG (Islam gate) have been proposed by researchers, but only few of them are having parity preserving feature. Thus, there is a need of research toward the design and analysis of reversible logic circuits using new reversible gates in which multiple parameters can be optimized.

1.2 Contributions of the Book

Design of digital circuits in QCA is in a growing stage. We examine design of different modules that will eventually become parts of QCA-based complex circuits. We focus (i) to design reversible logic circuits considering metrics of ancilla inputs, garbage outputs, gate counts; (ii) to devolve new design structures in order to obtain an optimal layout for some of the basic logic circuits considering key metrics such as wire delays, cell counts, and circuit area based on emerging nanotechnology such as QCA. As a developing technology, there are a number of research areas in QCA. This book has the following contributions toward the design and synthesis of some combinational/sequential reversible and non-reversible logic circuits using QCA.

- An introduction to different aspect of both irreversible and reversible logic gates along with the performance metrics which are used for evaluation.
- Design of fundamental reversible logic gates in QCA is provided such as Fredkin, Peres, and Toffoli. This will help beginners to explore QCA-based circuits with different aspects such as layout design, clocking, and crossover.
- Presents an optimal 5-input majority gate using single-layer QCA technology. Novel designs of QCA-based D flip-flops, shift register, and a new robust RAM cell with set and reset ability based on the gate have been introduced, which are more efficient compared to the existing designs. Novel architectures of adder circuits are developed to evaluate the suitability of the presented gate.
- Introduces an efficient one-bit reversible ALU using an existing reversible gate in QCA which utilizes a minimum number of QCA cells and clock delay. The design and verification of the QCA layouts are performed using the QCADesigner tool.
- Highlight a new design for implementation of 3-input XOR that uses explicit interactions between QCA cells to produce the expected results ignoring the conventional designing methods. To show the efficacy of the novel XOR gate, adder and divider are constructed based on it. The results confirmed that the presented structures have outperformed all prior designs in terms of complexity, area occupation, and input-to-output clock delay as compared to most of the coplanar designs.
- Investigates different clocking floor plan and clocking schemes.
- Tutorials on layout design using QCADesigner (freeware) and power estimation tool are also provided in Appendices.

A large part of the contents depicted in this book are also equally applicable to other technologies (like resonant tunneling diodes, single electron transistors, etc.) that utilize majority logic in other form.

1.3 Organization of the Book

The remainder of this dissertation is organized as follows: Chap. 2 provides an overview of QCA technology and the logic associated with it. It also includes a brief discussion of various types of QCA implementation, currently under research. Chapter 2 also describes the background and a comprehensive literature survey pertaining to this research. Chapter 3 discusses different aspect of both irreversible and reversible logic gates along with the performance metrics which are used for evaluation. In Chap. 4, we explore the QCA implementation of primitive reversible logic gates. In particular, new reversible structures are proposed and compared with existing design in this chapter. In Chap. 5, we discuss two different designs of QCA Ripple Carry Adders by customizing the fundamental block, i.e., the full adder circuit. We start the discussion with designing RCA using 5-input majority gate-based full adder. Further, designs are evolved considering a compact form full adder that relied on a novel XOR structure. Chapter 6 is devoted to design and implementation of iterative computational unit such as binary divider using QCA. Different types of dividers are introduced. Specifically, non-restoring divider is discussed that realized by iterative cellular arrays for parallel divisions. In Chap. 7, we examined reversible arithmetic logic unit (ALU) and its implementation in QCA framework. In particular, we analyze and validate one of the reversible ALU designs. In Chap. 8, we presented design of different D flip-flops and RAM cell with set and reset ability in QCA. Chapter 9 presents an overview of different clocking schemes. This study can help to develop new routing algorithms and design of an arbitrary large feasible QCA circuit toward advancement of QCA technology. The concluding remarks and the suggested future direction in terms of extensions to the problems are addressed in this book, and other ideas for further refinements are given in Chap. 10. Quick tutorials on CAD tool QCADesigner and QCAPro are provided in Appendices at the end.

References

1. J. Fortes, Future challenges in VLSI system design, in *IEEE Computer Society Annual Symposium on VLSI, 2003. Proceedings*. IEEE Computer Society (2003), pp. 5–7
2. T. Skotnicki, J. Hutchby, T. King, H.S. Wong, F. Boeuf, The end of CMOS scaling. IEEE Circuits Dev. Mag. **21**(1), 16–26 (2005)
3. The International Technology Roadmap for Semiconductors: Edition-assembly & packaging, Semiconductor Industry Association (San Jose, CA, 2005), https://www.itrs.net/links/2005ITRS/AP2005.pdf
4. C.S. Lent, P.D. Tougaw, W. Porod, G.H. Bernstein, Quantum cellular automata. Nanotechnology **4**(1), 49–57 (1993)
5. S.J. Tans, A.R.M. Verschueren, C. Dekker, Room-temperature transistor based on a single carbon nanotube. Nature **393**(6680), 49–52 (1998)
6. A. Bachtold, P. Hadley, T. Nakanishi, C. Dekker, Logic circuits with carbon nanotube transistors. Science **294**(5545), 1317–1320 (2001)

7. Y. Cui, C.M. Lieber, Functional nanoscale electronic devices assembled using silicon nanowire building blocks. Science **291**(5505), 851–853 (2001)
8. M.E. Flatte, G. Vignale, Unipolar spin diodes and transistors. Appl. Phys. Lett. **78**(9), 1273–1275 (2001)
9. E. Terzioglu, M. Beasley, Complementary Josephson junction devices and circuits: a possible new approach to superconducting electronics. IEEE Trans. Appl. Superconduct. **8**(2), 48–53 (1998)
10. J.C. Gallop, *SQUIDS, the Josephson Effects and Superconducting Electronics* (Adam Hilger, 1991)
11. J.M. Tour, Molecular electronics. Synth. Test. Compon. Acc. Chem. Res. **33**(11), 791–804 (2000)
12. F. Carter, *Molecular Electronic Devices* (Marcel Dekker, New York, 1982)
13. M.A. Kastner, The single-electron transistor. Rev. Mod. Phys. **64**(3), 849–858 (1992)
14. H.W.C. Postma, T. Teepen, Z. Yao, M. Grifoni, C. Dekker, Carbon nanotube single-electron transistors at room temperature. Science **293**(5527), 76–79 (2001)
15. P. Mazumder, S. Kulkarni, M. Bhattacharya, J.P. Sun, G.I. Haddad, Digital circuit applications of resonant tunneling devices. Proc. IEEE **86**(4), 664–686 (1998)
16. T. Yang, R.A. Kiehl, L.O. Chua, Tunneling phase logic cellular nonlinear networks. I. J. Bifurc. Chaos **11**, 2895–2912 (2001)
17. G. Moore, Cramming more components onto integrated circuits. Electronics **38**(8), 114–117 (1965)
18. P.D. Tougaw, C.S. Lent, Logical devices implemented using quantum cellular automata. J. Appl. Phys. **75**(3), 1818–1825 (1994)
19. C.S. Lent, P.D. Tougaw, W. Porod, Quantum cellular automata: the physics of computing with arrays of quantum dot molecules, in *Workshop on Physics and Computation, 1994. PhysComp '94, Proceedings*, Nov 1994, pp. 5–13
20. S. Kassa, R.K. Nagaria, An innovative low power full adder design in nano technology based quantum dot cellular automata. J. Low 156 Power Electron. **12**(2), 107–111 (2016)
21. G. Singh, R.K. Sarin, B. Raj, A novel robust exclusive-or function implementation in qca nanotechnology with energy dissipation analysis. J. Comput. Electron. **15**(2), 455–465 (2016)
22. P. Ball, Computer engineering: feeling the heat. Nature **492**, 174–176 (2012)
23. R.F. Service, Computer science. What it'll take to go exascale. Science **335**(6067), 394–396 (2012)
24. L.F. Rahman, F.B. Arith, M.I.B. Idris, M.B.I. Reaz, Low power in nano-scale CMOS memory. J. Theor. Appl. Inf. Technol. **20**(612) (2014)
25. R. Kusuma, A. Choubey, G. Sreekanth, Reduction of leakage power in CMOS MUX using leakage control transistor in 90 nm technology. Int. J. VLSI Embed. Syst. IJVES **04**(06108) (2013)
26. R. Sharma, R.K. Baghel, L. Gupta, Novel approach for current application with low power dissipation, Int. J. Eng. **1**(5) (2012)
27. Y. Dhillon, A. Diril, A. Chatterjee, A. Singh, Sizing CMOS circuits for increased transient error tolerance, in *Proceedings. 10th IEEE International On-Line Testing Symposium*. IEEE Computer Society (2004), pp. 11–16
28. S. Komatsu, M. Fujita, *Low Power and Fault Tolerant Encoding Methods for On-chip Data Transfer in Practical Applications*, vol. E88-A (Institute of Electronics, Information and Communication Engineers, 1992)
29. R. Landauer, Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **5**(3), 183–191 (1961)
30. R.W. Keyes, R. Landauer, Minimal energy dissipation in logic. IBM J. Res. Dev. **14**(2), 152–157 (1970)
31. K. Shizume, Heat generation required by information erasure. Phys. Rev. E **52**, 3495–3499 (1995)
32. R. Dillenschneider, E. Lutz, Memory erasure in small systems. Phys. Rev. Lett. **102**, 210601 (2009)

33. T. Sagawa, M. Ueda, Minimal energy cost of thermodynamic information processing: measurement and information erasure. Phys. Rev. Lett. **102** (2009)
34. A. Bérut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, E. Lutz, Experimental verification of Landauer's principle linking information and thermodynamics. Nature **483**(7388), 187–189 (2012)
35. A.O. Orlov, C.S. Lent, C.C. Thorpe, G.P. Boechler, G.L. Snider, Experimental test of Landauer's principle at the sub-KBT level. Jpn. J. Appl. Phys. **51**(6S), 06FE10 (2012)
36. R.W. Keyes, Fundamental limits in digital information processing. Proc. IEEE **69**(2), 267–278 (1981)
37. S. Lloyd, Ultimate physical limits to computation. Nature **406** (2000)
38. J.D. Meindl, J.A. Davis, The fundamental limit on binary switching energy for terascale integration (TSI). IEEE J. Solid-State Circuits **35**(10), 1515–1516 (2000)
39. E. Fredkin, T. Toffoli, Conservative logic. Int. J. Theor. Phys. **21**(3), 219–253 (1982)
40. C.H. Bennett, Logical reversibility of computation. IBM J. Res. Dev. **17**(6), 525–532 (1973)
41. A.O. Orlov, I. Amlani, G. Toth, C.S. Lent, G.H. Bernstein, G.L. Snider, Experimental demonstration of a binary wire for quantum-dot cellular automata. Appl. Phys. Lett. **74**(19), 2875–2877 (1999)
42. R.P. Cowburn, M.E. Welland, Room temperature magnetic quantum cellular automata. Science **287**(5457), 1466–1468 (2000)
43. C.S. Lent, B. Isaksen, M. Lieberman, Molecular quantum-dot cellular automata. J. Am. Chem. Soc. **125**(4), 1056–1063 (2003)
44. A. Imre, G. Csaba, L. Ji, A. Orlov, G.H. Bernstein, W. Porod, Majority logic gate for magnetic quantum-dot cellular automata. Science **311**(5758), 205–208 (2006)
45. V.V. Zhirnov, R.K. Cavin, J.A. Hutchby, G.I. Bouriano, Limits to binary logic switch scaling—a gedanken model. Proc. IEEE **91**(11), 1934–1939 (2003)
46. N. Gershenfeld, Signal entropy and the thermodynamics of computation. IBM Syst. J. **35**, 577–586 (1996)
47. D. Maslov, Reversibility in the future computer chip design. CompuTerra **14**(538), 48–51 (2004)
48. I.L.C.M.A. Nielsen, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000)
49. R.C. Merkle, Two types of mechanical reversible logic. Nanotechnology **4**(2), 114 (1993)
50. E. Knill, R. Laamme, G.J. Milburn, A scheme for efficient quantum computation with linear optics. Nature **409**(6816), 46–52 (2001)
51. B. Parhami, Fault-tolerant reversible circuits, in *40th Asilomar Conference Signals Systems, and Computers* (2006), pp. 1726–1729
52. D.K. Pradhan, *Fault-tolerant Computer System Design*. Prentice-Hall (1996)
53. R.P. Feynman, Quantum mechanical computers. Optics News **11**(2), 11–20 (1985)

Chapter 2
QCA Background

2.1 Introduction

This chapter provides an overview of QCA technology and the logic associated with it. It also includes a brief discussion of various types of QCA implementation currently under research along with a comprehensive literature survey pertaining to this work. As device feature sizes approach quantum limits, fundamental effects will make further scaling difficult, requiring a departure from the FET-based paradigm and necessitating revolutionary approaches to computing. CMOS devices are being reached their fundamental limits. Beyond this limits, the feature size cannot be reduced without compromising with proper functioning of the device. This limitation may pave the way for new alternative device technologies like Quantum-dot Cellular Automata (QCA), Tunneling Phase Logic (TPL), Single-Electron Tunneling (SET), and Carbon Nanotube (CNT). QCA could be a feasible alternative which promises operation at high frequency with low power consumption and high-density device. QCA technology offers a new horizon in information computation, which is based on the confinement and mutual repulsion of electrons. Unlike the traditional transistor-based circuits, a change in logic value from 1 to 0 does not yield discharging of the capacitor and it does not have to dissipate all its energy during transition. The information is transferred as a result of the propagation of polarization between two cells, due to the Coulombic interaction of electrons and no flow of current exist. An individual QCA cell holds a limited number of charges. These charges are permitted only in certain predefined locations within the cell while tunneling is allowed between these locations. QCA works on the mechanism where Coulomb effects dominate over tunneling. And due to Coulomb effects, this cell interacts with its neighbors.

The concept of Quantum-dot Cellular Automata (QCA) was introduced by Tougaw and Lent [1] in 1993. They proposed this new technique to be an alternative method for fabricating electronic devices. The concept was very theoretical in the beginning. Many mathematical equations were used to prove its feasibility

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_2

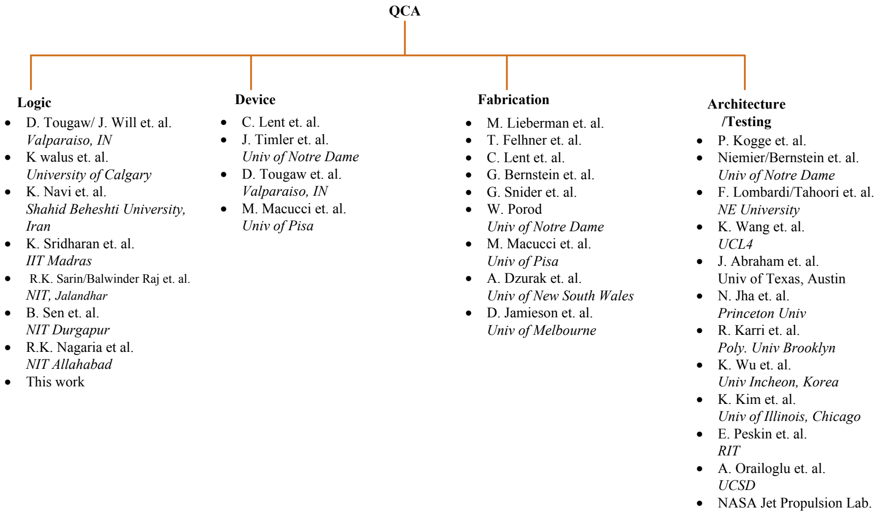


Fig. 2.1 Research groups

and strengths. In the following decade, this topic drew more attention and improved in both design and fabrication. Mainly, there are two categories of current QCA research, physical implementation and nano-architecture. This thesis is focused on the nano-architecture. The logic design is independent of the implementation technique. Once the implementation becomes more stable for fabrication, the design can be set into a real hardware device. There are a number of research groups in leading research laboratories around the world working on QCA. Some of the leading research groups currently involved in different areas of QCA research are depicted in Fig. 2.1.

2.2 QCA Basics

One of the proposed implementations of the Quantum Cellular Automata is the Quantum-dot Cellular Automata. Quantum-dot Cellular Automata is not a physical implementation yet, it is rather a lower-level abstraction, since there are several ways to build the quantum dots and connect them. Quantum dots can be any charge containers, with discrete electrical energy states (there may be more than two states, but only two are used), sometimes called artificial atoms. Some molecules have well-defined energy states and, therefore, are suitable for supporting the operation of QCA systems. Small metal pieces can also behave as quantum dots, if the energy states an electron can occupy are distinguishable, instead of the usual energy band. This means that the difference between two consecutive energy states must be well above the thermal noise energy ($k_\mathrm{b}T$, being k_b the Boltzmann constant and T the absolute temperature).

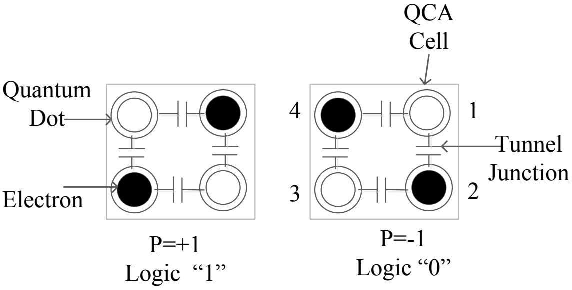
The basic cells are made of four dots placed in the corners of a square, populated with two excess electrons. These dots are well known as quantum dots or qdots. The charge of the electron is localized by the quantum dot. The dot is basically a region of space with energy barriers surrounding it. These barriers are large and high enough so that the charge within it is quantized to a multiple elementary charge. Given the electrostatic interactions (repulsion) between the charges, these will tend to occupy diagonally opposed quantum dots. There are only two stable configurations, as there are only two diagonals in a square, and these two stable configurations are the two lower energy states referred above: they encode the binary values ‘0’ and ‘1’. In the absence of any environmental conditions, the two configurations have the same electrostatic energy. The state of the neighboring cells makes one of the configurations to be the preferred low-energy configurations. The bi-stability of QCA is based on the quantization of charge and it is essential to identify the relationship between the energy levels of a single particle and the energy levels of the dot.

According to the existing Coulombic interaction between the electronic charges, they can occupy diagonal antipodal sites through tunneling junctions. The numbering of the dots in the cell goes clockwise starting from the dot on the top right. A polarization P in a cell, which measures the extent to which the electronic charge is distributed among the four dots, is therefore defined as:

$$p = \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{\rho_1 + \rho_2 + \rho_3 + \rho_4} \tag{2.1}$$

where ρ_i is the electronic charge in each dot of a four-dot QCA cell. Once polarized, a QCA cell can be in any one of the two possible states depending on the polarization of charges in the cell. Because of coulombic repulsion, the two most likely polarization states of QCA can be denoted as $P = +1$ and $P = -1$ as shown in Fig. 2.2. The two states depicted here are called ‘most likely’ and not the only two polarization states are because of the small (almost negligible) likelihood of existence of an erroneous state. In QCA architecture, information is transferred between neighboring cells by mutual interaction from cell to cell. Hence, if we change the polarization of the driver cell (left most cell also know as input cell), first its nearest neighbor changes its polarization, then the next neighbor and so on. Figure 2.2 depicts the transfer of polarization between neighboring QCA cells. When the driver cell (input)

Fig. 2.2 Two possible polarizations of a QCA cell



is $P = -1$ (or $P = +1$), a linear transfer of information among its neighboring cells leads to all of them being polarized to $P = -1$ (or $P = +1$).

2.3 Mechanics of QCA Device Operation

To figure out the operation of a simple QCA cell, we first examine the motion of an electron in an infinite potential well. The walls of infinite potential well hinder electron to tunnel between adjacent dots. Electrons in an infinite potential well exist as a wavefunction $\Psi(x, y, z)$ that gives us the probability of finding an electron within that potential well. This probability is proportional to $|\Psi(x, y, z)|^2$. Solution to the Schrodinger's wave equation for a free electron ($V = 0$) is given by:

$$\frac{\mathrm{d}^2\psi}{\mathrm{d}x} + \frac{2m}{\hbar}(E - V)\psi = 0 \tag{2.2}$$

where V is the potential acting on the particle, E is the energy of the particle, and m is the mass. Taking $V = 0$ for free electron, we obtain:

$$\frac{\mathrm{d}^2\psi}{\mathrm{d}x} + \frac{2m}{\hbar}(E)\psi = 0 \tag{2.3}$$

Using $k^2 = 2\ m/h^2$, this reduces to

$$\frac{\mathrm{d}^2\psi}{\mathrm{d}x} + k^2\psi = 0 \tag{2.4}$$

Solution of Schrodinger's equation for this wavefunction is a sin/cos function, and it also gives the value of the energy of an electron within a potential well. The electron can only have certain discrete energies (E_n) matching the allowed wavefunctions. A lower (higher) energy electron will have a smaller (larger) value of k (wavevector) and a larger (smaller) wavelength (see Fig. 2.3). Since the boundary conditions demand the wavefunction to be zero at the walls of the well, the wavevector can only take discrete quantities and hence the electron can only exist in quantized energy levels. The spacing between adjacent energy levels depends on the width of the potential well. If we consider the height of the potential well as finite, there is a possibility of electrons tunneling out of the potential well. Figure 2.4 shows an example of an electron tunneling across a finite potential well. The potential energy (PE) of point A is less than that of point D. Hence, a car released from point A can at most make it to C but not E. When the car is at the bottom of the hill, its energy is totally kinetic energy (KE). The energy barrier (between C and D) prevents the car from making it to E. In quantum theory, on the other hand, there is a chance that the car could tunnel through (leak) the energy barrier between C and E and emerge on the other side of the hill at E. Figure 2.4 shows the wavefunction of the electron when it is

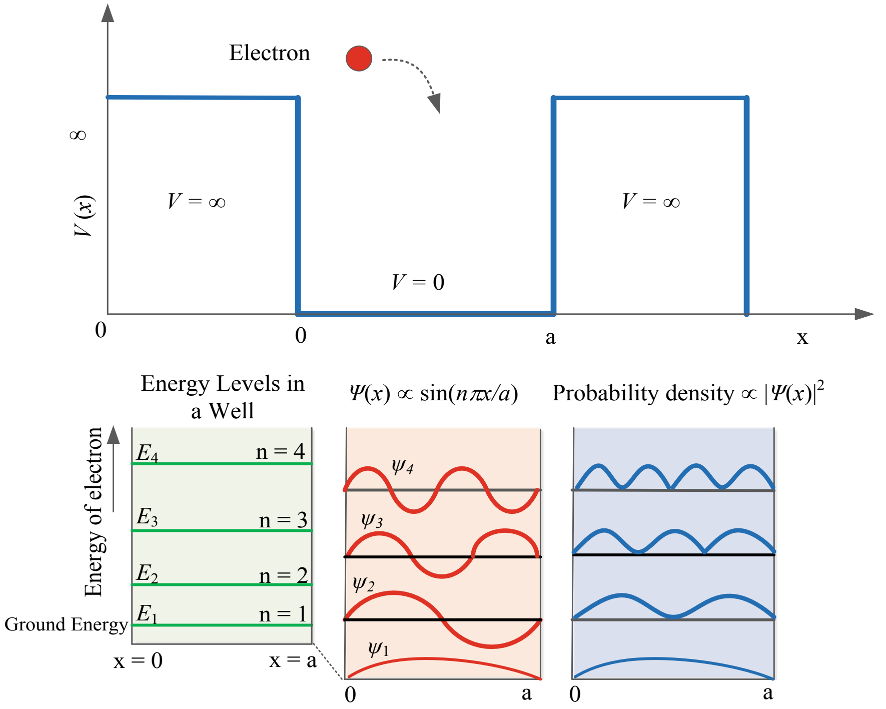


Fig. 2.3 Various quantized energy states of an electron in a one-dimensional infinite potential well. Possible wavefunctions and the probability distributions for the electron are shown [2]

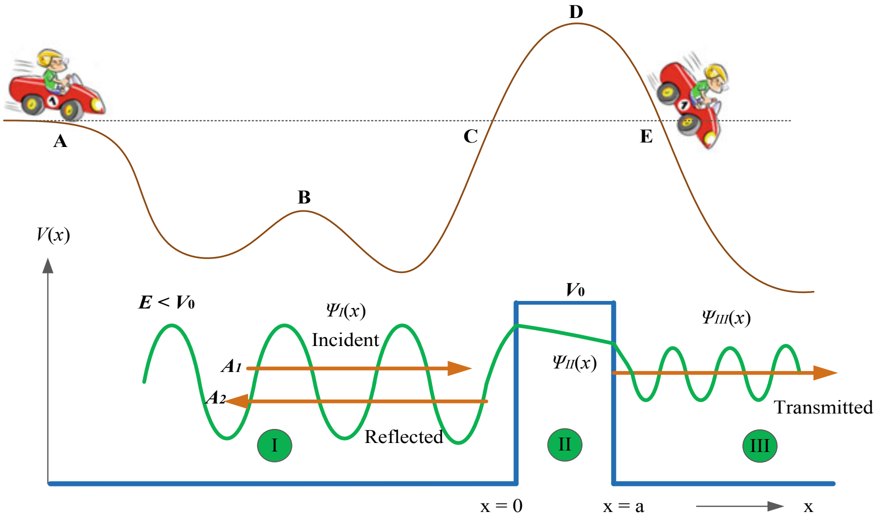


Fig. 2.4 Roller coaster example for tunneling phenomenon across a finite potential wall [2]

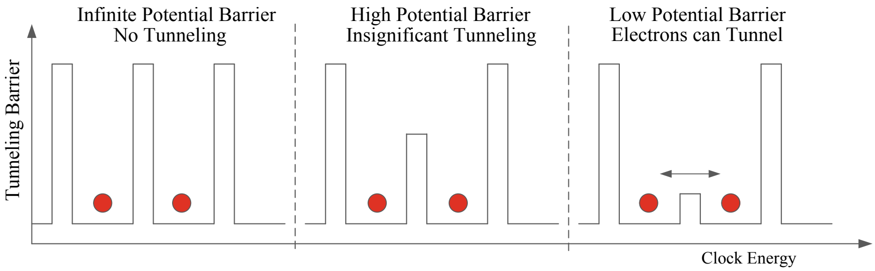


Fig. 2.5 Clock energy variation to control the tunneling barrier. While the clock energy given to a QCA cell increases, the tunneling barriers lower and allow the electron to tunnel across to the other side

incident on a PE barrier (*Vo*). The interference of the incident and reflected waves give $y_I(x)$. There is no reflected wave in region III. In region II, the wavefunction decays with x because $E < Vo$. Solving the Schrodinger equation for the finite barrier region (II) yields an exponential decay function. This is the main difference to the outer regions of the infinite well, where the wavefunction must be zero. Solutions for I and III are the same as for the infinite potential well. However, boundary conditions now demand that the wavefunction matches the exponential function in region II, causing nonzero amplitude in region III. Since the probability of finding an electron is proportional to the square of the amplitude, therefore, there is a nonzero probability to find the electron on the outside, i.e., it can escape from region I. Taking this into account we now look at a simple QCA cell with two electrons placed in neighboring potential wells (called dots). In case of an infinite potential barrier between the dots, electrons are not allowed to tunnel within the dots. As the potential barrier decreases, the possibility of an electron to tunnel across the potential barrier increases. When the potential barriers are very low, electrons can tunnel freely across the two quantum dots. In QCA technology, clock energy is provided as a means to lower or raise the tunneling barriers as we will see in Sect. 2.6. Figure 2.5 shows how the tunneling barriers between two dots are lowered (raised) when the clock energy supplied to the QCA cell is raised (lowered). The work done in raising and lowering of tunneling barriers controlled by the clock energy can be termed as leakage power dissipation as this will take place even if the QCA cell does not switch state. In a similar way, a clock controls the tunneling barriers in a four-dot QCA cell used in this work.

Since in practice it is not possible to implement an infinite potential well to prevent the electrons from tunneling across, there is always a finite possibility of some electronic charge escaping the QCA cell over a long period of time. However, in this work, we have neglected any loss of charge. Electrons in higher energy states within a potential well are more prone to tunnel across if the tunneling potential is of finite height. Thermal errors are caused when the electrons settle in higher energy orbits and are more likely to tunnel across the barriers as compared to when they are in ground state.

A single QCA cell can be modeled by a Hamiltonian of the form. The Hamiltonian of the extended Hubbard type is used to describe a single isolated cell as follow [1]:

$$\begin{aligned} H^{\text{cell}} &= \sum_{i,\sigma}(E_0 + V_i)\hat{n}_{i,\sigma} + \sum_{i>j,\sigma} t_{i,j}\left(\hat{a}^{\dagger}_{i,\sigma}\hat{a}_{i,\sigma} + \hat{a}^{\dagger}_{j,\sigma}\hat{a}_{j,\sigma}\right) \\ &\quad + \sum_{i} E_Q \hat{n}_{i,\uparrow}\hat{n}_{i,\downarrow} + \sum_{i>j,\sigma,\sigma'} V_Q \frac{\hat{n}_{i,\sigma}\hat{n}_{j,\sigma'}}{\left|R_i - R_j\right|} \end{aligned} \tag{2.5}$$

This model represents each quantum dot as a site and ignores internal degrees of freedom of the cell. Here, E_0 is the ground state energy for an electron in a single dot, n_i is the number density for site *i; $t_{i,j}$* is the coupling to neighboring dots, E_Q is the energy to put two electrons on a single dot, V_Q is the strength of the Coulomb interaction, and the R are the positions of the dots. The basis states for this Hamiltonian are taken to be states in which each electron is in the ground state of one of the individual dots. This Hamiltonian can be diagonalized directly to determine the two-particle states of the system. It should be emphasized that Eq. (2.5) is a model in the sense that the strengths of the various terms are put in as constants. The Hamiltonian used to model the cell includes four terms in which the first term represents the on-site energy of each dot. Here, the $\hat{a}^{\dagger}_{i,\sigma}\left(\hat{a}_{i,\sigma}\right)$ is the annihilation (creation) operators for an electron on site i with spin σ. $\hat{n}_{i,\sigma}$, represents the number operator for electrons of spin σ on site i and V_i is the potential energy of an electron at dot i due to charges outside the cell. $t_{i,j}$ is the tunneling energy between site '*i*' and '*j*'. The third terms is the Coulombic cost to put two electrons of opposite spin on a single dot, and the Coulombic interaction between the charge densities on different dots within a cell is calculated in the last term.

The stationary state of the cell is given by solving the time-dependent Schrödinger equation.

$$\hat{H}_{\text{cell}}|\psi_i\rangle = E_i|\psi_i\rangle \tag{2.6}$$

where $|\psi_i\rangle$ and E_i are the *i*th eigenstate and eigenvalue of the Hamiltonian, respectively. These eigenstate is found using the site-ket basis for the cases of electrons with opposite spin and the 16 possible states as follows:

$$\begin{aligned} |\varphi_1 &= \left| \begin{matrix} 0\,0\,0\,1 \\ 0\,0\,0\,1 \end{matrix} \right\rangle \\ |\varphi_2 &= \left| \begin{matrix} 0\,0\,0\,1 \\ 0\,0\,1\,0 \end{matrix} \right\rangle, \ldots, \\ |\varphi_{16} &= \left| \begin{matrix} 1\,0\,0\,0 \\ 1\,0\,0\,0 \end{matrix} \right\rangle \end{aligned} \tag{2.7}$$

Then, the Hamiltonian matrix and its associated eigenvectors will be calculated, in which matrix elements are evaluated as follows:

$$H_{i,j} = \left\langle \varphi_i | \hat{H} | \varphi_i \right\rangle \tag{2.8}$$

The ground state of the cell is defined as:

$$|\psi_0\rangle = \sum_j \psi_j^0 |\varphi_j\rangle \tag{2.9}$$

Here, $\left|\boldsymbol{\varphi}_{\boldsymbol{j}}\right\rangle$ and ψ_0 are the jth basis vector and its coefficient, respectively. Coefficient of the basis vector is found by the direct diagonalization of the Hamiltonian matrix. For the case of weak tunneling energy between the sites of the cell (less than the columbic energy), the electrons will remain largely localized and resulting in a polarized cell. If the tunneling energies become comparable to the coulomb energies (more than the coulomb energy), the polarization of the cell is eliminated. In this way, the quantity of the cell polarization is defined as:

$$p = \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{\rho_1 + \rho_2 + \rho_3 + \rho_4} \tag{2.10}$$

where ρ_i denotes the expectation value of the ground state charge configuration as follows:

$$\rho_i = \langle \psi_0 | n_i | \psi_0 \rangle \tag{2.11}$$

We can solve Schrödinger's equation for a system composed of many cells, the ground state of the entire system is found by iteratively solving each cell ground state [1, 3, 4].

2.4 QCA Implementation Techniques

The fundamental component for QCA computation is a bistable cell capable of interacting with its local neighbors. The cell is not necessarily being in quantum-mechanical coherent at all times; as a result, several non-quantum-mechanical realizations of QCA have developed. There are four distinct techniques for physically implementing QCA: metal-based [5], semiconductor [6], molecular [7], and magnetic [8]. In this section, a brief description of each implementation is provided.

2.4.1 Metal Island

Recent works have demonstrated that the metal island-based QCA devices are feasible and work correctly at cryogenic temperature. Metal QCA consists of four metal

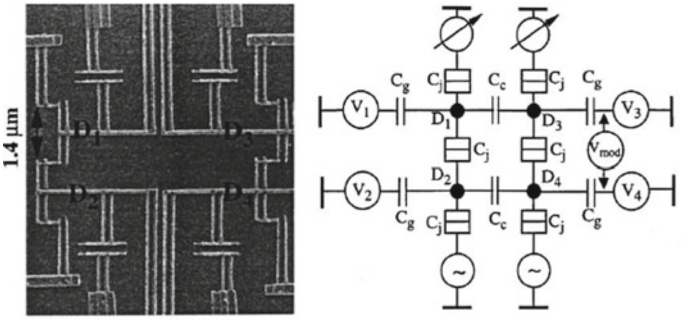


Fig. 2.6 SEM image of metal-dot QCA cell and corresponding schematic diagram [11]

islands, which are relatively large. Metal junction QCA [9, 10] was the first fabrication method considered to show the concept of QCA, where metallic tunnel junctions and very small capacitors are used to build the QCA cells. It was not aimed to compete with the existing technology in terms of speed and practicality, as its structural properties are not suitable for scalable designs. The basic idea of metal-dot QCA is to build quantum dots using aluminum islands. The cell size is approximately 60 nm by 60 nm, with junction capacitance of 400 aF [9]. The method has the advantages of an easier fabrication process, reliability, and ease of modeling and analyzing. However, it has one major drawback, which is the operating temperature. The prototype only operates at 10 K or below. The required quantum-mechanical effects only happen at this operating temperature. Metal-dot QCA is meant as a proof-of-concept implementation. In [11], authors reported a SPICE model development for QCA cells. Figure 2.6 shows a scanning electron microscope (SEM) image of the metal-dot QCA and its diagram.

2.4.2 Semiconductor

Semiconductor QCA implementations [12] can possibly be used to realize QCA devices with the same highly advanced semiconductor fabrication processes used to realize CMOS devices [13]. Semiconductor quantum dots are nanostructures formed using electron beam lithographically defined gates on heterostructure materials such as InAs/GaAs [14] and GaAs/AlGaAs [15, 16]. These structures can be modeled as 3-D quantum wells. Consequently, they show energy quantization effects even at distances several hundred times larger than the material system lattice constant. Cell polarization is encoded as charge position, and quantum-dot interactions depend on electrostatic coupling [17]. And recently published implementation is based on Silicon [18] (shown in Fig. 2.7a). Unfortunately, current semiconductor patterning tech-

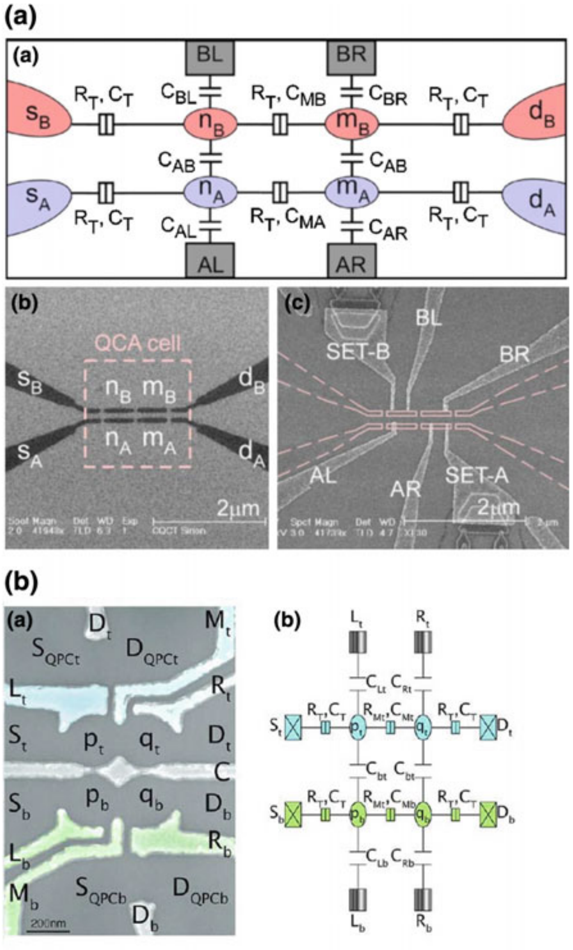


Fig. 2.7 **a** Silicon-based QCA schematic and SEM images [18], **b** electron micrograph of a GaAs/AlGaAs QCA cell with Simplified circuit equivalent of the four-dot cell [15]

nologies do not allow for a small enough size scale to make room temperature operation possible. Therefore, semiconductor QCA suffers from the same temperature and speed limitations found with metal-dot QCA.

2.4.3 Molecular QCA

Molecular QCA [20–22] concept consists of building QCA devices out of single molecules. Majority of the work so far has been presented by the research group at Notre Dame. The basic concept of molecular QCA is that each molecular QCA cell consists of a pair of identical allyl groups as shown in [19] and Fig. 2.8. The molecule shown in Fig. 2.8 is also known as a 1, 4-diallyl butane radical cation. This is formed

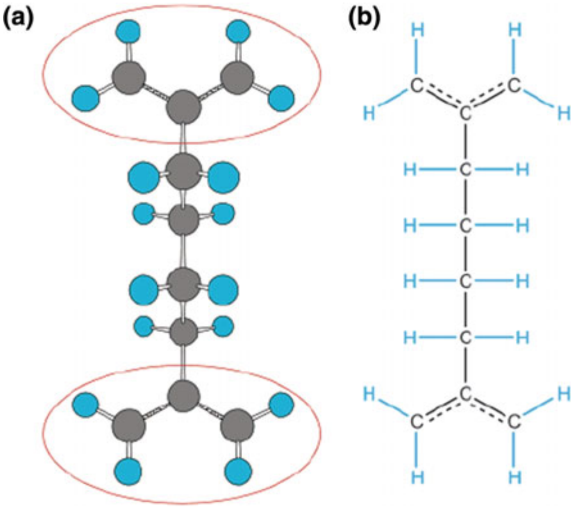


Fig. 2.8 Two views of molecule as a QCA cell [19]

by two allyl groups connected by a butyl bridge in between. This molecule is neutral on one end and the other end behaves as a cation. This molecule has an extra hole or electron that allows the quantum tunneling effect needed by QCA to happen. If an electrical field is placed near one end of the molecule, it can create either a repelling or attracting force. It has been calculated that the molecule in Fig. 2.8 has nonlinear switching characteristics, which make it an ideal switch. When the molecules are placed next to each other with a distance of seven angstroms, the electrostatic interaction will cause the holes to be at opposite ends, which makes the propagation of the electron feasible to create the state of the QCA cell. Figure 2.9 shows the different states of the molecular QCA. Part (a) is *a* +1 state, part (b) is a non-ideal state which is not needed, and part (c) is a −1 state. At this scale, the required quantum-mechanical effects can happen at room temperature. Molecular QCA is believed to have the following advantages: high density, high clock frequency from the gigahertz range to the terahertz range, low power consumption, and low power loss. An individual molecular QCA cell has been demonstrated. However, no complete circuit

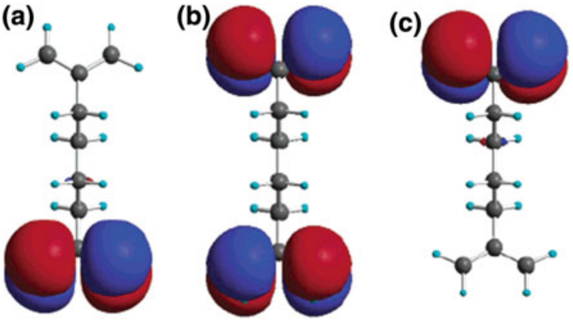


Fig. 2.9 Different possible states of molecule **a** show a +1 state, **b** show a non-ideal state that is a unwanted state, and **c** show a −1 state [20]

using molecular QCA has yet been demonstrated [23]. While fabrication methods are currently being researched, no one method has been predominating. Efforts are on to fabricate molecular QCA circuits using self-assembly monolayer methods [24, 25]. The molecules themselves are produced by standard chemical procedure [26, 27].

2.4.4 Magnetic QCA

A basic cell in magnetic QCA is a nanomagnet [28, 29]. These nanomagnets are arranged in various grid-like fashions to accomplish computing [30]. Cells in magnetic QCA are enumerated based on their single-domain magnetic dipole moments and are inherently energy minimums [31]. There are several popular schemes of magnetic QCA that have been proposed: Cowburn and Wellands nanodot QCA Automata [32], Parish and Forshaws Bistable Magnetic QCA [33], and Csaba et al., Field Couple Nanomagnets [34]. Cowburn and Wellands have fabricated the magnetic QCA model that has been described here. A nanomagnet consists of a single circular nanodot. These nanodots were made of a magnetic Supermalloy (mainly Ni). The nanodots are 110 nm in diameter and had a thickness of 10 nm. In order to have a single domain in the nanodots, it was found that the nanodots must have a size of about 100 nm and below. Nanodots are placed 20 nm apart on a straight line. The basic operation is to use an oscillating field on the dot to have it point to a certain direction to represent the binary value. Magnetic QCA cell is capable of operating at room temperature. Other advantages include high density and low power loss. The operating frequency is low (in the MHz range) when compared to CMOS [9]. A NOT gate and a majority gate have been demonstrated [28, 35]. Figure 2.10 shows a SEM image of a fabricated magnetic QCA network. Figure 2.11 shows the implementation of a majority gate using nanodots by Imre et al. [35].

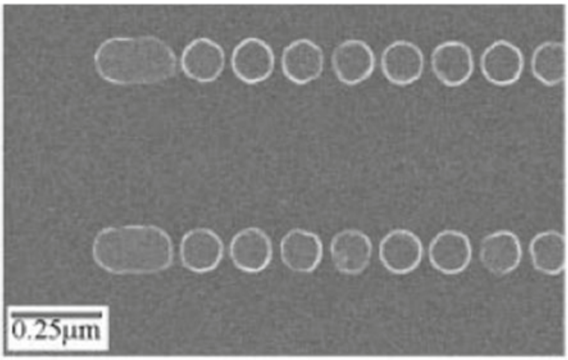


Fig. 2.10 SEM image of a room temperature MQCA network [32]

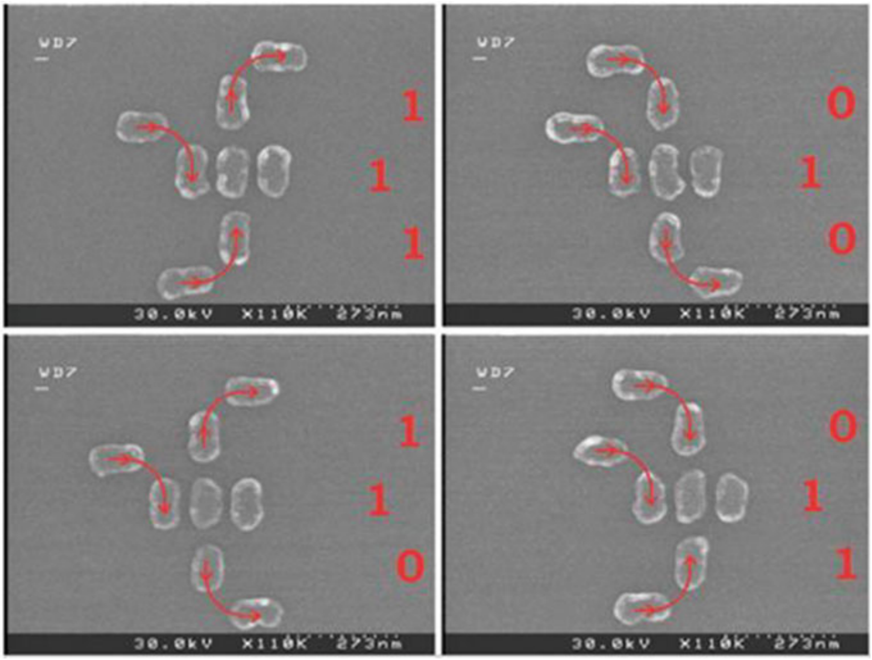


Fig. 2.11 Implementations of a majority gate for magnetic quantum-dot cellular automata [35]. The arrows drawn superimposed on the SEM images illustrate the resulting magnetization direction due to a horizontally applied external clock field

2.5 QCA Devices

This section explains the basic operation of QCA technology and its associated components, such as a cell, wire, majority gate, and inverter. The layout designs of the QCA circuits are the combination of all the mentioned components. Two different QCA structures of the fundamental gates are illustrated in Figs. 2.12a, b, namely the inverter (INV) and the majority gate (MV). In addition, Fig. 2.12c shows cascade of QCA cells to propagate binary data which represents a QCA wire. The 3-input majority gate function is described by the following equation:

$$MV3(A, B, C) = F = AB + BC + CA \tag{2.12}$$

2.6 QCA Clocking

In traditional VLSI technology, clocking mechanism is used to control the timing in sequential circuits. In QCA technology, a pipeline-based clock mechanism is

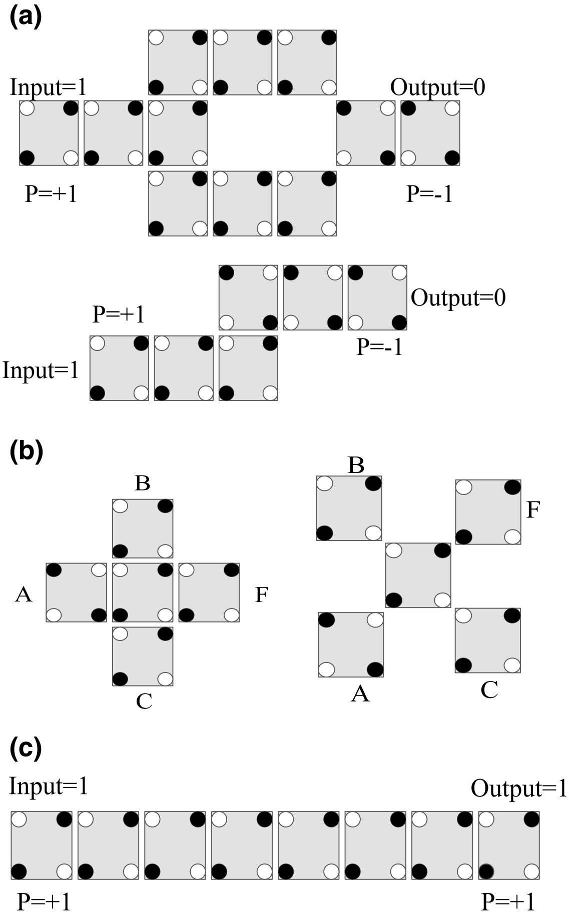
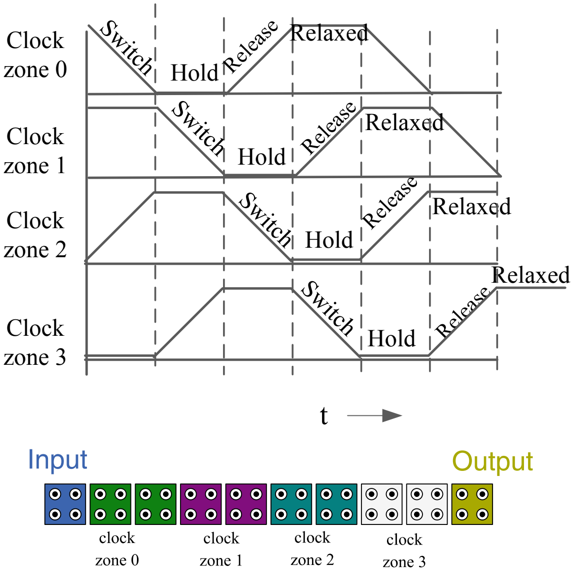


Fig. 2.12 QCA primitives: **a** two different realizations of inverter, **b** original majority gate (OMG) and rotated majority gate (RMG), and **c** QCA wire

essentially required for both sequential and combinational designs. This mechanism not only controls the data flow, but also supplies power for the cells. For the clocking purpose, four clocks are applied, i.e., clock 0, clock 1, clock 2, and clock 3. These clocks are 90° out of phase [36, 37] as apparent in Fig. 2.13. Each clock in QCA comprises of four distinct clock phases: switch, hold, release, and relax as depicted in Fig. 2.13 [38]. In switch state, cells start polarized and inter-dot barriers are raised and QCA cell attends one of the polarization states depending on the state of driving cell. During this phase, the real computation occurs. During the hold phase, cells have a fixed polarization to drive the succeeding stage. In the release phase, cells start unpolarized and during the final stage, inter-dot barriers stay lowered and a cell has no fixed polarization.

Fig. 2.13 QCA clocking with four phases



2.7 QCA Wire Crossing

In QCA structures, fabrication of interconnection between components needs to be handled efficiently for a better stability. Till date, there are two different types of crossover methods commonly utilized, coplanar and multilayer. The multilayer crossover uses more than one layer of cells (analogous to multiple metal layers in a conventional IC), shown in Fig. 2.14a but yields high cost due to, for instance, fabrication issue [39]. In coplanar crossover strategy, wire crossing is done by two different cells. These cells are orthogonal to each other, so they operate without affecting neighboring cells. The first wire consists of cells of 90° orientations and second wire has only 45° orientations, as depicted in Fig. 2.14b. The main drawback

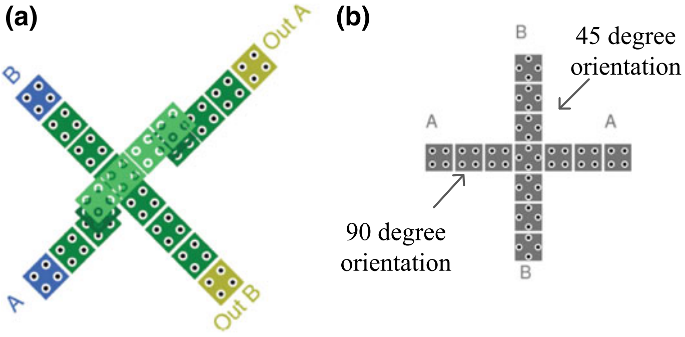
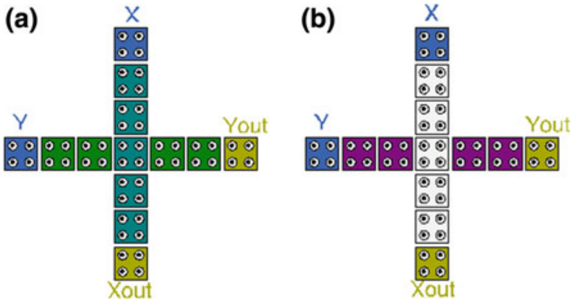


Fig. 2.14 Wire crossing **a** multilayer and **b** coplanar

of this scheme is that any misalignment of cells during fabrication may cause a cross-coupling between the two wires. Works have been done to mitigate such effects and to increase the robustness of the circuits, but all these ends up with large area overhead [40, 41]. In this work, wire crossing utilizes clock zone-based crossover [42] where cells on the switch phase can cross cells on the release phase and cells on the hold phase cross cells on the relax phase without polarization effect as depicted in Fig. 2.15a, b, respectively. This scheme takes advantage of two zones of the four-phase zone-based clocking scheme. For illustrations of Fig. 2.15b, the input/output signals with aforementioned clocking correspond to Clock0, and Clock2 are shown in Fig. 2.16, which is identically produced by Coherence vector and Bistable simulation

Fig. 2.15 Wire crossing using clock zone **a** cells on the switch phase cross cells on the release phase and **b** cells on the hold phase cross cells on the relax phase



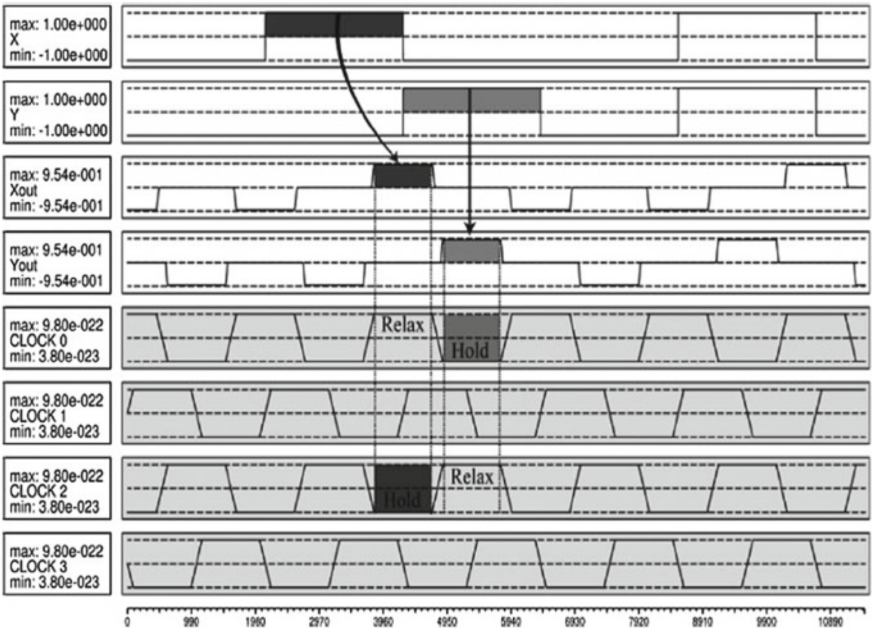


Fig. 2.16 I/O and clocking signals for clock zone-based crossover

engines with default QCADesigner parameters. This figure also indicates that the relax and hold phases of Clock0 coincide with the hold and relax phases of Clock2, respectively. Therefore, when the central cell is clocked by Clock2 (Clock0), signal X (Y) passes through.

2.7.1 Kink Energy and Cell Robustness

In QCA implementation of larger designs, designers are more concerned about increasing robustness/stability of whole QCA structure [43]. As the wire length increases, the switching probability of QCA cell decreases; similarly, QCA cell switches successfully for smaller wire length. Moreover, the circuits operate at higher clock rates. It has been experimentally shown that the number of cells in a wire length for healthy transmission of a signal is 28 (90°) or 27 (45°) [43]. At higher operating temperatures, due to thermal fluctuation, the QCA cell characteristic deviates [44], i.e., a kink to occur. To avoid kinks, the maximum number of cells is given by [44]

$$N \leq \mathrm{e}^{E_\mathrm{k}//\mathrm{k}_\mathrm{b}T} \tag{2.13}$$

where E_k is the kink energy, k_b is Boltzmann's constant and operating temperature T.

2.8 Modeling QCA Designs

There are several approximate simulators available at the layout level, such as the bistable simulation engine and the nonlinear approximation methods. The coherence vector-based method does explicitly estimate the polarizations, but it is appropriate when one needs full temporal dynamics simulation (Bloch equation), and hence is extremely slow. Coherence vector simulations are generally accepted as the most accurate simulation engine for clocked QCA due to the quantum mechanical properties which are integrated in the simulations. They also provide information on power, speed, and reliability and include temperature and other electrical properties. QCADesigner is an easy and useful program to design and simulate QCA circuits. QCADesigner is not just a switch-level simulator. It simulates QCA using the quantum mechanics of QCA. Once the cell is placed in the schematic, it is easy to change its clock zone or the rotation of the cell. To simulate the circuit, there are two steps to do. The first step is to set up the simulation engine, and the second step is to set up the simulation type. Simulation engine has two choices, coherence vector and bistable. The simulation type is to allow the user to set up the input test vectors. All the simulations were done in QCADesigner v2.0.3 using bistable approximation and coherence vector simulation engine [45]. The parameters of the simulation are the default values used by QCADesigner as shown in Table 2.1.

Table 2.1 Bistable approximation and coherence vector parameters model

Parameter	Bistable approximation	Coherence vector
Cell size	18 * 18 nm^2	18 * 18 nm^2
Number of samples	12,800	12,800
Convergence tolerance	0.001000	–
Radius of effect	65 nm	80 nm
Relative permittivity	12.9	12.9
Clock high	9.8e−22 J	9.8e−22 J
Clock low	3.8e−23 J	3.8e−23 J
Clock amplitude factor	2	2
Layer separation	11.500 nm	11.500 nm
Maximum iterations per sample	100	–
Relaxation time	–	4.135e−14 s
Time step	–	1e−016 s
Total simulation time	–	7e−011 s

2.8.1 QCA Power Dissipation Model

Work by Timler and Lent initially developed a power estimation model for QCA-based circuit [46]. A Hamiltonian matrix is used to measure energy related to a QCA cell. By considering Hartree–Fock approximation [47] and mean-field approach, Coulombic interaction between QCA cells [46, 48], and the Hamiltonian matrix for an array of cell is expressed as

$$H = \begin{bmatrix} -\frac{E_k}{2}\sum_{t} c_i f_{ij} & -\gamma \\ -\gamma & \frac{E_k}{2}\sum_{i} c_i f_{ij} \end{bmatrix} = \begin{bmatrix} -\frac{E_k}{2}\left(C_{j,1}+C_{j+1}\right) & -\gamma \\ -\gamma & \frac{E_k}{2}\left(C_{j,1}+C_{j+1}\right) \end{bmatrix} \tag{2.14}$$

where $f_{i,j}$ is a geometrical factor representing electrostatic interactions between cell i and cell j due to the geometrical distance and polarization of the ith juxtaposed cell is represented by C_i. If the space between neighboring cells are equal, then $f_{i,j}$ is interpreted as the kink energy, which can be calculated using the electrostatic interaction between all electrons in two cells, i and j, as

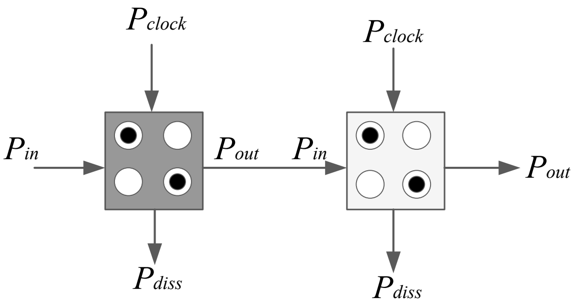


Fig. 2.17 Power flows between two QCA cells

$$E_{ij} = \frac{1}{4\pi\varepsilon_0\varepsilon_r}\sum_{n=1}^{4}\sum_{m=1}^{4}\frac{q_{i,n}q_{j,m}}{\left|r_{i,n} - r_{j,m}\right|} \tag{2.15}$$

At each clock cycle, the expectation value of QCA cell energy is expressed as

$$E = \langle H \rangle = \frac{\hbar}{2}\cdot\vec{\Gamma}\cdot\vec{\lambda} \tag{2.16}$$

where $E = \langle H \rangle = \frac{\hbar}{2}\cdot\vec{\Gamma}\cdot\vec{\lambda}$ is the Planck constant, $\vec{\Gamma}$ is the energy environment vector of the cell, and coherence vector is represented as $\vec{\lambda}$. The Hamiltonian vector is presented as

$$\vec{\Gamma} = \frac{1}{\hbar}\left[-2\gamma, 0, E_k\left(C_{j-1} + C_{j+1}\right)\right] \tag{2.17}$$

Here, $\left(C_{j-1} + C_{j+1}\right)$ represents the sum of neighboring polarizations. Power flow between neighboring cells is shown in Fig. 2.17. As mentioned in [46], P_{in} and P_{out} are the inflow signal power and the released signal power for a QCA cell. During the *switch* phase, P_{clock} amount of energy transfer to the cell as inter-dot barriers are raised. Similarly, in the *release* phase, the energy gets returned to the clocking circuit as barriers are reduced. During this process, a small power is dissipated in the clocking circuit named as P_{diss} [48, 49]. The total instantaneous power for a cell is given as

$$P_t = \frac{\mathrm{d}E}{\mathrm{d}t} = \frac{\hbar}{2}\left[\frac{\mathrm{d}\vec{\Gamma}}{\mathrm{d}t}\cdot\vec{\lambda}\right] + \frac{\hbar}{2}\left[\vec{\Gamma}\cdot\frac{\mathrm{d}\vec{\lambda}}{\mathrm{d}t}\right] = P_1 + P_2 \tag{2.18}$$

where P_1 combines the difference of input and output signal powers and clocking power to the cell. The term P_1 can be written:

$$P_1 = P_{\text{in}} - P_{\text{out}} + P_{\text{clock}} \tag{2.19}$$

Here, P_{clock} is the amount of transferred energy into the cell by the clock, P_{in} is the signal power in from the left side cell, and P_{out} is the signal power out to the right side cell. The power gain of each cell is then:

$$\text{gain} = \frac{P_{\text{out}}}{P_m} \tag{2.20}$$

The term P_2 is the cell dissipated power to the environment [49]. In QCA circuits, power consumption occurs in each cell during a quasi-adiabatic clocking scheme. It is noteworthy that a considerable amount of energy is transferred to the cell as the barriers are being raised. Most of that energy is returned to the clock as the barriers are being lowered. The difference between these two amounts is the consumed power which is categorized into two types: switching and leakage powers. The switching power occurs when the cell actually changes the state from ‘0’ to ‘1’ or inversely. Leakage power is dependent on the clock energy changing to polarize or depolarize a cell [50, 51].

According to [46], the Hamiltonian and coherence vectors can be used to calculate the energy dissipation in one clock cycle $T_{\text{cc}} = [-T, T]$ as

$$E_{\text{diss}} = \frac{\hbar}{2}\int_{-\pi}^{T} \vec{\Gamma}\cdot\frac{\mathrm{d}\vec{\lambda}}{\mathrm{d}t}\mathrm{d}t = \frac{\hbar}{2}\left(\left[\vec{\Gamma}\cdot\vec{\lambda}\right]_{-T}^{T} - \int_{-T}^{T}\vec{\lambda}\cdot\frac{\mathrm{d}\vec{\Gamma}}{\mathrm{d}t}\mathrm{d}t\right) \tag{2.21}$$

The upper bound power dissipation model in [51] is presented as

$$P_{\text{diss}} = \frac{E_{\text{diss}}}{T_{\text{cc}}} < \frac{\hbar}{2T_{\text{cc}}}\vec{\Gamma}_{+} \times \left[-\frac{\vec{\Gamma}_{+}}{\left|\vec{\Gamma}_{+}\right|}\tanh\left(\frac{h\left|\vec{\Gamma}_{+}\right|}{k_B T}\right) + \frac{\vec{\Gamma}_{-}}{|\vec{\Gamma}_{-}|}\tanh\left(\frac{h|\vec{\Gamma}_{-}|}{k_s T}\right)\right] \tag{2.22}$$

Here, $\vec{\Gamma}_{+}$ and $\vec{\Gamma}_{-}$ represent $\vec{\Gamma}(+T)$ and $\vec{\Gamma}(-T)$, respectively, k_{B} defines the Boltzmann constant, and T is the temperature.

Authors in [52] addressed a power dissipation model by considering above concepts and developed a power estimation tool known as QCAPro. This helps to evaluate the total power loss in a QCA circuit as a combination of leakage and switching power when clock changes.

2.9 Summary

An overview of QCA devices and associated logic circuits are explained in this chapter. It includes basics of QCA, comprehensive review from the literature, various types of implementation techniques along with clocking, and wire crossing mechanisms. A brief discussion of various types of design and simulation tools is also provided.

References

1. C.S. Lent, P.D. Tougaw, W. Porod, G.H. Bernstein, Quantum cellular automata. Nanotechnology **4**(1), 49–57 (1993)
2. S.O. Kasap, *Principles of Electronic Materials and Devices* (McGraw Hill, 2002)
3. P.D. Tougaw, C.S. Lent, Logical devices implemented using quantum cellular automata. J. Appl. Phys. **75**(3), 1818–1825 (1994)
4. C.S. Lent, P.D. Tougaw, A device architecture for computing with quantum dots. Proc. IEEE **85**(4), 541–557 (1997)
5. G. Toth, C.S. Lent, Quasi-adiabatic switching for metal-island quantum-dot cellular automata. J. Appl. Phys. **85**, 2977–2984 (1999)
6. G. Toth, C.S. Lent, Quantum computing with quantum-dot cellular automata. Phys. Rev. A **63**, 052315 (2001)
7. C.S. Lent, B. Isaksen, Clocked molecular quantum-dot cellular automata. IEEE Trans. Electron Dev. **50**(9), 1890–1896 (2003)
8. M.C.B. Parish, *Modeling of physical constraints on bistable magnetic quantum cellular automata*. Ph.D. Dissertation. University of London, UK (2003)
9. G. Bernstein, Quantum-dot cellular automata: computing by field polarization, in *Proceedings of the 40th Annual Design Automation Conference*, Series ACM (2003), pp. 268–273
10. M. Liu, C.S. Lent, High-speed metallic quantum-dot cellular automata, in *2003 Third IEEE Conference on Nanotechnology*. *IEEE-NANO 2003*, vol. 2, Aug 2003, pp. 465–468
11. A.O. Orlov, I. Amlani, G. Toth, C.S. Lent, G.H. Bernstein, G.L. Snider, Experimental demonstration of a binary wire for quantum-dot cellular automata. Appl. Phys. Lett. **74**(19), 2875–2877 (1999)
12. C. Single, R. Augke, F. Prins, D. Wharam, D. Kern, Towards quantum cellular automata operation in silicon: transport properties of silicon multiple dot structures. Superlattices Microstruct. **28**(5), 429–434 (2000)
13. C. Smith, S. Gardelis, A. Rushforth, R. Crook, J. Cooper, D. Ritchie, E. Linfield, Y. Jin, M. Pepper, Realization of quantum-dot cellular automata using semiconductor quantum dots. Superlattices Microstruct. **34**(3), 195–203 (2003)
14. S. Suraprapapich, S. Panyakeow, C.W. Tu, Effect of arsenic species on the formation of (Ga)InAs nanostructures after partial capping and regrowth. Appl. Phys. Lett. **90**(18), 183112 (2007)
15. F. Perez-Martinez, I. Farrer, D. Anderson, G.A.C. Jones, D.A. Ritchie, S.J. Chorley, C.G. Smith, Demonstration of a quantum cellular automata cell in a GaAs/AlGaAs heterostructure. Appl. Phys. Lett. **91**(3), 032102 (2007)
16. S. Gardelis, C.G. Smith, J. Cooper, D.A. Ritchie, E.H. Linfield, Y. Jin, Evidence for transfer of polarization in a quantum dot cellular automata cell consisting of semiconductor quantum dots. Phys. Rev. B **67**, 033302 (2003)
17. K. Walus, R.A. Budiman, G.A. Jullien, Impurity charging in semiconductor quantum-dot cellular automata. Nanotechnology **16**(11), 2525 (2005)
18. M. Mitic, M.C. Cassidy, K.D. Petersson, R.P. Starrett, E. Gauja, R. Brenner, R.G. Clark, A.S. Dzurak, C. Yang, D.N. Jamieson, Demonstration of a silicon-based quantum cellular automata cell. Appl. Phys. Lett. **89**(1), 013503 (2006)
19. C.S. Lent, B. Isaksen, M. Lieberman, Molecular quantum-dot cellular automata. J. Am. Chem. Soc. **125**(4), 1056–1063 (2003)
20. M. Lieberman, S. Chellamma, B. Varughese, Y. Wang, C. Lent, G. Bernstein, G. Snider, F. Peiris, Quantum-dot cellular automata at a molecular scale. Ann. NY Acad. Sci. **960**, 225–239 (2002)
21. Y. Lu, M. Liu, C. Lent, Molecular quantum-dot cellular automata: from molecular structure to circuit dynamics. J. Appl. Phys. **102**(3), 034311 (2007)
22. Z. Li, M. Lieberman, W. Hill, XPS and SERS study of silicon phthalocyanine monolayers: umbrella vs octopus design strategies for formation of oriented sams. Langmuir **17**(16), 4887–4894 (2001)

23. K. Walus, F. Karim, A. Ivanov, Architecture for an external input into a molecular QCA circuit. J. Comput. Electron. **8**(1), 35–42 (2009)
24. W. Hu, K. Sarveswaran, M. Lieberman, G.H. Bernstein, High-resolution electron beam lithography and DNA nano-patterning for molecular QCA. IEEE Trans. Nanotechnol. **4**(3), 312–316 (2005)
25. Q. Hang, Y. Wang, M. Lieberman, G.H. Bernstein, A liftoff technique for molecular nanopatterning. J. Nanosci. Nanotechnol. **3**(4), 309–312 (2003)
26. J. Jiao, G.J. Long, F. Grandjean, A.M. Beatty, T.P. Fehlner, Building blocks for the molecular expression of quantum cellular automata. Isolation and characterization of a covalently bonded square array of two ferrocenium and two ferrocene complexes. J. Am. Chem. Soc. **125**(25), 7522–7523 (2003)
27. Z. Li, A.M. Beatty, T.P. Fehlner, Molecular QCA cells. Structure and functionalization of an unsymmetrical dinuclear mixed-valence complex for surface binding. Inorg. Chem. **42**, 5707–5714 (2003)
28. R.P. Cowburn, Digital nanomagnetic logic, in *61st Device Research Conference. Conference Digest* (Cat. No. 03TH8663), June 2003, pp. 111–114
29. J.F. Pulecio, S. Bhanja, Reliability of bi-stable single domain nano magnets for cellular automata, in *2007 7th IEEE Conference on Nanotechnology (IEEE NANO)*, Aug 2007, pp. 782–786
30. S.A. Haque, M. Yamamoto, R. Nakatani, Y. Endo, Magnetic logic gate for binary computing. Sci. Technol. Adv. Mater. **5**(1–2), 79–82 (2004)
31. R.P. Cowburn, D.K. Koltsov, A.O. Adeyeye, M.E. Welland, D.M. Tricker, Single-domain circular nanomagnets. Phys. Rev. Lett. **83**, 1042–1045 (1999)
32. R.P. Cowburn, M.E. Welland, Room temperature magnetic quantum cellular automata. Science **287**(5457), 1466–1468 (2000)
33. M.C.B. Parish, M. Forshaw, Physical constraints on magnetic quantum cellular automata. Appl. Phys. Lett. **83**(10), 2046–2048 (2003)
34. G. Csaba, A. Imre, G.H. Bernstein, W. Porod, V. Metlushko, Nanocomputing by field-coupled nanomagnets. IEEE Trans. Nanotechnol. **1**(4), 209–213 (2002)
35. A. Imre, G. Csaba, L. Ji, A. Orlov, G.H. Bernstein, W. Porod, Majority logic gate for magnetic quantum-dot cellular automata. Science **311**(5758), 205–208 (2006)
36. K. Walus, G.A. Jullien, Design tools for an emerging soc technology: quantum-dot cellular automata. Proc. IEEE **94**(6), 1225–1244 (2006)
37. Y. Wang, M. Lieberman, Thermodynamic behavior of molecular-scale quantum-dot cellular automata (QCA) wires and logic devices. IEEE Trans. Nanotechnol. **3**(3), 368–376 (2004)
38. C.S. Lent, M. Liu, Y. Lu, Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling. Nanotechnology **17**(16), 4240 (2006)
39. M. Crocker, M. Niemier, X.S. Hu, M. Lieberman, Molecular QCA design with chemically reasonable constraints. J. Emerg. Technol. Comput. Syst. **4**(2), 9:1–9:21 (2008)
40. S. Bhanja, M. Ottavi, F. Lombardi, S. Pontarelli, QCA circuits for robust coplanar crossing. J. Electron. Test. **23**(2), 193–210 (2007)
41. S. Bhanja, M. Ottavi, F. Lombardi, Novel designs for thermally robust coplanar crossing in QCA, in *Proceedings of the Conference on Design, Automation and Test in Europe: Proceedings, ser. DATE '06*. 3001 Leuven, Belgium (European Design and Automation Association, Belgium, 2006), pp. 786–791
42. S. Shin, J.-C. Jeon, Y. Kee-Young, Design of wire-crossing technique based on difference of cell state in quantum-dot cellular automata. Int. J. Control Autom. **7**, 153–164 (2014)
43. K. Kim, K. Wu, R. Karri, Towards designing robust QCA architectures in the presence of sneak noise paths, in *Design, Automation and Test in Europe*, Mar 2005, pp. 1214–1219
44. C.S. Lent, P.D. Tougaw, W. Porod, Quantum cellular automata: the physics of computing with arrays of quantum dot molecules, in *Workshop on Physics and Computation, 1994. PhysComp '94, Proceedings*, Nov 1994, pp. 5–13
45. K. Walus, T.J. Dysart, G.A. Jullien, R.A. Budiman, Qcadesigner: a rapid design and simulation tool for quantum-dot cellular automata. IEEE Trans. Nanotechnol. **3**(1), 26–31 (2004)

46. J. Timler, C.S. Lent, Power gain and dissipation in quantum-dot cellular automata. J. Appl. Phys. **91**(2), 823–831 (2002)
47. P.D. Tougaw, C.S. Lent, Dynamic behavior of quantum cellular automata. J. Appl. Phys. **80**, 4722–4736 (1996)
48. W. Liu, S. Srivastava, L. Lu, M. O'Neill, E.E. Swartzlander, Are QCA cryptographic circuits resistant to power analysis attack? IEEE Trans. Nanotechnol. **11**(6), 12391251 (2012)
49. S. Srivastava, S. Sarkar, S. Bhanja, Estimation of upper bound of power dissipation in QCA circuits. IEEE Trans. Nanotechnol. **8**(1), 116–127 (2009)
50. T. Ghani, K. Mistry, P. Packan, S. Thompson, M. Stettler, S. Tyagi, M. Bohr, Scaling challenges and device design requirements for high performance sub-50 nm gate length planar CMOS transistors, in *2000 Symposium on VLSI Technology. Digest of Technical Papers* (Cat. No.00CH37104), June 2000, pp. 174–175
51. M. Taheri, R. Akbar, F. Safaei, M.H. Moaiyeri, Comparative analysis of adiabatic full adder cells in cnfet technology. Eng. Sci. Technol. Int. J. **19**(4), 2119–2128 (2016)
52. S. Srivastava, A. Asthana, S. Bhanja, S. Sarkar, QCAPro—an error-power estimation tool for QCA circuit design, in 2011 IEEE International Symposium of Circuits and Systems (ISCAS), May 2011, pp. 2377–2380

Chapter 3
Fundamental of Reversible Logic

All classical logic circuits are physically irreversible, because these circuits comprise irreversible logic. With irreversible logic, all the energy transferred by the power supply is finally converted into heat. Irreversible logic does not allow traverse the state sequences in the reverse direction to gain the initial state after the end of logical computation. This chapter discusses different aspects of both irreversible and reversible logic gates.

3.1 Irreversible Function

A useful entity in reversible logic is the Boolean function$f(x_1, x_2, \ldots, x_n)$ that receives n bits as arguments and generates one bit. Since 2^n different values can be obtained for n bits, one can determine 2^{2n} viable n-bit functions that can be generated using n bits. For this purpose, it will be sufficient to take a block diagram for irreversible computation, where $z = f(x_1, x_2, \ldots, x_n)$ represents a single-valued function on n discrete inputs, as depicted in Fig. 3.1. In the classical computation, we generally use base-2 arithmetic to represent the inputs and outputs, where $x_1, x_2, \ldots, x_n$, and y are binary variables, or bits, taking either 0 or 1. In this case, the Boolean expression $f(x_1, x_2, \ldots, x_n)$ is known as an n-bit irreversible function.

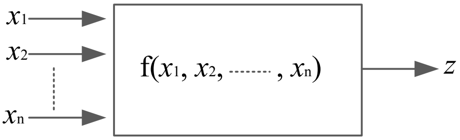


Fig. 3.1 Block diagram for an n-bit irreversible Boolean function

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_3

3.2 Irreversible Gates

Any switching function can be "built up" from a basic set of Boolean functions that operate on a small number of inputs, say two or three at a time, taken from $x_1, x_2, \ldots,$ x_n. These basic functions are called logic gates that can be used repetitively in order to implement 'f' on its inputs. A set of gates that can devolve any Boolean function is said to be universal gates. The logical operations AND ($\wedge$) and OR ($\vee$) demonstrate, respectively, the concept of logical conjunction and disjunction. The logical function AND is interpreted to result true if and only if both inputs are true. Conversely, for the logical OR function the output returns true if one of the inputs evaluates to true, where FALSE and TRUE are considered synonymously for bit values 0 and 1, respectively.

One suitable method to specify a logic gate in a compact form is using 'truth table.' A truth table lists all possible input combinations with relation between the values and the result of the operation. For instance, the truth table of AND gate is given in Table 3.1, and its circuit symbol is depicted in Fig. 3.2. One AND gate is inherently irreversible as inputs cannot be derived from the knowledge of output. Particularly, when output is zero, corresponding inputs can be 00, 01, or 10, i.e., no one-to-one relationships exist among inputs and outputs. This results in information lost when output of AND gate is zero. Further, the truth table of OR gate is shown in Table 3.2, and its circuit symbol is depicted in Fig. 3.3. It can be easily verified from truth table that the output is 1 for inputs 01, 10, or 11, so no one-to-one relationships exist among inputs and outputs. Thus, the OR gate loses some information for specific output. Another variant of the OR gate that performs modulo sum operation without including carry is known as exclusive-OR (XOR) gate. An XOR gate is normally two-input logic gate, where output is only logical 1 when only one input is logical 1. When both inputs are equal, that is either both are 1 or both are 0, the output will be logical 0. The truth table of XOR is given in Table 3.3, and its symbol is depicted in Fig. 3.4. It can be referred from the truth table that no one-to-one relationships exist among inputs and outputs, so logically irreversible gate.

Table 3.1 Truth table of AND gate

x	y	$x \bigwedge y$
0	0	0
0	1	0
1	0	0
1	1	1

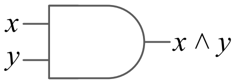


Fig. 3.2 Graphical symbol

Table 3.2 Truth table of OR gate

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

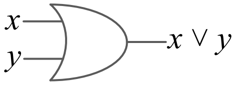


Fig. 3.3 Graphical symbol

Table 3.3 Truth table of XOR gate

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

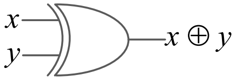


Fig. 3.4 Graphical symbol

These logic blocks are the major elements for all modern computing devices, where irreversible computation is carried out using series of irreversible logic units those work on a few bits at any moment.

3.3 Irreversible Parameters

Classical logic gates are analyzed and compared using various parameters such as fan-out, power dissipation, and propagation delay.

3.3.1 Fan-Out

The fan-out, generally known as loading of an irreversible gate, indicates the number of inputs that are driven by the gate without compromising its normal operation. When the gate output is generally associated with other gates inputs, the driving gate can supply limited amount of current. This is expressed by a number which provides information about the amount of current accessible at the gate output and the current required by the individual input of gates. Therefore, those circuits which are not following this rule may not work correctly as the circuits unable to supply the demanded power [1].

3.3.2 Power Dissipation

All electronic circuits need some amount of power to do the operations. The power dissipation of a gate is the power required by the gate for its operation and is expressed in milliwatts (mW). This shows the power supplied to each individual gate from power source, rather than the power delivered from other gates. Power dissipation in a circuit is directly related to the heat produced by that circuit. Excessive heat dissipation can raise working temperature, and the corresponding gates start shift out from its normal operating range by generating erroneous outputs. Hence, power dissipation of each gate realization should maintain at low possible value [1].

3.3.3 Propagation Delay

Propagation delay of a logic gate is the time required for the signal to travel from input of logic gate to output. Generally, signal through the gate takes certain time for the effect of change in input to be visible at the output. This time interval is well known as the propagation delay of the gate. The time between the input and output transitions is not a suitable measure to calculate the delay time of a gate for two reasons: First, the input and the output signals considered for the gates are not the ideal waveforms studied in theory. The non-ideal input and output signals to a NOT gate are depicted in Fig. 3.5, where the transitions between high level and low level take finite time. Secondly, the input to a gate has to reach the threshold level before the gate starts to change state. Therefore, the delay time of a gate is calculated at 50% point of input–output switching, defined as reference voltage level V_{ref}.

3.4 Reversible Logic

Nanotechnology offers new ways of computing in emerging nanoelectronics and possible solution for electronics industry at the end of roadmap. Power management plays a vital role in the advancement of recent computational systems. For nanotechnology-based system, the total heat dissipation is relay upon the number of bits erasure when some computation is performed. Authors in [2] reported that for any technology, the conventional logic-based circuits dissipate heat in factor of $kT\ln 2$ joules (k is Boltzmann's constant and T is the temperature during operating condition) for each bit erasures. In this direction, lossless computing may be used as

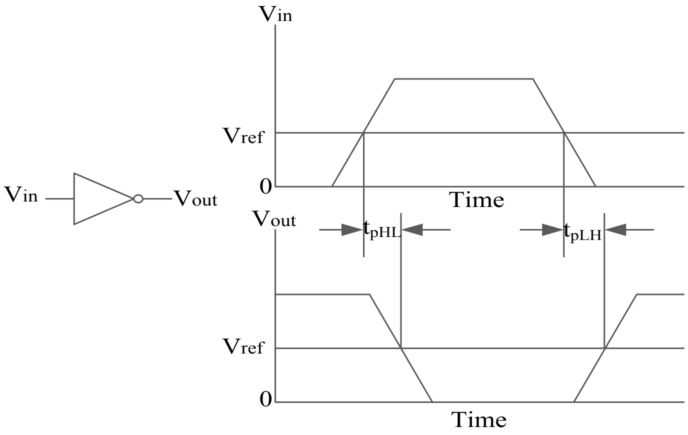


Fig. 3.5 Propagation delays for a NOT gate, measured at the midpoints of transitions (t_{pHL}: delay, when output switches from high to low, after input switches from low to high. t_{pLH}: when output switches from low to high, after input switches from high to low)

possible solution which allows a logical operation without employing information loss, known as Reversible operation. For reversible computing, there must exist a bijective mapping among input and output pattern, i.e., output set can be used to restore the status of input set. Thus, circuit based on reversible logic possibly can work in nanoscale, providing lossless computation due to information lost.

There are two different ways to attain reversibility: (i) logical reversibility and (ii) physical reversibility. Logical reversibility means if there exist input set and output set, then each element of input is paired with only one element of output, i.e., all the elements are paired one-to-one. In addition, inputs are reconstructed from the output information. Physical reversible is attainable for system, which enables computation in reversible order. Generally, logical reversibility can be obtained in two ways. First, the intermediate values in the forward direction must be stored for later usage. So that during backward computation, these stored values can be used. Once the information are used, the stored values are erased. This requires a dynamic structure to retain all intermediate states, where Bennett clocking [3] can be used. Second, all computations are done without keeping intermediate values, i.e., by using dedicated reversible logic blocks such as Toffoli gate. Researchers in [4, 5] demonstrate the generic relationship between logical reversibility and physical reversibility. Table 3.4 lists some primitive reversible gates.

3.4.1 Reversible Function

An n-input, n-output function $f(x_1, x_2, \ldots, x_n)$ of n variable is a reversible function if it maps distinct input to distinct output, i.e., a $n * n$ bijective function mapping

Table 3.4 Primitive reversible gates

Gate name	Size	Input ⟷ output
Toffoli [6]	3 × 3	$A \longleftrightarrow P = A$
		$B \longleftrightarrow Q = B$
		$C \longleftrightarrow R = AB \oplus C$
Fredkin [7]	3 × 3	$A \longleftrightarrow P = A$
		$B \longleftrightarrow Q = A'B + AC$
		$C \longleftrightarrow R = A'C + AB$
Peres [8]	3 × 3	$A \longleftrightarrow P = A$
		$B \longleftrightarrow Q = A \oplus B$
		$C \longleftrightarrow R = AB \oplus C$
Feynman (CNOT) [9]	2 × 2	$A \longleftrightarrow P = A$
		$B \longleftrightarrow Q = A \oplus B$

between input and output. In general, a truth table or a permutation can be used to represent a reversible function. Let us define reversible function f: I → O, where I is the input set and O is the output set, which can be expressed as a set of decimal number $(0, 1, \ldots, 2^n - 1)$ for each n-bit binary pattern. So the reversible function $f = (O_0, \ldots, O_N)$ can be expressed as permutation of input set $(I_0, \ldots, I_N)$, where $N = 2^n - 1$. An identity function $f_I = [0, 1, \ldots, N]$ is a special reversible function, which reproduces input binary patterns at the output.

If a given objective function (permutation of inputs) can be realized by cascade of reversible gates $\{G_1, G_2, \ldots G_m\}$ starting from identity function, then cascade of same gates in reverse order, i.e., $\{G_m, G_{m-1}, \ldots G_1\}$ transform the given function into identity function, i.e., computation can be done in both directions.

3.5 Reversible Gate

A *reversible gate* realizes a reversible function. If a reversible gate has k inputs, and therefore k outputs, then we call it a $k * k$ reversible gate. In reversible gates, the number of inputs is equal to the number of outputs and any input pattern maps to a unique output pattern. Different gate libraries are available, among which NCT, NCTSF, and NCTSFP (NOT, CNOT, TOFFOLI, SWAP, FREDKIN, PERES) libraries are adopted for synthesizing a reversible function.

3.5.1 NOT Gate

A NOT gate is a 1 × 1 gate as drawn in Fig. 3.6a. This gate shows quantum cost of 1, as it comprises one 1 × 1 gate.

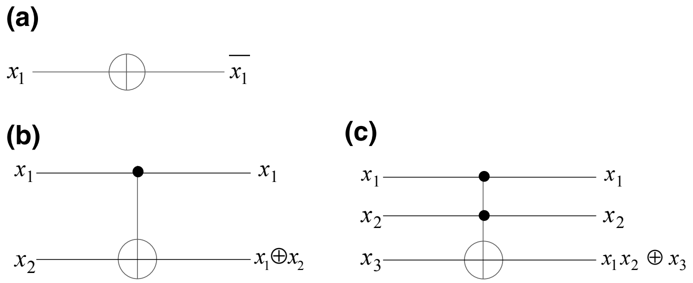


Fig. 3.6 Logic symbol for **a** NOT gate, **b** Feynman gate, and **c** Toffoli gate

3.5.2 Feynman Gate (CNOT Gate)

The Feynman gate (FG) or CNOT is a 2 × 2 reversible gate which pairs inputs ($x1$, $x2$) to outputs ($1, x1 \oplus x2$). The quantum cost of this gate is 1, as it consists of a 2 × 2 gate. Figure 3.6b shows symbolic representation of the Feynman gate along with the relations that govern it. The Feynman gate can be used to provide more fan-out in reversible logic.

3.5.3 Toffoli Gate

The Toffoli gate is another most widely used universal reversible gate represented as TOF($C; T$) or C^nNOT($x_1, x_2, \ldots, x_{n+1}$), where $C = (x_{i_1}, \ldots, x_{i_k}) \subset X$ a set of control lines, and a single target line $T = \{x_j\}$, $C \cap T = \Phi$, $X = \{x_1, x_2, \ldots, x_{n+1}\}$ input variables. It pairs a Boolean set $(x_1^0, x_2^0, \ldots, x_{n+1}^0)$ to $(x_1^0, x_2^0, \ldots, x_{j-1}^0, x_j^0 \oplus x_{i_1}^0 x_{i_2}^0 \ldots x_{i_k}^0, x_{j+1}^0, \ldots, x_{n+1}^0)$. For $n = 0$, a generalized Toffoli gate is noted as NOT gate without any control lines. For $n = 1$, the CNOT gate commonly known as Feynman gate has one control line [9]. For $n = 2$, the C^2NOT gate also known as Toffoli gate having two control lines [6], as depicted in Fig. 3.6c. These three gates constitute the universal NCT library, which is redrawn in Fig. 3.6.

3.5.4 Fredkin Gate

For the set of input variables $X = \{x_1, x_2, \ldots, x_n\}$, the generalized Fredkin gate can be represented as Fred Fred($C; T$) or Fred($x_1, x_2, \ldots, x_n$), where $C = \{x_{i_1}, \ldots, x_{i_k}\} \subset X$ a set of control lines, and target line $T\{x_j, x_l\}$ and $C \cap T = \Phi$ It maps a Boolean pattern $(x_1^0, x_2^0, \ldots, x_n^0)$ to $(x_1^0, x_2^0, \ldots, x_{j-1}^0, x_l^0 x_{j+1}^0, \ldots, x_{l-1}^0 x_j^0, x_{l+1}^0, \ldots, x_n^0)$ iff $x_{i_1}^0 x_{i_2}^0 \ldots x_{i_k}^0 = 1$. Otherwise the passed input will not be interchanged. For $n = 0$, a gate with no control signal

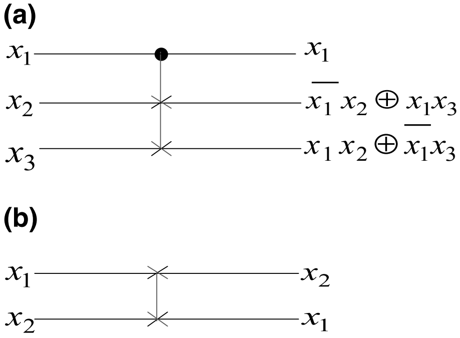

(a)
x_1
x_2
x_3
x_1
$\overline{x_1}\, x_2 \oplus x_1 x_3$
$x_1\, x_2 \oplus \overline{x_1} x_3$
(b)
x_1
x_2
x_2
x_1

Fig. 3.7 **a** Fredkin gate and **b** SWAP gate

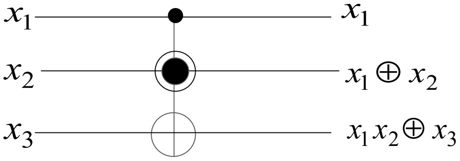


Fig. 3.8 Peres gate

Fred(x_1, x_2) is called SWAP, as it exchanges the values on signal x_1, x_2. For $n = 1$, a gate with one control signal Fred$(x_1; x_2, x_3)$ is commonly known as Fredkin gate [7]. NCTSF library forms by adding these gates to the NCT library. Figure 3.7 shows SWAP and Fredkin gates.

3.5.5 Peres Gate

Peres gate [8] $P(x_1; x_2, x_3)$ has one control line x_1 and two target lines x_2 and x_3. It represents a cascade of a C^2NOT$(x_1; x_2, x_3)$ and a CNOT(x_1, x_2). It maps input (x_1, x_2, x_3) to $(x_1,\ x_1 \oplus x_2, x_2 \oplus x_3)$. The general structure of Peres gate is illustrated in Fig. 3.8.

3.5.6 Quantum Gate

Quantum bit, or qubit, in quantum system is analogous to bit in classical computation. It can be described as mathematical objects. Authors in [10] show physical realization and connection with abstract mathematical objects. Classical bit has a state either 0 or 1, whereas a qubit has two possible states $|0\rangle$ and $|1\rangle$ (Notation '$|\rangle$' is called the Dirac notation [11]). The difference between bits and qubits is that a qubit can be in state other than $|0\rangle$ and $|1\rangle$. There is also the possibility of storing information as a superposition of states may be written as:

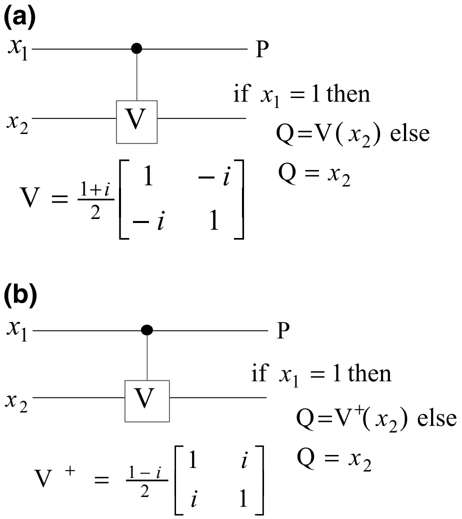

(a)
x_1
P
x_2
V
if $x_1 = 1$ then
Q=V(x_2) else
Q = x_2
$V = \frac{1+i}{2}\begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}$
(b)
x_1
P
x_2
V
if $x_1 = 1$ then
Q=V$^+$(x_2) else
Q = x_2
$V^+ = \frac{1-i}{2}\begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$

Fig. 3.9 **a** Controlled-*V* gate and **b** controlled-*V*+ gate

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{23}$$

where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$.

A matrix describing the single qubit gate is unitary if $UU^\dagger = I$ where $U^\dagger$ is the conjugate transpose of U and I is the identity matrix. An n-qubit quantum gate is a device which performs a $2^n \times 2^n$ unitary operation U on n qubits in a specific period of time. The popular elementary gates are the NOT, CNOT, controlled-*V*, and controlled-*V*+ (NCV) gates [9, 12]. Reversible logic synthesis can be realized with a combination of elementary quantum gates [10]. Figure 3.9 shows elementary controlled-*V* and controlled-*V*+ gates. The realization of Toffoli, Peres, SWAP, and Fredkin gates using elementary gates are illustrated in Fig. 3.10.

3.6 Reversible and Quantum Logic Design Metrics

3.6.1 Quantum Cost (QC)

Classical reversible logic circuits can be realized using elementary quantum gates (1 × 1, and 2 × 2 gates); moreover, the quantum technology is one of the best ways for implementation of reversible logic circuits. Quantum cost of a logic circuit is defined as the number of elementary quantum gates that need to realize a reversible or quantum logic circuit [13]. Each quantum gate (1 × 1, and 2 × 2 gates) adds a QC of Δ. A 3 × 3 Toffoli gate can be implemented using five primitive gates, thus QC of 5. The well-known 3 × 3 Fredkin, Peres, and SWAP gates have QC of 5, 4, and 3, respectively. QC can be calculated either implementing new gate using only

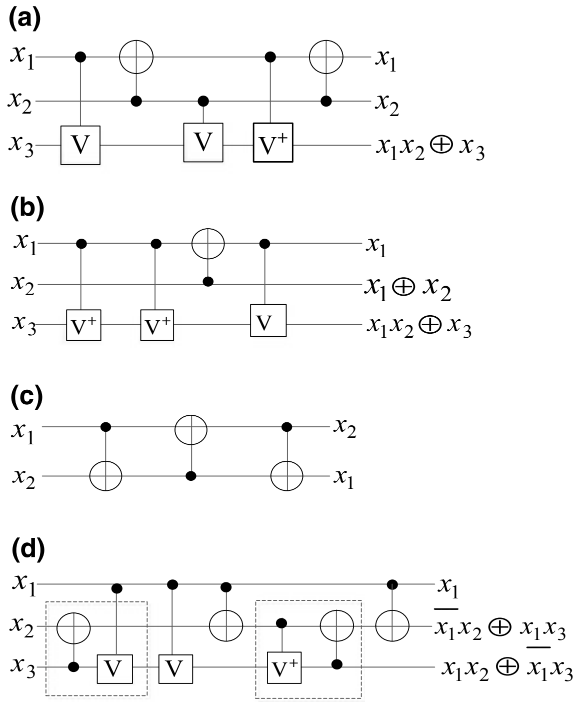


Fig. 3.10 **a** Toffoli gate quantum realization, **b** Peres gate quantum realization, **c** SWAP gate quantum realization, and **d** Fredkin gate quantum realization

1 × 1, and 2 × 2 gates, or synthesize the new gate using the well-known gates and add up QC of each known gate to result the total QC. The QC, size, and delay of well-known reversible and quantum gates are illustrated in Table 3.5.

Toffoli gate is one of the widely used universal gates in many literatures. Table 3.6 showcases quantum cost of a generalized Toffoli gate. The number of circuit lines and control lines is denoted by n and c, respectively.

Table 3.5 Common reversible, quantum gates and corresponding quantum cost, size, and delay

Name	Size	QC	Delay Δ
Toffoli	3 × 3	5	4
Fredkin	3 × 3	5	5
SWAP	2 × 2	3	2
Peres	3 × 3	4	3
CNOT	2 × 2	1	1
NOT	1 × 1	1	1
Controlled-V	1 × 1	1	1
Controlled-V+	1 × 1	1	1

Table 3.6 Quantum cost of generalized Toffoli gates

c	$(n-c+1) \geq$[a]	Cost
0	0	1
1	0	1
2	0	5
3	0	13
4	2	26
4	0	29
5	3	38
5	1	52
5	0	61
6	4	50
6	1	80
6	0	125
7	5	62
7	1	100
7	0	253
8	6	74
8	1	128
8	0	509
9	7	86
9	1	152
9	0	1021
>9	$c-2$	$12(c+1)-34$
>9	1	$24(c+1)-88$
>9	0	$2^{c+1}-3$

[a] Lines neither taken as control or target line

3.6.2 Gate Count (GC)

The number of reversible or quantum gates that are required to realize a circuit is known as the GC of that circuit. GC is used to compare the designs where all the gates are similar in size and type. For instance, reversible logic circuits are implemented using one Toffoli gate (QC = 5), and if there is a possibility that same circuit can be implemented by two primitive gates (QC = 2), then just looking at number of gates one cannot select the former one.

3.6.3 Ancilla Input, Garbage Output

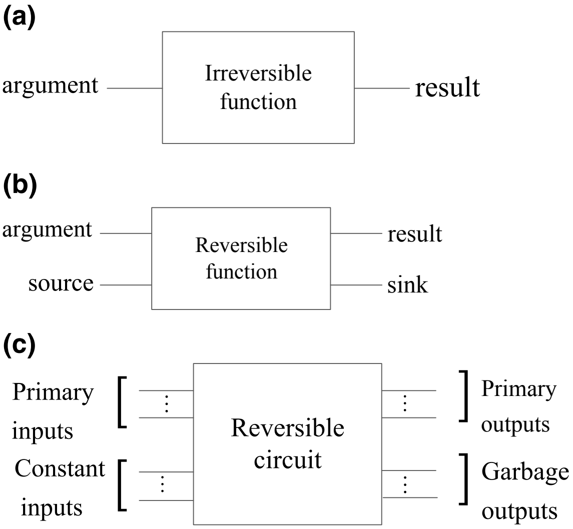
For n variable, there exist $2^n!$ unique reversible functions and maximum $2^{n^{2^n}}$ multiple output irreversible functions. In order to get reversible specification out of irreversible specification, additional input (source) and/or auxiliary output (sink) need to be added. The remaining inputs called argument, and the remaining outputs called result [6]. For example, AND gate in Fig. 3.11 can be inferred that to obtain desired specification, the source lines must be initialized with constant values (0, 1) also known as ancilla (constant) inputs. There are possibilities where values on sink line are independent of argument, thus cannot be utilized as constant inputs in the next level called the garbage lines (see Fig. 3.12).

The following formulas hold good for total outputs and inputs in a reversible circuit.

Fig. 3.11 Reversible AND gate implementation

c	x_1	x_2	y	g_1	g_2
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	0	1	1

Fig. 3.12 **a** An irreversible function, **b** reversible function having source (constant) input and sink (garbage) output, and **c** reversible circuit having inputs and outputs



Argument + auxiliary inputs = result + auxiliary outputs.
Argument + source = result + sink.
Primary i/p + constant = primary o/p + garbage.

For an irreversible function, if one output pattern repeats M number of times in function table, then $\lceil \log_2 M \rceil$ number of garbage bits need to be added to design a reversible logic circuit [14]. Addition of constant inputs is equally important for converting irreversible module to reversible module. More often, these constant values are taken as "*don't care*", which further used to optimize a reversible logic function. Similarly, addition of garbage outputs can be considered for efficient design, as information loss reduces with reductions in number of garbage outputs.

3.6.4 Delay

Delay measures logical depth of reversible logic circuits. The delay of each primitive gate (1 * 1, 2 * 2 gates) is considered a unit delay Δ. For forming a 3 * 3 or larger new gate, these primitive gates are utilized. Hence, effective delay for new gate is estimated by considering logical depth of individual 1 * 1 and 2 * 2 reversible units. Table 3.5 indicates the delay for other reversible or quantum gates with respect to the unit delay.

Example 1 Figure 3.13 shows a 2-bit binary adder, which has QC of 41. This circuit is composed of several gates in a cascade manner. Analysis shows that the delay in the critical path of this circuit is 32 and GC of 12. Figure 3.13 also depicts that to implement this reversible specification, we need one ancilla input and two garbage outputs.

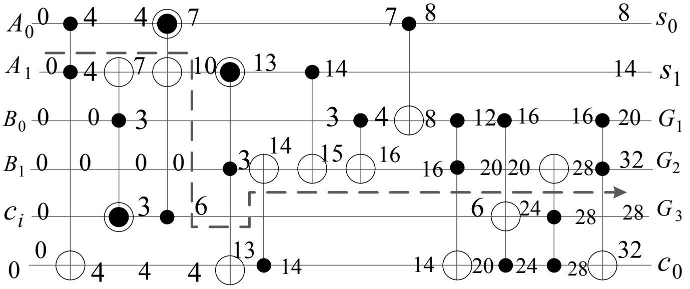

A_0 0 4 4 7 7 8 8 s_0
A_1 0 4 7 10 13 14 14 s_1
B_0 0 0 3 3 4 8 12 16 16 20 G_1
B_1 0 0 0 0 3 14 15 16 16 20 20 28 32 G_2
c_i 0 3 6 6 24 28 28 G_3
0 0 4 4 4 13 14 14 20 24 28 32 c_0

Fig. 3.13 Reversible realization of binary adder of size 2

3.7 Summary

This chapter discusses the theory of reversible and irreversible computing, corresponding logic gates, and circuits along with the performance metrics which are used for evaluation.

References

1. M.M. Mano, *Digital Design* (Prentice Hall, 2005)
2. R. Landauer, Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **5**(3), 183–191 (1961)
3. C.S. Lent, M. Liu, Y. Lu, Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling. Nanotechnology **17**(16), 4240 (2006)
4. A. Berut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, E. Lutz, Experimental verification of Landauer's principle linking information and thermodynamics. Nature **483**(7388), 187–189 (2012)
5. A.O. Orlov, C.S. Lent, C.C. Thorpe, G.P. Boechler, G.L. Snider, Experimental test of Landauers principle at the sub-KBT level. Jpn. J. Appl. Phys. **51**(6S), 06FE10 (2012)
6. T. Toffoli, *Reversible computing* (Springer, Berlin, 1980), pp. 632–644
7. E. Fredkin, T. Toffoli, Conservative logic. Int. J. Theor. Phys. **21**(3), 219–253 (1982)
8. A. Peres, Reversible logic and quantum computers. Phys. Rev. A **32**, 3266–3276 (1985)
9. R.P. Feynman, Quantum mechanical computers. Found. Phys. **16**(6), 507–531 (1986)
10. I.L.C.M.A. Nielsen, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000)
11. P.A.M. Dirac, A new notation for quantum mechanics. Math. Proc. Cambridge Philos. Soc. **35**(3), 416–418 (1939)
12. D. Maslov, D.M. Miller, Comparison of the cost metrics for reversible and quantum logic synthesis. IET Comput. Dig. Techn.
13. W.N.N. Hung, X. Song, G. Yang, J. Yang, M. Perkowski, Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **25**(9), 1652–1663 (2006)
14. D. Maslov, G.W. Dueck, Reversible cascades with minimal garbage. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **23**(11), 1497–1509 (2004)

Chapter 4
Design of Reversible Gates in QCA

In this chapter, we explore the QCA implementation of primitive reversible logic gates. We also consider the advantages of using proposed reversible structures to realize complex and efficient QCA circuits. For QCA layout of the proposed structures, we have employed QCADesigner, and corresponding simulation results are analyzed.

4.1 Fundamental Gates Using QCA

Reversible gates are fundamental blocks of reversible logic circuits. These gates result unique mapping between the input and the output sets allowing the number of inputs equal to the number of outputs. Figure 4.1 illustrates the block diagrams of primitive reversible gates.

Several investigations have been conducted to realize reversible logic circuits on QCA during last decade [6–11]. Authors in [6] have shown QCA architecture for various reversible gates using majority gates as the fundamental unit. It is observed that some of these gates are not suitable as the output cells are not highly polarized signals. In particular, the Peres and Fredkin gates' output *P* is found to be 5.80e−001 and 8.63e−001, respectively. This leads to 14% reduction in strength of output signal compared to input signal, which sets limit for the drivability of the designs. Mohammadi et al. [7] developed a QCA-based reversible circuit for Feynman, Toffoli, and Fredkin gates considering both rotated and regular cells. The presented structures are focused on majority gate structure, but the key issues in the designs are high complexity (in terms of delay and area) and need additional majority logic synthesis algorithm for an efficient QCA design. In other works [8–10], QCA implementable majority gate-based Feynman and Peres gates are explored. Their designs are less optimized and require further reduction in majority and inverter gates. Authors in [11] proposed optimal reversible gate structure in QCA using 5-input majority gates. However, these layouts have common drawbacks that require more constant inputs

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_4

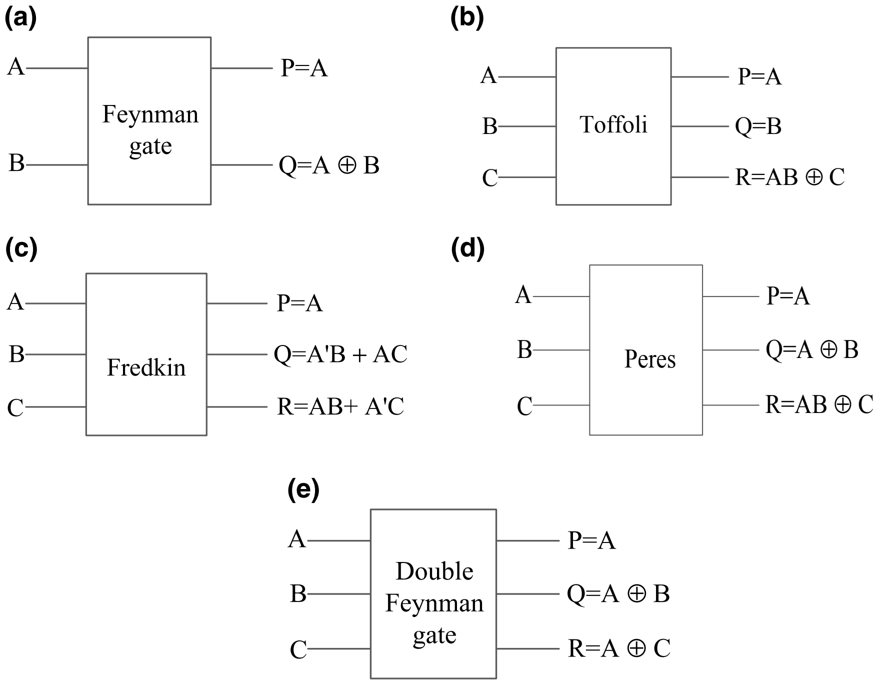


Fig. 4.1 Block diagram of reversible gates—**a** Feynman [1], **b** Toffoli [2], **c** Fredkin [3], **d** Peres [4], and **e** double Feynman [5]

(garbage/ancilla inputs). The designs presented by Sing et al. [11] cannot be realized without employing any wire crossover in cascade designs. In addition, outputs of Fredkin gate are not highly polarized signals. In particular, the output loses at least 11.4% of the input signal, which may affect signal integrity.

4.2 Novel QCA Structure for Reversible Logic Gates

Designs of different reversible gates using QCA have been examined in [6–11]. The fundamental unit in most primitive reversible gates is a XOR gate. It is observed that the complexity of the XOR gate not only affects the complexity but also the delay of the reversible gate layout. The conventional design of XOR requires three majority gates and two inverters. In particular, for given inputs A and B, the output function XOR2 is denoted by Eq. 4.1.

$$\text{XOR2} = A \oplus B = A'B' + AB \tag{4.1}$$

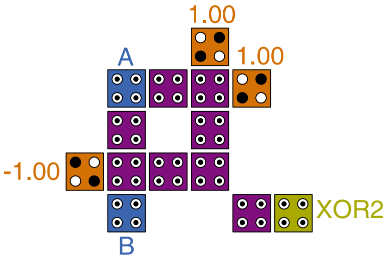


Fig. 4.2 Layout of XOR gate [12]

Contrary to existing designs that follow Eq. 4.1 for realization of reversible gates, the presented reversible structures utilize an optimized design for the 2-input XOR gate that uses explicit interactions between QCA cells, as shown in Fig. 4.2 [12].

4.2.1 Design of Feynman Gate

The logic diagram of the presented Feynman gate is depicted in Fig. 4.3a. This relates inputs (A, B) to outputs ($P = A$, $Q = A \oplus$ B). Figure 4.3b represents the corresponding QCA layout that involves one 2-input XOR gate. It is worth noting that output P connected to input A using array of cells to provide necessary wire

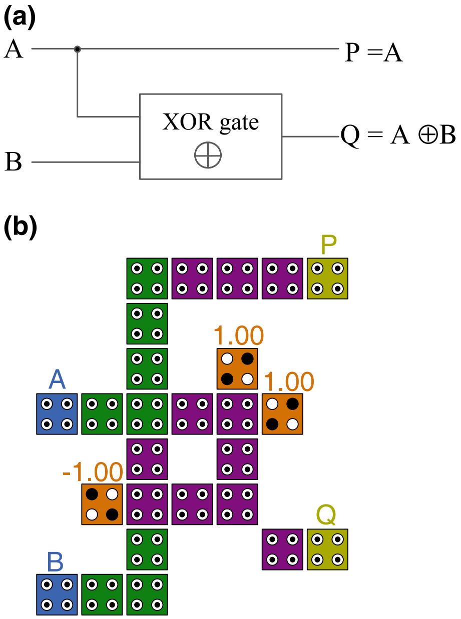


Fig. 4.3 **a** Feynman gate block diagram and **b** layout for Feynman gate

delay, whereas output Q requires the optimal XOR gate which has lower complexity than the existing XOR designs with 3-input majority gates.

4.2.2 Design of Double Feynman Gate

The schematic of the proposed Double Feynman gate is shown in Fig. 4.4a, which maps inputs (A, B, C) to outputs $(P = A, Q = A \oplus B, R = A \oplus C)$. The presented layout for the Double Feynman gate uses the novel XOR gate to produce an area-efficient design. Figure 4.4b shows the proposed layout for the gate with two 2-input

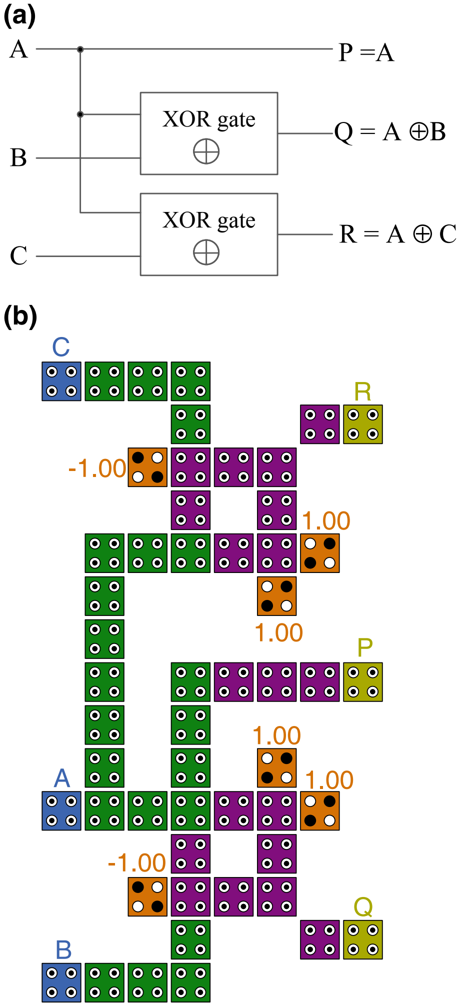


Fig. 4.4 **a** Double Feynman gate block diagram and **b** layout for double Feynman gate

XOR gates. Note the input A is passed to P using array of cells to provide necessary wire delay, whereas for the realization of outputs Q and R, two 2-input XOR gates are required.

4.2.3 Design of Toffoli Gate

The block diagram for the proposed Toffoli gate is shown in Fig. 4.5a, which relates inputs (A, B, C) to outputs ($P = A$, $Q = B$, $R = AB \oplus C$). An efficient QCADesigner layout for the gate is shown in Fig. 4.5b. It involves one 3-input majority gate and one 2-input XOR gate. The 3-input majority gate results an AND (setting third input as ‘0’) operation between inputs A and B. It can be observed that outputs P and Q are obtained directly from inputs A and B, respectively, using array of cells to provide necessary wire delay. The resulting intermediate output is given by

$$I_1 = MV(A, B, 0) = AB \tag{4.2}$$

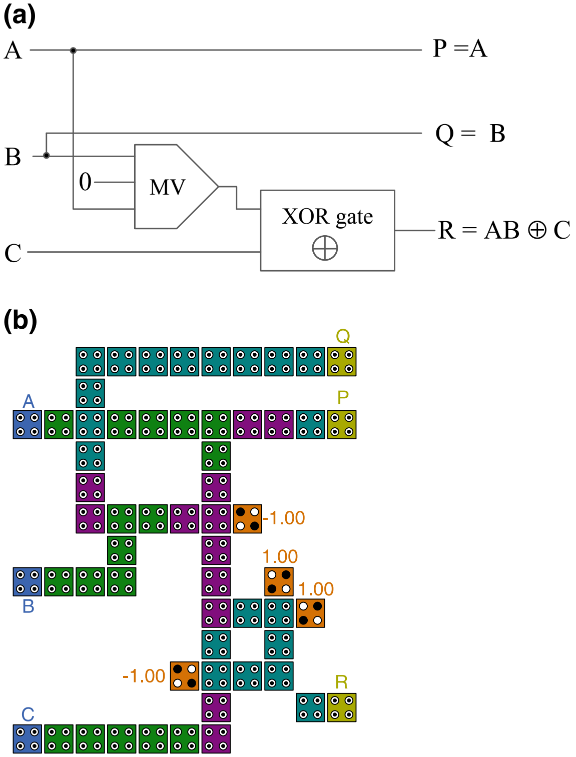


Fig. 4.5 **a** Toffoli gate block diagram and **b** layout for Toffoli gate

A 2-input XOR gate takes input C, and the intermediate result I_1 for computation of output R is given by

$$R = I_1 \text{ XOR } C \tag{4.3}$$

4.2.4 Design of Fredkin Gate

The schematic of the proposed Fredkin gate is depicted in Fig. 4.6a, which links inputs (A, B, C) to outputs ($P = A$, $Q = A'B + AC$, $R = AB + A'C$). For implementation, it uses an inverter, two 2-input XOR gates along with two 3-input majority gates. The QCADesigner layout of the presented Fredkin gate is shown in Fig. 4.6b. It can be observed that the layout involves proper wire crossing using clock zone-based coplanar crossover. Further, it is worth noting that output P connected to input A using array of cells to provide necessary wire delay, while outputs Q and R are realized using 2:1 multiplexer units.

The output Q is given as follows:

$$\begin{aligned} Q &= MV(A \oplus B, B, C) \\ &= MV(A'B + AB', B, C) \\ &= (A'B + AB')B + BC + C(A'B + AB') \\ &= A'B + A'BC + AB'C + BC = \Sigma(2, 3, 3, 5, 3, 7) \\ &= \Sigma(2, 3, 5, 7) = A'B + AC \end{aligned} \tag{4.4}$$

The output R is defined by

$$\begin{aligned} R &= MV(A' \oplus B, B, C) \\ &= MV(AB + A'B', B, C) \\ &= (AB + A'B')B + BC + C(AB + A'B') \\ &= AB + A'B'C + BC + ABC = \Sigma(1, 3, 6, 7) = AB + A'C \end{aligned} \tag{4.5}$$

4.2.5 Design of Peres Gate

The schematic of the proposed QCA Peres gate is depicted in Fig. 4.7a, which relates inputs (A, B, C) to outputs ($P = A$, $Q = A \oplus B$, $R = AB \oplus C$). The circuit consists

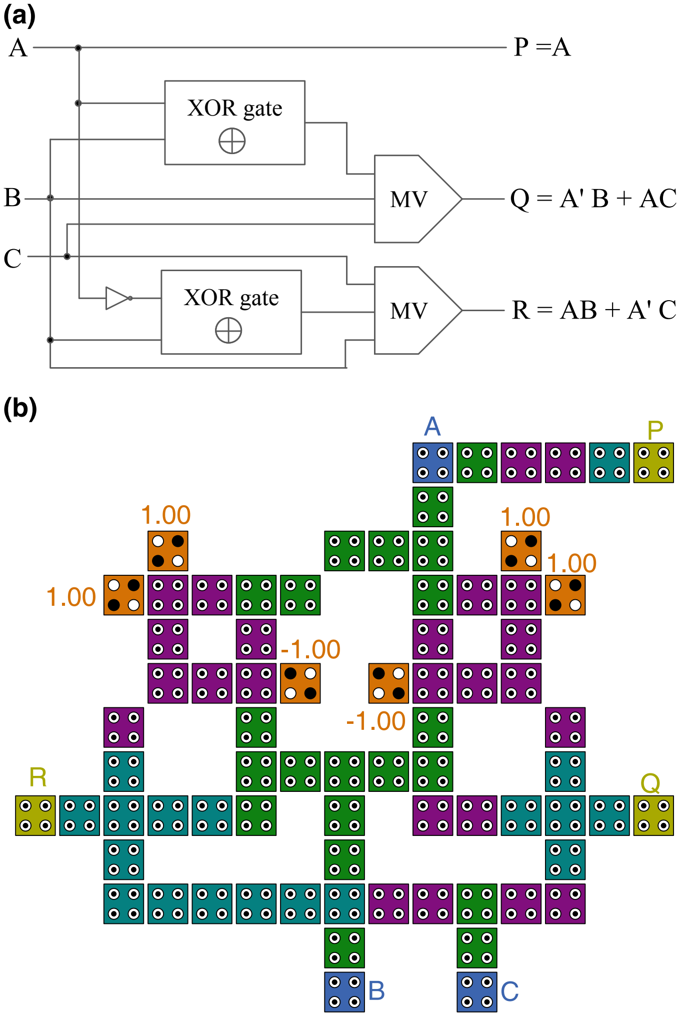


Fig. 4.6 **a** Fredkin gate block diagram of and **b** layout for Fredkin gate

of one 3-input majority gate and two 2-input XOR gates. The majority gate realizes AND gate (making one input to ‘0’), denoted by $AB = M(A, B, 0)$. Figure 4.7b shows the QCADesigner layout for the proposed Peres gate. It can be observed that the layout facilitates proper wire crossing using clock zone-based coplanar crossover.

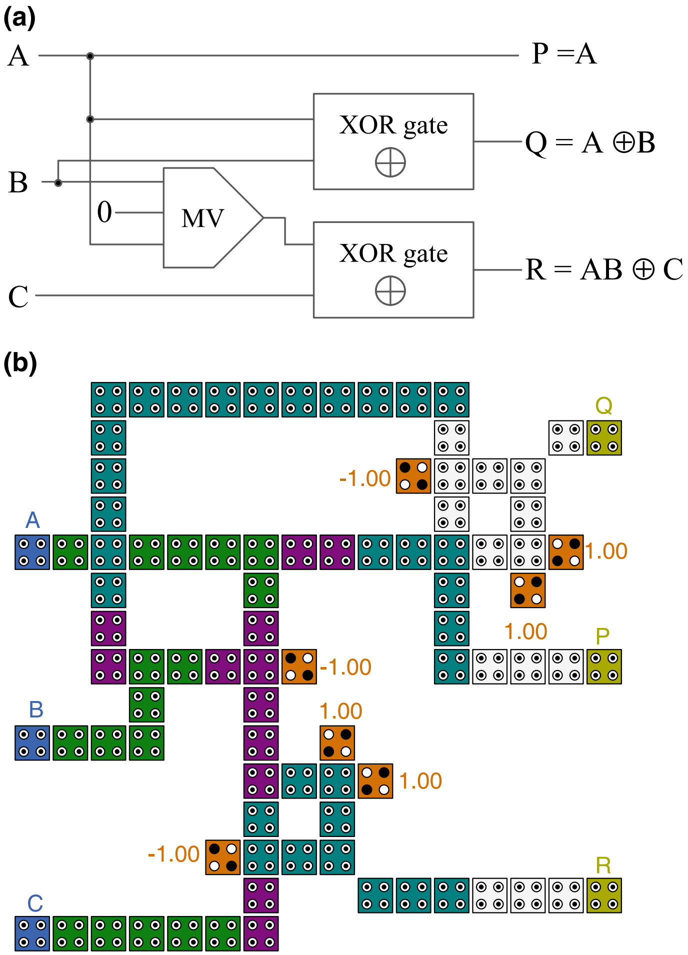
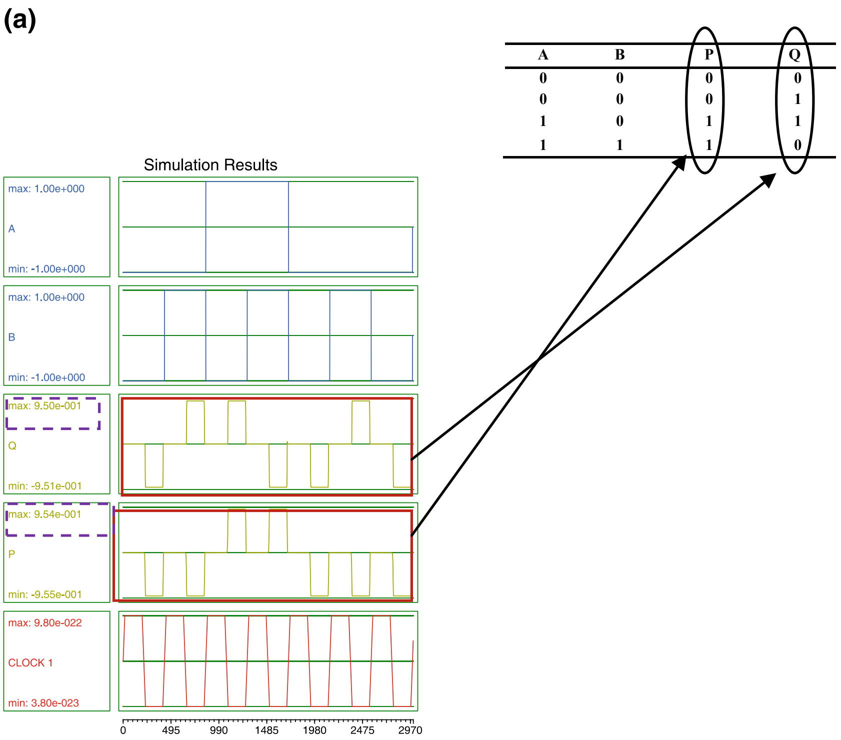


Fig. 4.7 **a** Peres gate block diagram and **b** layout for Peres gate

4.3 Simulation Results and Discussion

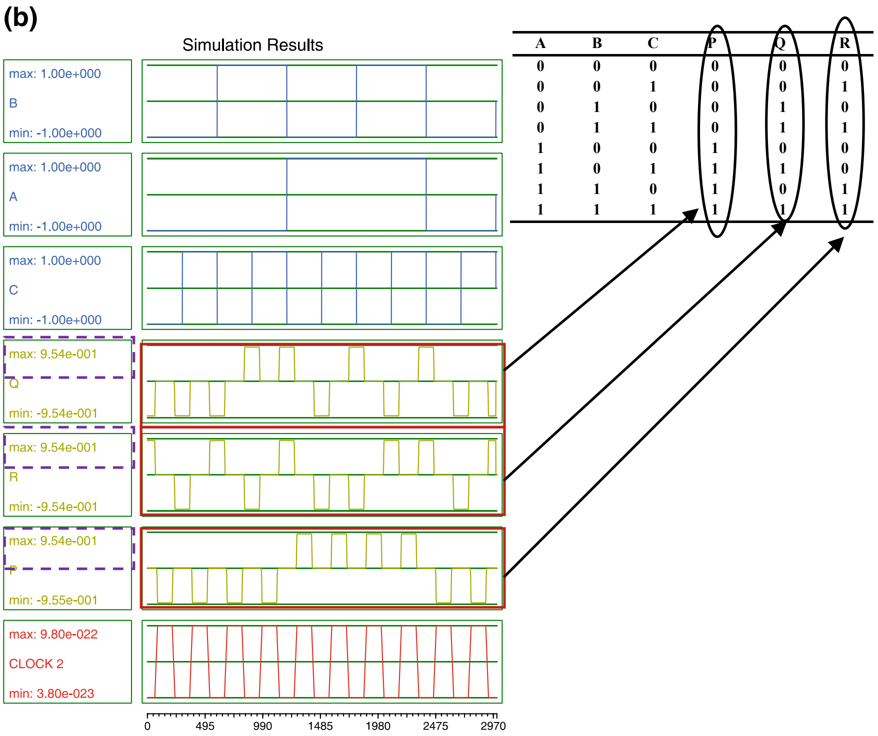
The QCA layouts of the proposed structures are done using QCADesigner [13], and coherence vector simulation engine is being considered to simulate the layouts using default settings. Figure 4.8a shows the simulation results for the proposed Feynman gate. It can be observed that the gate performs correctly for all possible input combinations . There is a delay of two clock phases for the gate to generate the correct outputs. The QCA realization of Feynman gate requires 26 cells and spans 0.03 μm^2 QCA area. Figure 4.8b illustrates the simulation waveforms for the proposed Fredkin gate. According to the waveform, the circuit performs correctly and produces correct outputs in three clock phases. The Fredkin design needs 68 cells for the layout and the spans over an area of 0.06 μm^2. The QCADesigner-based simulation waveform for the Toffoli gate is given in Fig. 4.9a. It shows the proper functioning of the gate, and correct outputs are resulted in three clock phases as indicated by red boxes. Corresponding QCA implementation takes 59 cells and the area equal to 0.034 μm^2.

Figure 4.9b represents the simulation waveforms of the Peres gate. It is noted that the correct outputs are generated after a delay of three clock phases. The gate is synthesized using 88 cells in an area of 0.097 μm^2. Figure 4.10 depicts the simulation waveform of the Double Feynman gate. This simulation result indicates that the circuit works correctly for all input vectors, and the outputs are generated after two clock phases identified by red boxes. For implementation, it needs 53 cells in an



A	B	P	Q
0	0	0	0
0	0	0	1
1	0	1	1
1	1	1	0

Fig. 4.8 Result of simulating the proposed gate in QCADesigner **a** Feynman gate and **b** Fredkin gate



A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Fig. 4.8 (continued)

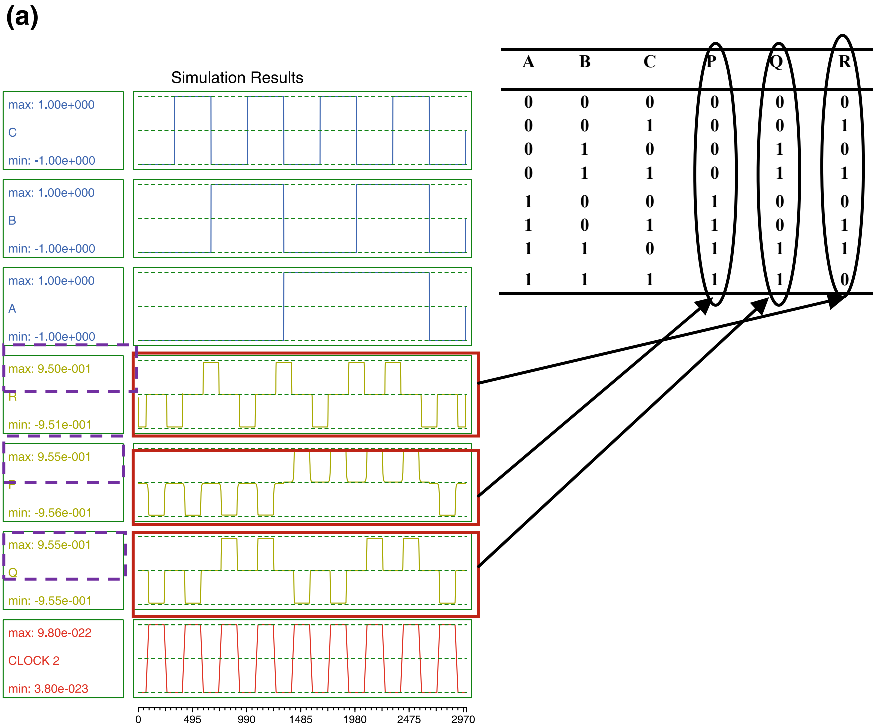
area of 0.058 μm^2. From the results, we can see that the proposed gates outputs are highly polarized signals represented by the purple boxes. Consequently, the proposed reversible structures establish signal integrity with a high drivability attribute and can be used as building blocks in complex QCA-based designs.

The post-implementation of the proposed structures is summarized in Tables 4.1, 4.2, 4.3, 4.4 and 4.5. Various QCA parameters such as total number of cells, types of crossover, area, and delay have been considered for comparison with existing designs. From Table 4.1, it is observed that the proposed Feynman gate achieves 18.7, 33% reduction in cell count and delay when compared with the best design in [11]. For implementation of Toffoli and Fredkin gates, we consider coplanar wire crossing to encourage cascade designs unlike the previous designs [6, 11, 14]. From Tables 4.2 and 4.3, we can see that designs reported in [11] considered being best among all existing design, but the designs require additional cells and area for any types of wire crossing. It is worth noting that the proposed Toffoli and Fredkin gates are more suitable for cascade design, and performance metrics are close to the best design in [11]. In particular, the Fredkin and Toffoli gates require delay of 0.75 clocks, which is equal to the delay of the designs presented in [11]. Table 4.4

demonstrates comparison of proposed Peres gate and the previous designs in [6, 10]. The proposed Peres gate shows 9.3 and 25% reduction in cell count and delay than the designs in [10]. Table 4.5 reveals that the proposed Double Feynman structure achieves performances close to the structure in [14].

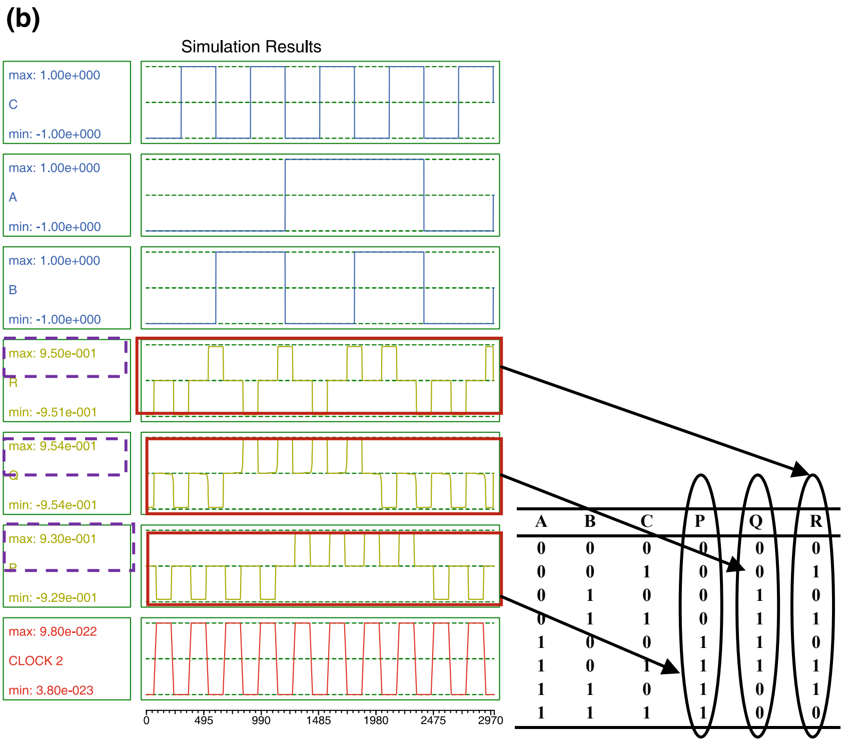
4.3.1 Summary

In this chapter, we have examined QCA realization of various reversible logic gates. We analyzed the performance of the proposed gates and the existing gates using conventional parameters. Results show that the proposed structures outperform previous reversible gate designs and thus suitable for application toward complex nanoscale architectures in QCA.



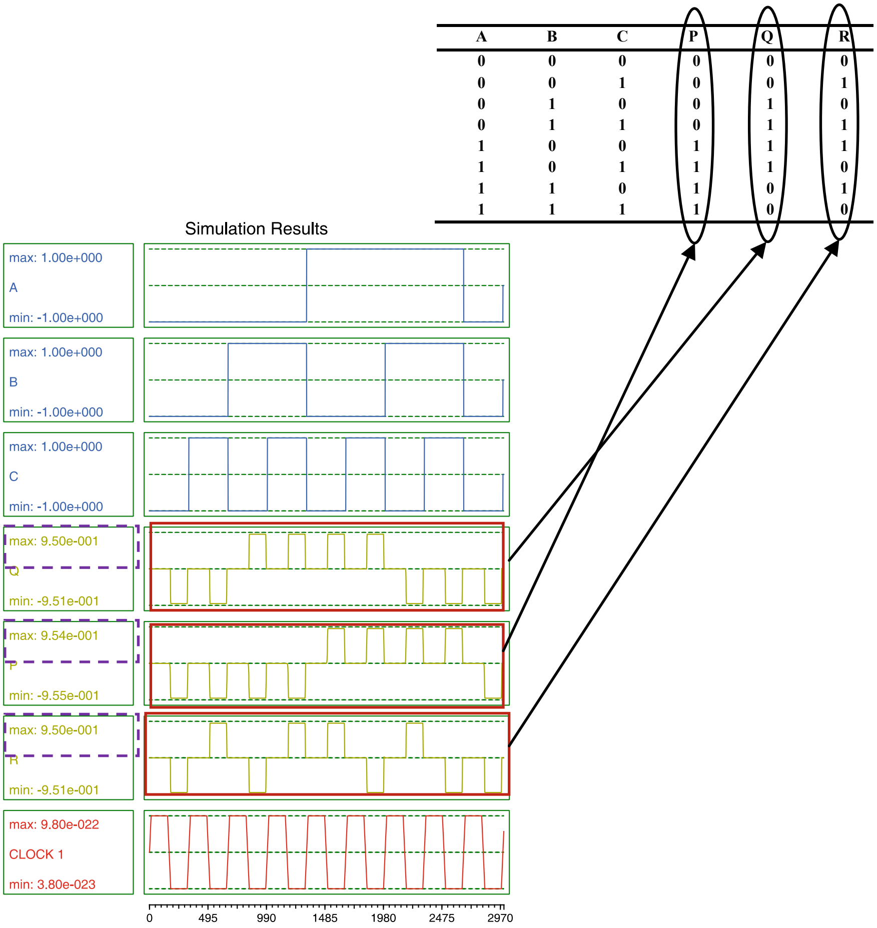
A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Fig. 4.9 Result of simulating the proposed gate in QCADesigner **a** Toffoli gate and **b** Peres gate



A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Fig. 4.9 (continued)



A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	0	0

Fig. 4.10 Result of simulating the proposed Double Feynman gate in QCADesigner

Table 4.1 Comparison of proposed Feynman gate with existing works

Feynman gate design	Cell count	Area (μm^2)	Delay (clocks)	Types of wire crossing in cascade design
[7]	78	0.09	1	Coplanar
[8]	54	0.038	0.75	Multilayer
[6]	53	0.07	0.75	–
[9]	37	0.023	0.75	–
[11]	32	0.03	0.75	Not required
[14]	34	0.036	0.5	–
Proposed	26	0.03	0.5	Not required

Table 4.2 Comparison of proposed Toffoli gates with existing works

Toffoli gate design	Cell count	Area (μm^2)	Delay (clocks)	Types of wire crossing in cascade design
[7]	170	0.23	1	Coplanar
[6]	57	0.06	0.75	–
[11]	45	0.045	0.75	–
Proposed	59	0.06	0.75	Coplanar

Table 4.3 Comparison of proposed Fredkin gate with existing works

Fredkin gate design	Cell count	Area (μm^2)	Delay (clock cycles)	Types of wire crossing in cascade design
[7]	178	0.21	1	Coplanar
[14]	100	0.092	0.75	–
[6]	97	0.10	0.75	–
[11]	73	0.065	0.75	–
Proposed	75	0.08	0.75	Coplanar

Table 4.4 Comparison of proposed Peres gate with existing works

Peres gate design	Cell count	Area (μm^2)	Delay (clocks)	Types of wire crossing in cascade design
[6]	117	0.18	0.75	–
[10]	97	0.075	1	Multilayer
Proposed	88	0.097	0.75	Coplanar

Table 4.5 Comparison of proposed Double Feynman gate with existing works

Double Feynman gate design	Cell count	Area (μm^2)	Delay (clocks)	Types of wire crossing in cascade design
[6]	93	0.19	0.75	Coplanar/multilayer
[14]	51	0.06	0.5	–
Proposed	53	0.058	0.5	Not required

‘–’ not available

References

1. R.P. Feynman, Quantum mechanical computers. Opt. News **11**(2), 11–20 (1985)
2. Toffoli, T.: Reversible computing, Tech memo - MIT/LCS/TM-151, MITLab for Comp. Sci. 1980
3. Fredkin, E., Toffoli, T.: *Conservative Logic in Collision-Based Computing* (Springer, Berlin, 2002), pp. 47–81
4. A. Peres, Reversible logic and quantum computers. Phys. Rev. A Gen. Phys. **32**(6), 3266–3276 (1985)
5. Parhami, B.: Fault-tolerant reversible circuits. In: Fortieth Asilomar Conference on Signals, Systems and Computers, ACSSC'06, pp. 1726–1729 (2006)
6. M. Abdullah-Al-shafi, M.S. Islam, A.N. Bahar, A review on reversible logic gates and it's QCA implementation. Int. J. Comput. Appl. **128**(2), 27–34 (2015)
7. Z. Mohammadi, M. Mohammadi, Implementing a one-bit reversible full adder using quantum-dot cellular automata. Quantum Inf. Process. **13**, 2127–2147 (2014)
8. J.C. Das, D. De, Reversible binary to grey and grey to binary code converter using QCA. IETE J. Res. **61**(3), 223–229 (2015)
9. J.C. Das, D. De, Reversible comparator design using quantum dot-cellular automata. IETE J. Res. **62**(3), 323–330 (2016)
10. J.C. Das, D. De, Novel low power reversible binary incrementer design using quantum-dot cellular automata. Microprocess. Microsyst. **42**, 10–23 (2016)
11. G. Singh, R.K. Sarin, B. Raj, Design and analysis of area efficient QCA based reversible logic gates. Microprocess. Microsyst. **52**, 59–68 (2017)
12. A.M. Chabi, A. Roohi, H. Khademolhosseini, S. Sheikhfaal, Towards ultra-efficient QCA reversible circuits. Microprocess. Microsyst. **49**, 127–138 (2017)
13. K. Walus, T. Dysart, G. Jullien, R. Budiman, QCA designer: a rapid design and simulation tool for quantum-dot cellular automata. IEEE Trans. Nanotechnol. **3**(1), 26–29 (2004)
14. A.N. Bahar, S. Waheed, M.A. Habib, A novel presentation of reversible logic gate in Quantum-dot Cellular Automata (QCA), in *2014 International Conference on Electrical Engineering and Information & Communication Technology*, Dhaka (2014), pp. 1–6

Chapter 5
Designs of Adder Circuit in QCA

In this chapter, we discuss two different designs of QCA Ripple Carry Adders by customizing the fundamental block, i.e., the full adder circuit. We start the discussion with designing RCA using 5-input majority gate-based full adder. Further, designs are evolved considering a compact form full adder that relied on a novel XOR structure. Parts of the analysis presented in this chapter have been discussed in [1, 2].

5.1 Introduction

Adders are the rudimental components of an arithmetic circuit in any computer architecture. So studies of optimal multibit adders have received great deal of attention in QCA. Among all adders, Ripple Carry Adder is the simplest form of adder with more computational time. Other improved forms of RCA are CFA (carry-flow adder) and CSA (carry-skip adder) reducing delay factor. A CLA (carry-lookahead adder) allows faster addition but at the cost of higher complexity. Similarly, other fast adders derived from CLA are Brent–Kung adder (BKA) and Kogge–Stone adder (KSA); use prefix operation.

5.2 Adder Designs

Several designs of QCA adders have been reported by the researchers, which include both single-layer and multilayer designs [2–8]. Such designs employ 3-input majority voter (MV3) gate and inverters (INV) as the basic units. The first coplanar QCA full adder (FA) demonstrated in [3] named as Tougaw adder. For implementation, this design incorporates five MV3 gates, three INVs, and nine crossovers. RCA can be designed by cascade of 1-bit Tougaw adders. Wang et al. [4] proposed a simplified version of 1-bit FA using a reduction in number of majority gates. Wang adder allows

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_5

three MV3 gates, two INVs, and six coplanar wire crossovers. An efficient design shown in Hanninen and Takala [5] was mainly an improved version of Wang adder. This robust design achieves significant reduction in number of crossovers. The 1-bit Takala adder contains three MV3 gates, two INVs, and three wire crossovers. Zhang et al. [6] presented a multilayer implementation of Wang adders. This revised 1-bit adder is formed by three MV3 gates and two INVs, but the number of crossovers and delay is reduced to three and one clock cycles, respectively. In spite of reduction in design complexity, the multilayer crossover imposes design constraint due to fabrication difficulty. Moreover, coplanar and multilayer crossovers carry different cost. RCA architectures realized by various topologies for 1-bit FA were analyzed, and it is inferred that one-bit addition is performed within the four clock phases. A delay optimized QCA CFA is demonstrated in [7] which contributes only one clocking zone delay per bit. Apart from lower computation delay, a 1-bit Cho adder is associated with two multilayer crossovers. Several parallel-prefix adders are discussed in [8]. Among these, the BKA provides most optimal delay and area as compared to other adders, particularly for operands of large size. In addition, two other faster adders were also proposed, namely CLA and CFA, by the same authors. Recent works in [9–11] showcase different adders incorporating 5-input majority gates. Results indicate significant improvement in area occupation and latency over some of adders mentioned above.

5.3 Full Adder Design with 5-Input Majority Gate

Existing 5-input majority gate designs have been studied in this section. Analyses are being carried out to find the best optimal and power-efficient design for 5-input majority voter gate (MV5), which can be used as an essential block of a full adder.

We have examined a novel adder circuit to evaluate the suitability of the most optimal MV5 gate. Figure 5.1 presents gate-level illustration of a full adder using 3-input majority voter (MV3), MV5 gates, and inverters. Given three inputs A, B,

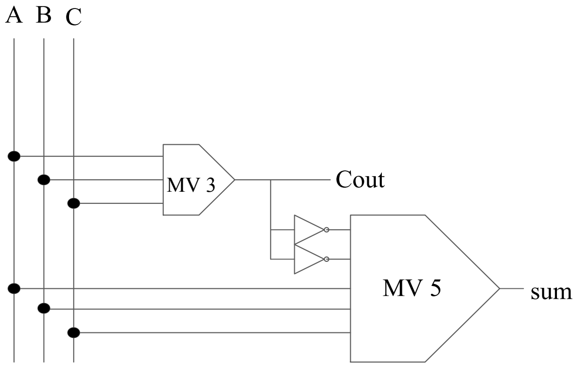


Fig. 5.1 Gate-level diagram for full adder

and C for a MV3 gate, Output Carry is presented by:

$$C_{\text{out}} = MV3(A, B, C) \tag{5.1}$$

The Sum for the full adder employs a MV5 gate and is expressed using Eq. 5.2.

$$\text{Sum} = MV5\left(\bar{C}_{\text{out}}, \bar{C}_{\text{out}}, C_{\text{in}}, A, B\right) \tag{5.2}$$

5.3.1 5-Input Majority Gates

A 5-input majority gate is basic building block in QCA which relies on majority function that results 1 only if 3 or more of its inputs are 1. Its concise truth table is depicted in Table 5.1 (taking summation of inputs), where output is shown ‘1’ for input combination which contains at least three ‘1’s. The Boolean expression is governed by Eq. 5.3:

$$\begin{aligned} MV5(A, B, C, D, E) = {} & ABC + ABD + ABE + ACD + ACE \\ & + ADE + BCD + BCE + BDE + CDE \end{aligned} \tag{5.3}$$

5.3.1.1 Structural Analysis

A novel symmetric structure for MV5 gate is reported in [1]. Its QCA cells’ arrangement is shown in Fig. 5.2. For correct operation, this gate needs ten cells—six input and output cells along with four middle cells. It is observed that the presented layout is utilizing only single layer, which turns out more flexible in terms of cell accessibility. Further analysis reveals that one of the five inputs and output can be relocated in different ways, making the design more dynamic and flexible. Note that unlike some of the existing designs, the input and output cells are not surrounded by other cells, so input cells are easily connected in cascade designs.

Table 5.1 Concise truth table of MV5

SUM (*A, B, C, D, E*)	Majority (*A, B, C, D, E*)
0	0
1	0
2	0
3	1
4	1
5	1

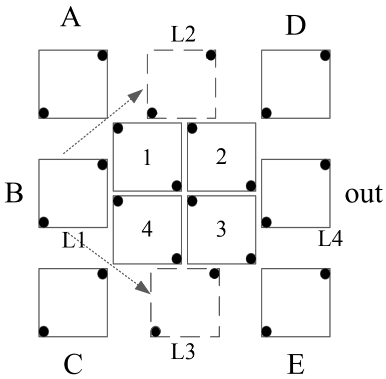


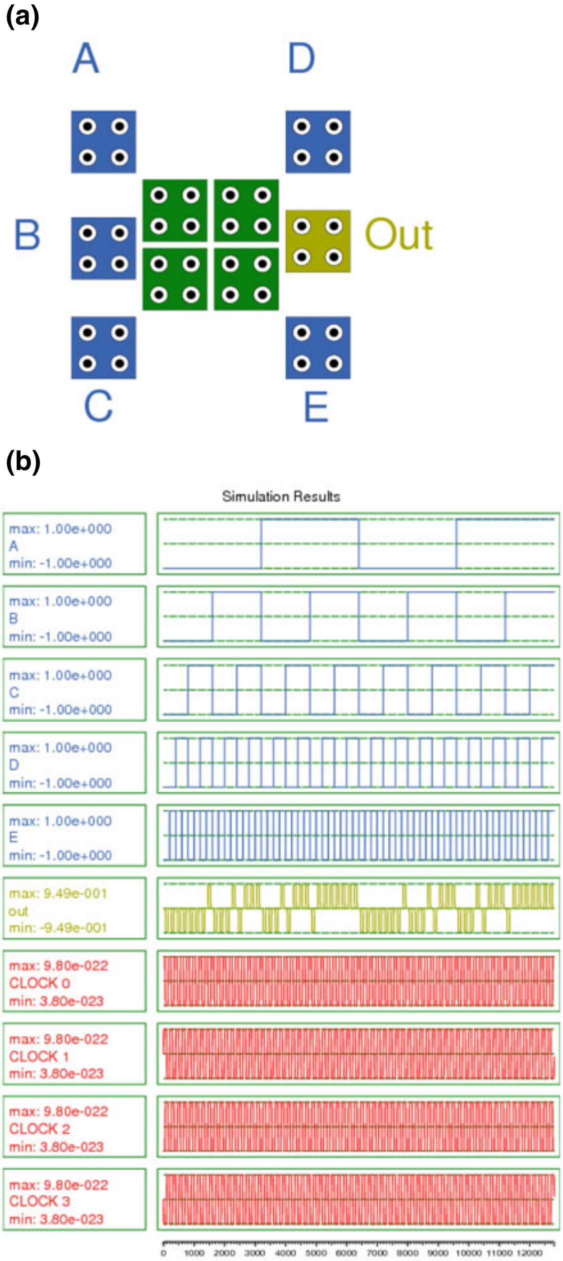
Fig. 5.2 Symmetric structure of MV5 [1]

QCA implementation for the MV5 is depicted in Fig. 5.3a. Figure 5.3b demonstrates the precise operation of the presented MV5 gate with input and output waveforms. Results show that presented circuit is 0.25 clock cycle to generate correct outputs while achieving better output polarization. The MV5 gate can be customized to a 3-input AND or a 3-input OR. By setting two inputs of MV5 gate to logic-0 leads to AND gate while logic-1 leads to OR gate (Fig. 5.4). Simulation results of the AND and OR gate are given in Fig. 5.5a, b. Table 5.2 compares several designs of 5-input majority gates considering common QCA metrics. The most commonly discussed metrics in the literature are a number of QCA cells, consumed area, polarization, and delay. According to Table 5.2, the presented design [1] is also good that achieved equal performance with respect to existing designs. Although the structures in [9] and [12] have better QCA area utilization, these designs are not suitable for coplanar design as some of the cells limit accessibility. Further, the single-layer design in [1] offers better accessibility to all input and output cells. From Table 5.2, we can figure out that the novel structure in [1] has lower cells and area requirement when compared to prior designs [10, 13–18].

5.3.1.2 Power Analysis

To calculate power dissipation of the designs, QCAPro has been used as a power estimator tool [19]. Estimation of power dissipation is carried out taking three tunneling energy values: 0.5 E_k, 1.0 E_k, and 1.5 E_k at 2 K temperature. Besides, the thermal layouts of different MV5 gates have been shown in Fig. 5.6 for estimation of energy dissipation at tunneling energy of 0.5 $\mathrm{E_k}$. Here cells those dissipate more average energy dissipation are indicated by dark color. Table 5.3 depicts energy dissipation results out of different designs [9, 10, 12–15, 18]. It can be shown that design in [1] exhibits lower energy dissipation when compared with previous designs. For ease of understanding, graphs showing average leakage energy, average switching energy, and total energy dissipation are demonstrated in Figs. 5.7, 5.8, and 5.9. This comparative analysis indicates that the MV5 by [1] offers better energy-efficient layout

Fig. 5.3 **a** QCA layout for MV5 [1] and **b** simulation waveforms



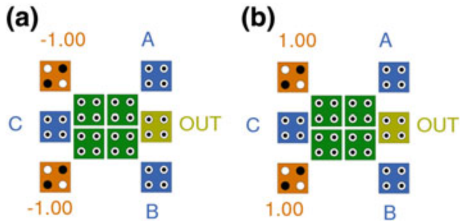


Fig. 5.4 QCA layouts **a** 3-input AND logic and **b** 3-input OR logic

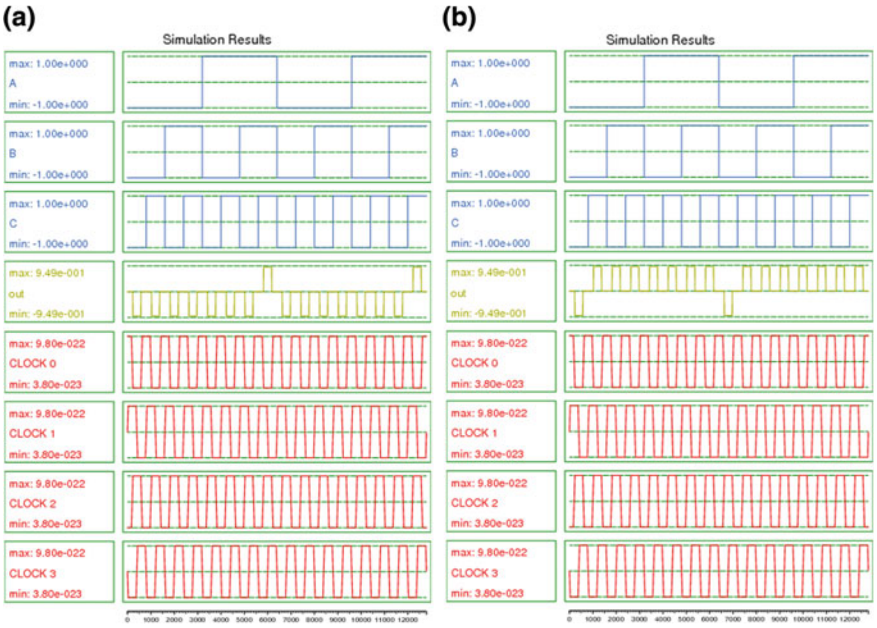


Fig. 5.5 QCADesigner simulations **a** for AND logic and **b** for OR logic

as compared to the best designs. This leads to realize ultra-low-power and complex QCA structures.

5.3.2 Full Adder Design

The full adder design described in [1] exploits the simplest module for a full adder that composes one MV3, one MV5 gate, and two inverters, as drawn in Fig. 5.1. The layout for the FA is redrawn in Fig. 5.10 using coplanar QCA technology. The Sum output computation needs four clock phases that include two levels of majority gates. This FA layout exploits coplanar crossing of wire with non-adjacent clock

Table 5.2 Comparison of different 5-input majority gates

5-input majority gate designs	Cell count	Polarization (e^{-001})	QCA area (nm^2)	Accessibility to the input and output cells using single layer
Angizi et al. [13]	23	9.52	24,564	Yes
Hashemi et al. [14]	20	–	19,044	Yes
Akeela et al. [15]	18	9.53	16,284	Yes
Hashemi et al. [16]	18	9.5	16,284	Yes
	17	9.5	18,644	Yes
Roohi et al. [10]	13	8.24	9604	Yes
Sen et al. [17]	13	9.54	9604	No
Sheikhfaal et al. [18]	11	9.48	9604	Yes
Navi et al. [12]	10	9.96	4524	Output cell is surrounded by the other cells; no single-layer accessibility to the output cell needs multilayer layout
Navi et al. [9]	10	9.5	7644	No
Sasamal et al. [1]	10	9.49	9604	Yes

zones, i.e., wires with 180° phase changes. Same has been presented in Fig. 5.10 using boxes. The carry output exhibited a delay of three clock phases and employs one rotated MV3 gate. QCA realization needs 55 cells with active area of 0.04 μm^2. Such module achieves superior results compared to other designs and can be used for more basic arithmetic logic like RCA to reduce the worst-case delay.

This QCA structure follows a generic strategy that counts lower clock phases for the coplanar wire crossing. It must be noted that the largest clock zone dimension comprises thirteen cells and has at least two cells. The simulation result for full adder is shown in Fig. 5.11. Results indicate polarization values of two outputs sum and carry as 9.540e−001, 9.520e−001, respectively. There is degradation in strength of polarization of 4.6%, within the allowed noise margin for inputs A, B, and C_{in} of polarization 1.00e+000 to retain signal integrity. Further, it is inferred that the first correct output is generated for both the outputs after four phases maximum denoted by black boxes.

5.4 Ripple Carry Adder Design

Let A_{i}, B_{i}, and C_{i} be inputs to a full adder. Then, the computation for *carry* (C_{i+1}) and sum (S_{i}) outputs is given by the following equations:

$$C_{i+1} = MV3(A_i, B_i, C_i) = A_i \cdot B_i + B_i \cdot C_i + C_i \cdot A_i \tag{5.4}$$

$$S_i = MV5\left(\bar{C}_{i+1}, \overline{C_{i+1}}, C_i, A_i, B_i\right) \tag{5.5}$$

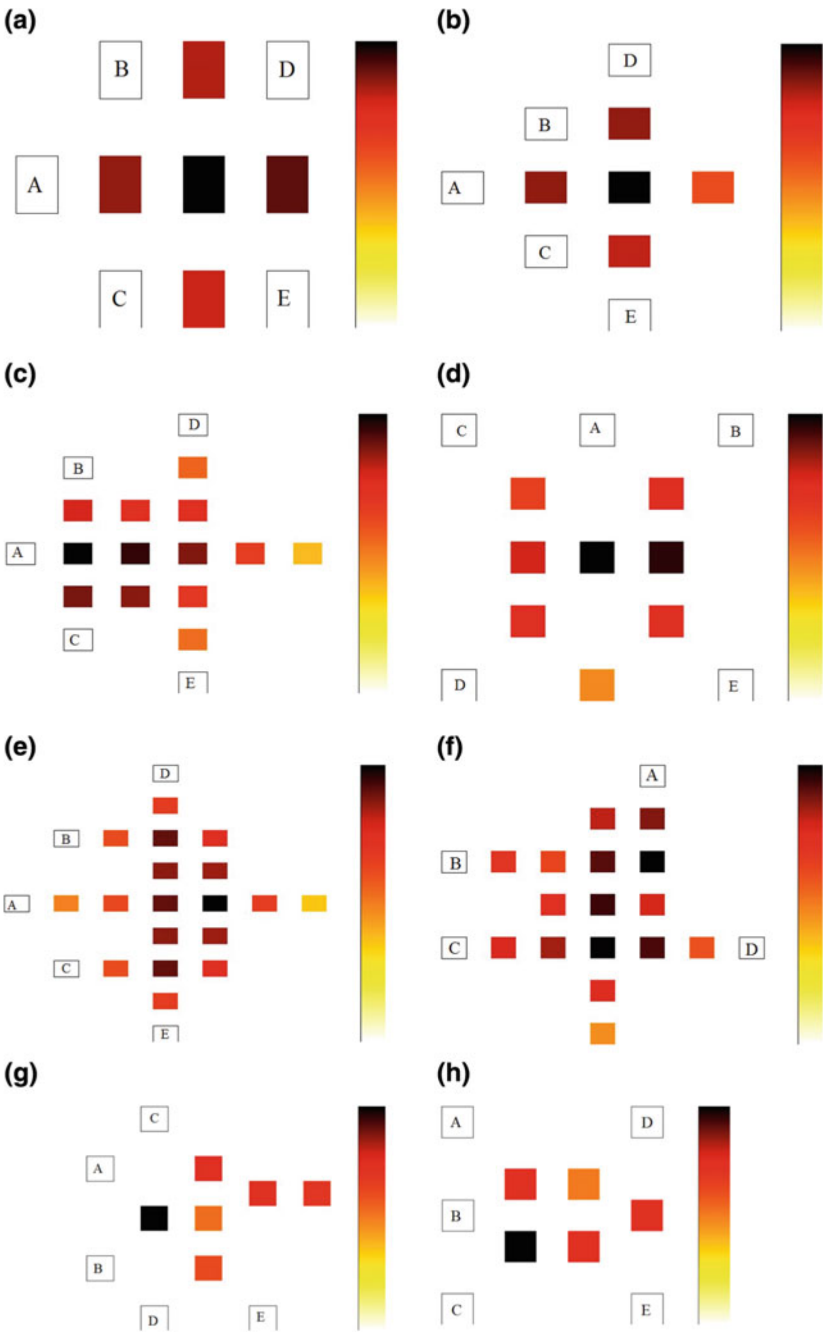


Fig. 5.6 Power dissipation maps for 5-input majority gates at 2 K temperature and tunneling energy of 0.5 E_k **a** design in [12], **b** design in [9], **c** design in [15], **d** design in [10], **e** design in [13], **f** design in [14], **g** design in [18], and **h** design in [1]

Table 5.3 Power consumption comparative analysis

Designs	Avg. leakage energy dissipation (meV)			Avg. switching energy dissipation (meV)			Total energy dissipation (meV)		
	0.5 E_k	1 E_k	1.5 E_k	0.5 E_k	1 E_k	1.5 E_k	0.5 E_k	1 E_k	1.5 E_k
Previous design [13]	4.44	14.25	26.61	45.51	41.59	37.29	49.96	55.84	63.90
Previous design [14]	4.41	13.55	24.73	31.24	28.31	25.21	35.66	41.85	49.94
Previous design [15]	3.44	10.67	19.52	32.66	29.89	27.01	36.1	40.56	46.53
Previous design [9]	1.28	4.14	7.69	11.53	10.37	9.16	12.81	14.51	16.85
Previous design [10]	3.38	8.95	15.03	9.23	7.7	6.41	12.61	16.65	21.44
Previous design [12]	1.35	4.25	7.8	10.94	9.84	8.7	12.29	14.09	16.5
Previous design [18]	2.99	7.73	12.35	3.69	2.77	2.15	6.68	10.5	14.5
[1] (Fig. 5.6h)	2.00	5.53	9.41	5.9	4.80	3.90	7.90	10.34	13.31

Fig. 5.7 Average leakage energy dissipation for different MV5 gates at different tunneling energies ($T = 2.0$ K)

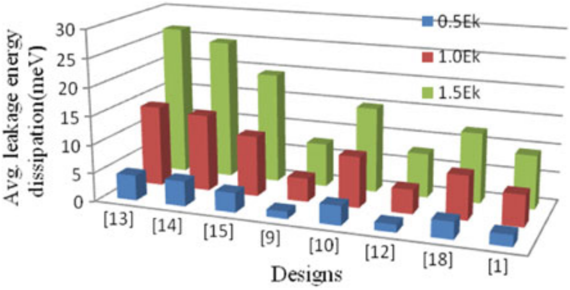


Fig. 5.8 Average switching energy dissipation for different MV5 gates at different tunneling energies ($T = 2.0$ K)

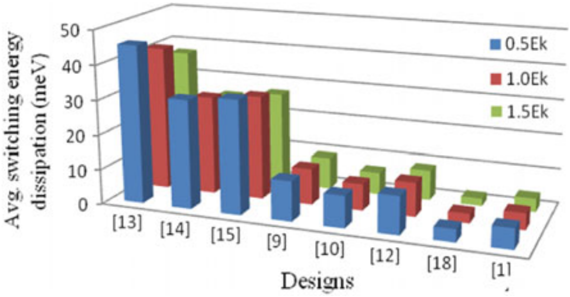
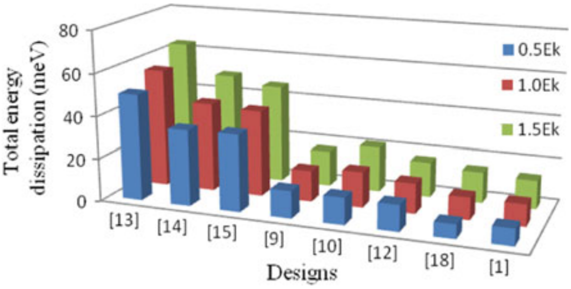


Fig. 5.9 Total energy dissipation for different MV5 gates at different tunneling energies ($T = 2.0$ K)



A n-bit RCA is realized through a number of cascaded n FAs. This means a carry path is formed out of the FA units as presented in Fig. 5.12a. Consider a QCA structure as reported in [1], where FA outputs the carry after three clock phases. That contributes total delay of $((n/2) + 0.5)$ for n-bit adder. Note this expression includes extra one clock phase required to generate final Sum output. One can notice that correctly designed RCA depends on addition of proper delay to the next stage operand and the Sum output. So once the subsequent stage produces the carry output, the next stage is ready to accept the inputs. The presented n-bit RCA is used to add n-bits in parallel once initial latency $((n/2) + 0.5)$ is over. One 4-bit layout incorporates the novel FA layout in [1] is shown in Fig. 5.12b. For implementation, this design needs 311 cells and outputted final results after clock phases of 10 cycles. For design, this adder introduces four rotated MV3 and MV5 gates, and eight inverters. To make

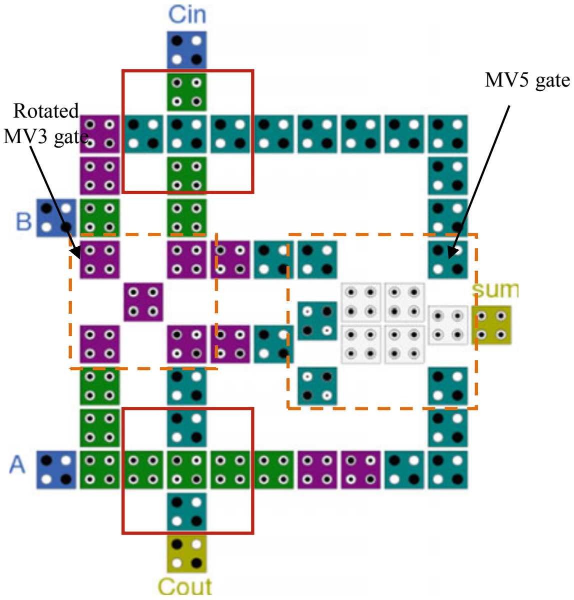


Fig. 5.10 QCA layout for FA

more reliable model for *n*-bit RCA, authors have considered maximum 13 cells along with QCA zone for a kink-free operation.

In order to verify the presented structures, exhaustive testing is done by considering all possible sequences of the input vectors. Figure 5.13 shows the correct behavior of the proposed 4-bit RCA using the input/output values. Herein, the output is obtained after 10 clock phases delay. The two, three, four, and eight-bit designs were undergone exhaustive vector set for simulation, and larger designs were analyzed using various operand sizes.

5.5 Results and Discussion

This section contains a brief analysis of various structures of adders in the literature followed by power analysis of FA modules. Table 5.4 illustrates different performance figures of FA designs. This indicates that FA layout in [1] offers excellent result in terms of QCA metrics as compared to most existing designs, except design in [10]. Particularly, the adder examined here [1] achieves superior performance with all coplanar designs [4, 5, 11, 14, 20], while overall delay is same as of [4, 6, 8, 10, 20]. It can be also noted that this coplanar design exhibits reduction in cells and area compared with [6–9, 12, 21–23] and close to [10, 24] those follow multilayer layout.

Next, we have evaluated performance of *n*-bit RCA which exists in literature by using conventional QCA parameters. It is observed that the architecture in [1] has achieved better solution for multibit adder with respect to delay, area, and cell

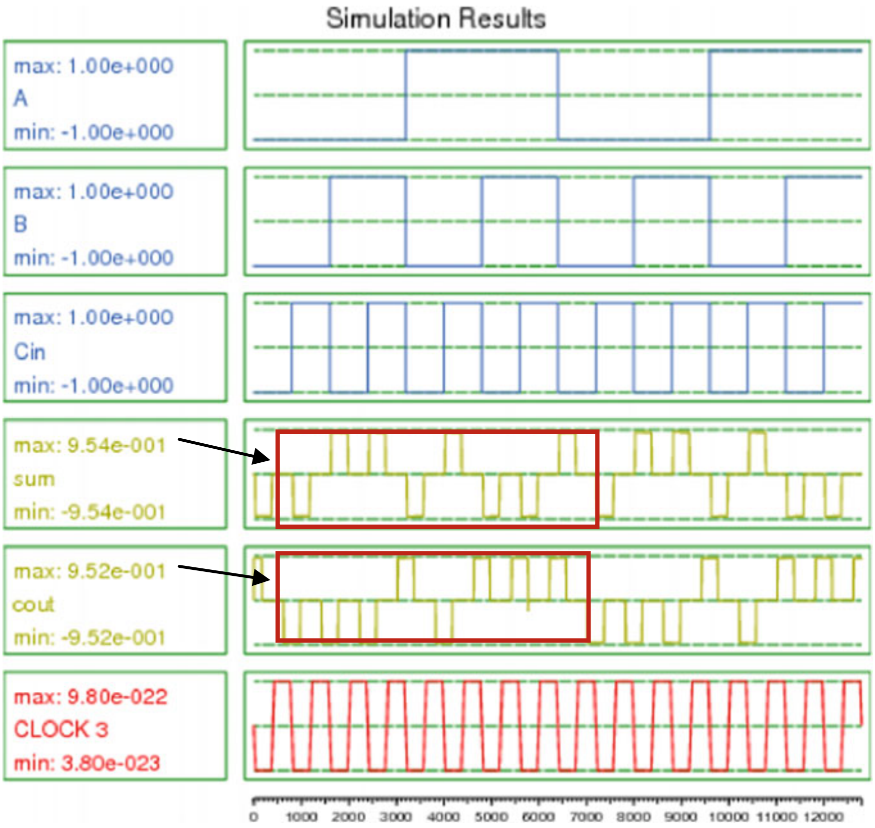


Fig. 5.11 Simulation results obtained for the full adder

counts when compared with both coplanar and multilayer designs. Table 5.5 shows the details of QCA based metrics for various multi-bit adders particularly for $n \in [4, 8, 16]$. Results reveal that the adder in [1] offers 44% cell count reduction for 4-bit and 50% for 16-bit in comparison to best RCA architecture [5]. Further results show that RCA design in [1] consumes lesser area with respect to design [5] by 38, 58, and 47% for 4-, 8-, and 16 bits, respectively. This confirms the efficient layout of FA that leads to logic advantage of RCA. Note that the multibit adder by authors [1] faces considerable speed loss than other multilayer designs [8, 21, 26] while offers advantages with respect to area and cell count.

Our novel architecture for RCAs attains a constant throughput, since operations are done in pipelined fashion and executing a number of additions in parallel. This modularity helps designer to construct more practical layout by considering realistic clock distribution like two-dimensional wave clocking method [28]. Table 5.6 gives a comparison among different RCA designs in terms of delay and throughput. The pipeline structure enables the proposed structure in [1] to maintain throughput of 2, as two Sum outputs resulted after each 0.5 clock cycle.

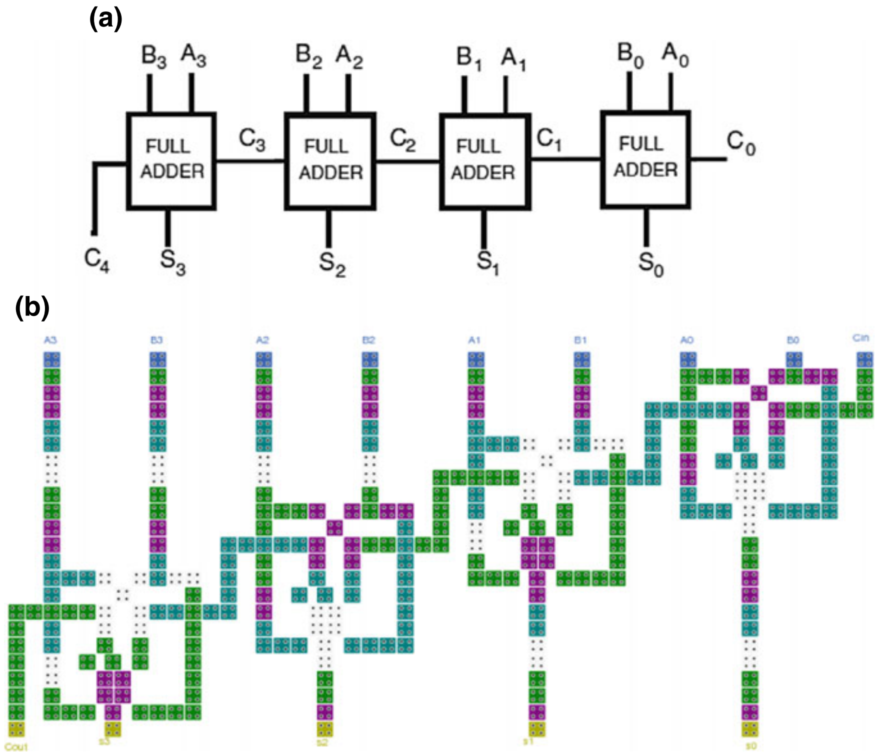


Fig. 5.12 A 4-bit RCA **a** logic diagram and **b** QCA layout

5.5.1 Power Analysis

To calculate power dissipation of the designs, QCAPro has been used as a power estimator tool [19]. For estimation of power dissipation, three tunneling energy values are considered (0.5 E_k, 1.0 E_k, and 1.5 E_k) at 2 K temperature. The thermal layout for the presented FA is shown in Fig. 5.14 for estimation of energy dissipation at tunneling energy of 0.5 E_k. Here cells those dissipate more average energy dissipation are indicated by dark color. Table 5.7 depicts energy dissipation results out of different FA designs [1, 4, 5, 11, 14, 29] in terms of average leakage and switching energy dissipation. Result indicates that design in [1] offers better power efficiency when compared with previous designs. The power dissipation profile for RCA of size 4 is depicted in Fig. 5.15 with tunneling energy of 0.5 E_k at 2 K temperature, and the presented RCA dissipates energy due to switching and leakage around 0.417 and 0.105 eV at 0.5 E_k energy level. So it can be concluded that the use of novel design by [1] not only delivers best area-delay trade-off but also equally energy-efficient design.

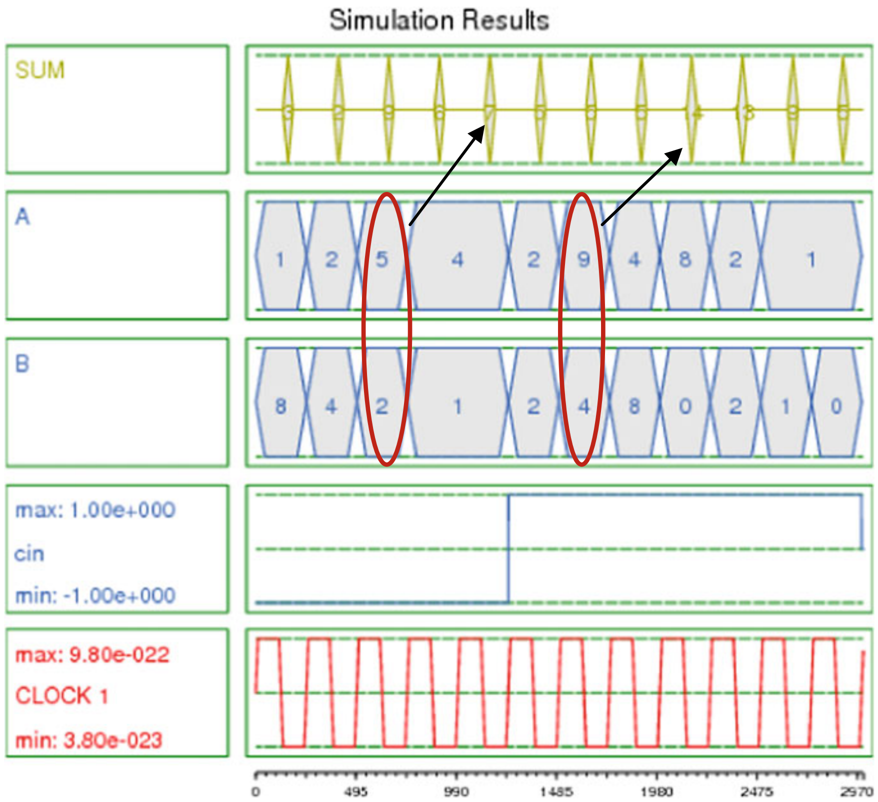


Fig. 5.13 Simulation waveforms of the 4-bit RCA

5.6 XOR Gate Design: A Novel Structure

In previous discussion, we have considered a full adder with MV5 gate for deriving ‘Sum’ output. Generally, a QCA-based 3-input XOR gate can be designed by combining 2-input XOR gates. Since there is more than one gate level in this design, the design complexity and latency increase, leading to circuit performance reduction. Several researches have been made in this direction to determine optimal cell layout. This section discusses a compact view of full adder that considers a novel XOR structure for its realization. The novel three-input XOR structure is based on explicit interactions between QCA cells ignoring the conventional designing methods [30]. The layout and simulation results for the structure are illustrated in Fig. 5.16. It is observed that this design requires two clock phases to deliver correct output, shown by solid box in Fig. 5.16b.

Table 5.4 Comparison of full adders

Full adder designs	Cell count	Area (μm)^2	Delay (clock cycles)	Crossover type
[21]	135	0.14	1.25	Multilayer
[6]	93	0.087	1	Multilayer
[22]	82	0.09	0.75	Multilayer
[8]	79	0.064	1	Multilayer
[7]	73	0.080	0.75	Multilayer
[9]	73	0.04	0.75	Multilayer
[12]	61	0.03	0.75	Multilayer
[23]	60	>0.07	0.5	Multilayer
[24]	58	0.03	0.75	Multilayer
[10]	52	0.04	1	Multilayer
[25]	220	0.36	3	Coplanar (rotated cells)
[20]	145	0.16	1	Coplanar (rotated cells)
[4]	105	0.17	1	Coplanar (rotated cells)
[5]	102	0.097	2	Coplanar (rotated cells)
[11]	95	0.09	1.25	Coplanar (rotated cells)
[14]	71	0.06	1.5	Coplanar (rotated cells)
[1]	55	0.04	1	Coplanar (clock zone based)

5.6.1 Full Adder Using New XOR Structure

Let consider a FA using MV3 gate has three inputs A, B, and C_{in} (Input Carry); then Output Carry is denoted as:

$$C_{\text{out}} = MV3(A, B, C_{\text{in}}) = A \cdot B + B \cdot C_{\text{in}} + C_{\text{in}} \cdot A \tag{5.6}$$

The Sum for FA is given as follows:

$$\text{Sum} = A \oplus B \oplus C_{\text{in}} \tag{5.7}$$

The QCA design for an efficient FA is illustrated in Fig. 5.17. This FA is composed of one of the most compact XOR gates that based on new cell arrangement (NG), ignoring use of MV3 modules as indicated by red box in Fig. 5.17b.

This QCA structure follows a generic strategy that counts lower clock phases for the coplanar wire crossing. The largest clock zone dimension comprises thirteen cells and has at least two cells. Figure 5.18 shows the simulation results for 1-bit FA. It is inferred that the first correct output is generated for both the outputs after two phases denoted by black boxes. This can be easily verified that there is degradation in strength of polarization of 11.4% for Sum output when inputs A, B, and C_{in} of polarization 1.00e+000. This means the design needs further improvement toward the signal integrity and robustness.

Table 5.5 Comparison results

Adders	n-bits	Cell count	Area (µm)^2	Delay (clock phases)	Crossover type
[27] CLA	16	4489	3.65	17	Coplanar (rotated cells)
[27] BKA	8	1462	1.06	10	Coplanar (rotated cells)
[5] RCA	4	558	0.85	20	Coplanar (rotated cells)
	8	1528	2.93	36	Coplanar (rotated cells)
	16	4652	10.85	68	Coplanar (rotated cells)
[8] BKA	4	680	0.645	7	Multilayer
	8	1782	1.49	10	Multilayer
	16	4350	3.55	16	Multilayer
[8] RCA	4	339	0.25	7	Multilayer
	8	712	0.745	11	Multilayer
	16	1602	1.996	19	Multilayer
[26] CLA	4	–	–	–	Multilayer
	8	1785	1.456	9	Multilayer
	16	4114	3.672	15	Multilayer
[26] CFA	4	–	–	–	Multilayer
	8	1143	1.018	9	Multilayer
	16	2817	2.453	13	Multilayer
[21] RCA	4	651	1.2	17	Multilayer
	8	1499	3.56	33	Multilayer
	16	3771	11.77	65	Multilayer
[21] CLA	4	1575	1.89	14	Multilayer
	8	3988	5.53	26	Multilayer
	16	10,217	15.51	41	Multilayer
[1] RCA	4	311	0.52	10	Coplanar (clock zone based)
	8	807	1.23	18	Coplanar (clock zone based)
	16	2358	5.71	34	Coplanar (clock zone based)

Table 5.6 Performance of n-bit adders

Design	Latency (clock cycles)	Throughput
RCA [25]	$3n$	1
RCA [5]	$n + 1$	1
RCA [4, 20]	n	1
RCA [1]	$(n/2) + 0.5$	2

5.7 New Results on Ripple Carry Adder

Let A_i, B_i, and C_i be inputs to a FA. Then, the computation for carry (C_{i+1}) and sum (S_i) outputs is given by the following equations:

$$C_{i+1} = MV3(A_\mathrm{i},\ B_\mathrm{i}, C_\mathrm{i}) = A_\mathrm{i} \cdot B_\mathrm{i} + B_\mathrm{i} \cdot C_\mathrm{i} + C_\mathrm{i} \cdot A_\mathrm{i} \tag{5.8}$$

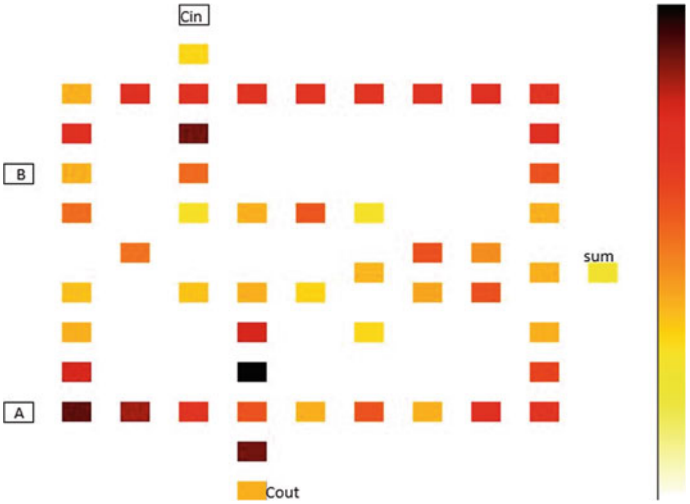


Fig. 5.14 Power dissipation map for full adder gate at 2 K temperature and tunneling energy of 0.5 E_k

$$S_i = (A_i \oplus B_i \oplus C_i) \tag{5.9}$$

A n-bit RCA is constructed of n FA modules placed in cascade manner. This establishes a carry path in RCA as FAs are placed in a series fashion as shown in Fig. 5.19a. The QCA implementation shows each FA delivers carry output after eight clock phases, so total delay is found to be ((n + 1/4)). For correct circuit operation, the Sum output of one stage must be fed to the next stage in two clock phases. In addition, the inputs in subsequent stage delayed accordingly, i.e., as long as the carry is not received from the previous stage, the new operand will not appear. A 4-bit RCA structure is illustrated in Fig. 5.19b considering one of the most compact FA modules in the literature, which leads to area-efficient layout. This design needs 195 cells for its realization and provides the final result after latency of five clock phases. It comprises four rotated MV3 and four new XOR gates [30]. The first set of inputs for simulation given in Fig. 5.20 corresponds to $X[3{:}0] = 2$; $Y[3{:}0] = 0$ (since the initial carry is set to 0). The output S [4:0] = 2 is available after 5 clock phases.

To show practical realizability of the presented architectures, we have drawn QCA layout for a RCA of size $n = 32$. Figure 5.21 depicts the QCA layout for the 32-bit adder which utilizes 2969 QCA cells, the delay of 33 clock cycles, and active area of 0.043 μm^2. Note the spiral paths were considered for reduce effective wire area for the 32-bit architecture.

Table 5.7 Energy dissipation comparison of coplanar QCA full adder designs[a]

Full adder designs	Avg. Avg. leakage energy dissipation (meV)				Avg. switching energy dissipation (meV))		Total energy dissipation (meV)		
	0.5 E_k	1.0 E_k	1.5 E_k	0.5 E_k	1.0 E_k	1.5 E_k	0.5 E_k	1.0 E_k	1.5 E_k
[29]	162.81	455.99	779.72	488.06	403.54	332.70	650.87	859.53	1112.42
[5]	58.48	160.50	272.30	101.20	83.07	68.20	159.68	243.57	340.5
[4]	38.26	124.98	230.51	281.48	246.17	210.92	319.74	371.15	441.43
[11]	25.77	79.67	144.52	103.81	91.89	80.08	129.58	171.56	224.6
[14]	20.75	61.99	110.46	100.16	87.83	75.98	120.91	149.82	186.44
[1]	17.36	51.96	91.53	81.20	68.76	57.67	98.57	120.72	149.19

[a] QCAPro allows only power evaluation of a coplanar QCA design

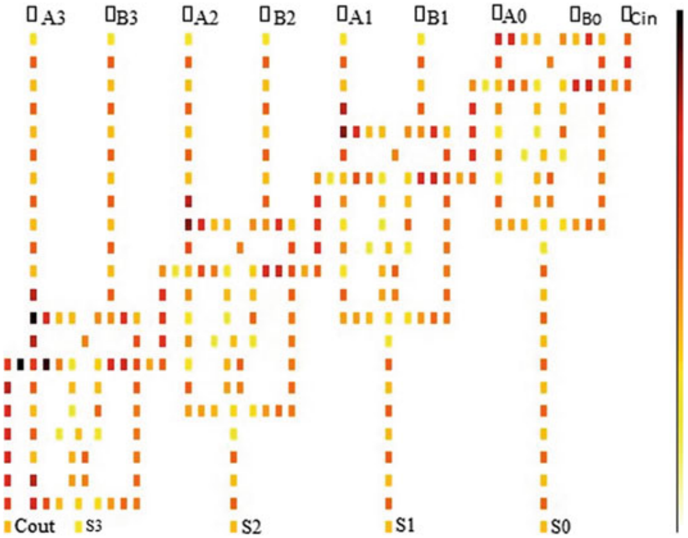


Fig. 5.15 Power dissipation map for 4-bit RCA at 2 K temperature and tunneling energy of 0.5 E_k

5.7.1 Power Analysis

To evaluate power dissipation in FA designs, QCAPro has been used as a power estimator tool [19], where total energy dissipation in a QCA circuit is calculated as summation of dissipation due to both leakage and switching. For estimation of power dissipation, three tunneling energy values are considered (0.5 E_k, 1.0 E_k, and 1.5 E_k) at 2 K temperature. The thermal layout profile for the examined FA is shown in Fig. 5.22 for estimation of energy dissipation at tunneling energy of 0.5 E_k. Here cells those dissipate more average energy dissipation are indicated by dark color. Table 5.8 depicts energy dissipation results out of different FA designs [4, 5, 11, 29, 31] in terms of average leakage and switching energy dissipation. From Table 5.8, we noted that FA in [30] receives at least 27% less average energy dissipation at 2 K temperature and energy level of 1.0 E_k when compared with all other competitors.

5.8 Performance Evaluation and Comparisons

In order to identify the efficacy of the presented architecture, a detailed analysis and comparison with available designs are given here. For performance evaluation of adders, we consider the general QCA parameters, i.e., complexity, latency, area, and the newly suggested parameters in [32]. Table 5.9 summarizes FA designs in QCA.

This shows that the FA in [2] attains superior results in all QCA design parameters as compared to the most optimal design in [31]. Further, this design exhibits

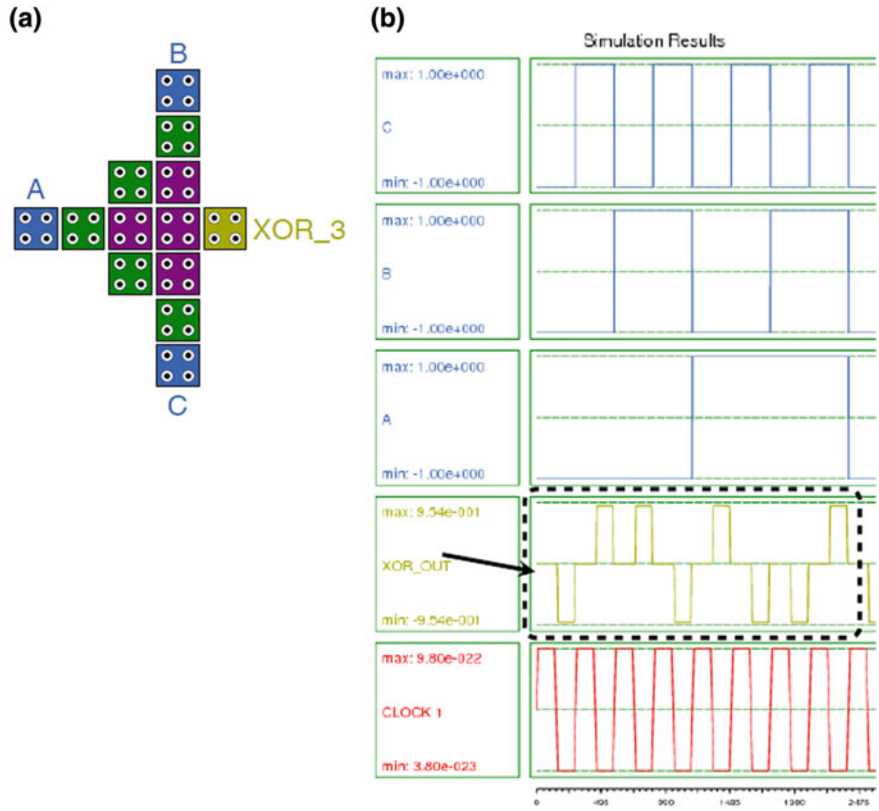


Fig. 5.16 XOR gate (a) QCA layout and (b) simulation waveforms [30]

significantly reduced complexity and QCA area when compared to previous multilayer structures except in [34], while its latency is lower than that for all designs reported in the literature. Table 5.10 lists area of n-bit version of most of the relevant adders. However, for those adders a generalized expression of area for n-bit versions is not obtained; they are marked with '–'. Figure 5.23 depictes the area metric of all representative designs. The adder in [2] has smallest area among all single and multilayer adders, which appears to be mainly due to the usage of new QCA full adder structure. Table 5.11 describes comparison of n-bit adders with respect to number of MVs, INVs, delay, and layer type. Those information that are not available are marked as '-'. Liu et al. [32] mentioned that the fabrication complication in QCA circuit is directly related to the number of wire crossing. To find the cost of multilayer crossover, authors have taken a multiplication factor of three with respect to a coplanar crossover. The overall complexity of QCA layout depends upon the MVs (M), INVs (I), and crossover (C). Authors in [32] describe this complexity metric as Eq. 5.10.

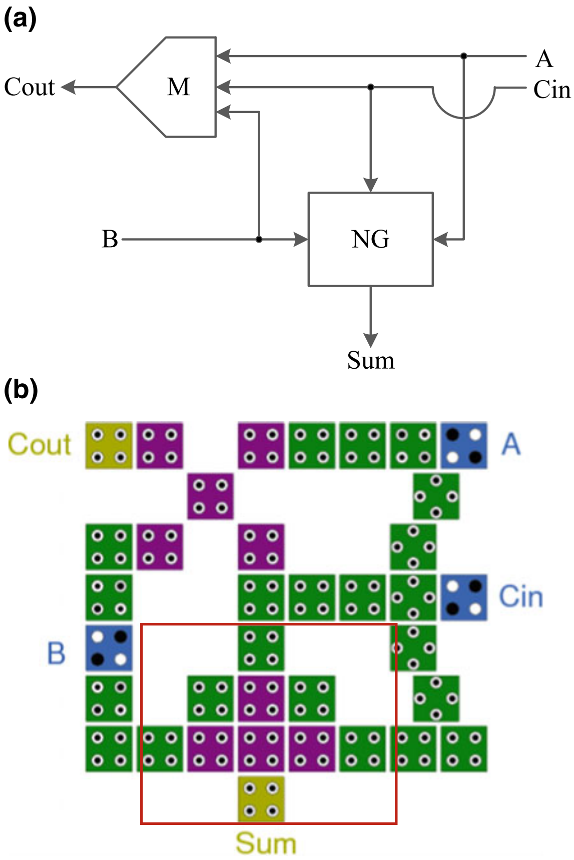


Fig. 5.17 FA (a) Schematic and (b) QCA layout

$$\text{Complexity} = M + I + C \tag{5.10}$$

Authors in [2] introduced a multiplication factor of two to find equivalent number of 3-input majority gate for a 5-input majority gate. Comparison of the complexity of n-bit adder is presented in Fig. 5.24. From this figure, we can observe that adder in [2] needs lesser number of MVs, INVs, and crossovers with respect to best existing adders. However, BK multilayer adder proved to be least efficient in terms of overall complexity. Figure 5.25 depicts the cost of crossing for various n-bit adders. It is worth noting that design in [2] achieves lower cost of crossing, while adders by [3, 9, 10] offer higher cost of crossings. Delay of a QCA design plays a major role in all conventional design parameters. Figure 5.26 gives the comparison of multibit adders with respect to delay. It has been noted that the adders in [2, 31, 33, 34] possess smaller delay when compared with the existing best designs except BK adder, while Takala adder [5] found to be least preferable.

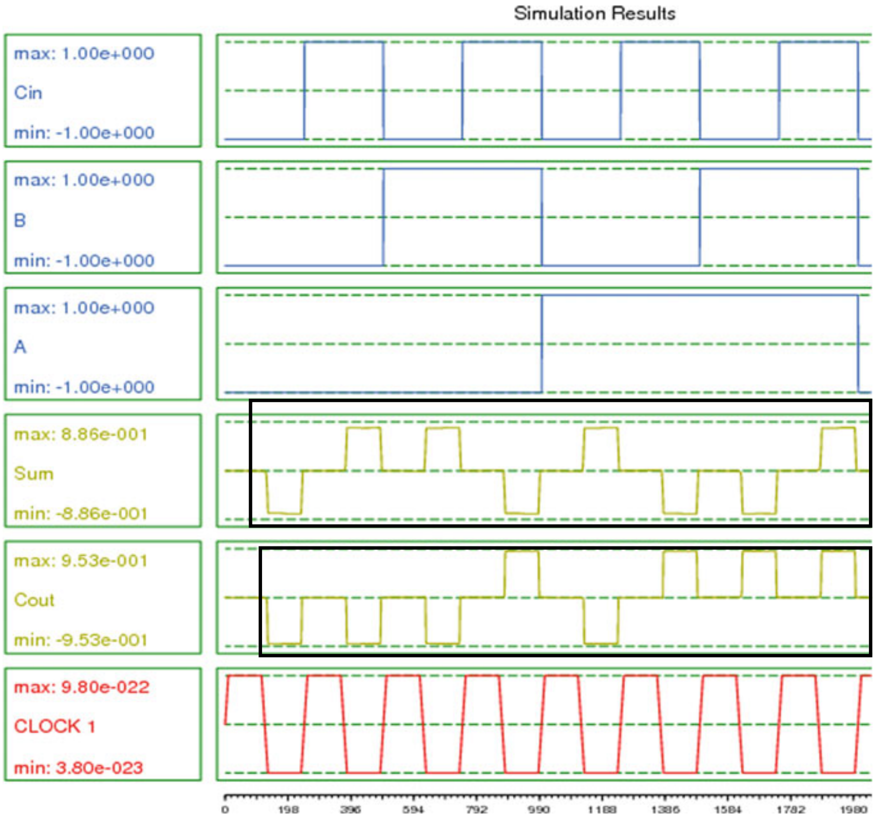


Fig. 5.18 Simulation results obtained for the FA

One aspect of power dissipation in logic gates is due to information loss during computation. Gates like 3-input majority gate is inherently irreversible as the inputs and outputs are not mapped one to one. This unbalanced mapping results information loss. As suggested in [35] and recently validated in [36], heat generated out of the system is because of reduction in entropy due to information lost. A generalized function for power dissipation in QCA circuit is devised by Sasamal et al. [2] utilizing majority gates and inverters as in Eq. 5.11.

$$P_d = (M \cdot X) + (I \cdot Y) \tag{5.11}$$

where X defines the maximum power consumption by a MV3 gate (e.g., 42.9 meV when input switching from '000' to '111' at 1.0 E_k, $T = 2$ K) and Y represents maximum power consumption by an inverter (e.g., 30.2 meV when input switching from 0 to 1 or vice versa at 1.0 E_k, $T = 2$ K) [37]. A MV5 gate has higher degree of imbalanced with respect to MV3 gate. In general, MV5 has at least twice the degree

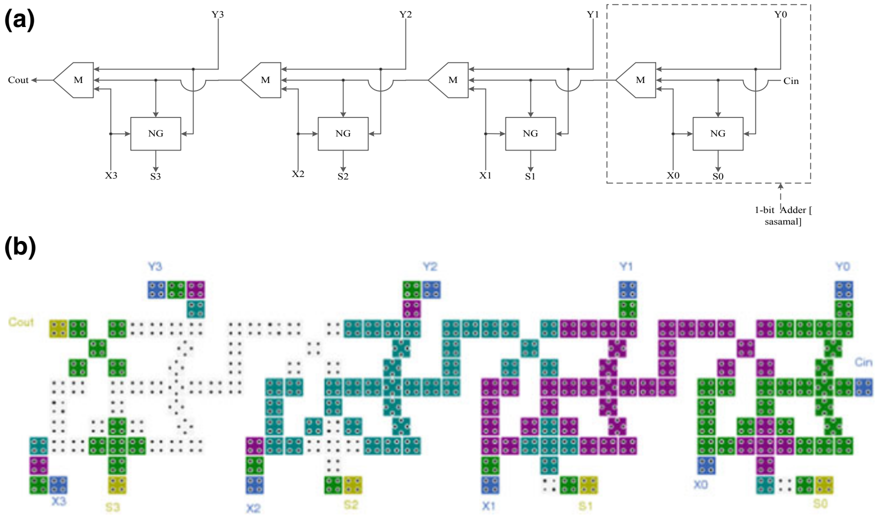


Fig. 5.19 A 4-bit RCA (a) schematic and (b) QCA layout

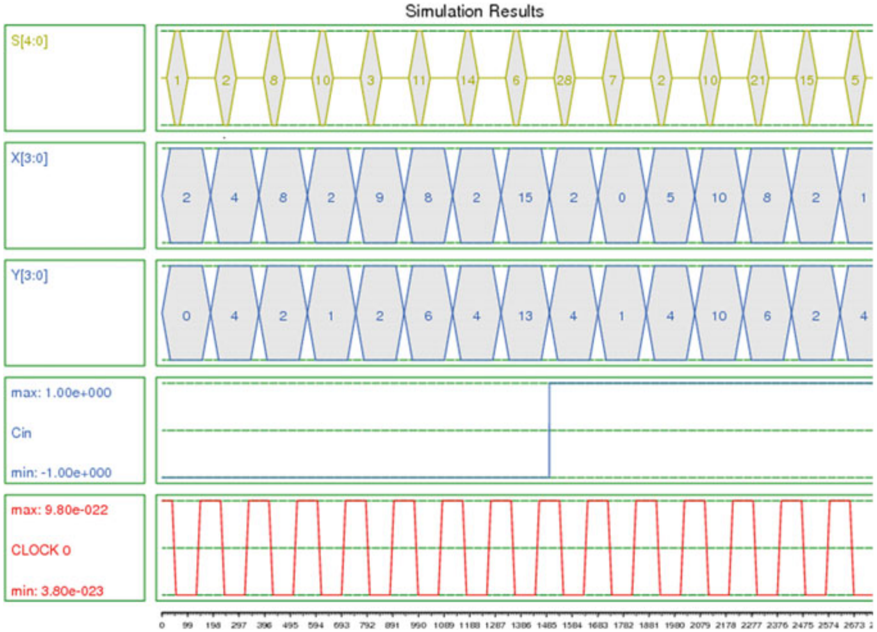


Fig. 5.20 Simulation waveforms of the 4-bit RCA

Fig. 5.21 Implementation of 32-bit RCA with QCA cells

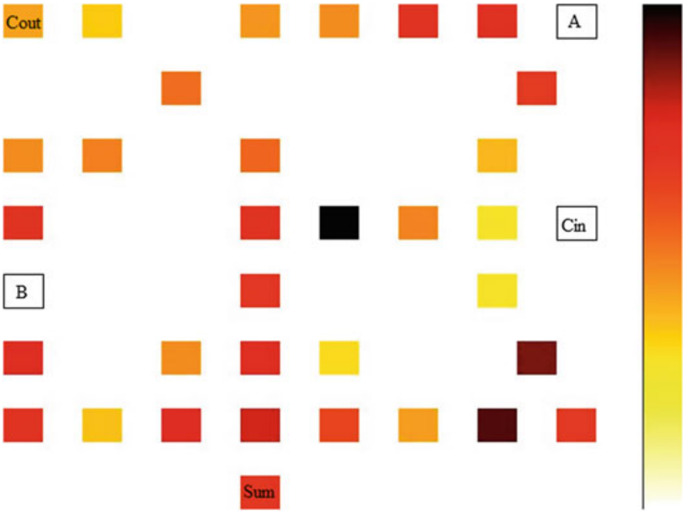


Fig. 5.22 Power dissipation map for one-bit FA gate at 2 K temperature and tunneling energy of 0.5 E_k

of imbalance compared to a MV3 gate. For calculation of power dissipation in a QCA layout utilizing MV5 gates, above assumption has been taken. Besides this, power dissipation due to QCA wire is neglected as no information is lost in a wire [38].

Figure 5.27 presents comparison of irreversible power dissipation for n-bit adders. It is observed that adder designed in [2] focuses toward more power-efficient QCA layout among all existing designs, and *KSA* is the least power-efficient one. Very recently, authors in [32] devised a new cost function more related to QCA circuits. This function focuses on power dissipation, complexity, and latency, as expressed in Eq. 5.12.

$$\text{QCA}_{\text{cost}} = \left(M^k + I + C^l\right) * T^p, \qquad 1 \le k, l, p \tag{5.12}$$

where M, I, C, and T stand for number of MVs, INVs, crossovers, and delay (clock cycles), respectively. Coefficients k, l, and p are to be suitably changed to emphasize power dissipation, complexity, and latency respectively based on the requirement specified by a specific application. It can be observed that a family of cost functions

Table 5.8 Power consumption comparison of single-layer QCA FA designs[a]

FA designs	Avg. leakage energy dissipation (meV)			Avg. switching energy dissipation (meV)				Total energy dissipation (meV)	
	0.5 E_k	1.0 E_k	1.5 E_k	0.5 E_k	1.0 E_k	1.5 E_k	0.5 E_k	1.0 E_k	1.5 E_k
[29]	162.81	455.99	779.72	488.06	403.54	332.70	650.87	859.53	1112.42
[5]	58.48	160.50	272.30	101.20	83.07	68.20	159.68	243.57	340.5
[4]	38.26	124.98	230.51	281.48	246.17	210.92	319.74	371.15	441.43
[11]	25.77	79.67	144.52	103.81	91.89	80.08	129.58	171.56	224.6
[31]	16.43	51.09	92.60	105.94	92.68	79.61	122.37	143.77	172.21
[2]	18.10	52.34	89.65	62.02	52.09	43.00	81.5	104.42	132.65

[a] QCAPro allows only power evaluation of a coplanar QCA design

Table 5.9 Comparison of QCA FAs

Full adder designs	Cell count	Area (μm^2)	Delay (clock cycles)	Crossover type
[3]	192	0.20	NA	–
[21]	135	0.14	1.25	Multilayer
[6]	93	0.087	1	Multilayer
[22]	86	0.10	0.75	Multilayer
[8]	79	0.064	1	Multilayer
[9]	73	0.04	0.75	Multilayer
[23]	60	>0.07	0.75	Multilayer
[10]	52	0.04	1	Multilayer
[33]	38	0.02	0.75	Multilayer
[34]	23	0.01	0.75	Multilayer
[29]	292	0.62	3.5	Coplanar (rotated cells)
[25]	220	0.36	3	Coplanar (rotated cells)
[20]	145	0.16	1.25	Coplanar (rotated cells)
[4]	105	0.17	1	Coplanar (rotated cells)
[5]	102	0.097	2	Coplanar (rotated cells)
[11]	95	0.09	1.25	Coplanar (clocking based)
[31]	59	0.043	1	Coplanar (clocking based)
[2]	36	0.02	0.5	Coplanar (rotated cells)

may be determined by opting different values of k, l, and p. According to [32], the authors opted $k = l = 2, p = 1$ for the most general form of cost function, i.e., cost $=$ $(M^2 + I + C^2) * T$. We examined all previous adder designs considering cost function described above as given in Fig. 5.28. This clearly shows that the architecture in [2] has achieved lower cost function too. It is further concluded that use of the new FA layout in [2] can be used to develop multibit RCA circuits with substantial energy savings.

5.9 Summary

In this chapter, an efficient layout of 5-input majority gate is examined using single-layer QCA technology. To show the advantages of this gate, QCA-based adders of various lengths are demonstrated. The comparison results show that the presented designs have significant improvements over existing designs. Further, a compact structure for 3-input XOR gate is presented based on explicit interaction of cells, which utilize a new arrangement of cells. The efficacy of compact XOR gate is verified by designing RCAs as specific application.

Table 5.10 Area of QCA adders (UNIT: μm^2)

Adder	4-bit	8-bit	16-bit	32-bit	n-bit
[10]	–	–	–	–	–
[31]	0.208	0.492	1.292	3.814	$\left[0.85 \times 2^{(\log_2 n-2)}\right] \times \left[0.24 + \left(2^{(\log_2 n-2)} - 1\right) \times 0.045\right]$
[33]	0.24	0.59	1.55	–	–
[23]	0.4	1.10	2.72	–	–
[21]	1.2	3.56	11.77	42.14	–
[22]	0.405	0.94	2.44	7.30	$\left[0.9 \times 2^{(\log_2 n-2)}\right] \times \left[0.45 + \left(2^{(\log_2 n-2)} \times 0.08\right)\right]$
[11]	0.967	3.474	13.09	50.77	$[0.02(15n-1)] \times [0.02(8n+9)]$
[20]	–	–	–	–	–
[3]	2.121	7.4	27.70	107.75	$\left[1 + 0.8 \times \left(2^{(\log_2 n-2)} - 1\right)\right] \times \left[2^{(\log_2 n-1)}\right]$
[6]	1.41	4.86	17.97	69.25	$\left[1.6 \times 2^{(\log_2 n-2)}\right] \times \left[0.7 \times 2^{(\log_2 n-2)}\right]$
[8] (Brent–Kung)	0.645	1.49	3.55	10.77	$\left[0.7 \times 2^{(\log_2 n-2)}\right] \times \left[0.7 + 2^{(\log_2 n-5)}\right]$
[8] (Kogge–Stone)	0.738	2.40	6.60	18.36	–
[4]	2.323	8.647	33.12	130.47	$\left[2^{(\log_2 n-2)}\right] \times \left[2^{(\log_2 n-1)}\right]$
[5]	0.892	2.99	11.03	42.37	$\left[0.7 \times 2^{(\log_2 n-2)}\right] \times \left[2^{(\log_2 n-2)}\right]$
[9]	0.31	0.95	>2.9	–	–
[34]	0.1770	0.7130	2.8600	11.4550	$[0.02(4n-1)] \times [0.02(7n-2)]$
Proposed [2]	0.189	0.432	1.220	3.402	$\left[0.81 \times 2^{(\log_2 n-2)}\right] \times \left[0.23 + \left(2^{(\log_2 n-2)} - 1\right) \times 0.04\right]$

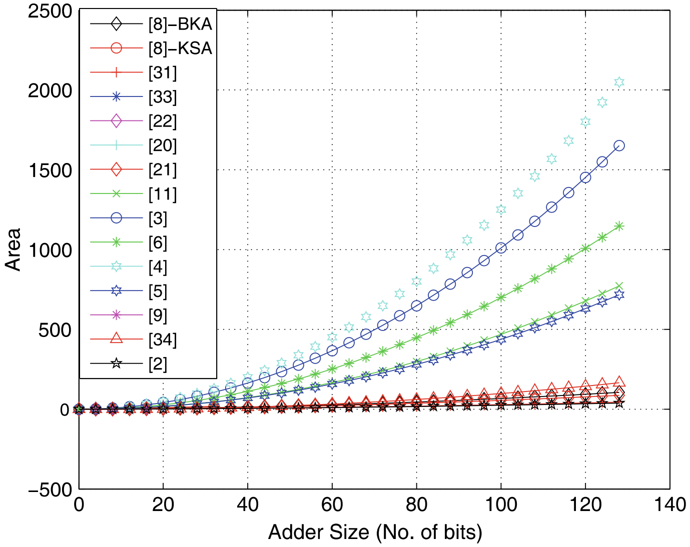


Fig. 5.23 Comparison of various adders with respect to area

Table 5.11 Summary of *n*-bit QCA adders

Adder	No. of MVs	No. of INVs	Layer type	No. of crossing	Delay (clock cycles)
[10]	$2n$	$2n$	Multilayer	$3n$	$\frac{2n+1}{4}$
[31]	$3n$	$2n$	Coplanar	$2n$	$\frac{n+3}{4}$
[33]	$3n$	n	Multilayer	$2n$	$\frac{n+2}{4}$
[23]	$3n$	$2n$	Multilayer	$2n$	$\frac{n+2}{4}$
[21]	$3n$	$2n$	Multilayer	$2n$	$n+\frac{1}{4}$
[22]	$3n$	$2n$	Multilayer	$2n$	$\frac{n+2}{4}$
[11]	$3n$	n	Coplanar	$2n$	$n+\frac{1}{4}$
[20]	$3n$	$2n$	Coplanar	$5n$	$n+\frac{1}{4}$
[3]	$5n$	$3n$	Coplanar	$9n$	$n+\frac{1}{4}$
[6]	$3n$	$2n$	Multilayer	$3n$	n
[8] (Brent–Kung)	$8n - 3\log_2 n - 4$	n	Multilayer	$n(\log_2 n - 3) + \log_2 n + 3n + 3$	$\frac{2\log_2 n+3}{4} + \frac{n\left(\log_2 n-2\right)}{32}$
[8] (Kogge–Stone)	$n(3\log_2 n - 1) + 5$	n	Multilayer	$\sum_{i}^{[\log_2 n]} \left(n-2^i\right)\cdot\left(2^i-1\right) + 3n$	–
[4]	$3n$	$2n$	Coplanar	$6n$	$n+\frac{1}{4}$
[5]	$3n$	$2n$	Coplanar	$3n$	$n+1$
[9]	$2n$	$2n$	Multilayer	$3n$	$\frac{2n+1}{4}$
[34]	$2n$	$6n$	Multilayer	$3n$	$\frac{n+2}{4}$
Proposed [2]	$3n$	0	Coplanar	N	$n+\frac{1}{4}$

'–' entry in table indicates where standard expression is not obtained

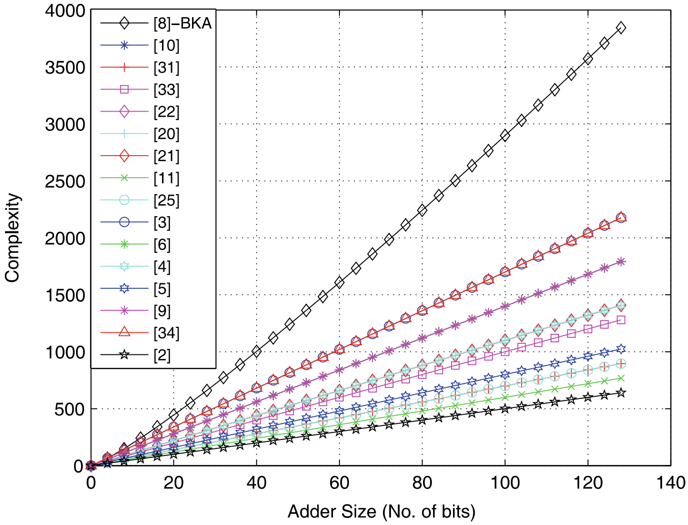


Fig. 5.24 Comparison of various adders with respect to complexity

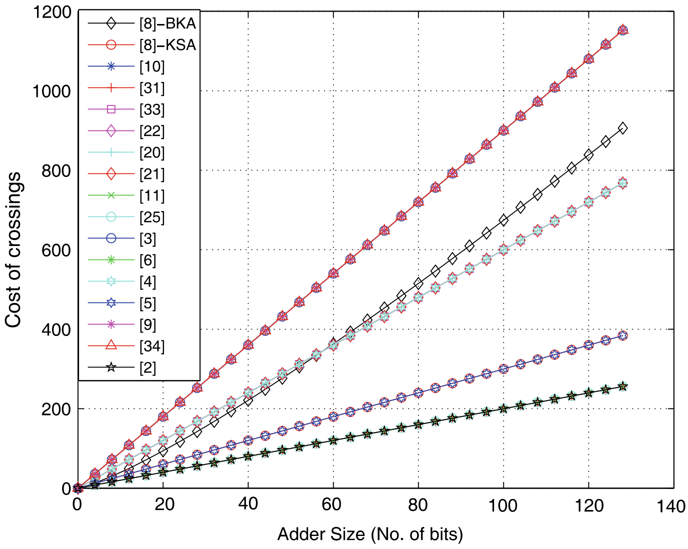


Fig. 5.25 Comparison of various adders with respect to cost of crossing

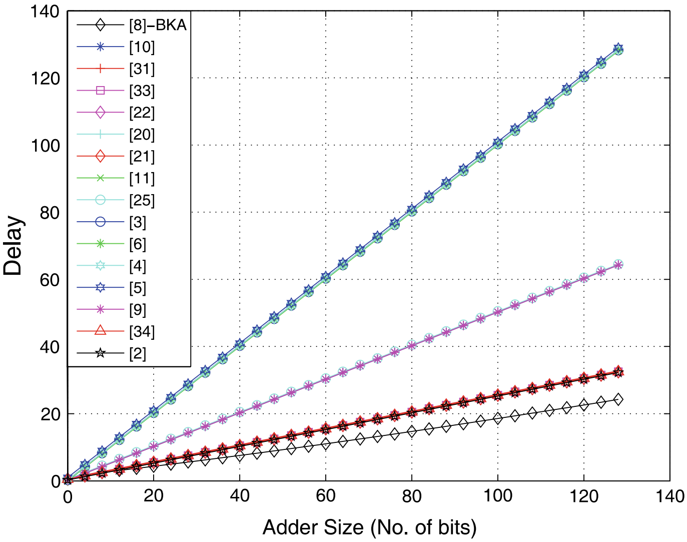


Fig. 5.26 Comparison of various adders with respect to delay

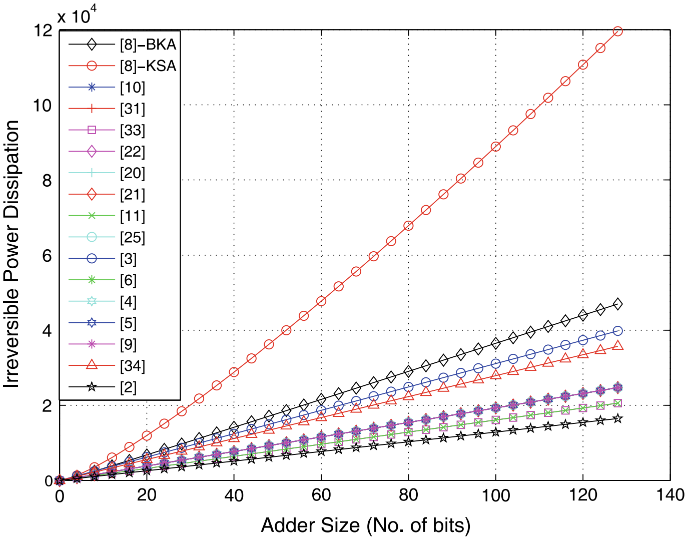


Fig. 5.27 Comparison of various adders with respect to irreversible power dissipation

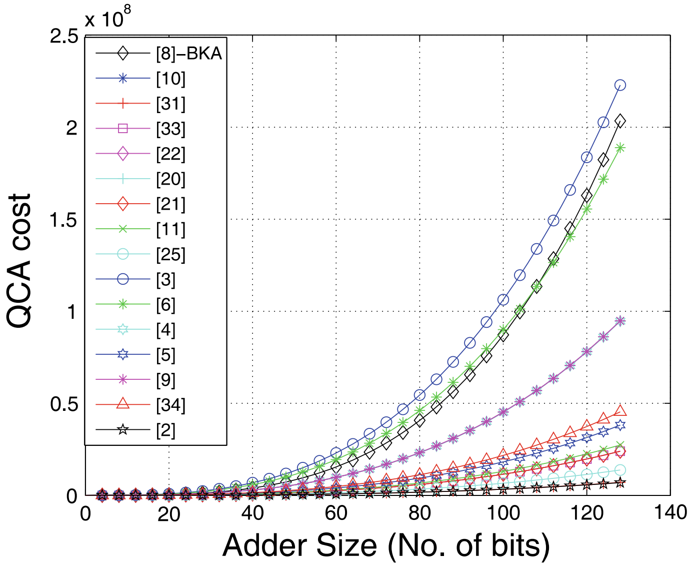


Fig. 5.28 Comparison of various adders with respect to QCA cost function = $(M^2 + I + C^2) * T$

References

1. T.N. Sasamal, A.K. Singh, A. Mohan, An optimal design of full adder based on 5-input majority gate in coplanar quantum-dot cellular automata. Optik-Int. J. Light Electr. Opt. **127**(20), 8576–8591 (2016)
2. T.N. Sasamal, A.K. Singh, U. Ghanekar, Efficient design of coplanar ripple carry adder in QCA. IET Circuits Dev. Syst. (2018). https://doi.org/10.1049/iet-cds.2018.0020
3. P.D. Tougaw, C.S. Lent, Logical devices implemented using quantum cellular automata. J. Appl. Phys. **75**(3), 1818–1825 (1994)
4. W. Wang, K. Walus, G.A. Jullien, Quantum-dot cellular automata adders, in *2003 Third IEEE Conference on Nanotechnology, 2003. IEEE-NANO 2003*, vol. 1 (2003), pp. 461–464
5. I. Hanninen, J. Takala, Binary adders on quantum-dot cellular automata. J. Signal Process. Syst. **58**(1), 87–103 (2010)
6. R. Zhang, K. Walus, W. Wang, G.A. Jullien, Performance comparison of quantum-dot cellular automata adders, in *2005 IEEE International Symposium on Circuits and Systems*, May 2005, pp. 2522–2526
7. H. Cho, Adder and multiplier design in quantum-dot cellular automata. IEEE Trans. Comput. **58**(6), 721–727 (2009)
8. V. Pudi, K. Sridharan, Low complexity design of ripple carry and brent kung adders in qca. IEEE Trans. Nanotechnol. **11**(1), 105–119 (2012)
9. K. Navi, R. Farazkish, S. Sayedsalehi, M.R. Azghadi, A new quantum-dot cellular automata full-adder,”. Microelectron. J. **41**(12), 820–826 (2010)
10. A. Roohi, K.H. Hosseini, S. Sayedsalehi, A symmetric quantum-dot cellular automata design for 5-input majority gate. J. Comput. Electron. **13**, 701–708 (2014)

11. S. Angizi, E. Alkaldy, N. Bagherzadeh, K. Navi, Novel robust single layer wire crossing approach for exclusive or sum of products logic design with quantum-dot cellular automata. J. Low Power Electron. **10**, 259–271 (2014)
12. K. Navi, S. Sayedsalehi, R. Farazkish, M.R. Azghadi, Five-input majority gate, a new device for quantum-dot cellular automata. J. Comput. Theor. Nanosci. **7**, 1546–1553 (2010)
13. S. Angizi, S. Sarmadi, S. Sayedsalehi, Design and evaluation of new majority gate-based RAM cell in quantum-dot cellular automata. Microelectron. J. **46**, 43–51 (2015)
14. S. Hashemi, K. Navi, A novel robust QCA full-adder. Procedia Mater. Sci. **11**, 376–380 (2015)
15. R. Akeela, M.D. Wagh, A five input majority gate in quantum dot cellular automata, in NSTI-Nanotech, vol. 2 (2011)
16. S. Hashemi, M. Tehrani, K. Navi, An efficient quantum-dot cellular automatafull-adder. Sci. Res. Essays **7**, 177–189 (2012)
17. D.B. Sen, A. Rajoria, B. Sikdar, Design of efficient full adder in quantum-dot cellular automata. Sci. World J. **2013**, 250802 (2013)
18. S. Sheikhfaal, S. Angizi, S. Sarmadi, M.H. Moaiyeri, S. Sayedsalehi, Designing efficient QCA logical circuits with power dissipation analysis. Microelectron. J. **46**(6), 462–471 (2015)
19. S. Srivastava, A. Asthana, S. Bhanja, S. Sarkar, Qcapro—an error-power estimation tool for QCA circuit design, in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, May 2011, pp. 2377–2380
20. R. Zhang, K. Walus, W. Wang, G.A. Jullien, A method of majority logic reduction for quantum cellular automata. IEEE Trans. Nanotechnol. **3**(4), 443–450 (2004)
21. H. Cho, E.E. Swartzlander, Adder designs and analyses for quantum-dot cellular automata. IEEE Trans. Nanotechnol. **6**(3), 374–383 (2007)
22. H. Cho, Adder and multiplier design and analysis in quantum-dot cellular automata. Ph.D. dissertation, Faculty of the Graduate School, University of Texas Austin (2006)
23. F. Bruschi, F. Perini, V. Rana, D. Sciuto, An efficient quantum-dot cellular automata adder, in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–4
24. M. Mohammadi, S. Gorgin, M. Mohammadi, Design of non-restoring divider in quantum-dot cellular automata technology. IET Circuits Dev. Syst. **11**(2), 135–141 (2017)
25. K. Kim, K. Wu, R. Karri, The robust QCA adder designs using composable QCA building blocks. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **26**(1), 176–183 (2007)
26. V. Pudi, K. Sridharan, New decomposition theorems on majority logic for low-delay adder designs in quantum dot cellular automata. IEEE Trans. Circuits Syst. II Express Briefs **59**(10), 678–682 (2012)
27. S. Perri, P. Corsonello, New methodology for the design of efficient binary addition circuits in QCA. IEEE Trans. Nanotechnol. **11**(6), 1192–1200 (2012)
28. V. Vankamamidi, M. Ottavi, F. Lombardi, Two-dimensional schemes for clocking/timing of QCA circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **27**(1), 34–44 (2008)
29. A. Vetteth, K. Walus, G.A. Jullien, et al., Quantum dot cellular automata carry-look-ahead adder and barrel shifter, in *Proceedings of IEEE Emerging Telecommunications Technologies*, Dallas TX (2002), pp. 2–4
30. T.N. Sasamal, A.K. Singh, U. Ghanekar, Design of non-restoring binary array divider in majority logic-based QCA. Electron. Lett. **52**(24), 2001–2003 (2016)
31. D. Abedi, G. Jaberipur, M. Sangsefidi, Coplanar full adder in quantum-dot cellular automata via clock-zone-based crossover. IEEE Trans. Nanotechnol. **14**(3), 497–504 (2015)
32. W. Liu, L. Lu, M. O'Neill, E.E. Swartzlander, A first step toward cost functions for quantum-dot cellular automata designs. IEEE Trans. Nanotechnol. **13**(3), 476–487 (2014)
33. M. Mohammadi, M. Mohammadi, S. Gorgin, An efficient design of full adder in quantum-dot cellular automata (QCA) technology. Microelectron. J. **50**, 35–43 (2016)
34. A. Roohi, R.F. DeMara, N. Khoshavi, Design and evaluation of an ultra-area-efficient fault-tolerant QCA full adder. Microelectron. J. **46**(6), 531–542 (2015)
35. R. Landauer, Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **5**(3), 183–191 (1961)

36. A. Berut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, E. Lutz, Experimental verification of Landauer's principle linking information and thermodynamics. Nature **483**(7388), 187–189 (2012)
37. W. Liu, S. Srivastava, L. Lu, M. O'Neill, E.E. Swartzlander, Are QCA cryptographic circuits resistant to power analysis attack? IEEE Trans. Nanotechnol. **11**(6), 1239–1251 (2012)
38. C.S. Lent, M. Liu, Y. Lu, Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling. Nanotechnology **17**(16), 4240 (2006)

Chapter 6
Array Dividers in QCA

In this chapter, we present design and implementation of iterative computational unit such as binary divider using QCA. Different types of dividers are introduced. Specifically, non-restoring divider is discussed that realized by iterative cellular arrays for parallel divisions. Performance metrics of different existing dividers are analyzed. Some of the works in this chapter have been reported in [1].

6.1 Introduction

Different QCA-based arithmetic logic circuits have been reported in several works [1–14]. Among these, the design of dividers is the major concern as it plays a pivotal role in evaluating the overall performance of a processing unit.

Division algorithms can be grouped into two classes, according to their iterative operator. The first one, where subtraction is the iterative operator, contains many familiar algorithms such as restoring and non-restoring division. These divisions are based on shift, subtraction, and addition operations which are relatively slow, as their execution time is proportional to the operand size.

The second class of algorithm relates to a higher speed class, where multiplication is the iterative operator. These algorithms converge quadratically where execution time is preoperational to $\log_2$ of the divisor size. Thus, the main difficulty is the evaluation of a reciprocal.

In this work, we examine the design and structure of the iterative cellular arrays for parallel divisions using non-restoring division algorithm. In these array dividers, iterative units are utilized to form the parallel divider structure.

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_6

6.2 Non-restoring Binary Array Divider (NRD)

In binary NRD, the partial remainder is measured out of an addition or subtraction of the dividend and the divisor with right shifted. The quotient bit is calculated by the sign of partial remainder which also decides whether to add or subtract the shifted divisor in the next cycle. 'Non-restoring divider' (NRD) algorithm has more advantages over 'restoring Divider' (RD) algorithm because it overcomes problems like delay due to the restoration process, realizing control logic, and unnecessary power dissipation.

The iterative unit in the NRD is the complement adder/subtractor (CAS) block. Each block has four inputs: A_i, B_i, P, and C_i, and two outputs: S_i and C_0 as depicted in Fig. 6.1.

Let n represents the number of bits for the algorithm, i is the iteration value, R_i defines partial remainder, q represents quotient set, Y and r define the divisor and the final remainder, respectively. Then the non-restoring division is determined by equations [8]:

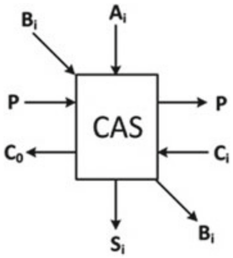
$$q_{i+1} = \begin{Bmatrix} 1, \text{ if } & R_i > 0 \\ 0, \text{ if } & R_i < 0 \end{Bmatrix} \tag{6.1}$$

$$R_{i+1} = \begin{Bmatrix} 2R_i - Y, \text{ if } & R_i > 0 \\ 2R_i + Y, \text{ if } & R_i < 0 \end{Bmatrix} \tag{6.2}$$

$$r = \begin{Bmatrix} 2^{-n} \cdot R_n, & \text{if } & R_i > 0 \\ 2^{-n} \cdot (2R_n + Y), & \text{if } & R_i < 0 \end{Bmatrix} \tag{6.3}$$

To construct an n-bit NRD, a 2D array of CAS blocks is required, and such structure needs N^2 CAS blocks. A 5 × 5 NRD block diagram is illustrated in Fig. 6.2. This divider divides an 8-bit number ($x1x2x3x4x5x6x7x8$) by a 4-bit number ($y1y2y3y4$) and generates a 5-bit quotient ($q0q1q2q3q4$) at the left side of the structure; remainder ($r4r5r6r7r8$) is produced at the bottom of the structure.

Fig. 6.1 A CAS block



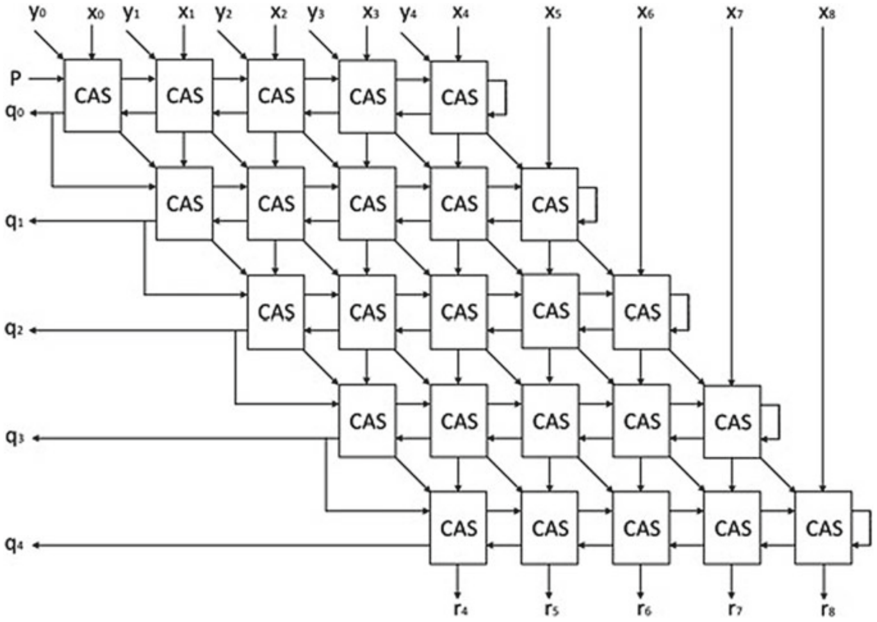


Fig. 6.2 Structure of a 5 × 5 non-restoring array divider

6.3 Non-restoring Divider (NRD) in QCA

As discussed in Chap. 2, majority function is the basic building block of any QCA circuit. So, QCA designs are realized by simply converting the Boolean equation to its corresponding majority logic-based circuit. Although various synthesis algorithms are devised to convert Boolean function into its equivalent majority-based circuit, the majority expressions by these methods do not guarantee a simplified majority expression for all the circuits. For some cases, the majority-based circuit requires more gates and levels when compared with the original Boolean function. The complexity and latency of the individual gates have an influential role in the performance of overall system. So, the overall performance of the circuit can be reduced by reducing number of gate levels.

The presented coplanar NRD is based on an efficient XOR gate where the numbers of gates and levels are reduced compared to the existing designs for 3-input XOR that use majority gates (3-input and 5-input majority gates). In addition, multilayer QCA designs face challenges to fabricate as well as feasibility of many designs leaves them unfavorable. Therefore, most of the designers restrict themselves by opting for single layer QCA designs.

The coplanar QCA divider is based on the non-restoring algorithm, which requires arrays of ‘complement adder/subtractor cells’ (CAS cells). Each cell is composed of a 2-input XOR function and a 1-bit full adder. The analysis shows the divider not only better than the prior single layer QCA designs, but also exhibit considerable superiority over the existing multilayer designs. QCADesigner has been considered to validate the functional correctness of all the designs.

6.3.1 Design of CAS Cell

The block diagram of the CAS cell is depicted in Fig. 6.3a, where the exclusive-OR gate takes two inputs, B_i and P (control signal), and results in XOR_OUT. The full adder operates on three inputs, A_i, XOR_OUT, and C_i (Input Carry). The carry (C_{o}) and sum (S_{i}) of the full adder are determined by:

$$C_{\mathrm{o}} = A_i \cdot \text{XOR\_OUT} + \text{XOR\_OUT} \cdot C_i + C_i \cdot A_i \tag{6.4}$$

$$S_i = A_i \oplus \text{XOR\_OUT} \oplus C_i \tag{6.5}$$

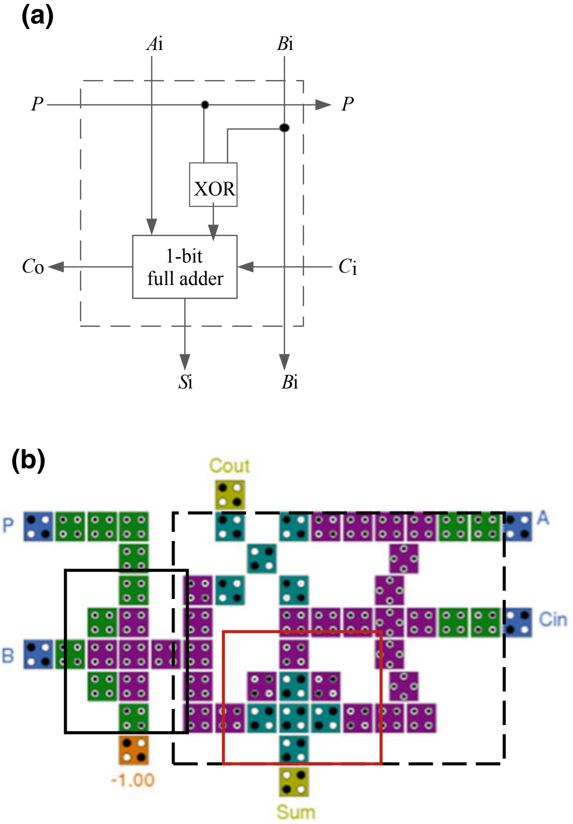


Fig. 6.3 CAS cell. **a** Block diagram, **b** QCADesigner layout

The QCA design of the CAS block is presented in Fig. 6.4b. The CAS cell uses a compact structure for 3-input XOR gate that depend on QCA cells interactions (shown in the red box) without considering the majority function [1]. A 2-input XOR design can be designed by making third input to 0. The XOR function is illustrated in solid square, which requires only 13 quantum cells and requires an area of 0.02 μm^2, and its delay is 0.5 clock cycles (Fig. 3.35b).

The 1-bit adder is shown in dashed square that comprises only 38 cells and an area occupation of 0.03 μm^2 with 0.5 clock cycles delay (Fig. 6.3b). Our proposed XOR gate and non-restoring divider have better performance than the prior best designs as shown in Tables 6.1 and 6.2.

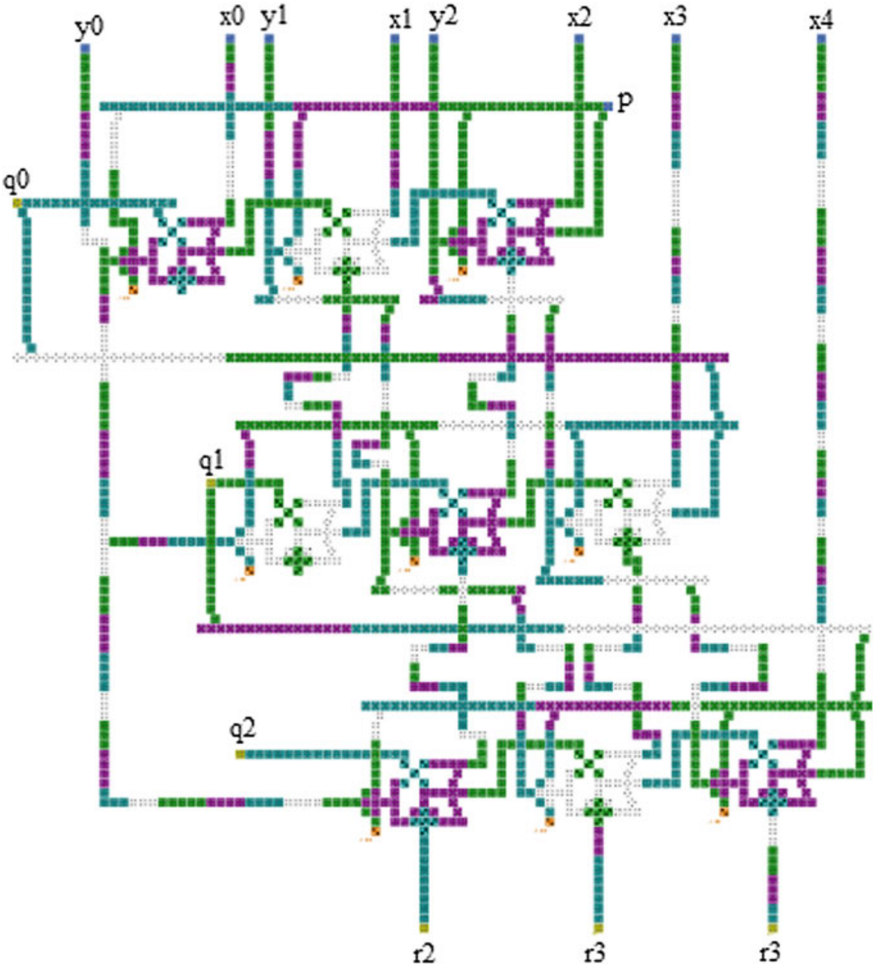


Fig. 6.4 QCA implementation of 3 × 3 non-restoring binary divider

Table 6.1 Comparison of 2-input XOR gates

Designs	Cell count	Latency (clock cycle)	Area (μm^2)	Layer type
[9]	60	1.5	0.09	Coplanar
[10]	67	1.25	0.06	Coplanar
[11]	29	0.75	0.03	Not needed
[1]	14	0.5	0.02	Not needed

Table 6.2 Comparison of non-restoring basic cell

Designs	Cell count	Latency (clocks)	Area (μm^2)	layer type
[12]	235	1.75	0.35	Multilayer
[13]	147	2.25	0.27	Coplanar
[1]	60	0.75	0.08	Coplanar

For the purpose of illustration, we have presented the layout of the 3 × 3 NRD in Fig. 6.4, which can be extended up to $n \times n$ divider. It comprises a 2-D array of adders that help to propagate the carry. A control signal P manages the operation of the CAS cell as a subtractor or an adder. The divisor, dividend, and quotient are represented by $(y_0.y_1y_2)$, $(x_0.x_1x_2x_3x_4)$, and $(q_0.q_1q_2)$, respectively. The leftmost bit y_0 and x_0 are used for the signs. Due to the compact CAS cells, the proposed divider layout achieves minimal latency and area. The QCA implementation of the 3 × 3 NRD requires 1686 cells covering an area of 3.40 μm^2 with 6.75 clock cycles' delay.

6.4 Simulation Results

QCADesigner [15] with default settings has been used for layout and simulation of NRD divider. Figures 6.5 and 6.6 depict the simulation results for the CAS cell and the 3 × 3 non-restoring binary divider, respectively.

6.5 Comparison and Discussion of Results

The comparison results of existing and presented QCA dividers are shown in Table 6.3. It can be perceived that the presented structure excels at all the best-reported designs reported in [7, 12, 13]. Particularly, the 4 × 4 divider in [1] achieves 38 and 63% improvements in latency, and area occupation, respectively, when compared to the best results.

Plot of latency versus different operand size for NRD, RD, and presented NRD is shown in Fig. 6.7. For an n-bit operand size NRD and RD, the latency is given by $3n^2 - 0.75$ and $4n^2 - 1$, respectively. In the presented coplanar n-bit divider, some

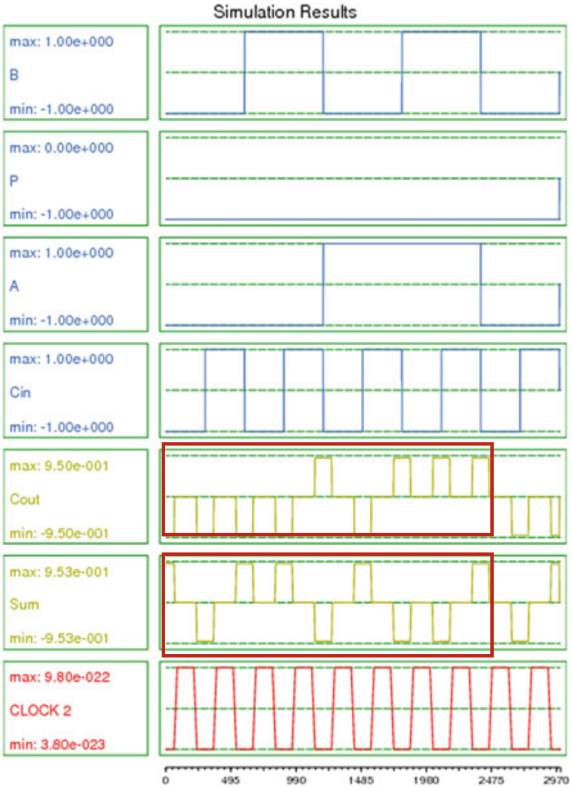


Fig. 6.5 Simulation results of QCA CAS cell

of the wire crossings are done by Clock zone-based crossover approach [16] and the rest are implemented by wires of 45° and 90° cells. Hence, the presented divider achieves an overall latency of $3n^2/4$ [1].

6.6 Summary

In this chapter, we have examined QCA-based non-restoring binary array divider. The presented divider design is based on a compact structure for 3-input XOR gate which utilizes a new configuration of cells. It has been shown that the divider is offered significant improvement over all existing coplanar and multilayer dividers.

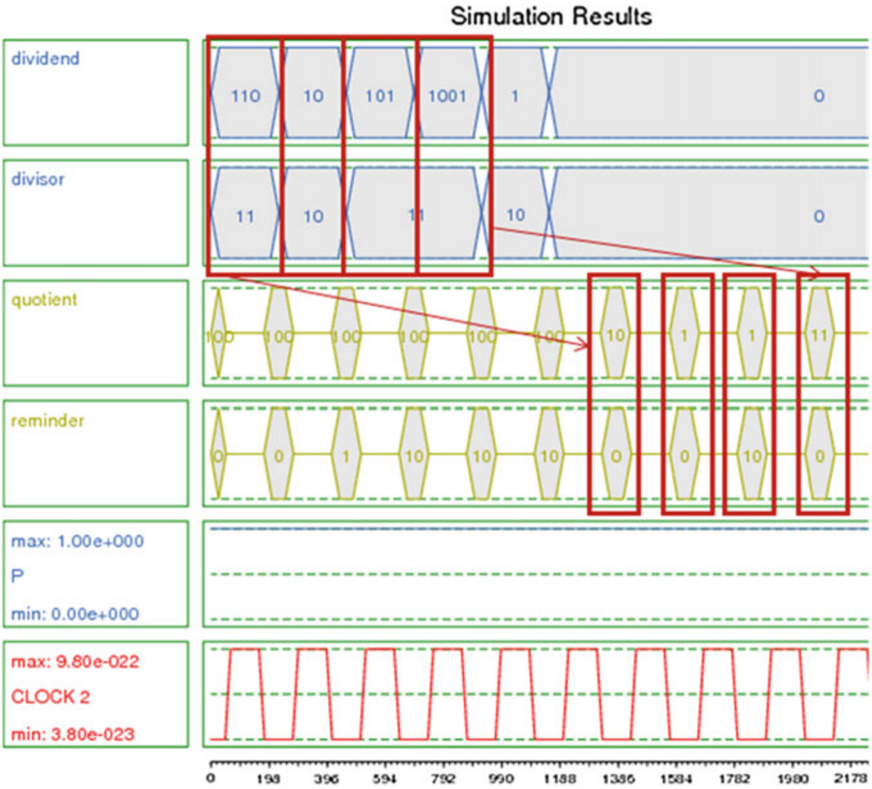


Fig. 6.6 Simulation diagram of 3 × 3 non-restoring binary divider

Table 6.3 Comparison of dividers

Dividers	Cell count	Latency (clock cycle)	Area (μm^2)	Layer type
3 × 3 RD [7]	6451	37	86.22	Coplanar
3 × 3 NRD [13]	3742	26.25	6.22	Coplanar
3 × 3 NRD [1]	1686	6.75	3.40	Coplanar
4 × 4 NRD [13]	6865	47.25	10.95	Coplanar
4 × 4 NRD [12]	5351	16.5	15.51	Multilayer
4 × 4 NRD [1]	3180	12	6.5	Coplanar

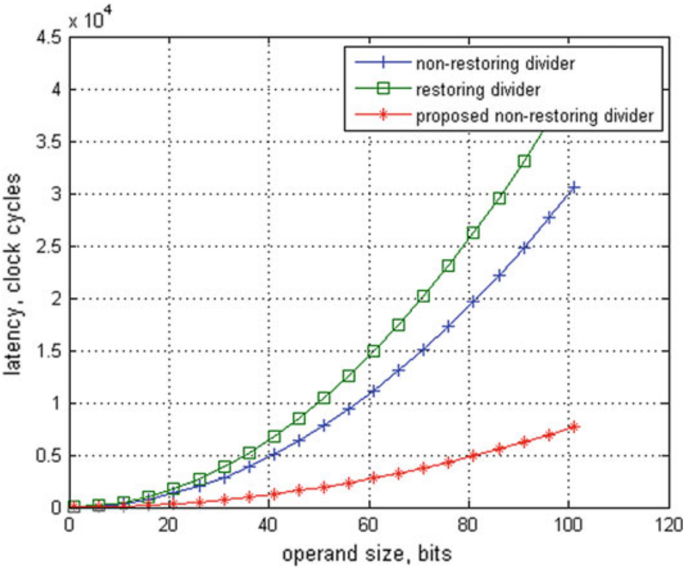


Fig. 6.7 Latency of QCA dividers for different operand sizes

References

1. T.N. Sasamal, A.K. Singh, U. Ghanekar, Design of non-restoring binary array divider in majority logic-based QCA. IET Lett. **52**(24), 2001–2003 (2016)
2. V. Pudi, K. Sridharan, Low complexity design of ripple carry and Brent-Kung adders in QCA. Trans. Nanotechnol. **11**, 105–119 (2012)
3. K. Navi, R. Farazkish, S. Sayedsalehi et al., A new quantum-dot cellular automata full-adder. Microelectron. J. **41**(12), 820–826 (2010)
4. Bruschi, F., Perini, F., Rana, V., et al.: An efficient quantum-dot cellular automata adder. *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, Mar 2011, pp. 1–4
5. C. Labrado, H. Thapliyal, Design of adder and subtractor circuits in majority logic-based field-coupled QCA nanocomputing. Electron. Lett. **52**(6), 464–466 (2016)
6. D. Abedi, G. Jaberipur, M. Sangsefidi, Coplanar full adder in quantum-dot cellular automata via clock-zone based crossover. Trans. Nanotechnol. **14**(3), 497–504 (2015)
7. Kim, S.W., Swartzlander, E.E.: Restoring divider design for quantum-dot cellular automata, in *11th IEEE International Conference on Nanotechnology*, Portland Marriott, USA, Aug 2011, pp. 1295–1300
8. C. Maurus, V.C. Hamacher, An augmented iterative array for high speed binary division. Trans. Comput. **c-22**(2), 172–175 (1973)
9. M.T. Niemier, Designing digital systems in quantum cellular automata. M.S. thesis, University of Notre Dame, Indiana (2004)
10. S. Angizi, E. Alkaldy, N. Bagherzadeh, K. Navi, Novel robust single layer wire crossing approach for exclusive or sum of products logic design with quantum-dot cellular automata. J. Low Power Electron. **10**, 259–271 (2014)
11. A.M. Chabi, S. Sayedsalehi, S. Angizi, K. Navi, Efficient QCA exclusive-or and multiplexer circuits based on a nanoelectronic-compatible designing approach. Int. Sch. Res. Not. **2014**, 1–9, Article ID 463967 (2014)

12. S. Sayedsalehi, M.R. Azghadi, S. Angizi, K. Navi, Restoring and non-restoring array divider designs in quantum-dot cellular automata. Inf. Sci. **311**, 86–101 (2015)
13. H. Cui, L. Cai, X. Yang, C. Feng, T. Qin, Design of nonrestoring binary array divider in quantum-dot cellular automata. Micro Nano Lett. **9**(7), 464–467 (2014)
14. T.N. Sasamal, A.K. Singh, A. Mohan, An optimal design of full adder based on 5-input majority gate in coplanar quantum-dot cellular automata. Optik **127**, 8576–8591 (2016)
15. K. Walus, T.J. Dysart, G.A. Jullien et al., QCA designer: a rapid design and simulation tool for quantum-dot cellular automata. IEEE Trans. Nanotech. **3**(1), 26–31 (2004)
16. S.H. Shin, J.C. Jeon, K.Y. Yoo, Design of wire-crossing technique based on difference of cell state in quantum-dot cellular automata. Int. J. Control Autom. **7**, 153–164 (2014)

Chapter 7
Design of Arithmetic Logic Unit in QCA

7.1 Introduction

In this chapter, we examined the reversible arithmetic logic unit (ALU) and its implementation in QCA framework. ALU is one of the fundamental components as it defines the performance of any processing systems. This chapter is structured in four sections. First section discusses different ALU structures in QCA. In Sect. 7.2, we analyze and validate one of the reversible ALU designs in QCA framework. Section 7.3 inspects the complexity of different reversible and non-reversible ALU structures with comparative analysis. Section 7.4 presents the summary of the chapter.

In recent years, numerous designs have reported, showing the efficiency of QCA technology toward high-density device. Meanwhile, designing circuits that enable lossless computation without information lost, i.e., reversible computing also gaining equal prominence. The energy dissipation of a QCA-based design is notably lower compared to $k_{\mathrm{B}}T\ln 2$ as a result of clocked information preserving methodology [1]. These energy-saving attributes introduce scope for reversible logic implementation using QCA framework.

An ALU is the basis of many processing units to execute both arithmetic and logical operations. So, from the processing point of view the ALU design must possess high-density device with higher computing speed and energy-efficient architecture. Moreover, the overall complexity can be reduced by the proper selection of combinational modules like full adder and multiplexer. Traditional ALU carries out arithmetic operations like addition, subtraction, and logical operations like AND, OR, XOR, or NOT taking two operands in parallel, and final output is computed by the usage of a multiplexer. Figure 7.1 presents one logic diagram for an ALU. Generally, there exist two design perspectives for an ALU construction: (1) multiplexer-based approach and (2) controller structure-based approach. Syamala et al. [2] reported multiplexer-based reversible ALU that enables four arithmetic and logical functions. Authors also demonstrated controller structure-based ALU that provides five arithmetic and logical operations. The multiplexer-based ALU realization is preferred as it turns

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_7

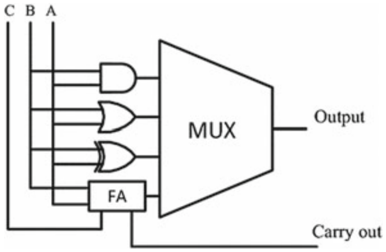


Fig. 7.1 Block diagram for an ALU [3]

out an efficient method in terms of speed and computation. We have categories all QCA-based ALUs into reversible and non-reversible design. In these contexts, various ALU designs in QCA are analyzed by different researchers [3–15]. For reversible ALU (RALU) design, the basic components must follow reversibility. Moreover, to realize RALU reversible gates can be used as the basic constituent.

Authors in [4] proposed reversible ALU and validated using QCA technology. This ALU incorporates four new reversible gates to design separate arithmetic and logical units. Result indicates the proposed design achieves superior performance with respect to other designs. But, the application-specific reversible gates limit the versatility of the design. Sen et al. [5] suggested a reversible ALU, considering a 2:1 multiplexer as the building block. Results confirm the effectiveness of the design in terms of complexity where an optimal layout for 2:1 multiplexer is used with 19 QCA cells. The proposed design delivers total 16 logical and arithmetic operations. But, the garbage and constant lines are relatively high. The complete QCA layout for the proposed RALU was not given. Authors in [6] devised a QCA-based reversible ALU. This design utilizes a compact 2:1 multiplexer using new RM gate. Authors also introduced fault-tolerant and attributed to the multiplexer which preferably tolerant to single missing fault only. In [7], authors analyzed the ALU proposed in [8] and implemented the same design using QCA. This design offers only four logical operations. For implementation, it needs 1096 cells, quantum cost of 13, and delay of 15 phases. Waje and Dakhole [9] proposed a 4-bit ALU utilizing QCA cells. For ADD operation, three logic depths required while one level required for XOR, AND, OR operations. The proposed 1-bit ALU incorporates two control lines and employs 2:1 multiplexer as the primary blocks. Authors also simulated 4-bit ALU. According to the authors, the proposed ALU has lesser number of components in comparison with CMOS counterpart and achieved better energy-efficient computation. A simple 12 ALU is presented by Ghosh et al. [10]. The proposed structure is good enough to provide total 12 operations based on the control inputs. This multilayer QCA-based ALU tried to minimize the QCA metrics, but suffers from low-strength polarized output which makes the design unsuitable for cascading connections. Teja et al. [11] investigated an ALU with extended functions. This ALU employs an arithmetic and logical function generator (ALFG) that generates 16 different functions using 4-bit patterns. The proposed architecture enables additional functions like shift, parity, rotate. Gadim et al. [12] presented multilayer ALU on QCA considering fault tolerance aspect. This ALU structure exploits a fault-tolerant rotated majority gate

Table 7.1 Existing ALU designs with QCA

ALU design	Implementation	Building block	Select line
[3]	Reversible	NHG	16
[4]	Reversible	RG1, RG2, RG3, RG4	12
[5]	Reversible	Multiplexer	16
[6]	Reversible	RM gate	16
[7]	Reversible	Gupta ALU	8
[9]	Non-reversible	Full adder, multiplexer	5
[10]	Non-reversible	Full adder, multiplexer	12
[11]	Non-reversible	ALFG	16
[12]	Non-reversible	Full adder, multiplexer	4
[13]	Reversible	Testable (t) adder	14
[14]	Non-reversible	Full adder, multiplexer	12
[15]	Reversible	RUG	19

full adder and multiplexer. Results confirm that the proposed design is tolerant to stuck-at faults. The authors in [13] have reported 1-bit ALU that exploits a testable (t) adder. Further, this structure is implemented on QCA. Comparison results suggest that this design needs more enhancements in area optimization. Babaie et al. [14] reported a multilayer QCA-based 1-bit ALU. This design uses a new full adder and ALU specific 4:1 multiplexer. It enables 12 different operations—4 logical and 8 arithmetic. Authors also examined the energy efficiency of the proposed structures. Naghibzadeh and Houshmand [3] proposed a reversible ALU exploring a new reversible 4 × 4 gate known as NHG. The circuit was synthesized and realized using QCA framework. This structure provides 16 functions with less number of garbage and constant lines. Comparison result indicates superiority of the proposed design in terms of design parameters. Sasamal et al. [15] discussed an efficient reversible ALU and implemented on QCA. The fundamental block used for the design was a 3 × 3 universal reversible gate. Authors proposed a HLDQ model for the 3 × 3 gate to showcase the fault tolerance capability. In the next section, we have analyzed the ALU reported by Sasamal et al. [15]. Table 7.1 shows existing ALU design in QCA framework.

7.2 Design of Arithmetic Unit and Logic Unit

Reversible ALU using emerging technology like QCA allows computing toward zero power dissipation. This module can be used for many high-end processing units. In this section, we have analyzed RALU by Sasamal et al. [15], which combine two separate blocks of reversible arithmetic and reversible logic unit. Authors have

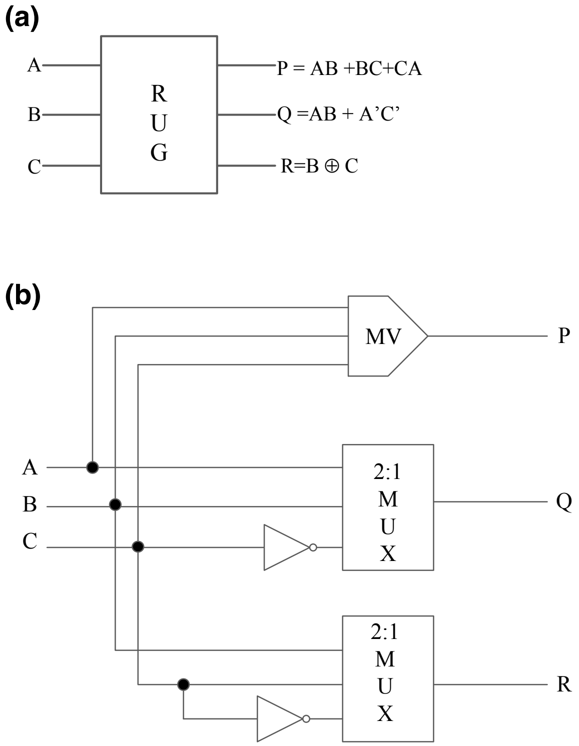


Fig. 7.2 **a** Reversible universal gate and **b** logic diagram of RUG

considered a universal gate; RUG for the synthesis of RALU. Figure 7.2b depicts logic diagram of RUG, and all associated relations between input and output are presented in Fig. 7.2a.

For synthesis of RUG, two 2:1 multiplexers, one MV3 and two inverters, are required. The QCA layout for the RUG is given in Fig. 7.3, where it composes 7 MV3 gates, 2 inverters. This single-layer structure needs 211 QCA cells with an effective area of 0.27 μm^2 and delay of 4 clock phases. A programmable RALU takes less power in respect of traditional ALU that can be configured by putting suitable logic on the selection lines. This helps in selecting which values to be send to the output terminal. The proposed RALU is based upon the instruction set instruction set architecture which insures maximum operations with minimum selection lines and enhances programmability. Moreover, this architecture follows modularity that encourages flexibility in the design process. Figure 7.4 completes the block diagram for RALU and comprises two main modules: (a) RLU and (b) RAU. It can be seen that the RUG has been used as the basis of all modules in addition to FG to facilitate fan-in at the input end. In order to select one of the module outputs, one additional RUG is used at the output.

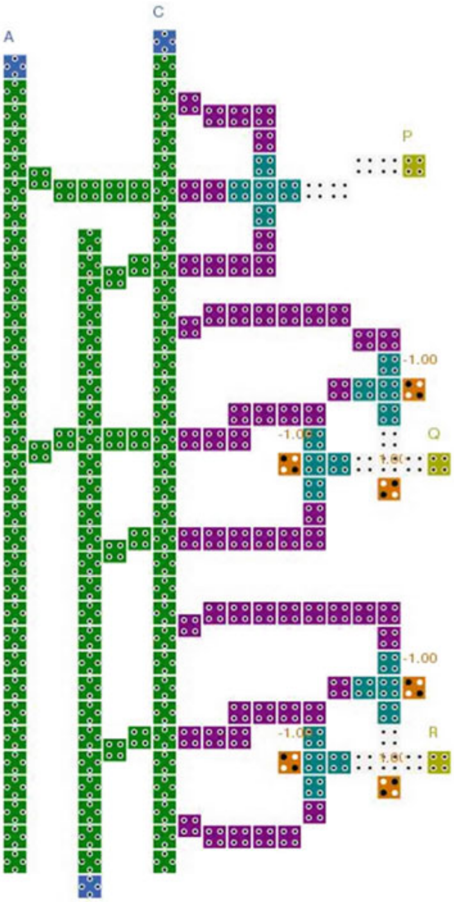


Fig. 7.3 QCA layout for RUG [15]

7.2.1 *Reversible Arithmetic Unit*

The reversible arithmetic unit (RAU) is shown in Fig. 7.5. It employs one Toffoli gate and two RUGs to provide operations such as increment, decrement, 2's complement subtraction. The presented RAU includes five inputs and five outputs out of which three are garbage outputs. This structure follows the instruction summarized in Table 7.2. It can be observed from the table that if control inputs vary from 000 to 111, different arithmetic functions can be realized. The cell layout for RAU design is shown in Fig. 7.6. For implementation, it needs 18 MV3 gates and clock phases of 16. Note, signal C1 is interpreted as Carry_input.

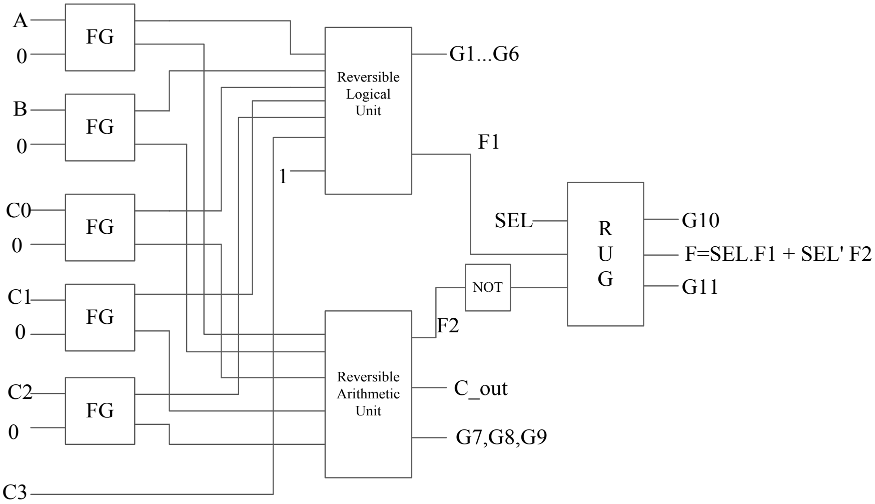


Fig. 7.4 Reversible arithmetic logic unit (ALU)

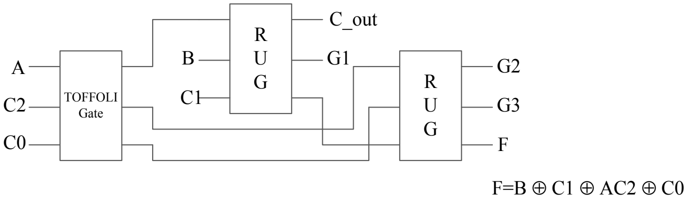


Fig. 7.5 Reversible arithmetic unit

Table 7.2 Reversible arithmetic unit functionality

$C0$	$C1$	$C2$	F	Operation
0	0	0	B	Transfer
0	0	1	$A + B$	Addition
0	1	0	$B + 1$	Increment B
0	1	1	$A + B + 1$	Addition with carry
1	0	0	$B + 1$	Increment B
1	0	1	$A + B$	1’s complement subtraction
1	1	0	$B - 1$	Decrement B
1	1	1	$A + B + 1$	2’s complement subtraction

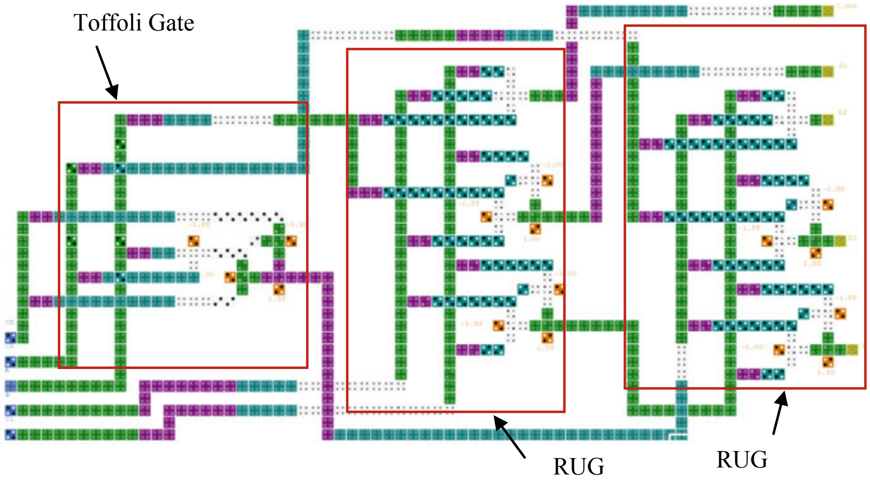


Fig. 7.6 Reversible arithmetic unit implemented in QCA

7.2.2 *Reversible Logic Unit*

The reversible logic unit (RLU) is shown in Fig. 7.7. It employs one Feynman gate and three RUGs to provide operations such as NOT, AND, NAND, OR. The presented RAU includes seven inputs and seven outputs out of which six are garbage outputs. This structure follows the instruction that summarized in Table 7.3. It can be observed from the table that if control inputs vary from 0000 to 1111, different logical functions can be realized. The cell layout for RLU design is shown in Fig. 7.8. For implementation, it needs 24 MV3 gates and clock phases of 12.

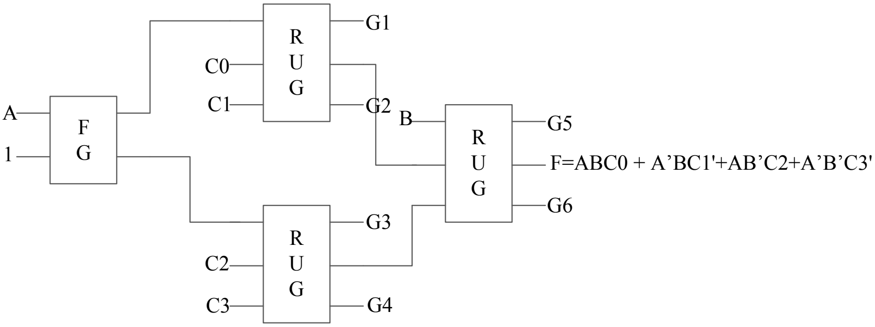


Fig. 7.7 Reversible logic unit

Table 7.3 Reversible logic unit functionality

$C0$	$C1$	$C2$	$C3$	F	Operation
0	0	0	0	A'	NOT
0	0	1	0	$(A \cdot B)'$	NAND
0	0	1	1	$(A \oplus B)$	XOR
0	1	0	0	$(A + B)'$	NOR
0	1	0	1	0	–
0	1	1	0	B'	NOT
1	0	0	1	B	COPY
1	0	1	0	1	Constant
1	0	1	1	$(A + B)$	OR
1	1	0	0	$(A \oplus B)'$	XNOR
1	1	0	1	$A \cdot B$	AND
1	1	1	1	A	COPY

7.3 Simulation and Discussion

All the designs are simulated and validated using QCADesigner-2.0.3 [16]. A summary of non-reversible and reversible ALU is cited in Tables 7.4 and 7.5, respectively. For comparison of RALU designs, we have considered QCA post layout parameters like delay, number of cells, effective area. Besides, we also examined reversible logic design parameters like garbage output and constant inputs. Table 7.4 lists the detailed comparison of non-reversible ALU designs with QCA technology. From Table 7.4, it is noted that the proposed design in [14] offers considerable advantages in terms of all QCA metrics. Particularly, this multilayer design has 17% improvements in cell counts and 25% improvement in delay and can perform more operations in comparison to the best design [9]. Table 7.5 lists the detailed comparison of reversible ALU designs with QCA technology. From Table 7.5, it is worth noting that the design in [15] offers more of number of operations; but at the cost of QCA area, we observed that the proposed design in [3] is suitable for reversible QCA implementation considering area, speed, and number of instructions in comparison with other best designs.

7.4 Summary

In this chapter, we first discussed different aspects of ALU in QCA. Further, we demonstrated an area-efficient reversible ALU which comprises two sub-modules, i.e., RLU and RAU. Both the sub-modules are synthesized using an existing gate (RUG). QCADesigner tool has been considered for design and simulation purposes.

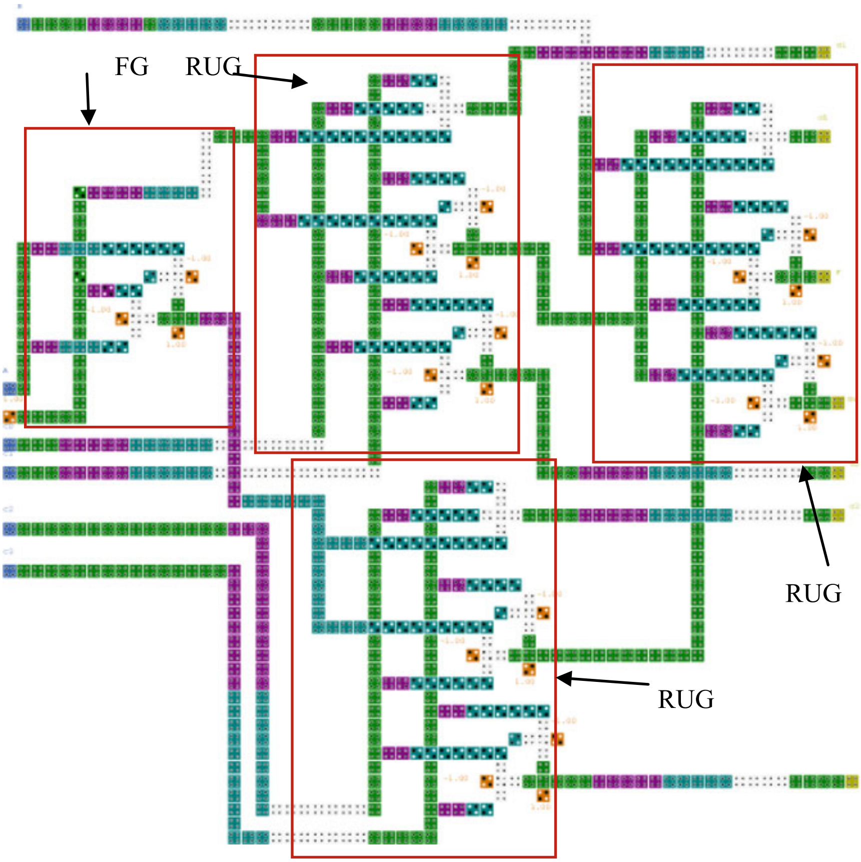


Fig. 7.8 Reversible logic unit implemented in QCA

Table 7.4 Comparison of non-reversible ALU designs with QCA technology

ALU designs	No. of operations	Cell counts	Area μm^2	Delay (clocks)	Crossover type
[10]	12	–	0.79	5	Multilayer
[9]	5	393	0.85	3	Coplanar
[11]	16	35596	11.37	9	Coplanar
[12]	4	430	0.78	3	Multilayer
[14]	12	324	0.245	2.25	Multilayer

Table 7.5 Comparison of reversible ALU designs with QCA technology

ALU designs	No. of operations	Cell counts	Area	Delay (clocks)	No. of garbage outputs	No. of constant inputs	Crossover type
[4]	12	647	0.624	3	20	3	Coplanar
[7]	4	1096	–	15	4	0	Multilayer
[15]	19	>2000	>5	5.25	11	6	Coplanar
[5]	16	2370	4.01	6	16	10	Coplanar
[6]	16	2370	4.44	6	11	10	Coplanar
[13]	14	–	2.34	2.25	8	0	Coplanar
[3]	16	670	–	4	2	1	Multilayer

References

1. R. Landauer, Irreversibility and heat generation in the computational process. IBM J. Res. Dev. **5**, 183–191 (1961)
2. Y. Syamala, A.V.N. Tilak, Reversible arithmetic logic unit. In: Electronics Computer Technology (ICECT), International Conference (2011)
3. A. Naghibzadeh, M. Houshmand, Design and simulation of a reversible ALU by using QCA cells with the aim of improving evaluation parameters. J. Comput. Electron. https://doi.org/10.1007/s10825-017-1004-9
4. A. Kamaraj, P. Marichamy, Design and implementation of arithmetic and logic unit (ALU) using novel reversible gates in quantum cellular automata, in *Proceedings of 4th International Conference on Advanced Computing and Communication System (ICACCS)*, pp. 1–8 (2017)
5. B. Sen, M. Dutta, D.K. Singh, D. Saran, B.K. Sidkar: QCA multiplexer based design of reversible ALU. In: *IEEE International Conference on Circuits and Systems (ICCAS)* (2012). doi:10.1109/ICCircuitsAndSystems.2012.6408309
6. B. Sen, M. Dutta, M. Goswami, B.K. Sidkar, Modular design of testable reversible ALU by QCA multiplexer with increase in programmability. Microelectron. J. (2014). https://doi.org/10.1016/j.mejo.2014.08.012
7. J.F. Chaves, D.S. Silva, V.V. Camargos, O.P.V. Neto, Towards reversible QCA computers: reversible gates and ALU. In: *IEEE 6th Latin American Symposium on Circuits & Systems (LASCAS)* (2015)
8. P. Gupta, A. Agrawal, N. Jha, *An algorithm for synthesis of reversible logic circuits* (Comput. Aided Des. Integr. Circuits Syst, IEEE Trans., 2006). https://doi.org/10.1109/TCAD.2006.871622
9. M.G. Waje, P.K. Dakhole, Design and implementation of 4-bit arithmetic logic unit using quantum dot cellular automata, in *Proceedings of 3rd IEEE International Advanced Computing Conference (IACC)* (2013), pp. 1022–1029
10. B. Ghosh, A. Kumar, A.K. Salimath, A simple arithmetic logic unit (12 ALU) design using quantum dot cellular automata. Adv. Sci. Focus **1**(4), 279–284 (2013)
11. V.C. Teja, S. Polisetti, S. Kasavajjala QCA based multiplexing of 16 arithmetic & logical subsystems—a paradigm for nano computing. In: *3rd IEEE International Conference on Nano/micro Engineered and Molecular Systems, 2008. NEMS 2008* (2008), pp 758–763 (2008)
12. M.R. Gadim, N.J. Navimipour, A new three-level fault tolerance arithmetic and logic unit based on quantum dot cellular automata. Microsyst. Technol. **24**(2), 1295–1305 (2018)
13. M. Goswami, B. Sen, R. Mukherjee, B.K. Sikdar, Design of testable adder in quantum-dot cellular automata with fault secure logic. Microelectron. J. **60**, 1–12 (2017)

14. S. Babaie, A. Sadoghifar, A.N. Bahar, Design of an efficient multilayer arithmetic logic unit in quantum-dot cellular automata (QCA). IEEE Trans. Circuits Syst. II Express Briefs **66**(6) (2019)
15. T.N. Sasamal, A.K. Singh, A. Mohan, Efficient design of reversible ALU in quantum-dot cellular automata. Optik **127**(15), 6172–6182 (2016)
16. K. Walus, T. Dysart, G. Jullien, R. Budiman, QCADesigner: a rapid design and simulation tool for quantum-dot cellular automata. IEEE Trans. Nanotechnol. **3**(1), 26–29 (2004)

Chapter 8
Design of Registers and Memory in QCA

In this chapter, we consider the design of sequential circuits in QCA. In particular, we presented the design of different *D* flip-flops and RAM cell with set and reset ability in QCA. The analysis for the proposed designs is carried out using rotated majority gate (MV3) and an efficient 5-input majority gate (MV5).

8.1 Introduction

Designing an efficient set of QCA-based flip-flops and memory cells at the logic and layout levels is one of the important aspects of implementing modern digital systems. Flip-flops are the primary circuit elements for realizing large-scale sequential circuits. In QCA, the functioning of the flip-flops is designed by triggering methods, the type of cells, and clocking schemes. Further, these flip-flops are used to demonstrate memory circuits, such as registers and random-access memories. The performance of all these elements is varied upon the complexity and input-to-output delay, which directly correlated to the performance of registers and memories. Hence, optimization can be done at their initial stages of development which is based on the selection of suitable basic building blocks. Herein, efficient set of majority gates are used to design optimized structures of register and RAM with proper clocking zone. Proposed designs utilize a modified 3-input and previously presented 5-input majority gate [1] to achieve a reduction in cell count and delay with least area occupancy.

8.2 Design of Registers

A register is formed by cascading flip-flops. This section presents different types of *D* flip-flops that determine the performance of registers. Various research works explore in this direction [1–12]. Level-triggered *D* flip-flop utilized both regular and

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_8

rotated QCA cells, while facing more delay in [2]. In [3], authors examined QCA layouts of different *D* flip-flops using less efficient 2:1 multiplexer. The overall delay of these designs can be reduced further by incorporating an optimal multiplexer. The authors in [4] presented a dual-edge-triggered *D* flip-flop that employs both regular and rotated cells. The design reported in [4] contains one-cell clocking zone, which may lead to incorrect results. Moreover, this design needs further improvement in input-to-output clock synchronization.

8.2.1 Design of D *Flip-Flops*

We present a new level- and edge-triggered QCA *D* flip-flops in this section. The QCA implementations of these designs are based on a 2:1 multiplexer which combines the benefits of MV3 and MV5 gates. The description of the presented structures is as follows.

Two compact schematics for a 2:1 multiplexer are presented in Fig. 8.1. The former one based on MV3 gates, while the later one uses MV5 and MV3 gates. QCA implementations of 2:1 multiplexer are depicted in Fig. 8.2a incorporate rotated and conventional 3-input majority gates. These designs deliver correct outputs in clock zone 1, i.e., after delay of 2 clock phases. In addition, these designs use 18 cells and QCA area of 0.016 μm^2. All inputs are in clock zone 0. The majority gates are kept in clock zone 0 and those implement *A*. Sel and *B*. Sel′. There is a change in zone for output majority gate that implements OR logic function. Thus, the resulted output is obtained in clock zone 1. The proposed 2:1 multiplexer is shown in Fig. 8.2b. This requires one MV3, an inverter at the input end and one MV5 gate [1] to generate the

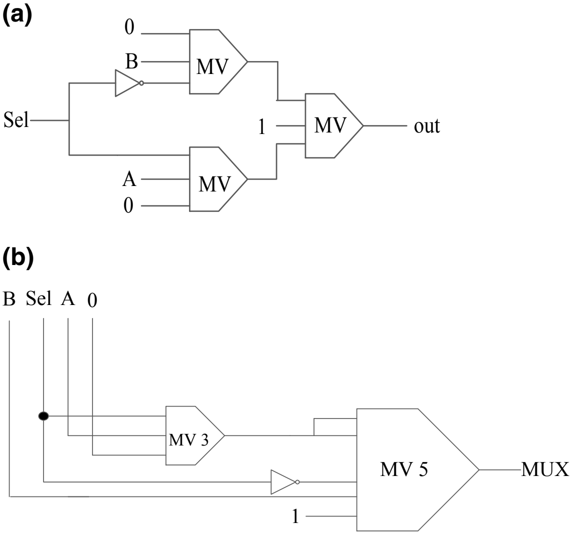


Fig. 8.1 Schematics of 2:1 multiplexer **a** MV3 gate-based, **b** MV3 gate-, MV5 gate-based

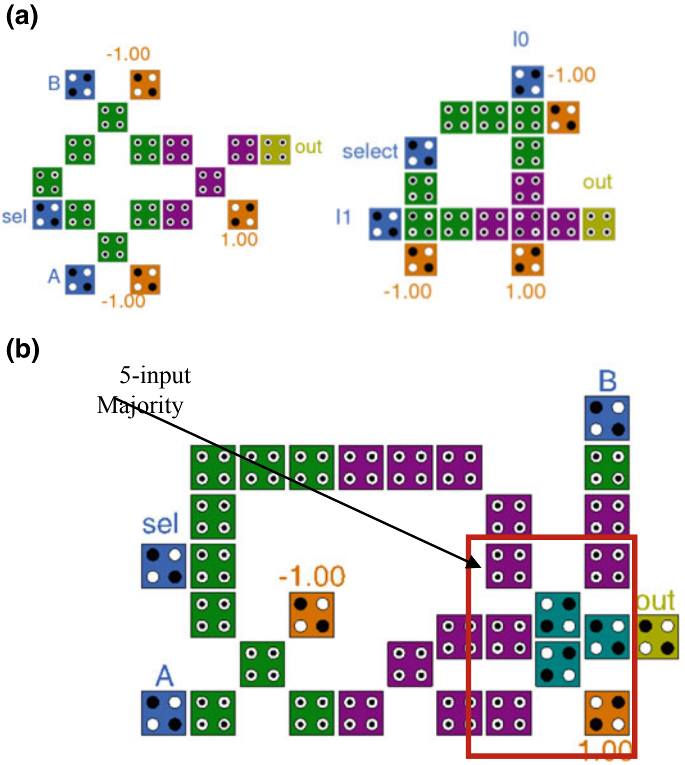


Fig. 8.2 QCA layouts of 2:1 multiplexer **a** MV3-based [12], **b** proposed

output. QCA layout of the proposed design has delay of 0.75 clock cycles, requires 32 cells, and resulted area of 0.04 μm^2.

QCADesigner-2.0.3 tool [13] is used for the simulation of the proposed layout. Corresponding simulation waveform is presented in Fig. 8.3. Result shows the correct operation of the 2:1 multiplexer and final output generated in clock 2. For instance in case of sel = '1', the layout outputs *S.A*, i.e., output follows input '*A*'. For simplicity, we have shown clock 2, which drives the output cell of the multiplexer, i.e., the output of the MV5 gate. Table 8.1 illustrates different QCA metrics of existing designs and presents layout. It is observed that the multiplexers are built upon MV3 offering more superiority than the design with MV5.

8.2.2 Design of Level-Triggered D *Flip-Flop*

A *D* flip-flop is one of the rudimental elements in the serial circuit design, which is used to store one bit of data and one of the fundamental structures that are suitable

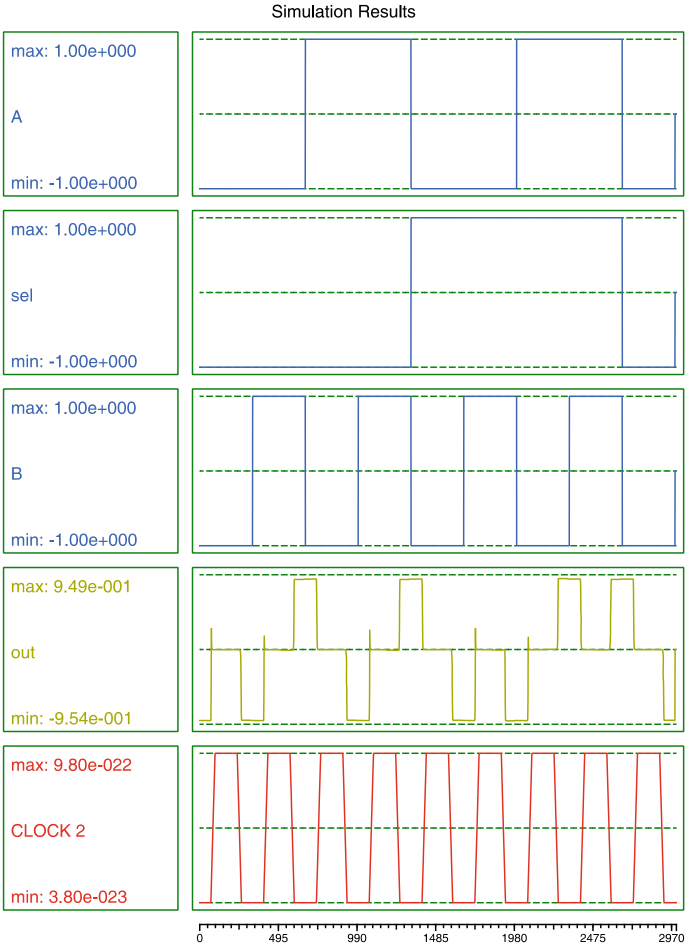


Fig. 8.3 Simulation diagram for proposed 2:1 multiplexer using MV3 and MV5

Table 8.1 Comparison results of QCA multiplexers

2:1 multiplexer	Area (μm^2)	Cell count	Delay (clock cycles)	Crossover type
[5]	0.14	88	1	Coplanar
[6]	0.14	66	1	Coplanar
[7]	0.08	46	1	Multilayer
[8]	0.07	56	1	Coplanar
[9]	0.06	36	1	Multilayer
[10]	0.03	27	0.75	Coplanar
[11]	0.02	26	0.5	Coplanar
[12]	0.016	18	0.5	Coplanar
Proposed	0.04	32	0.75	Coplanar

Fig. 8.4 Schematic of 2:1 multiplexer-based level-triggered *D* flip-flop

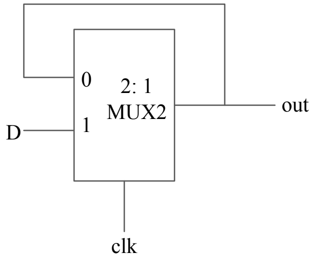


Table 8.2 Operation of QCA-based level-triggered D flip-flop

Clock (clk)	*D*	*Q*(output)
0	X	$Q(t\ -1)$
0	X	$Q(t\ -1)$
1	0	0(input)
1	1	1(input)

for use in shift registers. A level-triggered *D* flip-flop has two inputs (*D* and CLK), an output (out) where output depends both on the current input '*D*' in addition to the past output. The block diagram of *D* flip-flop is shown in Fig. 8.4 using 2:1 multiplexer. The level-triggered *D* flip-flop is governed by equation CLK. $D +$ CLK$'$ out, where 'out' represents the present state of *D* flip-flop. Based on Table 8.2, when the 'clk' = '1', the input '*D*' will propagate to output 'Q_t'; otherwise, the output is not altered (Q_{t-1}). QCA implementation for level-triggered QCA *D* flip-flop using rotated MV3 is shown in Fig. 8.5a [12]. Cellular representation of the proposed structure is depicted in Fig. 8.5b and requires a MV5, rotated MV3, and an inverter. According to this figure, the proposed implementation has coplanar design using regular QCA cells without any wire crossing, which involves two levels of majority gates. First-level MV3 gate enables one AND gate that is driven by clock zone 0, whereas the second-level MV5 gate receives inputs as 'clk' and intermediate output of the first stage majority gate. The final output of the design, i.e., output cell of MV5, is assigned to clock zone 2. A chain of QCA cells is added at the output that creates a feedback path with four consecutive clocking zones. This helps in storing 1-bit of data *until* clk= '0'. The number of cells which are used in this structure is 36 cells in 0.044 μm^2 *area.* The signal delay of the presented design is 3 clock phases to deliver correct outputs. The simulation result of the presented structure is illustrated in Fig. 8.6. Results indicate that the proposed QCA layout for level-triggered *D* flip-flop performs correctly. Output waveform is obtained accurately after 0.75 clock cycle delay. This figure also demonstrates when the clock is active, the output will be equivalent to *D* input, and otherwise, the output is equal to the stored value in the loop.

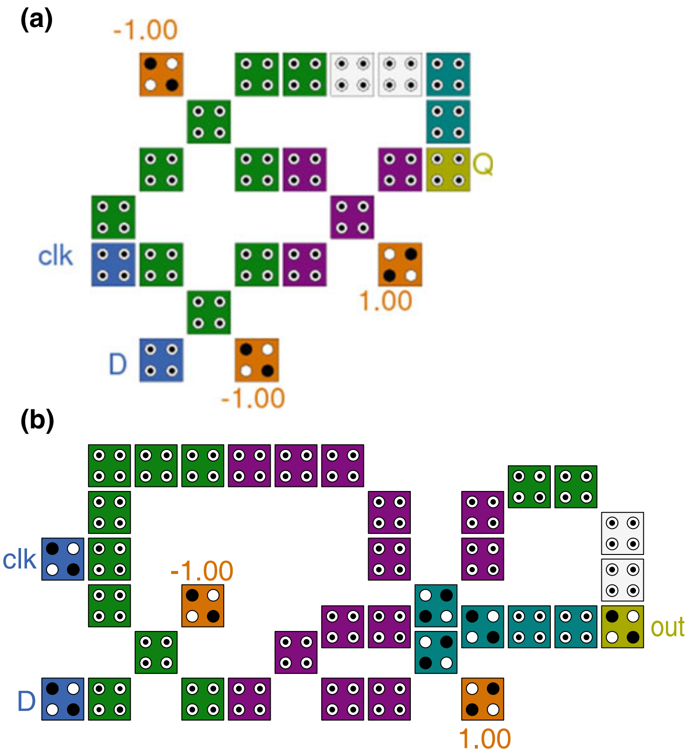


Fig. 8.5 Proposed level-triggered *D* flip-flops in QCA **a** using 3-input majority gate [12], **b** using 5-input majority gate

8.2.3 Design of Positive Edge-Triggered D *Flip-Flop*

Yang et al. [4] reported a structure for edge-sensitive flip-flop which is depicted by a rectangular box in Fig. 8.7. According to authors, this unit employs one MV3 gate and an inverter. The MV3 gate takes a clock input, delay version of clock input and implements a 2-input AND gate.

The operation of the presented positive edge-triggered *D* flip-flop is illustrated in Table 8.3. According to the table, input is transferred to the output when the intermediate signal becomes '1' that represents rising edge of the clock signal. QCA layout for the proposed structure is shown in Fig. 8.7b, and utilizes a MV5 and two rotated MV3 gates with an inverter. The design has single-layer QCA regular cells which involve three layers of majority gates. First layer enables positive edge clock signal, while the second and third layers constitute a *D* flip-flop. The number of cells which are used is 59 cells, and the area occupied is 0.06 μm^2. The signal delay of the presented design is 2 clock cycles to deliver correct outputs.

Figure 8.8 demonstrates the precise operation of the presented rising edge-sensitive *D* flip-flop with input and output waveforms. Result indicates when rising

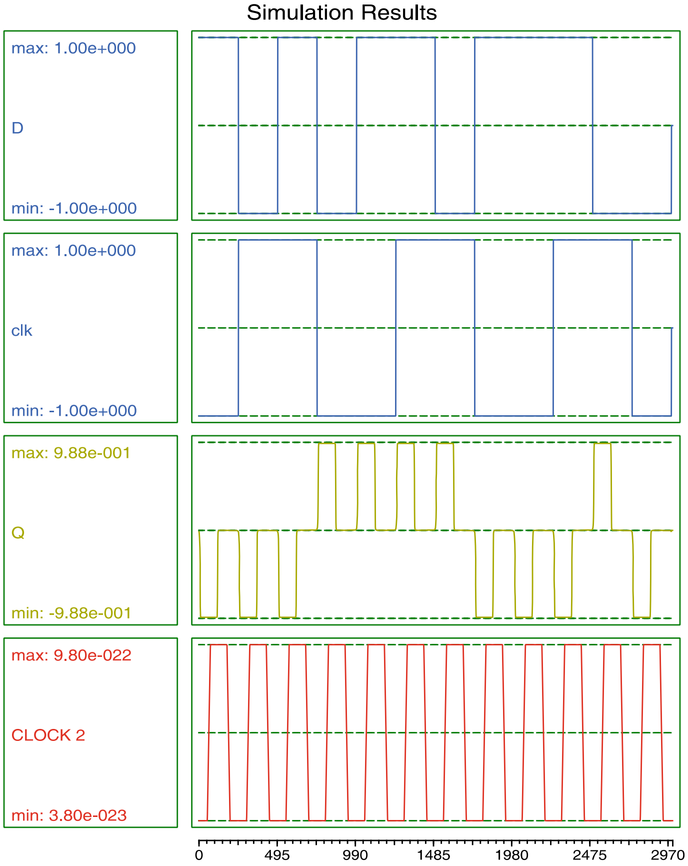


Fig. 8.6 Simulation results of the proposed level-triggered *D* flip-flops

edge of clock is active, the output will be equivalent to *D* input, and otherwise, the output is equal to the stored value in the loop. Output waveform is obtained accurately after 2 clock cycles' delay.

8.2.4 Design of Negative Edge-Triggered D *Flip-Flop*

QCA layout for presented falling edge *D* flip-flop structure is depicted in Fig. 8.9b. It comprises a MV5 and two rotated MV3 gates with an inverter three. The design has QCA regular cells which involves three layers of majority gates. First layer enables positive edge clock signal, while second and third layers constitute a *D* flip-flop. This structure has a clock input (clk), a data input '*D*', and an output from the last stage. The operation of the presented *falling edge*-triggered *D* flip-flop is illustrated

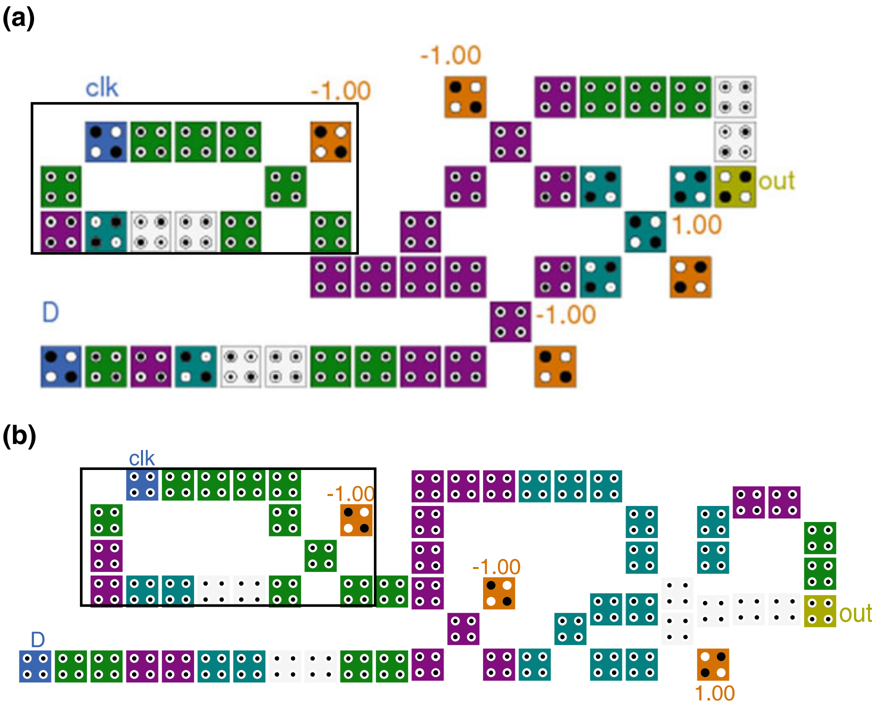


Fig. 8.7 QCA implementation of positive edge-triggered *D* flip-flops **a** [12], **b** proposed

Table 8.3 Operation of QCA-based positive edge-triggered *D* flip-flop

clk (t)	clk ($t-1$)	$\overline{\mathrm{clk}(t-1)}$	I	Output ($Q(t)$)
0	0	1	0	$Q(t-1)$
0	1	0	0	$Q(t-1)$
1	0	1	1	D(input)
1	1	0	0	$Q(t-1)$

in Table 8.4. According to Table 8.4, ‘I’ is the intermediate output of the first level of MV3 gate implementing *an AND gate which takes* $\overline{\mathrm{clk}(t)}$ and clk($t-1$). When the intermediate signal becomes ‘1’ that represents rising edge of the clock signal, the value of input data (D) will be sent to output. The number of used cells in the presented structure is 59 cells with a total area of 0.06 $\mu\mathrm{m}^2$ and latency of 2 clock cycles. Proposed falling edge *D* flip-flop has been implemented at a single layer. Also, it is worth noting that all the signals are properly synchronized with at least two cells in one clock zone.

Figure 8.10 demonstrates the correct operation of the presented falling edge-triggered *D* flip-flop with input and output waveforms. Different binary values are considered at input ‘*D*’. When the clock is active, i.e., during falling edge, the output

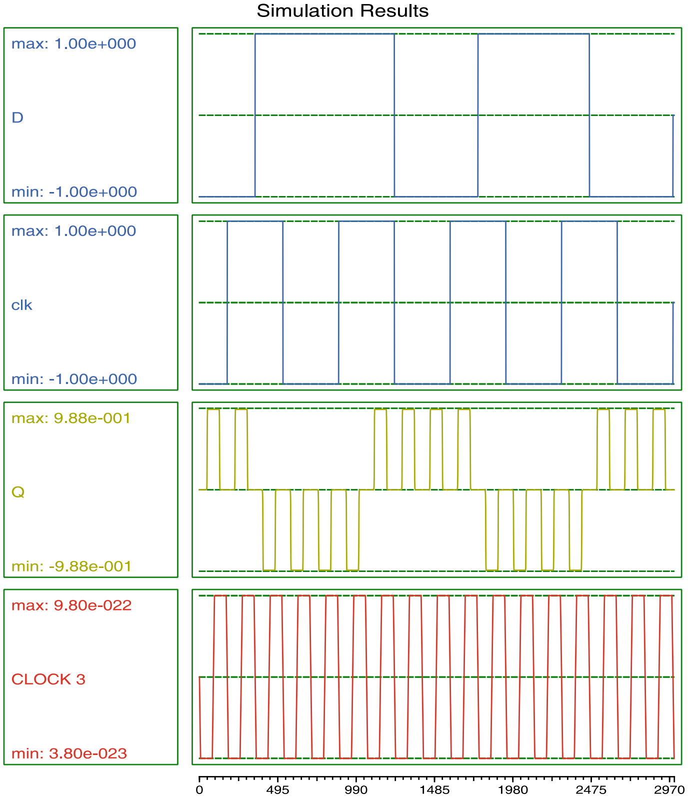


Fig. 8.8 Simulation results of the proposed positive edge-triggered *D* flip-flops

will be equivalent to *D* input, and otherwise, the output is equal to the stored value in the loop. Output waveform accurately is obtained after 2 clock cycles' delay.

8.2.5 Design of Dual-Edge-Triggered D *Flip-Flop*

QCA layout for the proposed structure is given in Fig. 8.11b. The design employs two edge-sensitive blocks for both rising and falling edge. The operation of the new design is specified in Table 8.5. It constitutes four levels of majority operations. MV3's at level one has a clock input (clk), inverted clock input, and clock input with one clock cycle delay (Fig. 8.11). The negative edge-triggered unit results intermediate signal I_1, while negative edge trigger unit results output I_2. The second-level MV3 gate is dedicated for the OR gate to output $I_3 = I_1 + I_2$ and decides the kind of edge-triggered. When $I_3 =$ '1', rising or falling edge of the clock is achieved and

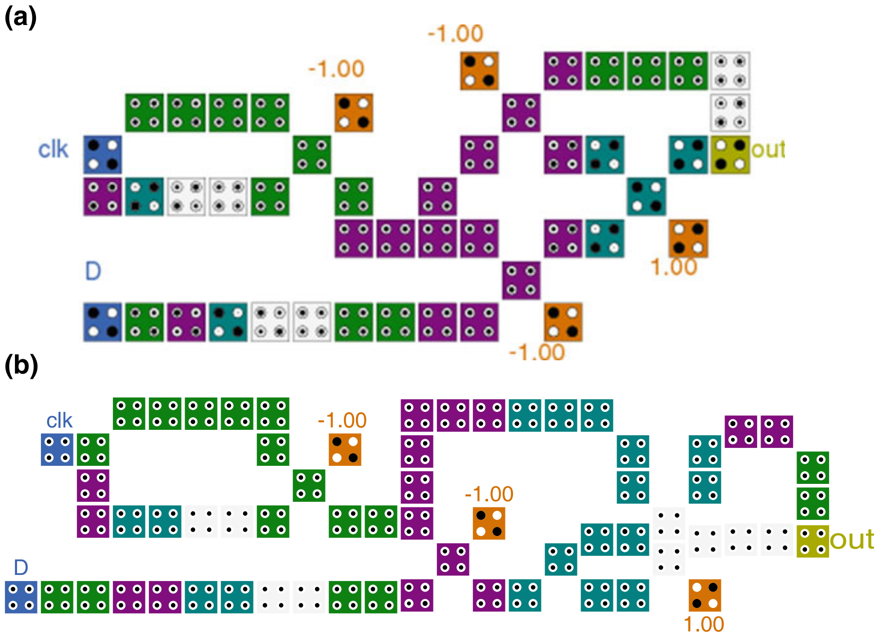


Fig. 8.9 QCA implementation of the negative edge-triggered *D* flip-flops **a** in [12], **b** proposed

Table 8.4 Operation of QCA-based negative edge-triggered *D* flip-flop

clk (t)	$\overline{\text{clk}(t)}$	clk ($t - 1$)	I	Output ($Q(t)$)
0	1	1	1	*D*(input)
0	1	0	0	$Q(t - 1)$
1	0	1	0	$Q(t - 1)$
1	0	0	0	$Q(t - 1)$

corresponding input ‘*D*’ can be transferred to output using third- and fourth-level majority gates those constitute a level-triggered *D* flip-flop.

The presented design utilizes four rotated MV3 gates and one MV5 and requires 91 cells. The QCA area of the layout is 0.11 μm^2 and delay of 9 clock zones. Figure 8.12 depicts QCADesigner waveform of the proposed *D* flip-flop. Result verifies the operations listed in Table 8.5. It also indicates that both the edges of clk signal are considered for transmitting the input ‘*D*’, and the output ‘*Q*’ is achieved after 9 clock zones.

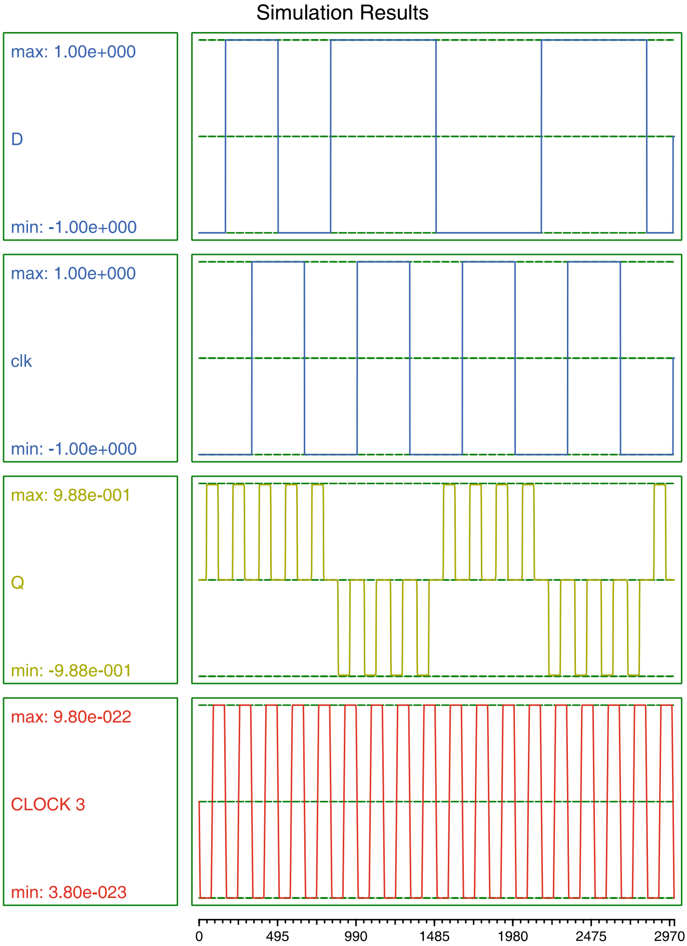


Fig. 8.10 Simulation results of the proposed negative edge-triggered *D* flip-flops

8.2.6 *Comparison Results*

The simulations were used to study structural complexity between proposed designs and existing designs. Table 8.6 compares the proposed QCA-based *D* flip-flops and the previously reported flip-flops designs. The most commonly discussed metrics in the literature are the cell count, consumed area, and input-to-output delay. According to Table 8.6, the proposed QCA-based *D* flip-flops have better performance than existing ones [3, 14]. However, the *D* flip-flop circuits in [12] provide better performance in terms of area, cell count, and delay.

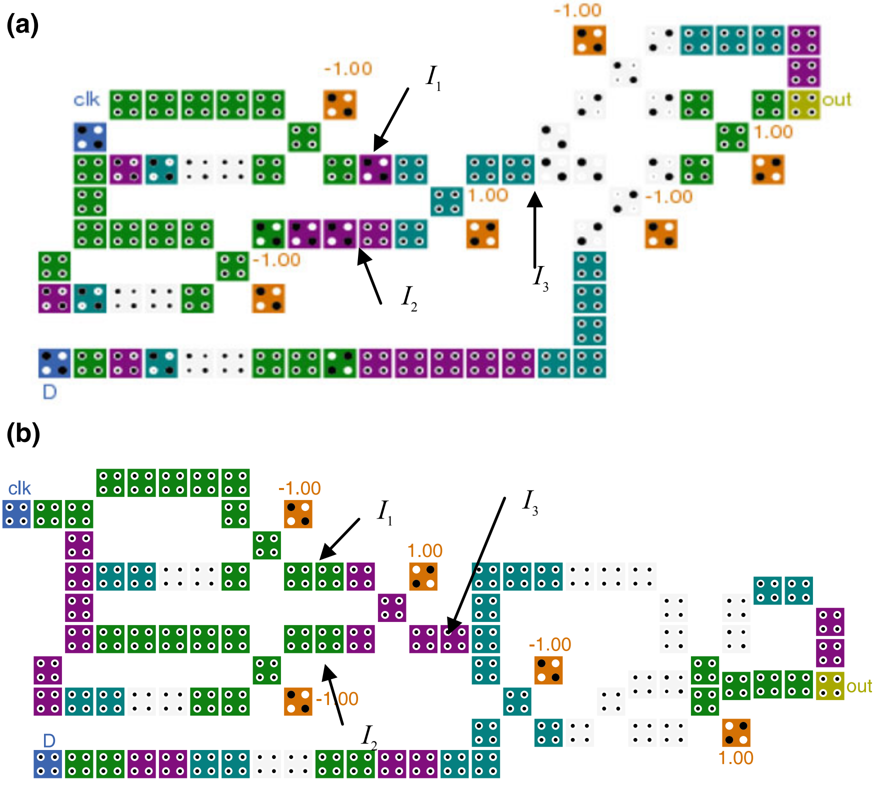


Fig. 8.11 QCA implementation of the proposed dual-edge-triggered D flip-flops **a** in [12], **b** proposed

Table 8.5 Dual-edge-triggered D flip-flop operational table

clk (t)	$\overline{\text{clk}(t-1)}$	I_1	$\overline{\text{clk}(t)}$	clk (t − 1)	I_2	I_3 = (MV(I_1, I_2, 1)	$Q(t)$ (output)
0	0	0	1	1	1	1	D (input)
0	1	0	1	0	0	0	$Q(t-1)$
1	0	0	0	1	0	0	$Q(t-1)$
1	1	1	0	0	0	1	D (input)

8.3 Design of Memory in QCA

A random-access memory (RAM) is one of the basic forms of memories designed to store and retrieve the data from any internal locations. So, constructing an efficient, dense, and simple QCA memory structure is of great importance where performance is varied upon the selection of majority gates and proper clocking zone.

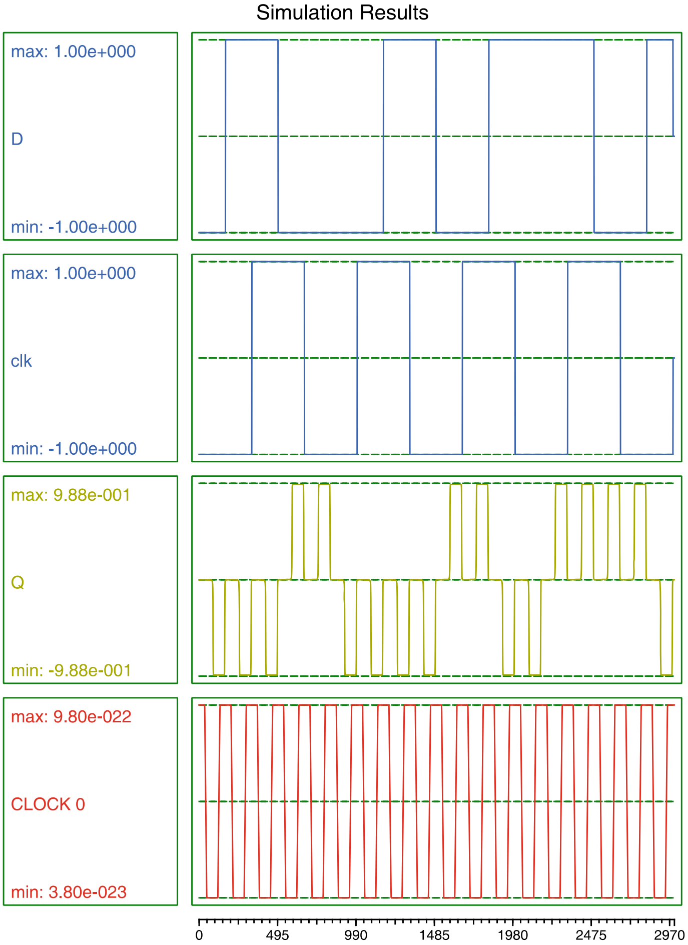


Fig. 8.12 Simulation waveforms of the dual-edge-triggered *D* flip-flop

QCA-based RAM cell is structurally differentiated by two mechanisms: (1) line-based; (2) loop-based. A line-based RAM structures need three consecutive clocking zones as shown in Fig. 8.13a, such that one of the three zones is in hold phase. It enables to retain one memory state at a given time. In the former structure, a loop with four consecutive clock zones is considered to store 1-bit of data, as depicted in Fig. 8.13b. Here, QCA cells x; y; z; c; o constitute a majority gate with inputs x; y; z, center cell c and output o. To retain the data in the memory cell, the inputs A, B to respective cells x; y are assigned 0; 1 or 1; 0, which allows the majority gate to compute the stored data bit. Loop-based RAM cell has received high attention from many designers because it requires fewer number of clock phases, whereas line-based RAM cell uses additional clock zones with increase in complexity.

Table 8.6 Comparison of QCA *D* flip-flops structure

		Area (μm^2)	Cell count	Clock delay phases	Crossover type	Majority gate types
QCA *D* flip-flop (level-triggered)	[15]	0.08	66	1.5	Coplanar[c]	OMG[a]
	[3]	0.05	48	1	Coplanar[c]	OMG[a]
	[12]	0.02	23	0.5	Coplanar[c]	RMG[b]
	Proposed	0.044	36	0.75	Coplanar[c]	RMG[b], New 5-input majority gate
Positive edge-triggered QCA *D* flip-flop	[3]	0.09	84	2.75	Coplanar[c]	OMG[a]
	[12]	0.04	47	1.75	Coplanar[c]	RMG[b]
	Proposed	0.077	63	2	Coplanar[c]	RMG[b], New 5-input majority gate
negative edge-triggered QCA *D* flip-flop	[3]	0.09	84	2.75	Coplanar[c]	OMG[a]
	[12]	0.04	47	1.75	Coplanar[c]	RMG[b]
	Proposed	0.077	64	2	Coplanar[c]	RMG[b], New 5-input majority gate
QCA dual-edge *D* flip-flop	[4]	0.16	116	3	Coplanar[c]	OMG[a]
	[3]	0.14	120	3.25	Coplanar[c]	OMG[a]
	[12]	0.1	81	2.25	Coplanar[c]	RMG[b]
	Proposed	0.119	91	2.25	Coplanar[c]	RMG[b], New 5-input majority gate

[a] OMG: Original 3-Input Majority Gate
[b] RMG: Rotated Majority Gate
[c] Designs can be constructed using 90° and 45° QCA cells

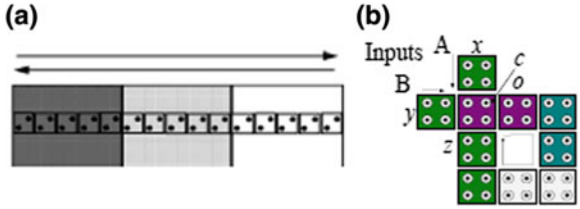


Fig. 8.13 Memory cell **a** line-based, **b** loop-based

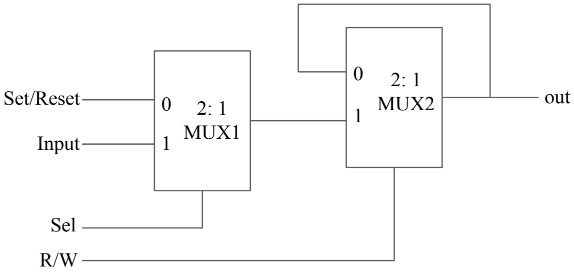
The two types of QCA-based RAM cell designs can be found in [3, 4, 14–18], respectively. In [14], authors presented a RAM cell lacking set/reset signal. Their design requires both regular and rotated cells for the implementation.

The RAM structure in [16] is not well-optimized in terms of area utilization and delay. A multiplexer-based RAM cell is constructed in [2], but this design requires more optimal multiplexer to reduce the overall complexity. Vankamamidi et al. reported a memory cell that employs larger number of cells and a higher latency [17]. Authors in [18] proposed a multilayer implementation of RAM cell but multilayer design offers high cost, for instance, due to fabrication issue.

8.3.1 Novel RAM Cell Design

Novel designs of QCA-based random-access memory (RAM) cell are examined in this section which offers a modest reduction in the cell count, area usage, and computation delay. The proposed RAM unit structure is illustrated in Fig. 8.14 which combines the benefits of 2:1 multiplexer, with set/reset attribute. Corresponding layout is drawn in Fig. 8.15. For implementation, this employs two MV5 gates and two rotated MV3 gates. First level of multiplexer (Mux1) is used to select 'input'

Fig. 8.14 Schematic design of the memory cell



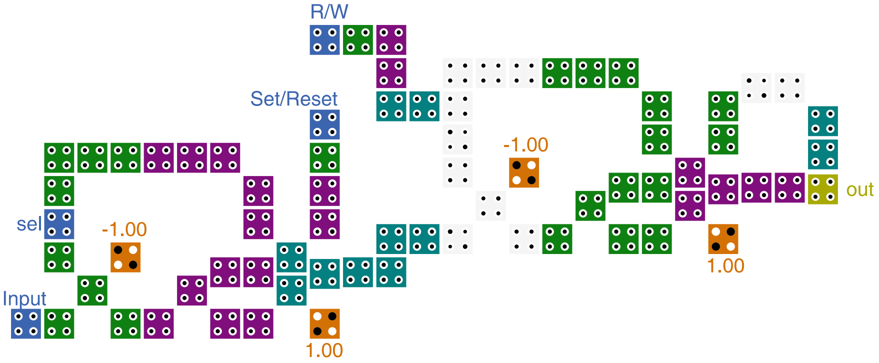


Fig. 8.15 QCA implementation of the proposed memory cell with set/reset signal

Table 8.7 Function table of QCA-based memory cell with set/reset ability

Read/Write(R/W)	Select(Sel)	Set/Reset	Output (out(t))
1	1	X	Input
1	0	0	0(reset)
1	0	1	1(set)
0	X	X	out(t $-$ 1)

or ‘set/reset’ by placing appropriate value on ‘Sel’ line. Second level of multiplexer (Mux2) is used to select previous output or Mux1 output by setting ‘R/W’ line. The functionality of RAM cell is summarized in Table 8.7. In particular, when the ‘R/W’ is set to ‘1’, the value of inputs (input or set/reset) will be sent to output (out), and if it is deactivated, the output is not changed, i.e., out(t) = out(t − 1). Further, setting R/W = ‘1’ and Sel= ‘1’ result write mode and the new input data transferred to the output. If the select (Sel) signal is set to ‘0’, the output of the circuit changes with respect to ‘set/reset’ signal. For the design, the output cell is assigned to clock zone 1 and a correct output is obtained after 6 clock zones.

8.3.2 Results and Discussion

The output result of the proposed RAM cell with set/reset ability is demonstrated in Fig. 8.16 that provides input and output waveforms due to different clock zones. This indicates in normal mode that the inputs are transferred to the output after 1.25 clock delays by setting both ‘Sel’ and ‘R/W’ signals to ‘1’. Further, by making ‘set/reset’ signal to ‘0’ or ‘1’, the corresponding output is set or reset respectively with delay of 6 clock zones. Furthermore, Fig. 8.16 confirms that the design for RAM cell operates correctly and provides expected outputs after 1.5 clock cycle delay.

Table 8.8 compares the proposed QCA-based RAM cell and the prior reported RAM designs. The most commonly discussed metrics in the literature are a number of QCA cells, area usage, number of control cells, area delay product (ADP), and input-to-output latency. From Table 8.8, we can refer that proposed RAM cell has the lowest delay among all existing designs which can be attributed due to the optimization of the majority gate. It is worth noting that the presented structure provides better performance in terms of area and cell count than the design in [3] with set/reset attribute. In particular, our design achieves 14 and 32% improvement in latency and cell count than the design in [3]. It is also observed that the RAM structure requires only 4 control cells, which is the same as reported in the literature [14, 18]. All designs are based on coplanar structures that provide good chances to increase the performance particularly for designing complex QCA circuits or even may be used as a basic building block for nanoscale registers and counters.

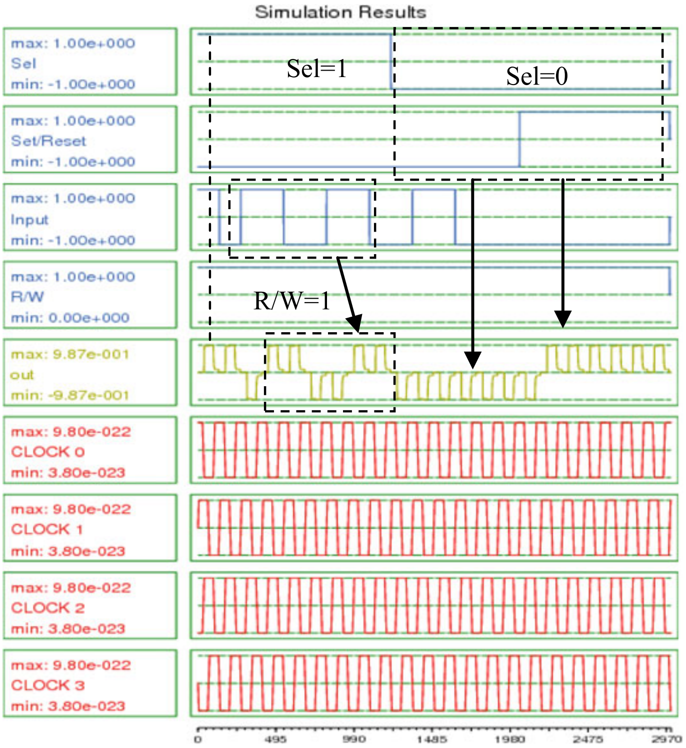


Fig. 8.16 Simulation waveforms of the proposed memory cell with set/reset ability

8.4 Summary

In this chapter, new QCA layouts for *D* flip-flops and memory cell are explored. We began with a novel 2:1 multiplexer which is used as the fundamental block of all proposed structures. Analysis of 2:1 Multiplexer has been made considering a compact MV5 and rotated MV3 gates. Results indicate that the offered designs outperform previous designs in respect of QCA parameters.

Table 8.8 Comparison results of QCA RAM cell structures

QCA memory cell	Crossover type	Set/reset ability	Area (μm^2)	Cell count	Control cells	Delay	Clock phases	ADP	Majority gate types
[17]	Coplanar[c]	No	0.31	233	5	1.75	7	0.543	OMG[a]
[18]	Multilayer	No	0.25	173	4	1.5	6	0.375	OMG[a]
[16]	Coplanar[c]	No	0.16	158	6	2	8	0.32	OMG[a]
[14]	Coplanar[c]	No	0.11	100	5	3	12	0.33	OMG[a]
	–	No	0.07	63	4	2	8	0.14	
[3]	–	Yes	0.13	109	6	1.75	7	0.227	OMG[a]
Proposed	–	Yes	0.129	74	4	1.5	6	0.187	RMG[b], New 5-input majority gate

[a] OMG: original 3-input majority gate

[b] RMG: rotated majority gate

– Designs do not need any crossover wire and can be constructed using 90° QCA cells

[c] Designs can be constructed using 90° and 45° QCA cells

References

1. R. Jaiswal, T.N. Sasamal, Efficient design of exclusive-OR gate using 5-input majority gate in QCA. IOP Conf. Ser.: Mater. Sci. Eng. **225**, 012143 (2017)
2. S. Angizi, S. Sarmadi, S. Sayedsalehi, Design and evaluation of new majority gate-based RAM cell in quantum-dot cellular automata. Microelectron. J. **46**, 43–51 (2015)
3. S. Hashemi, K. Navi, New robust qca d flip-flop and memory structures. Microelectron. J. **43**(12), 929–940 (2012)
4. X. Yang, L. Cai, X. Zhao, Low power dual-edge triggered d flip-flop structure in quantum dot cellular automata. Electron. Lett. **46**(12), 825–826 (2010)
5. K. Kim, K. Wu, R. Karri, The robust QCA adder designs using composable QCA building blocks. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **26**(1), 176–183 (2007)
6. V. Mardiris, C. Mizas, L. Fragidis, V. Chatzis, Design and simulation of a qca 2 to 1 multiplexer, in *Proceedings of the 12th WSEAS International Conference on Computers, ser. ICCOMP'08* (World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2008), pp. 572–576
7. T. Teodosio, L. Sousa, QCA-LG: a tool for the automatic layout generation of QCA combinational circuits, in *Norchip 2007*, Nov 2007, pp. 1–5
8. V.A. Mardiris, I.G. Karafyllidis, Design and simulation of modular 2^n to 1 quantum-dot cellular automata (QCA) multiplexers. Int. J. Circuit Theory Appl. **38**(8), 771–785 (2010)
9. S. Hashemi, M.R. Azghadi, A. Zakerolhosseini, A novel QCA multiplexer design, in *2008 International Symposium on Telecommunications*, Aug 2008, pp. 692–696
10. A. Roohi, H. Khademolhosseini, S. Sayedsalehi, K. Navi, A novel architecture for quantum-dot cellular automata multiplexer. Int. J. Comput. Sci. **8**, 55–60 (2011)
11. R. Sabbaghi-Nadooshan, M. Kianpour, A novel QCA implementation of MUX-based universal shift register. J. Comput. Electron. **13**(1), 198–210 (2014)
12. T.N. Sasamal, A.K. Singh, U. Ghanekar, Design of QCA-based *D* flip flop and memory cell using rotated majority gate. In: B. Panigrahi, M. Trivedi, K. Mishra, S. Tiwari, P. Singh (eds) *Smart Innovations in Communication and Computational Sciences. Advances in Intelligent Systems and Computing*, vol. 670 (Springer, Singapore, 2019)
13. K. Walus, T.J. Dysart, G.A. Jullien, R.A. Budiman, Qcadesigner: a rapid design and simulation tool for quantum-dot cellular automata. IEEE Trans. Nanotechnol. **3**(1), 26–31 (2004)
14. M.A. Dehkordi, A.S. Shamsabadi, B.S. Ghahfarokhi, A. Vafaei, Novel ram cell designs based on inherent capabilities of quantum-dot cellular automata. Microelectron. J. **42**(5), 701–708 (2011)
15. A. Vetteth, K. Walus, V.S. Dimitrov, G.A. Jullien, *Quantum-Dot Cellular Automata of Flip-Flops*, vol. 2500 (ATIPS Laboratory, 2003
16. K. Walus, A. Vetteth, G.A. Jullien, V.S. Dimitrov, Ram design using quantum-dot cellular automata, in *Technical Proceedings of the 2003 Nanotechnology Conference and Trade Show* (2003), pp. 160–163
17. V. Vankamamidi, M. Ottavi, F. Lombardi, A line-based parallel memory for QCA implementation. IEEE Trans. Nanotechnol. **4**(6), 690–698 (2005)
18. B. Taskin, B. Hong, Improving line-based QCA memory cell design through dual phase clocking. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **16**(12), 1648–1656 (2008)

Chapter 9
Clocking Schemes for QCA

One aspect that is important for QCA circuits is clocking schemes. For the advancement of QCA technology, it is necessary to enable the specification of standard cells, the development of placement and routing algorithms. This chapter discusses the different arrangements of underlying clocking circuit onto which a QCA circuit is overlaid.

9.1 Introduction

QCA clocking is used for information synchronization and also to provide necessary power during computation. It generally constitutes multiphase. In QCA architecture, an individual cell is not clocked separately, rather group of cells as an array are clocked simultaneously by a clock zone. Different arrays are timed in pipelining manner such that an array performs certain computation and act an input to the succeeding array. Multiphase clocking plays an important role in logic computation. It controls the height of interdot barrier of a cell which allows an array of cell to hold the logic value when the successor array of cells during computation. In addition, when an array performs computation, the successor array must be in release state so that it will not interfere with ongoing computation. The next section discusses various clocking schemes.

9.2 Clocking Schemes

Several clocking mechanisms for QCA circuits have been discussed in [1–5]. The simplest form of clock arrangement is discussed in [1]. Where, the information propagation in QCA designs is done along one direction as a 1-D technique, as shown in Fig. 9.1. For clocking the QCA cells in 1-D fashion four phases (adiabatic

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_9

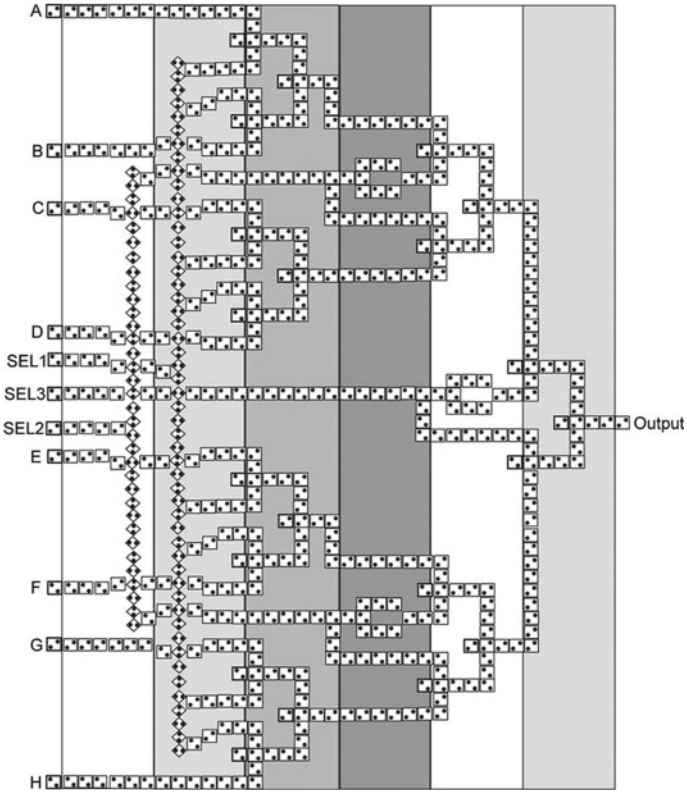


Fig. 9.1 1-D clocking [1]

switching) are required. This technique partitions a design into different zones only along the *x*-axis where the clock zones are vertical and non-uniform width. The data runs horizontally and there exist no feedback paths through the QCA clock zones. The height of a clock zone depends on the complexity of the design, whereas the lengths of horizontal lines are limited by the width of zones. As the width of the clocking zone decides the clock rate, the clock rate for a narrow clock zone is faster than a larger width clock zone.

Janez et al. framed a design rule and floorplan that consider uniform and regular clocking zones; simplifying manufacturability [2]. Their work employs a unidimensional arrangement of clock zones for only combinational circuits. However, authors were not adopted any clocking scheme.

Following these limitations, authors in [3] introduced two clocking schemes: "trapezoidal clocking" and universal clocking cell as depicted in Fig. 9.2a, b. Though the logic propagates in one direction from one zone to another, this clocking scheme allows QCA wires to form a feedback loop. It is worth noting that a signal is traveled

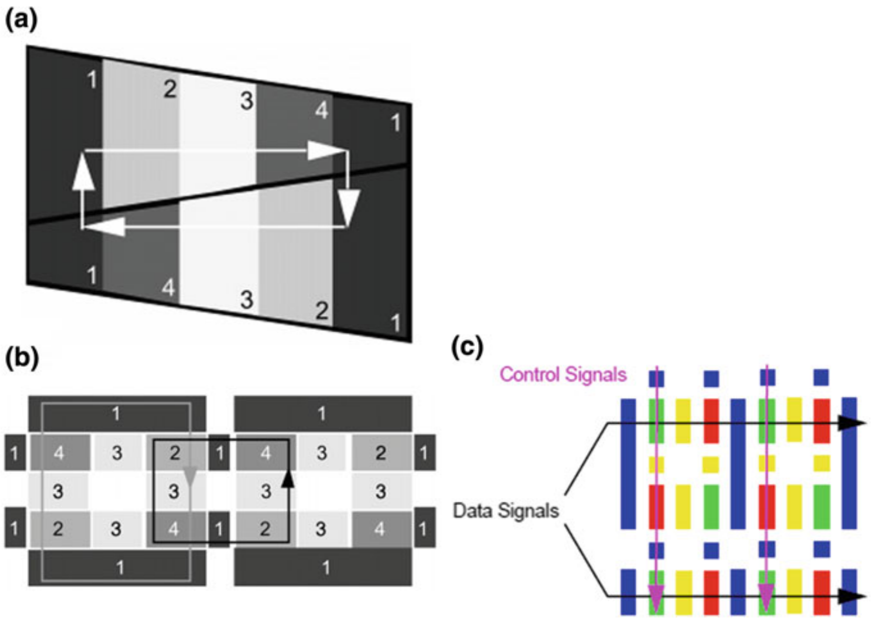


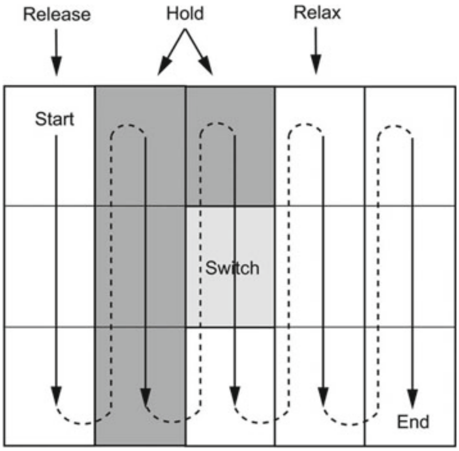
Fig. 9.2 **a** Trapezoidal clocking, **b** universal clocking cell, and **c** universal clocking floorplan [3]

properly with a feedback path. This floorplan also reduces wasted area. Furthermore, a universal clocking floorplan is designed for complicated QCA architecture that allows multiple loops and feedback paths through the QCA clocking phases as illustrated in Fig. 9.2c. However, no proper clocking circuitry is defined for both the clocking mechanisms. In addition, both models fall short since the sizes of the clocking zones are not uniform.

Vankamamidi et al. [4] devised two new schemes to overcome the shortcomings with 1-D clocking techniques. A given QCA design is partitioned along *y*-axis and *x*-axis, which restricts long vertical lines. The first scheme was based on a 2-D partitioning of the design into a grid of zones. The QCA designs under 2-D schemes are robust to thermal fluctuations and can be operated at higher temperatures, mostly independent of size. In general, for the proper functioning of QCA design, all the signals in a driving zone need to be available in the switch phase of the next subsequent zone. Figure 9.3 depicts the concept of 2-D clocking scheme that permits a signal runs both *x*-axis and *y*-axis. If a zone change from hold phase to release phase when the adjacent zone in the same column ends its switch phase, then signals from that zone are not available in the switch phase of the adjacent zone in the next column. This prevents signal flow to the right which leads to an erroneous result of the QCA devices. Thus, all zones in a column must keep in the hold phase until the adjacent zones in the next column are in the switch state.

This clocking mechanism ensures minimum changes to the 1-D clocking scheme. The presented 2-D clocking mechanism in [4] is close to the existing 1-D clocking

Fig. 9.3 2-D clocking [4]

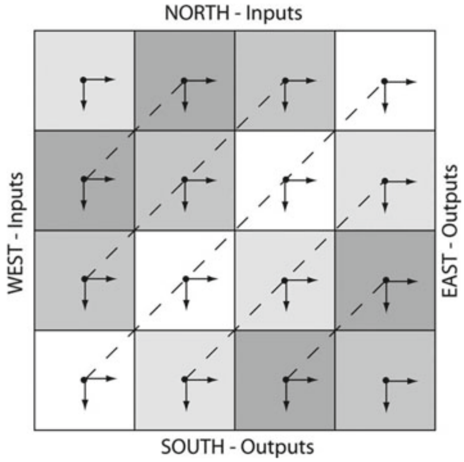


case as a zone is in release state as soon as the zones located along both axes are in switch state. The computation time for the 2-D mechanism is ($Z_x \times Z_y$), which shows quadratic relation with the number of zones along both the dimensions. It is worth noticing that the effective computation time is the sum of the clock periods of all columns in a QCA design; which is nearly the same for both the 1- and 2-D schemes. Even though the 2-D scheme finds a solution for long vertical lines, it does not provide performance improvements in terms of throughput over 1-D scheme. Moreover, the required clocking circuit for 2-D mechanism is quite challenging than the 1-D clocking mechanism.

One of the major concerns in the clocking techniques for QCA is the inability to create the feedback paths, which is a downside specially for sequential elements. In both 1-D and 2-D clocking mechanisms, information flow is inherently one directional. As reported in [4], the second scheme was based on a 2-D wave propagation of signals within a grid of zone. This scheme defines blocks of square zones with uniform size and orthogonal interconnections. Figure 9.4 depicts a schematic for the 2-D diagonal wave scheme (2DDWave), where the diagonally located zones are switched in parallel manner. This schematic enables low computation time while showing modest implementation complexity. To maintain analogy to the 2-D systolic array, individual zone receive information only from two adjacent zones (left and top) and allows its outputs to the other two zones (right and bottom). This permits equal number of zones in each column that results same outputs as the 1- and 2-D techniques. However, it is revealed that 2DDWave scheme does better with respect to computation speed than 1- and 2-D clocking techniques, which increases linearly along both the axes determined by (Z_x+ Z_y). To allow feedback paths in a 2DD clocking-based layout authors adopted the “trapezoidal clocking” mechanism [3]. Authors in [4] also presented a clocking circuitry.

Campos et al. [5] have proposed and implemented a new universal, scalable, and efficient (USE) clocking scheme, flexibility to create feedback paths of any length, and a well-defined clocking circuitry. It eliminates most of the problems in

Fig. 9.4 2DD wave clocking [4]



the existing clock schemes discussed above such as to ensure feedback paths and easy routing of QCA circuit layouts. This clocking scheme is based on the principle of neighborhood zones. Clock zones with adjacent numbers are always placed close to each other, while zones with non-adjacent numbers are distant. Figure 9.5 illustrates the structure of USE mechanism. Here, each block represents a single zone that contains cells, and the arrows show the direction of a logic value flow between QCA cells positioned in adjacent zones. USE allows the designer to implement long wires as well as feedback loops of any length. For large QCA circuits, the unit structure can be repeated as desired, as depicted in Fig. 9.5. Usually, all the zones in a row/column have a fixed direction of logic propagation, while corresponding neighboring zones in a row/column always retain opposite direction. This implies that there exist a number of possible routing paths, thus permits the most compact QCA structures.

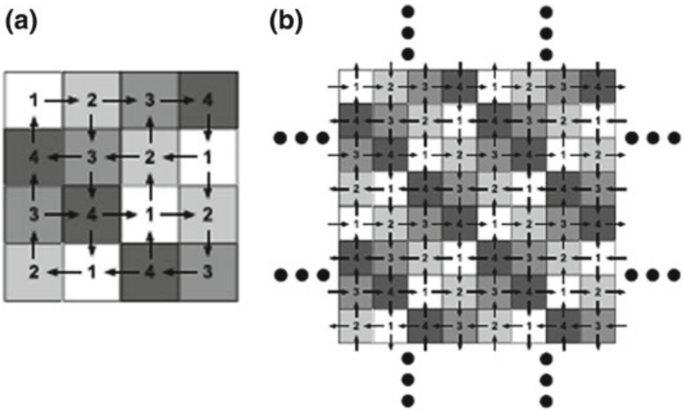


Fig. 9.5 USE clocking **a** unit structure, **b** extended version of USE [5]

Note that USE scheme facilitates both the coplanar and multilayer wire crossings QCA layout.

Authors also developed a clocking circuitry that generates required electric fields for each clock zones with low complexity. Note the current integrated circuit fabrication technologies are feasible in order to realize its clocking circuitry.

9.3 Comparative Analysis

Performance comparison of various clocking schemes is presented in Table 9.1. Different metrics specific to QCA clocking such as feedback path and design flexibility have been considered as evaluation purpose. Following observations are made:

- In 1-D scheme, the data runs horizontally and there exist no feedback paths through the QCA clock zones.
- There is no proper clocking circuitry is defined for the trapezoidal clocking mechanisms. In addition, this model falls short since the sizes of the clocking zones are not uniform.
- 2-D scheme finds a solution for long vertical lines, but no feedback path through different clock zones. The required clocking circuit for 2-D mechanism is quite challenging than the 1-D clocking mechanism.
- 2DD wave scheme allows feedback loop and low computation time. In addition, this scheme proves less flexible due to medium area overhead.
- Implementation using USE was better in terms of all metrics in comparison when compared with all previous clocking schemes.

Table 9.1 Comparison of different clocking schemes

Attributes	1-D [1]	Trapezoidal clocking [3]	2-D [4]	2DDWave [4]	USE [5]
Feedback path	No	Yes	No	Yes	Yes
Clocking circuitry	Modest	–	Complex	Modest	Simple
Zone size	Non-uniform	Non-uniform	Non-uniform	Uniform	Uniform
Design flexibility	Low	Low	Low	Medium	High
Area overhead	High	Medium	High	medium	Low
Computation time	High	Medium	High	Low	Low

9.4 Summary

In this chapter, we have discussed briefly different clocking schemes. This study can help to develop new routing algorithms, and design of an arbitrary large feasible QCA circuit toward the advancement of QCA technology. Comparisons of different schemes are also given. Different metrics specific to QCA clocking such as feedback path, design flexibility have been considered as evaluation purpose. The next chapter summarizes the work presented in this book and explores directions for further research.

References

1. C.S. Lent, P.D. Tougaw, A device architecture for computing with quantum dots. Proc. IEEE **85**(4), 541–557 (1997)
2. M. Janez, P. Pecar, M. Mraz, Layout design of manufacturable quantum-dot cellular automata. Microelectron. J. **43**(7), 501–513 (2012)
3. M.T. Niemier, P.M. Kogge, Problems in designing with QCAs: Layout = timing. Int. J. Circuit Theory Appl. **29**(1), 49–62 (2001)
4. V. Vankamamidi, M. Ottavi, F. Lombardi, Two-dimensional schemes for clocking/timing of QCA circuits. IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **27**(1), 34–44 (2008)
5. C.A.T. Campos, A.L. Marciano, O.P. Vilela Neto, F.S. Torres, USE: a universal, scalable, and efficient clocking scheme for QCA. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **35**(3), 513–517 (2016)

Chapter 10
Conclusion and Possible Future Direction

10.1 Summary of This Research

The QCA paradigm encodes bit information by charge configuration within a cell instead of current switches of transistors in conventional CMOS circuits. This revolutionary approach provides an alternate way for transistor-less computation at the nanoscale. This research has provided designs and simulation results for new nanoelectronics computing architecture-based digital design on the Quantum-dot Cellular Automata (QCA) paradigm. In this work, several new designs have been developed and analyzed for reversible and non-reversible logic circuits using QCA technology, which will help in improving the logic computation and information flow, provide features for easy physical implementation.

First, we explored the QCA implementation of primitive reversible logic gates. Second, we have presented a compact 5-input majority gate using single layer QCA technology. We have put forth this gate to design few efficient QCA circuits, which are more efficient compared to the existing designs. In addition, we have examined a new design for the implementation of 3-input XOR that uses explicit interactions between QCA cells to produce the expected results. In order to show the efficacy of this XOR gate, adder and divider are constructed based on it. Further, we discussed the architecture of an efficient 1-bit reversible ALU using an existing reversible gate in QCA, which utilizes a minimum number of QCA cells and clock delay. The proposed designs are evaluated in terms of metrics such as the QCA cells, delay, and garbage outputs. New QCA structures for D flip-flops, shift register, and memory cell are proposed, simulated, and evaluated. The basic component of sequential designs is an efficient set of multiplexer circuits. All the QCA layouts are validated and simulated using the QCADesigner. Finally, we have discussed briefly different clocking schemes. This study can help to develop new routing algorithms, and design of an arbitrary large feasible QCA circuit toward the advancement of QCA technology. We have discussed the observations from the results at the end of each chapter.

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2_10

10.2 Future Direction

Some of the extensions of this work are listed below:

- In addition to dot displacement, investigations need to be performed on cell displacement, missing dots, and missing cells.
- Clocking would need to be explored as well, for both the four and five dot QCA models. But even with no defect, the current model is only functional at very low temperatures.
- Additional work needs to be done to improve the strength of the crossover. One possible solution is to move away from the traditional planar scheme and utilize vertical inputs. Moving QCA from a planar system to a multilevel scheme would allow for much more intricate circuitry with fewer cells.
- To improve the temperature sensitivity, the model would need to be reduced to the molecular scale. As confinement is increased to the molecular scale, the energy states raise well above room temperature.
- Investigation of design of fault-tolerant nanocircuits based on reversible logic and conservative logic.
- Development of a comprehensive synthesis tool for reversible arithmetic circuits in the QCA framework.

Appendix A
Tutorial on QCADesigner 2.0.3

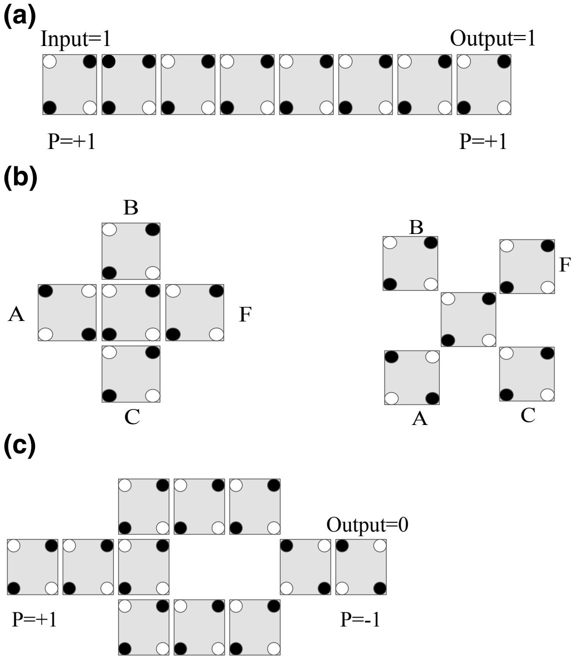
This tutorial provides quick and handy information that will help designer understand how to create a simple QCA-based design, perform simulation and store the outputs.

A.1 Layout of QCA Basic Elements

This section explains the layouts of basic components such as wire, 3-input majority gate, and inverter (see Fig. A.1). Figure A.1a shows cascade of QCA cells which represents a QCA wire. In a coplanar design only one layer allowed and the default layer is “main cell layer” (see the box above design area). The single cell can be created by clicking “Cell” followed by left click on the design window. Note a left click in the design area only append one cell. So we can get desired number of cells by left clicking as many times as we need. An array of cells can be obtained by clicking “Array” and dragging the mouse in the design area to a require length. To delete one or more cells click “Select” followed by selecting unwanted cell/cells and pressing delete button. Figure A.1b shows layout of a 3-input majority gate as described by $M(A, B, C) = \text{MV3} = AB + BC + AC$. The input/output mode of a cell can be identified by double clicking corresponding cell which pop up “Cell Function” window. All the cells in a QCA circuit are clocked appropriately using “Clock Box” above design area and choosing correct clock zone from the drop list. The layout design for an inverter is obtained in similar a manner. QCA structure of an inverter is illustrated in Fig. A.1c.

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2

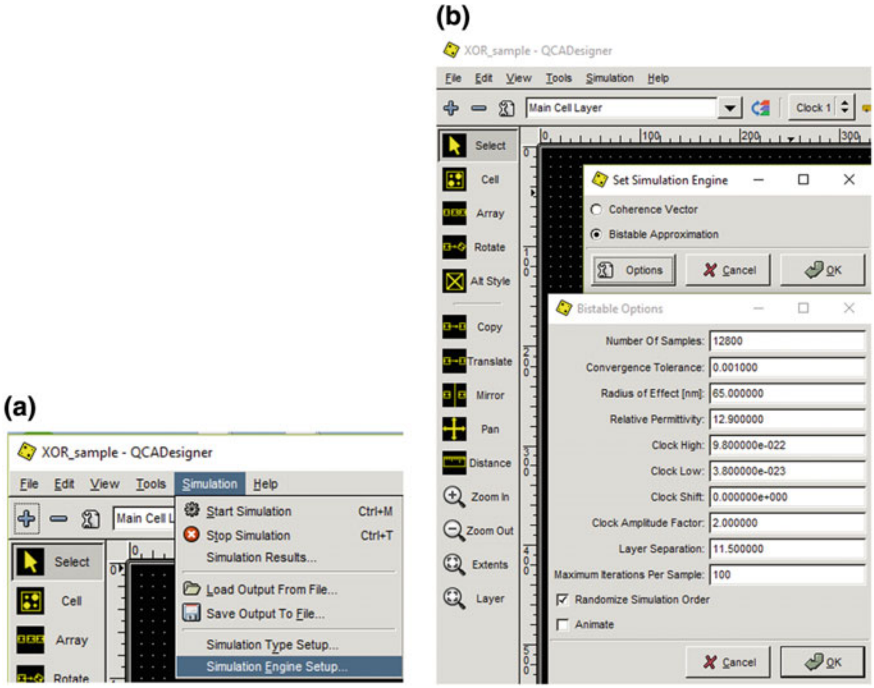
Fig. A.1 QCA primitives **a** wire, **b** majority gate (MV3), and **c** inverter



A.2 *Way to Simulate*

QCADesigner 2.0.3 enables user to specify type of simulation engines: Coherence vector and bistable approximation are depicted in Fig.A.2. In addition to this, it allows user to specify the way in which the input is to be provided. To select simulation engine, click **Simulation** and select **Simulation Engine Setup**, as shown in Fig.A.2a. The default engine appears for simulation is "bistable approximation" (see Fig. A.2b). In bistable approximation engine, designer can change number of samples, maximum iteration per sample, etc. Figure A.3 shows, the coherence vector engine where designer gets the opportunity to set temperature, relaxation time, total simulation time, etc. In general, bistable approximation engine is preferably used for simulations of basic circuits, whereas coherence vector type is appropriate for studying thermal robustness of designs.

To select different formats of input, click **Simulation** and select **Simulation Type Setup**, as shown in Fig. A.4a. The vector table simulation type allows designer to define the input vectors (see Fig. A.4b). User can easily insert a new vector by clicking on the "+" button. In exhaustive simulation type, considers all possible combinations of the input vectors to simulate the circuit. After the desired input type is specified, click "Start Simulation" to initiate the simulation.



Set Simulation Engine

○ Coherence Vector

● Bistable Approximation

Bistable Options

Number Of Samples:	12800
Convergence Tolerance:	0.001000
Radius of Effect [nm]:	65.000000
Relative Permittivity:	12.900000
Clock High:	9.800000e-022
Clock Low:	3.800000e-023
Clock Shift:	0.000000e+000
Clock Amplitude Factor:	2.000000
Layer Separation:	11.500000
Maximum Iterations Per Sample:	100

☑ Randomize Simulation Order

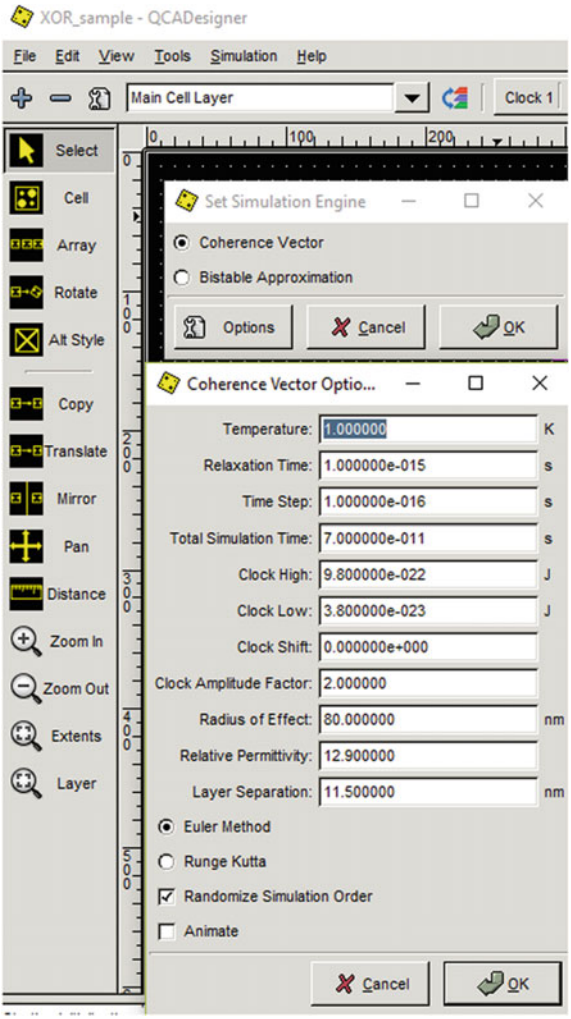
□ Animate

Fig. A.2 Type of simulation engines

A.3 *Clocking a Device*

In QCA, all the cells are clocked to controls the data flow and to supply power for the cells. Different parts of the QCA layout are assigned with appropriate clock zones for correct functionality of the circuit. For illustration purpose, a 2-input XOR QCA layout is considered given by OUT $= I_0 \oplus I_1$, where I_0, I_1 are inputs (see Fig. A.5). Both the inputs are assigned to clock zone 0. In addition, clock zone 0 also drives two inverters that yield $I_0'$and I_1'. Thereafter, clock zone 1 is assigned to two majority gates that realize two main terms $I_0 \cdot I_1'$ and $I_1 \cdot I_0'$. Final output needs one majority gate that implement OR gate is assigned to clock zone 2. Note that one of the inputs of AND gate and OR gate is indicated by polarization -1 and $+1$, respectively.

Fig. A.3 Coherence vector engine



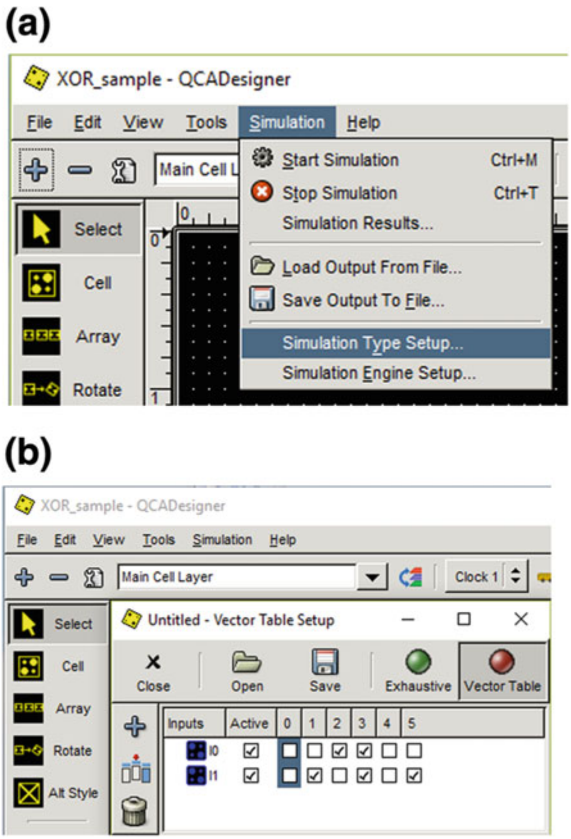
● Coherence Vector
○ Bistable Approximation

Temperature: 1.000000 K
Relaxation Time: 1.000000e-015 s
Time Step: 1.000000e-016 s
Total Simulation Time: 7.000000e-011 s
Clock High: 9.800000e-022 J
Clock Low: 3.800000e-023 J
Clock Shift: 0.000000e+000
Clock Amplitude Factor: 2.000000
Radius of Effect: 80.000000 nm
Relative Permittivity: 12.900000
Layer Separation: 11.500000 nm

● Euler Method
○ Runge Kutta
☑ Randomize Simulation Order
□ Animate

A.4 *QCA Layout Evaluation Parameters*

Once the layout of the circuit is completed, one can extract parameters like cell occupied area, number of cells, and delay for the corresponding design. Figure A.6 shows selection of all the cells of the design by making a box. Thereafter, the total cell count and cell area display at the bottom of the window. Total delay of the circuit is calculated from the number of clock zones, for instance, the delay for the given circuit in Fig. A.5 is 0.75 clocks or 3 clock phases. Note designer can increment a clock zone of an array of cells or whole layout via “Tools → Increment cell clocks.”



Inputs	Active	0	1	2	3	4	5
I0	☑	☐	☐	☑	☑	☐	☐
I1	☑	☐	☑	☐	☑	☐	☑

Fig. A.4 Coherence vector **a** simulation type setup and **b** input vectors setup

A.5 Different Types of Crossover

As discussed earlier, there are two different types of crossover methods commonly used, coplanar and multilayer. In coplanar crossover strategy is shown in Fig. A.7a, wire crossing is done by two different cells. The default wire consists of cells of 90° orientations while for 45° orientation cells select the required cell/cells followed by clicking “Rotate” button. Another coplanar crossover strategy is shown in Fig. A.7b, which takes advantage of two zones of the four-phase zone-based clocking scheme. Therefore, signals *A* and *B* pass through with coplanar crossover if the horizontal cells are assigned clock 0 (clock 1) and the vertical cells are assigned clock 2 (clock 3) or vice versa.

The multilayer design steps are illustrated in Figs. A.8, A.9, A.10, A.11, A.12, A.13, A.14, A.15 and A.16. A multilayer crossover requires three layers: Main cell Layer, middle Layer, and Upper Layer. QCADesigner provides options for alternate drawing style which can be used to link different layers in a multilayer design. This is accomplished by clicking on “Alt Style” and selecting one from the drop list, as shown in Fig.A.8.

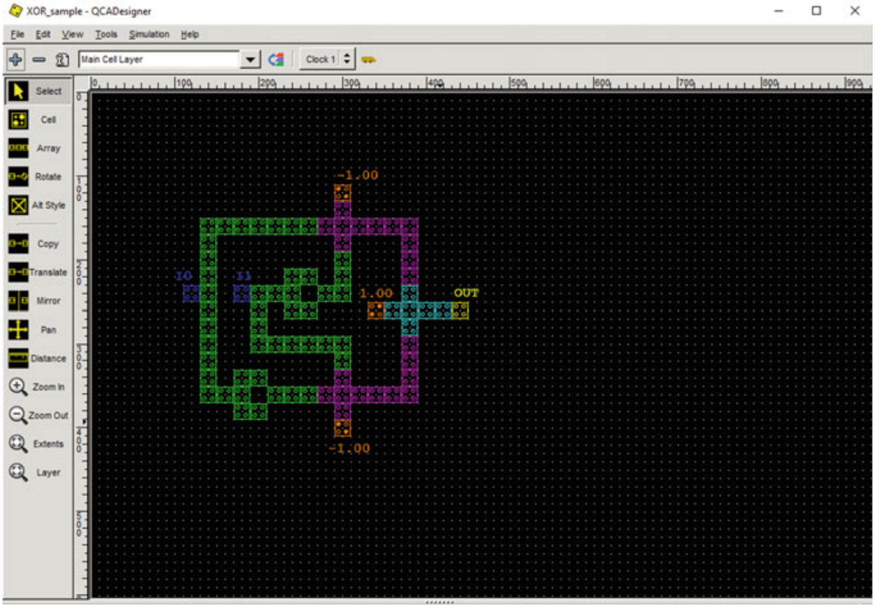


Fig. A.5 Sample XOR_gate

A.6 Storing Layout and Simulation Result

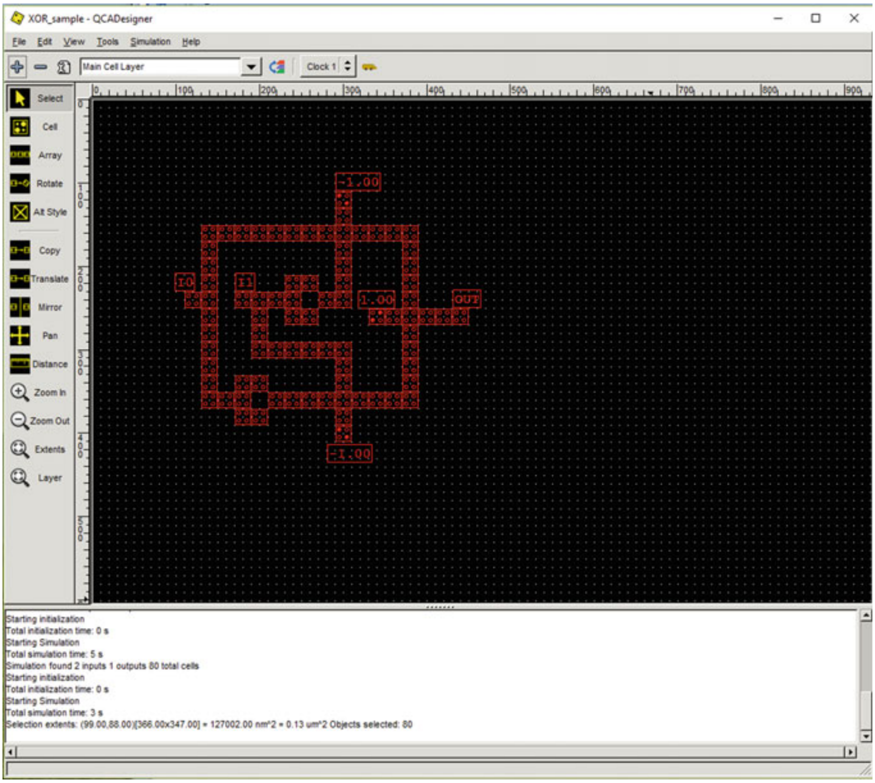
To save the layouts select File → Print → Printer. The window of Fig.A.17 will be opened. In addition, print option enables setting like margins, scale, and printed objects. For coplanar layouts, deselect substrate and drawing layer in **Printed Objects**. Whereas, multilayer layouts keep all the three layers active. In the file name field of Fig.A.18, enter the desired circuit layout name. In this case, the file name used was **XOR_sample.eps** or **XOR_sample.qca** (Note: .qca file for QCA layout and .eps file for print the layout as an image file).

Further, the simulation waveform can be saved by selecting "Simulation → Start simulation → Save". While the waveform in Fig. A.19 can be saved as image file (.eps) by selecting "Simulation → Start simulation → print."

A.7 Summary

This tutorial briefly describes QCADesigner 2.0.3, a tool for QCA-based circuits.

The steps provided in this section describe different crossover, how to extract design metrics, how to set up the simulation engine, and how to save the results. Please refer QCADesigner (http://www.mina.ubc.ca/qcadesigner) for additional documentation.



```
Starting initialization
Total initialization time: 0 s
Starting Simulation
Total simulation time: 5 s
Simulation found 2 inputs 1 outputs 80 total cells
Starting initialization
Total initialization time: 0 s
Starting Simulation
Total simulation time: 3 s
Selection extents: (99.00,88.00)[366.00x347.00] = 127002.00 nm^2 = 0.13 um^2 Objects selected: 80
```

Fig. A.6 Sample layout

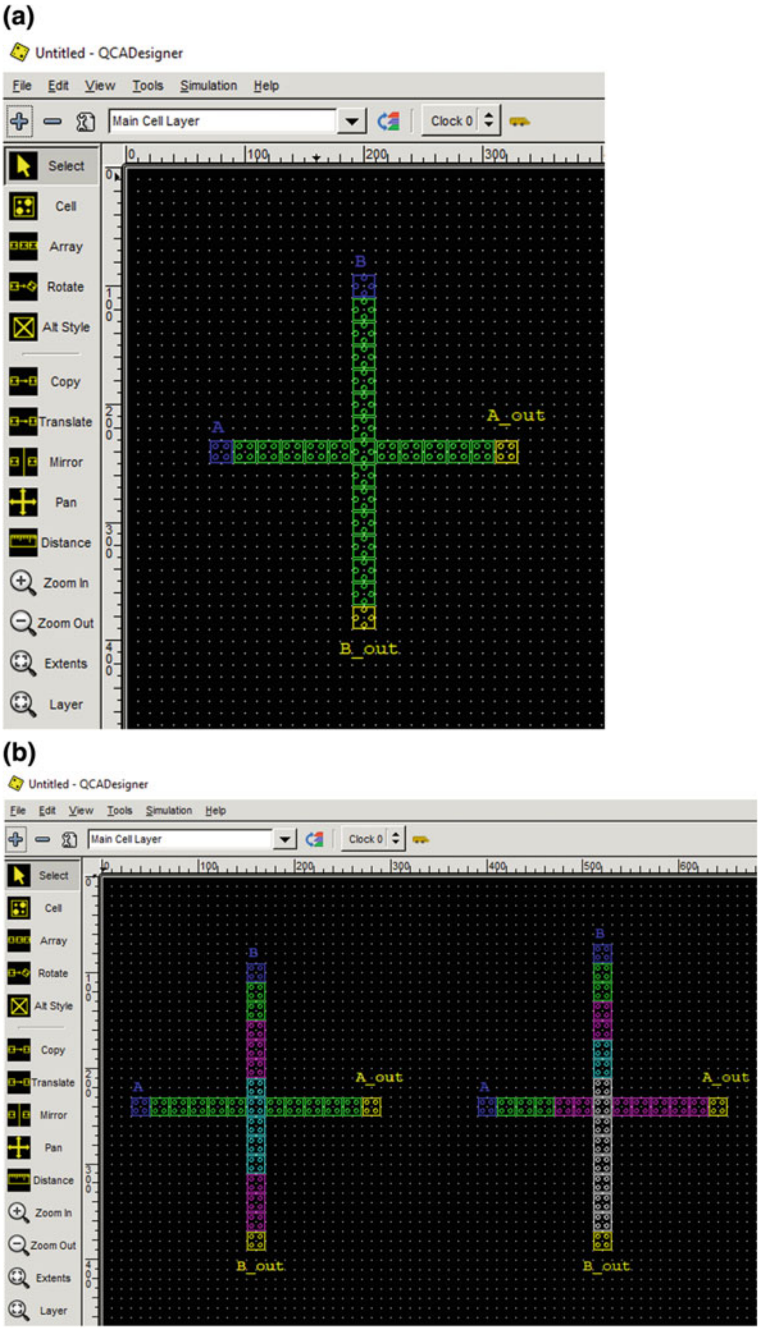


Fig. A.7 Coplanar crossover **a** with two types of cells and **b** with different phases

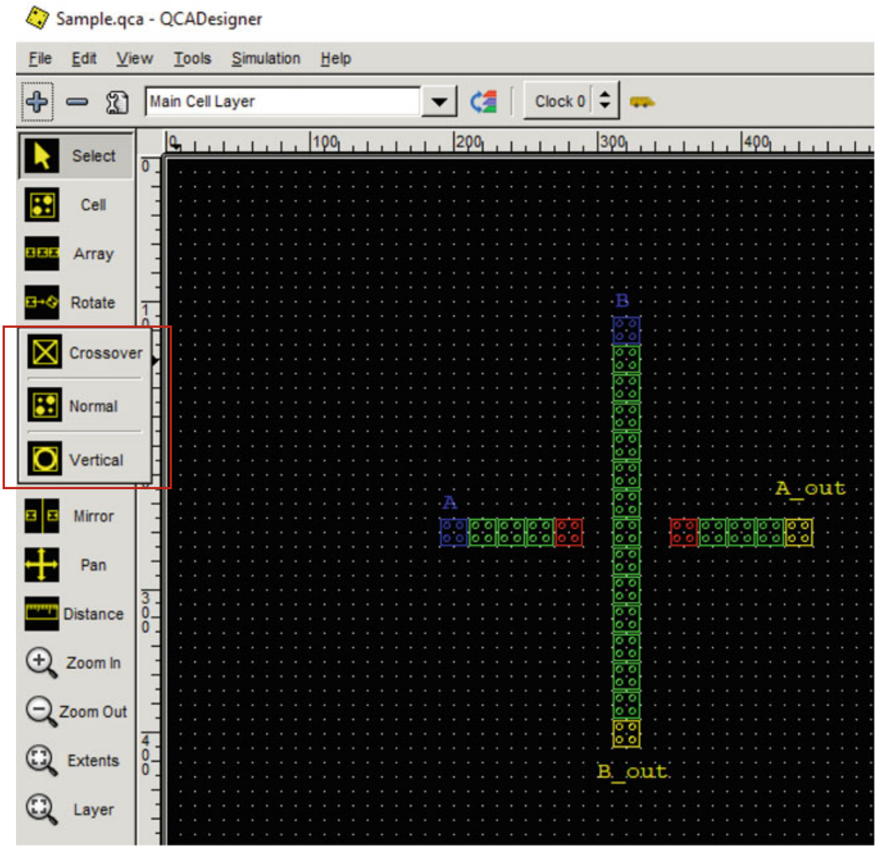
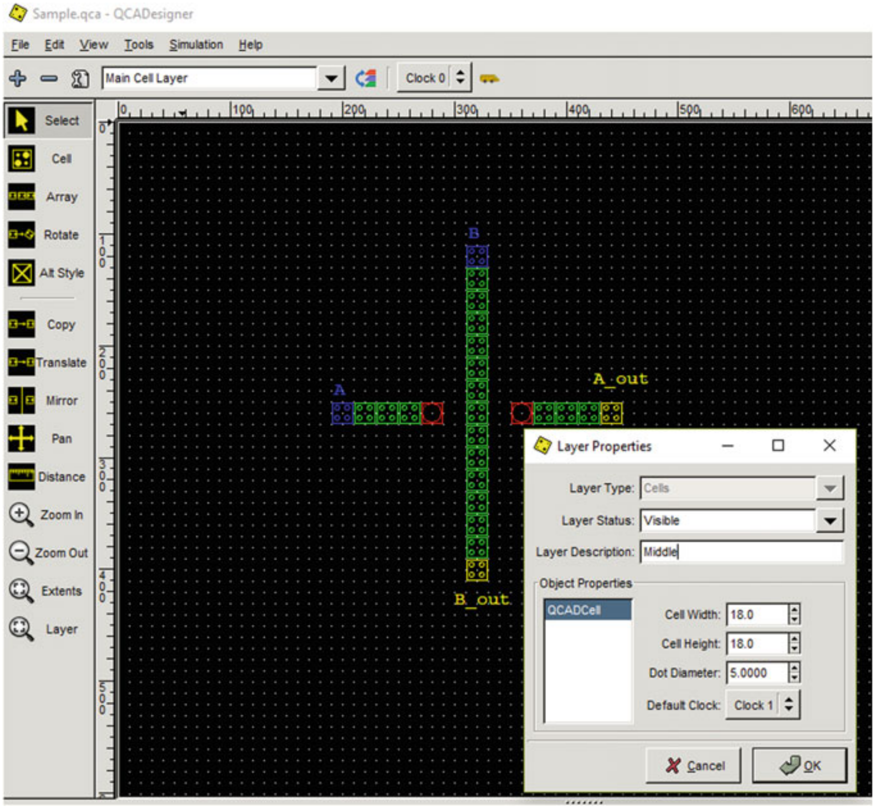


Fig. A.8 Multilayer crossing-clip 1



Layer Type: Cells

Layer Status: Visible

Layer Description: Middle

Cell Width: 18.0

Cell Height: 18.0

Dot Diameter: 5.0000

Default Clock: Clock 1

Fig. A.9 Multilayer crossing-clip 2

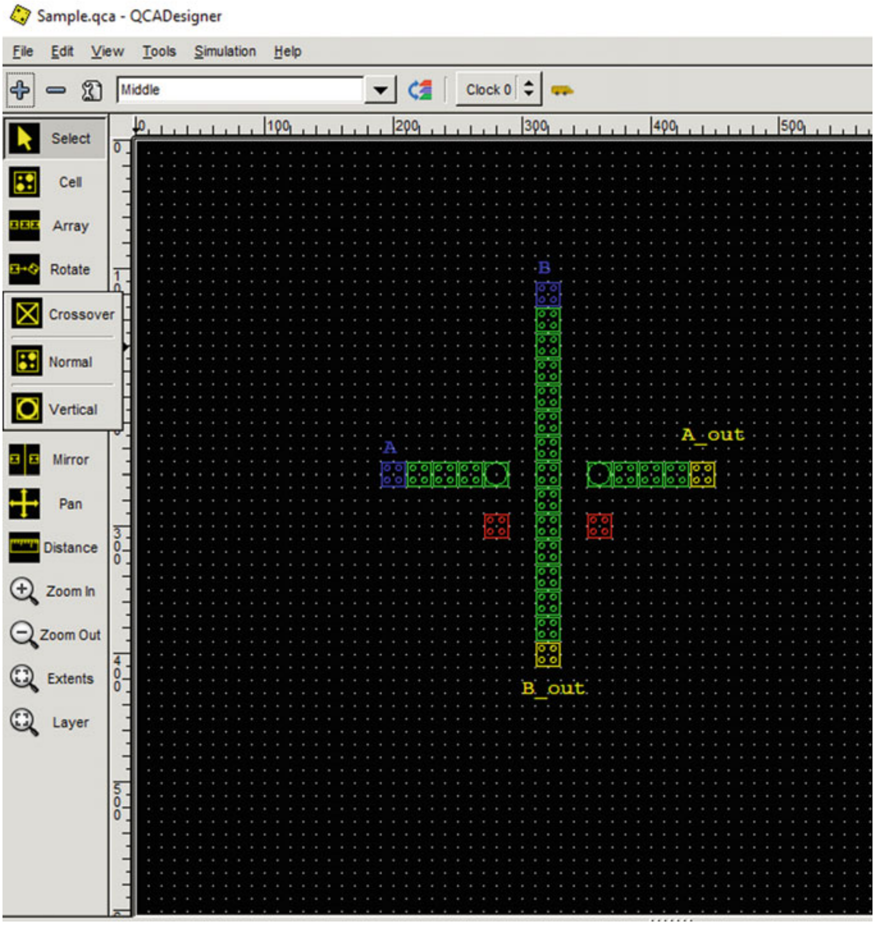


Fig. A.10 Multilayer crossing-clip 3

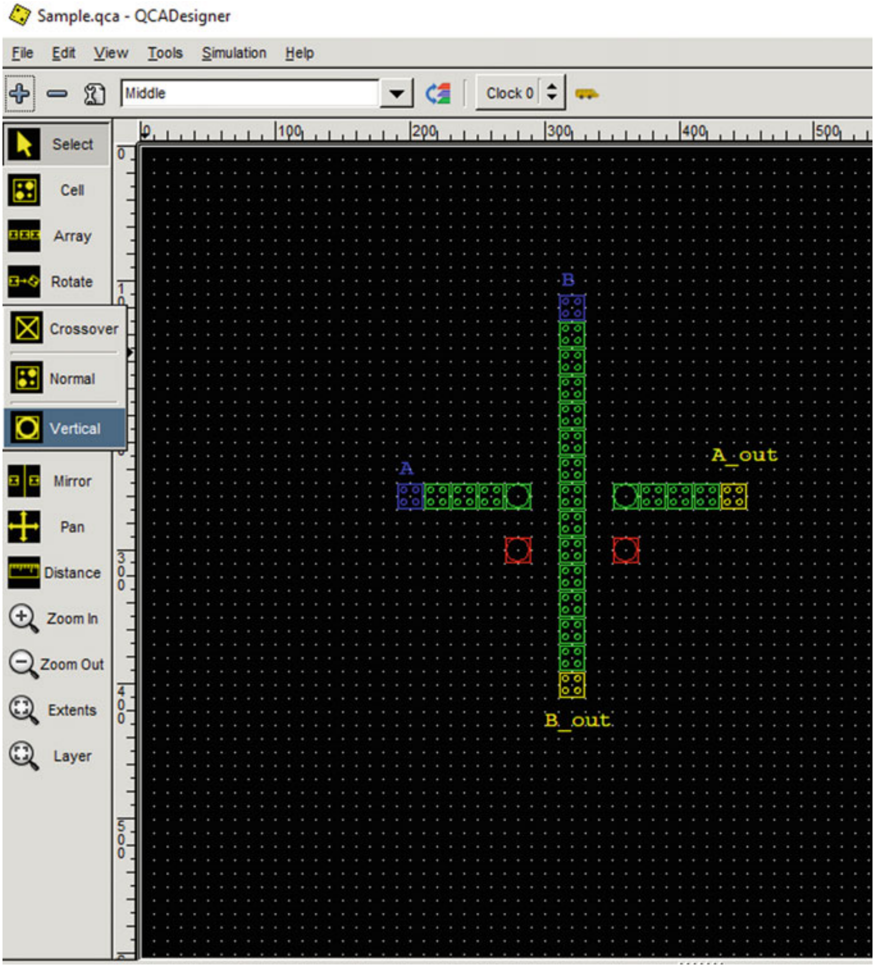
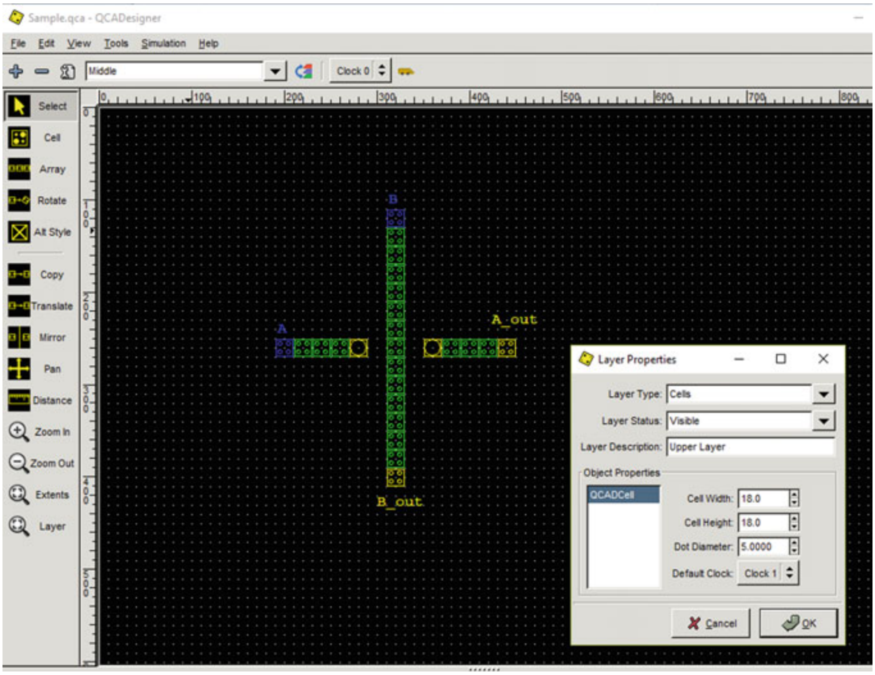


Fig. A.11 Multilayer crossing-clip 4



Layer Type: Cells

Layer Status: Visible

Layer Description: Upper Layer

Cell Width: 18.0

Cell Height: 18.0

Dot Diameter: 5.0000

Default Clock: Clock 1

Fig. A.12 Multilayer crossing-clip 5

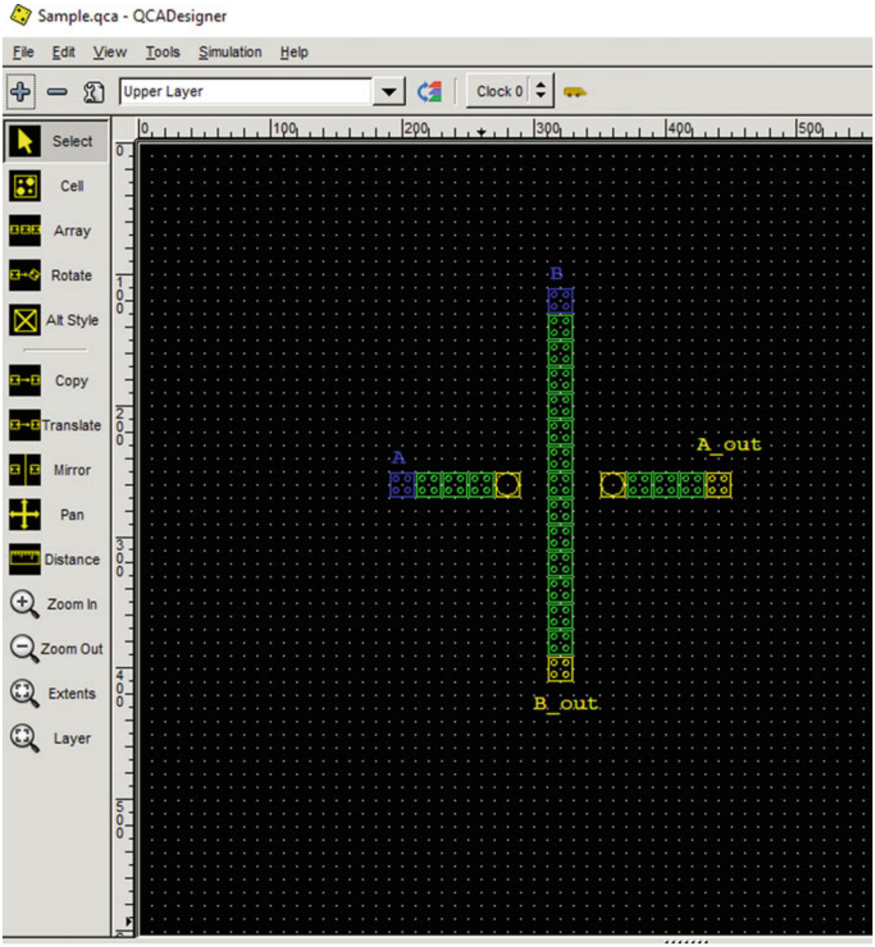


Fig. A.13 Multilayer crossing-clip 6

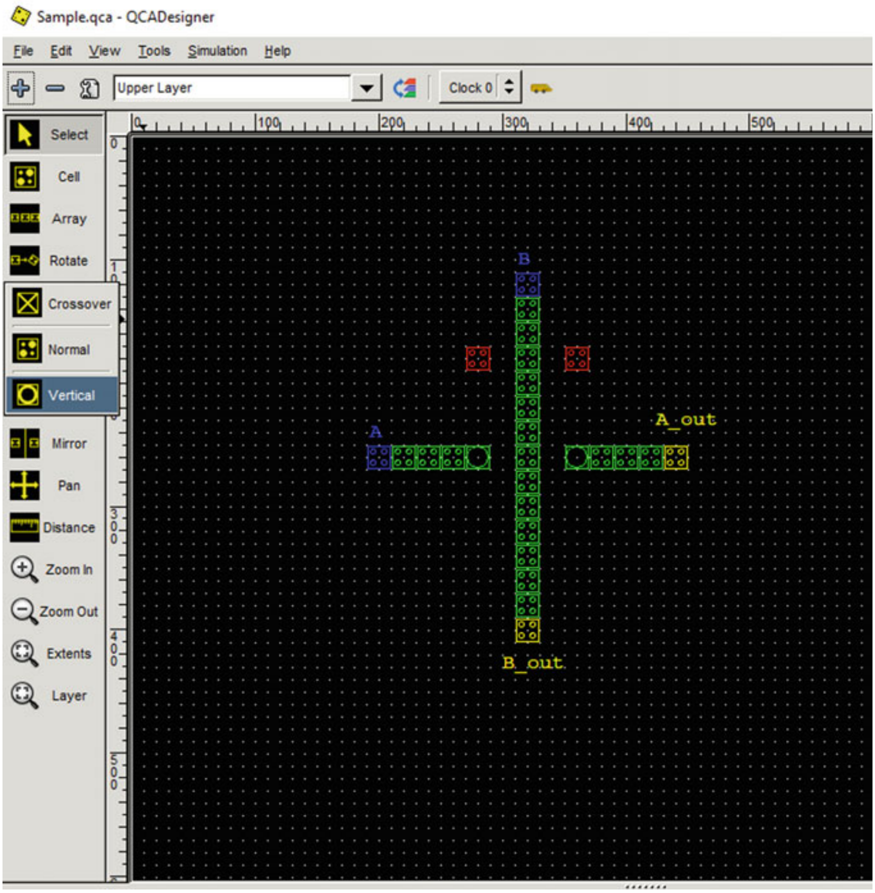


Fig. A.14 Multilayer crossing-clip 7

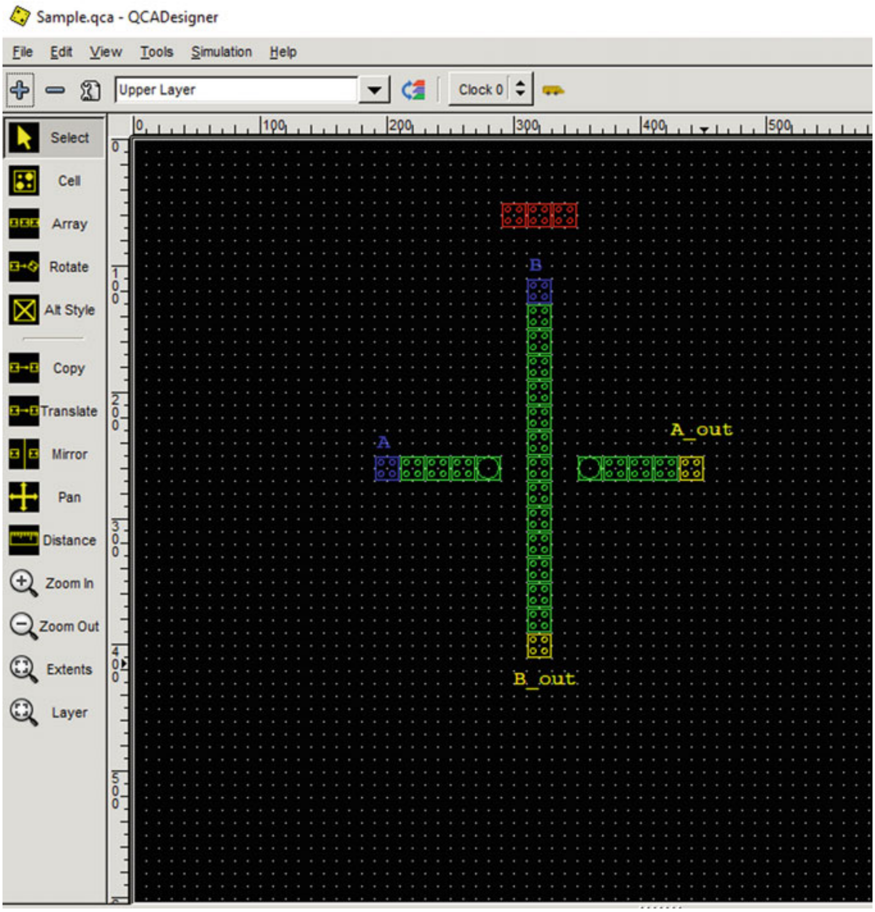


Fig. A.15 Multilayer crossing-clip 8

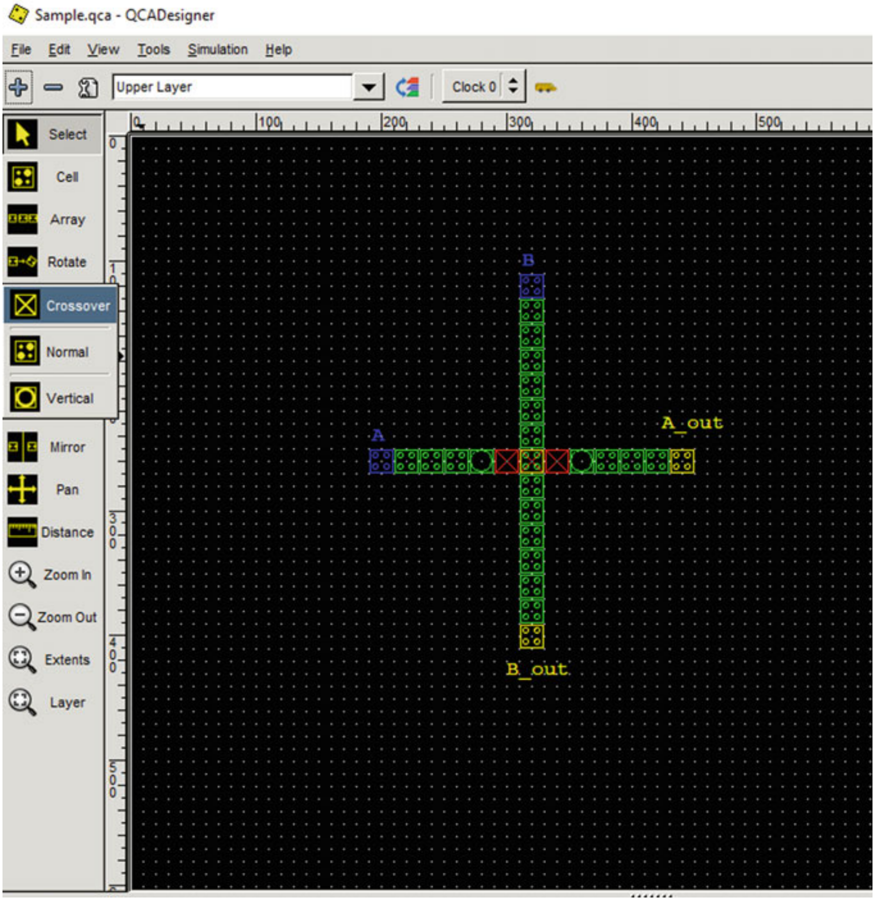
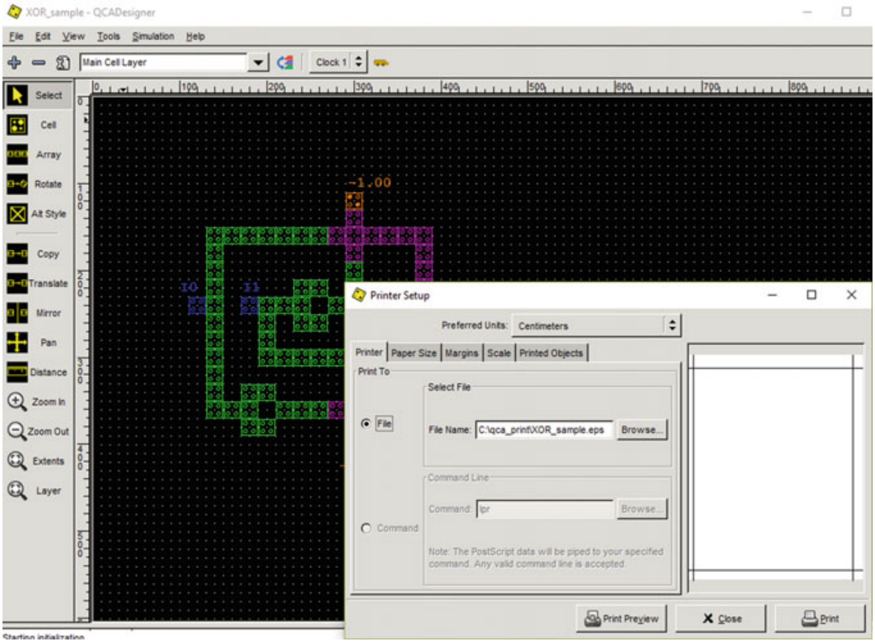


Fig. A.16 Multilayer crossing-clip 9



Preferred Units: Centimeters

● File

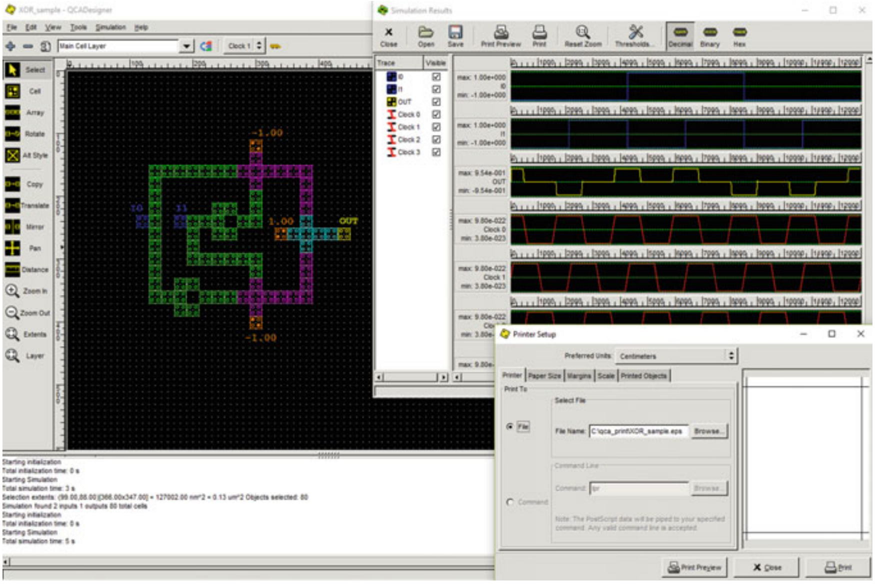
File Name: C:\qca_print\XOR_sample.eps

○ Command

Command: lpr

Note: The PostScript data will be piped to your specified command. Any valid command line is accepted.

Fig. A.17 Storing result



Preferred Units: Centimeters

● File

File Name: C:\qca_print\XOR_sample.eps

○ Command

Command: lpr

Note: The PostScript data will be piped to your specified command. Any valid command line is accepted.

```
Starting initialization
Total initialization time: 0 s
Starting Simulation
Total simulation time: 3 s
Selection extents: (99.00,86.00)[368.00x347.00] = 127002.00 nm^2 = 0.13 um^2 Objects selected: 80
Simulation found 2 inputs 1 outputs 80 total cells
Starting initialization
Total initialization time: 0 s
Starting Simulation
Total simulation time: 5 s
```

Fig. A.18 File rename

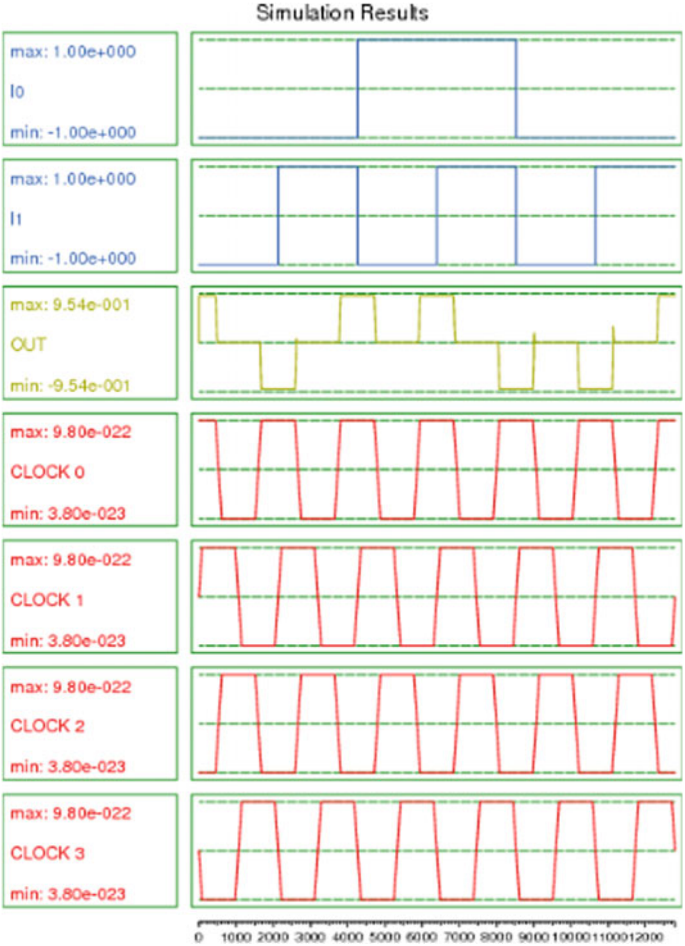


Fig. A.19 Simulation waveform

Appendix B
Tutorial on QCAPro

This appendix serves as a quick reference to (i) how to handle QCAPro, (ii) estimate of power loss in a QCA circuit, and (iii) measure of output polarization. Additional information about QCAPro can be fetched from the webpage of the authors of QCAPro.

B.1 How to run the tool?

Currently the QCAPro ver 1.0 runs in Linux (32/64 bit). The tool reads a **.qca** file generated by QCADesigner ver 1.40/1.41 as input. It is not recommended to use the latest versions of QCADesigner as QCAPro ver 1.0 does not support multilayer QCA layout.

Input files for the tool:

This tool reads three files as follows:

I. A QCA layout file of the design generated from QCADesigner ver 1.40/1.41 (.qca)
II. An input/output vector file (to check the design accuracy and isolate erroneous cells).
III. A switching vector file (to estimate average switching power dissipation in the QCA circuit over all switching vectors).

Outputs generated from the tool:

I. Check the QCA design for all input /output vector combinations listed in the input/output vector file.
II. Display a layout of the design with approximate error probabilities of each intermediate cell and the output cells.

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2

III. A layout of the design with approximate average switching power dissipation of each intermediate cell and the output cells for the input vector sets given in switching vector.

IV. Data file with maximum/minimum power dissipation and the corresponding input switching vectors.

V. A **.net** file generated that gives the Bayesian network of the QCA layout.

B.2 Measuring Power and Polarization of a Circuit

A 3-input XOR gate is considered (part of 1-bit full adder from Sect. 5.5) for illustration purpose as depicted in Fig. B.1. It takes three data inputs A, B, and C to generate output XOR_3 = $A \oplus B \oplus C$. To measure power dissipation of XOR layout, we need three files: QCA layout in Fig. B.1, input/output vector set, and switching vector set (e.g., .txt files) as shown in Figs. B.2a, b, respectively.

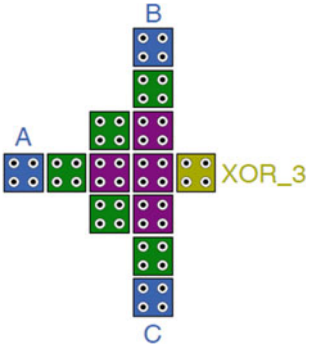
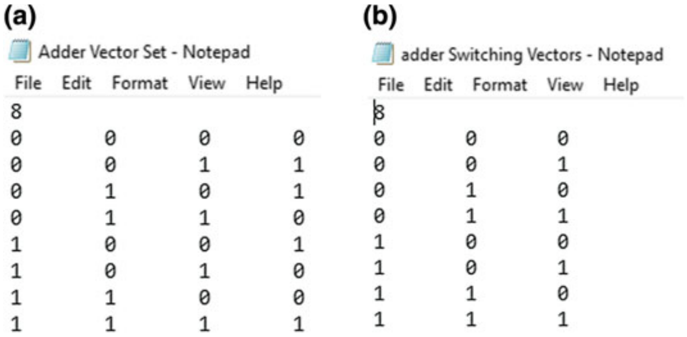


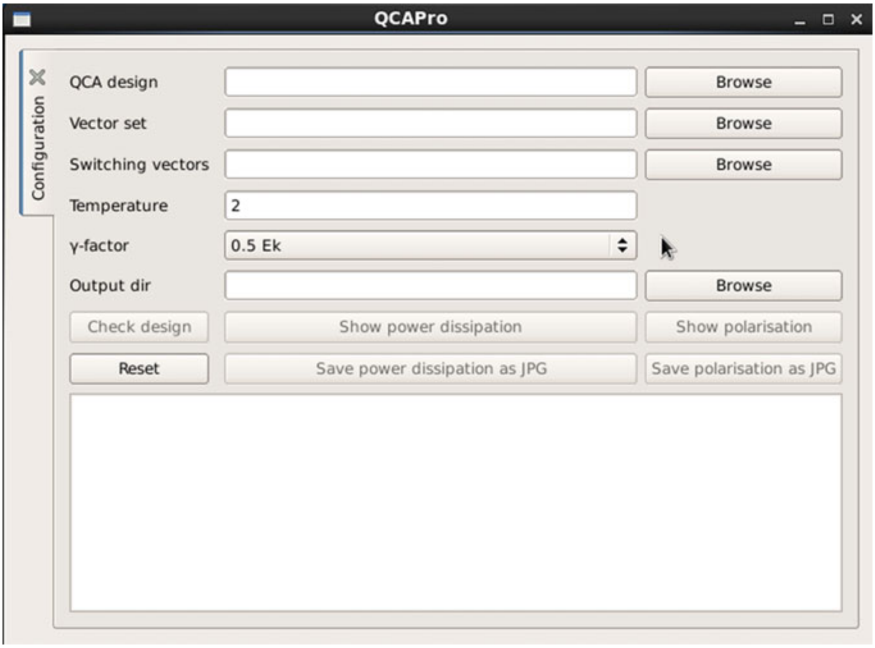
Fig. B.1 QCA layout for XOR gate based on explicit interaction of cells


(a)
Adder Vector Set - Notepad
File Edit Format View Help
(b)
adder Switching Vectors - Notepad
File Edit Format View Help

```
8
0    0    0    0
0    0    1    1
0    1    0    1
0    1    1    0
1    0    0    1
1    0    1    0
1    1    0    0
1    1    1    1
```

```
8
0    0    0
0    0    1
0    1    0
0    1    1
1    0    0
1    0    1
1    1    0
1    1    1
```

Fig. B.2 Input files to QCAPro **a** input/output vector set and **b** switching vector set



Temperature 2

γ-factor 0.5 Ek

Fig. B.3 QCAPro main window

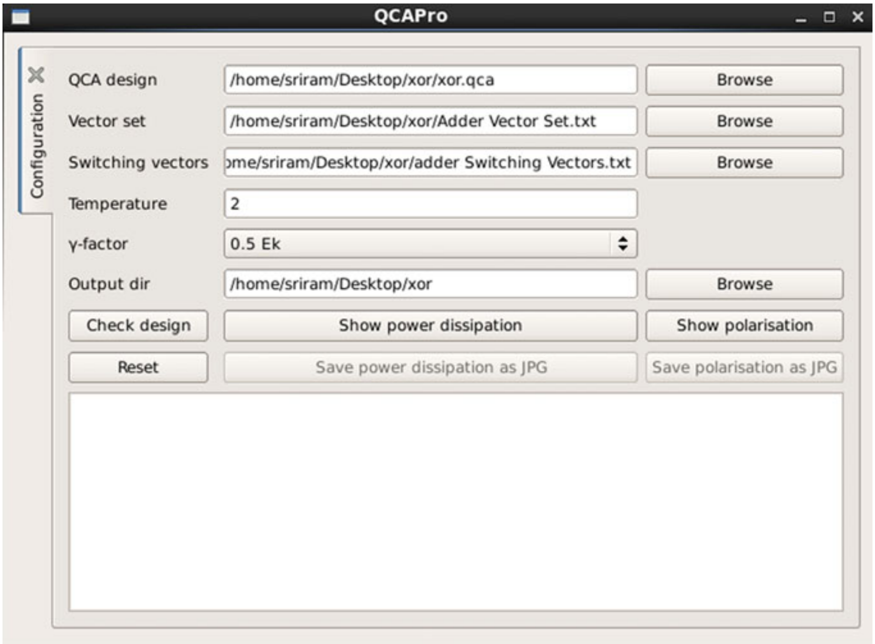
Figures B.3 and B.4 show snapshots of QCAPro while performing design check of QCA XOR circuit. QCAPro main window allows designer to select QCA design, vector set, and switching vector file from appropriate folder. Different temperature can be selected, where default temperature is 2 K. For evaluation, three different tunneling energies can be taken (0.5 E_k, 1.0 E_k, and 1.5 E_k).

Click **Check design** as shown in Fig. B.5 to perform a quick design check to test the value of outputs for all possible combinations of inputs. If there is an error detected, QCAPro generates an error message and also the input vector set for which the expected output did not match the output obtained. Thereafter, designer can generate power dissipation map or a map of polarization error for the design.

Click **Show power dissipation** to generate a power dissipation map that displays the thermal hotspots. Figure B.6 shows energy dissipation map of the XOR design with tunneling energy of 0.5 E_k. QCAPro also enables to store result as an image file (.JPG).

Click **Show Polarization** to estimate polarization map of QCA circuit for any particular input vector. Figure B.6 shows a map of polarization error for the QCA design.

QCAPro generates a text file for thermal layout with average, maximum, and minimum power dissipation in a QCA circuit, as shown in Fig. B.7.



QCA design /home/sriram/Desktop/xor/xor.qca

Vector set /home/sriram/Desktop/xor/Adder Vector Set.txt

Switching vectors ome/sriram/Desktop/xor/adder Switching Vectors.txt

Temperature 2

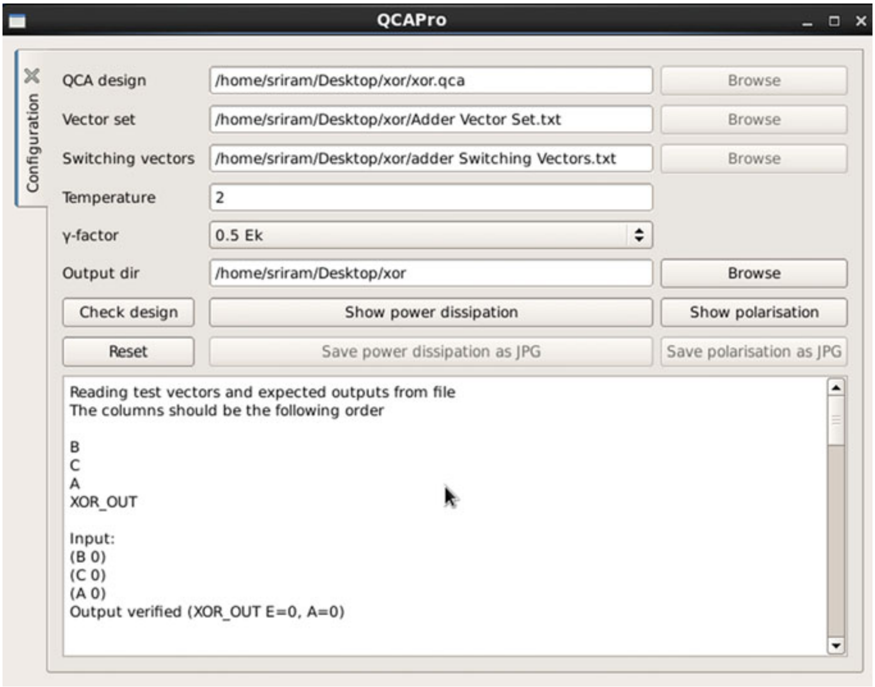
γ-factor 0.5 Ek

Output dir /home/sriram/Desktop/xor

Fig. B.4 QCAPro File selection

B.3 Summary

In this appendix, we have briefly described QCAPro to estimate error and power dissipation in QCA circuit design.



QCA design: /home/sriram/Desktop/xor/xor.qca

Vector set: /home/sriram/Desktop/xor/Adder Vector Set.txt

Switching vectors: /home/sriram/Desktop/xor/adder Switching Vectors.txt

Temperature: 2

γ-factor: 0.5 Ek

Output dir: /home/sriram/Desktop/xor

```
Reading test vectors and expected outputs from file
The columns should be the following order

B
C
A
XOR_OUT

Input:
(B 0)
(C 0)
(A 0)
Output verified (XOR_OUT E=0, A=0)
```

Fig. B.5 Check design

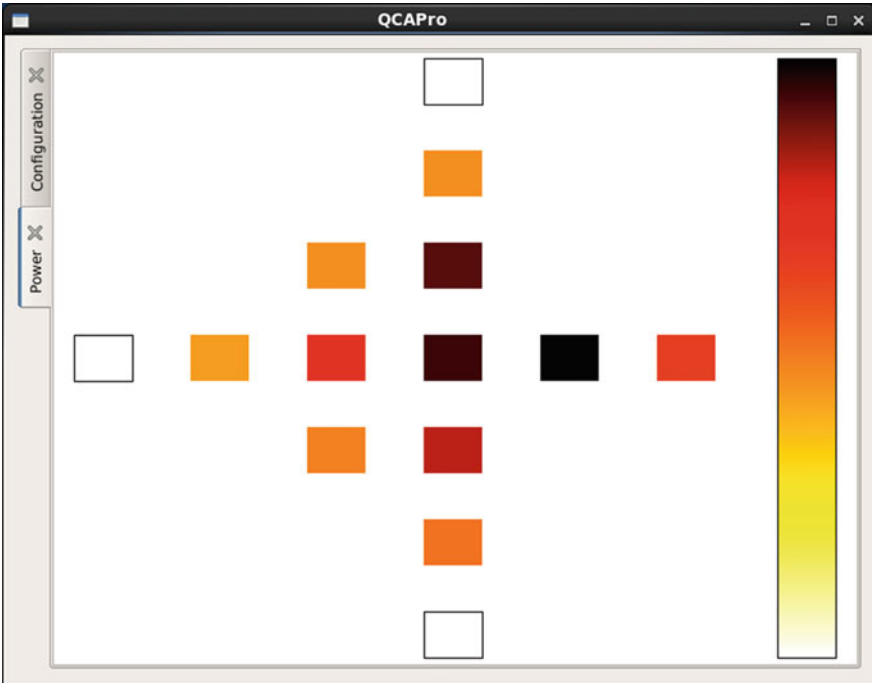


Fig. B.6 Energy dissipation map of the XOR

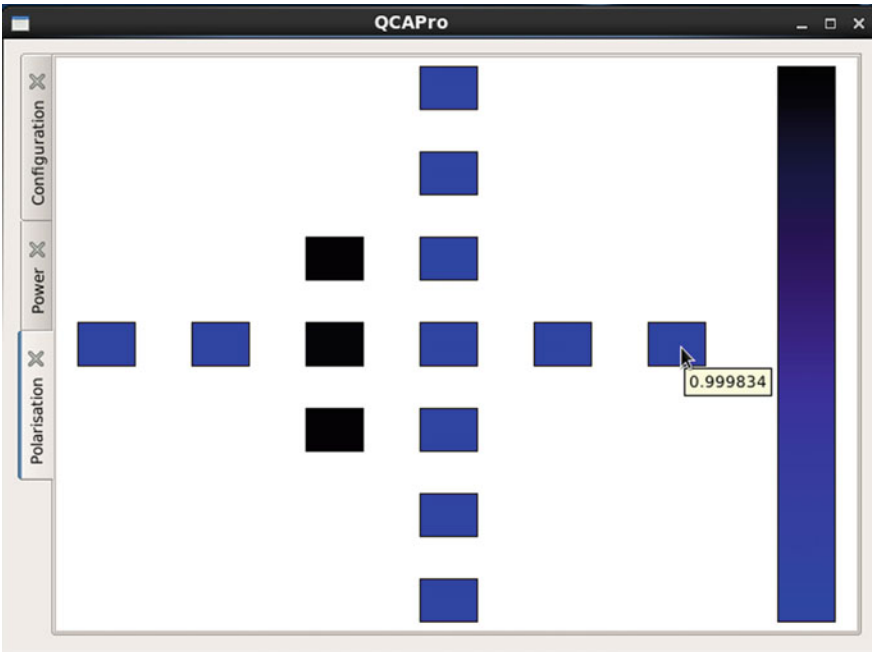
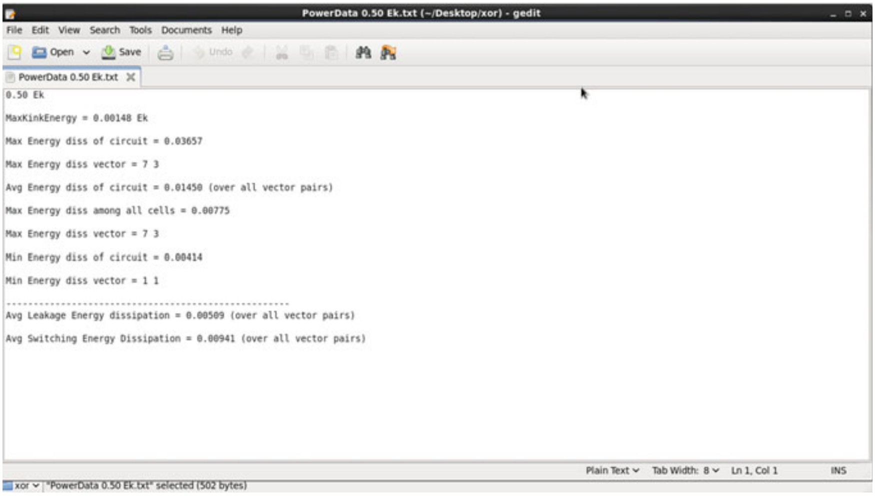


Fig. B.7 Polarization error map



```
0.50 Ek

MaxKinkEnergy = 0.00148 Ek

Max Energy diss of circuit = 0.03657

Max Energy diss vector = 7 3

Avg Energy diss of circuit = 0.01450 (over all vector pairs)

Max Energy diss among all cells = 0.00775

Max Energy diss vector = 7 3

Min Energy diss of circuit = 0.00414

Min Energy diss vector = 1 1

----------------------------------------------------
Avg Leakage Energy dissipation = 0.00509 (over all vector pairs)

Avg Switching Energy Dissipation = 0.00941 (over all vector pairs)
```

Fig. B.8 Energy dissipation report

Index

T. N. Sasamal et al., *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Studies in Computational Intelligence 879,
https://doi.org/10.1007/978-981-15-1823-2