Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 11, No. 3, September 2023, pp. 621~629
ISSN: 2089-3272, DOI: 10.52549/ijeei.v11i3.4367

Optimized Reversible Logic Multiplexer Designs for Energy-Efficient Nanoscale Computing

C.Vijesh Joe[1], Haewon Byeon[2], Anand Kumar Singh[3], C. Ramesh Kumar[4], Aaquil Bunglowala[5], Anu Tonk[6]

[1]Assistant Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore Campus, Vellore, India.
[2]Department of AI-Big Data, College of AI Convergence, Inje University, Gimhae, 50834, Republic of Korea
[3]Assistant Professor, Manav Rachna International Institute of Research and Studies (MRIIRS), Sector-43, Aravali Hills, Faridabad, Haryana-121001, India.
[4]Professor, Applied Mathematics, Rungta College of Engineering & Technology, Rungta Educational Campus, Kurud Rd, Kohka, Bhilai, Chhattisgarh 490024, India.
[5]Professor, STME NMIMS Indore, Super Corridor NMIMS, Indore, MP, India.
[6] Assistant Professor -Sr Scale, Dept. of Multidisciplinary Engineering, The NorthCap University, Sector 23 A, Huda, Gurugram, Haryana - 122017, India.

Article Info

Article history:

Received Dec 4, 2022
Revised Jun 17, 2023
Accepted Aug 6, 2023

Keyword:

Quantum computing
Reversible logic gates
Garbage output
Ancilla inputs
TwinSJ gate
AJ gate
SJ gate

ABSTRACT

Nano- and quantum-based low-power applications are where reversible logic really shines. By using digitally equivalent circuits with reversible logic gates, energy savings may be achieved. Reducing garbage output and ancilla inputs is a primary emphasis of this study, which aims to lower power consumption in reversible multiplexers. Multiplexers with switchable 2:1, 4:1, and 8:1 ratios may be built using the SJ gate and other simple reversible logic gates. The number of ancilla inputs has been cut in half from four to zero, and the amount of garbage output has been cut in half as well, from eight to three, making the 2:1 multiplexer an improvement over the prior design. New 4:1 multiplexer has 10' ancilla inputs, up from 2' in the previous designs. The proposed 4:1 multiplexer also cuts waste production in half from the current 5-to-6 bins per day. The 8:1 multiplexer has two ancilla inputs and nine trash outputs, while the current architecture only has one of each. The functionality of the VHDL and Xilinx 14.7-coded designs is validated by ISIM simulations.

Corresponding Author:

Haewon Byeon,
Department of AI-Big Data,
College of AI Convergence, Inje University,
Gimhae, 50834, Republic of Korea
Email: bhwpuma@naver.com

1. INTRODUCTION

Reversible logic has gained a lot of interest in recent years due to its potential to provide low-loss or virtually lossless digital circuits. One thing that catches people's attention is the connection between high energy use and data loss. For classical systems, Rolf Landauer proposed [1] that information loss results in energy waste proportional to KTln2, where KT is Boltzmann's constant and T is the temperature. As shown in [2], reversible logic may be used to develop low-power circuits by obviating the heat dissipation often associated with data corruption. Nanotechnology, such as Quantum-dot Cellular Automata (QCA), has the potential to significantly reduce power consumption in contrast to CMOS [3, 4]. Garbage outputs, high gate counts, high quantum costs, and extra inputs are only some of the problems of reversible logic. Researchers have concentrated on designing solutions for particular usage in an effort to lower these costs [5]. This may be achieved by reducing the number of inputs and outputs in reversible logic gate VLSI circuits that are not being

used [6]. The advent of the pandemic has magnified the importance of some of the newer applications of reversible logic. Some of them are:

Quantum computers can't function without reversible logic. If quantum states are to remain in a reversible form, quantum gates must have an inverse operation. The creation and implementation of quantum algorithms and devices rely heavily on reversible logic. Reversible logic offers a viable answer to the rising challenge of energy efficiency in computer systems. It is utilised in the construction of low-power digital circuits such as arithmetic and computing circuits like adders and multipliers. Reversible logic reduces power dissipation by cutting down on heat loss and increasing energy efficiency. Reversible logic is essential in security and cryptography processes. Reversible cryptographic circuits, which are built using reversible gates, provide enhanced security features including untraceability and the avoidance of information leakage [3].

In the emerging field of DNA computing, DNA molecules are used to process data. DNA-based technology may be used to create reversible logic gates, allowing DNA computers to do two-way computations without using extra power. One kind of error correction method that makes advantage of reversible logic is quantum error correcting codes. In order to fix faults in quantum or conventional data without adding new irreversible processes, error correction circuits use reversible gates. Reversible logic might be useful in neuromorphic computing, a branch of computing that attempts to emulate the brain's structure and operation. Synaptic actions and neural network computations may be implemented quickly and easily with the help of reversible gates. Since light is used to process information in optical computing, reversible logic is essential. Optical components may be utilised to construct reversible gates, leading to reversible optical circuits that can be put to use in fast, low-power computation [7]. The contributions of this article are given below,

- Using reversible logic in the architecture of IoT devices allows for more efficient communication while also reducing power consumption.
- For effective data processing, reversible logic circuits may be employed in remote sensing applications requiring contactless measurement.
- Low-power digital pens and other gadgets may be possible with the help of reversible circuitry in the future of digital ink technology.
- Online testing and networking: Online system testing and networking might benefit from the reduced power usage and increased data processing capabilities made possible by reversible logic.
- Reversible logic's adaptability has made it possible to create circuits that are both energy efficient and durable, enhancing the user experience and extending the battery life of wearable computing devices.

2. REVERSIBLE LOGIC

2.1. Reversible logic fundamentals

Each operation in reversible logic has the same number of inputs as outputs. On the other hand, conventional logic sometimes produces more results than it takes in. Reversible multiplexers are a kind of reversible circuit used to choose and transmit one of two inputs. Feynman, Double Feynman, Toffoli, Fredkin, Peres, TSG, and Sayem gates are all examples of eversible logic gates that may be used in reversible logic circuits [8]. A fundamental characteristic of reversible logic is the equalization of inputs and outputs in the circuit. To guarantee that the sum of the inputs is equal to the sum of the outputs, reversible logic circuits often use auxiliary inputs and garbage outputs. Ancilla inputs are optional inputs used to ensure the gate may be operated in both directions. Even if they are neither useful or essential to the gate's primary activity, trash outputs are required for reversibility [9].

Using reversible circuitry and cutting out unnecessary inputs and outputs may help save or even eliminate power consumption. Therefore, reversible logic circuits excel in situations where minimizing power consumption is crucial. In addition to reversible logic gates, combinational circuits like as decoders, D flip-flops, T flip-flops, latches, and RAM cells may be built with reduced garbage outputs and ancilla inputs. This improves the efficiency of these circuits without increasing their power consumption [10]. Low-power and low-loss computational systems rely heavily on reversible logic, especially in quantum computing and low-power digital circuit design [11]. Since, reversible circuits consume less energy, they might eventually replace traditional logic [12]. The key features and configuration options for reversible logic gates are outlined below.

i. Inputs and outputs of reversible gates always correspond one to one, thus signals may go in both directions.
ii. Since each input combination yields a distinct output combination, reversible gates are able to do exact calculations.
iii. In order to facilitate easier hardware implementation, reversible gates are designed, resulting in more compact and efficient circuits.
iv. Garbage outputs, or undesired signals, are created as a byproduct of gate operations and should be kept to a minimum. Reduced trash outputs are one way that reversible gates help to save energy.

v. Power consumption and circuit complexity may be decreased while the number of gates required to do a given calculation is decreased with the help of reversible logic circuits.
vi. A deterministic behavior is guaranteed by the constant input values ('0' or '1') that reversible gates expect.
vii. Fan-out is set to zero (vii) Fan-out refers to the number of outputs that are wired to a gate's actual output. In most cases, reversible gates have a fan-out value of zero, indicating that just a single output is being driven.
viii. By not allowing feedback or loopback connections, reversible gates help keep the input-output mapping constant.

2.2. Reversible logic gate parameters

i. Garbage output is the term given to signals generated by gates that are neither needed nor fully exploited. To reduce power consumption and circuit complexity, designs using reversible logic should prioritize reducing trash outputs.
ii. The complexity, power consumption, and performance of a reversible logic circuit are all affected by the total number of gates it uses. If you want to design efficient reversible circuits, you need to limit the number of gates you use.
iii. The difficulty of a reversible gate may be estimated using a metric called "Quantum Cost." The number of Toffoli or Fredkin gates, both of which are examples of basic reversible gates, is shown.
iv. A matrix called the gate matrix represents the reversible logic gate
v. The mapping between input and output states is described using matrix operations.

2.3. Reversible Logic gates

The paper explores the use of reversible logic gates to decrease power consumption in VLSI circuits while taking into account a number of monetary considerations. Unnecessary data inputs, high quantum costs, large gate counts, and meaningless data outputs are all instances [13].

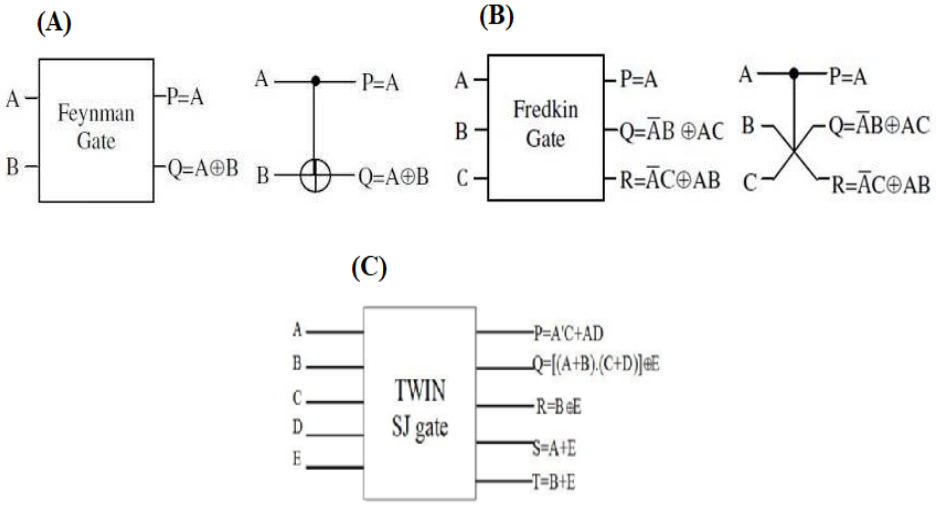

Figure 1. The block diagrams of the reversible gates: (a) the Feynman gate, (b) the Fredkin gate and (c) the TWIN SJ gate

The goal is to develop energy-efficient designs that can reduce costs. The primary goal of this study is to minimize the number of unnecessary inputs and outputs in very large scale integration (VLSI) circuits [15]. Garbage outputs, signals having no purpose, and ancilla inputs, used in reversible logic circuits but serving no vital function, may be produced via gates. The Feynman gate and the Fredkin gate, two reversible logic gates, will be examined to this aim. The Feynman gate cannot be used using the fan-out approach, which involves connecting several gates through a single output. In order to decrease fan-out and increase circuit efficiency, the Feynman gate is used in this research. The versatile Fredkin gate (shown in Figure 1 as a functional block diagram) is used in this investigation.

Table 1. Truth Table – SJ Gate

INPUT				OUTPUT			
A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	1	1	0	1	0	0	0
0	1	1	1	1	1	0	1
1	0	0	0	0	0	0	0
1	0	0	1	1	1	1	0
1	0	1	0	0	0	0	0
1	0	1	1	1	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	1	1	1	1
1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1

Among the common reversible gates built and used in the probed circuit is an SJ gate. With properly configured inputs, the SJ gate can carry out a variety of logical functions. The SJ gate acts as a 2:1 multiplexer in this circuit. The SJ gate functions as a multiplexer when input A is a logic "0" or "1," hence, it is sensitive to the value of input B. Input B or C is chosen and passed on to the output P depending on the value of A. P is the multiplexer's output and it is the input that was picked based on the value of A. In this setup, the SJ gate may function as a 2:1 multiplexer. The circuit's decision of whether to produce B or C as the output depends on the value of A, enabling a wide range of logical operations and signal-routing possibilities (Table 1).

i. P = A'C + AD: This is the OR function of A'C and AD. A'C is true when A is false and C is true. AD is true when A is true and D is true. Therefore, P is true when either A'C or AD is true.
ii. Q = D: This is a tautology, which means that it is always true.
iii. R = A.D: This is the AND function of A and D. A.D is true when both A and D are true. Therefore, R is true when both A and D are true.
iv. S = B.D: This is the AND function of B and D. B.D is true when both B and D are true. Therefore, S is true when both B and D are true.

3. REVERSIBLE LOGIC IMPLEMENTATION

Researchers have created a broad range of reversible multiplexers using reversible logic gates including the Feynman, Fredkin, BJN, and Peres gates. Due to the high gate count of these designs, GO and CI are more common than in others [16]. The 4:1 multiplexer is built from a special BJN gate, Feynman gates, and Peres gates. This method complicates production due to the number of gates that must be integrated. The Controlled Operation Gate (COG) is also created, which is a new kind of reversible gate that may be used as a 2:1 multiplexer. Both reversible and conventional multiplexers have been evaluated on cost metrics and gate count. However, the size of the chip is not necessarily proportional to the number of gates, since various gates might use varied amounts of space depending on the number of transistors employed. It is thought that quantum cost is a better comparative metric [17]. There has been study into the fabrication of Fredkin gates for usage as 2:1 multiplexers. In the past, reversible gates have been used to build a variety of circuits, including 2:1

multiplexers, D flip-flops, and PIPO shift registers. The proposed study optimises the design of a 2:1 multiplexer by decreasing the number of fixed inputs and unnecessary outputs. In another configuration, an 8:1 multiplexer is constructed from a pair of Fredkin gates functioning as 2:1 multiplexers and a 4:1 multiplexer. The writers are incorrect in claiming that there are only seven inputs and only two outputs. The proposed layout for the 8:1 multiplexer will lessen the amount of garbage it produces [18].

Several methods of building a 4:1 multiplexer are examined. One method employs three TKS gates, while another makes use of a VSMT gate. The second way is superior since it requires fewer reversible gates and garbage outputs. Further investigation results in a 4:1 multiplexer with eleven constant inputs, forty-three quantum cost, and sixteen garbage outputs [19]. The goal is to reduce the amount of constant resources used and unnecessary outputs. There are five pointless outputs and no useful inputs from the proposed procedure [20, 21]. The fundamental goal of this effort is to design efficient multiplexers that only take in the data that is absolutely essential. Multiplexers, sometimes called data selectors, take in a number of inputs and choose only one to send out. The project aims to employ the two ideas offered to develop multiplexers with more pristine input and output. The TWINSJ and AJ gates are used to build multiplexers in the first design. The creation of the 5x5 reversible TWINSJ gate was motivated by its potential use in multiplexers. Control signals allow one input to be chosen from several possibilities. As was previously mentioned, the AJ gate is a combination device capable of both AND and OR operations. It might be used to do logical operations and choose the correct input for multiplexers based on the control signals. In order to efficiently transport signals from numerous inputs to the required output in response to control signals, the multiplexers in Design1 make heavy use of the TWINSJ gate and the AJ gate. Effective and reversible multiplexer circuits may be built with the help of these gates.

The 'O' output of the multiplexer (Figure 2) is connected to the 'P' output of the TwinSJ gate. The output is based on either input (C or D), depending on the value of the selection bit (A). The gate's "B" and "E" inputs are supplementary features that may be utilised to modify its behaviour. TwinSJ gates often include three garbage outputs (called "G1," "G2," and "G3") that may produce unwanted or useless signals. Reversible logic design often reduces these wasteful outputs to cut down on power consumption and increase efficiency. If the TwinSJ gate is set up and utilized as a 2:1 multiplexer, then various input combinations may be sent to the output 'P' depending on the value of the selection bit 'A'. The gate features a special input-to-output pattern and, in reverse, has the same number of outputs as inputs.

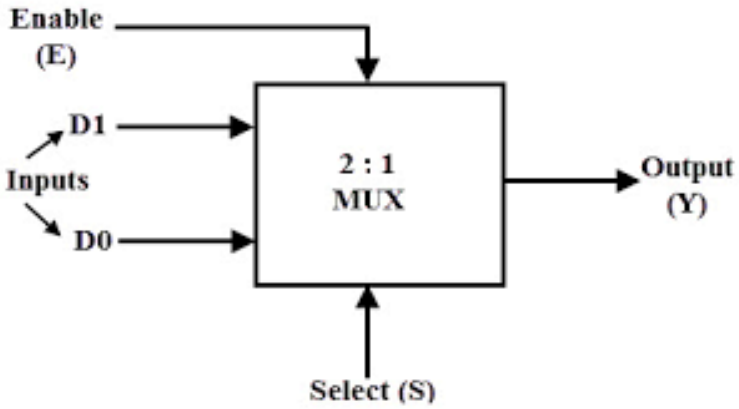

Figure 2. Multiplexer

The simulation results illustrate how the input selection and output routing is carried out based on the values of the selection bit and the input signals, revealing the predicted behaviour of the 2:1 multiplexer constructed using the TwinSJ gate. It is possible to use these simulations to check the functionality and efficiency of the constructed reversible multiplexer circuit. Electronic circuits known as multiplexers accept many inputs and selectively send one of them out an output based on the presence of control or select signals. There are four inputs (Vin1, Vin2, Vin3, and Vin4) and two select signals (S0 and S1) that determine which input is utilised. By linking together four 4:1 multiplexers, an 8:1 multiplexer may be built. All 4:1 multiplexers are constructed from a quartet of 2:1 multiplexers. The 2:1 multiplexer is built up of reversible Feynman gates. The copying function of the Feynman gate allows the input (A) to be accessible at both outputs when the other input (B) is zero. The 2:1 multiplexer makes advantage of this property. Select signals (S0 and S1) and input signals (Vin1, Vin2, Vin3, and Vin4) combine to form the multiplexer's output. The output may choose from four distinct input signals (Vin1, Vin2, Vin3, and Vin4) depending on the state of the select signals (00, 01, 10, and 11).

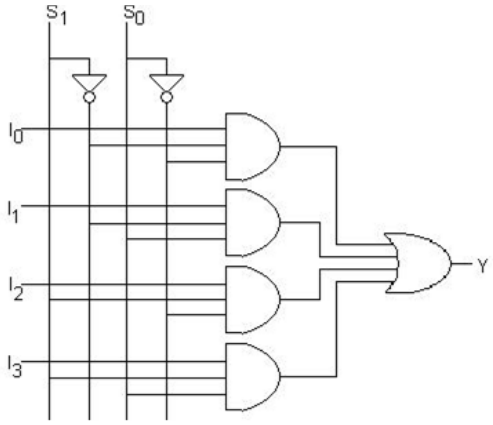

Figure 3. 4:1 Multiplexer using TWIN SJ gate

The 4:1 multiplexer (Figure 3) requires the use of three AJ gates and two TWINSJ gates for its construction. The AJ gate may function as either an AND or an OR gate, whereas the TWINSJ gate is a 2:1 multiplexer. Vin1 and Vin2 enter via TWINSJ gate 1, while Vin3 and Vin4 enter through TWINSJ gate 2. TWINSJ gate 1's output is governed by the E input. Select signals S0 and S1 may be accessed via the S and T outputs, respectively, when E is high ('1'). The third AJ gate performs logic operations on the intermediate data. AJ gate 3 represents the multiplexer's chosen output. The combined signals from the multiplexer's inputs (Vin1, Vin2, Vin3, and Vin4) and the two select signals (S0 and S1) are output at the Muxout terminal. This leads to the following expression:

$$Muxout=S1' [(S0'.Vin1 + S0.Vin2).S0'] S0'.Vin3 + S0.Vin4 + S0.S1 \quad (1)$$

As a function of the selected signals and the input values, this equation reflects the selected input signal. In this experiment, we build a TWINSJ gate, two 4:1 multiplexers, and an 8:1 multiplexer (Figure 4). When deciding which of eight inputs to send to the output, an 8:1 multiplexer takes into account the values of other selected signals. Two 4:1 multiplexers have been combined to create this 8:1 multiplexer. Each 4:1 multiplexer uses the input signals to choose one of four possible inputs. The construction process is further aided by the usage of a TWINSJ gate. It's possible for the TWINSJ gate to act as a 2:1 multiplexer in either way. One of the 4:1 multiplexer's outputs may be connected to the 8:1 multiplexer's final output. To clone a signal, one may use a FEYNMAN gate. When input B can be set to zero to replicate the signal. Both of these outputs may be used to connect to the second input (labelled "A" here).

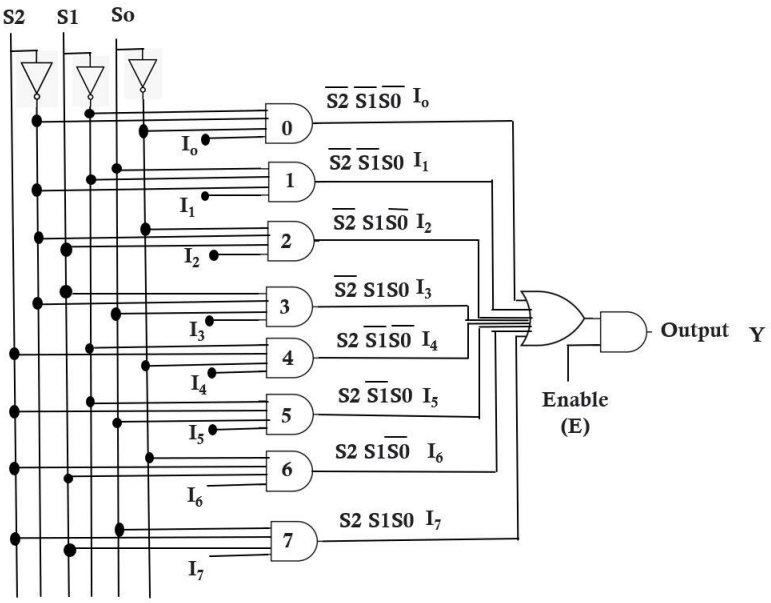

Figure 4. 8:1 Multiplexer

SJ gates and Fredkin gates are used to build multiplexer circuits in Design 2, with an emphasis on reducing the number of garbage outputs and ancillary inputs. The SJ gate, a reversible gate, is an integral part of multiplexer circuit design. By reducing the number of useless outputs and extraneous inputs, we may improve the circuit's efficiency and make it more energy-efficient. In Plan 2, the SJ gate is configured as a 2:1 multiplexer. The SJ gate with the 'P' output representing the maximum output. The equation for a multiplexer is P = A'C + AD, where 'A' is the selection bit and 'C' and 'D' are the inputs. The "C1" auxiliary input and the "G1," "G2," and "G3" garbage outputs round out the standard SJ gate arrangement. Ancilla inputs and trash outputs are essential to the proper functioning of reversible logic circuits. The three SJ gates used in the design form a 4:1 multiplexer. The output, or Muxout, of the 4:1 multiplexer is defined by the input signals (Vin1, Vin2, Vin3, and Vin4) and the select signals (S0 and S1). The resulting expression is as follows:

$$Muxout = [(S0'.Vin1+S0.Vin2).S1']+[(S0'.Vin3+S0.Vin4).S1] \quad (2)$$

With the use of the select signals S0 and S1, the output may be routed to one of the four input signals (Vin1, Vin2, Vin3, or Vin4) based on the binary combinations 00, 01, 10, and 11. Three SJ gates are used as building blocks in the circuit to create the 4:1 multiplexer. Each SJ gate's role in the larger multiplexer architecture is determined by its arrangement, and the SJ gate itself serves as a 2:1 multiplexer. The select signal S1 may be accessed through the SJ gate 1's Q output, making it a repeater. The inputs of SJ gate 1 are linked to Vin1 and Vin2.

The proposed 8:1 multiplexer makes use of both SJ gates and Fredkin gates. The multiplexer's inputs are numbered V_IN1 through V_IN8. The circuit's output may arbitrarily choose one of the input signals thanks to the employment of SJ gates and Fredkin gates. The usage of Feynman gates for signal recycling allows us to reduce the number of inputs and outputs that aren't essential. By employing the copy gate functionality of the Feynman gate, the select signals SEL0 and SEL1 may be utilised several times. Fredkin gates have a selection bit (SEL3) that determines which output (called the "muxoutput") is produced. The results from the SJ gates are sent into the Fredkin gate, which then uses them to determine its own output. The "muxoutput" is chosen among the outputs of the SJ gates by use of a Fredkin gate, with the SEL3 bit serving as the selection bit. The final output of the gate is determined by a Fredkin gate, which is connected to the outputs of the SJ gates and uses the selection signal SEL3 from the SJ gates.

The Muxout equation, which takes into account the select signals (S0, S1, and S2) and their associated input values (VIN1 to VIN8), expresses the process of signal input selection. Multiple permutations of the choice signals are used to zero in on the right input signal [10]. The architecture employs a total of seven gates: two Fredkin gates, one Feynman gate, and five SJ gates. In response to select signals, SJ gates (which also function as copying gates) make the actual selection. The Feynman gate conserves resources by recycling some signals, using fewer inputs and producing fewer outputs. The Fredkin gate uses the SEL3 select signal as the basis for its output decision. It has been discovered that the circuit may accept a variety of selection signal configurations, and that by watching the resultant signal, the proper multiplexer's functioning can be verified. The results of the verification procedure. There are twelve main outlets and one additional outlet for disposal use.

4. SIMULATION & RESULTS

All VHDL-coded circuits are simulated and tested in the Xilinx development environment to guarantee low quantities of garbage output and irrelevant input. In this circuit, the 'P' output of the SJ gate is wired to the muxoutput of the 2:1 multiplexer. If 'A' equals 0, then 'P' will choose 'B' as the output, and if not, then 'A' will be used instead. On the other hand, if 'A' equals 1, then 'P' will choose 'C' as the answer. varied input configurations of "A," "B," and "C" will likely result in varied behaviours from the multiplexer circuit. Using it during testing may assist ensure the circuit is functioning as expected. To provide further context to the results, Vin1 in the sample has an ON and OFF duration of 100ns, Vin2 has an ON and OFF period of 100ns, Vin3 has an ON and OFF length of 200ns, and Vin4 has an ON and OFF period of 40ns. The results of constant muxout and S0/S1 status checks. If S0 = 1 and S1 = 0, then Vin2 is formed, but if S0 = 0 and S1 = 1, then Vin3 is generated. When both S0 and S1 are 1, the terminal will make Vin4.

- Vin1: 100ns ON and 100ns OFF signal
- Vin2: 100ns ON and 20ns OFF signal
- Vin3: 20ns ON and 100ns OFF signal
- Vin4: 40ns ON and 200ns OFF signal

The values of the muxout, as well as the present states of the select signals S0 and S1, are shown at a number of different time periods. Vin2 is selected as the output signal during the time period in which S0 equals "1" and S1 equals "0." When S0 is set to '0' and S1 is set to '1,' Vin3 is the signal that is used as the output during the time period in question. Vin4 is selected as the output signal for the relevant time period when both S0 and S1 are equal to the value '1'. By comparing the select signal values with the accompanying muxout results, as shown in Table 2 and 3, we can see how the multiplexer circuit picks the input signal based on the select signal values and their temporal correlations. This is accomplished by comparing the select signal values with the results of the muxout.

- When SEL0, SEL1, and SEL2 are all 0, V_IN1 is selected at the output.
- When SEL0 is made '1' at 0.12 µsec, V_IN2 is selected at the output.
- When SEL0 is '0' and SEL1 is '1' at 2.5 µsec, V_IN3 is selected at the output.
- At 3.5 µsec, when SEL0 and SEL1 are both '1', V_IN4 is selected at the output.

Each conceivable combination of selection bits 100, 101, 110, and 111 is verified in the same way by observing and validating the relevant outputs. At 7.5 microseconds, with SEL0, SEL1, and SEL2 all set to 1, V_IN8 is selected as the output's input. It is probable that the results of the simulation which shows the temporal correlations between the selection bit values (SEL0, SEL1, and SEL2) and the corresponding output signals. Analysing these results will prove that the multiplexer circuit selects the correct input signal according to the given selection bit combinations.

Table 2. Comparison of designs

Multiplexer	Design 1		Design 2	
	Ancilla Input	Garbage Output	Ancilla Input	Garbage Output
2:1	3	6	1	4
4:1	1	8	1	6
8:1	5	15	3	12

Table 3. Comparison of outputs

Multiplexer	Existing design		Proposed Design		Improvement %	
	Ancilla Input	Garbage Output	Ancilla Input	Garbage Output	Ancilla Input	Garbage Output
2:1	4	8	0	3	100	62.5
4:1	2	9	1	6	50	33.33
8:1	4	12	2	9	50	25

Table 2 is a comparison of the two designs, and it is assumed that the main differences and enhancements are highlighted there. Measurements of garbage production and auxiliary input might be used to objectively assess the effectiveness and efficiency of each design. Many reversible multiplexers, both current and prospective, are compared in Table 3. Most of the time, it gives you trash to work with and some extra input parameters for each design, so you can compare and contrast how well they function. Each new concept includes an explanation of how it improves upon previous ideas, highlighting any reductions in waste production or auxiliary input.

5. CONCLUSION

Using the Reversible Twin SJ gate (Design1) and the SJ gate (Design2), the reversible multiplexers developed in this research significantly outperform prior designs in terms of trash outputs and ancilla inputs. The SJ gate-based second design is the most functional option. The suggested design uses the SJ gate to create a 2:1 multiplexer and a 4:1 multiplexer, both of which need less ancilla inputs ('0') than current systems. Additionally simplifying things is the 8:1 multiplexer's need for just a single ancilla input. The recommended designs also drastically cut down on garbage production. With just 3 garbage outputs, the 2:1 multiplexer architecture is a huge improvement over the prior design (62.5% improvement, to be precise). The wasted outputs are reduced from 10 to 6 using the 4:1 multiplexer. In addition, the 8:1 multiplexer architecture reduces garbage outputs by 25%, from 11 to 9. Improvements in garbage outputs and ancilla inputs are especially useful in the construction of low-power computers and future systems since multiplexers are required for CPU operations like choosing peripheral devices and accessing memory.

ACKNOWLEDGMENT

This research is supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF- RS-2023-00237287, NRF-2021S1A5A8062526) and local government-university cooperation-based regional innovation projects (2021RIS-003).

REFERENCES

[1] C. Tsai, "Reversible data hiding and lossless reconstruction of binary images using pair-wise logical computation mechanism," *Pattern Recognition*, vol. 38, no. 11, pp. 1993-2006, 2005.
[2] P. M. B. Vitanyi, "Logical depth for reversible Turing machines with an application to the rate of decrease in logical depth for general Turing machines," *Theoretical Computer Science*, vol. 778, pp. 78-80, 2019.
[3] J. C. Das and D. De " Computational fidelity in reversible quantum-dot cellular automata channel routing under thermal randomness," *Nano Communication Networks,* vol. 18, pp. 17-26, 2018.
[4] S. Mohammad, " Fault-tolerant quantum reversible full adder/subtractor: Design and implementation," *Optik*, vol. 253, No. 168543, 2022.
[5] A. Samson, " A structural approach to reversible computation," *Theoretical Computer Science*, vol. 347, pp. 441-464, 2005.
[6] F. M. Weidner, and J. D. Schwab, " Leveraging quantum computing for dynamic analyses of logical networks in systems biology*," Patterns*, vol. 4, No. 100705, 2023.
[7] S. S. Samrin, and R. Patil, " Design of logic gates using reversible gates with reduced quantum cost," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 136-141, 2022.
[8] M. Awais, and P. Maah "Energy Efficient Design of Fast Fourier Transform using Reversible Logic," *Sustainable Energy Technologies and Assessments,* vol. 52, No. 102257, 2022.
[9] C. Bertini, and R. Leporini, "A fuzzy approach to quantum logical computation," *Fuzzy Sets and Systems*, vol. 317, pp. 44-60, 2022.
[10] Z. Li, and S. Zhang, "An efficient three-level nano-design for reversible gate using quantum dot-cellular automata with cost analysis," *Materials Science and Engineering: B,* vol. 294, No. 116526, 2023.
[11] D. Mukhopadhyay, and P. Dutta, "A study on energy optimized 4 dot 2 electron two dimensional quantum dot cellular automata logical reversible flip-flops" *Microelectronics Journal*, vol. 46, pp. 519-530, 2015.
[12] M. M. Hossain, M. H. Rehman, A. Ahmed, and A. B. M. A. Rahman, "Design of low power reversible binary coded decimal adder," I*nternational Journal of Computer Science and Information Security*, vol. 12, no. 5, pp. 127-135, 2014.
[13] L. Singh, R. Saxena, and G. Zho, "A compact realization of Feynman Reversible and NOR logic gate using Plasmonic waveguide based MZI for all-optical signal processing," *Optics Communications*, vol. 522, No. 128707, 2022.
[14] F. Orts, G. Ortega, and E.F. Combarro, "A review on reversible quantum adders," *Journal of Network and Computer Applications,* vol. 170, N. 102810, 2020
[15] G. J. Martínez, and A. Adamatzky, "On logical gates in precipitating medium: Cellular automaton model," *Physics Letters A*, vol. 372, pp. 5115-5119, 2008.
[16] M. Tostado-Veliz, S. Kamel, and H. M. Hasanien, " A mixed-integer-linear-logical programming interval-based model for optimal scheduling of isolated microgrids with green hydrogen-based storage considering demand response," *Journal of Energy Storage,* vol. 48, No. 104028, 2022.
[17] Leverrier, K., M. J. Bremner, A. M. Childs, and M. B. Plenio. "Exploring trade-offs between physical realizability and optimal quantum circuit complexity*." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* vo.469.no.2159 (2012).
[18] J. P. Klein, and T. H. Leete, " A biomolecular implementation of logically reversible computation with minimal energy dissipation," *Biosystems,* vol. 52, pp. 15-23, 1999.
[19] B. Sen, M. Dutta, and M. Goswami, "Modular Design of testable reversible ALU by QCA multiplexer with increase in programmability," *Microelectronics Journal*, vol. 45, pp. 1522-1532, 2014.
[20] M. Li, and J. Tromp, "Reversible simulation of irreversible computation," *Physica D: Nonlinear Phenomena*, vol. 120, pp. 168-176, 1998.
[21] H. M. Gaur, and A. K. Singh, "In-depth Comparative Analysis of Reversible Gates for Designing Logic Circuits," *Procedia Computer Science*, vol. 125, pp. 810-817, 2018.