J Comput Electron (2015) 14:726–746
DOI 10.1007/s10825-015-0707-z

Optimized fault tolerant designs of the reversible barrel shifters using low power MOS transistors

Md. Shamsujjoha[1] · Fahmida Hossain[2] · Md. Nawab Yousuf Ali[1] · Hafiz Md. Hasan Babu[2]

Published online: 4 June 2015

Abstract This paper demonstrates the reversible logic synthesis for the unidirectional, bidirectional and universal barrel shifters. The proposed shifters are constructed using only Fredkin and Feynman double gates. Initially, these two gates are designed and schematized using standard low power p-MOS 901 and n-MOS 902 models with average channel length 45 nm, delay 0.030 ns and density 1200 kgates/mm^2. These schemas can detect faulty signal in its primary outputs. Thus, all the proposed shifters inherently tolerate single level fault. Moreover, the presented generalized algorithm for the proposed unidirectional barrel shifter has further been used to build base structures of the proposed bidirectional and universal shifters. In addition, lower bounds on the numbers of constant inputs and garbage outputs of the reversible barrel shifter have been proposed. It has been evidenced that the proposed circuits are constructed with these optimum constant inputs and garbage outputs. The simulation results prove the functional correctness of the proposed circuits. The comparative results show that the proposed designs perform much better and have significantly better scalability than the existing approaches.

✉ Md. Shamsujjoha
dishacse@yahoo.com

Fahmida Hossain
fahmidacsedu@gmail.com

Md. Nawab Yousuf Ali
nawab@ewubd.edu

Hafiz Md. Hasan Babu
hafizbabu@hotmail.com

[1] Department of Computer Science & Engineering, East West University, Dhaka 1212, Bangladesh

[2] Department of Computer Science & Engineering, University of Dhaka, Dhaka 1000, Bangladesh

Keywords Barrel shifter · Fault tolerant · Garbage output · Low power design · Reversible logic

1 Introduction

Connection between the computational limit and entropy was pointed out by Maxwell in [1] and Szilard in [2], which entail that "$KTln(N)$ joules of energy must be dissipated during every computation, where K is Boltzmann's constant of 1.38×10^{23} J K^{-1} and T is the absolute temperature of the environment in kelvin and N is an integer proportional to the number of computed bit". Later Landauer proved that the reversible system can reduce these energy dissipations [3]. In [4], Bennett showed that the reversible circuits dissipate $KT\ ln(1)$ energy, which is much lower than the corresponding irreversible circuits. An irreversible system can store the information which is produced during a computation rather than erasing, but it doesn't always provide a unique path from each current state to its previous state [5]. Energy used to store these information is unrecoverable [6]. It has also been pointed out that an irreversible system has a fundamental lower limit to the energy dissipation during a computation which equals to $KT\ ln(2)$ for each erased bit. Although the power dissipations due to information loss in irreversible circuits are negligible at present, but it will be a significant concern in near future if Moore's Law continues to be in effect [7]. Moore predicted in [8] that "the number of transistors and resistors on a chip doubles in every 18 months". This assumption implies that the power dissipation in the circuit must be decreased, otherwise internal overheating can demolish the future hardware chips [5,7]. Therefore, the reversible circuit will play an extensively crucial role in the upcoming days. Moreover, a reversible circuit is viewed as a special quantum circuit, as the quantum evolution must be reversible [9].

In addition, fault tolerant reversible circuits can detect faulty signal in the primary outputs through parity checking. Parity checking has widely been used to detect errors in the storage hardware or transmission of information because most arithmetic and other processing functions don't consider to preserve the parity of data [5]. If the parity of the input data is maintained throughout the computation, then the intermediate checking wouldn't be required [10]. As a result, an entire circuit can preserve parity if its individual gate is parity preserving [11]. Over the past few years, both reversible and fault tolerant circuitry gained remarkable interest in the field of DNA-technology [12], nano-technology [13], optical computing [14], program debugging and testing [15], discrete event simulation [5], modeling of biochemical systems [16] and in the development of highly efficient algorithms [17].

On the other hand, barrel shifter is the in-built component of many computing systems as it can shift multiple bits in a single cycle and hence it attains much importance in designing the processors [5], low-density parity-check decoders [6], optical networks [18] and test generation [19]. Among the various barrel shifters, logarithmic shifter has the simplest structure which is also more area efficient as it doesn't require any underneath decoder circuitry [20]. In these consequences, this paper proposes the implementation techniques of the reversible fault tolerant barrel shifters using nano-meter MOS transistors. There are few non-fault tolerant reversible designs of barrel shifters in literature [21–23]. However, all these approaches are not generalized and scalable except our proposed designs [7]. Thus, the main objective of this research is to develop a generalized structure of the reversible barrel shifters in addition to the fault detection capabilities. The MOS realization of the proposed reversible fault tolerant circuits are considered because of its scalability [5,6,24,25]. The rest of the paper is organized as follows. Section 2 summarizes necessary backgrounds and related works. Then we present a generalized synthesis of the proposed reversible fault tolerant unidirectional logarithmic barrel shifters in Sect. 3. We already published a part of this mentioned work in [7]. This section proposes few extension of the published work such as MOS realization and generalized algorithmic representation. Section 4 presents the hierarchical design of the bidirectional barrel shifter based on the proposed unidirectional shifter which proves the scalability of the proposed extended unidirectional design. We also present the design methodologies of the proposed universal shifter in Sect. 5. The performances of all the proposed designs are evaluated in Sect. 6. Finally, based on all these discussions, a conclusion is drawn in Sect. 7.

2 Basic definitions and literature review

This section starts with the basic definitions of the reversible and fault tolerant logic synthesis. Then we briefly define reversible gates in addition to their fundamental properties. Transistor realizations of the popular reversible gates using standard low power p-MOS 901 and n-MOS 902 models have also been shown. Finally, the section ends with a discussion on the existing works.

2.1 Reversible and fault tolerant gates

An $n \times n$ *reversible gate* is shown in Fig. 1. This block uniquely maps between input vector $I_v = (I_0, I_1, ..., I_{n-1})$ and output vector $O_v = (O_0, O_1, \ldots, O_{n-1})$ denoted as $I_v \leftrightarrow O_v$ [5]. The unique mapping (one-to-one mapping) among all inputs and outputs for each input-output sequence results no information loss. There are two prime requirements for the reversible logic synthesis which are given below:

- There should be equal number of inputs and outputs.
- There should be one-to-one correspondence among all input-output sequences.

A *Fault tolerant gate* is a reversible gate that constantly preserves the same parity between inputs and outputs. More specifically, an $n \times n$ fault tolerant gate clarifies the following property between the input and output vectors:

$$I_0 \oplus I_1 \oplus ... \oplus I_{n-1} = O_0 \oplus O_1 \oplus ... \oplus O_{n-1} \quad (1)$$

The parity preserving property of Eq. 1 allows to detect a faulty signal from the circuit's primary output. Researchers showed that if a reversible circuit is drawn using only reversible fault tolerant gates, the entire circuit preserves parity and hence it is able to detect a faulty signal in its primary outputs [5–7].

2.2 Quantum cost

The *quantum cost* for all 1 × 1 and 2 × 2 reversible gates are considered as 0 and 1, respectively [5–7]. Hence, the quantum cost of a reversible circuit is the total number of 2 × 2 quantum gates. In other words, numbers of controlled-V (square root of NOT, i.e., SRN), controlled-V^+ (hermitian

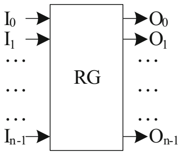


Fig. 1 Block diagram of an $n \times n$ reversible gate

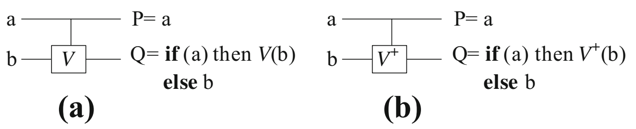


Fig. 2 Block diagram of the controlled **a** V gate **b** V^+ gate

of SRN) and 2 × 2 Ex-OR gates represent the quantum cost of the circuit.

A controlled-V gate is shown in Fig. 2a. In a controlled-V gate, when the control signal a = 0, the qubit b will pass through the controlled part and remain unchanged, i.e., we will have $Q = b$. If the value of a = 1, then $Q = V(b)$, where,

$$V = (i + 1)/2 \begin{pmatrix} 1 & -i \\ -i & -1 \end{pmatrix}$$

In the above matrix, i is the basic imaginary unit, i.e., $i = \sqrt{-1}$. A controlled-V^+ gate is shown in Fig. 2b. In a controlled-V^+ gate, when the control signal a = 0, the qubit b will pass through the controlled part and keep it unchanged, i.e., we will have $Q = b$. If the value of a = 1, then $Q = V^+(b)$, where V^+ is the hermitian of V ($V^+ = V^{-1}$, when a = 1). Thus, $V \times V$ or $V^+ \times V^+$ creates a unitary matrix of NOT gate, whereas, $V \times V^+$ or $V^+ \times V$ is an identity matrix (I) describing just a quantum wire.

2.3 Hardware complexity, PDP and garbage output

In a logic circuit or reversible logic circuit, the path consisting of maximum number of gates from any input to any output is known as critical path. However, it is an NP complete problem to find the critical path specially for large circuits [5,7]. Thus, researchers used to select the path which is the most likely candidates for the critical path, and *delay* of a logic circuit is the delay of this path. The critical path delay calculation has several other requirements such as, “each gate need to perform computation in unit time and all inputs to the circuit must be known before the computation begins”. Thus, the researchers determined the *hardware complexity* of the circuit which is considered as a seemingly equivalent performance evaluation criteria. The number of basic operations needed to realize the circuit is referred to as the hardware complexity of the circuit. Actually, a constant complexity is assumed for each basic operation of the circuit, such as, α for Ex-OR, β for AND, γ for NOT etc. Then, the total hardware complexity is calculated with respect to these assumed complexities. For example, the hardware complexity of Fig. 3 is α, since it can be realized with a 2×2 Ex-OR operation.

The power delay product (*PDP*) is a figure of merit that directly correlated with both the energy efficiency and delay of a circuit. The real value of PDP is power consumption times delay. However, generally PDP is presented in normalized form [26,27].

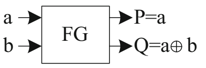


Fig. 3 Block diagram of a 2 × 2 reversible Feynman gate

Unwanted or unused output of a reversible gate is known as *garbage output*, i.e., the output which are needed only to maintain the reversibility are known as garbage output [5, 6]. For example, to perform the exclusive OR between two inputs, a 2 × 2 reversible Feynman gate (FG) can be used. This realization produces an extra dummy output along with the desired output signal, which is needed to preserve the reversibility. This extra output is the garbage output denoted by P as shown in Fig. 3. It is important to note that heavy price is paid off for each garbage output [28].

2.4 Reversible and fault tolerant gates

In this section we define more reversible and fault tolerant gates and their properties.

2.4.1 Feynman double gate

Input vector (I_v) and output vector (O_v) for 3 × 3 reversible Feynman double gate ($F2G$) are defined as follows: $I_v = (a, b, c)$ and $O_v = (a, a \oplus b, a \oplus c)$. Block diagram of $F2G$ is shown in Fig. 4a. Figure 4b represents the quantum equivalent realization of $F2G$. From Fig. 4b, we find that it is realized with two 2 × 2 Ex-OR gates i.e., its quantum cost is 2. According to our design procedure, 12 transistors are required to realize the Feynman double gate reversibly. This transistor realization is shown in Fig. 4c. Figure 4d represents the corresponding timing diagram of the transistor realization of the reversible Feynman double gate.

2.4.2 Fredkin gate

Input and output vectors for 3 × 3 Fredkin gate (FRG) are defined as follows: $I_v = (a, b, c)$ and $O_v = (a, a'b \oplus ac, a'c \oplus ab)$. Block diagram of FRG is shown in Fig. 5a. Figure 5b represents the quantum equivalent realization of FRG. In Fig. 5b, each rectangle is equivalent to a 2 × 2 quantum primitive. Therefore, its quantum cost is considered as one. Thus, the quantum cost of FRG is five. To realize the FRG, four transistors are needed as shown in Fig. 5c and its corresponding timing diagram is shown in Fig. 5d. Fredkin gate can also be used to swap 2nd and 3rd inputs using first input as a controlled input. Referring to the input-output combinations of Fredkin gate, when a = 1, the inputs b and c will be swapped. The resulting value of the outputs are $Q = c$ and $R = b$. Whereas, if a = 0, then outputs P,

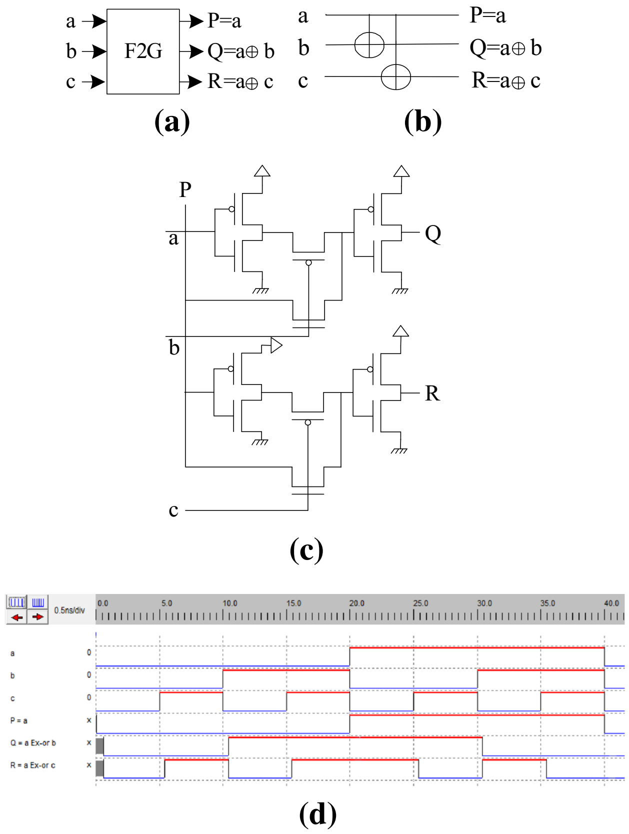


Fig. 4 Reversible 3 × 3 Feynman double gate. **a** Block diagram. **b** Quantum equivalent realization. **c** Transistor realization. **d** Timing diagram [6]

Q and R are directly connected to inputs $a(= 0)$, b, and c, respectively. The reversible gates discussed above maintain the property of Eq. 1 which is shown in Table 1.

2.5 Existing barrel shifters

Barrel shifter has n-input and n-output lines for data transmission and k control inputs, where k=log$_2 n$. An adaptive structure of the basic (n, k) unidirectional logarithmic barrel shifter is shown in Fig. 6. It has k ($k = log_2 n$) stages which are controlled by k control bits. Control bit S_j (j=0 to $k - 1$) of a stage determines whether to shift (or rotate) the input data or not for that stage. If S_j is set to high, then j^{th} stage will shift or rotate the input 2^j times, otherwise input will remain unchanged.

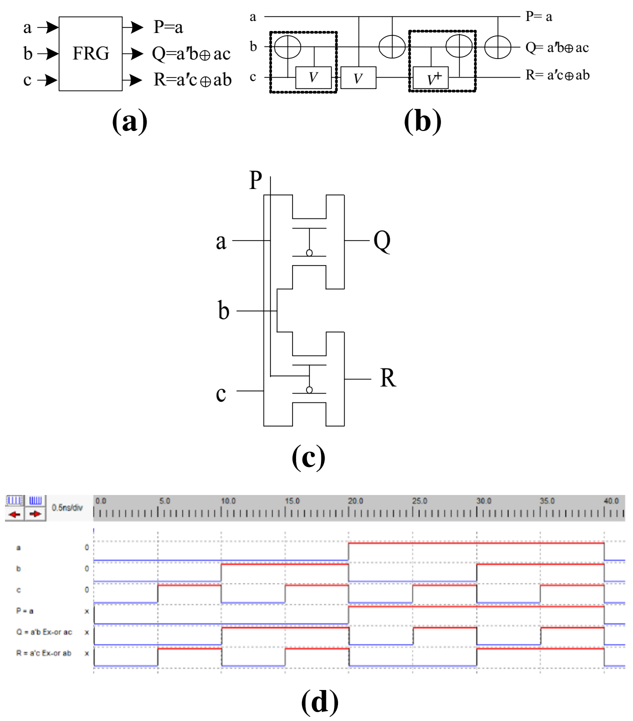


Fig. 5 Reversible 3 × 3 Fredkin gate. **a** Block diagram. **b** Quantum equivalent realization. **c** Transistor realization. **d** Timing diagram [6]

Table 1 Truth table for $F2G$ and FRG

Input			Output: $F2G$			Output: FRG			
A	B	C	P	Q	R	P	Q	R	Parity
0	0	0	0	0	0	0	0	0	Even
0	0	1	0	0	1	0	0	1	Odd
0	1	0	0	1	0	0	1	0	Odd
0	1	1	0	1	1	0	1	1	Even
1	0	0	1	1	1	1	0	0	Odd
1	0	1	1	1	0	1	1	0	Even
1	1	0	1	0	1	1	0	1	Even
1	1	1	1	0	0	1	1	1	Odd

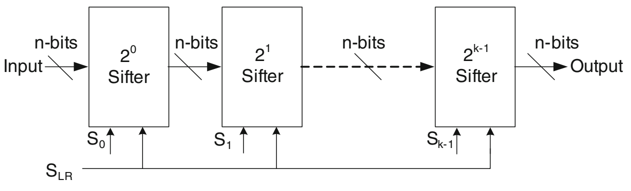


Fig. 6 Adaptive structure of (n, k) logarithmic barrel shifter

Gorgin et al. presented the first paper on reversible unidirectional barrel shifter in [22]. An improved design of the reversible unidirectional barrel shifter presented in [21]. Our proposed work on unidirectional shifter outperforms these mentioned works by a huge margin as shown in [7]. As mentioned in Sect. 1, this paper shows the extension of our proposed work in transistor level realizations and algorithmic representation. The design methodology of a (8, 3) reversible logarithmic bidirectional arithmetic and a logical barrel shifter presented in [23]. Here, the circuits are constructed with Feynman and Fredkin gates in addition to many elements of the theory for the (n, k) circuit. Besides these, the design methodology of a reversible (4, 2) universal barrel shifter is presented in [29]. Except our proposed work [7], all the other existing works are unscalable and generalized i.e., not generalized. Thus, a single structure for all the shifters under a generalized and scalable architecture is needed.

3 Proposed reversible fault tolerant unidirectional logarithmic barrel shifters

In this section, an enhancement of our work [7] on unidirectional barrel shifter is presented. The proposed adaptive structure of the unidirectional shifters and algorithm have extensively used to design other shifters which are being presented in the remaining sections.

Theorem 1 *A reversible unidirectional barrel shifter can be realized with at least k garbage outputs and no constant input, where n is the number of data bits and* $k = log_2 n$.

Proof The unidirectional barrel shifter with n data bits has k control inputs, where $k = \log_2 n$, i.e., total of $n+k$ inputs. Thus, according to the property of reversibility, it should have at least $(n + k)$ outputs among which n bits are the primary outputs. So, the number of garbage outputs is k, which means that no constant input is required. □

3.1 Proposed reversible fault tolerant unidirectional logarithmic rotators

Figures 7 and 8 show the architecture of the proposed (4, 2) and (8, 3) logarithmic rotators, respectively. From these figures, we find that the proposed rotators are designed using only reversible fault tolerant Fredkin gates. Thus, according to our discussion in Sect. 2.1, the proposed rotators preserve parity. Proposed unidirectional rotators follow the adaptive structure of logarithmic barrel shifter, *e.g.*, the 1^{st} stage rotates 2^0 bit, the 2nd stage rotates 2^1 bits and so on. Rotation occurs only when control signal is set to high. If any stage's control signal is set to low then instead of rotating that stage just passes the input to the next stage.

Example 1 Let the value of n be 4. Then, we have (4, 2) reversible barrel shifter. Figure 7 shows the architecture of the proposed (4, 2) reversible fault tolerant barrel shifter (circuits for rotation) which has 2 two garbage outputs and no constant input. If we replace n with 4 in Theorem 1, we get two garbage outputs and no constant inputs as well. Thus, the proposed rotators are the optimal design in-terms of garbage output and constant input as it corresponds to Theorem 1.

3.1.1 Design and working procedure of the proposed reversible fault tolerant unidirectional rotators

Let the data inputs for the proposed (n, k) rotator be $I_0, I_1, I_2, \ldots, I_{n-3}, I_{n-2}, I_{n-1}$ and the control inputs be $S_0, S_1, \ldots, S_{k-1}$; where n represents the number of inputs and k is the number of stages equals to $\log_2 n$. Each of these stages require a chain of $(n - 2^j)$ Fredkin gates. The inputs and outputs for each Fredkin gate in a stage j can be rewritten as $A(i, j)$, $B(i, j)$, $C(i, j)$ and $P(i, j)$, $Q(i, j)$, $R(i, j)$, respectively. Here, j represents the j^{th} stage, where $n - 1 \geq i \geq 0$ and $k - 1 \geq j \geq 0$. The working procedure of the proposed reversible fault tolerant rotator is as follows[1]: Let $n = 4$, then we have a (4, 2) rotator. It takes $I_0 I_1 I_2 I_3$ as data inputs and S_0 and S_1 as control inputs. If both the control inputs are set to high ($S_0 = 1$, $S_1 = 1$), then data inputs will be rotated $2^0 + 2^1$ times to the right. Sequence of the rotate operations will be $I_1 I_2 I_3 I_0$ for the first stage and $I_3 I_0 I_1 I_2$ for the next. On the other hand, if both control inputs are set to low, then the input sequence will remain unchanged. For example, when control input S_0 is set to low, the 1st Fredkin gate (FRG) of first row will select input I_0, 2nd will select I_1 and the 3rd one will select I_2 also I_3 will be generated from another output of the 3rd FRG. Finally, the selected four outputs will be used as inputs of 1st and 2nd FRG at 2nd row, which are controlled by the control input S_1. Besides, if the 1st control input is set to high, then the selection sequence will be I_1, I_2, I_3 and I_0 which are 1st, 2nd and 3rd FRG at 1st row, respectively. Figure 7c, d shows these working procedures. From the above figures, we find that the proposed fault tolerant left and right rotators are very similar to each other except the input-output positions. Hence, only the architecture and algorithm of the proposed reversible fault tolerant (n, k) right rotator is presented here which is shown in Fig. 9 and Algorithm 1, respectively.

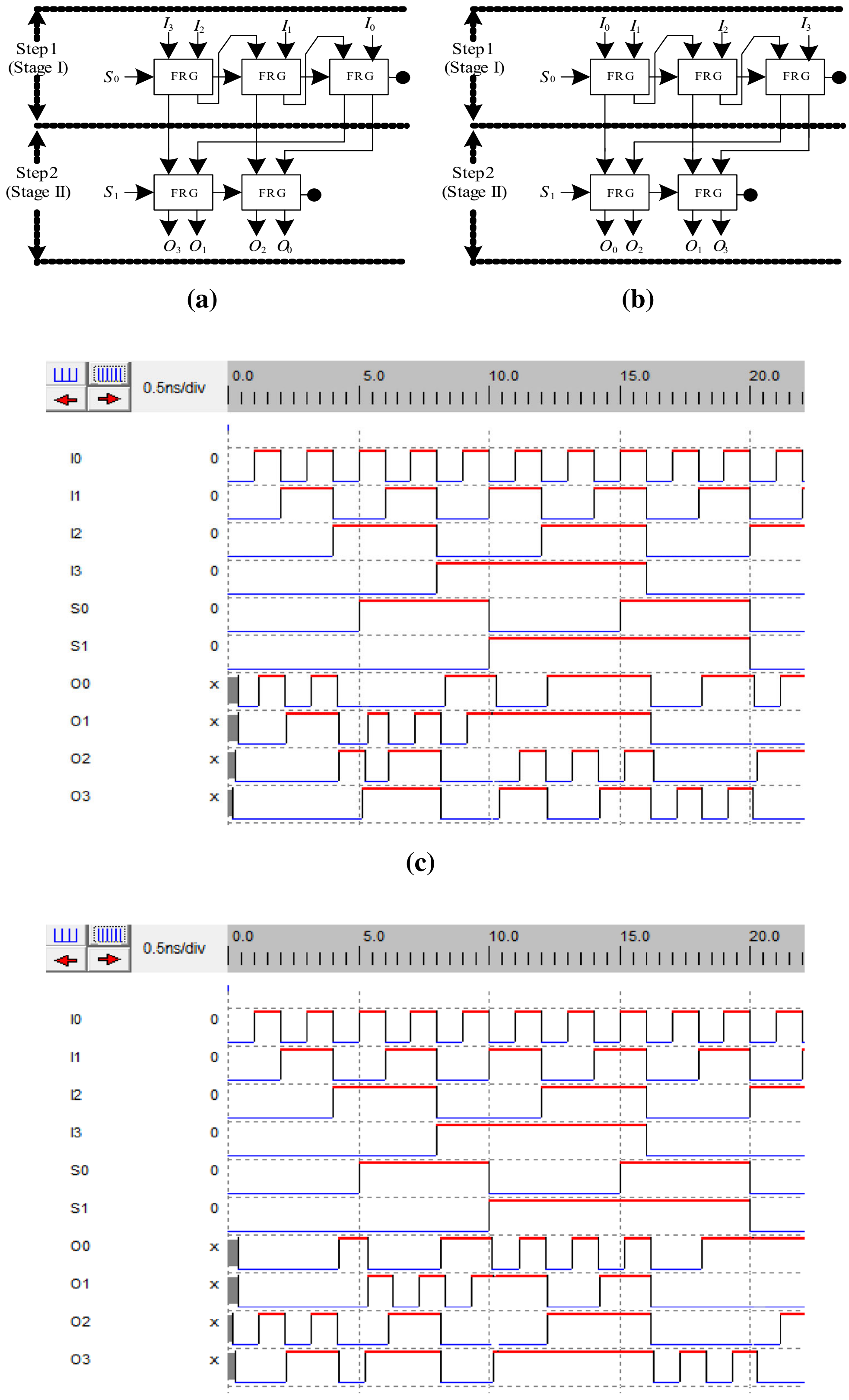
Lemma 1 *Let FR be the required number of gates for (n, k) fault tolerant unidirectional rotator, where n and k* ($k = log_2 n$) *are the numbers of data bits and control inputs, respectively. Then,* $FR = nk - \sum_{j=0}^{k-1} 2^j$.

Proof A (n, k) reversible fault tolerant unidirectional logarithmic rotator has k stages and n input bits. According to our design procedure, each j^{th} stage requires $(n\text{-}2^j)$ Fredkin gates, where $j = 0$ to $(k - 1)$. So, the required number of gates for a (n, k) reversible fault tolerant rotator is $(n - 2^0) + (n - 2^1) + (n - 2^2) + \cdots + (n - 2^{k-2}) + (n - 2^{k-1}) = nk - \sum_{j=0}^{k-1} 2^j$. □

Lemma 2 *Let QC be the total quantum cost for (n, k) reversible fault tolerant unidirectional logarithmic rotator. Then,* $QC = 5nk - 5\sum_{j=0}^{k-1} 2^j$.

Proof In Lemma 1, we proved that the proposed reversible fault tolerant (n, k) rotator can be realized with

[1] This is for the proposed unidirectional right rotator's circuit. Unless mentioned explicitly, the design and working procedure elaborated in this paper is for the right rotator/shifter.



(a)

(b)

(c)

(d)

Fig. 7 Proposed (4, 2) reversible fault tolerant unidirectional logarithmic barrel shifter. **a** Circuit for left rotation. **b** Circuit for right rotation. **c** Corresponding timing diagram of (4, 2) left rotator. **d** Corresponding timing diagram of (4, 2) right rotator

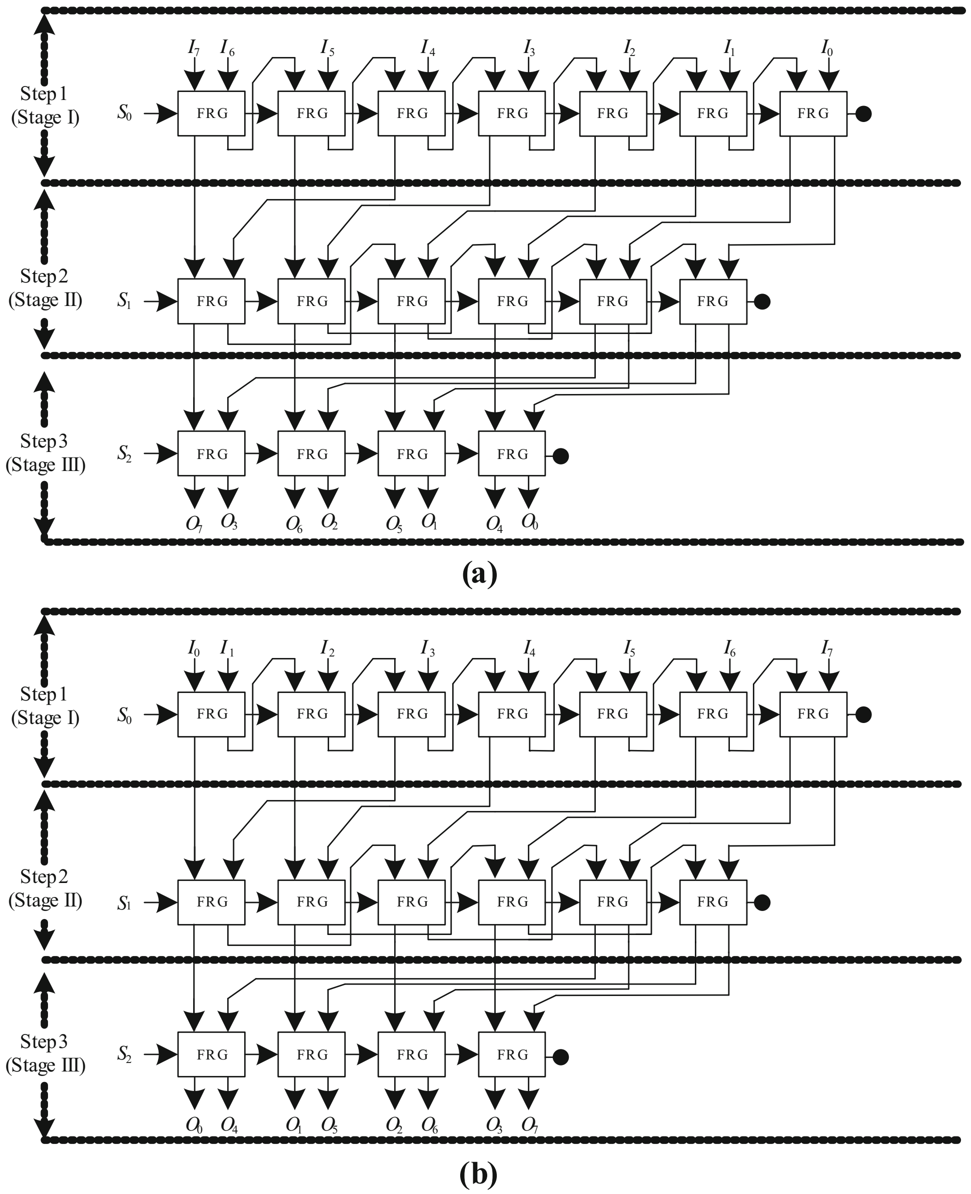

Step 1
(Stage I)
S_0
I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
FRG
Step 2
(Stage II)
S_1
Step 3
(Stage III)
S_2
O_7 O_3 O_6 O_2 O_5 O_1 O_4 O_0
(a)
Step 1
(Stage I)
S_0
I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7
FRG
Step 2
(Stage II)
S_1
Step 3
(Stage III)
S_2
O_0 O_4 O_1 O_5 O_2 O_6 O_3 O_7
(b)

Fig. 8 Proposed (8, 3) reversible fault tolerant unidirectional logarithmic barrel shifter. **a** Circuit for left rotation. **b** Circuit for right rotation

$(nk - \sum_{j=0}^{k-1} 2^j)$ reversible fault tolerant Fredkin gates. According to our discussion in Sect. 2, quantum cost of each Fredkin gate is 5. Therefore, total quantum cost of the (n, k) reversible fault tolerant unidirectional logarithmic rotator is $(5nk - 5\sum_{j=0}^{k-1} 2^j)$. □

Lemma 3 *Let* α, β *and* γ *be the hardware complexity for two-input Ex-OR, AND, NOT calculations, respectively. Then the total hardware complexity (HC) for (n, k) fault tolerant unidirectional logarithmic rotator is* $(2nk - \sum_{j=0}^{k-1} 2^{j+1})\alpha + (nk - \sum_{j=0}^{k-1} 2^j)(4\beta + \gamma)$.

Proof Lemma 1 proved that the proposed reversible fault tolerant (n, k) unidirectional logarithmic rotator can be realized with $(nk - \sum_{j=0}^{k-1} 2^j)$ Fredkin gates. Hardware complexity of a Fredkin gate is $(2\alpha+4\beta+\gamma)$ [5]. So, the hardware complexity of the (n, k) unidirectional rotator is $(2nk - \sum_{j=0}^{k-1} 2^{j+1})\alpha$ $+(4nk - \sum_{j=0}^{k-1} 2^{j+2})\beta$ $+ (nk - \sum_{j=0}^{k-1} 2^j)\gamma$=$2nk - \sum_{j=0}^{k-1} 2^{j+1})\alpha$+ $(nk - \sum_{j=0}^{k-1} 2^j)(4\beta+\gamma)$. □

3.2 Proposed reversible fault tolerant unidirectional logical shifters

Design procedure of the proposed (n, k) reversible fault tolerant unidirectional logarithmic logical shifters is similar to the proposed rotators but in reverse direction. Each k stages of

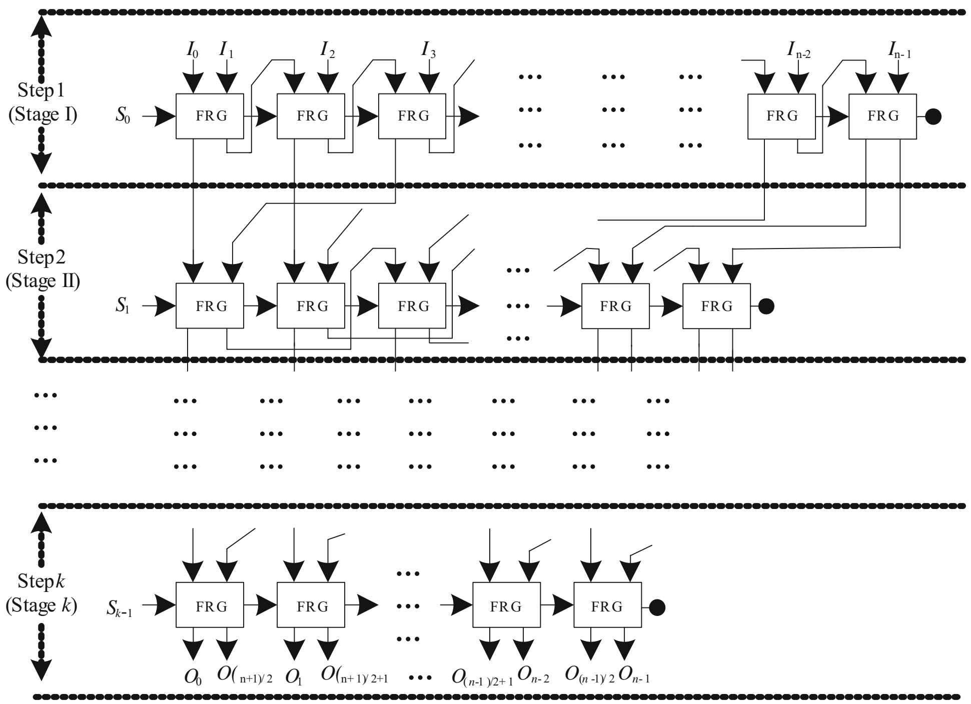


Fig. 9 Proposed (n, k) reversible fault tolerant unidirectional logarithmic right rotator

the proposed (n, k) logical shifter is responsible for shifting input data by 2^{k-1} to 2^0 bits. The architecture of the proposed (4, 2) and (8, 3) reversible fault tolerant unidirectional logical shifters are shown in Figs. 10 and 11, respectively. The working procedure of the proposed (n, k) unidirectional right logical shifter is as follows: Inputs to the 1st stage are,

$A(i, k-1) = S_{k-1}$, **for** $i \leftarrow (n-1)$ to 0
$B(i, k-1) = i_i$, **for** $i \leftarrow (n-1)$ to 0
$C(i, k-1) = 0$, **for** $i \leftarrow (n-1)$ to $((n-1) - 2^{k-1} + 1)$
$C(i, k-1) = R_m$, **for** $i \leftarrow ((n-1) - 2^{k-1})$ to 0; R is 3^{rd} output of the m Fredkin gate, where $m \leftarrow (n-1)$ to $((n-1)\text{-}2^{k-1} + 1)$.

The controlling input for all Fredkin gates in the first stage of the proposed logical shifter is set to $A(i, k-1) = S_{k-1}$. When S_{k-1}=1,

$P(i, k-1) = S_{k-1}$, **for** $i \leftarrow (n-1)$ to 0
$R(i, k-1) = i_i$, **for** $i \leftarrow (n-1)$ to 0
$Q(i, k-1) = 0$, **for** $i \leftarrow (n-1)$ to $(n-1) - 2^{k-1} + 1$
$Q(i, k-1) = i_m$, **for** $i \leftarrow (n-1) - 2^{k-1}$ to 0 and $m = (n-1)$ to 2^{k-1}

When S_{k-1}=0, we have following values at the outputs.

$$P(i, k-1) = S_{k-1},\ \textbf{for}\, i \leftarrow (n-1)\ \text{to}\,0$$
$$Q(i, k-1) = i_i,\ \textbf{for}\, i \leftarrow (n-1)\ \text{to}\,0$$
$$R(i, k-1) = 0, \textbf{for}\, i \leftarrow (n-1)\ \text{to}\,0$$

The second stage of the proposed logical shifters uses inputs from the first stage.

$A(i, k-2) = S_{k-2}$, **for** $i \leftarrow (n-1)$ to 0
$B(i, k-2) = Q(i, k-1)$, **for** $i \leftarrow (n-1)$ to 0
$C(i, k-2) = 0$, **for** $i \leftarrow (n-1)$ to $((n-1) - 2^{k-2} + 1)$
$C(i, k-2) = O_m$, **for** $i \leftarrow ((n-1) - 2^{k-2} + 1)$ to 0 and O is 3^{rd} output of the m Fredkin gate; where $m \leftarrow (n-1)$ to $((n-1) - 2^{k-2} + 1)$; and $m = (n-1)$ to 2^{k-2}.

Since, S_{k-2} works as the input A of the Fredkin gates, we will have following values at the outputs, when S_{k-2} = 1,

$P(i, k-2) = S_{k-2}$, **for** $i \leftarrow (n-1)$ to 0
$R(i, k-2) = Q(i, k-1)$, **for** $i \leftarrow (n-1)$ to 0
$Q(i, k-2) = 0$, **for** $i \leftarrow (n-1)$ to $(n-1) - 2^{k-2} + 1$
$Q(i, k-1) = Q(m, k-1)$, **for** $i \leftarrow (n-1) - 2^{k-2}$ to 0; $m = (n-1)$ to 2^{k-2}.

When S_{k-2}=0, we will have following values at the outputs.

$$P(i, k-2) = S_{k-2},\ \textbf{for}\, i \leftarrow (n-1)\ \text{to}\,0$$
$$Q(i, k-2) = Q(i, k-1),\ \textbf{for}\, i \leftarrow (n-1)\ \text{to}\,0$$
$$R(i, k-2) = Q(i, k-1),\ \textbf{for}\, i \leftarrow (n-1)\ \text{to}\,0.$$

Similar design strategy are used for the remaining stages. From Figs. 11 and 12, we find that the proposed (4, 2) and (8, 3) left and right logical shifters are identical to each other except the input-output positions. so, only the architecture and algorithm of the proposed fault tolerant (n, k) right logical shifting is presented which is shown in Fig. 13 and Algorithm 2 respectively.

Algorithm 1: Algorithm for the proposed reversible fault tolerant (n, k) unidirectional right rotator, ***UR(n, k, F2G, FRG)***

```
Input  : Data input set $I(I_0, I_1, ..., I_{n-1})$ and Control input set
         $S(S_0, S_1, ..., S_{k-1})$ Feynman double gate ($F2G$) and
         Fredkin gate ($FRG$)
Output: Reversible fault tolerant $(n, k)$ unidirectional rotator
         circuit
begin
    $i = input,\ \ o = output$
    for $j \leftarrow 0$ to $k-1$ do
        $S_j \rightarrow first.i.FRG$
        if $j$=0 then
            $I_{j+1} \rightarrow second.i.FRG$
            for $l \leftarrow 1$ to $n-1$ do
                $third.o.frg \rightarrow first.i.FRG$
                $third.o.frg_{l-1} \rightarrow second.i.FRG$
                $I_{j+2} \rightarrow third.i.FRG$
        else
            for $m \leftarrow 0$ to $n-k$ do
                $third.o.frg \rightarrow first.i.FRG$
                if $j \geq k$ then
                    $third.(o-2^j).FRG \rightarrow second.i.FRG$
                else
                    $Stage_{j-1}.second.o.FRG \rightarrow second.i.FRG$
                $Stage_{j-1}.third.o_{j-m}.FRG \rightarrow third.i.FRG$
        return if $j = (k-1)$ then
            for $q \leftarrow 0$ to $n-j$ do
                $second.o.FRG \rightarrow Desire\ Ouptut$
                if $q = (n-j)$ then
                    $third.o.FRG \rightarrow Garbage\ Output$
                else
                    $third.o.FRG \rightarrow Desire\ Output$
        $Remaining.last.third.o \rightarrow Garbage\ Output$
```

Lemma 4 *Let **F** be the required number of gates for a (n, k) reversible fault tolerant logical shifter, where n is the number of data bits and $k = log_2n$. Then, **F** = nk.*

Proof A (n, k) logical shifter has k stages (0 to $k-1$) and n input bits. According to our design procedure, n Fredkin gates are required for each stage. Therefore, total number of Fredkin gates required for reversible fault tolerant (n, k) logical shifter is $\sum_0^{k-1} n = nk$. □

Lemma 5 *A (n, k) reversible fault tolerant unidirectional logical shifter can be realized with 5nk quantum cost and ($2nk\ \alpha + 4nk\ \beta + nk\ \gamma$) hardware complexity.*

Proof Lemma 4 proved that the unidirectional (n, k) logical shifter can be realized with $nk\ FRG$. Each FRG is realized with 2 quantum cost and $2\alpha + 4\beta + \gamma$ hardware complexity. Hence, the (n, k) fault tolerant unidirectional logical shifter is realized with $5nk$ quantum cost and ($2nk\ \alpha + 4nk\ \beta + nk\ \gamma$) hardware complexity. □

4 Proposed reversible fault tolerant bidirectional barrel shifters

Initially this proposes a lower bound on the number of garbage output of the reversible fault tolerant bidirectional shifter. Then we presents the detail conversion procedures of the proposed unidirectional shifters to bidirectional shifter.

Theorem 2 *A reversible bidirectional barrel shifter can be realized with at least* k+1 *garbage outputs and no constant input, where n is the number of data inputs and $k = log_2n$.*

Proof Reversible bidirectional barrel shifter with an n-bit data input has $k+1$ control inputs i.e., $n+k+1$ inputs in total. Thus according to our discussion in Sect. 2 and Theorem 1, there must be $n+k+1$ output bits in a reversible bidirectional barrel shifter. However, among these output bits, there are only n primary outputs. So, there are at least $k+1$ garbage outputs which means that no constant input is required. □

4.1 Proposed fault tolerant bidirectional rotators

Algorithm 3 shows the working procedures of the bidirectional barrel shifter. Here, control signal (S_{LR}) determines the direction of shift or rotate operations (shown in lines 4 and 6). Figure 13a, b shows the architecture and corresponding simulation of the proposed (4, 2) reversible fault tolerant bidirectional rotators. Comparing the Fig. 13 with Fig. 7, we find that the proposed bidirectional rotator is only the extension of the proposed (4, 2) unidirectional rotators with two additional gates. These additional gates are needed to maintain the direction of rotate operations.

Lemma 6 *A* (n, k) *reversible fault tolerant logarithmic bidirectional rotator can be realized with $n(k-1)+(k+1)$ reversible fault tolerant gates, where* n *is the number of data inputs and $k = log_2n$.*

Proof A (n, k) reversible fault tolerant bidirectional rotator has k stages and n input bits. According to our design procedure, each j^{th} stage requires (n-2^j) Fredkin gates, where j=0 to $(k-1)$. In addition to that, one Fredkin gate and $(k-1)$ Feynman double gates are needed to maintain bidirectionality. Therefore, total number of reversible fault tolerant gates required for a (n, k) reversible fault tolerant logarithmic rotator is $\sum_{j=0}^{k-1}(n-2^j)+1+(k-1) = n(k-1)+(k+1)$. □

Lemma 7 *A* (n, k) *reversible fault tolerant bidirectional rotator can be realized with $5n(k-1)+2(k+4)$ quantum cost and $2(nk-n+k+1)\ \alpha + 4(nk-n+2)\ \beta + 2(nk-n+2)$*

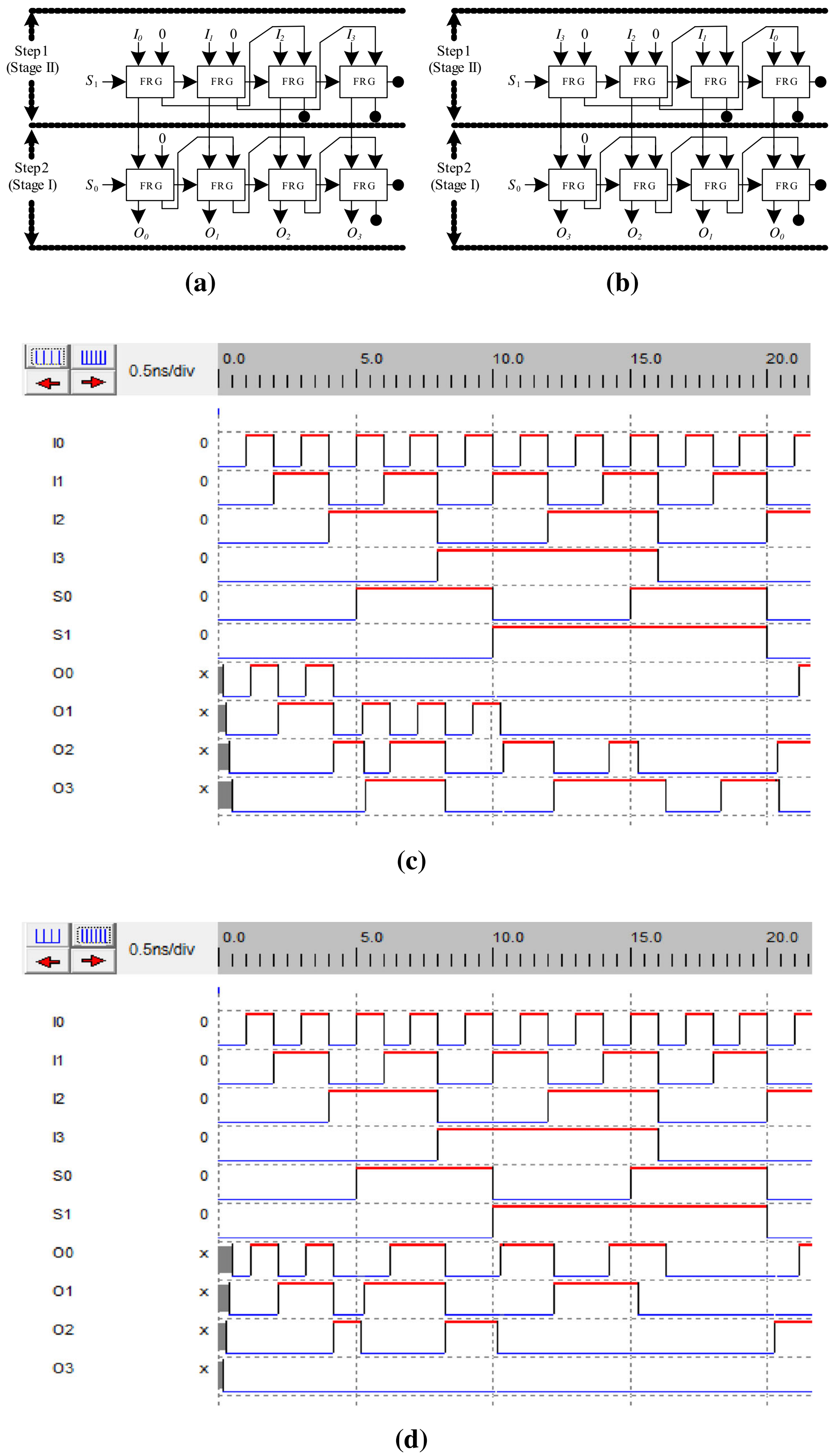


Fig. 10 Proposed (4, 2) reversible fault tolerant unidirectional logarithmic barrel shifter. **a** Circuit for left logical shifting. **b** Circuit for right logical shifting. **c** Corresponding timing diagram of (4, 2) left logical shifter. **d** Corresponding timing diagram of (4, 2) right logical shifter

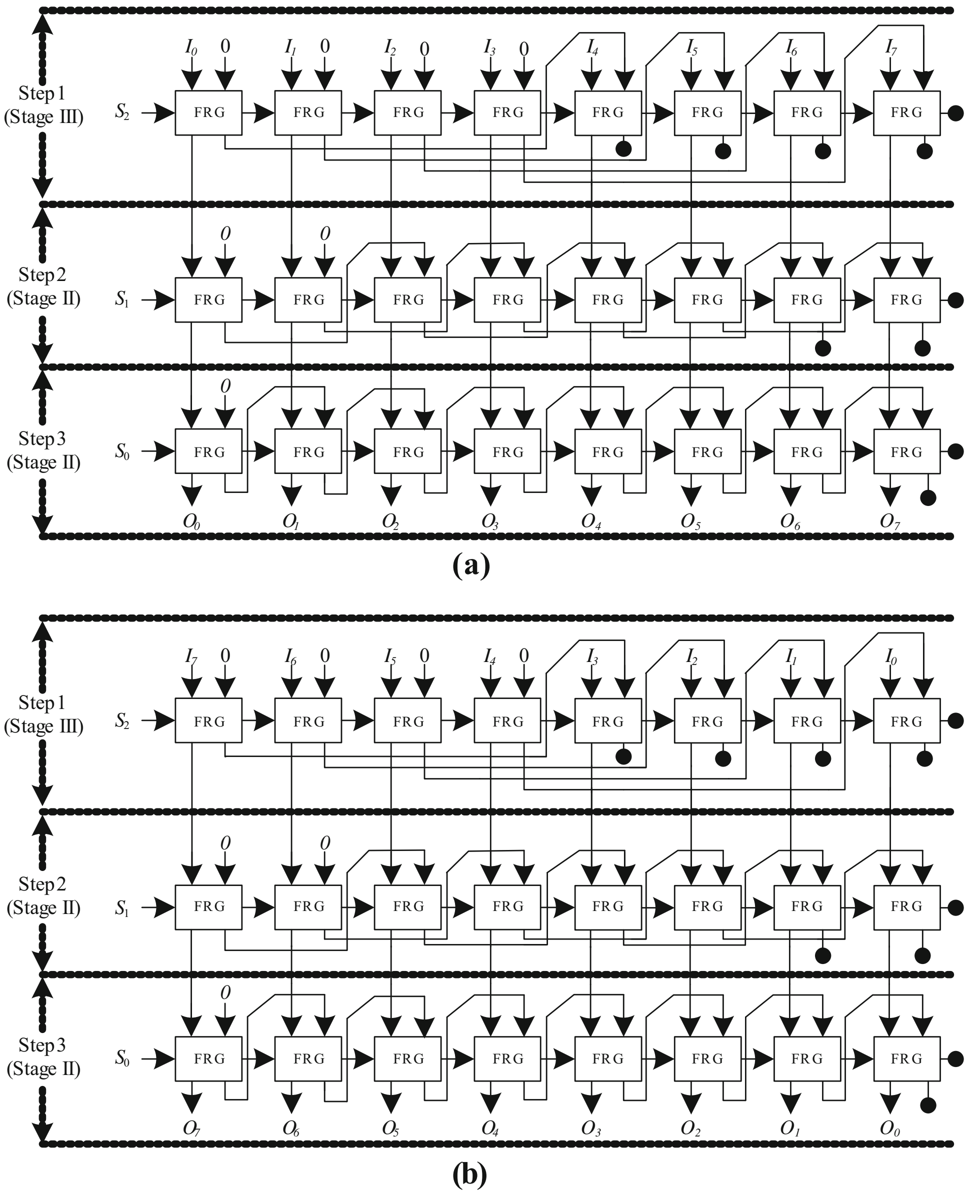

Step 1
(Stage III)
Step 2
(Stage II)
Step 3
(Stage II)
S_2
S_1
S_0
I_0 0 I_1 0 I_2 0 I_3 0 I_4 I_5 I_6 I_7
0 0
0
FRG
O_0 O_1 O_2 O_3 O_4 O_5 O_6 O_7
(a)
I_7 0 I_6 0 I_5 0 I_4 0 I_3 I_2 I_1 I_0
O_7 O_6 O_5 O_4 O_3 O_2 O_1 O_0
(b)

Fig. 11 Proposed (8, 3) reversible fault tolerant unidirectional logarithmic logical shifter. **a** Circuit for left logical shift. **b** Circuit for right logical shift

γ hardware complexity, where n *is the number of data inputs, $k = log_2 n$ and α, β and γ are the hardware complexity for two input Exclusive-OR, AND, NOT calculations respectively.*

Proof In Lemma 6, we have proved that the (n, k) reversible fault tolerant bidirectional rotator can be realized with $\sum_{j=0}^{k-1}(2n-2^j)+1$ Fredkin gates and $(k-1)$ Feynman double gates. According to our discussion in Sect. 2, the quantum cost of each Fredkin and Feynman double gates is 5 and 2, respectively. On the other hand, hardware complexity of each Fredkin and Feynman double gate is $2\alpha + 4\beta + \gamma$ and 2α respectively [5]. Thus, total quantum cost of the proposed (n, k) bidirectional rotator is

$$\left(5\left(\sum_{j=0}^{k-1}(n-2^j)+1\right)+2(k-1)\right)$$
$$= 5n(k-1)+2(k+4)$$

and hardware complexity is

$$\left(\sum_{j=0}^{k-1}(n-2^j)+1\right)(2\alpha+4\beta+\gamma)+2(k-1)\alpha$$
$$= 2(nk-n+k+1)\alpha+4(nk-n+2)\beta$$
$$+2(nk-n+2)\gamma.$$

□

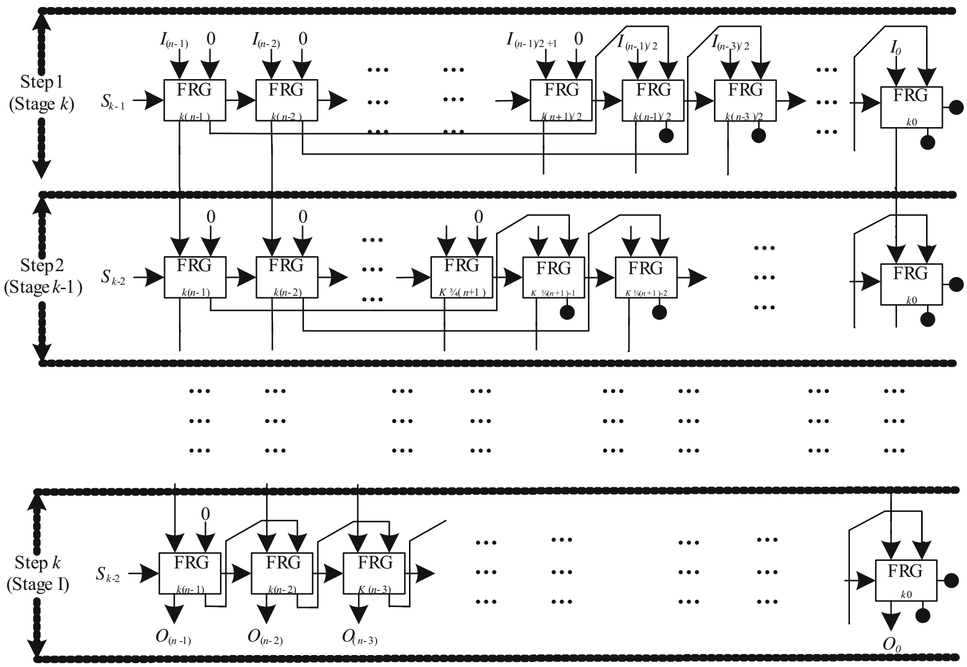


Fig. 12 Proposed (n, k) reversible fault tolerant logarithmic unidirectional logical right shifter

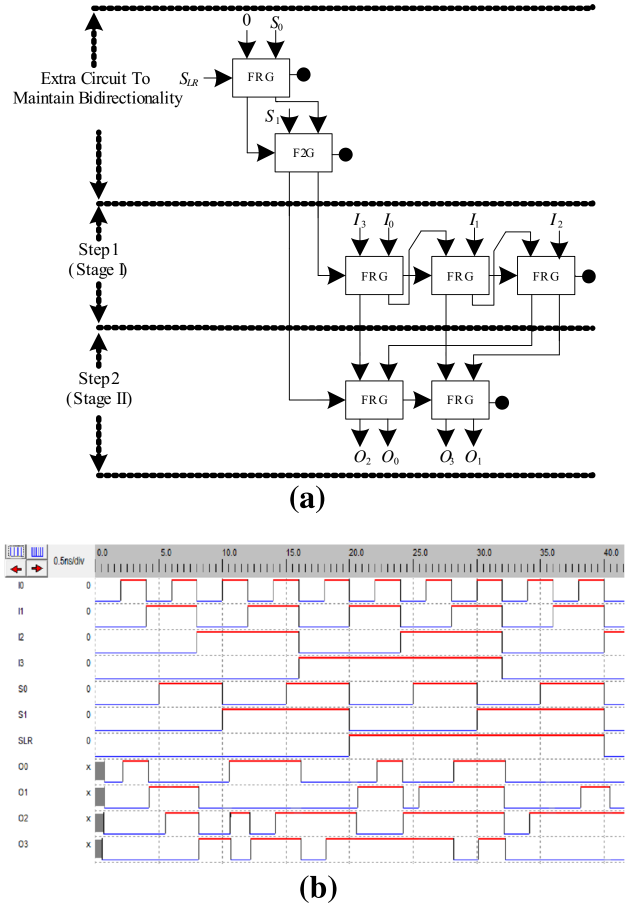


Fig. 13 Proposed (4, 2) reversible fault tolerant bidirectional rotator. **a** Architectural block diagram. **b** Corresponding timing diagram

Algorithm 2: Algorithm for the (n, k) unidirectional right logical shifter, ***URLS(n, k, F2G, FRG)***

Input : Data input set $I\,(I_0, I_1, ..., I_{n-1})$ and Control input set $S(S_0, S_1, ..., S_{k-1})$ and Fredkin gate (FRG)

Output: Reversible fault tolerant (n, k) unidirectional logical shifter circuit

```
begin
    $i = input,\ \ o = output$
    for $j \leftarrow 0$ to $k-1$ do
        $S_j \rightarrow first.i.FRG$
        for $l \leftarrow 0$ to $\lceil n/2 - 2^j \rceil$ do
            $0 \rightarrow third.i.FRG$
        if $j = 0$ then
            for $m \leftarrow 0$ to $n-1$ do
                $I_{n-j} \rightarrow second.i.FRG$
        else
            for $p \leftarrow 0$ to $n-2$ do
                $second.o_{j-1}.FRG \rightarrow second.i_j.FRG$
        $remaining.third.i_j.FRG \leftarrow third.o_{n/2-2^j}.FRG$
        return if $j = (k-1)$ then
            for $q \leftarrow 0$ to $n-1$ do
                $second.o.FRG \rightarrow Desire\ Ouptut$
        $remaining.o \rightarrow Garbage\ Output$
```

The architectural block diagram and algorithmic designs of the proposed reversible fault tolerant (n, k) bidirectional rotator is shown in Fig. 14 and Algorithm 4, respectively. Lines 3–14 of the algorithm design the additional circuitry, which is needed to maintain bidirectionally. Although line 15 calls Algorithm 1 to built the base circuitry of unidirectional rotators.

Algorithm 3: Operation in bidirectional barrel shifter

```
input : Data input set $I(I_0, I_1, ..., I_{n-1})$
        Control input set $S(S_0, S_1, ..., S_{k-1})$
        Direction determiner control signal $S_{LR}$
output: Output data set $O(O_0, O_1, ..., O_n - 1)$
begin
    for $j \leftarrow 0$ to $k-1$ do
        if $S_j$=1 & $S_{LR}$ = 1 then
            $I \longleftarrow$ Shift/Rotate $I$, $2^j$ Times Left
        else if $S_j$=1 & $S_{LR}$ = 0 then
            $I \longleftarrow$ Shift/Rotate $I$, $2^j$ Times Right
        else
            $I \longleftarrow I$
    return $O \longleftarrow I$
```

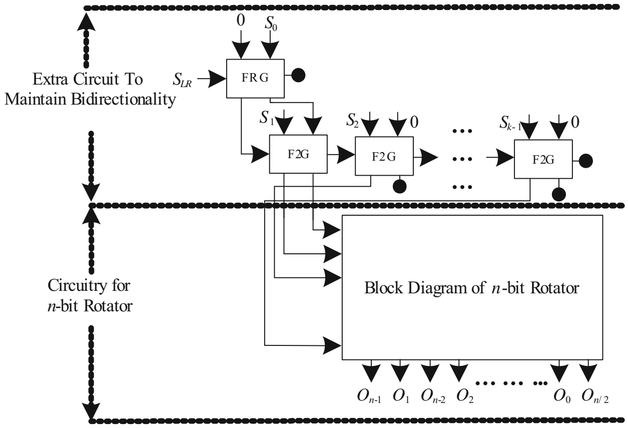

0
S_0
Extra Circuit To Maintain Bidirectionality
S_{LR}
FRG
S_1
S_2
0
S_{k-1}
0
F2G
F2G
F2G
Circuitry for n-bit Rotator
Block Diagram of n-bit Rotator
O_{n-1} O_1 O_{n-2} O_2 O_0 $O_{n/2}$

Fig. 14 Proposed reversible fault tolerant bidirectional rotator: circuitry for n-data bits

4.2 Proposed reversible fault tolerant bidirectional logical shifters

Algorithm 5 presents the design procedure of the proposed (n, k) reversible fault tolerant bidirectional logical shifter. The proposed shifter is capable of logical shifting the inputs in both left and right directions depending on the value of the control signal S_{LR}.

According to Algorithm 5, architecture of the proposed (4, 2) and (n, k) fault tolerant bidirectional logical shifter is shown in Figs. 15a and 16, respectively. Figure 15b represents the corresponding timing diagram of Fig. 15a. According to the procedure, both the left and right unidirectional schema can be used for the base extension of the bidirectional schema form the unidirectional one. Here, bidirectional logical shifter is schematized through the extension of the proposed unidirectional logical shifter.

Lemma 8 *Let* n *be the number of data input bits for* (n, k) *reversible fault tolerant logarithmic bidirectional logical shifter and* $k = log_2 n$. *Let* F_3 *be the required number of gates for* (n, k) *reversible fault tolerant bidirectional logical shifter. Then,* $F_3 = n(k+1)$

Proof A (n, k) reversible fault tolerant bidirectional logical shifter has k stages and n input bits. According to our design procedure, number of Fredkin gates required for each j^{th} stage is (n), where $j = k - 1$ to 0. Also we need n additional Fredkin gates to maintain bi-directionality. Therefore, total number of gates required is, $\sum_{j=0}^{k-1} n + n = n(k+1)$ □

Lemma 9 *Let* n *be the number of data input bits for* (n, k) *reversible fault tolerant logarithmic bidirectional logical shifter and* $k = log_2 n$. *Let* G_3 *be the required number of garbage outputs and for* (n, k) *reversible fault tolerant bidirectional logical shifter. Then,* G_3=$(n+k)$.

Proof A (n, k) reversible fault tolerant bidirectional logical shifter has k stages and total of n input bits. According to our design procedure, each stage produces ($n - 2^j$) garbage outputs, where, $j = 0$ to $(k-1)$. Extra control input which is needed to preserver the bi-directionality produces another garbage output. So, total garbage outputs produced by a (n, k) reversible fault tolerant bidirectional logical shifter is,

$$\sum_{j=0}^{k-1} (n - 2^j) + 1 = (n+k)$$

□

Lemma 10 *Let* n *be the number of data input bits for* (n, k) *reversible fault tolerant logarithmic bidirectional logical shifter and k be the number of number of stages equals to* $log_2 n$. *Then the* (n, k) *reversible fault tolerant bidirectional logical shifter can be realized with,* $5n(k+1)$ *quantum cost and* $(2nk + 2n)$ α + $(4nk + 4n)$ β+ $(nk + n)$ γ *hardware complexity; where,* α, β *and* γ *are the hardware complexity for two input Exclusive-OR, AND, NOT calculations respectively.*

Proof In Lemma 8 we have proved that the (n, k) reversible fault tolerant bidirectional logical shifter can be realized with $n(k+1)$ Fredkin gate. According to our earlier discussion (Sect. 2.2), the quantum cost of each Fredkin is 5. So, total quatum cost of the proposed (n, k) reversible fault tolerant bidirectional logical shifter is $5n(k+1)$. As shown in Sect. 2.2, the hardware complexity of each Fredkin gate is $2\alpha + 4\beta + \gamma$. Thus, the hardware complexity of the proposed (n, k) reversible fault tolerant bidirectional logical shifter is, $(2nk + 2n)$ α+$(4nk + 4n)$ β+$(nk + n)$ γ. □

Algorithm 4: Algorithm for the proposed reversible fault tolerant bidirectional rotator, ***UBR(n, k+1, F2G, FRG)***

```
Input : Data input set $I(I_0, I_1, ..., I_{n-1})$ and Control input set
        $S(S_0, S_1, ..., S_{k-1})$, Directional signal $S_{LR}$, Fredkin gate
        $(FRG)$ and Feynman double gate $(F2G)$
Output: Reversible fault tolerant bidirectional rotator circuit for
        $n$-data bits
begin
    $i = input,\ o = output$
    $S_{LR} \rightarrow first.i.FRG, 0 \rightarrow second.i.FRG,$
    $S_0 \rightarrow third.i.FRG$
    for $j \leftarrow 1$ to $k-1$ do
        if $j$=1 then
            $second.o.FRG \rightarrow first.i.F2G$
            $S_j \rightarrow second.i.F2G$
            $third.o.FRG \rightarrow third.i.FRG$
        else
            $second.o_{j-1}.F2G \rightarrow first.i_j.F2G$
            $S_j \rightarrow second.i.F2G$ , $0 \rightarrow third.i.FRG$
        Call UR(n, k (third output of first F2G as $S_0$, and second
        output of $F2G_j$ as $S_1$ to $S_{k-1}$), F2G, FRG)
    return Desire output of UR as Desire Output
    Garbage output of UR as Garbage Output
```

Algorithm 5: Algorithm for the proposed reversible fault tolerant bidirectional logical shifter, ***BLS(n, k+1, F2G, FRG)***

```
Input : Data input set $I(I_0, I_1, ..., I_{n-1})$ and Control input set
        $S(S_0, S_1, ..., S_{k-1})$, Direction determiner control signal
        $S_{LR}$, Fredkin gate $(FRG)$
Output: Reversible fault tolerant bidirectional logical shifter's
        circuit for $n$-data bits
begin
    $i = input,\ o = output$
    $S_{LR} \rightarrow first.i.FRG$
    for $j \leftarrow 1$ to $n/2$ do
        $second.o.FRG_j \rightarrow I_{n-j-1}$
        $third.o.FRG_j \rightarrow I_j$
    Call URLS(n, k, F2G, FRG)
    $third.FRG_{n/2}.o \rightarrow first.i.FRG$
    $second.i.FRG_l \leftarrow URLS.O_0$
    $third.i.FRG_l \leftarrow URLS.O_{n-1}$
    for $l \leftarrow 1$ to $n/2$ do
        $third.o.FRG_{l+1} \rightarrow first.i.FRG_l$
        $URLS.O_{l-1} \rightarrow second.i.FRG_l$
        $URLS.O_{(l+n-1)mod(n-1) \rightarrow third.i.FRG_l}$
    return for all bottom level $FRG$ do
        $first.o.FRG$ & $second.o.FRG$ as Desire Output
        and
        Remaining Output as Garbage Output
```

5 Proposed reversible fault tolerant universal barrel shifters

Universal shifter performs logical shifting, arithmetic shifting and rotate operation in a single circuit. As shown earlier, the proposed left shifter can be designed from the proposed right shifters by interchanging input-output. Hence, initially we propose the reversible fault tolerant universal right shifter here. The proposed universal right shifter is designed from the proposed logical right shifter in addition with some extra gates. The additional gates are needed to support the arithmetic shifting and rotate operations. There are two extra control inputs used for this purpose. Table 2 summarizes the operations of these control signals. Algorithm 6 represents the design procedure of the proposed reversible fault tolerant universal right shifter for n data inputs. According to the algorithm, proposed (8, 3) universal shifter circuit is shown in Fig. 17.

The working procedure of the above algorithm is as follows: The first stage of (n, k) universal right shifter requires 2^{k-1} number of additional Fredkin gates in addition to the chain of n Fredkin gates. The inputs and outputs for these additional Fredkin gates are represented as $A_r(i, j)$, $B_r(i, j)$, $C_r(i, j)$ and $P_r(i, j)$, $Q_r(i, j)$, $R_r(i, j)$ respectively. The inputs are passed to these additional Fredkin gates as follows:

$A_r(i, k-1) = S_{RR}$, **for** $i \leftarrow (2^{k-1}-1)$ to 0

$B_r(i, k-1) = Q_r$, **for** $i \leftarrow (2^{k-1}-k)$ to 0 from right, remaining $B_r = 0$, except the last stage; for the last stage, if S_{RS}=1 then B_r= I_7 else $B_r = 0$.

$C_r(i, k-1) = i_m$, **for** $i \leftarrow 2^{k-1}$ to 0, and $m = (2^{k-1}-1)$ to 0.

In the first stage, the additional Fredkin gates are controlled by the control signal S_{RR}. Thus, for all $i \leftarrow (2^{k-1}-1)$ to 0, the controlling input $A_r(i, k-1)$ is set to S_{RR} . This assigning of S_{RR} to $A_r(i, k-1)$ facilitates logical right shift or right rotation operation.

For $i \leftarrow (2^{k-1}-1)$ to 0, the second input of the additional Fredkin gates are connected Q_r from right, remaining B_r = 0, except the last stage; for the last stage B_r= I_7, if S_{RS}=1 else B_r=0. This facilitates the arithmetic right shift operation by 2^{k-1} bits because when the control signals S_{RS} = 1 and S_{RR} = 0 we will have $Q_r(i, j) = B_r(i, j)$. The input sequences for the third input of the additional Fredkin gates are set to $C_r(i, k-1) = i_m$, for $i \leftarrow 2^{k-1}$ to 0, and $m = (2^{k-1}-1)$ to 0. This facilitates the right rotate operation by 2^{k-1} bits because when the control signals S_{RS} = 0 and S_{RR} = 1 we will have $Q_r(i, k-1) = C_r(i, k-1)$ and the logical right shift operation, when S_{RS} = 0 and S_{RR} = 0. The original inputs

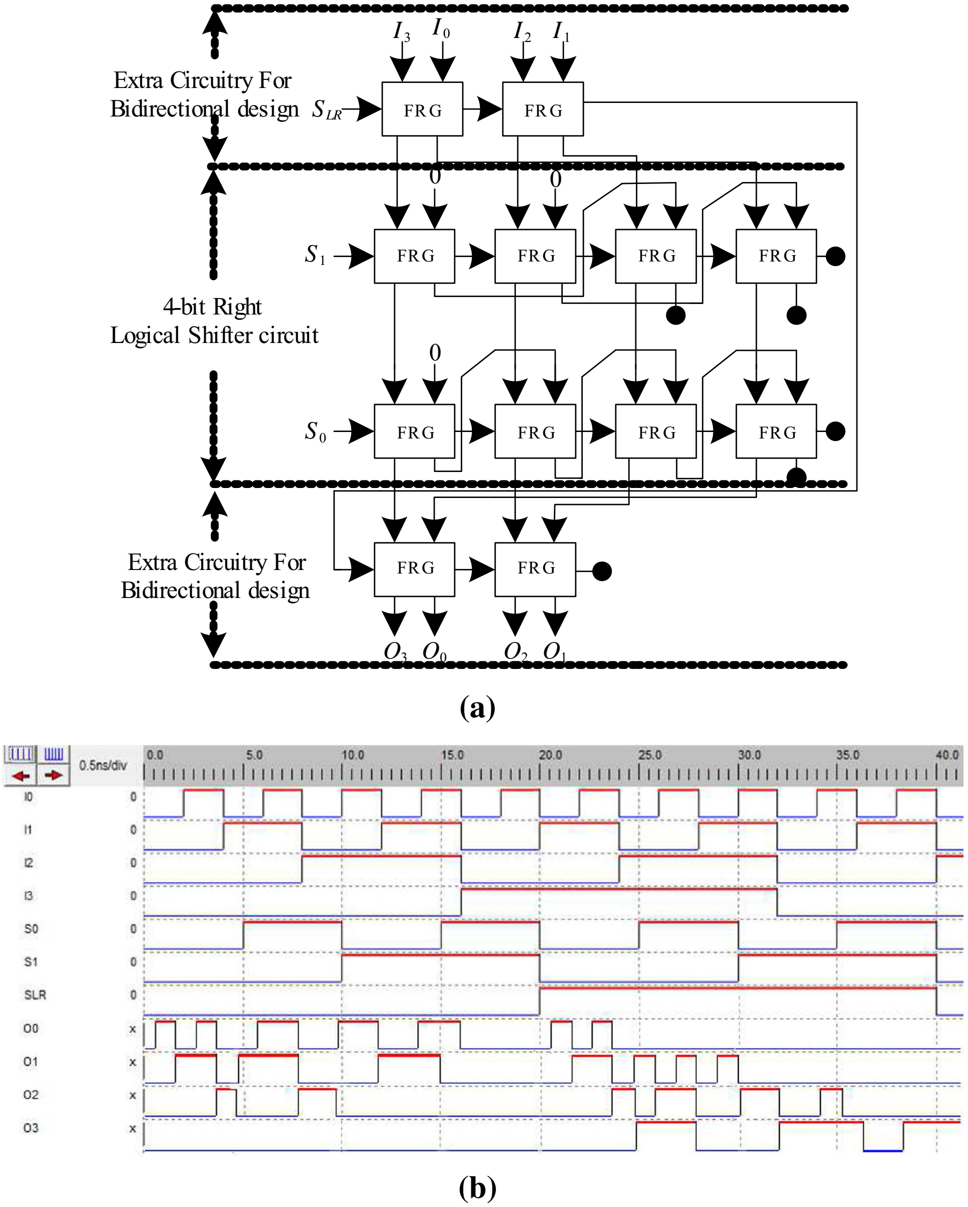


Fig. 15 Proposed (4, 2) bidirectional logical shifter. **a** Architectural block diagram. **b** Corresponding timing diagram

$I_{n-1}, I_{n-2}, I_{n-3}, \ldots, I_2, I_1, I_0$ will work as the inputs to the chain of n Fredkin gates as used in the design methodology for (n, k) reversible logical right shifter. These original inputs and $Q_r(i, k-1)$ outputs of the additional Fredkin gates are assigned to the chain of the n Fredkin gates as follows:

$A(i, k-1) = S_{k-1}$, **for** $i \leftarrow (n-1)$ to 0

$B(i, k-1) = i_i$, **for** $i \leftarrow (n-1)$ to 0

$C(i, k-1) = Q_r(m, k-1)$, **for** $i \leftarrow (n-1)$ to $((n-1) - 2^{k-1} + 1)$,

and $m = 2^{k-1}$ to 0

$C(i, k-1) = i_m$, **for** $i \leftarrow ((n-1) - 2^{k-1})$ to 0, and $m = (n-1)$ to 2^{k-1}.

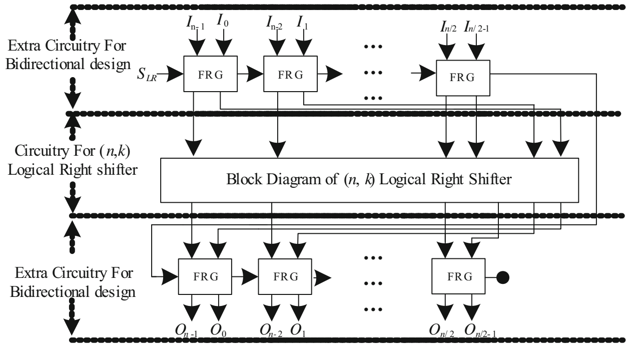

Extra Circuitry For Bidirectional design
I_{n-1} I_0 I_{n-2} I_1 $I_{n/2}$ $I_{n/2-1}$
S_{LR}
FRG
Circuitry For (n,k) Logical Right shifter
Block Diagram of (n, k) Logical Right Shifter
Extra Circuitry For Bidirectional design
O_{n-1} O_0 O_{n-2} O_1 $O_{n/2}$ $O_{n/2-1}$

Fig. 16 Proposed (n, k) bidirectional logical shifter

Table 2 Operations on a (n, k) universal right shifter

S_{RS}	S_{RR}	Operation
0	0	Logical right shift
0	1	Right rotation
1	0	Arithmetic right shift

Algorithm 6: Algorithm for the proposed reversible fault tolerant (n, k) universal right shifter, ***URS(n, k, F2G, FRG)***

```
Input  : Data input set $I(I_0, I_1, ..., I_{n-1})$ and Control input set
         $S(S_0, S_1, ..., S_{k-1})$, $S_{RS}$, $S_{RR}$ and Fredkin gate ($FRG$)
Output: Reversible fault tolerant $(n, k)$ universal shifter circuit
begin
    $i = input,\ \ o = output$
    for $j \leftarrow 0$ to $log_2 n$ do
        $I_{n-1} \rightarrow first.i_j.frg$
        $S_{RS} \rightarrow second.i_j.frg$
        $third.o.frg \rightarrow first.i_j.f2g$
        $first.o.f2g \rightarrow first.i_{j+1}.f2g$
        $0 \rightarrow remaing\ input\ of\ all\ gates$
    for $k \leftarrow 0$ to $log_2 n$ do
        for $l \leftarrow n-1$ to $0$ do
            $I_l \rightarrow first.i_l.f2g$
            $0 \rightarrow second.i_l.f2g$ & $third.i_l.f2g$
        for $m \leftarrow (n-1)/2$ to $0$ do
            $S_{RR} \rightarrow first.i_m.frg$
            if $m \leq k/2$ then
                $0 \rightarrow second.i_m.frg$
            else
                $second.f2g_j.o \rightarrow second.i_m.frg$
                $third.f2g_k.o \rightarrow third.i_m.frg$
        for $p \leftarrow n-1$ to $0$ do
            $S_j \rightarrow first.frg_p.i$
            $second.f2g_l.o \rightarrow second.frg_p.i$
            if $p \leq (n-1)/2$ then
                $second.frg_m.o \rightarrow third.frg_p.i$
            else
                $third.f2g_k.o \rightarrow third.frg_p.i$
        if $k = log_2 n$ then
            return   for $q \leftarrow n-1$ to $0$ do
                $second.o.FRG\ as\ Desire\ Output$
        else
            all remaining Outputs as garbage
```

Control input for all the Fredkin gates in the 1^{st} stage is set to S_{k-1}, which facilitates either shift or no shift operation. Except these changes in the input-output mapping working procedure of this stage is similar to the above stage thus is not explained separately. Similarly, one can design the other stages of the reversible universal right shifter. Table 3 represent the summarized output of the architecture shown in Fig. 17 for the initial inputs $I_7, I_6, I_5, I_4, I_3, I_2, I_1, I_0$.

Lemma 11 *Let UGT be the required number of gates for the (n, k) fault tolerant universal shifter, where n is the number of data inputs and* $k = log_2 n$*. Then,*

$$UGT = \frac{1}{4}(5n + 8nk)$$

Proof A (n, k) reversible fault tolerant universal shifter has k stages and n input bits. According to our design procedure each stage requires $(2n - 2^j)$ Fredkin gates. An additional Fredkin gate is also required to determine the type of shifts or rotations. For efficient copying of output, we require n Feynman double gates at each stage. We also need $n/4$ Feynman double gates for extra circuitry, which determines type of shifting and rotation. Therefore, total number of gates required for a (n, k) fault tolerant universal shifter is

$$\sum_{j=0}^{k-1}(2n - 2^j) + 1 + \sum_{j=0}^{k-1} n + \frac{n}{4} = \frac{1}{4}(5n + 8nk)$$

□

Lemma 12 *Let UGO be the number of garbage outputs for the* (n, k) *fault tolerant universal shifter, where* n *is the number of input bits and* $k = log_2 n$*. Then,*

$$UGO = n(2k + 1) + (k + 2)$$

Proof A (n, k) fault tolerant universal shifter requires $(\sum_{j=0}^{k-1}(2n - 2^j) + 1)$ Fredkin gates and $(nk + n/4)$ Feynman double gates. According to our design procedure, each FRG produces one garbage output and and the remaining gates combinedly produce $n/2$ garbage outputs. So, total garbage outputs produced by an (n, k) fault tolerant universal shifter is,

$$\sum_{j=0}^{k-1}(2n - 2^j) + nk + \frac{n}{4} = n(2k + 1) + (k + 2)$$

□

Lemma 13 *Let* α, β *and* γ *be the hardware complexity for two-input Ex-OR, AND, NOT calculations, respectively. Also let,* n *is the number of input bits and* $k = log_2 n$*. Then the propsed universal shifter can be realized with* $\frac{1}{2}(11n + 14nk)$ *quantum cost and* $(n/2(8k+5)\alpha + 4n(k+1)\beta + n(k+1)\gamma)$ *hardware complexity .*

Proof A (n, k) fault tolerant universal shifter requires $(\sum_{j=0}^{k-1}(2n - 2^j) + 1)$ Fredkin gates and $(nk + \frac{n}{4})$ Feynman double gates. Quantum cost of each Fredkin and Feynman double gate is 5 and 2 respectively; and their hardware complexity is $2\alpha + 4\beta + \gamma$ and 2α, respectively. So, total quantum cost of the proposed (n, k) reversible fault tolerant universal

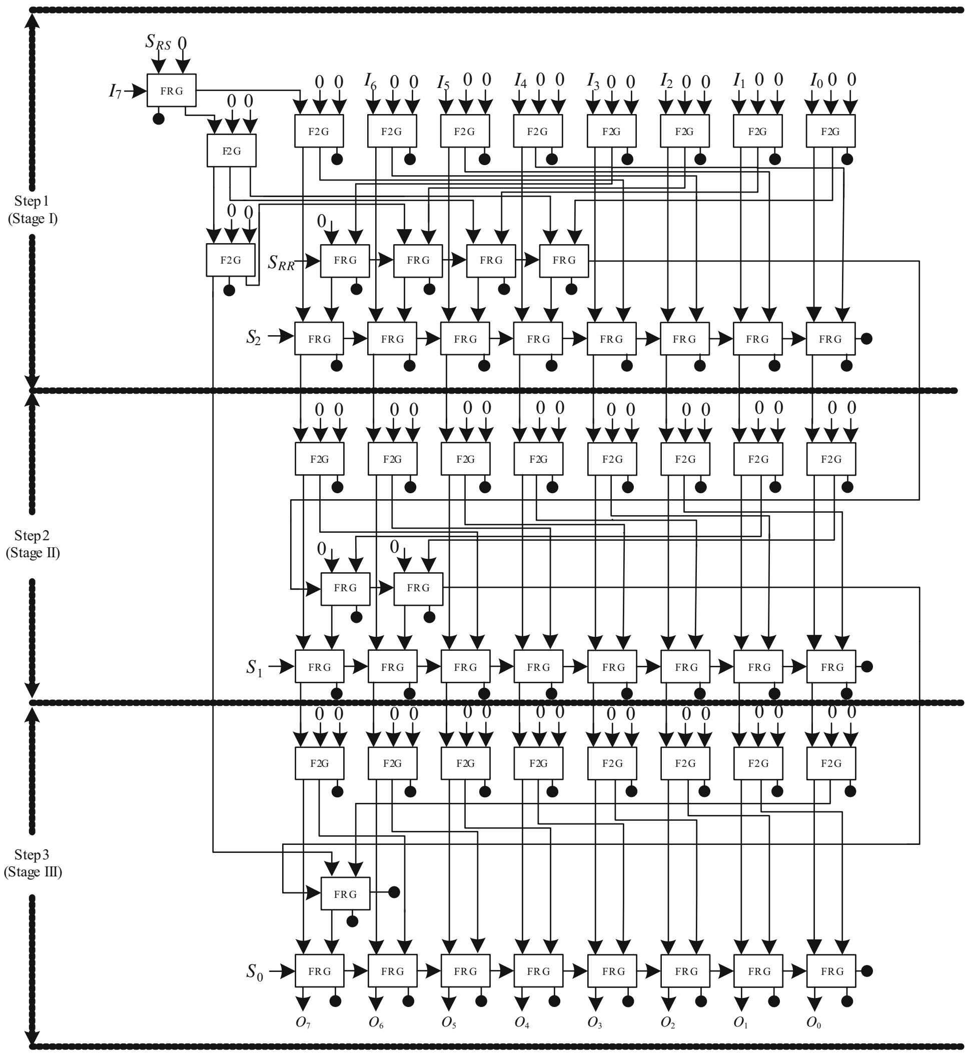


Fig. 17 Proposed (8, 3) reversible fault tolerant logarithmic universal shifter (circuit for universal right shift)

Table 3 Operations on a (8, 3) universal right shifter

Function	Arithmetic shift ($S_{RS} = 1$, $S_{RR} = 0$)	Logical shift ($S_{RS} = 0$, $S_{RR} = 0$)	Rotate operation (S_{RS}=0, $S_{RR} = 1$)
Stage:I $S_2 = 1$	$I_7 I_7 I_7 I_7 I_7 I_6 I_5 I_4$	$0000 I_7 I_6 I_5 I_4$	$I_3 I_2 I_1 I_0 I_7 I_6 I_5 I_4$
Stage:II $S_1 = 0$	$I_7 I_7 I_7 I_7 I_7 I_7 I_7 I_6$	$000000 I_7 I_6$	$I_5 I_4 I_3 I_2 I_1 I_0 I_7 I_6$
Stage:III $S_0 = 0$	$I_7 I_7 I_7 I_7 I_7 I_7 I_7 I_6$	$000000 I_7 I_6$	$I_5 I_4 I_3 I_2 I_1 I_0 I_7 I_6$

shifter is $5\sum_{j=0}^{k-1}(2n - 2^j) + 1) + 2(nk + n/4) = \frac{1}{2}(11n + 14nk)$ and the total hardware complexity is $2(\sum_{j=0}^{k-1}(2n - 2^j)+1)+(nk+n/4)\alpha+4\sum_{j=0}^{k-1}(2n-2^j)+1)\beta+\sum_{j=0}^{k-1}(2n-2^j) + 1)\gamma = n/2(8k + 5)\alpha + 4n(k + 1)\beta + n(k + 1)\gamma$ □

Algorithm 7 demonstrates the design procedure of the proposed bidirectional universal shifter. The extra control input S_{LR} defines shift-rotate direction. According to the algorithm the architecture of proposed bidirectional universal shifter is shown in Fig. 18. The figure shows that there is an additional Fredkin gate after the (n, k) universal right shifter. The additional gate preserves the sign bit when S_{LA} is set to high. The sign bit of the input data is copied for arithmetic right shift using additional Feynman double gate. Table 4 shows the operational procedure of the (n, k) reversible fault tolerant universal bidirectional shifter.

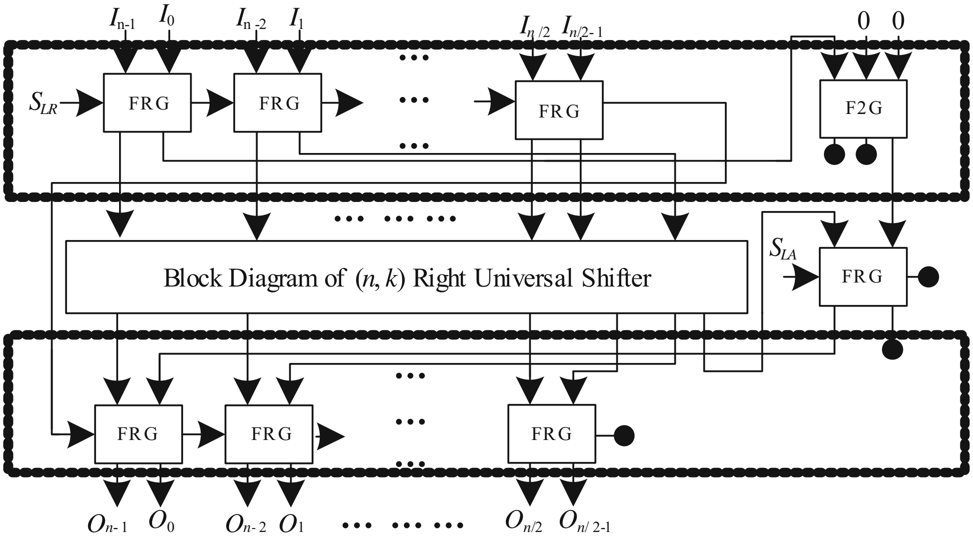


Fig. 18 Proposed (n, k) reversible fault tolerant bidirectional universal shifter

Algorithm 7: Algorithm for the proposed reversible fault tolerant bidirectional universal shifter, ***BUS(n, k+4, F2G, FRG)***

Input : Data input set $I(I_0, I_1, ..., I_{n-1})$ and Control input set $S(S_0, S_1, ..., S_{k-1})$, S_{LR}, S_{RS}, S_{RR}, S_{LA}, Fredkin gate (FRG) and Feynman Double gate $F2G$

Output: Reversible fault tolerant bidirectional universal shifter's circuit for n-data bits

```
begin
    $i = input,\ \ o = output, k = 0, m = 0$
    for $j \leftarrow (n-1)/2$ to 0 do
        $S_{LR} \rightarrow first.i.FRG, I_j \rightarrow second.i.FRG$
        $I_k \rightarrow third.i.FRG, k = k + 1$
    Call URS(n, k+2, F2G, FRG)
    $third.FRG_{n/2}.o \rightarrow first.i.F2G$
    $0 \leftarrow first\ \&\ second\ .i.F2G$
    $S_{LA} \rightarrow first.i.FRG$
    $URS(n, k+2, F2G, FRG).o_n \rightarrow second.i.F2G$
    $third.O.F2G \leftarrow third.i.FRG$
    for $l \leftarrow (n-1)/2$ to 0 do
        $S_{LR} \rightarrow first.i.FRG_l$
        $URS(n, k+2, F2G, FRG).o_l \rightarrow second.i.FRG_l$
        if $l = (n-1)/2$ then
            $frist.FRG.o \rightarrow third.i.FRG_l$
        else
            $URS(n, k+2, F2G, FRG).o_m \rightarrow second.i.FRG_l$
            $m = m + 1$
    return for $all\ bottom\ level\ FRG$ do
        $first.o.FRG\ \&\ second.o.FRG$ as Desire Output
        and
        Remaining Output as Garbage Output
```

Table 4 Operations on bidirectional universal shifter

S_{LR}	S_{RS}	S_{RR}	Operation performed
1	0	0	Logical left shift
0	0	0	Logical right shift
1	0	0	Arithmetic left shift
0	1	1	Arithmetic right shift
1	0	1	Left Rotation
0	0	0	Right Rotation

The working procedure of the Algorithm 7 is as follows: In the first stage, the input data $I_{n-1}, I_{n-2}, \ldots\ldots\ldots, I_1, I_0$ is provided to the $n/2$ Fredkin gates, which are controlled by the control signal S_{LR}. When S_{LR} is set to high, the input data is reversed. In the second stage, the outputs generated by the $n/2$ Fredkin gates are used as the inputs to the $(n, k+2)$ universal right shifter. The $(n, k+2)$ universal right shifter performs the desire right shifts depending on its inputs S_{RS} and S_{RR}. The stage III consists of a Fredkin gate, which is needed only if the arithmetic left shift operation is performed. This additional Fredkin gate helps in changing the 0^{th} bit of the output generated by the proposed $(n, k+2)$ universal right shifter if $S_{LR} = 1$ and $S_{RS} = 1$. Thus, when this condition is satisfied, the Fredkin gate will change the 0^{th} bit of the output generated by the (n, k) universal right shifter to the sign bit, i.e., I_{n-1}. If this stage is used, then the outputs of the final stage will be arithmetic left shifted result of the original inputs. The stage IV, consists of $n/2$ Fredkin gates as shown in Fig. 18, which are controlled by the control signal S_{LR}. If $S_{LR} = 1$ then the inputs passed to this stage are reversed by the $n/2$ Fredkin gates to produce the final output as logically left shifted input data or arithmetically left shifted input data or left rotated input data. Otherwise, the final outputs will be the same as the outputs of the Stage III, which is logically right shifted input or arithmetically right shifted input or right rotated input.

Lemma 14 *A reversible fault tolerant bidirectional universal shifter for n data bits can be realized with* $1/4(9n +$

Table 5 Comparative study of reversible unidirectional logarithmic barrel shifters

n,k	Proposed rotator unidirectional				Existing rotators unidirectional [21]					Proposed logical shifter unidirectional				
	GO	QC	PDP	HC	GO	QC	CI	PDP	HC	GO	QC	CI	PDP	HC
4,2	2	25	0.10	10α 20β 5γ	6	34	4	0.19	16α 24β 6γ	8	40	3	0.19	16α 32β 8γ
8,3	3	85	0.76	34α 68β 17γ	19	116	16	1.56	56α 80β 20γ	24	120	7	1.20	48α 96β 24γ
16,4	4	245	4.41	98α 196β 49γ	52	328	48	8.80	160α 224β 56γ	64	320	15	6.08	128α 256β 64γ
32,5	5	645	22.57	258α 516β 129γ	133	848	128	37.37	416α 456β 144γ	160	800	31	28.80	320α 640β 160γ

$8nk+8)$ *reversible fault tolerant gates* $\frac{1}{2}(21n+14nk+14)$ *quantum cost and* $\frac{1}{2}(9n+8nk+8)\ \alpha\ +\ (8n+4nk+4)$ $\beta+$ $(4n+2nk+2)\ \gamma$ *hardware complexity, where n is the number of inputs,* $k = log_2 n$, α, β *and* γ *are the hardware complexity for two-input Exclusive-OR, AND, NOT calculations respectively.*

Proof The proposed bidirectional universal shifter for n data bits is the extension of the proposed universal right shifter with some additional gates. As proved earlier, the universal right shifter can be realized with $\sum_{j=0}^{k-1}(2n-2^j)+nk+\frac{n}{4}$ garbage outputs. Additional gates that are needed to maintain bi-directionality in the universal shifter produce $\frac{n}{2}$ garbage outputs. Thus, total number of garbage outputs produced by the proposed bidirectional universal shifter is, $\sum_{j=0}^{k-1}(2n-2^j)+nk+\frac{n}{4}+\frac{n}{2}=1/4(n+2nk+5)$.

According to our previous discussion, quantum cost of each Fredkin and Feynman double gate are 5 and 2, respectively and their hardware complexity are $2\alpha+4\beta+\gamma$ and 2α, respectively. So, total quantum cost of the proposed (n, k) bidirectional universal shifter is, $5\sum_{j=0}^{k-1}(2n-2^j)+5+5(n+1)+2\sum_{j=0}^{k-1}n+\frac{n}{2}+2=\frac{1}{2}(21n+14nk+14)$ and total hardware complexity of the proposed bidirectional universal shifter is $(\sum_{j=0}^{k-1}(2n-2^j)+1+(n+1))(2\alpha+4\beta+\gamma)+2(\sum_{j=0}^{k-1}n+\frac{n}{4}+1)\alpha\frac{1}{2}(9n+8nk+8)\alpha+(8n+4nk+4)\beta+(4n+2nk+2)$. □

6 Performance evaluation of the proposed methods

In this section, we evaluate the performance of the proposed designs with respect to existing approaches in terms of number of garbage output (GO), quantum cost (QC), constant inputs (CI), hardware complexity (HC), and power delay product (PDP). Garbage output is considered as an important performance evaluation parameter, since it represents the number of unwanted overheads. As mentioned in Sect. 2.3, heavy price is paid off for each garbage output [5,6]. Thus, the circuit with fewer garbage outputs is desirable. In addition, the constant input is another major overhead, as it increases the number of bits for the realization. On the other hand, hardware complexity represents the required number of basic operations for a circuit. A reversible circuit with fewer hardware complexity means a circuit with lower operations that leads to the lower quantum cost. The reversible circuits with the fewer quantum cost are considered as beneficial as a quantum circuit with many qubits is very difficult to realize. In addition, power delay product is directly correlated with the energy efficiency specifically for the transistor based circuits. The circuit with lower power delay product is also desirable [26,27]. In the following tables we present the power delay product in normalized form. The standard normalization procedure based on frequency (input) has been used for this purpose.

To our best knowledge, there are two works [21,22] on reversible unidirectional barrel shifters. The work [21] greatly outperforms the work of [22]. Thus, we compare our proposed unidirectional work only with the method of [21] which is shown in Table 5[2]. Table 6 represents the performance of the proposed bidirectional shifters with the existing bidirectional shifter [23]. However, it is not possible for us to evaluate the performance of the proposed universal shifter with respect to the existing design [29] as the existing design presents only a structure of (4, 2) shifter without any further explanation. The minimization in the proposed circuit

[2] The proposed unidirectional rotators don't require any constant input. Thus, we omit the column of CI for the proposed rotator's part.

Table 6 Comparative study of reversible bidirectional logarithmic barrel shifters

n,*k*	Proposed rotators bidirectional					Proposed logical shifter bidirectional					Existing shifter bidirectional [23]				
	GO	QC	CI	PDP	HC	GO	QC	CI	PDP	HC	GO	QC	CI	PDP	HC
4,2	4	32	1	0.18	14α 24β 6γ	6	60	1	0.23	24α 48β 12γ	15	79	10	0.94	29α 52β 13γ
8,3	6	94	2	1.11	40α 72β 18γ	11	160	2	1.41	64α 128β 32γ	32	195	26	3.56	92α 132β 33γ
16,4	8	256	3	5.59	106α 200β 50γ	20	400	3	7.20	160α 320β 80γ	73	475	66	13.95	216α 324β 81γ
32,5	10	658	4	26.22	268α 520β 130γ	37	960	4	34.92	384α 768β 192γ	170	1131	62	94.85	544α 772β 193γ

is achieved at the low level transistors which is reflected in the above performance evaluation tables. The performance evaluation tables also proved that the proposed reversible fault tolerant designs perform much better than the existing non-fault tolerant designs in terms of all the performance evaluation parameters. Form these tables we also find that the performance of the proposed circuits increases as the circuit grows (more inputs).

7 Conclusions

In this paper, we presented the design methodologies of the reversible fault tolerant unidirectional, bidirectional and universal barrel shifters. The proposed circuits have the capability of detecting errors at its primary outputs, which are not available in the existing circuits [21,23,29]. We have also proposed two lower bounds on the numbers of garbage outputs and constant inputs of the reversible barrel shifters in Theorems 1 and 2, respectively. Example 1 has evidenced that the proposed unidirectional rotator is constructed with the optimum garbage output and constant input. The proposed unidirectional rotator has further been used to build the base structures of the other proposed shifters. We also present design algorithm for all the proposed reversible fault tolerant barrel shifters. The presented algorithms can be used to design the respective fault tolerant barrel shifters for any n where, n is number of data bits and $n = 2^m, m \geq 2, m \epsilon Z$. Moreover, in order to simulate the proposed circuits, we implement each individual gates of the proposed shifters using low power MOS transistors. The simulation results showed that the proposed shifters work accurately. Finally we have presented detail comparative results in Tables 5 and 6, respectively. The comparative results show that the proposed designs perform better than all its counterparts and are much more scalable.

References

1. Maxwell, J.C.: Theory of Heat, 4th edn. Longmans, Green and co., London (1875)
2. Szilard, L.: On the decrease of entropy in a thermodynamic system by the intervention of intelligent beings maxwell's demon 2: entropy, classical and quantum information computing. Zeitschrift fur Physik **53**, 840–856 (1929)
3. Landauer, R.: Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **5**, 183–191 (1961)
4. Bennett, C.H.: Logical reversibility of computation. IBM J. Res. Dev. **17**, 525–532 (1973)
5. Shamsujjoha, M., Hasan Babu, H.M., Jamal, L.: Design of a compact reversible fault tolerant field programmable gate array: a novel approach in reversible logic synthesis. Microelectron. J. **44**(6), 519–537 (2013)
6. Shamsujjoha, M., Hasan Babu, H.M.: A low power fault tolerant reversible decoder using mos transistors. In: Proceedings of the 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems, pp. 368–373, IEEE Computer Society (2013)
7. Shamsujjoha, M., Hasan Babu, H.M., Jamal, L., Chowdhury, A.R.: Design of a fault tolerant reversible compact unidirectional barrel shifter. In: Proceedings of the 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems, pp. 103–108, IEEE Computer Society (2013)
8. Moore, G.E.: Cramming more components onto integrated circuits. Electronics **38**, 114–117 (1965)
9. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information, 10 Anniversary edn. Cambridge University Press, Cambridge (2011)
10. Parhami, B.: Fault-tolerant reversible circuits. In: Fortieth Asilomar Conference on Signals, Systems and Computers, pp. 1726–1729 (2006)
11. Pastawski, F., Yoshida, B.: Fault-tolerant logical gates in quantum error-correcting codes. Phys. Rev. A **91**(1), 012305 (2015)
12. Ahmed, T., Sarker, A., Sharif, M.I., Rashid, S.M.M., Rahman, M.A., Hasan Babu, H. M.: A novel approach to design a reversible shifter circuit using DNA. In: 2013 IEEE International SOC Conference, pp. 256–261 (2013)
13. Akbar, E.P.A., Haghparast, M., Navi, K.: Novel design of a fast reversible wallace sign multiplier circuit in nano-technology. Microelectron. J. **42**(8), 973–981 (2011)

14. Taraphdar, C., Chattopadhyay, T., Roy, J.N.: Machzehnder interferometer-based all-optical reversible logic gate. Opt. Laser Technol. **42**(2), 249–259 (2010)
15. Mahammad, S.N., Veezhinathan, K.: Constructing online testable circuits using reversible logic. IEEE Trans. Instrum. Meas. **59**, 101–109 (2010)
16. Katz, E., Privman, V.: Enzyme-based logic systems for information processing. Chem. Soc. Rev. **39**, 1835–1857 (2010)
17. Miller, D.M., Maslov, D., Dueck, G.W.: A transformation based algorithm for reversible logic synthesis. In: Proceedings of the 40th annual Design Automation Conference, pp. 318–323 (2003)
18. Chaki, N., Chaki, R.: A scalable barrel shifter topology for all optical networks. In: Proceedings of the Fourth IASTED Asian Conference on Communication Systems and Networks, pp. 304–309, ACTA Press (2007)
19. Voyiatzis, D. Gizopoulos, Paschalis, A.: Accumulator-based test generation for robust sequential fault testing in DSP cores in near-optimal time. In: IEEE Transactions on Very Large Scale Integration (VLSI) Systems vol. 14, pp. 1079–1086 (2005)
20. Beerel, P. A., Kim, S., Yeh, P.C., Kim, K.: Statistically optimized asynchronous barrel shifters for variable length co-decs. In: Proceedings of the International Symposium on Low Power Electronics and Design, pp. 261–263 (1999)
21. Hashmi, I., Hasan Babu, H.M.: An efficient design of a reversible barrel shifter. In: 23rd International Conference on VLSI Design, 9th International Conference on Embedded Systems, pp. 93–98, IEEE Computer Society (2010)
22. Gorgin, S. Kaivani, A.: Reversible barrel shifters. In: ACS/IEEE International Conference on Computer Systems and Applications, pp. 479–483, IEEE Computer Society (2007)
23. Kotiyal, S., Thapliyal, H., Ranganathan, N.: Design of a reversible bidirectional barrel shifter. In: 11th IEEE Conference on Nanotechnology, pp. 463–468 (2011)
24. Vasudevan, D.P., Lala, P.K., Parkerson, J.P.: Online testable reversible logic circuit design using nand blocks. In: IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, pp. 324–331, IEEE Computer Society (2004)
25. Thapliyal, H., Vinod, A.P.: Design of reversible sequential elements with feasibility of transistor implementation. In: International Symposium on Circuits and Systems (ISCAS 2007), pp. 625–628, IEEE (2007)
26. Vujkovic, M., Sechen, C.: Optimized power-delay curve generation for standard cell ics. In: Proceedings of the 2002 IEEE/ACM International Conference on Computer-aided Design, pp. 387–394, ACM (2002)
27. Miyaoka, Y., Kataoka, Y., Togawa, N., Yanagisawa, M., Ohtsuki, T.: Area/delay estimation for digital signal processor cores. In: Proceedings of the 2001 Asia and South Pacific Design Automation Conference, pp. 156–161, ACM (2001)
28. Mathew, J., Singh, J., Taleb, A.A., Pradhan, D.K.: Fault tolerant reversible finite field arithmetic circuits. In: Proceedings of the 14th IEEE International On-Line Testing Symposium, pp. 188–189, IEEE Computer Society (2008)
29. Ravish Aradhya, H.V., Lakshmesha, J., Muralidhara, K.N.: Design optimization of reversible logic universal barrel shifter for low power applications. Int. J. Comput. Appl. **40**(15), 26–34 (2012)