International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Enhancing Efficiency and Performance with RFT Gate Designs

Ravi L. S.[*1], Naveen K. B.[2], B N Shobha[3], Nagesh R.[4]

Submitted:11/03/2024 **Revised**: 26/04/2024 **Accepted**: 03/05/2024

Abstract: Ensuring congruence between the input vector's parity and that of the output vector stands as a pivotal requirement within fault-tolerant reversible logic gate circuits. This parity-conserving attribute assumes paramount significance in discerning latent anomalies within the circuitry. Particularly in the realm of nanotechnology applications, this trait serves as a linchpin in crafting fault-tolerant systems. The realm of reversible logic gates (RLGs) designated for fault tolerance enjoys widespread recognition for its adeptness in nascent computing paradigms, exemplified by optical and quantum computing. Within our discourse, we introduce the concept of reversible fault-tolerant lookup tables (LUTs), wielding profound influence in the evolution of Configurable Logic Blocks (CLBs). This architectural construct of CLBs amalgamates fault-tolerant reversible logic constituents, encompassing D-Latches, Master-Slave Flip-flops, and Multiplexers. Our proposed architecture undergoes exhaustive simulation, logical scrutiny, and FPGA Spartan 3-based implementation. The resultant simulation outcomes and practical deployments corroborate the efficacy underlying the design of reversible fault-tolerant (RFT) gates. Particularly noteworthy are the discernible reductions observed in power dissipation and latency within these gates. In the precincts of 90 nm CLB technology, a conspicuous power decrease of around 95.5% is discerned, accentuated further to 98% in the context of 45 nm CLB technology. These empirical revelations serve to underscore the inherent potential of reversible fault-tolerant gate configurations, poised to augment efficiency and performance benchmarks across FPGA applications.

Keywords: *CLB, FPGA, Fault Tolerant, Unwanted Output, Look-Up Table, Reversible Logic.*

1. Introduction

Addressing the challenge of information loss during computation, reversible computing inherently tackles the issue of heat generation associated with such loss. Bennett's research underscores that achieving zero energy dissipation becomes attainable solely when all gates within a network exhibit reversibility [1]. Because of this capability, reversible computing has been included into several new technologies such as ultra-low-power CMOS design, quantum computing, optical computing, and nanotechnology [2]. Reversible logic is used in these areas to reduce power consumption.

Many methods for reversible logic synthesis have been developed in the quest to implement reversible functions [3–4]. These techniques aim to decrease a number of parameters, such as the number of gates, undesired outputs, quantum cost, latency, area, and power consumption. The crux of efficient reversible logic synthesis lies in optimizing these design criteria to realize the intended functionality while mitigating energy consumption and other resource utilization metrics.

[1]*Research Scholar, Dept. of Electronics and Communication Engg., BGS Institute of Technology, Adichunchanagiri University, B.G. Nagara, Mandya-571448, Assistant Professor, Rajeev Institute of Technology, Hassan.*
ORCID ID : 0009-0005-8405-5977
[2,3]*Professor, Dept. of Electronics and Communication Engg.,BGS Institute of Technology, Adichunchanagiri University, B.G. Nagara, Mandya.*
[4]*Asst.Professor,Dept. of Electronics and Communication Engg.,GSKSJTI,, Bangalore.*
Corresponding Author Email: naveenkb@bgsit.ac.in ravigowda1234@gmail.com

Maxwell and Szilard laid the groundwork for understanding the intrinsic relationship between a single bit and its associated minimum entropy [5]. They posited that in each fundamental computational operation, there exists a minimum threshold of energy dissipation, denoted as $KT\ ln(m)$. In this case, T represents the ambient absolute temperature, while K stands for Boltzmann's constant (1.38×10-23 J/K), and m correlates with an integer value tied to the quantity of processed bits [5].

Subsequently, the Landauer principle elucidated that circumventing energy dissipation is contingent upon achieving reversibility within the system [6]. Bennett expanded upon the concept of reversible computation, elucidating that in systems where no information is lost, energy dissipation tends towards zero, represented by $KT\ ln(1)$ [7].

On the flip side, systems that undergo irreversible processes are constrained by a fundamental threshold in energy dissipation, quantified as $KT\ ln(2)$ for every bit erased [8]. Despite their capacity to retain information rather than wiping it out, irreversible systems often lack a clear trajectory from one state to its precursor, resulting in irretrievable energy expenditure unless computational actions are reversible.

The imperative to curtail energy dissipation stands paramount in the trajectory of enhancing computer hardware efficacy, particularly in light of the potential escalation in power dissipation stemming from heightened hardware density, should Moore's Law persist. According

to Gordon Moore's prediction, transistor and resistor counts on a chip will double every 18 months. This highlights the need of reducing power dissipation to avoid overheating situations [9].

Within this framework, reversible logic emerges as a pivotal player. Reversible circuits can be thought of as a particular subset of quantum circuits as reversibility constraints also apply to quantum evolution [10]. Reversible fault-tolerant circuits, facilitating the identification of flawed signals sans additional data corruption, are gaining traction. Parity checking, a prevalent technique for detecting single-level faults in communication, finds application in reversible fault-tolerant circuits [11].

The field of reversible and fault-tolerant circuits has attracted significant interest recently from a variety of fields, including quantum-dot cellular automata, DNA technology, nanotechnology, optical computing, program testing and debugging, and database recovery [12-14]. These strides underscore the significance of minimizing energy dissipation and ensuring fault tolerance within computational systems.

Reprogrammability, faster processing, more density, no more ongoing engineering expenses, and the capacity to carry out massively parallel processes are some of these advantages of FPGAs. An FPGA is structurally made up of a network of interconnections and programmable logic modules. These logic blocks can be designed to do complicated combinational operations as well as sequential ones, which will affect the FPGA's overall performance and density [15].

The most prevalent logic blocks in SRAM-based FPGAs include Look-Up Tables (LUTs) and Plessey blocks. LUTs, while extensively utilized, grapple with exponential complexity amplification concerning inputs [16].

Researchers have been drawn to the amalgamation of FPGA capabilities and the energy-conserving attributes of reversible logic. This interest has precipitated the development of reversible FPGAs aimed at curtailing power consumption [17].

An alternative method has been suggested, which focuses mostly on latency as a performance metric and uses fault-tolerant approaches to setup specific parts of the Plessey Logic Block on FPGAs. Nevertheless, this approach has not yielded noteworthy enhancements and has failed to tackle problems related to space and energy usage [18]. This research proposes improved design techniques to reduce power and area footprints while improving metrics like as gate count, unwanted outputs, quantum cost, and latency, by utilizing the foundation established in reversible logic. The objective of this research is to propel the domain of reversible FPGAs forward by addressing pivotal performance benchmarks and augmenting power efficiency and area utilization.

2. Literature Survey

The importance of fault-tolerant and reversible logic synthesis was discussed in detail in the preceding section. A thorough explanation of the transistor representations required for MOS transistor circuit implementation has been included in this section. The channel length was 0.12 mm, and the average delay was 0.030 ns, according to the results. Throughout this discourse, signals exhibiting stability less than 0.001 ns have been identified as glitches.

Venkata et al. [19] undertook a study concentrating on the integration of LUTs in FPGAs, employing multiplexers and pass-transistors to tackle power dissipation concerns. They underlined the significance of n-MOS transistor low-power multiplexers, which are essential parts of FPGA routing blocks and CLBs.

Xie et al. [20] introduced two innovative FPGA implementations leveraging memristor-based CLBs alongside an associated automatic design flow. They emphasized the advantages of memristor technology, including its high integration density, low standby power consumption, and non-volatile nature. These implementations underwent benchmarking using Toronto 20 and were juxtaposed with cutting-edge solutions concerning area and delay considerations.

Li et al. [21] achieved outstanding performance in terms of area power delay product (APD) by using an inventive LUT design that incorporates a decoding transmission gate. In addition, a multi-threshold voltage strategy was employed to minimize circuit leakage current while maintaining functioning. Their proposed design exhibited a remarkable 47.4% enhancement in APD compared to the structures under comparative analysis. Moreover, they devised an area-saving ring vibration test circuit for assessing input-output propagation delays, achieving a substantial 158% area saving relative to a simplistic test chain. The simulations were conducted utilizing the H spice-circuit simulator tool.

Gao et al. [22] carried out an extensive analysis of the design and layout of Xilinx Spartan-3 FPGAs and proposed a circuit configuration that is specifically tailored for dynamic burn-in testing. Using the CLB structure, they put SliceLs into 4-input XOR and 4-input NXOR modes and SliceMs into RAM chain and shift register modes to construct configuration circuits. The outcomes of the simulation showed that the suggested configuration circuit optimized resource consumption by providing better input-controllability and output-observability. Xilinx Spartan-3 and SliceMs were the primary elements employed for the analysis.

The use of Quantum-dot Cellular Automata (QCA) technology for establishing CLBs in Xilinx FPGAs was examined in a research by Zhang et al. [23]. Using QCA Designer software, they designed and simulated a number of QCA-based circuits, such as multiplexers, decoders, and 8-bit LUTs. Their analysis brought to light the special qualities of QCA technology, including its high speed, density, and low power usage.

3. Materials and Methods

3.1. Reversible Gate and Properties

The n*n configuration, which preserves parity between the input and output ports, can be used to identify reversible gates. There is a direct connection between the input and output arrays of reversible gates as the number of outputs always matches the number of inputs. A reversible gate may provide a distinct output array for each input array thanks to this fundamental feature, and vice versa.

A circuit must have reversible gates in order to be considered reversible, and the number of gates used is usually used to measure circuit complexity. Therefore, reducing gate consumption to improve overall functionality and efficiency is always at the center of the circuit optimization quest.

3.1.1. Gate Structure and Parameters

The reversible gate is the fundamental component of a reversible circuit, serving as its center. It creates full and direct linkages between the input and output vectors by acting similarly to a bijection function. This crucial feature ensures that each input signal has a unique counterpart in the output, hence simplifying the retrieval process. Figure 1 shows a schematic illustration of a reversible gate.

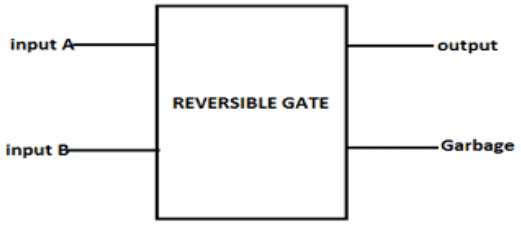


Fig. 1. Structure of reversible gate

Let O_v stand for the output vector and I_v for the input vector. The input vector may be written as follows: $I_v = (I_0, I_1, I_2, \ldots I_{n-1}, I_n)$, and similarly, the output vector as $Ov = (O_0, O_1, O_2, \ldots, O_{n-1}, O_n)$. Each specific input corresponds to a distinct relationship with the output, represented as $I_v \leftrightarrow O_v$. The schematic depiction of an $n \times n$ reversible gate is shown in Figure 2.

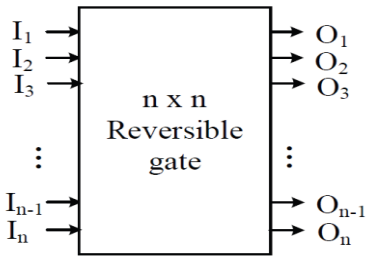


Fig. 2. Block diagram for $n \times n$ reversible gate

3.1.2. Unwanted Output

An additional output that is introduced to a circuit with an n-input and m-output function in order to ensure reversibility is known as an unwanted output. By maintaining a one-to-one mapping between inputs and outputs, this extra output helps to maintain the circuit's reversibility. The function is made reversible by adding unwanted outputs, which allow the original input to be restored from the output. Because they enable bidirectional computing and preserve information, unwanted outputs are essential to reversible logic architecture.

3.1.3. Quantum Cost

The quantum cost represents the resources associated in implementing every reversible gate [24]. Each reversible gate is associated with a particular quantum circuit that is built from basic quantum gates like the V, NOT, V+, and Ex-OR (CNOT) gates. In this case, V gate is a square-root-of-NOT gate, and V+ is the Hermitian conjugate of the V gate. The quantum circuit constructed from these fundamental quantum gates represents the reversible gate, and the quantum cost quantifies the resources needed to do this. It serves as a gauge for reversible circuit complexity and efficiency.

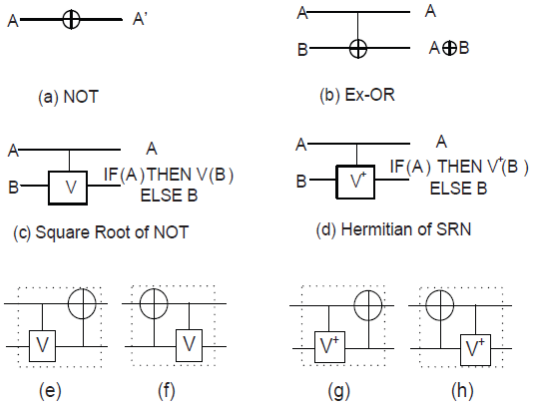


Fig. 3. Basic quantum gates and their symmetric patterns

The representations of the four basic quantum gates are shown in Figures 3(a)–(d), respectively. A unit cost is associated with each of these elementary gates, as well as their symmetrical equivalents (Figure 3(e)-(h)) [24]. The quantum cost of a reversible gate depends on the total number of fundamental gates and their symmetric

configurations in the corresponding quantum circuit.

Figure 4 shows a quantum circuit representation of the reversible Toffoli gate, which is a necessary component for constructing reversible logic. This Toffoli gate's quantum cost, which represents the resource cost required for its physical manifestation, is defined as 5.

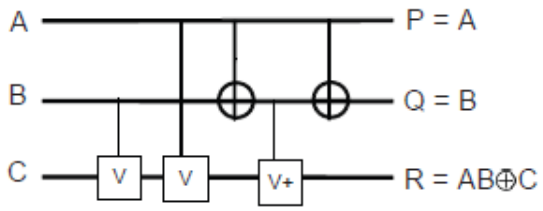


Fig. 4. Quantum circuit of Toffoli gate

Within reversible circuit design, quantum cost stands out as a pivotal cost metric, tightly linked with the tangible resources required for circuit implementation. Throughout the design phase, striving for a reduced quantum cost holds paramount importance, signaling the efficiency of the design endeavor. Hence, the endeavor to minimize quantum cost emerges as a central objective in reversible circuit development, paving the path toward resource-efficient and economically viable implementations.

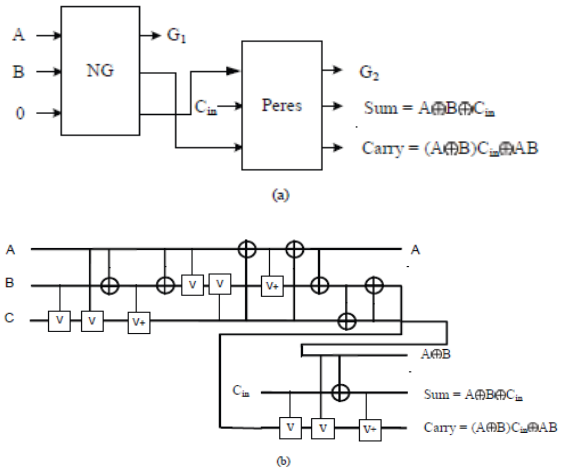
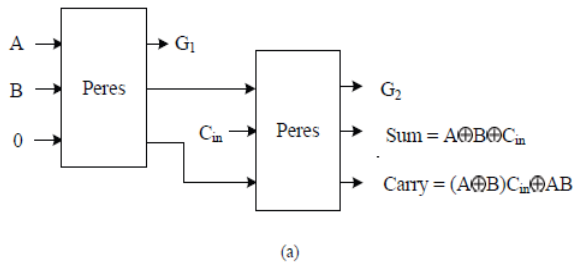


Fig. 5. Reversible full-adder (a) Block diagram (b) Quantum circuit

Two quantum circuits representing the reversible full-adder are shown in Figures 5(a) and 5(b), with quantum costs of 17 and 8, respectively. The latter circuit is more efficient since it has a smaller quantum cost.



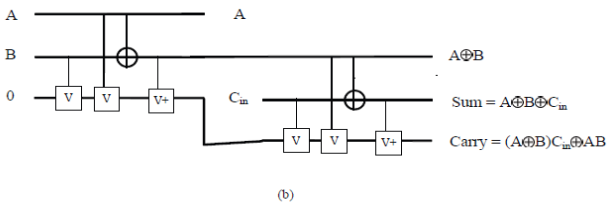


Fig. 6. Reversible full adder 2 (a) Block diagram (b) Quantum circuit

In the realm of reversible logic strategy, minimizing the quantum cost is one of the main objectives. Researchers invest substantial effort in minimizing this expense as much as feasible. Several studies [25–26] have focused on reducing the cost of circuits and simplifying quantum circuits.

Different templates for lowering quantum cost are shown in Figure 7(a)–(c), along with methods for streamlining and improving the quantum circuit design process. These templates provide strategies to reduce resource usage and increase the effectiveness of implementations of reversible logic.

Furthermore, important features of the V and V+ gates—which are crucial parts of reversible logic architecture—are highlighted in Figures 7(d)–(e). Understanding these characteristics enables designers to efficiently utilize the unique characteristics of these gates, which further helps to minimize quantum cost and optimize reversible circuit implementations.

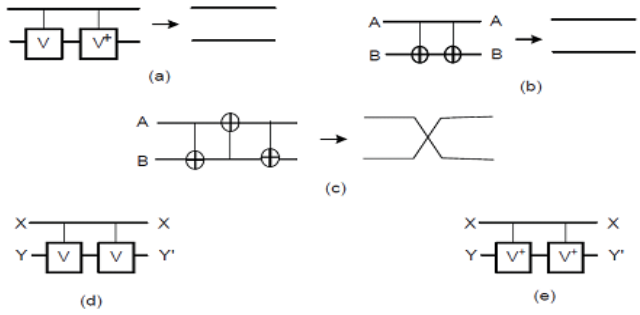


Fig. 7. Rules in minimization of quantum circuit

3.1.4. Delay

In circuit design, delay is an important factor that directly affects the efficiency and performance of the circuit. Irrespective of their computational intricacy, previous studies conventionally regarded the count of reversible gates on the critical path as equivalent to one delay unit. This technique made the analytical process more efficient and made the evaluation of circuit timing easier to understand.

An array of reversible gates, each serving a different purpose inside the circuit, make up reversible circuits. Many reversible gates with different characteristics and

advantages have been presented and studied in academic publications. Reversible gates that are commonly mentioned include:

- Toffoli Gate: It is a multipurpose reversible gate that performs a controlled-NOT operation. Because of its broad use, it is essential in reversible technology.
- Fredkin Gate: It is often referred to as the controlled-SWAP gate, is helpful in situations requiring reversible data manipulation since it swaps the second and third input bits if the first input bit is true.
- Feynman Gate: Named in homage to physicist Richard Feynman, this gate embodies the reversible iteration of the AND operation, producing a true output solely when all input bits are true.
- Peres Gate: Executing the reversible XOR operation, the Peres gate yields a true output when the count of true input bits is odd, thereby facilitating various computational tasks.
- CNOT Gate: Another crucial kind of universal reversible gate is the Controlled-NOT gate, which changes the state of the second input bit in response to the truth value of the first input bit. It finds widespread application in both quantum computing and reversible circuit design domains.

These gates, along with others described in the literature, form the building blocks of reversible circuits, allowing designers to construct complex and efficient computational systems while minimizing energy consumption and maximizing performance.

3.1.5. Feynman Gate (CNOT Gate or Ex-OR Gate)

As a 2-input 2-output reversible gate, the Feynman gate (FG) exhibits the input-output pattern shown below:

$$(A, B) \leftrightarrow (P = A, Q = A \oplus B) \qquad (1)$$

In this case, A and B stand for the input variables, whereas P and Q stand for the corresponding output variables. Figures 8(a) and (b) depict the quantum circuit and block diagram of the Feynman gate, respectively. It is noteworthy that the Feynman Gate's quantum circuit consists of a single Ex-OR gate with a delay of 1Δ and a quantum cost of 1. This gate, sometimes called an Ex-OR gate, repeats input to solve the fan-out issue in reversible logic, as seen in Figure 8(c). Moreover, it produces the opposite signal to the input, as seen in Figure 8(d).

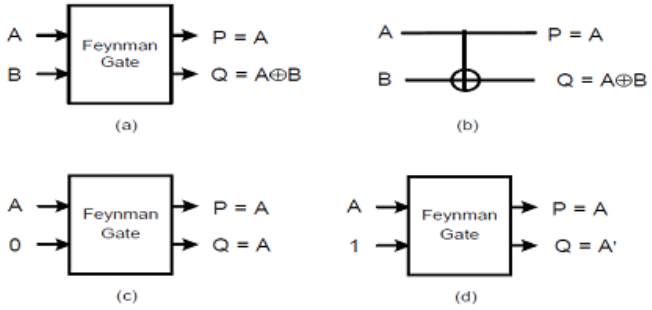


Fig. 8. Feynman Gate

3.1.6. Toffoli Gate (TG)

It is a reversible gate that has three inputs and three outputs. It represents the input-output pattern that is described as follows:

$$(A, B, C) \leftrightarrow (P = A, Q = B, R = AB \oplus C) \qquad (2)$$

Here, the equivalent output variables are P, Q, and R, while the input variables are A, B, and C. Figure 9 displays the TG's quantum circuit and block diagram. To create the TG quantum circuit, two Controlled-V gates, one Controlled-V + gate, and two Ex−OR gates are needed; the remaining basic gates need to be linked in a serial configuration. Consequently, the quantum cost of the Toffoli gate is 5 at a delay of 5Δ.

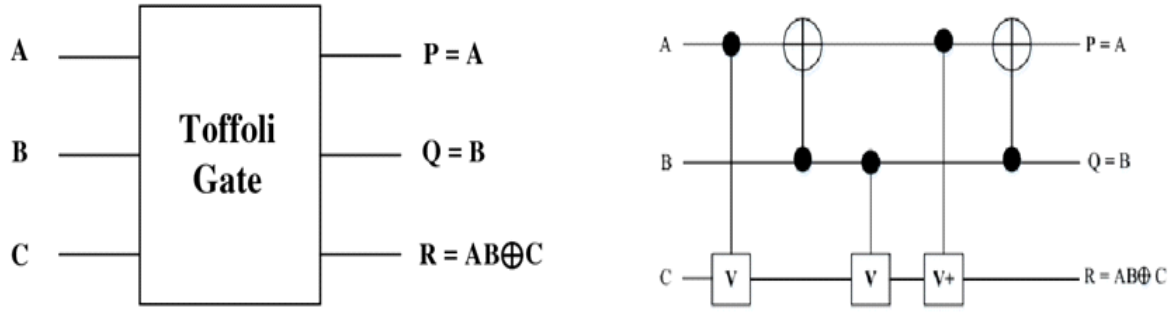


Fig. 9. Toffoli Gate and its implementation in Quantum Gate

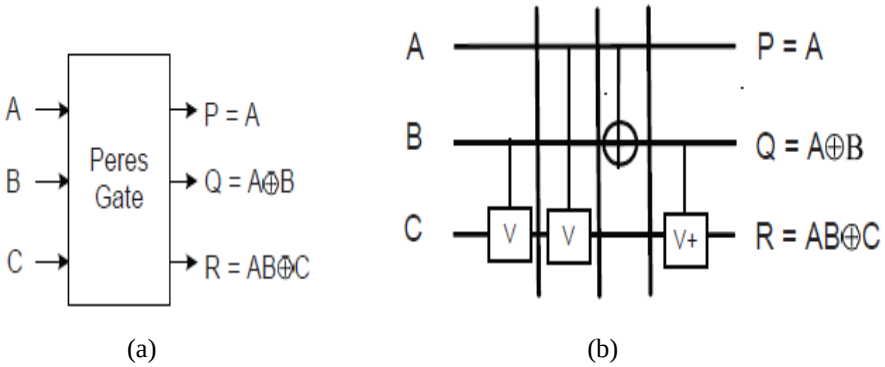
3.1.7. Peres Gate (PG)

Three inputs and three outputs make up the reversible PG, which symbolizes the input-output pattern that is described below:

$$(A, B, C) \leftrightarrow (P = A, Q = A \oplus B, R = AB \oplus C) \quad (3)$$

Here, the input variables are denoted by the letters A, B, and C, while the output variables are denoted by the letters P, Q, and R. The Peres gate (PG) is shown in Figure 10(a) [27], whereas Figure 10(b) shows the quantum circuit of the Peres gate (PG).

Two Controlled-V + gates, one Controlled-V gate, and one Ex−OR gate must be integrated to create the Peres gate's quantum circuit, which has a four-quantum cost. Essentially, the Peres Gate allows for the simultaneous creation of two output functions (from Q and R) by combining the Feynman Gate (FG) and Toffoli Gate (TG) functions. When constructing full-adder circuits or other circuits requiring an Ex-OR operation and comparable actions to those performed by the Toffoli gate, but at a lower cost, this gate is recommended.



(a) (b)

Fig. 10. (a) Peres Gate (b) Quantum circuit

3.1.8. Fredkin Gate (FRG)

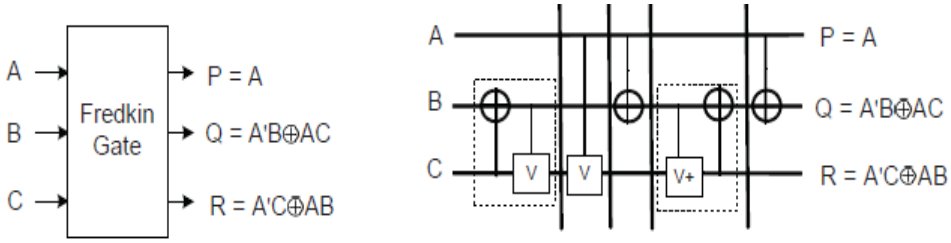
It is a 3-input 3-output reversible gate, encapsulates the input-output pattern shown below:

$$(A, B, C) \leftrightarrow (P = A, Q = A'B \oplus AC, R = AB \oplus A'C) \quad (4)$$

In this case, A, B, and C stand for the input variables, whereas P, Q, and R stand for the output variables. The block diagram and quantum circuit of the FRG are shown in Figures 11(a) and 11(b), respectively.

The two Ex-OR gates, one Controlled-V gate, two dotted rectangles, and one FRG make up its quantum circuit. As a result, five is the quantum cost of the FRG. Due to the symmetrical patterns and arrangement of the fundamental quantum gates, a delay of 5Δ is necessitated for the FRG.

The FRG is significant in reversible literature because it may provide one output directly as input and two distinct outputs as separate Boolean functions. Because the other two inputs swap in the output when the first input is 1, it is also known as the SWAP gate. FRG is also used in the building of multiplexer circuits.



(a) (b)

Fig. 11. (a) FRG Gate (b) Quantum circuit

4. Proposed Methodology

Within the domain of digital circuit design, a Look-Up Table (LUT) functions as a memory module, employing outputs to enact a truth table, wherein distinct input combinations yield specific logic outputs [25]. An n-input LUT is typically constructed using multiplexers and a memory cell, wherein the multiplexer's selection inputs align with the LUT's input bits, while the multiplexer's input bits are stored within a memory cell.

Several scholars have proposed enhanced designs for the reversible Plessey logic block within FPGAs in papers [29-30]. Conversely, in paper [25], researchers suggested a LUT-based reversible FPGA design. However, the precise structure of the LUT remained largely unexplored in [25] [30] [31].

BSP, FG, HNFG, FRG, and TG gates were utilized by

researchers in article [25] to build the logic components of the reversible FPGA. They did not, however, include a thorough description of the look-up table's architecture. Additionally, they did not optimize their architecture in terms of gate count, trash outputs, or quantum costs.

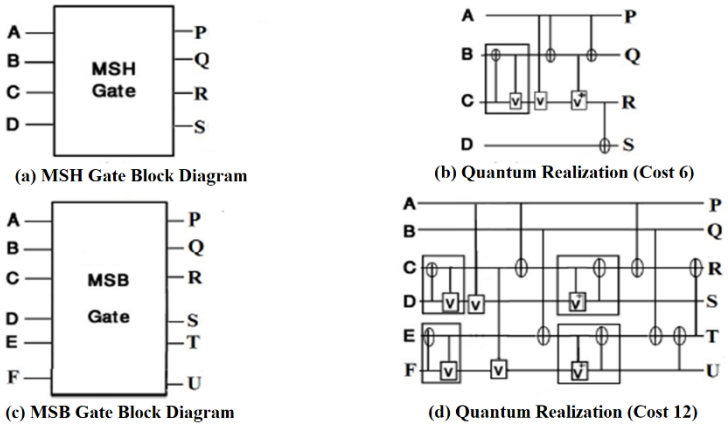
We provide block diagrams, quantum realizations, and truth tables for two new reversible fault-tolerant gates.

4.1. Mubin-Sworna-Hasan (MSH) Gate

In Figures 12(a) and 12(b), the proposed MSH gate with an input vector of [A, B, C, D] and an output vector of [P, Q, R, S] is displayed along with its quantum implementation.

4.2. Mubin-Sworna-Babu (MSB)Gate

Figure 12(c) depicts the block design of this gate, whereas Figure 12(d) shows the quantum implementation. [A, B, C, D, E, F] is its input vector, while [P, Q, R, S, T, U] is its output vector.



(a) MSH Gate Block Diagram

(b) Quantum Realization (Cost 6)

(c) MSB Gate Block Diagram

(d) Quantum Realization (Cost 12)

Fig. 12. MSH Gate and MSB Gate

Quantum realization of MSH Gate:

$$|A\ B\ C\ D> \ -> \ |A\ B\ C\ D> \ XOR\ |P\ Q\ R\ S> \quad (5)$$

Table 1. Truth table for MSH gate

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
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	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	1

Where, $S_1 = A,\ \ S_2 = B, S_3 = C,\ \ S_4 = D, S_5 = P,\ \ S_6 = Q, S_7 = R, S_8 = S$ for MSH Gate

Quantum Realization of MSB:

$$|A\ B\ C\ D\ E\ F> \ -> \ |A\ B\ C\ D\ E\ F> \ XOR\ |P\ Q\ R\ S\ T\ U> \quad (6)$$

Table 2. Truth table for MSB gate

G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	0	1	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	1	0	1	0	0	0	1	0	1
0	0	0	1	1	0	0	0	0	1	1	0
0	0	0	1	1	1	0	0	0	1	1	1
0	0	1	0	0	0	0	0	1	0	0	0

0	0	1	0	0	1	0	0	1	0	0	1
0	0	1	0	1	0	0	0	1	0	1	0
0	0	1	0	1	1	0	0	1	0	1	1
0	0	1	1	0	0	0	0	1	1	0	0
0	0	1	1	0	1	0	0	1	1	0	1
0	0	1	1	1	0	0	0	1	1	1	0
0	0	1	1	1	1	0	0	1	1	1	1

Where, $G_1 = A,\ G_2 = B, G_3 = C,\ G_4 = D, G_5 = E,\ G_6 = F, G_7 = P, G_8 = Q, G_9 = R,\ G_{10} = S, G_{11} = T,\ G_{12} = U$ for MSB Gate

4.3. An Innovative RFT Architecture for, 4×1 Multiplexer, Master-Slave Flip Flop, and D-latch

Our new RFT D-latch is schematically shown in Figure 13(a), where the desired output $S = Clk' \oplus Clk.D$ is obtained by means of a single MSH gate. It's interesting to note that this design only has one constant input and two redundant outputs (G_1 and G_2).

We now illustrate the architecture of our cutting-edge RFT D-latch in Figure 13(b). It employs a Feynman gate to provide the necessary output, $Q = D \cdot E \oplus E'$ in a feedback loop. Similar to the previous concept, this setup consists of merely two redundant outputs, G_1 and G_2, and one constant input.

The design of our new RFT write-based master-slave flip-flop is shown in Figure 13(c). As mentioned earlier, this design generates the necessary outputs, B and B+, by combining an MSB gate with a RFT D-latch. Additionally, we offer a novel MSH gate that we employ to construct a fault-tolerant, reversible 4×1 Mux that is managed by the subsequent formula:

$$Q = (M_0 \cdot I_0) \oplus (M_1 \cdot I_1) \oplus (M_2 \cdot I_2) \oplus (M_3 \cdot I_3) \quad (7)$$

As Property 1 explains, the minimum trash output of our proposed RFT multiplexer sets it apart.

$$O_{mux} = A'B'C + A'BE + AB'D + ABF \quad (8)$$

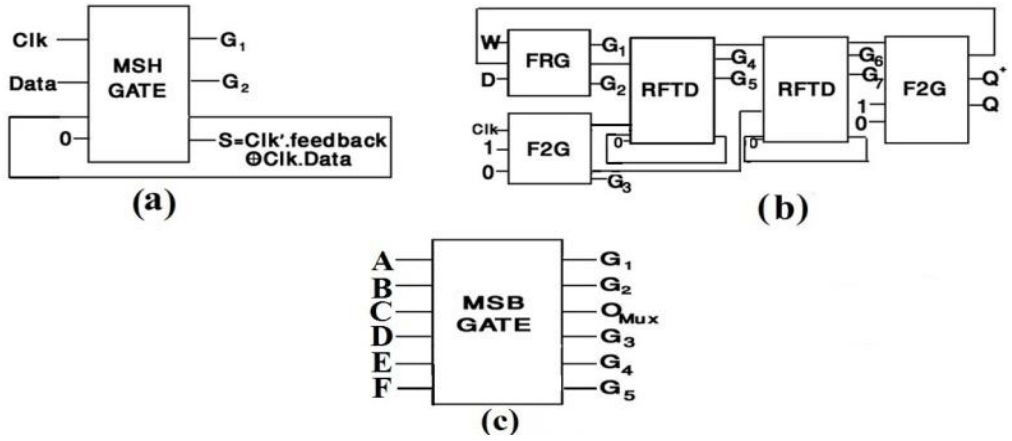


Fig. 13. The developed RFT block diagram includes (a) D-latch, (b) master slave flip-flop, (c) 4×1 multiplexer

4.3.1. D-latch

The D-latch is a crucial memory component in digital circuitry that enables data to flow from input to output in response to a clock signal. The D-latch stores the logical state on the data line as the clock input rises. Recent studies show that all D-latches, irrespective of design style, typically result in two superfluous outputs and a quantum cost of 6. After careful examination and investigation, we have shown that although the unnecessary output of a D-latch cannot be changed, its quantum cost may be reduced to 4 by including Feynman gates. The result of this modification is an overall quantum cost of 4(2×2), with a quantum cost of 2 per latch.

Let D represent the data input, the clock signal clk, and Q the D-latch's output. The following equation expresses the D-latch's distinctive operation:

$$Q^+ = D \cdot clk + clk' \cdot Q \quad (9)$$

$Q^+ = D$ is the result of the data input D propagating to the output when the clock (clk) takes on a value of 1. Conversely, when $clk = 0$, the latch preserves its preceding state, thus resulting in $Q^+ = Q$.

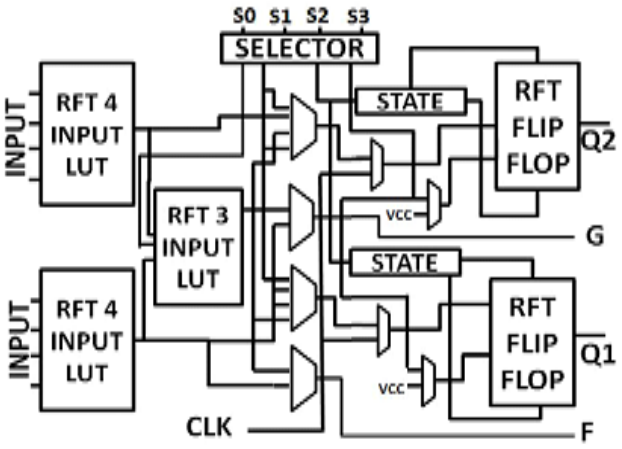


Fig. 14. Block schematic of the suggested fault-tolerant CLB

Algorithm 1: Design of an n-input Reversible Fault Tolerant (RFT) LUT based Configurable Logic Block

Input : Input from I/O block
Output: $Omux$, which will carry to next logic block

begin
  while *I/O Block is not empty* **do**
    $[S_0, S_1, .., S_n]$:= selection bits from I/O Block;
    Apply 4-input RFT LUT where Input:={ $[S_0, S_1, .., S_n]$}; Output:={Q_0};
    Apply RFT Flip Flop where Input:={ Q_0, clock signal}; Output:={Q or Q^+};
    Apply RFT 2×1 Mux where Input:={ Q_0 , Q or $Q+$}; Output:={ O_{mux}};
    $SelectionBit_{RFTMUX}$:={$Control_{bit}$};
  end while
end

4.4. FPGA-based Approach for Constructing an n-bit Adder Circuit

Employing the DSCH modeling tool, this research effort developed a complete adder circuit employing our suggested RFT-CLB architecture, as illustrated in Fig. 15. In this configuration, Block 1 estimates the carry (AB + AC + BC) while Block 2 calculates the total (A ⊕ B ⊕ C) of the three input bits (A, B, and C). Our design readily incorporates full adder logic, allowing for the assessment of both delay and power usage. Moreover, this implementation's flexibility allows for scalability, which makes it useful for building n-bit adders, as seen in Fig. 16.

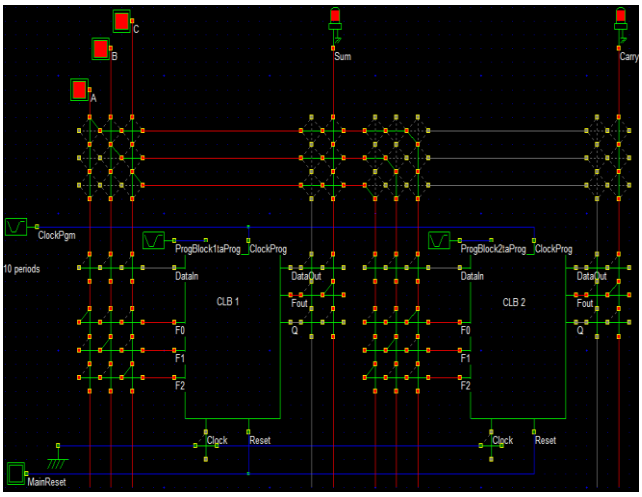


Fig. 15. Design of n-BIT Adder Circuit

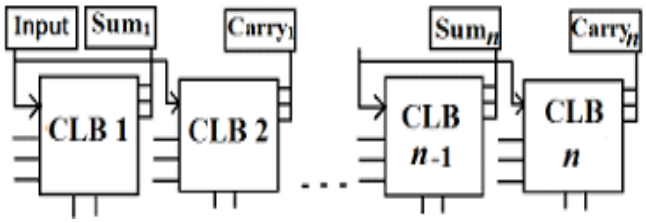


Fig. 16. An n-bit adder circuit's generalized block diagram

A computing unit called a full adder is able to add up three inputs and produce two outputs. initial and foremost, the initial two inputs, A and B, are joined by a third input, C, which is the input carry. As a result, the conventional output—which is the total of the inputs—is referred to as Sum, whereas the output carry is designated as Carry. Interestingly, the logic of a Full Adder is designed to accept eight inputs at once, which makes it easier to build a byte-wide adder and allows the carry bit to spread across adders. Table 3 offers an extensive illustration of the truth table that corresponds to the Adder circuit for the sake of clarity.

5. Results and Discussions

5.1. Results

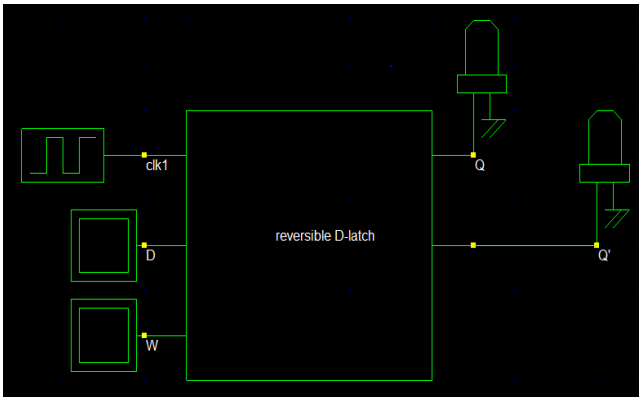


Fig. 17. Design of Reversible D-Latch

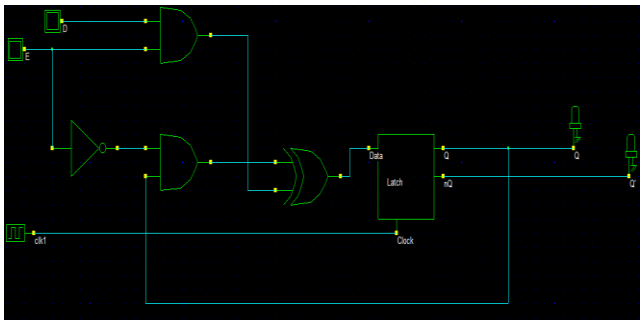


Fig. 18. Implementation of D-Latch

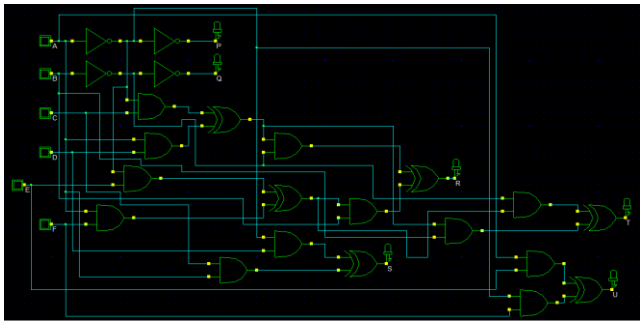


Fig. 19. Implementation of Master Slave SR Flip-flop

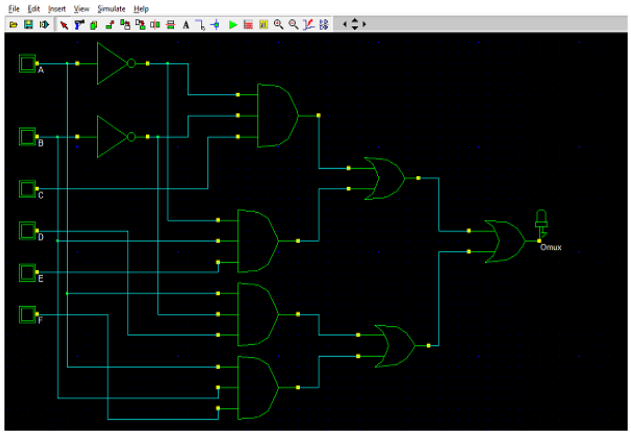


Fig. 20. Implementation of 4 × 1 multiplexer

Then select the Run simulation option

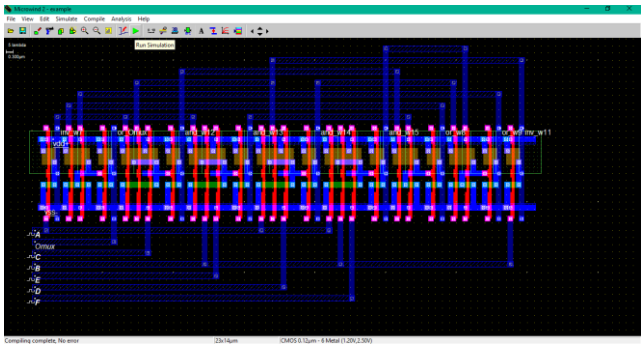


Fig. 21. Timing diagram

5.2. Performance Metrics

Four main criteria are used to assess the performance of reversible FPGA and act as the foundational benchmarks for our investigation. Below is a brief synopsis of these measurements.

- Gate Count: Reversible gates are essential in determining how a particular circuit is designed. By aggregating all utilized gates and determining their total count, we establish a significant performance indicator for reversible circuits. A lower count of required gates underscores the efficiency and high performance of the circuit design.
- Unwanted Outputs: To calculate unwanted outputs, one must count the number of output channels in a circuit that are not being used. This total number of unwanted outputs provides information about the design's overall effectiveness. Reducing the quantity of trash outputs is a symptom of an efficient and organized design process.
- Quantum Cost: One of the main performance metrics used to assess reversible circuits is quantum cost. It is the total number of NOT, Controlled-V, Controlled-V +, and CNOT gates required in order for a reversible gate to be executed.
- Delay: For delay calculations in reversible circuits, the reversible gates along the crucial route are substituted by their corresponding quantum counterparts. The total delay across the critical route determines the overall latency of the circuit. In this instance, the normalized unit-delay, Δ, represents the delay of each 1×1 or 2×2 reversible gate.

5.3. Performance Analysis

This section provides a performance study of the various LUT Logic Block components.

Table 3. Comparison of Output in Fault-Tolerant Master-Slave Flip-Flops

Methods	Gates	Quantum Cost	Unit Delay	Unwanted Output	No. of Transistors
Previous [3]	7	25	6	9	60

Previous [4]	5	21	5	7	38
Proposed 90 nm	4	10	4	2	6
Proposed 45 nm	4	8	2.3	2	6

Table 4. FT 4×1 Multiplexer Output comparisons

Methods	Gates	Quantum Cost	Unit Delay	Unwanted Output	No. of Transistors
Previous [3]	9	57	7	11	70
Previous [4]	1	12	1	5	12
Proposed 90 nm	1	5	1	3	5
Proposed 45nm	1	4	0.68	3	5

Table 5. Comparison of Outputs in Fault-Tolerant D-Latches

Methods	Gates	Quantum Cost	Unit Delay	Unwanted Output	No. of Transistors
Previous [3]	2	9	2	7	19
Previous [4]	1	6	1	4	10
Proposed 90 nm	1	2	1	1	3
Proposed 45 nm	1	1	0.68	1	3

Table 6. Comparative Analysis of Outputs in 4-i/p LUT-FPGAs

Methods	Gates	Quantum Cost	Unit Delay	Unwanted Output	No. of Transistors
Previous [3]	39	243	39	50	342
Previous [4]	5	60	5	19	60
Proposed 90 nm	5	56	5	5	37
Proposed 45 nm	5	51	4	5	37

Table 7. Comparative Assessment of CLB Performance

Methods	Gates	Quantum Cost	Unit Delay	Unwanted Output	No. of Transistors
Previous [3]	44	208	44	57	424
Previous [4]	18	170	18	57	192
Proposed 90 nm	12	156	15	32	112
Proposed 45 nm	12	156	15	32	112

A thorough comparison of the suggested design and previous designs is given in Tables 3–7, which show how the latter performed better in terms of latency, quantum cost, garbage output, gate count, and transistor count, among other metrics.

The settings of the FPGA are determined by these two variables, n and m. To shed light on the overall findings, we present the verdicts for three distinct methods: (i) altering m while preserving n, (ii) altering n whereas preserving m, and (iii) altering both m and n simultaneously.

See Table 8 for a comprehensive overview of the RFT n-BIT ADDER's performance evaluation.

Table 8. Comparative Analysis of n-Bit Adder

Adder	Power (mw)				
	1-bit	**4-bit**	**16-bit**	**64-bit**	**n-bit**
Existing	0.0082	0.0328	0.1312	0.5248	0.0082n
Proposed 90 nm	0.0025	0.0128	0.0512	0.2048	0.0032n
Proposed 45 nm	0.0010	0.0078	0.0241	0.1859	0.0016n
Adder	**Delay(ns)**				
	1-bit	**4-bit**	**16-bit**	**64-bit**	**n-bit**
Existing	77.88	311.52	1246.08	4984.32	77.88n
Proposed 90 nm	59.53	238.12	952.48	3809.92	59.53n
Proposed 45 nm	38.41	174.34	571.69	1764.60	31.17n

Figure 22 shows a graphical depiction of the power-delay product analysis for the current and proposed Adder designs. This graphic offers insightful information about the enhancements made possible by our suggested design.

Moreover, Table 8 displays a comparison study that emphasizes how much better our solution is in terms of both delay performance and power usage.

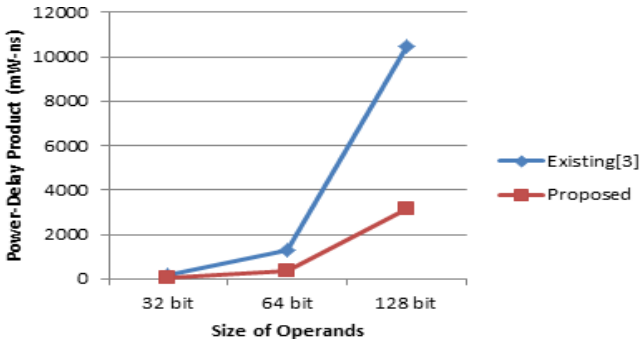


Fig. 22. Power-Delay Product Graph for RFT Adder

6. Conclusion

Reversible fault-tolerant CLB-based FPGA is a significant advancement in the area. These reversible CLBs are essential parts of FPGA designs and serve a crucial function. We have focused our research efforts on the thorough verification of the functionality built into the Reversible Fault-Tolerant (RFT) CLB. This component contains a 4:1 Multiplexer, D-Latches, Master-Slave Flip-Flops, 3-input and 4-input LUTs, and other parts.

Compared to traditional approaches, we have made substantial improvements to our RFT CLB and its component elements. Specifically, we have improved the design to minimize the quantity of transistors needed, reduce the number of unwanted outputs, minimizing the quantum cost, and shorten the gate count. These advancements are attainable because of the application of our unique reversible gate design.

In addition, the previously mentioned design's power consumption and hardware complexity are reduced when the recommended reversible gate is used. This decrease is especially noticeable when considering library technologies at 45 and 90 nm. This kind of advancement shows promise for more efficient and energy-efficient FPGA applications in modern semiconductor production processes.

Acknowledgements

The authors would like to thank BGS Institute of Technology, Adichunchanagiri University, B.G. Nagara, Mandya, for all of their support and encouragement in doing this study and releasing this manuscript.

Author contributions

The study's conception and design, experimental protocol supervision, data analysis, and paper composition fall within the purview of the lead author, Ravi L.S. He produced graphical representations, carried out data collection and analysis with skill, and made significant contributions to the writing of the text. He meticulously revised the text, provided insightful guidance during data interpretation, and actively participated in the study design. The manuscript was critically evaluated by the research supervisor, Naveen K.B.

Conflicts of interest

The authors declare no conflict of interest.

References

[1] Glück, R. and Yokoyama, T., 2023. Reversible computing from a programming language perspective. *Theoretical Computer Science*, *953*, p.113429.

[2] Bhattacharya, S. and Sen, A., 2021. A Review on Reversible Computing and it's applications on combinational circuits. *International Journal*, *9*(6).

[3] Shamsujjoha, M., Babu, H.M.H. and Jamal, L., 2013. Design of a compact reversible fault tolerant field programmable gate array: A novel approach in reversible logic synthesis. *Microelectronics journal*, *44*(6), pp.519-537.

[4] Haque, M.U., Sworna, Z.T. and Babu, H.M.H., 2016, January. An improved design of a reversible fault tolerant lut-based fpga. In *2016 29th International Conference on VLSI Design and 2016 15th*

International Conference on Embedded Systems (VLSID) (pp. 445-450). IEEE.

[5] Vidhya, N., Seethalakshmi, V., Arul, A., Dhanasekar, J. and Vinodhini, N., 2023, September. Era of reversible logic based quantum computing on low power applications. In *AIP Conference Proceedings* (Vol. 2764, No. 1). AIP Publishing.

[6] Proesmans, K., Ehrich, J. and Bechhoefer, J., 2020. Finite-time Landauer principle. *Physical Review Letters*, *125*(10), p.100602.

[7] Frank, M.P., 2020. *Fundamental Physics of Reversible Computing--An Introduction* (No. SAND2020-9778C). Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

[8] Chen, Z., Leugering, J., Cauwenberghs, G. and Chakrabartty, S., 2024. Energy-efficiency Limits on Training AI Systems using Learning-in-Memory. *arXiv preprint arXiv:2402.14878*.

Moore, M., 2020. Devices and systems™.

[9] Ravi L. S. and Naveen K. B., "Efficient Design of Configurable Logic Block with Customized LUT using Reversible Fault Tolerant Gates," International Journal of Intelligent Systems and Applications in Engineering (IJISAE), vol. 12, no. 16s, pp. 217-226, 2024.

[10] Naz, S.F. and Shah, A.P., 2023. Reversible gates: A paradigm shift in computing. *IEEE Open Journal of Circuits and Systems*.

[11] Noorallahzadeh, M., Mosleh, M., Ahmadpour, S.S., Pal, J. and Sen, B., 2023. A new design of parity preserving reversible Vedic multiplier targeting emerging quantum circuits. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, *36*(5), p.e3089.

[12] Singh, R. and Sharma, D.K., 2020. Fault tolerant reversible gate based sequential quantum dot cellular automata circuits: design and contemplation. *Journal of Nanoelectronics and Optoelectronics*, *15*(3), pp.331-344.

[13] Ahmadpour, S.S., Navimipour, N.J., Mosleh, M. and Yalcin, S., 2023. Nano-design of ultra-efficient reversible block based on quantum-dot cellular automata. *Frontiers of Information Technology & Electronic Engineering*, *24*(3), pp.447-456.

[14] Heikalabad, S.R., 2024. Reversible priority encoder in quantum-dot cellular automata. *Inorganic Chemistry Communications*, p.112401.

[15] Sozzo, E.D., Conficconi, D., Zeni, A., Salaris, M., Sciuto, D. and Santambrogio, M.D., 2022. Pushing the level of abstraction of digital system design: A survey on how to program fpgas. *ACM Computing Surveys*, *55*(5), pp.1-48.

[16] Rai, S., Nath, P., Rupani, A., Vishvakarma, S.K. and Kumar, A., 2021. A survey of FPGA logic cell designs in the light of emerging technologies. *IEEE Access*, *9*, pp.91564-91574.

[17] Tan, B.L., Mok, K.M., Chang, J.J., Lee, W.K. and Hwang, S.O., 2021. RISC32-LP: Low-power FPGA-based IoT sensor nodes with energy reduction program analyzer. *IEEE Internet of Things Journal*, *9*(6), pp.4214-4228.

[18] Haque, M.U., Sworna, Z.T. and Babu, H.M.H., 2016, January. An improved design of a reversible fault tolerant lut-based fpga. In *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)* (pp. 445-450). IEEE.

[19] Venkata, S.K., Roy, B.K. and Mohanty, P., 2022. Field programmable gate array implementation of an intelligent soft calibration technique for linear variable differential transformers. *F1000Research*, *11*, p.370.

[20] Xie, L., Du Nguyen, H.A., Taouil, M., Hamdioui, S., Bertels, K. and Alfailakawi, M., 2022, December. Non-volatile look-up table based FPGA implementations. In *2022 11th International Design & Test Symposium (IDT)* (pp. 165-170). IEEE.

[21] Li, Y., Zhang, Y., Liu, J., Gong, J., Wang, J., Lai, J., Tao, X. and Qu, G., 2022. AutoTEA: An Automated Transistor-level Efficient and Accurate design tool for FPGA design. *Integration*, *87*, pp.231-240.

[22] Gao, C., Liu, H., Wang, X. and Huang, J., 2023, October. Configuration circuit design for burn-in test of CLB based on structure. In *2023 Prognostics and System Health Management Conference (PHM)* (pp. 1-5). IEEE.

[23] Zhang, Y., Zhu, C., Cheng, X. and Xie, G., 2022. Design and Implementation of SRAM for LUT and CLB Using Clocking Mechanism in Quantum-Dot Cellular Automata. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *69*(9), pp.3909-3913.

[24] Seyedi, S., Otsuki, A. and Navimipour, N.J., 2021. A new cost-efficient design of a reversible gate based on a nano-scale quantum-dot cellular automata technology. *Electronics*, *10*(15), p.1806.

[25] Langenberg, B., Pham, H. and Steinwandt, R., 2020. Reducing the cost of implementing the advanced encryption standard as a quantum circuit. *IEEE Transactions on Quantum Engineering*, *1*, pp.1-12.

[26] Nash, B., Gheorghiu, V. and Mosca, M., 2020. Quantum circuit optimizations for NISQ architectures. *Quantum Science and Technology*, *5*(2), p.025010.

[27] Patidar, M., Kumar, D.A., William, P., Loganathan, G.B., Billah, A.M. and Manikandan, G., 2024. Optimized design and investigation of novel reversible toffoli and peres gates using QCA techniques. *Measurement: Sensors*, *32*, p.101036.

[28] Polash, M.M.A. and Sultana, S., 2020. Design of a LUT-based reversible field programmable gate array. *Journal of computing*, *2*(10), pp.103-108.

[29] Shamsujjoha, M., Babu, H.M.H. and Jamal, L., 2023. Design of a compact reversible fault tolerant field programmable gate array: A novel approach in reversible logic synthesis. *Microelectronics journal*, *44*(6), pp.519-537.

[30] Tara, N., Babu, H.M.H. and Jamal, L., 2022. Power efficient optimum design of the reversible plessey logic block of a field-programmable gate array. *Sustainable Computing: Informatics and Systems*, *16*, pp.76-92.

[31] Samrin, S.S., Patil, R., Itagi, S., Chetti, S.C. and Tasneem, A., 2022. Design of logic gates using reversible gates with reduced quantum cost. *Global Transitions Proceedings*, *3*(1), pp.136-141.