Design and FPGA Implementation of Reversible Logic-Based Multiplier for Efficient Cryptographic Operations

Mihir Lal Saha, Antarik Sinha, Agnik Maity, Malay Gangopadhyaya, Siddhartha Roy, and Sayan Chatterjee

Abstract The need for efficient and low-power hardware accelerators for cryptographic algorithms has led to significant advancements in digital design. This paper presents the design and FPGA implementation of a reversible logic gates-based multiplier optimized for cryptographic operations. Utilizing Peres and HNG gates, the design aims to minimize quantum cost, gate count, garbage output, and TRLIC, enhancing overall system performance and energy efficiency. The proposed multiplier is evaluated against existing designs, focusing on metrics such as quantum cost, garbage output, ancilla input, Total Reversible Logic Instruction Count (TRLIC), number of gates used, and delay. The FPGA implementation demonstrates the practicality and effectiveness of the design, showcasing its potential for real-world applications. The findings provide valuable insights into the trade-offs and benefits of using reversible multipliers, contributing to the development of future-proof hardware solutions in the realm of cryptographic operations. A multiplier circuit that is optimized has been designed and developed using HNG and PERES gates. Using Xilinx Vivado and Xilinx Ise 14.7, the simulations are performed. The NEXYS A7 (Artix 7 series) board has been used for FPGA implementation. Ultimately, it's mentioned that as in addition to improving the delay, there is a significant 50.36% reduction in the quantum cost.

M. L. Saha (✉) · A. Sinha · A. Maity · M. Gangopadhyaya
Department of Electronics and Communication Engineering, Institute of Engineering & Management, Kolkata School of University of Engineering & Management, Kolkata, West Bengal, India
e-mail: mihirlal.saha@iem.edu.in

M. Gangopadhyaya
e-mail: malay.ganguly@iem.edu.in

S. Roy
Department of Electronics and Communication Engineering, Bengal Institute of Technology, Kolkata, West Bengal, India

S. Chatterjee
Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata, West Bengal, India
e-mail: sayan.chatterjee@jadavpuruniversity.in

N. Das et al. (eds.), *Proceedings of International Conference on Data, Electronics and Computing*, Lecture Notes in Networks and Systems 1657,
https://doi.org/10.1007/978-981-95-3163-9_18

Keywords Reversible logic gates · Cryptography · FPGA · TRLIC · Xilinx · Quantum cost · Garbage outputs

1 Introduction

The increasing demand for efficient and low-power hardware accelerators for cryptographic algorithms has driven significant advancements in digital design. Cryptographic algorithms, particularly those involving public-key type cryptography such as RSA and Elliptic Curve type Cryptography, require extensive computational resources, with multiplication being a fundamental operation. Optimizing these operations is crucial for enhancing overall system performance and energy efficiency. Reversible logic, which operates without information loss, presents a promising approach to address these challenges. Unlike conventional irreversible logic gates, reversible logic gates have the potential to significantly reduce power dissipation, making them ideal candidates for energy-efficient cryptographic modules. Additionally, reversible logic is inherently compatible with quantum computing, providing a pathway to future-proof cryptographic hardware against the capabilities of quantum technologies. This work introduces a design of a reversible multiplier, optimized for cryptographic operations, and its implementation on a FPGA. The design utilizes Peres and HNG gates to achieve efficiency and minimize computational complexity. By focusing on reducing the gate count and ensuring minimal resource usage, the design meets the stringent demands of cryptographic applications. Key contributions of this work include a detailed design of the reversible multiplier, highlighting the integration of Peres and HNG gates to enhance performance and efficiency. The implementation process on an FPGA platform is described, demonstrating the practicality and performance of the design in real-world applications. A comprehensive evaluation and comparison of the proposed work with existing designs is conducted, focusing on key metrics such as garbage output, ancilla input, Total Reversible Logic Instruction Count (TRLIC), number of gates used, and delay. This thorough evaluation highlights the advantages and trade-offs of the proposed design, providing insights into its potential applications in cryptographic systems. By integrating reversible logic with modern FPGA technology, this approach addresses current power efficiency challenges and provides a robust framework for cryptographic module implementation. The significance of minimizing garbage outputs and ancilla inputs cannot be overstated, as these factors directly impact the overall efficiency and feasibility of implementing reversible logic in practical applications. Garbage outputs, which are unwanted intermediate results, and ancilla inputs, which are additional inputs required to maintain reversibility, both contribute to increased resource usage and power consumption. By designing a reversible multiplier that minimizes these aspects, the proposed solution offers a more streamlined and efficient approach to cryptographic computations. Furthermore, the use of Peres and HNG gates in the work enhances the efficiency of the reversible logic-based multiplier. Peres gates are known for their ability to perform multiple logical operations within

a single gate, thereby reducing the overall gate count and improving computational efficiency. HNG gates, on the other hand, offer enhanced functionality by combining multiple logical operations into a single gate, which also helps in cutting down on the quantity of gates utilized and the delay. The combination of these gates in the proposed design results in a multiplier that is not only efficient, but also highly suitable for cryptographic applications where power consumption and computational speed are critical. The implementation of the reversible multiplier on an FPGA platform further underscores the practicality of the design. FPGAs are widely used in hardware acceleration for cryptographic applications due to their flexibility and ability to handle complex computations efficiently. The FPGA implementation demonstrates the feasibility of deploying the proposed reversible multiplier in real-world cryptographic systems, showcasing its potential to improve the overall performance and efficiency of such systems. This work contributes to the ongoing efforts in developing efficient and future-proof hardware solutions for cryptographic applications. By leveraging the benefits of reversible logic and integrating Peres and HNG gates. This paper lays the groundwork for further research and development in the field of reversible logic-based computing and its application in cryptographic hardware, emphasizing the significant potential of reversible logic in advancing digital design.

2 Reversible Logic Gates

See Fig. 1.

Gate Count—The number of reversible logic gates used to design the integrated circuits.

Ancilla Input—It is the constant input of the reversible logic gate which is set as either logic 1 or logic 0 depending on the required Boolean expression.

Garbage Output—It is the unused output of the reversible logic gates that are not required for the particular Boolean function. They help to retain the reversibility property of the reversible logic gates.

Quantum Cost—Every reversible logic gates have a particular quantum cost which is calculated by the number of primitive gates present, i.e., not gate, controlled-v gate, controlled-v+ gate, and cnot gate.

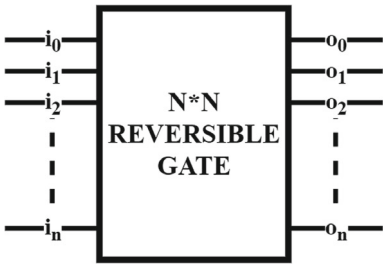

Fig. 1 Schematic of *n* * *n* reversible logic gate

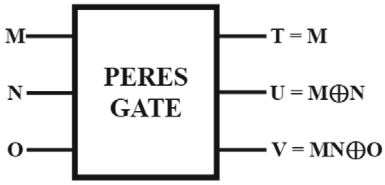

Fig. 2 Schematic of 3 * 3 Peres gate

These four factors—gate count, ancilla input, garbage output, and quantum cost—determine the performance and efficiency of a circuit designed by reversible logic gates. They act as a parameter to compare an existing design with new design. The number of inputs and outputs of reversible logic gates is equal. There is a one-to-one mapping between the inputs and outputs of the reversible logic gates. In reversible gates there are neither feedbacks nor direct fanouts. They are reversible as the inputs can be retrieved from the outputs. In reversible logic gates no information bits are lost, thus the power dissipation reduces as it overcomes the Landauer's cause of power dissipation. These make the reversible logic gates a better alternative over the conventional logic gates. Complex integrated circuits can be developed using reversible logic gates as the power dissipation is lower than conventional logic gates due to its reversibility property which prevent the loss of information bits. In the proposed 4-bit multiplier architecture, Peres and HNG gates have been used.

2.1 PERES Gate

Peres gate is a one kind of reversible logic gate with 3 inputs and 3 outputs. The Input (M, N, O) are the input vectors and Output (T, U, V) are the output vectors. The quantum cost of this 3 * 3 gate is 4. Figure 2 illustrates the PERES gate. The outputs of the Peres gate have the following Boolean expressions (Table 1):

$$\begin{aligned} T &= M \\ U &= M \oplus N \\ V &= MN \oplus O \end{aligned}$$

2.2 HNG Gate

HNG gate is a reversible logic gate with 4 inputs and 4 outputs. The Input (L, M, N, O) are the input vectors and Output (S, T, U, V) are the output vectors. Figure 3 illustrates the HNG gate. The quantum cost of this 4 * 4 gate is 6.

Table 1 Truth table of Peres gate

Inputs			Outputs		
M	N	O	T	U	V
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	0	0

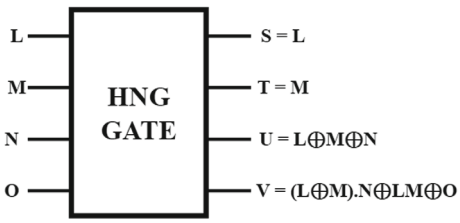

Fig. 3 Schematic of 4 * 4 HNG gate

The outputs of the HNG gate have the following Boolean expressions (Table 2):

$$\begin{aligned} S &= L \\ T &= M \\ U &= L \oplus M \oplus N \\ V &= (L \oplus M).N \oplus LM \oplus O \end{aligned}$$

3 4 × 4 Array Multiplier Circuit

3.1 Bitwise Product Block of 4 × 4 Array Multiplier

Figure 4 illustrates the bitwise product block of 4 × 4 array multiplier circuit. The bitwise product multiplier block has 5 inputs (a 4-bit number $i(i_0, i_1, i_2, i_3)$ and a bit of the other 4-bit number (j) and 4 outputs (m_0, m_1, m_2, m_3).

The function of the bitwise multiplier product block of the 4-bit multiplier is to multiply each bit of the second 4-bit number with each and every bit of the first 4-bit number.

Table 2 Truth table of HNG gate

Inputs				Outputs			
L	M	N	O	S	T	U	V
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	0	1	0
1	0	0	1	1	0	1	1
1	0	1	0	1	0	0	1
1	0	1	1	1	0	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	0	0
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	0

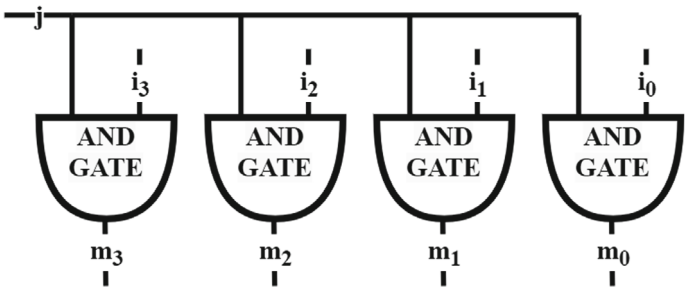

Fig. 4 Bitwise product block

This bitwise multiplier product block generates a partial product of the two 4-bit numbers. This partial product consists of a bit from each of the two numbers. In this way, the partial product of every combination of the two 4-bit number is obtained.

3.2 4-Bit Ripple Carry Adder

Figure 5 illustrates 4-bit RCA—Ripple Carry Adder; it is an essential part of a 4-bit array multiplier. The 4-bit RCA (Ripple Carry Adder circuit) block has 9 inputs which are the resulted output (two 4-bit number $i(i_0, i_1, i_2, i_3)$ and $j(j_0, j_1, j_2, j_3)$ a single bit input (k_{in})) and 5 outputs ($r_0, r_1, r_2, r_3, k_{\text{out}}$). The partial product outputs of the bitwise multiplier product block are arranged in a particular sequence which acts as an input for the RCA—Ripple Carry Add block. The block of ripple carry type adder adds the partial products calculated by the bitwise multiplier product block.

The output of the RCA (Ripple Carry Adder) block becomes the input for the next ripple carry adder block, and this continues up to the last ripple carries adder block. The output of the last ripple carry adder block gives us the resultant bits of the 4-bit multiplier. The circuit's output bits and inputs are related in the following ways:

$$
\begin{aligned}
r_0 &= i_0 \oplus j_0 \oplus k_{\text{in}} \\
r_1 &= i_1 \oplus j_1 \oplus k_1 \\
r_2 &= i_2 \oplus j_2 \oplus k_2 \\
r_3 &= i_3 \oplus j_3 \oplus k_3 \\
k_1 &= i_0 j_0 + j_0 k_{\text{in}} + k_{\text{in}} i_0 \\
k_2 &= i_1 j_1 + j_1 k_1 + k_1 i_1 \\
k_3 &= i_2 j_2 + j_2 k_2 + k_2 i_2 \\
k_{\text{out}} &= i_3 j_3 + j_3 k_3 + k_3 i_3
\end{aligned}
$$

Digital circuits that multiply two 4-bit binary integers to create an 8-bit result efficiently are called 4-bit array multipliers. The circuit is built with a matrix-like arrangement of AND gates and adders organized in a grid. The process creates a grid of partial type products by doing the multiplication process of every single bit of the multiplic and with each bit of the multiplier. After that, 4-bit ripple carry full adders are used to sum these partial products in an organized way and aggregate the results. Figure 6 illustrates the schematic of 4-bit array multiplier.

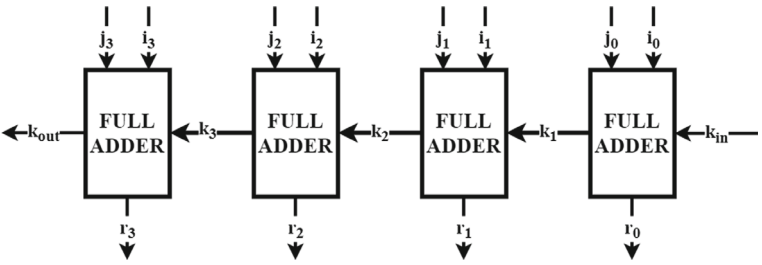

Fig. 5 Circuit diagram of 4-bit RCA (Ripple Carry Adder)

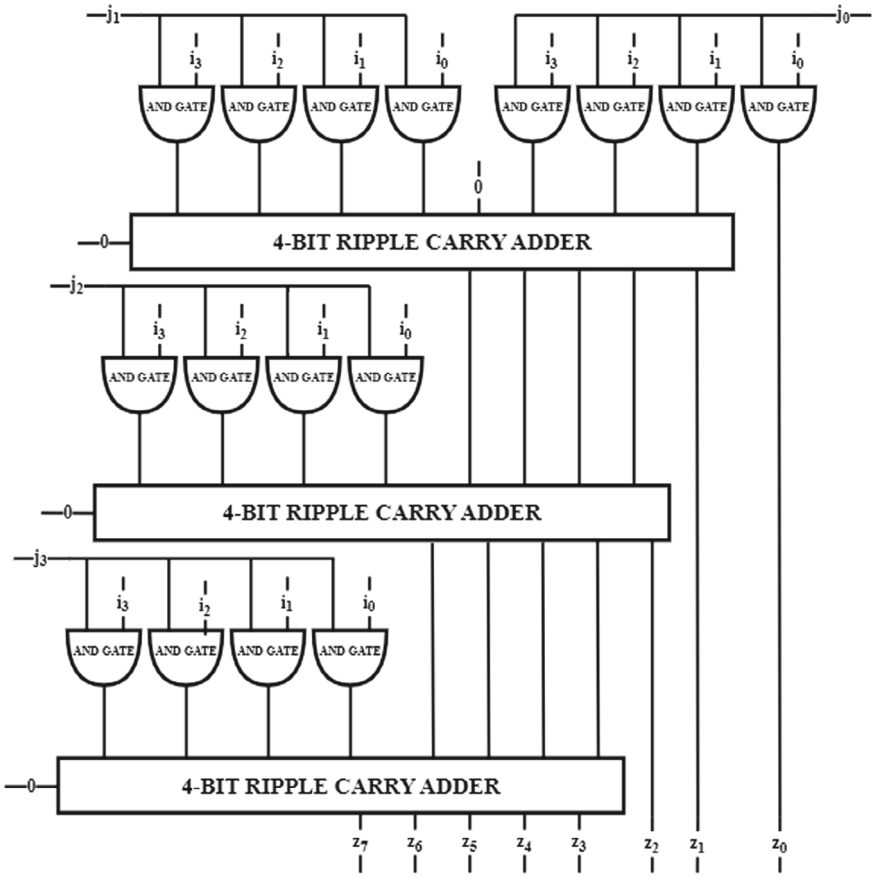

Fig. 6 Schematic of 4 × 4 array multiplier [1]

4 Proposed 4 × 4 Multiplier Using Reversible Logic

Each of the blocks of array multiplier has been replaced by an equivalent set of reversible logic gates in the proposed 4 × 4 multiplier circuit; the proposed circuit is illustrated in the following sections, where a combination of reversible gates is used in place of each individual block.

4.1 Bitwise Product Block Using Reversible Logic

A combination of four PERES gates are deployed to implement the bitwise product block of proposed multiplier circuit. The outputs of the bitwise product block are m_0, m_1, m_2, and m_3. Each processing a bit of the input (i_0–i_3) with a common input j,

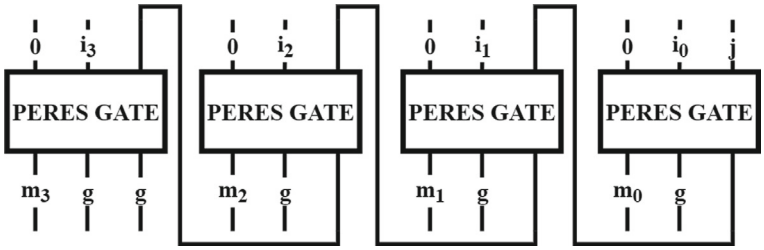

Fig. 7 Proposed bitwise product block

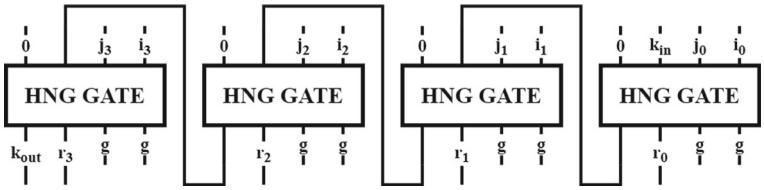

Fig. 8 Schematic of proposed 4-bit ripple carry adder using reversible HNG gates [2]

producing outputs m_0–m_3. Each Peres gate computes the XOR and AND operations, resulting in bitwise product outputs, while generating two garbage outputs. This configuration illustrates the use of reversible logic to implement a bitwise product block in a multiplier circuit, minimizing power dissipation (Fig. 7).

4.2 4-Bit Ripple Carry Adder Using Reversible Logic

The four-bit ripple carry adder is implemented using four HNG gates in total for implementing the proposed ripple carry adder. The inputs for each HNG gate include the respective bits of the numbers being added (i_0, i_1, i_2, i_3 and j_0, j_1, j_2, j_3) along with the carry inputs and outputs (k_{in}, r_0, r_1, r_2, r_3) and garbage output (g). The final carry output is labeled as k_{out}. This configuration allows for the sequential processing of each bit, from the least significant to the most significant, producing the correct sum and carry outputs for the entire four-bit binary addition (Fig. 8).

5 Simulation and Results

The designed architecture has been realized using hardware description language Verilog. The simulation and synthesis work has been done using Xilinx Vivado and Xilinx ISE. Total three types of input pattern has been taken for the simulation of the proposed work. Figure 9 illustrates the RTL of proposed design. Figures 10, 11, and

12 illustrates the outputs of test cases. The two four-bits input is represented by a, b and output is represented by p.

From Table 3, it can be observed that the proposed work has been constructed by using only 28 reversible gates which is having 32 ancilla inputs, 44 garbage outputs, and a quantum cost of 136. For the proposed design, a maximum improvement of 50.36% in quantum cost and 41.03% in TRLIC was attained compared to the other existing designs.

From Table 4, it can be observed that the delay of proposed design is better than existing design [3, 7–14]. Figure 13 illustrates the comparison of garbage output in bar graph (Table 5).

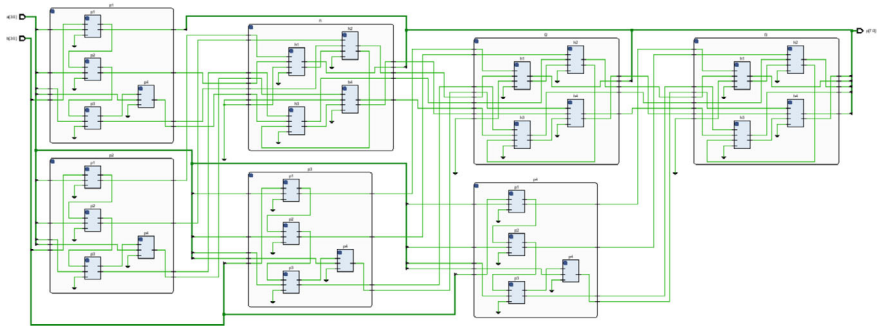

Fig. 9 RTL design of the proposed design

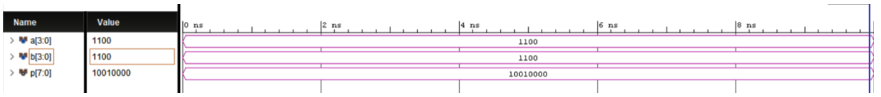

Fig. 10 Simulated output wave of proposed multiplier where $a = 1100$, $b = 1100$

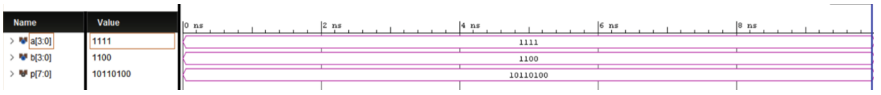

Fig. 11 Simulated output wave of proposed multiplier where $a = 1111$, $b = 1100$

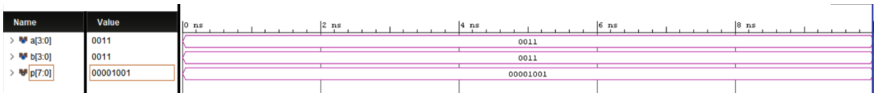

Fig. 12 Simulated output wave of proposed multiplier where $a = 0011$, $b = 0011$

Table 3 Comparison of existing design

Parameter	Gate count	Ancilla input	Garbage output	Quantum cost	% of improvement	TRLIC	% of improvement
Proposed design	28	32	44	136		240	
[3]	NR*	34	58	221	38.46%	NR*	
[4]	44	56	56	236	42.37%	392	38.77%
[5]	40	52	52	152	10.52%	296	18.91%
[6]	42	42	49	136	0%	269	10.78%
[7]	28	28	52	144	5.88%	252	4.76%
[8]	28	44	56	232	41.37%	360	33.33%
[9]	28	28	52	144	5.88%	252	4.76%
[10]	29	46	58	274	50.36%	407	41.03%
[11]	NR*	43	53	224	39.28%	NR*	

NR* Not Reported

Table 4 Comparison of delay

Parameter	Delay (ns)
Proposed design	4.561
[3]	10.422
[12] (1)	9.00
[12] (2)	10.00
[7]	9.00
[8]	11.00
[9]	11.91
[10]	11.00
[11]	11.00

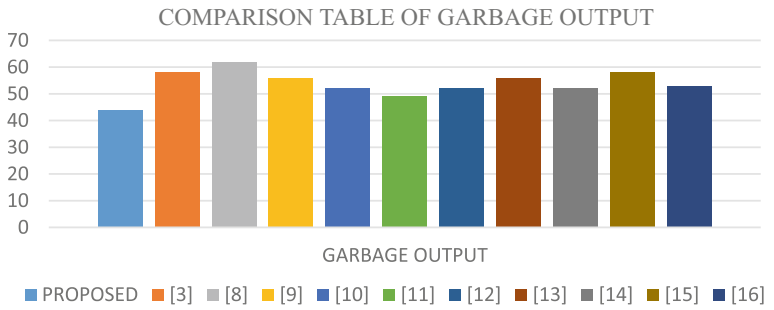

Fig. 13 Comparison of garbage output

Table 5 Comparison of power

Parameter	Static power	Dynamic power
Proposed design (mW)	0.234	11.66
[15] (W)	0.586	19.106
[16] (W)	0.081	0.00017

6 FPGA Implementation

Total number of three different types of input patterns as test case of 4 bit used for FPGA implementation of proposed multiplier. After the FPGA implementation, it's resulting different types of output from different test cases (Table 6).

NEXYS A7 is a 16-bits FPGA board. Here 8 bits of switches and led's (from right to left) used for the FPGA implementation. Figures 14, 15, and 16 illustrates the FPGA implementation. Simulated output and the output which are getting by FPGA functioning are similar. FPGA implementation of proposed multiplier has been done successfully.

Table 6 Test cases

Test case	Input		Output
1	1100	1100	10010000
2	1111	1100	10110100
3	0011	0011	00001001

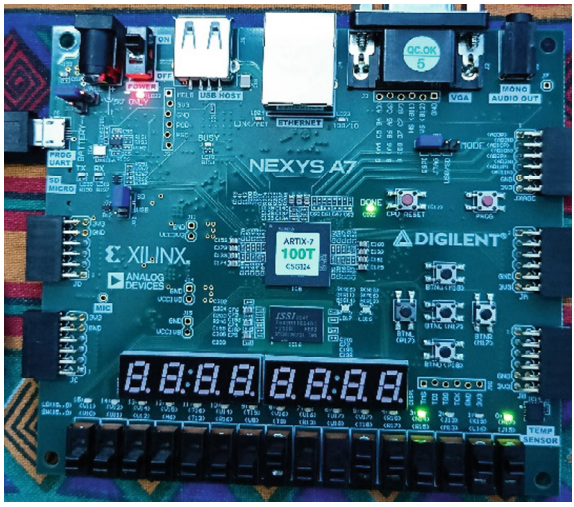

Fig. 14 FPGA implementation of test case 1

Fig. 15 FPGA implementation of test case 2

Fig. 16 FPGA implementation of test case 3

7 Conclusion

An optimized multiplier circuit using HNG and Peres gates has been designed and developed in this work, with simulations performed using Xilinx Vivado and Xilinx ISE tools. Implemented on the NEXYS A7 (Artix 7 series) FPGA board, the design demonstrates a significant 50.36% reduction in quantum cost and a 41.03% improvement in TRLIC compared to existing designs, along with reduced delay. These results highlight the effectiveness of reversible logic in creating efficient and high performance cryptographic modules, supporting future advancements in digital hardware design.

References

1. Qian J, Wang J (2014) A 4-bit array multiplier design by reversible logic
2. Halder S, Saha ML, Malvika, Talukdar J, Mummaneni K (2024) Design and implementation of an optimized high-speed vedic-based squarer circuit using reversible logic gates. In: Lecture notes in electrical engineering, vol 1067. Springer, Singapore
3. Durgam V, Ragini K, Durgam D (2021) Design of 4 × 4 reversible multiplier using reversible TSG gate. Int J Eng Res Electron Commun Eng (IJERECE) 8(3)
4. Sinha HP, Syal N (2012) Design of fault tolerant reversible multiplier
5. Bhagyalakshmi HR, Venkatesha MK (2010) An improved design of a multiplier using reversible logic gates
6. Zhou R, Shi Y, Cao J, Wang H (2010) Comment on design of a novel reversible multiplier circuit using HNG gate in nanotechnology. World Appl Sci J 10(2):161–165
7. Islam MS, Rahman MM, Begum Z, Hafiz MZ (2009) Low-cost quantum realization of reversible multiplier circuit. Inf Technol J 8:208–213. https://doi.org/10.3923/itj.2009.208.213
8. Shams M et al (2008) Novel reversible multiplier circuit in nanotechnology

9. Haghparast M et al (2008) Design of a novel reversible multiplier circuit using HNG gate in nanotechnology
10. Thapliyal H, Srinivas MB (2006) Novel reversible multiplier architecture using reversible TSG gate. In: IEEE international conference on computer systems and applications, Dubai, United Arab Emirates, pp 100–103. https://doi.org/10.1109/AICCSA.2006.205074
11. Thapliyal H et al (2005) A reversible version of 4 × 4 bit array multiplier with minimum gates and garbage outputs. Embedded Syst Appl
12. PourAliAkbar E, Mosleh M (2019) An efficient design for reversible wallace unsigned multiplier. Theor Comput Sci 773:43–52. ISSN 0304-3975. https://doi.org/10.1016/j.tcs.2018.06.007
13. Ariafar Z, Mosleh M (2019) Effective designs of reversible vedic multiplier. Int J Theor Phys 58:2556–2574. https://doi.org/10.1007/s10773-019-04145-0
14. Das A, Bha MJK (2014) Design optimization of vedic multiplier using reversible logic. Int J Eng Res Technol (IJERT) 03(03)
15. KR AJ, Sakkara S (2022) A novel, scalable N * N reversible logic multiplier design. In: 2022 3rd international conference on intelligent engineering and management (ICIEM), London, United Kingdom, pp 249–255. https://doi.org/10.1109/ICIEM54221.2022.9853070
16. Yugandhar K et al (2018) High performance array multiplier using reversible logic structure. In: 2018 international conference on current trends towards converging technologies (ICCTCT), pp 1–5