International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 3(1)(March)2026

Received: 10-1-2026 Accepted: 17-2-2026 Published: 27-3-2026

Design of Hamming Code Encoder and Decoder Using Reversible Logic

O. Homakesav
Dept. of Electronics and Communication Engineering
Annamacharya Institue of Technology and Sciences
Kadapa, India
homakesav@gmail.com

Pola Sruthi
Dept. of Electronics and Communication Engineering
Annamacharya Institue of Technology and Sciences
Kadapa, India
sruthip0222@gmail.com

T. Harshitha
Dept. of Electronics and Communication Engineering
Annamacharya Institue of Technology and Sciences
Kadapa, India
tumatiharshitha71@gmail.com

P. Sandya
Dept. of Electronics and Communication Engineering
Annamacharya Institue of Technology and Sciences
Kadapa, India
sandhyasandy5442@gmail.com

P. V. Deepak
Dept. of Electronics and Communication Engineering
Annamacharya Institue of Technology and Sciences
Kadapa, India
deepakpv9703@gmail.com

***Abstract*—In contemporary digital communication and memory systems, error detection and repair methods are essential to preserving data reliability. The Hamming code is one of the most popular error correction methods because it can identify and fix single-bit faults with little redundancy. However, continual switching activity in synchronous digital systems frequently results in increased power consumption for typical Hamming encoder and decoder circuits. In order to solve this problem, clock gating techniques have been incorporated into current designs in order to lower dynamic power by turning off the clock signal while it is idle. Even though clock gating increases power efficiency, next-generation low-power VLSI systems need more tuning. The construction of a low-power (7,4) Hamming code encoder and decoder utilizing reversible logic gates is presented in this study. In order to minimize information loss and heat dissipation while performing parity creation, syndrome computation, and error correction operations, the suggested architecture uses reversible gates such as Feynman and Double Feynman Gates. While the decoder computes syndrome bits to identify and fix single-bit faults in the incoming codeword, the encoder uses reversible logic networks to provide parity bits. Verilog HDL is used to implement the design, and FPGA synthesis tools are used to evaluate it. In comparison to traditional and clock-gated implementations, experimental results show that the suggested reversible logic-based architecture provides lower power usage and propagation delay. As a result, the suggested system offers a dependable, energy-efficient solution for contemporary VLSI communication and memory applications.**

Keywords— Error Detection and Correction, Reversible Logic, Low Power VLSI, Parity Bit Generator, Syndrome Generator, FPGA Implementation.

I. INTRODUCTION

Due to the growing need for high-speed communication, small hardware, and energy-efficient computing platforms, reliable data transmission and storage are critical components of contemporary digital systems. Because they convert data into formats appropriate for transmission and enabling the recipient to replicate the original data, encoders and decoders are essential parts of digital communication systems. These systems are widely used in telecommunication computer systems, networks, satellite communication, and memory storage applications where maintaining data integrity against interference and noise in a communication channel is crucial [1, 2]. Crosstalk, signal attenuation, channels noise, and device faults can all lead to errors in digital data exchange. To overcome this issue, detection and correction of errors (EDAC) algorithms are utilized to find and rectify errors without requiring retransmission. One of the most widely used EDAC algorithms is linear block codes due to its well-organized architecture and efficient error correction capability [3]. One of the most used linear block codes is the Hamming code, whose was first introduced by R. W. Hamming in 1950. It may correct single-bit errors and detect double-bit faults [4, 5].A codeword that allows error detection and correction at the receiver is created by adding parity bits to the original data bits in a Hamming coding system. To locate an error in the incoming data, the decoder recalculates the parity values and creates a syndrome. The popular (7,4) Hamming code adds three parity bits to four data bits to create a seven-bit codeword. XOR logic, parity generation networks, and syndrome computation circuits are typically used to create conventional Hamming encoder and decoder circuits [6–9]. Hamming codes are frequently employed in memory protection systems, embedded processors, and communication interfaces to increase system reliability as a result of the development of Very Large Scale Integration (VLSI) technology [10, 11]. However, power consumption has grown to be a significant issue in contemporary digital systems as circuit complexity rises and transistor size decrease [12]. In synchronous circuits, conventional Hamming encoder and decoder designs frequently result in needless switching activity, which raises dynamic power consumption because of constant clock operation [13, 14].

Clock gating has been proposed as an efficient low-power design method to solve this problem. When no data

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 3(1)(March)2026
Received: 10-1-2026 Accepted: 17-2-2026 Published: 27-3-2026

processing is needed, clock gating selectively turns off the clock signal for dormant circuit blocks. Clock gating greatly lowers dynamic power consumption while preserving the circuit's functional accuracy by avoiding needless switching activity in registers and combinational logic [15–17]. This approach is most useful in FPGA-based designs where clock networks contribute significantly to overall power usage [18]. In addition to clock-based optimization techniques, reversible logic has emerged as a practical technique for creating energy-efficient digital circuitry. Unlike conventional irreversible logic gates, reversible logic gates preserve a one-to-one mapping between inputs and outputs, ensuring that no information is lost during computation. According to Landauer's topic, information loss in irreversible circuits results in energy dissipation in the form of heat. Reversible logic is suitable for low-power VLSI and emerging nanotechnology applications since it reduces power dissipation by preventing this loss [19]. The primary objective of this work is to design and implement a power-efficient Hamming codes encoder and decoder using clock gating and reversible logic. Based on the conventional (7, 4) Hamming code, the proposed architecture incorporates reversible logic gates to reduce information loss and clock gating to reduce switching activity. The design is described using Verilog HDL, synthesis is done using Xilinx Vivado, and implementation is done on an FPGA platform. The foundational work "Design of Hamming Code Encoding and Decoding Based on Clock Gating Technique" is the main reference for standard architectural ideas and power optimization techniques [20, 21]. The findings show that the suggested architecture retains dependable mistake detection and correction capabilities while increasing power efficiency.

II. Existing Design

2.1 Hamming Code Generation

Faults that occur during the transmission of digital data are detected and corrected using linear block error correction codes called Hamming codes. In a Hamming coding system, a codeword is a coded sequence created by adding redundant parity bits to initial data bits. These redundant bits allow the receiver to detect and correct errors in the received data stream. For a message containing m data bits, the number of parity bits (r) required must meet the following criteria*:*

$$2r \geq m + r + 1$$

Where,

- m = number of data bits
- r = number of parity bits

This inequality ensures that sufficient redundant bits are available to represent all possible error locations within the codeword. For the commonly used (7, 4) Hamming code, the number of data bits is, m=4

Substituting into the above equation:

$$2r \geq 4 + r + 1$$

Testing values of r:

- r = 2→ $2^2 = 4 < 7$ (not sufficient)

- r = 3 → $2^3 = 8 \geq 8$

Therefore,

r=3

Thus, three parity bits are required to encode four data bits, producing a 7-bit Hamming codeword**.**

2.2 Conventional Hamming Code

The Hamming code encoder adds redundant parity bits to the original data bits to create an encoded codeword. In the widely used (7, 4) Hamming code, three parity bits are inserted to encode four input data bits into a seven-bit codeword. These parity bits give the receiver side the redundancy needed for error detection and correction [4, 5].The parity bits in this encoding system are positioned at codeword structure powers of two, namely places 1, 2, and 4. The data bits occupy the remaining positions. Using XOR operations on specific data bit combinations in accordance with predetermined parity equations, the encoder creates the parity bits. The parity bits *P*1, *P*2 and *P*4 for a (7, 4) Hamming code with data bits D0, D1, D2, and D3 are computed as follows:

$$P1=D0 \oplus D1 \oplus D3$$
$$P2=D0 \oplus D2 \oplus D3$$
$$P4=D1 \oplus D2 \oplus D3$$

The final encoded codeword is arranged as:

[P1 P2 D0 P4 D1 D2 D3]

The parity bit generator in conventional digital circuits is often created with XOR gate networks. These encoder architectures are commonly used in fault-tolerant digital systems, memory protection plans, and communication systems due to their simple implementation and reliable single-bit error correcting capability [6]–[9].

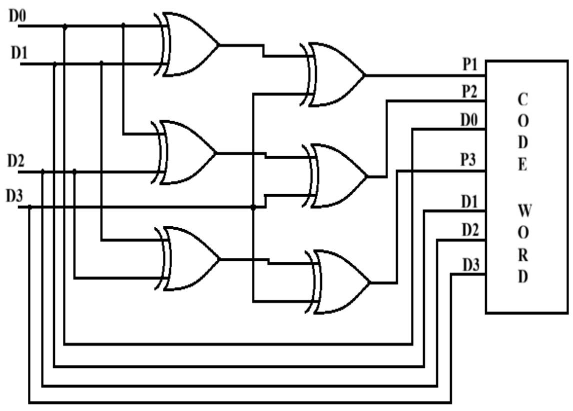


Fig. 1. Hamming Encoder.

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 3(1)(March)2026
Received: 10-1-2026 Accepted: 17-2-2026 Published: 27-3-2026

2.3 Hamming Code Decoder

Errors in the received codeword must be found and fixed by the receiver-side Hamming code decoder. After recalculating the parity values from the received bits, the decoder compares them to the parity bits that were transmitted. The result of this comparison is a group of bits known as syndrome bits, which show if an error has happened and pinpoint the location of the incorrect bit. Three syndrome bits, $S1$, $S2$ and $S4$ are present in the (7, 4) Hamming code. These are produced by doing XOR operations on particular combinations of the incoming codeword bits are used to calculate these bits. For a received codeword (D4, D3, D2, P4, D1, P2, P1)the syndrome bits are computed as:

$$S1=D3\oplus D1\oplus D0\oplus P1$$
$$S2=D2\oplus D3\oplus P2\oplus D0$$
$$S4=P4\oplus D1\oplus D2\oplus D3$$

The location of the fault within the codeword is indicated by the binary number formed by the three syndrome bits. There hasn't been a mistake if the syndrome value is 000. If the syndrome is not zero, the incorrect bit's location is immediately represented by the binary value, which can be fixed by flipping that bit. Like the parity generator in the encoder, the syndrome generator is usually constructed using XOR logic networks. Hamming codes are ideal for dependable digital communication and memory systems because of their decoding technique, which allows for automatic single-bit error correction and double-bit error detection [7–9].

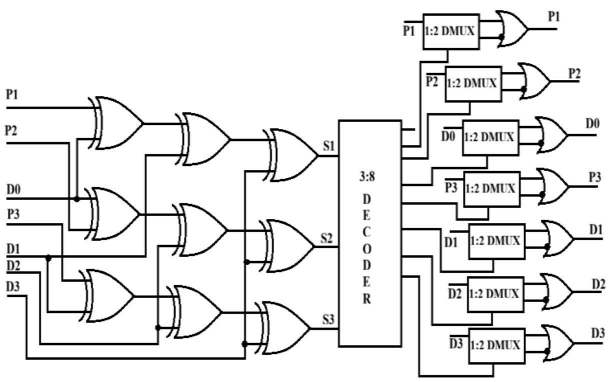


Fig. 2. Hamming Decoder.

2.4 Clock Based Hamming Code

To overcome the power consumption limitations of conventional designs, clock gating techniques have been introduced in existing Hamming code architectures. Clock gating is a widely used low-power design technique that reduces switching activity by disabling the clock signal when the circuit is not actively processing data.In the clock-gated Hamming encoder and decoder, the clock signal is enabled only when valid input data is available for encoding or decoding operations. When the circuit remains idle, the clock signal is automatically disabled, preventing unnecessary transitions in registers and associated combinational logic.

The circuit's switching power consumption is greatly decreased by regulating the clock activity. Only during active clock cycles does the encoder produce parity bits, and only when fresh data is received does the decoder compute syndrome bits. While preserving the Hamming code's ability to detect and fix errors, this selective clock activation increases energy efficiency. The architecture still uses traditional irreversible logic gates even though clock gating effectively lowers dynamic power consumption. According to thermodynamic principles, these gates inevitably result in information loss during computing, which causes further energy dissipation. Therefore, to obtain ultra-low-power digital systems, more optimization approaches are needed.

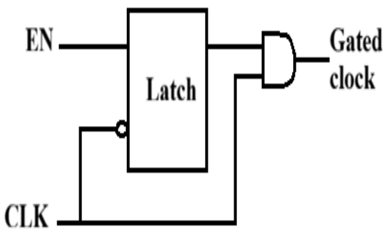


Fig. 3. Clock Gating Method.

III. PROPOSED DESIGN

Low-power digital circuit design has found success with reversible logic. Reversible gates, in contrast to traditional logic gates, preserve a one-to-one mapping between inputs and outputs, guaranteeing that no data is lost during processing. Landauer's principle states that heat is the result of energy dissipation when a single bit of information is lost. Consequently, in VLSI systems, reversible circuits aid in lowering power consumption and heat production. The suggested reversible Hamming coding architecture makes use of two basic reversible gates:

- FG or Feynman Gate
- F2G or Double Feynman Gate

These gates effectively carry out XOR operations, which are the main operations needed to compute the syndrome and generate parity in Hamming codes.

A. Feynman Gate

The Feynman Gate, also known as the Controlled NOT (CNOT) gate, is a 2×2 reversible logic gate. It consists of two inputs and two outputs. Let the input vector be (A, B)

The output vector is defined as,

$$Q = A\oplus B$$

Thus, the first output directly transfers the input value, while the second output produces the XOR operation between the two inputs. The Feynman gate is widely used in reversible logic circuits because:

- It performs XOR operation required in parity generation.
- It enables signal copying, which is not directly allowed in reversible logic.

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 3(1)(March)2026
Received: 10-1-2026 Accepted: 17-2-2026 Published: 27-3-2026

- It has low quantum cost and minimal delay.

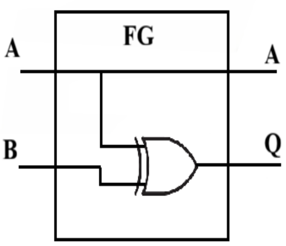


Fig. 4. Feynman Gate.

B. *Double Feynman Gate*

The Double Feynman Gate is a 3×3 reversible gate that produces two XOR outputs simultaneously. Let the input vector be, (A,B,C)

The output vector is given as,

$$Q=A\oplus B$$
$$R=A\oplus C$$

The Double Feynman gate is particularly useful in Hamming code circuits because parity generation requires multiple XOR combinations of data bits. By generating two XOR outputs simultaneously, the F2G gate reduces the number of required logic gates and improves circuit efficiency.

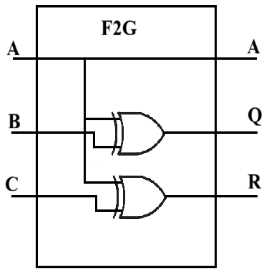


Fig. 5. Double Feynman Gate

C. *Reversible Logic Hamming Encoder*

The encoded codeword is created from the input data bits by the reversible logic Hamming encoder prior to transmission. Reversible logic gates, like the Feynman and Double Feynman gates, which can carry out the necessary logical operations while maintaining information, are used to implement the encoder in the suggested design. Unlike conventional logic circuits, reversible logic ensures that the number of inputs and outputs is equal, preventing information loss during calculation. The encoder block receives a 4-bit data word. Using the Hamming code encoding method, additional redundant bits are generated and put into the data stream in order to construct the encoded codeword. These useless bits allow errors to be identified and fixed at the receiving end. The reversible gate network performs the required parity generating operations while keeping a precise mapping between input and output signals. Because reversible logic does not destroy information during computation, the encoder circuit in the proposed architecture exhibits lower heat dissipation and improved power efficiency compared to conventional irreversible logic implementations. The encoder produces a 7-bit Hamming codeword, which includes data bits and parity bits, organized according to the Hamming code's structure. The encoder circuit uses a combination of Feynman and Double Feynman gates, arranged in a specific way. This design allows for an efficient encoding process, using fewer gates and consuming less energy.

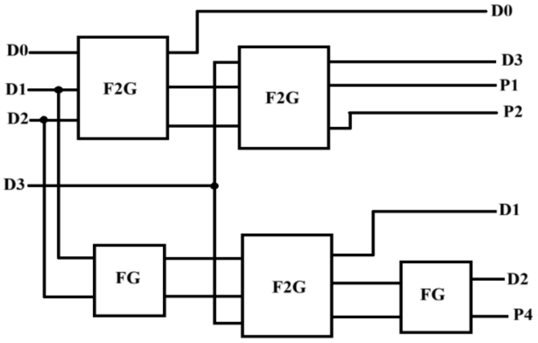


Fig. 6. Reversible Logic Encoder

D. *Reversible Logic Hamming Decoder*

Potential transmission errors are identified by the receiver-side reversible logic Hamming decoder. The decoder looks for bit errors in the encoded Hamming codeword after receiving it. The proposed approach minimizes information loss and maintains energy economy by using reversible logic gates to generate the decoder. The incoming codeword is processed by a network of reversible gates, which calculates new parity equations and generates a set of check values known as syndrome bits. These syndrome bits indicate whether the incoming data contains errors.

If the syndrome value indicates that there has been no error, the incoming data is considered accurate and the original data bits are recovered directly. However, if the condition indicates that a mistake exists, it also identifies the exact location of the error in the codeword. The system can effectively detect and locate single-bit errors due to this functionality. By keeping the decoding process maintains a reversible connection between inputs and outputs, reversible gates lower switching activity and improve the system's power efficiency.

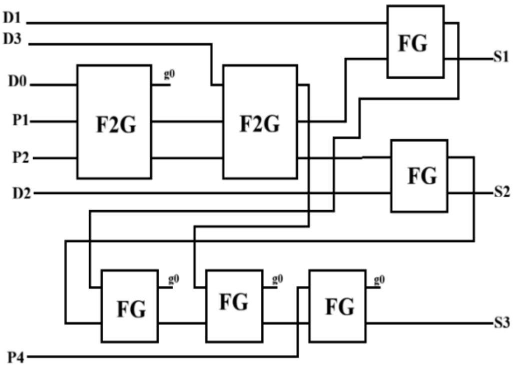


Fig. 7. Reversible Logic Decoder

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 3(1)(March)2026
Received: 10-1-2026 Accepted: 17-2-2026 Published: 27-3-2026

E. *Error Detection and Correction Block*

The error correction block flips the incorrect bit to restore the original data after the error location has been determined using syndrome bits. A decoder circuit uses the syndrome bits as input to activate the codeword's appropriate bit position. The incorrect bit is inverted via a reversible XOR gate that is controlled by the activated line.

TABLE 1. Syndrome Values and Error Position

Syndrome	Error Position
000	No error
001	Bit 1
010	Bit 2
011	Bit 3
100	Bit 4
101	Bit 5
110	Bit 6
111	Bit 7

The rectified codeword is obtained by inverting the incorrect bit. The rectified codeword is then used to extract the original data bits. Reversible logic gates are used to construct the correction mechanism, which minimizes switching activity and lowers overall power consumption.

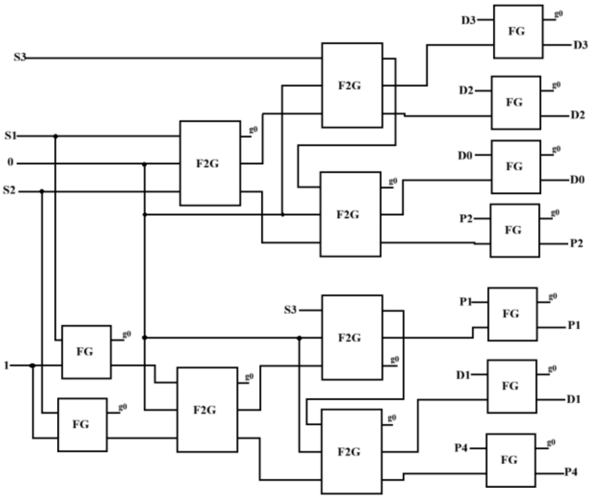
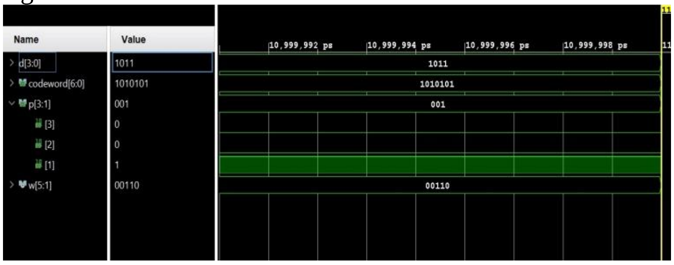


Fig. 8. Error Correction Block

IV. Results

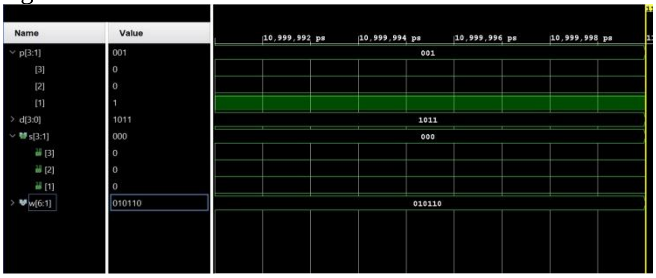
We developed and simulated the reversible logic-based Hamming encoder and decoder using Verilog HDL. The output waveforms confirm that the system encodes, decodes, and corrects errors exactly as expected. Figure 9 displays the simulation results for the reversible Hamming encoder. Given the input data D[3:0]=1011, the encoder outputs the parity bits P[3:1]=001. These bits are produced by XOR operations carried out by reversible gates, specifically Feynman and Double Feynman types. 1010101 is the last 7-bit codeword that matches the standard (7,4) Hamming code structure. This test confirms that the encoder can correctly provide parity while following reversible logic.



Name	Value
d[3:0]	1011
codeword[6:0]	1010101
p[3:1]	001
[3]	0
[2]	0
[1]	1
w[5:1]	00110

Fig. 9. Waveforms of Proposed Encoder

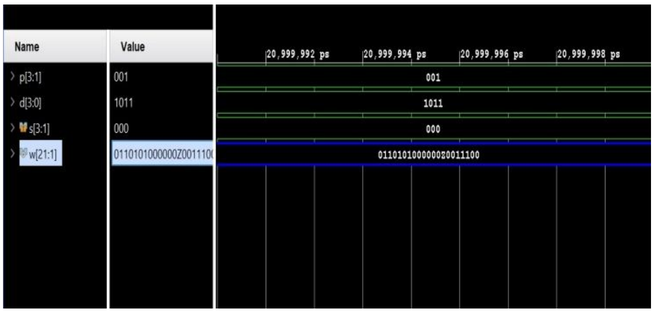
The simulation results for the reversible Hamming decoder are shown in Figure 10. The system finds the syndrome bits S[3:1] after applying the encoded codeword. In this case, the syndrome is 000, meaning that there are no errors in the data that was received. Therefore, the decoder returns the original message bits exactly as they were transmitted. This outcome shows that the logic used for syndrome generation and verification operates perfectly. The research show how reversible logic gates can perform the XOR-intensive operations needed for encoding and decoding. Because they prevent data loss, these circuits drastically lower redundant switching and power consumption when compared to traditional, irreversible logic circuits.



Name	Value
p[3:1]	001
[3]	0
[2]	0
[1]	1
d[3:0]	1011
s[3:1]	000
[3]	0
[2]	0
[1]	0
w[6:1]	010110

Fig. 10. Waveforms of Proposed Decoder

The error detection and correction unit checks the syndrome values. If the syndrome is not zero, it shows the exact part of the codeword that is wrong. The system then uses XOR-based logic to flip that specific bit back to its correct state. An mistake at the third bit position is indicated by a syndrome of 011 example, which causes the circuit to flip it. If the syndrome stays at 000, the data flows through unchanged because no errors were found. In summary, these simulation results show that the design can accurately handle error management, encoding, and decoding. By employing reversible logic, we ensured that the system operates effectively while maintaining energy efficiency, making it a perfect match for modern VLSI and high-speed communication tools.



Name	Value
p[3:1]	001
d[3:0]	1011
s[3:1]	000
w[21:1]	0110101000000Z001110(

Fig. 11. Waveforms of Error Detection and Correction Block

V. Conclusion

This study successfully developed and evaluated a low-power (7,4) Hamming code encoder and decoder based only on reversible logic. The main aim was to combine the accurate error-correction capabilities of Hamming codes with the power-saving benefits of reversible circuits. By using Feynman and Double Feynman gates to preserve an exact one-to-one input-output mapping, the structure successfully prevents data loss and lowers heat dissipation. When compared to traditional or clock-gated Hamming structures, this reversible logic method shows a notable decrease in switching activity, which improves energy performance. The encoder produces parity bits efficiently, while the decoder accurately computes syndrome bits to identify and fix single-bit faults.We used FPGA synthesis and Verilog HDL to verify that the design works as expected. The findings indicate that reversible logic is a very attractive path for the future development of low-power VLSI, particularly for embedded technology, memory hardware, and communication links where energy conservation is essential. Ultimately, this design offers an error control code method of operation that is dependable and scalable. Higher-order Hamming codes may eventually be supported by this system. The system should be tested on state-of-the-art nanotechnology or quantum computing hardware in the future to try to drastically lower quantum costs and garbage outputs.

References

[1] S. Garg and A. K. Sharma, "An introduction to various error detection and correction schemes used in communication," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 8, no. 4, pp. 1–6, April 2018.

[2] E. Khan and N. Pandey, "Review of binary codes for error detection and correction," International Journal of Computer Applications, vol. 180, no. 29, pp. 15–19, April 2018.

[3] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, 2nd ed. Oxford, U.K.: Oxford Univ. Press, 2010.

[4] R. W. Hamming, "Error detecting and error correcting codes," Bell System Technical Journal, vol. 29, no. 2, pp. 147–160, April 1950.

[5] M. D. Ercegovac and T. Lang, Digital Arithmetic. Amsterdam, Netherlands: Elsevier, 2003.

[6] T. Zhang and Q. Ding, "Design of (15,11) Hamming code encoding and decoding system based on FPGA," in Proc. International Conference on Computer Engineering and Technology, Kuala Lumpur, Malaysia, 2011, pp. 215–219.

[7] V. Badoler, "Implementation of multidirectional parity check code using Hamming code for error detection and correction," International Journal of Engineering Research and Technology, vol. 7, no. 5, pp. 210–214, May 2018.

[8] A. Kumar and S. Patel, "VLSI design of parity check code with Hamming code for error detection and correction," International Journal of Electronics and Communication Engineering, vol. 13, no. 2, pp. 85–90, 2019.

[9] J. Gu and C. H. Chang, "Ultra low-voltage, low-power compressor circuits," IEEE Transactions on Circuits and Systems *II*, vol. 50, no. 9, pp. 548–556, September 2003.

[10] M. Margala and N. G. Durdle, "Low-power low-voltage compressors for VLSI applications," *IEEE* Transactions on Very Large Scale Integration (VLSI) Systems, vol. 7, no. 2, pp. 156–161, June 1999.

[11] J. Liang, J. Han, and F. Lombardi, "New metrics for reliability of approximate adders," IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760–1771, September 2013.

[12] K. Prasad and K. K. Parhi, "Low-power compressor architectures," IEEE Transactions on Circuits and Systems *II*, vol. 48, no. 3, pp. 255–264, March 2001.

[13] D. Radhakrishnan and A. P. Preethy, "Low-power CMOS pass logic designs," IEEE Transactions on Circuits and Systems II, vol. 47, no. 2, pp. 123–129, February 2000.

[14] K. S. Chong, B. H. Gwee, and J. S. Chang, "Micropower low-voltage design techniques," *IEEE* Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 9, pp. 1065–1079, September 2005.

[15] S. Cheemalavagu, P. Korkmaz, K. V. Palem, and B. Erez, "Probabilistic CMOS switching for energy-efficient computing," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 5, pp. 552–565, May 2005.

[16] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in arithmetic circuits," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 4, pp. 556–569, April 2011.

[17] M. Siva Kumar, R. Srinivasan, and P. Ramesh, "Ultra-low power full adder design using mixed logic styles," International Journal of Electronics and Communication Engineering, vol. 12, no. 6, pp. 301–305, 2019.

[18] D. Ramamma and V. Ganesan, "Low power VLSI implementation of convolution encoder and decoder using Verilog HDL," International Journal of Advanced Engineering Research and Science, vol. 7, no. 3, pp. 112–118, March 2020.

[19] S. Shoba and R. Nakkeeran, "GDI based full adders for energy efficient arithmetic applications," Microelectronics Journal, vol. 48, pp. 1–8, 2016.

[20] Xilinx Inc., Vivado Design Suite User Guide. San Jose, CA, USA: Xilinx Inc., 2023.

[21] K. Kalaichevi, "Design of Hamming code encoding and decoding based on clock gating technique," Kronika Journal, vol. 25, no. 6, pp. 585–597, June 2025.