European Journal of Sustainable Development Research
2026, 10(1), em0339
e-ISSN: 2542-4742
https://www.ejosdr.com/

MODESTUM

Research Article

OPEN ACCESS

Design of optimized reversible multipliers using KMD gates for DSP applications

Kamaraj A [1*], Ilanchezhian P [2], Vallikkannu M [3], Thirumal L [4]

[1] MSEC, Sivakasi, INDIA
[2] Department of Information Technology, Sona College of Technology, Salem, INDIA
[3] Department of English, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, INDIA
[4] Department of ECE, Varuvan Vadivelan Institute of Technology, Dharmapuri, Tamil Nadu, INDIA
***Corresponding Author:** kamarajvlsi@gmail.com

Citation: A, K., P, I., M, V., & L, T. (2026). Design of optimized reversible multipliers using KMD gates for DSP applications. *European Journal of Sustainable Development Research, 10*(1), em0339. https://doi.org/10.29333/ejosdr/17276

ARTICLE INFO

Received: 19 May 2025

Accepted: 03 Oct. 2025

ABSTRACT

One kind of logic architecture, known as reversible logic, maps the output and input vectors one-to-one. According to the Landauer's principle from 1961, if a device is irreversible in its characteristics, then deleting a small amount will cause the release of heat energy of the order of kTln2 joules. Here, T represents the absolute temperature of the environment, and k is referred to as the Boltzmann constant. Quantum logic gates, which each carry out a fundamental unitary operation on 1, 2, or more qubits–two-state quantum systems–can be conceived of as the building blocks of a quantum computer, or as kinfolk's networks of quantum. Quantum networks need the use of reversible logic components. The multiplication procedure in this study makes use of KMD gates. The KMD gate can be used to create a variety of logic circuits, including AND, half adder, and full adder. The criteria for reversibility, fault tolerance, and universality are satisfied by these gates. This research presents the design and simulation of a unique reversible fault-tolerant Wallace multiplier, Baugh-Wooley multiplier, Vedic multiplier, and minimal complexity multiplier. The above multipliers are designed to obtain the addition and product terms with the least amount of delay and optimal slice utilization for sustainable technology. The complete design of the fundamental reversible logic gates was confirmed using the QCADesigner software version 2.0.3. The current designs of several multipliers are compared in this paper based on a number of parameters, such as garbage output (GO), gate count (GC), quantum cost (QC), area, and constant input (CI). The simulation results show improvements in area, delay, CIs, GO, QC, and GC over currently available architectures.

Keywords: vedic multiplier, Wallace multiplier, minimum complexity multiplier, Baugh-Wooley multiplier, quantum cellular automata, reversible logic, KMD gates

INTRODUCTION

Power dissipation occurs from irreversible hardware computing because of information loss. Irreversible hardware computation causes energy dissipation, as Landauer (1961) showed in the early 1960s, regardless of the realization mechanism. When the circuit implements non-bijective functions, information is lost. An irreversible logic circuit's output vectors cannot be used to retrieve the input vector from them. Reversible circuits can only be used to realize functions if the mapping between the input and output assignments is one-to-one. Reversible circuits do not lose any information as a result. Bennett (1973) states that the network could only have zero energy dissipation if it were made up of reversible gates. Therefore, reversibility will become a crucial component of circuit design. For any circuit to be reversible, certain requirements must be met.

If the output and input of a deterministic device can be independently extracted from each other, then the device is deemed logically reversible. The second requirement is that for a device to be considered physically reversible, it must be able to run backward. The most common issues with CMOS technology are stray capacitance, design complexity, interconnection noise, and gate leakage current. With its increasing popularity as a component of sophisticated design projects, quantum-dot cellular automata (QCA) appears as a strong contender for the upcoming cohort of digital structures. The most crucial QCA metrics are diminutive size, increased speed, superb switching, incredibly scalable features, and low power consumption. The location of electrons is used for logic assessment. According to recent studies, QCA circuits may operate at room temperature, achieve great densities, and switch quickly.

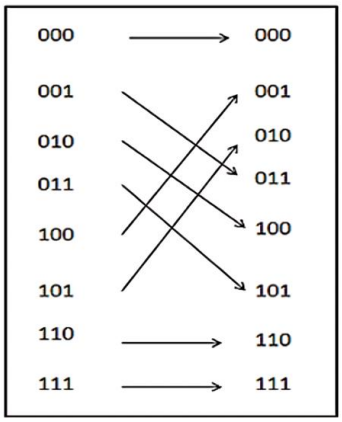
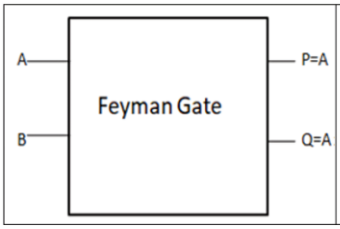


Figure 1. Example of reversible logic mapping (Source: Authors' own elaboration)



Inputs		Outputs	
A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Figure 2. Feynman gate (no information is lost–2 × 2) (Source: Authors' own elaboration)

Properties of Reversible Logic

Reversible logic gates and circuits have three important characteristics, that is

(1) one to one mapping,

(2) no information loss, and

(3) fault-tolerant.

Individual mapping

Reversible systems, also known as gates, are shown in **Figure 1** with a one-to-one mapping between the input and output vectors. Because of this, the output state vector may always be used to rebuild the input state vector (Riyaz & Sharma, 2023).

No information loss

In reversible computing, no information will be lost at any point. As seen in **Figure 2**, the reversible logic circuits always contain some inputs that are identical to the number of outputs.

Fault-tolerant

A fault in the circuit can be easily foreseen if it is considered to be fault-tolerant. A gate must meet two requirements to be considered fault-tolerant.

1. **XOR of input will match with the XOR of output:** For example, consider a 3 × 3 reversible gate with the following combinations of inputs: A, B, and C; outputs: P, Q, and R. Consequently, the condition $A \wedge B \wedge C = P \wedge Q \wedge R$ will be satisfied.
2. **Number of inputs should be equal to the number of outputs.**

Reversible Systems Performance Metrics

The following are quantifiable parameters in reversible logic. Garbage output (GO), the number of gates utilized, constant inputs (CIs), cell count, quantum cost (QC), and area are among them. For better reversible circuits, these performance factors should be at a minimum. Here are the definitions of these factors (A & P, 2018, 2019).

Quantum cost

It expresses the cost of the circuit in terms of a simple gate. The calculation is counting the number of 1 × 1 or 2 × 2 fundamental reversible logic gates required to build the circuit. Reduced QC guarantees reduced computational overhead and improved feasibility on quantum devices.

Garbage outputs

GO is the term for the reversible gate's unneeded output. GO is the term for the reversible gate's output that isn't utilized as a primary output or as an input to other gates. The circuit needs GOs to preserve the result of reversibility. More GOs lead to increased output line usage and affect circuit compactness, which can also increase decoherence in quantum circuits.

Constant input

Some inputs are maintained at either zero or one to get a particular output. We refer to these inputs as CIs. A high number of CIs affects scalability and efficiency because it raises hardware overhead and may necessitate more qubits in a quantum implementation.

Gate count

The number of reversible gates required to carry out the computation. In general, circuit complexity, area usage, and power dissipation grow with the number of gates. Minimizing the number of gates in reversible logic improves circuit efficiency by lowering latency and resource consumption.

Number of cells

It is a measure of the number of cells needed to build the desired reversible logic gates or logic circuit. An increase in the number of cells leads to unreliable data transmission and creates erroneous outputs.

Area

It is a measure of how much area (2D) has been occupied by the constructed reversible logic gates or logic circuits. Efficient placement of QCA cells and arrangement minimize area overhead.

LITERATURE SURVEY

Chudasama and Sasamal (2016) used a carry-save adder to build a 4 × 4 Vedic multiplier based on QCA. Misra and Bhoi (2018) presented a brand-new, precise technique for synthesizing and optimizing the Baugh-Wooley multiplier. Chu et al. (2020) presented a BCD adder with majority and XOR gates, and subsequent studies suggested decimal complete adders using various devices (Abedi & Jaberipur, 2017;

Cocorullo et al., 2016; Zhang et al., 2018). Sekar et al. (2021) presented a high-speed serial-parallel multiplier and an array multiplier with a comparatively high latency. This is an effective hardware circuit that has been applied to a variety of tasks, including intricate cryptography systems, filters, and basic arithmetic circuits.

Lau and Ruslan (2022) designed and implemented a low-power 4-bit reversible multiplier. The speed improvement and power consumption of a 4-bit reversible multiplier have been examined in this work. The Peres gate serves as the reversible logic gate in the multiplier, which is based on the Wallace design (Lau & Ruslan, 2022). A reversible logic-based inexact signed Wallace tree multiplier design was created by Raveendran et al. (2021). Comparing this work's 4:2 inexact compressor to the reversible logic-based realization of current state-of-the-art designs, it has the least amount of reversible logic realization metrics. By calculating the structural similarity index measure, which is determined to be 0.96 and 0.84 for image decomposition and smoothing, respectively, the suggested multiplier's effectiveness in image processing applications is calculated (Raveendran et al., 2021).

Rashno et al. (2020) devised a reversible Vedic multiplier with minimal power usage. 2 × 2 reversible multiplier of 4 numbers of Vedic blocks were created for this piece, and each was utilized appropriately. Afterward, a 4 × 4 Vedic reversible multiplier utilizing the four previously described multipliers. Comparing the design to the previous one, the results are superior regarding GOs, number of CIs, and QC. We were able to expand the concept to n × n multipliers, allowing us to create our suggested design in all dimensions. Additionally, a formula was developed to determine the QC of our suggested n × n reversible Vedic multiplier, enabling the computation of the QC before multiplier design (Rashno et al., 2020).

A Vedic reversible multiplier was proposed by Ariafar and Mosleh (2019). Two 4-bit reversible multipliers employing the Vedic method were given in this article. The Vedic method produces partial products and their sums concurrently in a parallel fashion, which speeds up the multiplication process. The circuit evaluation criteria, which include hardware complexity, QC, gate count (GC), number of CIs, and number of GOs, are compared among the proposed designs (Ariafar & Mosleh, 2019).

An effective reversible Wallace unsigned multiplier was designed by Pour Ali Akbar and Mosleh (2019). In this work, the depth of circuits is reduced using the Wallace tree method on two 4 × 4 reversible unsigned multiplier circuits. The partial products circuit in the first design is created utilizing TG and FG gates, with TG being utilized for partial product production and FG for fan-out. In the second architecture, TG and PG gates generated partial products; fan-out was not necessary. According to the evaluation results, the first design has the smallest delay of any circuit. Furthermore, compared to other designs, the second design performs better in terms of GOs, QC, and the number of CIs (Pour Ali Akbar & Mosleh, 2019).

In QCA, Chudasama et al. (2018) created a Vedic multiplier by utilizing a ripple carry adder. An 8-bit multiplier is built using the effective structure of a 4-bit Vedic multiplier. Moreover, ripple carries adders and full adders (FAs) are used to realize the additions of the created partial products. Additionally, the difficulty of the N × N multiplier design and a generalized structure of the N × N Vedic multiplier were designed (Chudasama et al., 2018). A minimum complexity reversible multiplier was proposed by Moshnyaga (2015). The implementation of the reversible multiplier with low complexity was defined by the proposed 4 × 4 unsigned digital multiplier in RL. It is noted that the multiplier design based on TG has less circuit complexity than current implementations of 4 × 4 unsigned multipliers (Moshnyaga, 2015).

From the above literature survey, it is observed that the reversible logic multipliers are popularly adapted for various applications, including DNN. The reversible multipliers have been realized at the RTL level and/or in quantum circuits. The RTL-level implementation of reversible circuits will not satisfy Laundauer's principle; hence direction of implementation of reversible logic must be on the quantum circuit. In this research work, four types of multipliers have been considered for QCA realization using the KMD gates-based circuit design. The performance characteristics are observed and discussed to validate the design at the end.

METHODOLOGY

Functional Description of Multipliers

Here, four multipliers were considered to be designed using reversible logic. They are minimum complexity multiplier, Baugh-Wooley multiplier, Wallace multiplier, and Vedic multiplier. The overall flowchart towards design of the above multipliers using the KMD gates is shown in **Figure 3**. The design process for a validated reversible multiplier begins with selecting an appropriate multiplier type based on design requirements. A functional analysis is then performed to understand partial product generation and to design the addition structure using FAs and half adders (HAs). The conventional hardware elements are mapped to reversible logic using KMD gates, where KMD gate 2 implements AND functionality, KMD gate 3 serves as an HA, and KMD gate 5 functions as an FA. The multiplier circuit is constructed by combining reversible AND gates for partial product generation and using reversible full/half stages for summation. The design is simulated in QCADesigner 2.0.3 to verify functionality. Performance is evaluated using key metrics such as GC, GOs, CIs, area (in μm^2), and QC. Finally, the proposed reversible multiplier design is compared with existing architectures to confirm its efficiency and validity.

Minimum Complexity Multiplier

There are two stages to the multiplication process. The generation of incomplete products occurs in the first stage. The generated partial products are combined to create the final product in the second phase. **Figure 4** illustrates how this 4 × 4 unsigned multiplier was created to lower circuit complexity. In terms of the quantity of gates, GOs, CIs, and QCs, this multiplier circuit has the least amount of complexity (Moshnyaga, 2015). The HNG gate and Peres have been used in the design of the minimal complexity multiplier. It offers GO of 40, CI of 40, number of gates of 36, and QC of 140 (Moshnyaga, 2015). Reversible gates with higher efficiency can be used to decrease the performance of these multipliers.

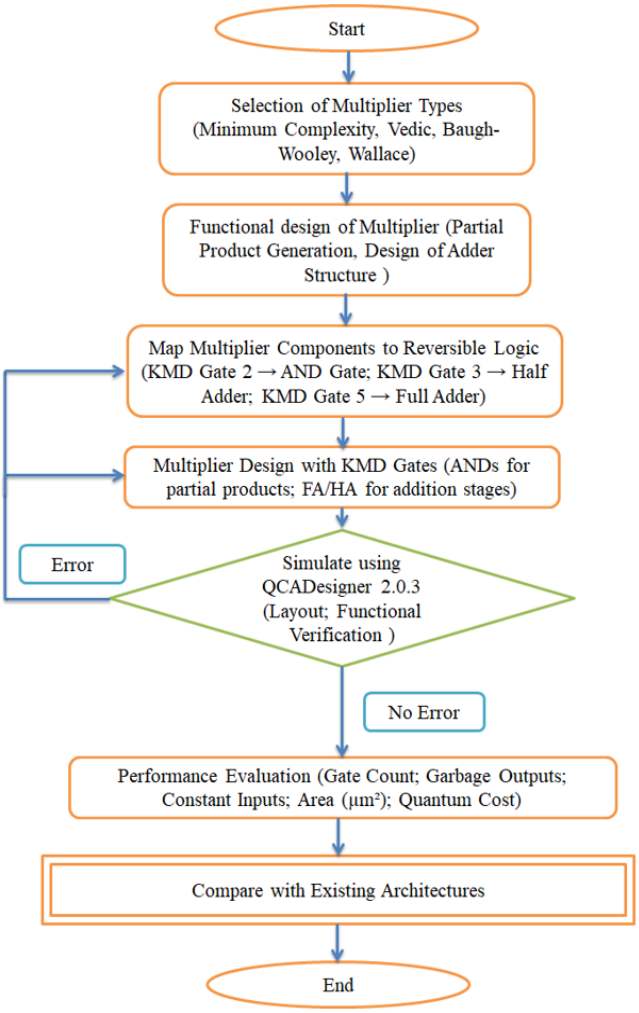


Figure 3. Methodology flowchart for the multiplier design process (Source: Authors' own elaboration)

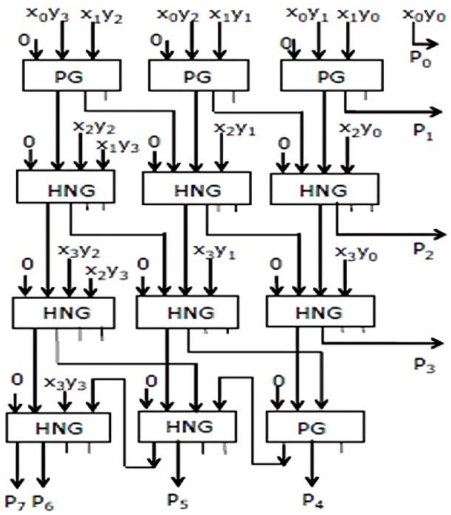


Figure 4. Architecture of minimum complexity multiplication method (Source: Authors' own elaboration)

Vedic Multiplier

An old Indian method of calculation is called Vedic mathematics. These techniques can be used for 2 × 2, 4 × 4, and N×N multipliers. This multiplier uses a multiplication algorithm for 4 × 4 multipliers. In this multiplier, a parallel partial product is generated by using Urdhva-Tiryagbhyam as in **Figure 5** and **Figure 6**.

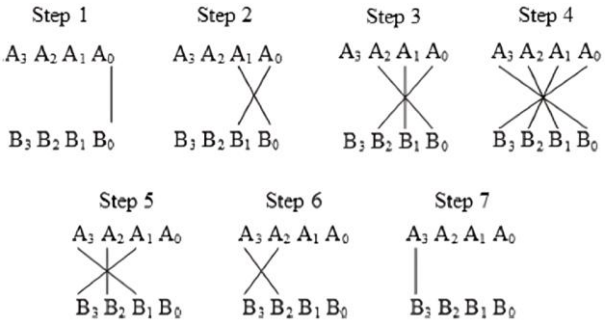


Figure 5. Partial product generation method of Vedic multiplier (Source: Authors' own elaboration)

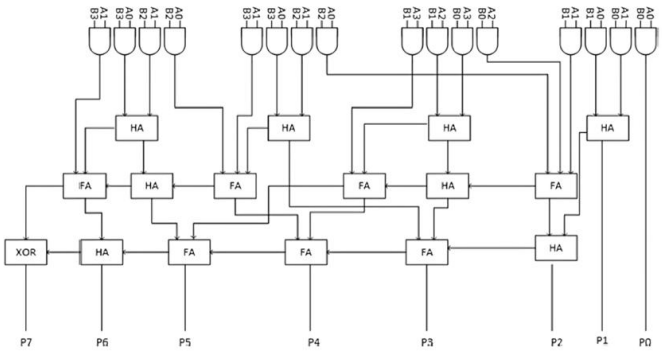


Figure 6. Architecture of Vedic multiplier structure (Source: Authors' own elaboration)

$$
\begin{array}{rcccc}
 & a_3 & a_2 & a_1 & a_0 \\
\text{x} & b_3 & b_2 & b_1 & b_0 \\
\hline
1 & \overline{a_3b_0} & a_2b_0 & a_1b_0 & a_0b_0 \\
 & \overline{a_3b_1} & a_2b_1 & a_1b_1 & a_0b_1 \\
 & \overline{a_3b_2} & a_2b_2 & a_1b_2 & a_0b_2 \\
1 & a_3b_3 & \overline{a_2b_3} & \overline{a_1b_3} & \overline{a_0b_3} \\
\hline
P_7 & P_6 & P_5 & P_4 & P_3 \quad P_2 \quad P_1 \quad P_0
\end{array}
$$

Figure 7. Multiplication logic of Baugh-Wooley (Source: Authors' own elaboration)

The primary cause of the multiplier's delay is the addition process of partial products (A & P, 2020; Chudasama et al., 2018; Khan et al., 2021; Sharma & Chattopadhyay, 2023). An ideal structure for FA and HA is employed to shorten this delay. To increase the speed of addition, the FA and HA are arranged to minimize carry propagation latency.

Baugh-Wooley Multiplier

In the complement form of 2, the Baugh-Wooley multiplier is applied to a pair of numbers. The advantage of the Baugh-Wooley multiplier is that it can be implemented with just a set of complete adders because it only requires a few logical operations at each stage of the multiplication process (Kishore et al., 2023; Gudivada & Sudha, 2020; Sridharan & Pudi, 2015). 4 × 4 Baugh-Wooley multiplier requires 12 FAs and one inverter as indicated in **Figure 7** and **Figure 8**.

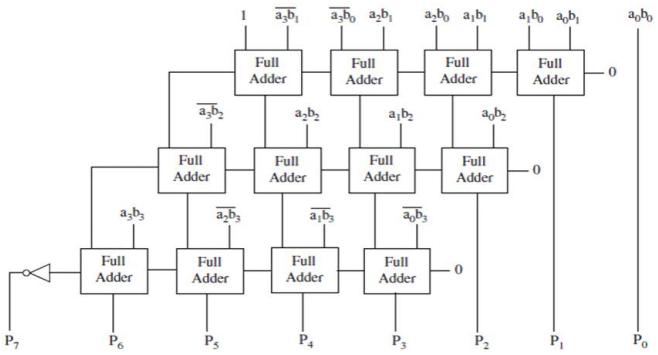


Figure 8. Architecture of Baugh-Wooley multiplier structure (Source: Authors' own elaboration)

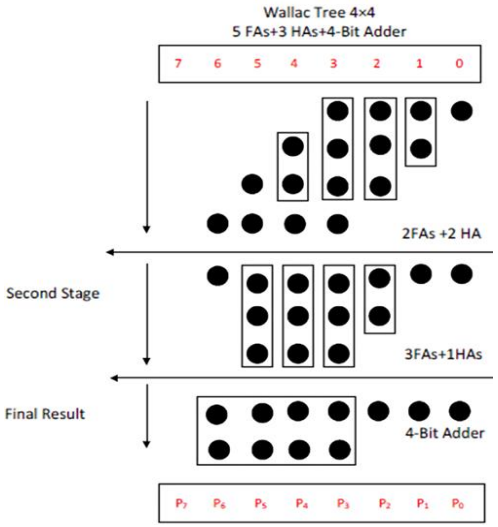


Figure 9. Architecture of Wallace multiplier (Source: Authors' own elaboration)

Figure 7 and **Figure 8** show the operation performed by the multiplier of Baugh-Wooley.

Wallace Multiplier

An electronic circuit used in digital systems to multiply two binary integers is called a multiplier. To build it, binary adders are used. A digital multiplier can be implemented using a range of computer arithmetic methods. The majority of methods entail calculating several partial products and adding them all together. Certain techniques are capable of efficiently summing up the partial products. Using the Wallace tree technique is one effective way to speed up the multiplication process (Faraji & Mosleh, 2018; Gudivada et al., 2021). This method can be used to produce hardware that can effectively conduct multiplication operations in parallel, as shown in **Figure 9**, using circuits for full and HAs that are built in three phases.

Three steps make up the Wallace-based multiplication operation:

1. To get n2 partial products, multiply (AND) each bit of the multiplicand by each bit of the multiplier.
2. Use the layers of an FA and an HA to decrease the number of partial products.
3. The second stage produced two n-sets, which were added to an n-bit adder.

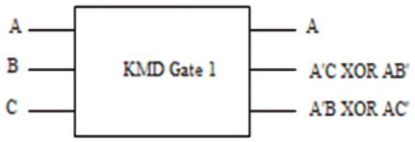


Figure 10. Block diagram of KMD gate 1 (Source: Authors' own elaboration)

Table 1. KMD gate 1 truth table

Inputs	Outputs
ABC	PQR
000	000
001	010
010	001
011	011
100	111
101	110
110	101
111	100

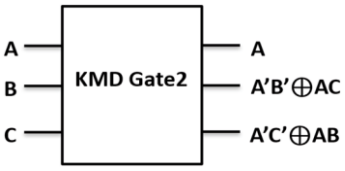


Figure 11. Block diagram of KMD gate 2 (Source: Authors' own elaboration)

Table 2. KMD gate 2 truth table

Inputs	Outputs
ABC	PQR
000	011
001	010
010	001
011	000
100	100
101	110
110	101
111	111

REQUIRED MODULES FOR MULTIPLIER DESIGN

The AND gate, HA, and FA modules are required for creating a multiplier. KMD gates could serve as the basis for these functions. The following are the KMD gates' functional descriptions.

KMD Gate 1

KMD gate 1 is the first fault-tolerant gate. This KMD gate 1 has three inputs as well as three outputs. **Figure 10** displays the KMD gate 1, and **Table 1** provides the related truth table.

KMD Gate 2

KMD gate 2 is equipped with three outputs and three inputs. It can alter this gate to an AND gate for a multiplier. **Figure 11** displays the KMD gate 2, and **Table 2** provides the related truth table.

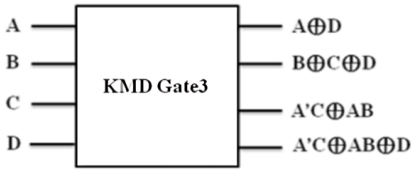


Figure 12. Block diagram of KMD gate 3 (Source: Authors' own elaboration)

Table 3. KMD gate 3 truth table

Inputs	Outputs
ABCD	PQRS
0000	0000
0001	1101
0010	0111
0011	1010
0100	0100
0101	1001
0110	0011
0111	1110
1000	1000
1001	0101
1010	1100
1011	0001
1100	1111
1101	0010
1110	1011
1111	0110

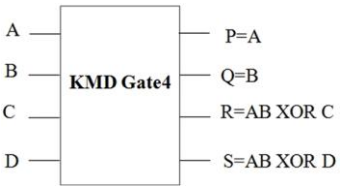


Figure 13. Block diagram of KMD gate 4 (Source: Authors' own elaboration)

Table 4. KMD gate 4 truth table

Inputs	Outputs
ABCD	PQRS
0000	0000
0001	0001
0010	0010
0011	0011
0100	0100
0101	0101
0110	0110
0111	0111
1000	1000
1001	1001
1010	1010
1011	1011
1100	1111
1101	1110
1110	1101
1111	1100

KMD Gate 3

KMD gate 3 has four inputs as well as four outputs. It is possible to transform this gate into an HA for a multiplier. **Figure 12** depicts the KMD gate 3, and **Table 3** provides the appropriate truth table.

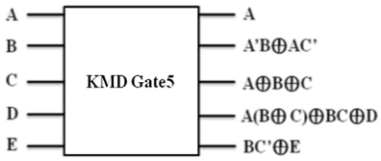


Figure 14. Block diagram of KMD gate 5 (Source: Authors' own elaboration)

Table 5. KMD gate 5 truth table

Inputs	Outputs
ABCD	PQRS
0000	0000
0001	0001
0010	0010
0011	0011
0100	0100
0101	0101
0110	0110
0111	0111
1000	1100
1001	1101
1010	1110
1011	1111
1100	1010
1101	1011
1110	1000
1111	1001

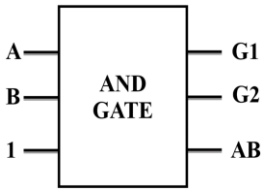


Figure 15. AND gate derived from KMD gate 2 (Source: Authors' own elaboration)

KMD Gate 4

KMD gate 4 has four inputs as well as four outputs. By making c = d = 0,the AND operation from this gate can be obtained. **Figure 13** depicts the KMD gate 4, and **Table 4** provides the related truth table.

KMD Gate 5

KMD gate 5 has five inputs as well as five outputs. This gate can be converted into an FA for a multiplier. **Figure 14** displays the KMD gate 5, and **Table 5** provides the appropriate truth table.

AND Gate

Figure 15 illustrates how to design an AND gate by configuring the third input of KMD gate 2 to set logic 1. The remaining outputs are called garbage outputs (G).

In that way, much functionality in a single gate can be constructed. **Figure 15** shows the schematic of an AND gate using KMD gate 2.

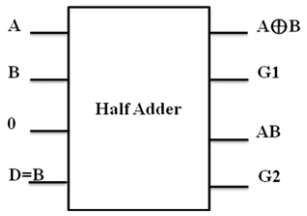


Figure 16. HA derived from KMD gate 3 (Source: Authors' own elaboration)

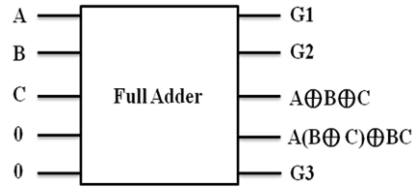


Figure 17. FA derived from KMD gate 5 (Source: Authors' own elaboration)

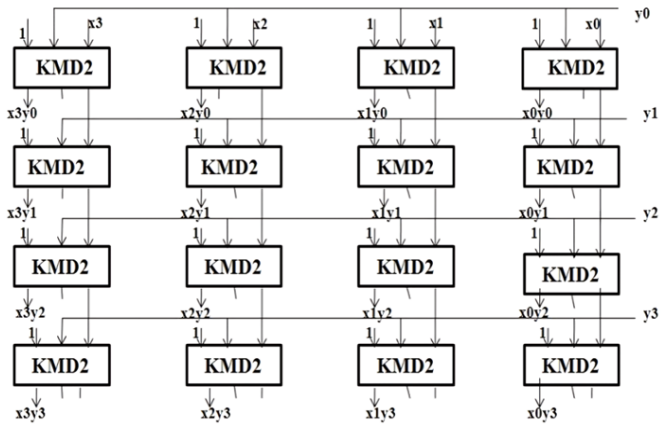


Figure 18. Proposed reversible fault tolerance minimal intricacy circuit for partial product generation using multiplier (Source: Authors' own elaboration)

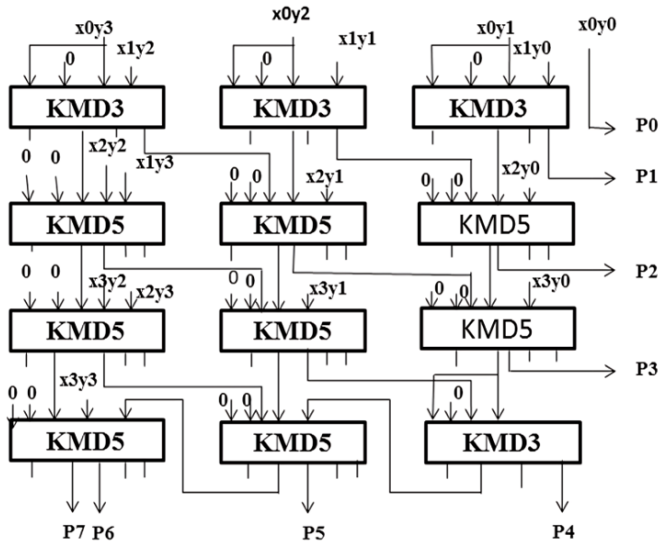


Figure 19. A reversible least complexity multiplier-addition array that is fault-tolerant is proposed (Source: Authors' own elaboration)

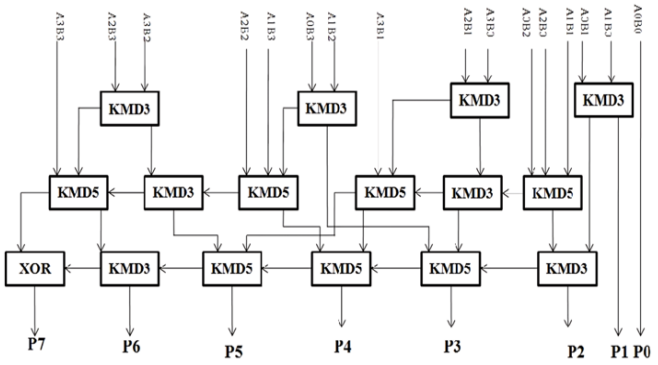


Figure 20. Fault-tolerant Vedic reversible multiplier (Source: Authors' own elaboration)

Half Adder

A half-adder can be built by placing logic 0 in the third input of KMD gate 3 and designating the fourth input as the second input. Garbage outputs (G) are the remaining outputs. In that way, much functionality in the single gate can be constructed. **Figure 16** shows the schematic of HA using KMD gate 3.

Full Adder

An FA can be built by setting logic 0 in KMD gate 5's fourth and fifth inputs. Garbage outputs (G) are the remaining outputs. In this manner, a single gate can be constructed to have different capabilities. The schematic of an FA utilizing a KMD gate 5 is shown in **Figure 17**.

PROPOSED MULTIPLIERS DESIGN

Tolerant of Faults Reversible Minimal Multiplier of Complexity

Reversible fault-tolerant KMD gates are used in the construction of this fault-tolerant reversible multiplier. This structure consists of a partial product and an addition array structure. The partial product section consists of 16 KMD2 gates, i.e., AND gates. Also, each KMD2 gate needs a CI of 1 for the AND operation. In this multiplier total of sixteen steps are needed to compute the product terms. **Figure 18** depicts the circuit for the fault-tolerant reversible partial product production.

The addition circuit in this multiplier is made up of FA and HA blocks. The KMD3 gate serves as the basis for the half-adder function. The half-adder's sum and carry can be obtained if c = 0 and b and d have the same value (b = d). The KMD5 gate is utilized for the FA. The FA with sum and carry can be obtained in this gate when the inputs d and e are the same (d = e = 0). Additionally, **Figure 19** displays the fault-tolerant reversible addition array block diagram.

Reversible Fault-Tolerant Vedic Multiplier

The reversible fault-tolerant three KMD gates are used in the design of the Vedic multiplier. Instead of sixteen steps, the multiplication procedure of this multiplier consists of just seven steps. KMD gates are used to compute the addition process. In this case, the KMD5 and KMD3 are the full and HAs, respectively. Additionally, **Figure 20** displays the fault-tolerant reversible Vedic multiplier block diagram.

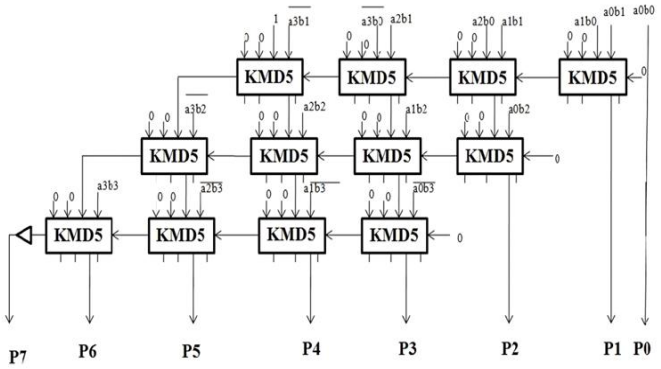


Figure 21. Proposed fault tolerant reversible Baugh-Wooley multiplier (Source: Authors' own elaboration)

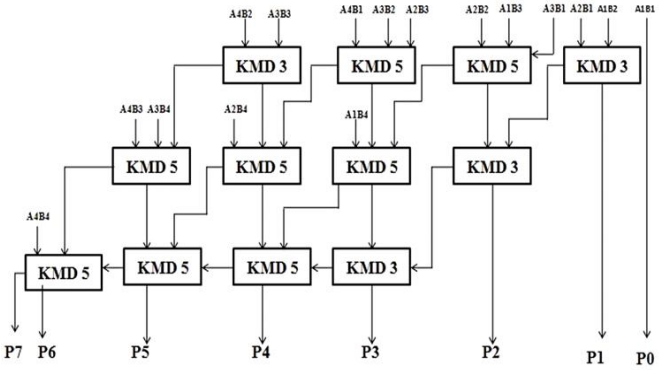


Figure 22. Proposed fault tolerant reversible Wallace multiplier (Source: Authors' own elaboration)

Reversible Fault-Tolerant the Baugh-Wooley Multiplier

The Baugh-Wooley multiplier computes the multiplication in sixteen steps in the same way as used in the minimum complexity multiplier. The addition process includes only FAs. One NOT gate is used for computing the most significant bit in the addition process. **Figure 21** depicts the fault-tolerant reversible Beck-Wooley multiplier.

Reversible Fault-Tolerant Wallace Multiplier

In this multiplier, the total multiplication and addition process is completed within three steps. This procedure requires FA and HA blocks. **Figure 22** depicts the fault-tolerant reversible Wallace multiplier construction.

RESULTS AND DISCUSSION

Using the QCADesigner version 2.0.3, every basic reversible logic gate's design was confirmed. The following default QCADesigner tool parameters have been utilized in the bistable approximation:

//Simulation engine setup in QCA//

number of samples (12,800)

convergence tolerance (0.001000)

radius of effect (65.00 nm)

relative permittivity (12.900000)

clock high(9.800000e-022)

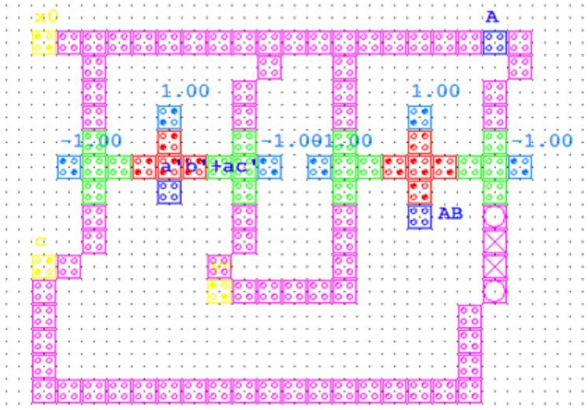


Figure 23. QCA layout of AND gate (of **Figure 14**) (Source: Authors' own elaboration)

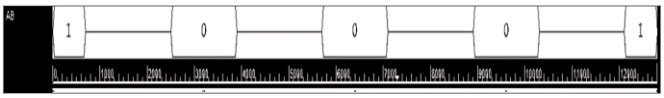


Figure 24. Simulation result of AND gate (Source: Authors' own elaboration)

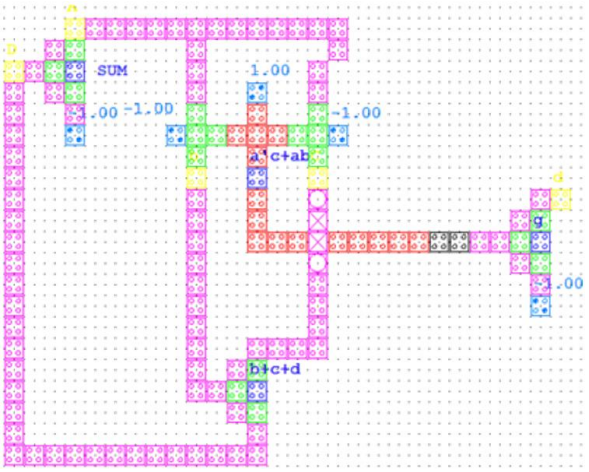


Figure 25. QCA layout of HA (of **Figure 15**) (Source: Authors' own elaboration)

clock low (3.800000e-023)

clock shift(0.000000e+000)

clock amplitude factor (2.000000)

layer separation (11.500000)

maximum iterations per sample (100).

Design of Multiplier Modules

AND gate using KMD gate 2

The AND gate and corresponding simulation result are shown in **Figure 23** and **Figure 24**, respectively. AND gate is obtained from KMD gate 2 by making c = 1.

Half adder using KMD gate 3

The HA structure is obtained from KMD gate 3 by setting the input c = 0 and b = d. This structure and the corresponding simulation result are shown in **Figure 25** and **Figure 26**, respectively.

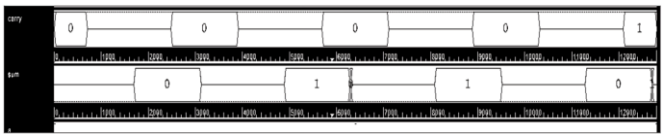


Figure 26. Simulation result of HA (Source: Authors' own elaboration)

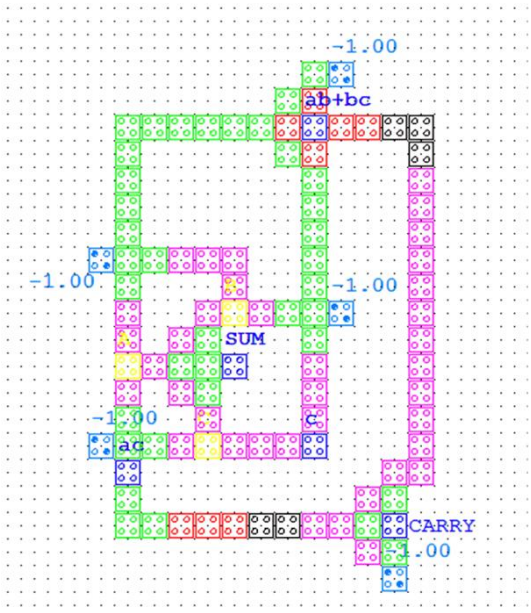


Figure 27. QCA layout of FA (of **Figure 16**) (Source: Authors' own elaboration)

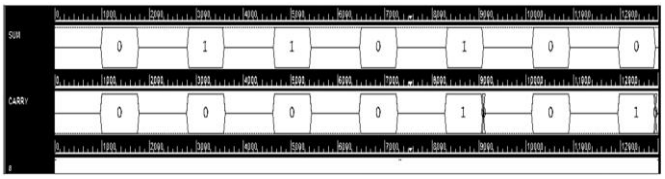


Figure 28. Simulation result of FA (Source: Authors' own elaboration)

Full adder using KMD gate 5

The FA structure is obtained from KMD gate 5 by setting the input d = e = 0. This structure and the corresponding simulation result are shown in **Figure 27** and **Figure 28**, respectively.

QCA Realization of Multipliers

Fault-tolerant reversible minimum complexity multiplier

Figure 29 and **Figure 30** display the layout and simulation result for the fault-tolerant reversible minimal complexity multiplier. The $X_3X_2X_1X_0$ and $Y_3Y_2Y_1Y_0$ are the inputs. The $P_6P_5P_4P_3P_2P_1P_0$ is the output. The inputs are given as 1010 (decimal equivalent of 10) and 1011 (decimal equivalent of 11) in the first simulation, which produces the output as 01101110 (decimal equivalent of 110).

Observations: **Table 6** shows the comparison of the existing multiplier with the proposed multiplier. The designs of a 4 × 4 bits minimum complexity multiplier have been designed on the QCADesigner tool. GC, GO, area, QC, and CI

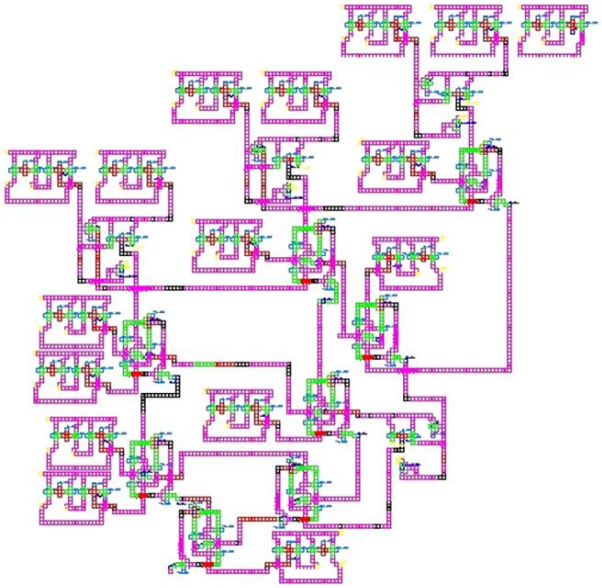


Figure 29. QCA layout of fault-tolerant reversible minimum complexity multiplier (Source: Authors' own elaboration)

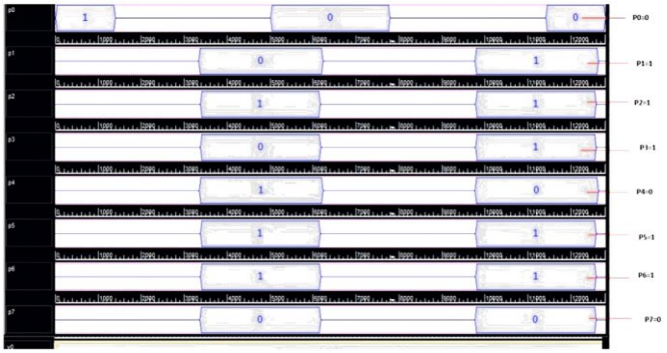


Figure 30. Simulation result of fault-tolerant reversible minimum complexity multiplier (Source: Authors' own elaboration)

Table 6. Comparison of proposed reversible fault-tolerant multiplier with existing minimum complexity multiplier

Parameter	Existing multiplier (Moshnyaga, 2015)	Proposed multiplier	Percentage of improvement
GC	36	28	22.22
CI	40	36	10.00
GO	40	38	5.00
Area (μm^2)	3	2.8	6.67
QC	140	128	8.57

are the parameters considered for comparison. The minimum complexity multiplier has an improvement of 22.22% in GC, 10% in CI, 5% in GO, 6.67% in area, and 8.57% in QC.

Fault-tolerant reversible Vedic multiplier

Layout and simulation results for the fault-tolerant reversible Vedic multiplier are shown in **Figure 31** and **Figure 32**. The $X_3X_2X_1X_0$ and $Y_3Y_2Y_1Y_0$ are the inputs. The $P_6P_5P_4P_3P_2P_1P_0$ is the output. The inputs are given as 1010 (decimal equivalent of 10) and 1011 (decimal equivalent of 11) in the first simulation, which produces the output as 01101110 (decimal equivalent of 110).

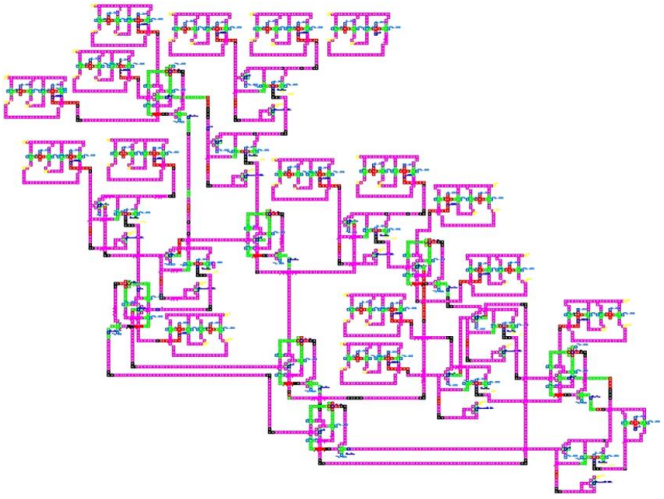


Figure 31. QCA layout of fault-tolerant reversible Vedic multiplier (Source: Authors' own elaboration)

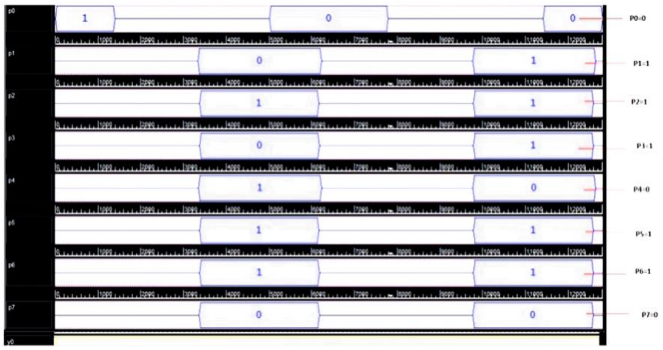


Figure 32. Simulation result of fault-tolerant reversible Vedic multiplier (Source: Authors' own elaboration)

Table 7. Comparison of proposed reversible fault-tolerant multiplier with existing Vedic multiplier

Parameter	Existing multiplier (Chudasama et al., 2018)	Proposed multiplier	Percentage of improvement
GC	80	32	60.00
CI	24	23	4.17
GO	60	57	5.00
Area (μm^2)	4.09	3.70	9.54
QC	321	301	6.23

Observations: Table 7 shows the comparison of the existing multiplier with the proposed multiplier. The 4 × 4 bits Vedic multiplier has been designed on the QCADesigner tool. The minimum complexity multiplier has an improvement of 60% in GC, 4.17% in CI, 5% in GO, 9.54% in area, and 6.23% in QC.

Fault-tolerant reversible Baugh-Wooley multiplier

The layout and simulation results for the fault-tolerant reversible Baugh-Wooley multiplier are shown in **Figure 33** and **Figure 34**. The $X_3X_2X_1X_0$ and $yY_3yY_2Y_1Y_0$ are the inputs. The $P_6P_5P_4P_3P_2P_1P_0$ is the output. The inputs are given as 1010 (decimal equivalent of 10) and 1011 (decimal equivalent of 11) in the first simulation, which produces the output as 01101110 (decimal equivalent of 110).

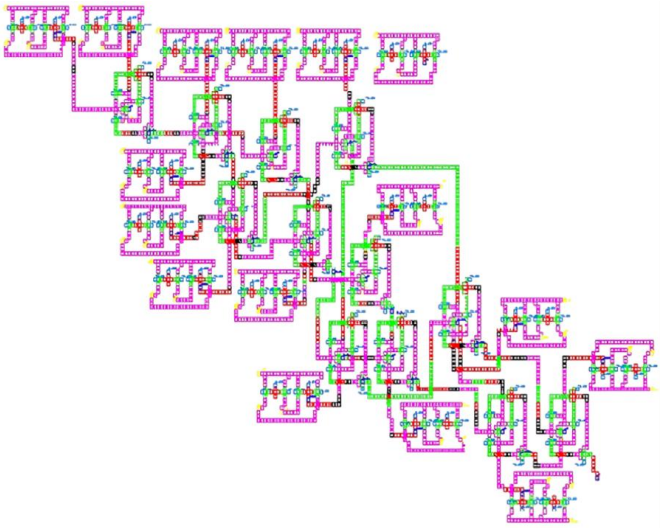


Figure 33. QCA layout of fault-tolerant reversible Baugh-Wooley multiplier (Source: Authors' own elaboration)

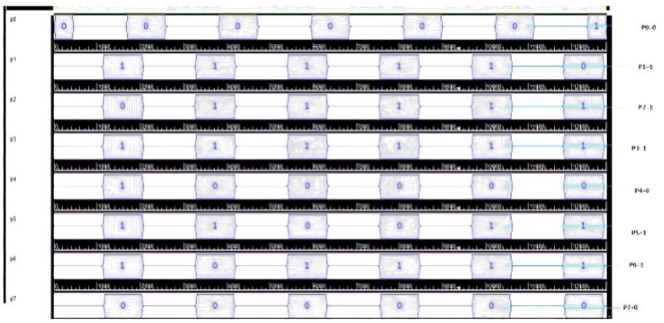


Figure 34. Simulation result of fault-tolerant reversible Baugh-Wooley multiplier (Source: Authors' own elaboration)

Table 8. Comparison of proposed reversible fault-tolerant multiplier with existing Baugh-Wooley multiplier

Parameter	Existing multiplier (Sridharan & Pudi, 2015)	Proposed multiplier	Percentage of improvement
GC	52	28	46.15
CI	36	36	-
GO	54	52	3.70
Area (μm^2)	1.80	1.75	2.78
QC	337	328	2.67

Observations: Table 8 shows the comparison of the existing multiplier with the proposed multiplier. The designs of the 4×4 bits Baugh-Wooley multiplier have been designed on the QCADesigner tool. The Baugh-Wooley multiplier has an improvement of 46.15% in GC, 3.7% in GO, 2.78% in area, and 2.67% in QC.

Fault-tolerant reversible Wallace multiplier

The layout and simulation results for the fault-tolerant reversible Wallace multiplier are shown in **Figure 35** and **Figure 36**. The $X_3X_2X_1X_0$ and $Y_3Y_2Y_1Y_0$ are the inputs. The $P_6P_5P_4P_3P_2P_1P_0$ is the output. The inputs are given as 1010 (decimal equivalent of 10) and 1011 (decimal equivalent of 11) in the first simulation, which produces the output as 01101110 (decimal equivalent of 110).

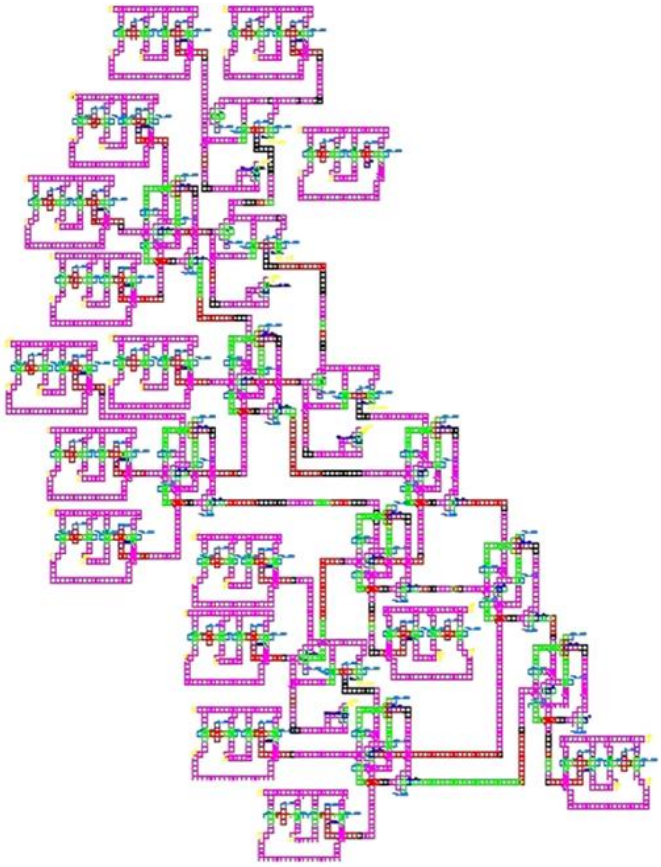


Figure 35. QCA layout of fault-tolerant reversible Wallace multiplier (Source: Authors' own elaboration)

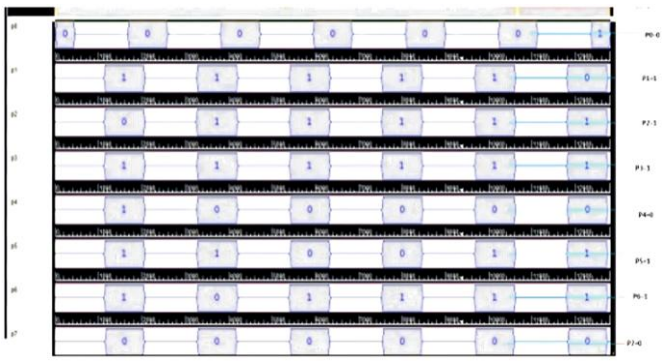


Figure 36. Simulation result of fault-tolerant reversible Wallace multiplier (Source: Authors' own elaboration)

Observations: Table 9 shows the comparison of the existing multiplier with the proposed multiplier. The design of the 4 × 4 bits Wallace multiplier has been designed on the QCADesigner tool. The Wallace multiplier has an improvement of 22.22% in GC, 7.14% in CI, 3.7% in GO, 8.13% in area, and 5.59% in QC.

From Table 10, it is observed that the minimum complexity multiplier has the lowest QC; the Vedic multiplier consumes less area; and the remaining parameters are almost similar to other multipliers.

Table 11 compares the suggested multipliers with the current multipliers based on the following parameters: cell count, area (μm^2), delay, area-delay product, and kind of interconnection layer. It has been observed that the existing literature has proposed multipliers for 2-bit or 3-bit computation. In the proposed method, all the multipliers have been designed and realized in a 4-bit computation. But, still, the proposed multipliers have competing performance advantage over the literature (Hanninen & Takala, 2007; Lu et al., 2010). Hence, the proposed multipliers have better performance than the existing multiplier designs in QCA.

Table 9. Comparison of proposed reversible fault-tolerant multiplier with existing Wallace multiplier

Parameter	Existing multiplier (Faraji & Mosleh, 2018)	Proposed multiplier	Percentage of improvement
GC	36	28	22.22
CI	28	26	7.14
GO	54	52	3.70
Area (μm^2)	3.69	3.39	8.13
QC	358	338	5.59

Table 10. Performance measure of the proposed multipliers

Parameter	Minimum complexity	Vedic	Baugh-Wooley	Wallace
GC	28	32	28	28
CI	36	23	36	26
GO	38	57	52	52
Area (μm^2)	2.80	3.70	1.75	3.39
QC	128	301	328	338

CONCLUSION

Reversible multipliers are widely used in the construction of complex circuits such as quantum computers, digital signal processing (DSP) applications, and higher-order multipliers. For the purpose of displaying binary data–0 and 1–on a QCA cell, basic Coulomb's law and the polarizing effect are employed in QCA designs. Researchers have been attempting to use QCA technology to implement traditional digital circuits ever since it was introduced. This comprises combinational circuits like the HA, FA, and multiplexer, as well as fundamental logic gates like NOT, AND, OR, XOR, and XNOR. Now, this research has been extended towards the design of multipliers on the QCA platform. In this research work, minimum complexity multiplier, Vedic multiplier, Baugh-Wooley, and Wallace multipliers are considered for the design and QCA realization. These multipliers are constructed using reversible logic and reversible gates. The KMD gates are utilized towards constructing these multipliers, and uniformity has been maintained throughout the design for the construction, i.e., only KMD gates are utilized for the construction. The functional verification has been carried on in the QCA platform. The designed multipliers are compared with the existing designs in view of QC, number of CIs, GO, cell count, area (μm^2), delay, area-delay product, and interconnection layer type. It has been observed that the proposed multipliers are performing better than the existing ones. In addition to that, a few of the existing designs are of 2-bit in nature, but still, the proposed multipliers have competing performance outcomes when compared with them. Hence, the proposed multipliers can be incorporated for DSP based applications to perform various complex computations.

Table 11. Performance comparison of the proposed multipliers

Year	Design	Cell count	Area (μm^2)	Delay	Area-delay product	Interconnection layer
2003	2-bit serial multiplier (Walus et al., 2003)	306	0.48	8.00	3.8400	Coplanar
2010	2 × 2 matrix multiplier (Lu et al., 2010)	7,102	15.69	20.00	313.8000	Multilayer
2007	2-bit multiplier (Hanninen & Takala, 2007)	1,598	1.76	7.00	12.3200	Coplanar
2023	3 × 3 array multiplier (Yan et al., 2023)	1,041	1.26	2.50	3.1500	Multilayer
2023	2 × 2 array multiplier (Yan et al., 2023)	439	0.49	1.75	0.8575	Multilayer
2023	4 × 4 minimum complexity	1,240	2.80	2.62	7.3360	Multilayer
	4 × 4 Vedic	1,310	3.70	2.71	10.0270	Multilayer
	4 × 4 Baugh-Wooley	1,338	1.75	1.80	3.1500	Multilayer
	4 × 4 Wallace	1,288	3.39	2.68	9.0852	Multilayer

The proposed reversible multiplier design is limited to 4 × 4 bit architectures, facing scalability challenges, increased complexity, and reliance on simulation-only validation. It lacks analysis of energy dissipation and real-world effects. Future work includes scaling to higher bit widths, exploring alternative reversible gates, implementing designs in QCA hardware, integrating into quantum-DSP applications, and employing machine learning based optimization algorithms for layout generation, aiming to improve performance, reduce GOs, and enhance practical applicability in quantum computing systems.

Author contributions: KA: conceptualization, investigation, methodology, writing – original draft; **IP:** resources, visualization; **VM:** validation, writing – review & editing; **TL:** software, writing – review & editing. All authors agreed with the results and conclusions.

Funding: No funding source is reported for this study.

Ethical statement: The authors stated that the study involved only secondary, anonymized, non-identifiable data and no human subjects were directly involved; therefore, ethics committee approval was exempted as per institutional policy and national guidelines.

AI statement: The authors stated that no AI tools are used for this manuscript.

Declaration of interest: No conflict of interest is declared by the authors.

Data sharing statement: Data supporting the findings and conclusions are available upon request from corresponding author.

REFERENCES

A, K., & P, M. (2018). Design of fault-tolerant reversible floating point division. *Journal of Microelectronics, Electronic Components and Materials, 48*(3), 161-172.

A, K., & P, M. (2019). Design of integrated reversible fault-tolerant arithmetic and logic unit. *Microprocessors and Microsystems, 69*, 16-23. https://doi.org/10.1016/j.micpro.2019.05.009

A, K., & P, M. (2020). Design of fault-tolerant reversible Vedic multiplier in quantum cellular automata. *Journal of the National Science Foundation of Sri Lanka, 47*(4), 371-384. https://doi.org/10.4038/jnsfsr.v47i4.9677

Abedi, D., & Jaberipur, G. (2017). Decimal full adders specially designed for quantum-dot cellular automata. *IEEE Transactions on Circuits and Systems II: Express Briefs, 65*(1), 106-110. https://doi.org/10.1109/TCSII.2017.2703942

Ariafar, Z., & Mosleh, M. (2019). Effective designs of reversible Vedic multiplier. *International Journal of Theoretical Physics, 58*(8), 2556-2574. https://doi.org/10.1007/s10773-019-04145-0

Bennett, C. H. (1973). Logical reversibility of computation. *IBM Journal of Research and Development, 17*(6), 525-532. https://doi.org/10.1147/rd.176.0525

Chu, Z., Li, Z., Xia, Y., Wang, L., & Liu, W. (2020). BCD adder designs based on three-input XOR and majority gates. *IEEE Transactions on Circuits and Systems II: Express Briefs, 68*(6), 1942-1946. https://doi.org/10.1109/TCSII.2020.3047393

Chudasama, A., & Sasamal, T. N. (2016). Implementation of 4 × 4 Vedic multiplier using carry save adder in quantum-dot cellular automata. In *Proceedings of the 2016 International Conference on Communication and Signal Processing* (pp. 1260-1264). IEEE. https://doi.org/10.1109/ICCSP.2016.7754355

Chudasama, A., Sasamal, T. N., & Yadav, J. (2018). An efficient design of Vedic multiplier using ripple carry adder in quantum-dot cellular automata. *Computers & Electrical Engineering, 65*, 527-542. https://doi.org/10.1016/j.compeleceng.2017.09.019

Cocorullo, G., Corsonello, P., Frustaci, F., & Perri, S. (2016). Design of efficient BCD adders in quantum-dot cellular automata. *IEEE Transactions on Circuits and Systems II: Express Briefs, 64*(5), 575-579. https://doi.org/10.1109/TCSII.2016.2580901

Faraji, H., & Mosleh, M. (2018). A fast Wallace-based parallel multiplier in quantum-dot cellular automata. *International Journal of Nano Dimension, 9*(1), 68-78.

Gudivada, A. A., & Sudha, G. F. (2020). Design of Baugh-Wooley multiplier in quantum-dot cellular automata using a novel 1-bit full adder with power dissipation analysis. *SN Applied Sciences, 2*, Article 813. https://doi.org/10.1007/s42452-020-2595-5

Gudivada, A. A., Kumar, K. J., Jajula, S. R., Siddani, D. P., Poola, P. K., Vourganti, V., & Panigrahy, A. K. (2021). Design of area-efficient high speed 4 × 4 Wallace tree multiplier using quantum-dot cellular automata. *Materials Today: Proceedings, 45*, 1514-1523. https://doi.org/10.1016/j.matpr.2020.07.677

Hanninen, I., & Takala, J. (2007). Pipelined array multiplier based on quantum-dot cellular automata. In *Proceedings of the 2007 18th European Conference on Circuit Theory and Design* (pp. 938-941). IEEE. https://doi.org/10.1109/ECCTD.2007.4529752

Khan, A., Bahar, A. N., & Arya, R. (2021). Efficient design of Vedic square calculator using quantum dot cellular automata (QCA). *IEEE Transactions on Circuits and Systems II: Express Briefs, 69*(3), 1587-1591. https://doi.org/10.1109/TCSII.2021.3107630

Kishore, P., Sirimalla, R., Sushma, K. S., & Reddy, R. S. (2023). Implementation of Braun and Baugh-Wooley multipliers using QCA. In *Proceedings of the 2023 2nd International Conference for Innovation in Technology* (pp. 1-4). IEEE. https://doi.org/10.1109/INOCON57975.2023.10101300

Landauer, R. (1961). Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development, 5*(3), 183-191. https://doi.org/10.1147/rd.53.0183

Lau, W. S., & Ruslan, S. H. (2022). The design and implementation of low-power 4-bit reversible multiplier. *Evolution in Electrical and Electronic Engineering, 3*(1), 37-43.

Lu, L., Liu, W., O'Neill, M., & Swartzlander Jr, E. E. (2010). QCA systolic matrix multiplier. In *Proceedings of the 2010 IEEE Computer Society Annual Symposium on VLSI* (pp. 149-154). IEEE. https://doi.org/10.1109/ISVLSI.2010.53

Misra, N. K., & Bhoi, B. K. (2018). Synthesis methods of Baugh-Wooley multiplier and non-restoring divider to enhance primitive's results of QCA circuits. In *Smart intelligent computing and applications. Smart innovation, systems and technologies* (pp. 237-245). Springer. https://doi.org/10.1007/978-981-13-1921-1_24

Moshnyaga, V. G. (2015). Design of minimum complexity reversible multiplier. In *Proceedings of the TENCON 2015-2015 IEEE Region 10 Conference* (pp. 1-4). IEEE. https://doi.org/10.1109/TENCON.2015.7373120

Pour Ali Akbar, E., & Mosleh, M. (2019). An efficient design for reversible Wallace unsigned multiplier. *Theoretical Computer Science, 773*, 43-52. https://doi.org/10.1016/j.tcs.2018.06.007

Rashno, M., Haghparast, M., & Mosleh, M. (2020). A new design of a low-power reversible Vedic multiplier. *International Journal of Quantum Information, 18*(3), Article 2050002. https://doi.org/10.1142/S0219749920500021

Raveendran, S., Edavoor, P. J., Kumar, Y. N., & Vasantha, M. H. (2021). Inexact signed Wallace tree multiplier design using reversible logic. *IEEE Access, 9*, 108119-108130. https://doi.org/10.1109/ACCESS.2021.3100892

Riyaz, S., & Sharma, V. K. (2023). Design of reversible Feynman and double Feynman gates in quantum-dot cellular automata nanotechnology. *Circuit World, 49*(1), 28-37. https://doi.org/10.1108/CW-08-2020-0199

Sekar, K. R., Marshal, R., & Lakshminarayanan, G. (2021). High-speed serial-parallel multiplier in quantum-dot cellular automata. *IEEE Embedded Systems Letters, 14*(1), 31-34. https://doi.org/10.1109/LES.2021.3098017

Sharma, V., & Chattopadhyay, M. K. (2023). Implementation of novel 2 × 2 Vedic multiplier using QCA technology. *Journal of Physics: Conference Series, 2603*, Article 012045. https://doi.org/10.1088/1742-6596/2603/1/012045

Sridharan, K., & Pudi, V. (2015). *Design of arithmetic circuits in quantum dot cellular automata nanotechnology*. Springer. https://doi.org/10.1007/978-3-319-16688-9

Walus, K., Jullien, G. A., & Dimitrov, V. S. (2003). Computer arithmetic structures for quantum cellular automata. In *Proceedings of the 37th Asilomar Conference on Signals, Systems & Computers* (pp. 1435-1439). IEEE. https://doi.org/10.1109/ACSSC.2003.1292223

Yan, A., Li, X., Liu, R., Huang, Z., Girard, P., & Wen, X. (2023). Designs of array multipliers with an optimized delay in quantum-dot cellular automata. *Electronics, 12*(14), Article 3189. https://doi.org/10.3390/electronics12143189

Zhang, T., Pudi, V., & Liu, W. (2018). New majority gate-based parallel BCD adder designs for quantum-dot cellular automata. *IEEE Transactions on Circuits and Systems II: Express Briefs, 66*(7), 1232-1236. https://doi.org/10.1109/TCSII.2018.2878717