Analog Integrated Circuits and Signal Processing (2024) 118:171–186
https://doi.org/10.1007/s10470-023-02224-1

A novel reversible gate and optimised implementation of half adder, subtractor and 2-bit multiplier

Siddhesh Soyane[1] · Ajay Kumar Kushwaha[1] · Dhiraj Manohar Dhane[1]

Received: 19 June 2022 / Revised: 29 November 2023 / Accepted: 30 November 2023 / Published online: 24 December 2023

Abstract
The paper proposes a novel 3×3 reversible gate which has varied functionality for logical and arithmetic operations. The advancements in VLSI demand higher operational speed and less time delay, which leads to increased complexity and more power dissipation in the design. The continuous evolution of DSP applications demands improvisation on the multiplier design that is faster and more power efficient. Reversible logic is an efficient solution to the above problems. In the paper, a basic 2×2 multiplier, the proposed novel gate, and its enhanced capability for implementing half adder-subtractor over existing basic reversible gates are discussed. The proposed designs were implemented on QCA Designer.

Keywords Reversible gates · Multiplier · Quantum cellular automata (QCA)

1 Introduction

In VLSI design, there is a trade-off between the low power design and the higher operating speeds or the minimum time delay. In most of these systems, low power consumption must be met whilst also achieving the equally challenging goals of high-speed operation and less time delay. In high-performance digital systems, such as microprocessors and digital signal processors (DSP) applications, the necessity for low-power design is becoming a critical challenge. As a result, low-power digital integrated circuit design has become a very active and ever evolving subject in VLSI design.

According to Landauer's principle, non-reversible logic computations must generate heat of the order of kT Joules for every bit of information lost, where k is Boltzmann's constant, and T is the absolute temperature for which the computations are done. The amount of heat dissipating at room temperature is small but not negligible [1]. Multipliers are the prime components in digital signal processors, microprocessors, micro controllers, etc. The architecture of the multiplier should be designed considering the power dissipation. Incorporating reversible logic into the architecture of the multiplier reduces the power losses to a significant value. The reversible logic can be the new normal in the terms power optimization circuits, nanotechnology, fast computing, and digital signal processing [2].

2 Vedic multiplication

The Vedic methods are based on natural principles that are followed by the human mind. Some of them, which are used for multiplication are listed down in Table 1 along with the name of their corollaries.

These principles are easier and faster in terms of the manual calculations involved in the multiplication than the conventional methods [3]. It's difficult to remember large numbers most of the time. However, for manual computations, picturing a line diagram and just adding two consecutive product terms is easier. This Vedic technique allows us to remember just few numbers. As a result, for manual calculations, Vedic multiplication is faster and more convenient [4–10]. "Urdhva Tiryakbhayam" is a popularly known method of Vedic math, where Urdhva means "vertically" and Tiryakbhayam means "diagonally" crosswise. Besides

✉ Ajay Kumar Kushwaha
akkushwaha@bvucoep.edu.in

Siddhesh Soyane
sjsoyanepg20-elect@bvucoep.edu.in

Dhiraj Manohar Dhane
dmdhane@bvucoep.edu.in

[1] Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India

Table 1 Vedic mathematics sutras and their corollaries

Serial no	Sutra	Corollary
1	Ekadhikena Purvena	Anurupaneya
2	Nikhilam	Sisyate Shesamjnah
3	Urdhva Tiryakbhayam	Adyamandeyanatyamantreyna
4	Parvartya Yojyet	Kevalaih Saptakam Gunyat
5	Purnapurnabhayam	Antyaordasakae

the commonly used Urdhva Tiryakbhayam sutra, Nikhilam sutra is also used for higher radix multiplication [11, 12].

3 Reversible logic

The relevance of reversible logic is established when we are dealing with low power, less area and time-efficient designs. Power dissipation, less chip area and time delay are significant design metrics in the digital design [13, 14].

The truth table of the combinational logic developed using the reversible logic is uniquely determined pattern [13]. The number of inputs and outputs is same in reversible logic gates. This discussion is restricted to two-valued logic functions describing switching logic.

A reversible logic circuit is characterized by the following:

1. Less number of reversible gates (less hardware complexity)
2. Less number of constant inputs.
3. Less number of garbage output.
4. Optimised Quantum cost.
5. Optimised Time Delay

The reversible logic gate employs a one-to-one mapping mechanism to help identify the outputs from the inputs. In reversible logic, the output vector is used to recover the input vector. It meets the requirements of Landauer's principle. Reversible logic is a promising option to the conventional digital logic for arithmetic operations with the optimized characteristics in terms of power, heat dissipation and delay.

Constant input: This can be defined as the number of inputs that are tied to either 0 or 1 value, to synthesize the given logical function.

Garbage output: The number of outputs that aren't employed in the synthesis of a function is referred to as garbage output. These are crucial; without them, reversibility is impossible to achieve.

Quantum cost: The quantum cost of a reversible circuit is calculated by counting the number of 2×2 logic gates such as controlled-Not, controlled-V, and controlled-V + gates required to implement the design [15].

Delay: It is defined in terms of the number of gates in the path from any input to any output, provided each gate performs computation in one unit of time and all inputs to the circuit are available before the computation begins [16].

Hardware complexity: Optimizing the hardware is one of the prime objectives of reversible logic. Hardware complexity is defined in terms of the number of EXOR, AND, and OR operations involving a design [17–19].

3.1 Toffoli gate

Toffoli gate is a 3×3 reversible gate. The first 2 output vectors are simple buffers, and the third output is the function of all three inputs. The inputs, A, B, and C are mapped to the outputs P, Q, and R as shown in Fig. 1. It is also considered as a universal reversible gate. The quantum cost of Toffoli gate is 5 [18].

3.2 Peres gate

The Peres gate is a 3×3 reversible gate with output "P" directly mapped to input "A". The remaining two output vectors mapping to combinational logic function with input variables. The block diagram of Peres gate is shown in Fig. 2. Quantum cost of Peres gate is 4 [16].

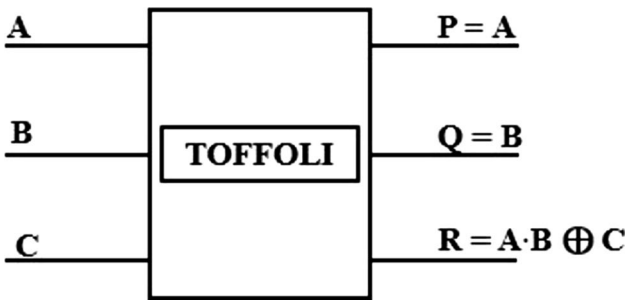

Fig. 1 Block diagram representation of the TOFFOLI gate

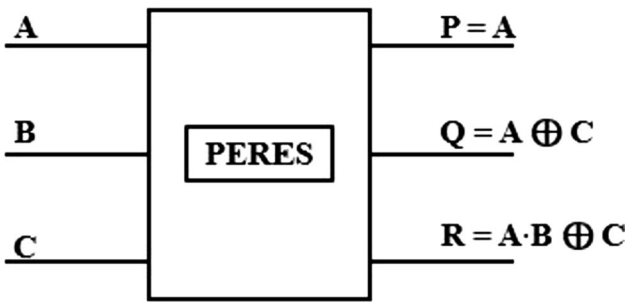

Fig. 2 Block diagram representation of the PERES gate

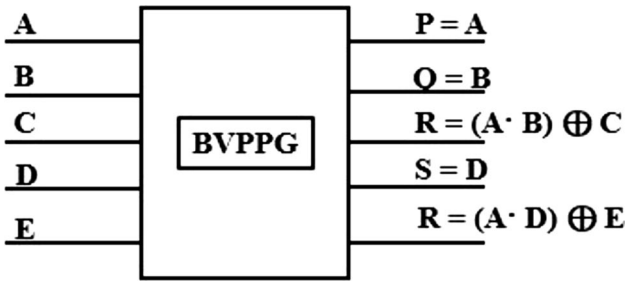

Fig. 3 Block diagram representation of the BVPPG gate

3.3 BVPPG gate

This gate has 5 input variables, (A, B, C, D, E), and its output vector is (P, Q, R, S, T). Figure 3 shows the implementation block of the BVPPG gate. It has the quantum cost of 5 [20–22].

4 QCA technology

Quantum Cellular Automata (QCA) is a substitute for conventional MOS-based designs of digital circuits and is a more power-efficient technology. The concept of Quantum-dot Cellular Automata (QCA) was introduced by Tougaw and Lent from Notre Dame University in 1993. One of the proposed implementations of the Quantum Cellular Automata is the Quantum-dot Cellular automaton. Quantum-dot Cellular Automata is a lower-level abstraction. Quantum dots are charge containers that have discrete electrical energy states [23]. A QCA cell is made up of four quantum dots. It is the basic computing element in QCA nanotechnology. In a cell, two electrons occupy diagonal position in the quantum cell owing to the Coulomb force, forming two configurations for encoding binary logical "0" and "1". The electrons in the cell interacts with each other using quantum–mechanical tunnelling. The polarization levels '+1' and '-1' represent the logic levels '1' and '0', respectively. Typically, QCA devices are described on the basis of symmetric square cells. Computational logic gates and memory structures can be correctly imitated with these symmetric square cells. Combinations of majority and not gates can realize any logic function. These structures can be implemented by assembling QCA cells in a specific geometric pattern to achieve the desired logic function. [24, 25].

In Fig. 4, the electron configurations within the cell for both polarization levels are depicted. Figures 5 and 6 show a basic inverter and a 3-input majority gate, respectively. In a majority gate, we have 3 inputs and 1 output, while the middle cell is the decision-making cell. AND and OR logic is realized with three input majority gates by setting the third input to '-1' and '+1' respectively.

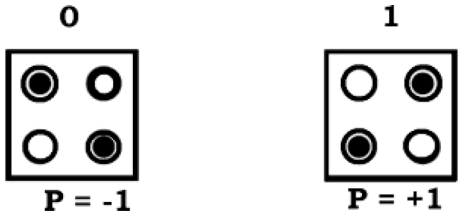

Fig. 4 Polarized QCA cell defining logic levels 0 and 1

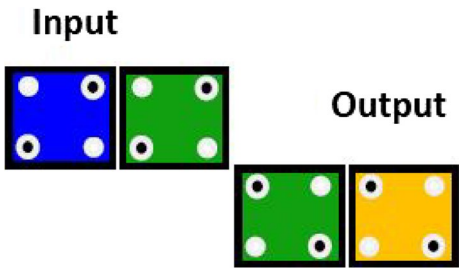

Fig. 5 QCA inverter

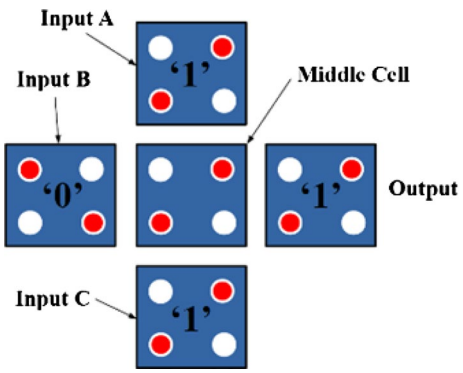

Fig. 6 QCA 3-input majority gate

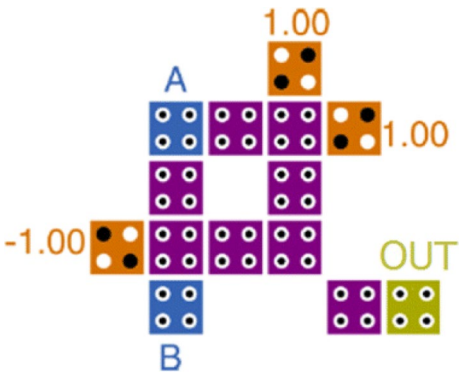

Fig. 7 QCA structure I of XOR gate

In Figs. 7 and 8, two strictures of the XOR gate, with cell counts of 14 and 13, respectively, are shown, which are used to realize the proposed designs. The latter structure, which is discussed in the results section, is used to optimise the design.

In QCA Designer, wire crossovers and interconnections in complex circuits can be realized using either a co-planar or multilayer approach. In a coplanar approach, two crossing wires are orthogonal to each other so that the crossing cells do not affect the neighbouring cells. The cells in the first wire are oriented at 90° and in the other crossing wire, they are oriented at 45° as depicted in Fig. 9. There is one more way for wire crossovers that does not require cell rotation. It is based on the advancing of the clocking phases from switch to relaxed and back to switch, which is discussed in Sect. 4.1.

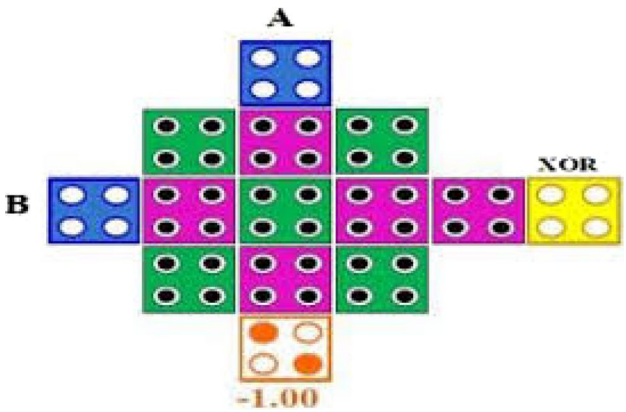

Fig. 8 QCA structure II of XOR gate

4.1 Clocking in QCA designer

Clocking in QCA is different than in conventional digital design. One of the main differences between them is that the latter circuit has no control over the clocks. This means that information is transmitted through each cell and not retained. Each cell erases its own state every clock cycle. Meta-stability is overcome by latching cell arrays to controlled clocking zones. It also facilitates the realization of a pipelined computing architecture. Clock 0 (switch), Clock 1 (hold), Clock 2 (released), and Clock 3 (relaxed) are the four clock zones that are applied systematically to each QCA cell in QCA circuits, with each zone having a phase difference of 90° with the others. This allows information to be pumped through the circuit as a result of the successive latching and unlatching of cells connected to different clock cycles. If a wire is clocked from left to right with ascending clocking zones, the information flows in the same direction as shown in Fig. 10. In QCA Designer, any single cell can be independently connected to any of the clocks, subject to the functionality of the circuit [26–28].

In this paper, wire crossovers in the proposed designs are based on the clocking zones, where cells latched to switch phase can cross cells latched to release phase, and cells on hold phase can cross cells in relaxed phase without having a polarization effect on the neighbouring cells [29]. The energy levels of a system are determined by the polarization and hence the interaction of the cells. In the ground

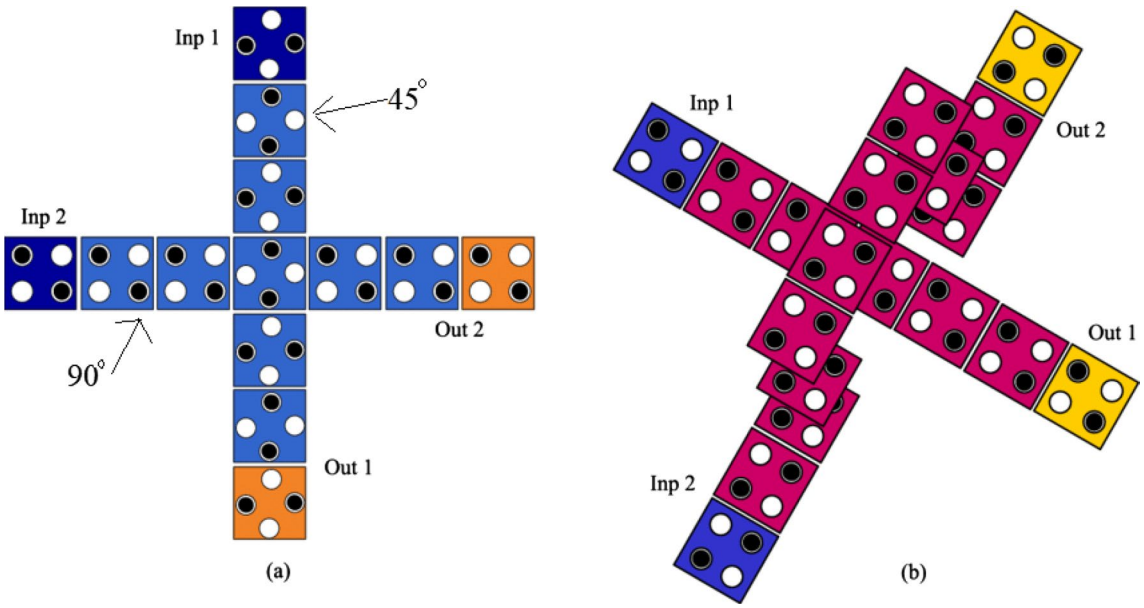

Fig. 9 Wire crossing **a** coplanar and **b** multilayer

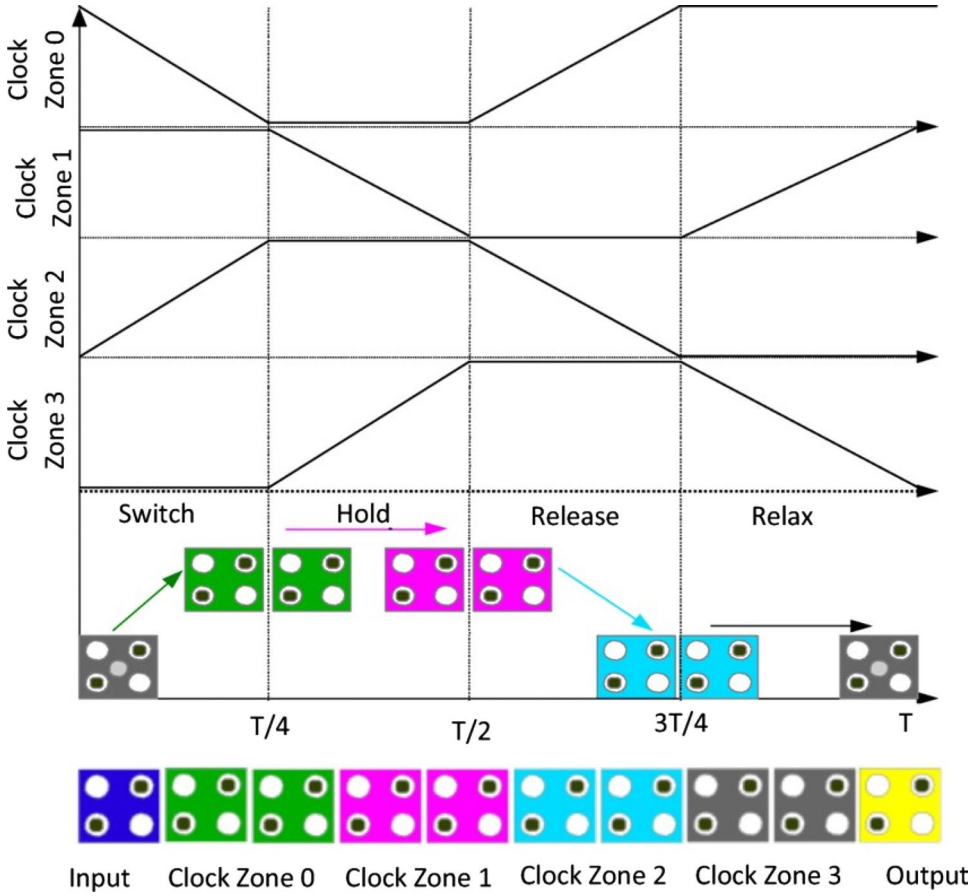

Fig. 10 QCA clocking with 4 phases

state, they are aligned, and in an excited state, they align oppositely to cell-to-cell repulsion and kink occurs [30, 31].

5 Proposed designs

The maximum benefits of power optimization can be obtained when implemented at the algorithmic and architectural level. The novel reversible gate design is a three-input three-output primary reversible gate with the attributes comparable to the Peres gate. The main feature of the proposed design is to improve functionality with optimised power and delay performance.

The proposed design, henceforward, shall be known as the SS gate. SS (Siddhesh Soyane) is the name of the proposed gate. Let us consider A, B, and C as the inputs and P, Q, and R as the outputs. The truth table for the SS gate is as given in Table 2.

The above truth table essentially translates into the following logic equations:

$$P = (A \cdot B') + (B \cdot C) \tag{1}$$

Table 2 Truth table of the proposed reversible gate

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

$$Q = (B' \cdot C) + (B \cdot C') \tag{2}$$

$$R = C' \tag{3}$$

The proposed SS gate with the input and output variables generates 3 outputs as shown in Fig. 11. First output, P, is the function of all 3 input variables. Second output is the EXOR between the inputs B and C while the third output R, is the complement of the third input variable C.

Quantum implementation of the SS gate uses 1 Controlled Not gate and 2 Controlled-V gates. The dotted box

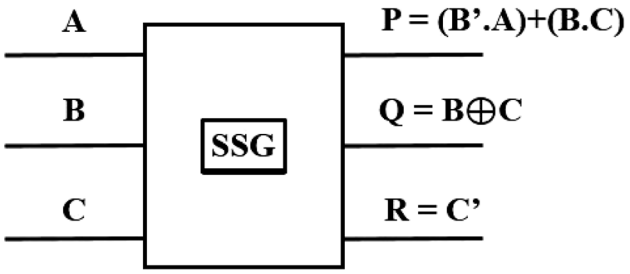

Fig. 11 Block diagram representation of the SS gate

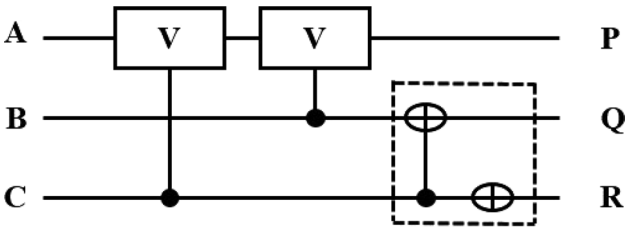

Fig. 12 Quantum implementation of SS gate

Table 3 A comparison table of basic reversible gates

Parameters	Peres	Toffoli	Fredkin	Proposed
Size	3×3	3×3	3×3	3×3
Delay	3	3	4	4
Quantum cost	4	5	5	3
Hardware complexity	3	2	6	4

in the Fig. 12 has a Controlled Not (2×2 reversible) gate with a QC of 1 and a Not (1×1 reversible) gate with QC of 0. Therefore, the QC of the dotted box is 1 and hence, the Quantum cost of the SS gate is 3. A comparison of the proposed SS gate with basic reversible gates is given in Table 3.

Figures 13 and 14 show the layout of the proposed reversible gate in the QCA design tool using both the above-discussed XOR gates, and its simulated waveform is shown in Fig. 15. It has three inputs, a, b, and c, and three outputs, p, q, and r. The logic is implemented using a majority gate with three inputs, an inverter, and an XOR gate.

5.1 SS Gate as half adder-subtractor

The improved functionality of the half adder-subtractor, both can be realized at the same time by using 2 SS gates as shown in the Fig. 16.

For SS gate to function as the half adder-subtractor we shall consider the variables B and C as it's two inputs. Therefore, the above circuit shall perform C + B and C−B

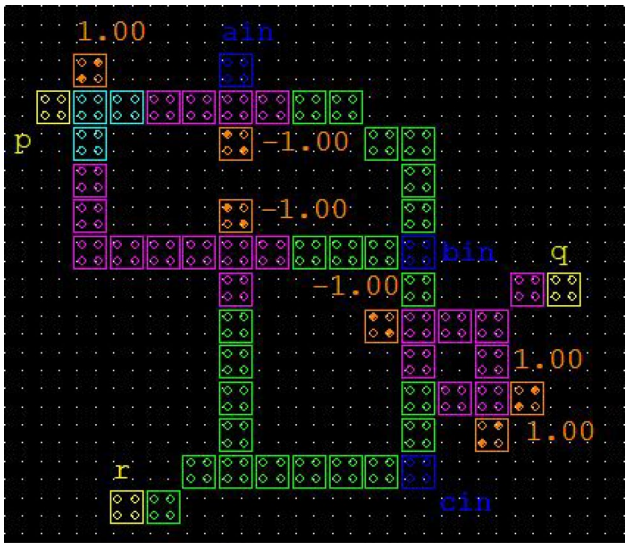

Fig. 13 Proposed QCA layout I of SS gate

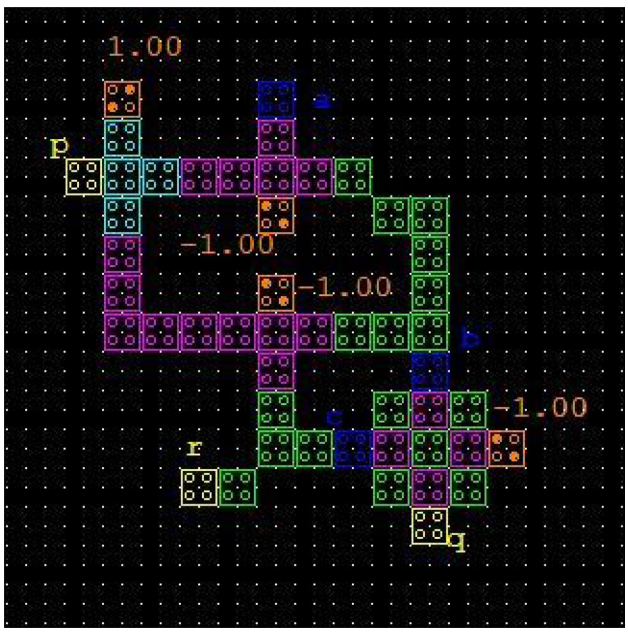

Fig. 14 Proposed QCA layout II of SS gate

operations. Input A is the constant input which is maintained at the '0' level. Consequently, it shall force the first half of the output Carry, (A'·B), to '0' value and only the (B·C) part produces the required carry for the addition. Similarly, in the Borrow, (A'·B), gives '0' value and (B·C'), remains as the borrow value. Sum or Difference (B ⊕ C) is the third valid output. G_1 and G_2 are the garbage values. Quantum cost of the proposed half adder-subtractor is 6.

The layouts of the half adder-subtractor using both the XOR structures implemented on the QCA tool are shown

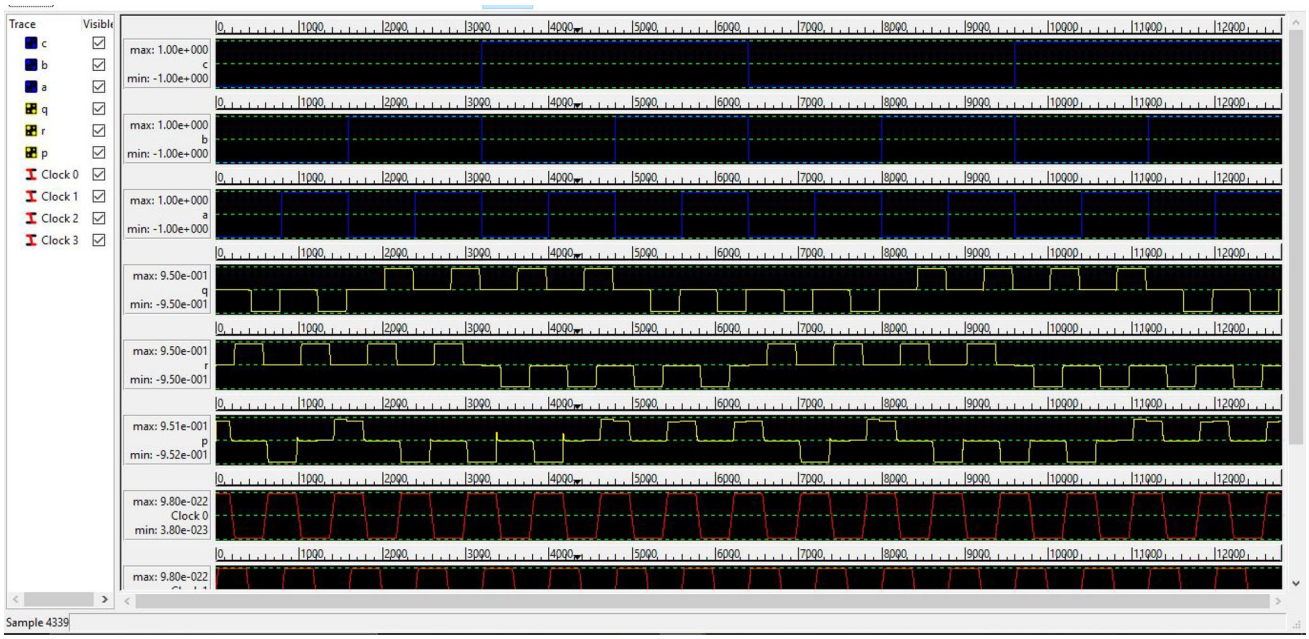

Fig. 15 Simulated waveform of SS gate

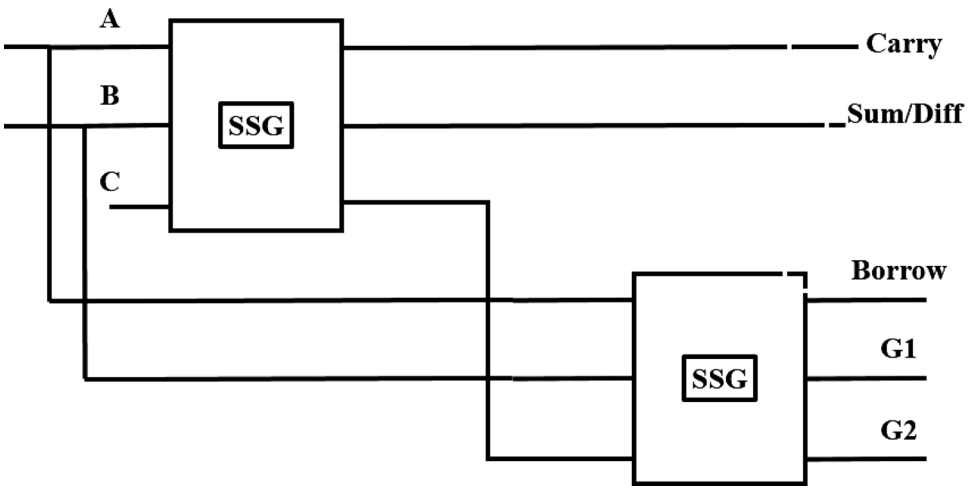

Fig. 16 Block diagram of half adder and half subtractor using SS gate

in Figs. 17 and 18, respectively. The simulated waveforms in Fig. 19.

Half adder-subtractor is also implemented using the SS gate. It requires two SS gates. Its QCA layout and simulated waveform are shown in Figs. 20 and 21, respectively.

5.2 Proposed 2-bit multiplier

This section proposes two circuits to implement 2-bit multiplier. The multiplication algorithm remains the same while the gates used and hence the parameters associated with the design are optimised in the later. The block diagram of layout I of the proposed 2-bit multiplier is shown in Fig. 22. It uses 2 SS gates and 2 BVPPG gates. The desired functionality is achieved with 4 input variables a_0, a_1, b_0, b_1, 6 constant inputs, $x_{1\text{-}6}$, maintained at value '0', and 8 garbage outputs, $g_{1\text{-}8}$.

The theoretical quantum cost of the design is 16 and the hardware complexity in terms of the logical operations as defined in the Sect. 3, $(6\alpha + 8\beta + 2\gamma) = 16$, where α is a 2 input XOR operation, β is a 2 input AND operation, and γ is a OR operation.

In Fig. 23, we have the QCA layout of the design I of 2-bit multiplier, it is a coplanar design employing the wire crossovers with the help of clocking zone, thus reduced complexity and ease of implementation.

The design II layout of the 2-bit multiplier is shown in the Fig. 24, it employs the 2 SS gates and 4 majority gates.

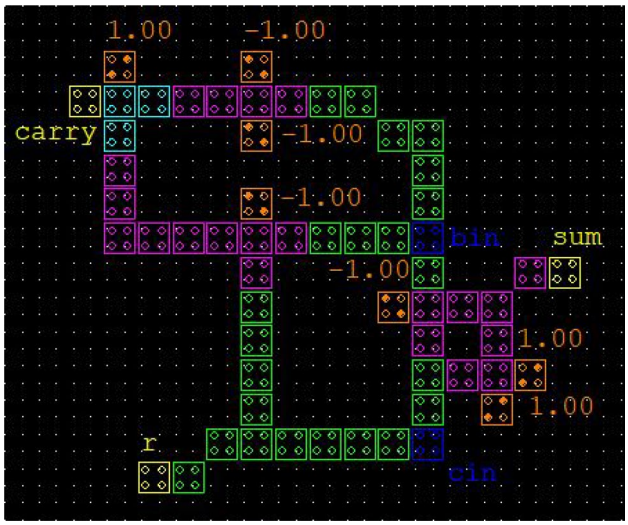

Fig. 17 QCA layout I of SS gate as half adder

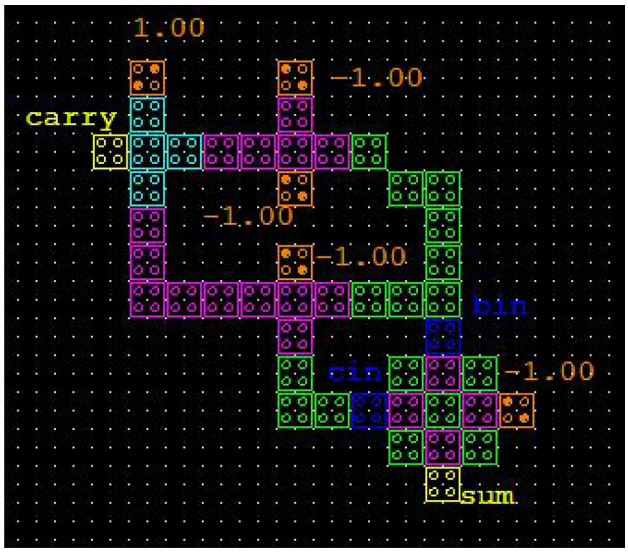

Fig. 18 QCA layout II of SS gate as half adder

It is also a coplanar design. Design layout II of the XOR gate is used. In this design too we use clock zones for wire crossovers, data and energy flow.

The simulated waveform is shown in Fig. 25. Total cell count, energy dissipation, area used, and latency and thus quantum cost is reduced in the design layout II compared to the design I.

5.3 General architecture for $2^n \times 2^n$ multiplier

In this section a general idea of $2^n \times 2^n$ is presented. The implementation of the 2-bit multiplier can be extended to 4-bits, 8-bits, and further. The idea of the $2^n \times 2^n$ (where n = 1, 2, 3…) can be formulated in the algorithm and generalised as shown in Fig. 26. The general block diagram of the $2^n \times 2^n$ consists of 4 multiplier blocks and 3 adder blocks. The multiplier blocks generate the partial products, and the two-stage adder adds the partial products in the required fashion.

For an n-bit multiplier, we need four (n/2) bit multiplier blocks. The input bit values to be fed to each of the multiplier blocks is shown in Fig. 26. The stage one adders consist of one n-bit adder and one (n + n/2) bit adder. The outputs of the stage one adders are then fed to a (n + n/2) adder at the second stage. The second stage adder gives the product bit values of q [(2n−1): (n/2)]. The first multiplier block directly yields the product bits q [(n/2)−1:0].

6 Results and discussions

The functional correctness of the SS gate, implementation of half adder-subtractor, 2-bit multipliers can be verified manually and through any simulation tool. QCA Designer was used to verify the theoretical and practical outcomes of the proposed designs. For the verification of the proposed designs the default parameters have been taken into consideration [23]. The designs are implemented in single layer with coplanar wire crossing only.

QCA Designer-E, an extended module of QCA designer was used for timing, energy dissipation analysis. QCA circuits are simulated using the bi-stable approximation and coherence vector simulation engines. Energy dissipation can be calculated using QCA Designer–E using Coherence vector energy simulation engine setup and considering parameters [27] shown in Fig. 27. The simulation was performed in coherence vector energy mode with the energy display option for each cell turned on. Total energy dissipation, S_b along with the error S_{bE} and average energy dissipation per cycle, A_b and error A_{bE} are associated with the energy dissipation. The design analysis, S_b, A_b, cell count and area associated with the proposed circuits are presented in Table 4. The designs used for comparison were taken from the papers citied in the reference section, simulated again on the QCA designer and then analysis was tabulate in the comparison tables.

In Table 5, the proposed design of the reversible gate used as a half adder is compared with the conventional logical implementation of the half adder [32, 33]. Though, conventional logic requires less number of cells and area, by the virtue, design based on reversible logic dissipates less energy. Further, in Table 6, the half adder-subtractor proposed in the paper is compared with a half-adder subtractor based on the reversible logic presented in [27]. Quantum cost is calculated as the product of area and latency (clock delay) [34]. Cell count and energy

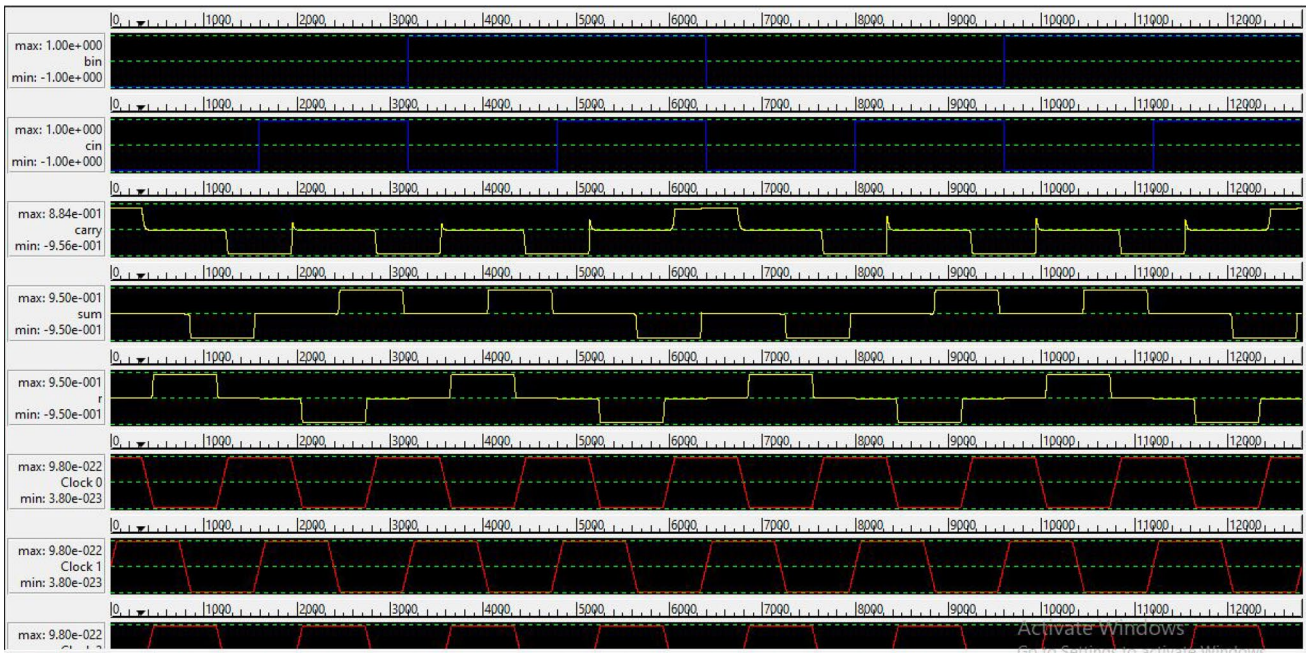

Fig. 19 Simulated waveform of half-adder

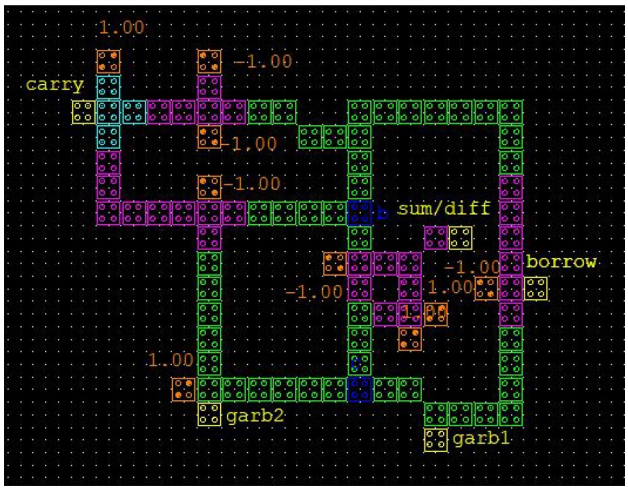

Fig. 20 Layout of half adder-subtractor

dissipation are found to be less in the proposed design. As discussed in Sect. 5.2, two layouts of a 2-bit multiplier are implemented using two different structures of the XOR gate; the analysed features of both layouts are compared with a design presented in [33]. The analysis is tabulated in Table 7, where we find the design II with better optimized parameters.

The design I of the 2-bit multiplier was also implemented on Xilinx ISE and the analysis and comparison with the designs in [20] are tabulated in the Table 8.

6.1 Implementing the design on kintex7 (KC705)

The general block diagram presented in Sect. 5.3 for the n-bit multiplier architecture is implemented on Kintex-7, the design utility and delay are compared with the multiplier-less squaring architectures in [13] and [14]. Delay and the number of LUTs used are compared in Tables 9 and 10, respectively.

It is observed that the proposed architecture produces more delay for lower bits, but with an increase in the bit value, the delay reduces and the area is also optimized.

7 Conclusion

The novel 3×3 reversible gate presented in the paper is compared with the existing basic reversible gates. It has been modified to work as a half-adder and half adder-subtractor. The proposed designs are implemented with two different XOR structures in the QCA layout. Designs realised with the latter structure are more optimised. The presented designs are comparable and better than some existing designs, as

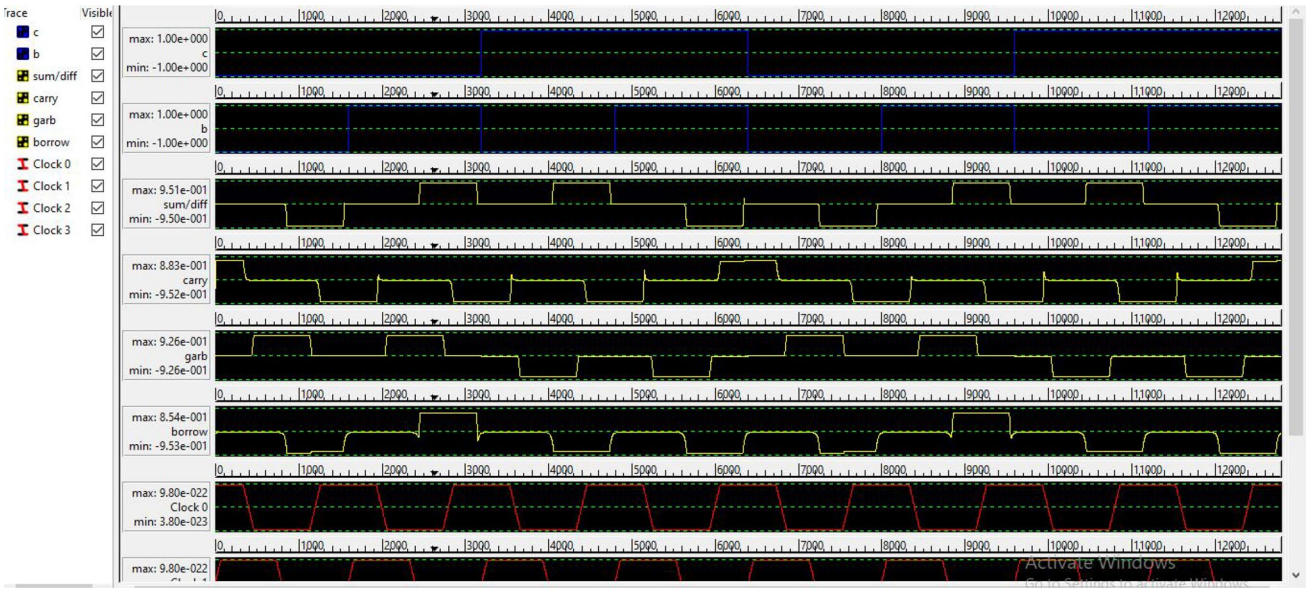

Fig. 21 Simulated waveform of half adder-subtractor

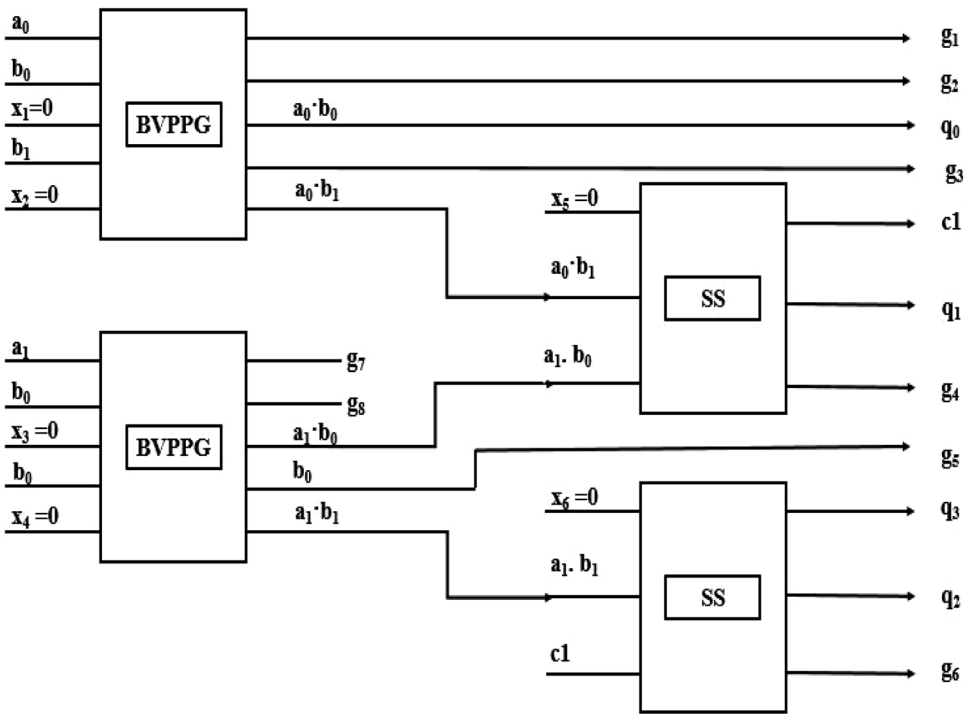

Fig. 22 Block diagram of 2-bit multiplier

found in the result and discussion sections. Design II of the 2 × 2 multiplier was optimised for the cell count and area, and hence the energy dissipation, over Design I. According to this study, clock zone-based crossover can reduce the complexity and improve the performance of the design. Further, a general 2^n bit multiplier architecture is presented exhibiting a simple 2 step optimised design. It is compared with a squaring architecture; as the value of bit n increased, the performance metrics improved.

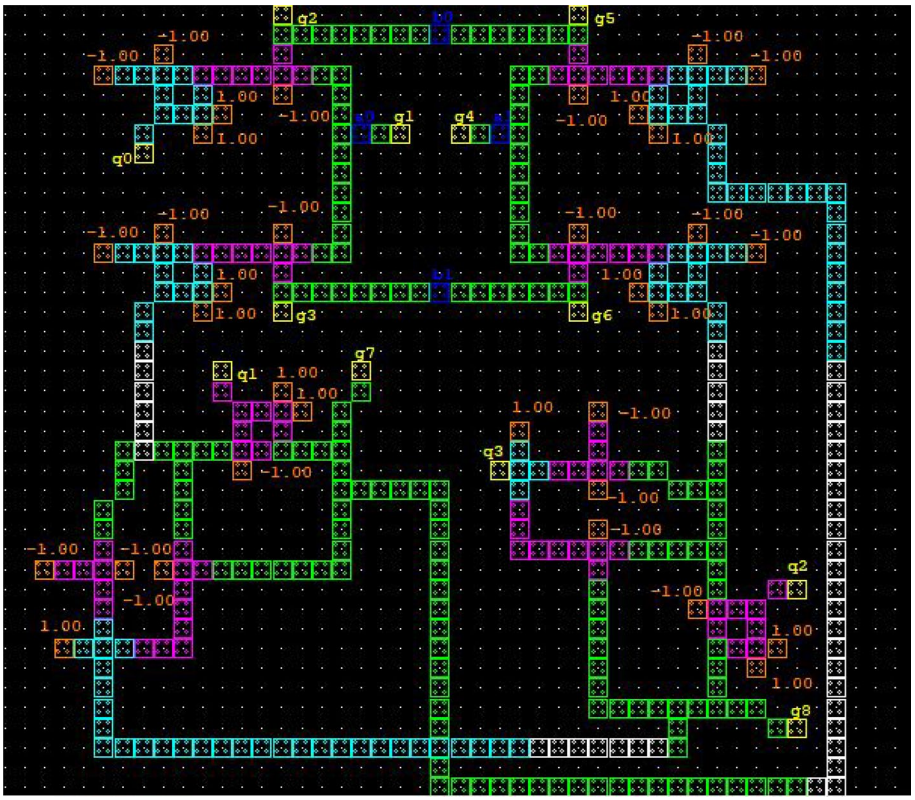

Fig. 23 QCA layout I of 2-bit multiplier

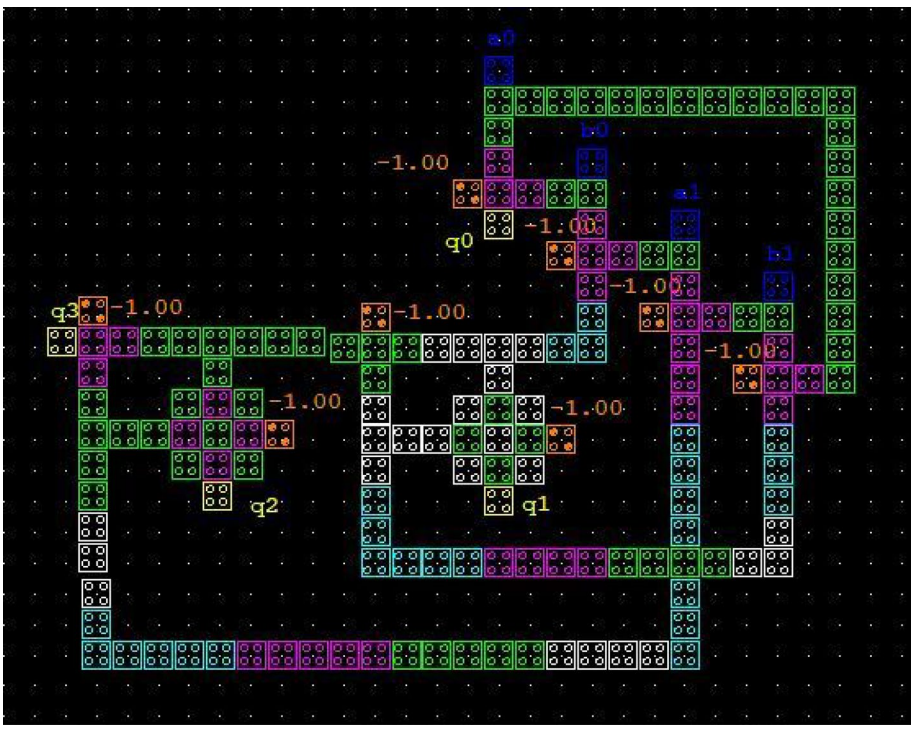

Fig. 24 QCA layout II of the 2-bit multiplier

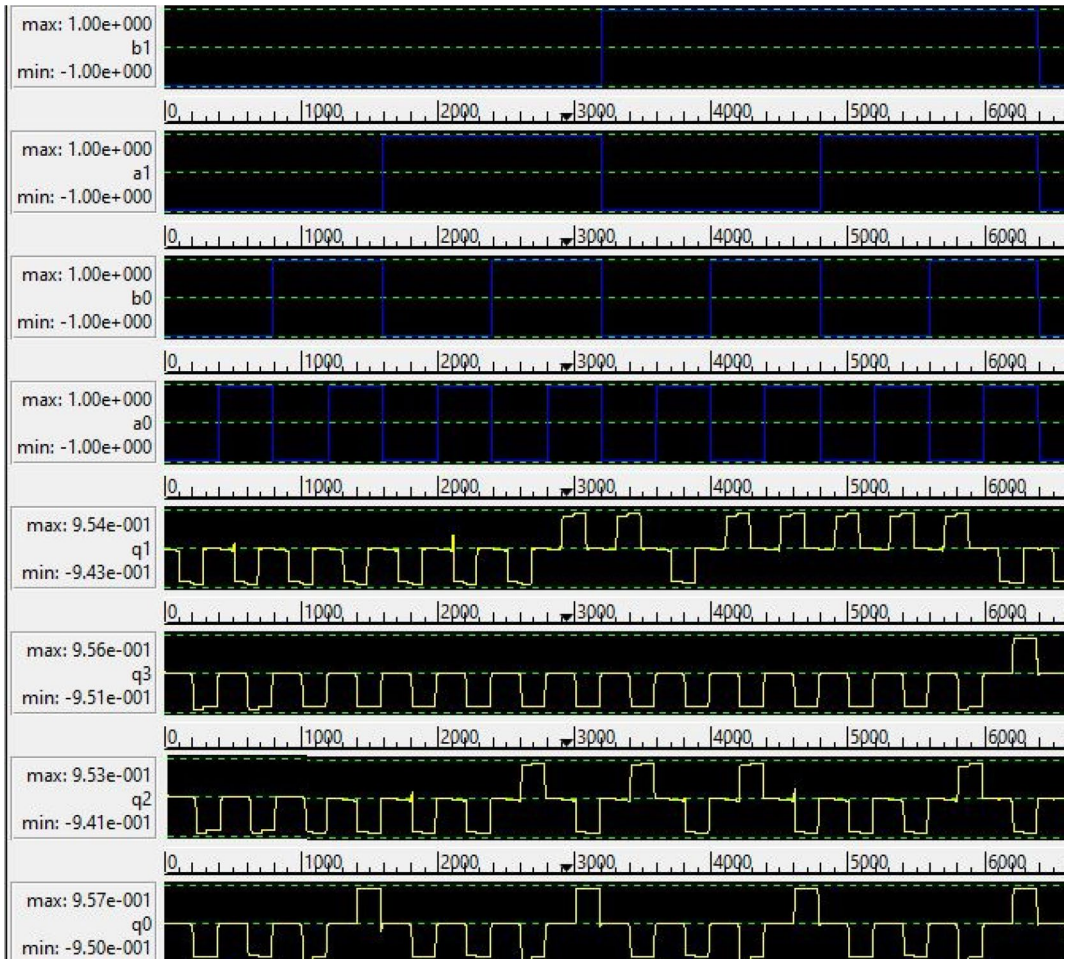

Fig. 25 Simulated waveform of 2-bit multiplier

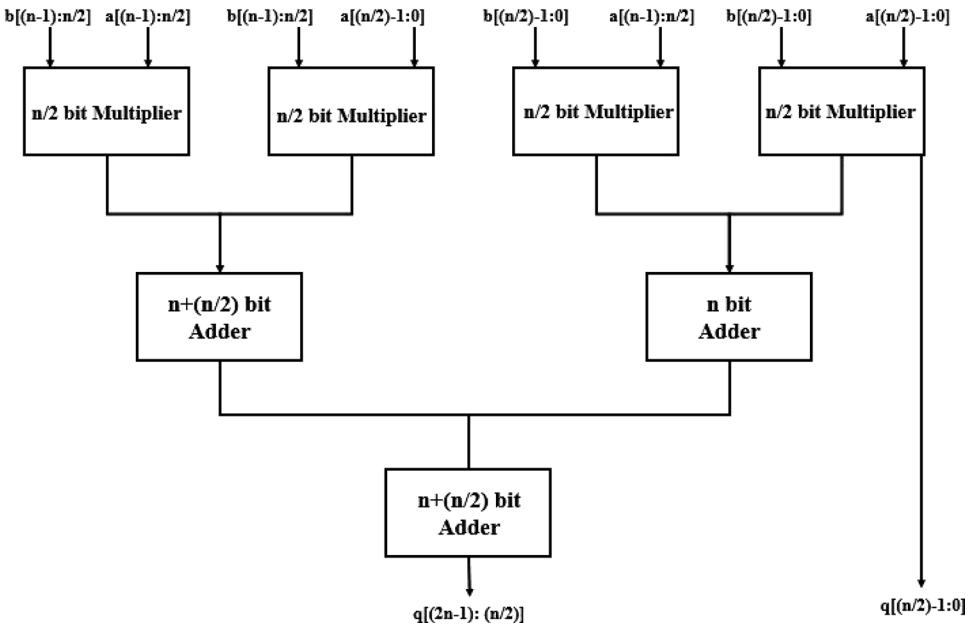

Fig. 26 General block diagram for $2^n \times 2^n$ multiplier

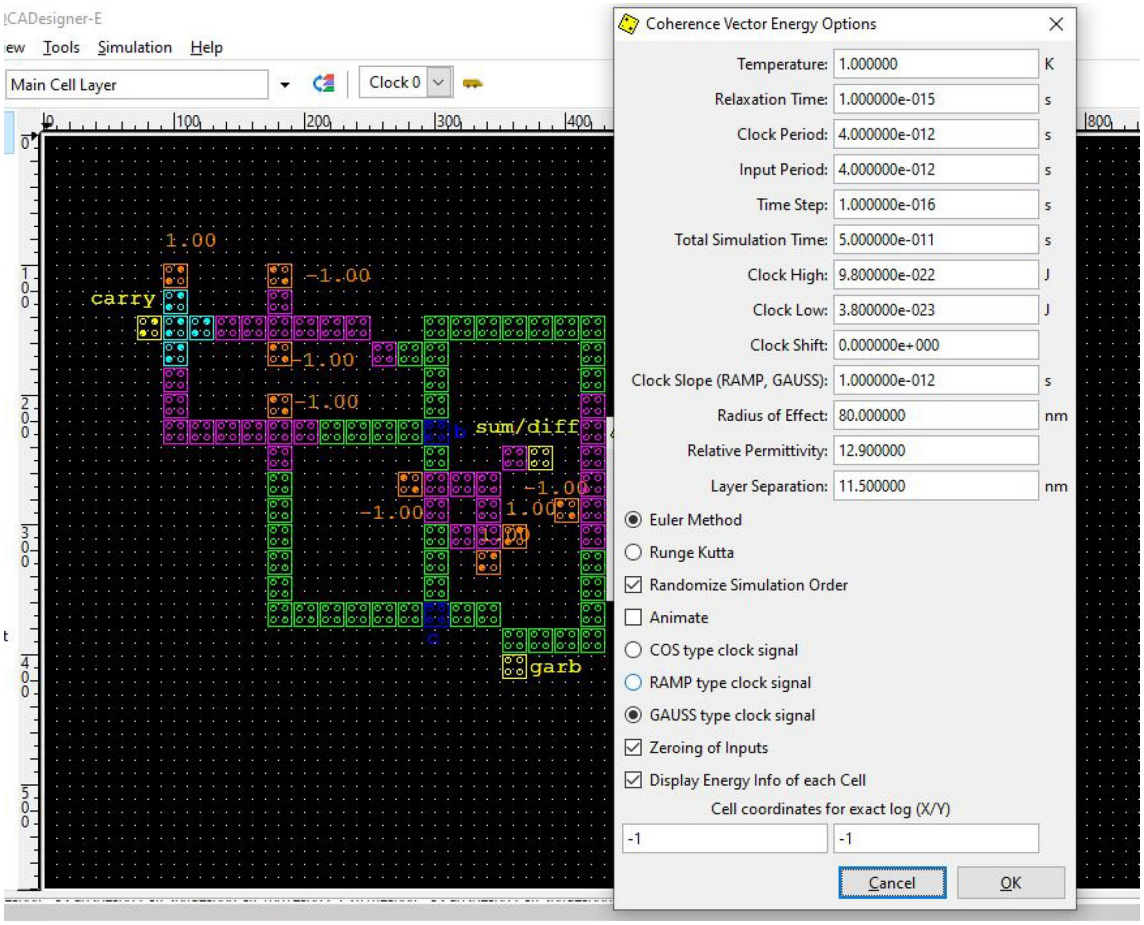

Fig. 27 QCA Designer engine setup for energy analysis

Table 4 Design analysis using QCA designer-E

Designs	Total energy dissipation	Average energy dissipation/cycle	Area occupied	Cell count	Quantum cost
SS gate I	0.0307 eV	0.0027 eV	0.10 μm^2	59	0.05
SS gate II	0.0293 eV	0.0026 eV	0.08 μm^2	49	0.04
Half adder [I]	0.0307 eV	0.0027 eV	0.10 μm^2	59	0.10
Half adder [II]	0.0293 eV	0.00267 eV	0.08 μm^2	47	0.08
Half add-sub	0.0383 eV	0.0034 eV	0.14 μm^2	89	0.14
2-bit multiplier [I]	0.1544 eV	0.0144 eV	0.64 μm^2	396	0.96
2-bit multiplier [II]	0.0675 eV	0.00614 eV	0.22 μm^2	162	0.110

Table 5 Half adder comparison between [32] and proposed design in QCA

Parameters	[32]	Proposed design I	Proposed design II
No of cells	48	59	47
Area	0.07 μm^2	0.10 μm^2	0.08 μm^2
Energy dissipation	0.0296 eV	0.033 eV	0.0293 eV
Quantum cost (area x latency)	0.07	0.10	0.08
Wire crossover	Coplanar based on clock zones	Coplanar based on clock zones	Coplanar based on clock zones

Table 6 Half adder-subtractor comparison between [27] and proposed design in QCA

Parameters	[27]	Proposed design
No of cells	137	89
Area	0.17 μm^2	0.14 μm^2
Energy dissipation	0.0607 eV	0.038 eV
Quantum cost (area x latency)	0.17	0.14
Wire crossover	Coplanar based on clock zones	Coplanar based on clocking zones

Table 10 Comparison of device utilization between squaring architectures designs in [13, 14] and the proposed design implemented on Kintex7 (KC705)

No. of bits	No. of LUTs in [13]	No. of LUTs in [14]	No. of LUTs proposed design
4	6	6	16
8	62	98	94
16	207	485	394
32	1343	1724	1131
64	1936	2876	1632

Table 7 2-bit multiplier comparison between [33] and proposed designs in QCA

Parameters	[33]	Proposed design I	Proposed design II
No of cells	203	396	162
Area	0.29 μm^2	0.64 μm^2	0.22 μm^2
Energy dissipation	0.0835 eV	0.1544 eV	0.0675 eV
Quantum cost (area x latency)	0.145	0.96	0.110
Wire crossover	Coplanar based on clock zones	Coplanar based on clocking zones	Coplanar based on clocking zones

Table 8 2-bit multiplier comparison between designs [20] and the proposed design

Parameter	Model 1 of [20]	Model 2 of [20]	Proposed design I of 2-bit multiplier
Constant input	6	6	6
Garbage output	10	8	8
Reversible gates	6	4	4
Path delay (ns)	8.752 ns	6.096 ns	5.05 ns
No of LUTs used	–	–	4
No of IO buffers	–	–	8
Quantum cost	24	18	16
Hardware complexity	20	14	16

Table 9 Delay comparison between squaring architecture designs in [13, 14] and the proposed design implemented on Kintex-7 (KC705)

No. of bits	Delay (ns) in [13]	Delay (ns) in [14]	Delay (ns) in proposed design
4	4.2	4.9	6.61
8	8.6	8.9	11.685
16	19.9	22.4	15.51
32	32.7	56.1	22.05
64	38.6	93.2	38.03

Acknowledgements The authors wish to thank the anonymous reviewers and the Editor-in-Chief very much for their helpful comments and suggestions.

Author contributions We have all worked on and reviewed this research article. The idea of "proposing a novel reversible gate" was done by Mr. Siddhesh. Simulated works in the QCAD tool are completed by Mr. Siddesh and Dr. Ajay Kumar Kushwaha. Paper writing is completed by Dr. Ajay Kumar Kushwaha and Dr. Dhiraj Manohar Dhane.

Funding Funding is not received for this research.

Data availability No data available.

Declarations

Conflict of interest All Authors of this work declare no conflict of interest.

References

1. Landauer, R. (1961). Irreversibility and heat generation in the computational process. *IBM Journal of Research and Development, 5*, 183–191. https://doi.org/10.1147/rd.53.0183
2. Saranya, K., & Vijeyakumar, K. (2021). A low area FPGA implementation of reversible gate encryption with heterogeneous key generation. *Circuits System and Signal Processing, 40*, 3836–3865. https://doi.org/10.1007/s00034-021-01649-1
3. Mehta, P., & Gawali, D. (2009). Conventional versus Vedic mathematical method for Hardware implementation of a multiplier. In *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies* (pp. 640-642). IEEE. https://doi.org/10.1109/ACT.2009.162.
4. Maharaja, J. S. S. B. K. T. (1986). *Vedic mathematics or sixteen simple sutras from the vedas*. Varanasi, India: Motilal Banarsidass.
5. Sujitha, S., & Kalith, B. (2021). High speed Power efficient Vedic arithmetic modules on Zedboard-Zynq-7000 FPGA. *International Journal of Circuit Theory and Applications*. https://doi.org/10.1002/cta.3110
6. Radwa, M. T., & Marwa, A. E. (2022). VHDL implementation of 16x16 multiplier using pipelined 16x8 modified Radix-4 booth multiplier. *International Journal of Electronics*. https://doi.org/10.1080/00207217.2022.2068198
7. PourAliAkbar, E., Navi, K., Haghparast, M., & Reshadi, M. (2020). Novel optimum parity-preserving reversible multiplier circuits. *Circuits System and Signal Processing, 39*, 5148–5168. https://doi.org/10.1007/s00034-020-01406-w
8. Kamaraj, A., Parimalah, A. D., & Priyadarshani, V. (2017). Realisation of Vedic Sutras for multiplication in Verilog. *SSRG International Journal of VLSI and Signal Processing, 4*(1), 25–29. https://doi.org/10.14445/23942584/IJVSP-V4I2P106
9. Bisoyi, A., Baral, M., & Senapati, M. K. (2014, May). Comparison of a 32-bit Vedic multiplier with a conventional binary multiplier. In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies* (pp. 1757-1760). IEEE. https://doi.org/10.1109/ICACCCT.2014.7019410.
10. Saranya, K., & Vijeyakumar, K. N. (2021). A novel *n*-decimal reversible radix binary-coded decimal multiplier using radix encoding scheme. *Circuits, System, and Signal Processing, 40*, 1743–1761. https://doi.org/10.1007/s00034-020-01549-w
11. Sahu, S. R., Bhoi, B. K., & Pradhan, M. (2020). Fast signed multiplier using Vedic Nikhilam algorithm. *IET Circuits Devices System, 14*, 1160–1166. https://doi.org/10.1049/iet-cds.2019.0537
12. Pradhan, M., & Panda, R. (2014). High speed multiplier using Nikhilam Sutra algorithm of Vedic mathematics. *International Journal of Electronics, 101*(3), 300–307. https://doi.org/10.1080/00207217.2013.780298
13. Reddy, B. N. K. (2020). Design and implementation of high performance and area efficient square architecture using Vedic Mathematics. *Analog Integrated Circuit and Signal Processing, 102*, 501–506. https://doi.org/10.1007/s10470-019-01496-w
14. Sethi, K., & Panda, R. (2015). Multiplier less high-speed squaring circuit for binary numbers. *International Journal of Electronics, 102*(3), 433–443. https://doi.org/10.1080/00207217.2014.897381
15. Dastan, F., & Haghparasat, M. (2011). A novel nanometric fault tolerant reversible divider. *International Journal of Physical Science, 6*(24), 5671–5681.
16. Biswas, A. K., Hasan, M. M., Chowdhury, A. R., & Babu, H. M. H. (2008). Efficient approaches for designing reversible Binary coded decimal adders. *Microelectronics Journal, 39*(12), 1693–1703. https://doi.org/10.1016/j.mejo.2008.04.003
17. Shamsujjoha, M., Babu, H. M. H., & Jamal, L. (2013). Design of a compact reversible fault tolerant field programmable gate array: A novel approach in reversible logic synthesis. *Microelectronics Journal, 44*(6), 519–537.
18. Fredkin, E., & Toffoli, T. (1982). Conservative Logic. *International Journal of Theoretical Physics, 21*, 219–253.
19. Abed, S., Khalil, Y., Modhaffar, M., & Ahmad, I. (2018). High-performance low-power approximate Wallace tree multiplier. *International Journal of Circuit Theory and Applications, 46*, 1–15. https://doi.org/10.1002/cta.2540
20. Kumar, K., Nagabhushana, M. R., Kedlaya, S.G. (2016). A Novel 2X2 Vedic multiplier architecture based on reversible logic. *International Journal of Electrical Electronics and Computer Science Engineering,* pp 20–23.
21. Dole, S., Shembalkar, S., Yadav, T., & Thakre, P. (2017). Design and FPGA implementation of 4X4 Vedic multiplier using different architectures. *International Journal of Engineering and Technical Research, 6*(4), 812–816. https://doi.org/10.17577/IJERTV6IS040673
22. Gunasekaran, K., Sudheer, C. L., Sornagopal, V., & Gnanasekaran, M. (2020). Design of 4-bit multiplier accumulator unit by using reversible logic gates in peres logic. *European Journal of Molecular and Clinical Medicine, 7*(9), 2415–2422.
23. Sasamal, T. N., Singh, A. K., & Mohan, A. (2020). Quantum-dot cellular automata based digital logic circuits: a design perspective. *Studies in Computational Intelligence*. https://doi.org/10.1007/978-981-15-1823-2_2
24. Singh, G., Sarin, R. K., & Raj, B. (2017). Design and analysis of area efficient QCA based reversible logic gates. *Microprocessors and Microsystems., 52*, 59–68. https://doi.org/10.1016/j.micpro.2017.05.017
25. Gassoumi, I., Touil, L., & Mtibaa, A. (2021). An efficient design of QCA full-adder-subtractor with low power dissipation. *Journal of Electrical and Computer Engineering*. https://doi.org/10.1155/2021/8856399
26. Walus, K., Dysart, T. J., & Juillien, G. A. (2004). QCA designer: A rapid design and simulation tool for quantum-dot cellular automata. *IEEE Transactions on Nanotechnology, 3*(1), 26–31. https://doi.org/10.1109/TNANO.2003.820815
27. Singh, S., Choudhary, A., & JainK, M. (2019). An optimized approach towards reversible adder/subtractor design on QCA. *IJ Modern Education and Computer Science, 10*, 47–53. https://doi.org/10.5815/ijmecs.2019.10.06
28. Vankamamidi, V., Ottavi, M., & Lombardi, F. (2006). Clocking and cell placement for QCA. *IEEE Conference on Nanotechnology*. https://doi.org/10.1109/NANO.2006.247647
29. Shin, S. H., Jeon, J. C., & Yoo, K. Y. (2014). Design of wire-crossing technique based on difference of cell state in quantum-dot cellular automata. *International Journal of Control and Automation, 7*(4), 153–164.
30. Taherkhani, E., Moaiyeri, M. H., & Angizi, S. (2017). Design of an ultra-efficient reversible full adder-subtractor in quantum-dot cellular automata. *Optik, 142*, 557–563. https://doi.org/10.1016/j.ijleo.2017.06.024
31. Menville, D., Mamum, M., & S. (2013). Quantum cost optimization for reversible sequential circuits. *International Journal of Advanced Computer Science and Applications, 4*(12), 15–21.
32. Moustafa, A. (2019). Efficient quantum-dot cellular automata for half adder using building block. *Quantum Information Review International Journal, 7*(1), 1–6.
33. Safoev, N., & Jeon, J. (2020). Design and evaluation of cell interaction based Vedic multiplier using quantum-dot cellular

automata. *Electronics, 9*, 1036. https://doi.org/10.3390/electronics9061036
34. Tripathi, D., & Wariya, S. (2021). An energy dissipation and cell optimization of Vedic multiplier topologies for nano computing applications. *Turkish Journal of Computer and Mathematics Education, 12*(14), 1490–1510.

Siddhesh Soyane received his Bachelor of Engineering degree from Mumbai University in 2019. He is currently pursuing the M.Tech. from Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India, wherein the conducted research is a part of the course with a good academic record. He is a verification engineer working in ASIC IP verification. His areas of research interests are reversible logic, power optimization and flash memories.

Ajay Kumar Kushwaha was born in 1986. He received his Doctor of Philosophy (Ph.D.) degree from the Department of Electronics Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, India, in 2017. In 2011, he received an M.Tech. degree from S.G.S.I.T.S. Indore in Microelectronics and VLSI Design. He received his B.E. degree from Indira Gandhi Govt. Engineering College Sagar in Electronics and Communication Engineering in 2009. He is currently an Associate professor in the Department of Electronics and Telecommunication Engineering at Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India. He has worked at the Government Engineering College in Raipur, the SRITW Warangal, the SR Engineering College in Warangal, and the Medi-Caps Group of Institutions in Indore. He has a keen research interest in analog and digital VLSI design, analog integrated circuits, chaotic circuits, current mode building blocks, etc. He has published research papers in international journals and international conferences. He has successfully executed two research projects sponsored by agencies of the Government of India (NPIU-MHRD).

Dhiraj Manohar Dhane received the Bachelor's degree (B.E.) from Govt. College of Engineering Jalgaon, India in 2001, Master's degree (M.Tech.) from Visvesvaraya Technological University Belgaum, India in 2006 and Ph.D. degree from Indian Institute of Technology Kharagpur, India in 2017. He is a senior member of IEEE and IEEE-EMB Society. He is currently an Associate Professor at the Bharati Vidyapeeth (Deemed University), College of Engineering, Pune, India. His current research interests include Machine learning in VLSI and developing intelligent transport system applications.