

Optimized Logarithmic Barrel Shifter in Reversible Logic Synthesis

Sajib Kumar Mitra and Ahsan Raja Chowdhury

Dept. of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh

Email: saajibmitra.csedu@yahoo.com, farhan717@yahoo.com

Abstract—Reversible logic attains the dominance in the realm of overwhelming research in logic synthesis and also has the significance in the context of quantum computing because of loss-less information processing. Due to low power dissipation, researchers are first designing smaller components with reversible gates, that eventually lead to design reversible computer. In this paper, we propose a robust architecture of logarithmic barrel shifter that performs bidirectional arithmetic and logical shifting, including rotate operation. Incorporating fault tolerance capability, the circuit is designed very efficiently that exhibits superior performance over state-of-the-art design methods in terms of minimum number of gates, garbage outputs, ancilla inputs, quantum cost, delay and others cost factors.

I. INTRODUCTION

According to Moore's law [1], the transistor count is doubled in every two years and it's performance continue to dominated until semiconductor circuits reach its physical limit. Further, irreversible logic computation necessarily generates $kT \cdot \log_2$ joules energy per bit information loss [2], where k is Boltzmann's constant and T is the absolute room temperature. On the other hand, Reversible Computing does not erase information, performs multiple operations in single cycle and is easily compliant to Fault Testing [3]. In terms of hardware design, binary computation is preferred over decimal computation which is due to the ease of building hardware for binary number. However, approximation is required while performing computation in binary hardware because of the complexity incur to represent most of the fractional decimals numbers. Shifting is considered as the most powerful bit-wise operation amongst any high speed computing system. Two popular types of shifting circuits: Logarithmic [4] and Array shifters [5]. In general, Logarithmic barrel shifter is widely used because of its simplicity and robustness. There exists several methodologies for designing barrel shifter in reversible domain, albeit each of them possess limitation to be considered as full-fledged reversible Barrel shifter. This paper attempts to overcome the limitations of the existing design approaches and improve the accuracy of the circuit.

Rest of the paper is organized as follows. A brief discussions of reversible logic, fault tolerant mechanism and shifting methodologies are presented in Section II, including the existing works on designing reversible barrel shifter. In Section III, the proposed optimized architecture of reversible logarithmic barrel shifter has been described along with the theoretical underpinning of the proposed design. Performance evaluation of the proposed design compared to existing design is presented in Section IV. Section V concludes the paper.

II. LITERATURE REVIEW

In this section, preliminaries in reversible logic and the standard performance metrics in reversible domain are discussed. This section also highlights the pros and cons of the existing designs of reversible logarithmic barrel shifter.

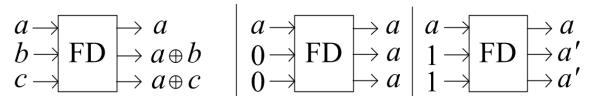
A. Reversible Logic

In 1973, Bennett revealed the secret of saving energy by preserving unique mapping between input and output vectors called Logical Reversibility of Computation [6]. A reversible gate has the property of maintaining one-to-one mapping between input and output vectors, while a reversible circuit is composed of reversible gates only. Among the conventional logic gates, NOT operation is the only one which itself is reversible. However, the other conventional logic operations have their reversible counterpart, which is known as $n \times n$ dimensional reversible gate. There exists a 2×2 (e.g., Feynman gate [7]) and several 3×3 reversible gates (e.g., Feynman Double gate [8], Fredkin gate [9], as shown in Fig. 1), which are widely used in designing reversible circuits.

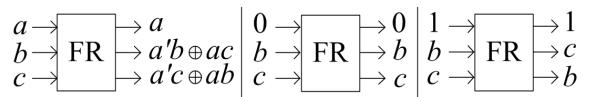
B. Fault Tolerant Technology

Logical reversibility recovers bit loss, although unable to detect bit error in circuit due to any noisy or faulty medium. On the other hand, Fault Tolerant (FT) reversible circuit is capable to prevent error at outputs [8]. The input and corresponding output parities of any FT gate is identical and the gates are used to form higher dimensional reversible FT circuits. Let I_v ($I_{n-1}, I_{n-2}, I_{n-3}, \dots, I_0$) and O_v ($O_{n-1}, O_{n-2}, O_{n-3}, \dots, O_0$) are the input and output vectors of an $n \times n$ reversible FT circuit, then the following equation must be preserved:

$$I_{n-1} \oplus I_{n-2} \oplus \dots \oplus I_0 = O_{n-1} \oplus O_{n-2} \oplus \dots \oplus O_0$$



(a) Controlling Outputs of Feynman Double Gate (FD)



(b) Fredkin Gate (FR) performs Swapping Inputs

Fig. 1. Two 3×3 Reversible Gates: (a) Feynman Double Gates and (b) Fredkin Gate

C. Shifting Methodology

Shifting is used in various arithmetic operations, such as multiplication, division, switching network, etc [10]. Rotate left/right, Shift logical left/right, arithmetic shifting are the example of shifting techniques, which are shown in Fig. 2.

D. Quantum Realization of Reversible Circuit

In Quantum Computing, unit of data is called qubit and the value of qubit is the superposition of $|0\rangle$ and $|1\rangle$. Every quantum operation is reversible if it is represented by a unitary matrix which is used to multiply the state of qubit(s) that produces output [11]. The comparative quantum realization of any reversible circuit is used to verify the operability of that circuit.

E. Measurement Metrics

The trend of the this research work is to minimize the cost factors of proposed design and speed up the performance of the overall circuit. Here we describe various cost factors and other metrics that are considered as stand performance metrics in reversible logic design. We also consider the full-adder circuit of Fig. 3, that is referred in various examples.

1) *Gate Count (GA)*: Total number of reversible gates which are used in a circuit is considered as gate count (GA) [12], [21], [22]. For the full-adder circuit in Fig. 3, GA is 4, as the circuit uses 4 gates to realize the function of a full-adder.

2) *Garbage Count (GB)*: In reversible Computation, unused output is called garbage and the garbage count is the total number of garbage outputs generated in a reversible circuit [13]. The full adder in Fig. 3 generates 3 garbage outputs.

3) *Hardware Complexity (HC)*: Total Number of two-input Ex-OR, AND and single input NOT operations, represented by α , β , and γ , respectively, in the output functions of any reversible circuit is called Hardware Complexity.

4) *Quantum Cost (QC)*: Quantum cost of any reversible circuit is the total number of 2×2 quantum gates that are used to represent equivalent quantum circuit [11]. The Quantum Cost of the full adder circuit in Fig. 3 is 11 (summing up the individual quantum cost of FD and NFT gates).

5) *Ancilla Inputs (AI)*: Total number of constant inputs in any reversible circuit is referred as Ancilla Inputs [14]. Fig. 3 shows that the total number of Ancilla Inputs is 2.

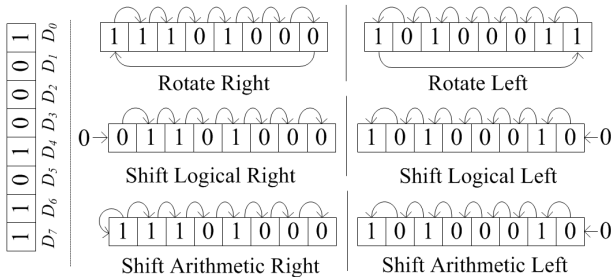


Fig. 2. Single bit left/right rotating, logical shift and arithmetic shift operations on 8-bit data (D_7 to D_0)

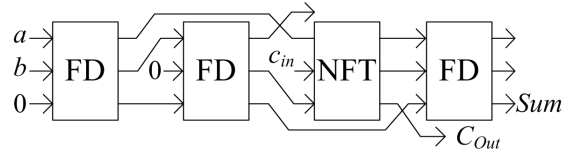


Fig. 3. Reversible Fault tolerant Full Adder circuit using NFT and FD [13]

6) *Delay (DL)*: Maximum number of gates that are placed in the path from any input to any output is called Delay (critical path delay or DL) [13]. In Fig. 3, the delay of Full adder is 4.

F. In Vogue Designs of Reversible Barrel Shifters

This section briefly outlines the existing design methodologies of Reversible Barrel Shifters.

Gorgin and Kaivani [15] proposed the novel architecture of reversible logarithmic barrel shifter using Fredkin gate (FR) as multiplexer to shift input data bits and Feynman gates (FG) to recover fan-out problem. However, each multiplexer node generates an extra output and every copying operation (rather using fan-out) needs ancilla bit to Feynman gate, which increases the total number of garbage outputs and ancilla inputs. Another generalized architecture of unidirectional (n , k) rotator has been proposed in [16] that reduced the number of excessive use of Feynman gates. Both the designs act as unidirectional barrel shifter and perform rotation rather doing arithmetic and logical shifting operations. Kotiyal et al. [17] proposed the utmost design of bidirectional barrel shifter capable of performing both arithmetic and logical shift operations. Authors of [18] designed the first fault tolerant reversible barrel shifter using Feynman Double gate and Fredkin gate. However, the design approach generate excessive number of garbage outputs to realize the circuit. Recently, Anjaneyulu et al. [19] proposed bidirectional architecture of reversible barrel shifter, which can perform arithmetic/logical shifting as well as rotating also where design used four control bits: (*left*, *rot*, *sra* and *sla*) instead of three (*left*, *sra* and *sla*). Although a compact design of barrel shifter is proposed in [20] which can perform logical shift and rotate operations using minimum number of gates and garbage outputs, due to unidirectional shifting and lacking of shift arithmetic options the method is considered as an incomplete one. Thapliyal et al. proposed another fault tolerant design of barrel shifter based on conservative reversible logic [14], although the internal architecture requires excessive number of Fredkin gates to design the circuit. A comparative analysis on the aforementioned existing methods are shown in Table I.

TABLE I. COMPARISON AMONG THE EXISTING DESIGNS

Existing Designs	Generalization of Quality Factors						BD	RT	FT	Shifting	
	GA	GB	QC	DL	HC	AI				AS	LS
[15]	Y	Y	Y					Y			
[16]	Y	Y		Y				Y			
[17]	Y	Y	Y			Y	Y			Y	Y
[18]								Y	Y		
[19]	Y	Y	Y			Y	Y	Y		Y	Y
[20]	Y	Y	Y	Y	Y	Y		Y	Y		Y
[14]	Y	Y				Y		Y	Y		

BD= Bidirectional Shifting, RT= Rotate Operation, FT= Fault Tolerant ability, AS= Arithmetic Shifting and LS= Logical Shifting

We note that the technique proposed in this paper caters the shortcomings of the methods mentioned in Table I, as well as include the following salient features:

- Generalized structure considering the standard metric
- Bidirectional capability with fault tolerant ability
- On-chip shifting (arithmetic/logical) and rotating

III. PROPOSED DESIGN

The architecture of proposed reversible (n, k) bidirectional fault tolerant arithmetic/logical barrel shifter circuit is presented here.

A. The Method

The construction of proposed design has promoted the rotation unit as the base of the bidirectional logarithmic barrel shifter. Then shift (shift logical) operation has been combined by attaching another operational unit that can replace the value of shifted data bit with 0 (Zero) during the execution of shift operation. The controlling direction of unidirectional shifting circuit has been suited through cascading directional control unit to simplex design. Finally, arithmetic control module has been placed inside the bidirectional shifter before kernel (i.e., rotating unit) and controls the propagation of sign bit (shift arithmetic right operation). The block diagram of proposed architecture, shown in Fig. 4, composed of following parts:

- Central Rotating Unit (CRU)
- Shift-Rotate Control Unit (SRCU)
- Direction Control Unit (DCU) and
- Sign Control Unit (SCU)

CRU is the core part of the proposed design that grabs information through input lines, $(I_{n-1}, I_{n-2}, \dots, I_0)$ and processes to output lines $(O_{n-1}, O_{n-2}, \dots, O_0)$ based on selection inputs $(S_{k-1}, S_{k-2}, \dots, S_0)$. Among the four units, DCU has been divided into two sub-units, i.e., DCU-I and DCU-II, that work concurrently to change the direction of shifted data. SCU is

activated only to propagate right most bit holding adjustment with DCU while performing Shift Arithmetic Right operation. The updated alternate values for particular operations (either arithmetic/logical shifting or rotating) are propagated through SCU which are used by SRCU. Three separate control lines C_{DU} , C_{SU} and C_{RU} are used to operate DCU, SCU and SRCU units, respectively. The construction of reversible bidirectional (n, k) logarithmic barrel shifter can be described through the following algorithm (Algorithm 1):

Algorithm 1. Construction of Reversible Bi-directional (n, k) Arithmetic/Logical Logarithmic Barrel Shifter

Input: Data Inputs $(I_{n-1}, I_{n-2}, \dots, I_0)$, Selection Inputs $(S_{k-1}, S_{k-2}, \dots, S_0)$ and Control Lines (C_{RU}, C_{DU}, C_{SU})

Output: Data Outputs $(O_{n-1}, O_{n-2}, \dots, O_0)$

Start

1. **If** $C_{RU} = 1$ **Then** [SRCU passes unchanged data inputs]
 2. **If** $C_{DU} = 1$ **Then** [DCU-I and DCU-II stay inactive]
 3. Perform Rotate Left Operation
 4. **Else** [DCU-I and DCU-II rotate data inputs by $n/2$ -bit]
 5. Perform Rotate Right Operation
 6. **End If**
 7. **Else** [SRCU swaps data inputs with 0s]
 8. **If** $C_{DU} = 1$ **Then** [DCU units perform Left Shifting]
 9. **If** $C_{SU} = 1$ **Then** [SCU is inactive when $C_{DU}=1$]
 10. Perform Shift Arithmetic Left Operation
 11. **Else**
 12. Perform Shift Logical Left Operation
 13. **End If**
 14. **Else** [DCU-I and DCU-II perform Right Shifting]
 15. **If** $C_{SU} = 1$ **Then** [SRCU propagates I_{n-1} (msb)]
 16. Perform Shift Arithmetic Right Operation
 17. **Else** [SRCU propagates 0s' instead of shifted bit]
 18. Perform Shift Logical Left Operation
 19. **End If**
 20. **End If**
 21. **End If**
- End**

According to above algorithm (Algorithm 1), the states of controlling inputs $(C_{RU}, C_{DU}$ and $C_{SU})$ and corresponding actions can be summarized in Table II. Also the activities of different modules (CRU, SRCU, SCU and DCU) when specific operation has been implemented are shown in Table II.

TABLE II. STATES OF CONTROLLING INPUTS TO PERFORM PARTICULAR SHIFT AND ROTATE OPERATIONS

States of control inputs			Operation	Activities of Modules			
C_{RU}	C_{DU}	C_{SU}		CRU	SRCU	SCU	DCU
1	1	X^\dagger	Rotate Left	✓	✓		✓
1	0	X^\dagger	Rotate Right	✓	✓		✓
0	1	0	Shift Logical Left	✓	✓		✓
0	0	0	Shift Logical Right	✓	✓		✓
0	1	1	Shift Arithmetic Left	✓	✓		✓
0	0	1	Shift Arithmetic Right	✓	✓	✓	✓

$^\dagger X =$ Don't Care [Rotate operation has no dependency on C_{SU}]

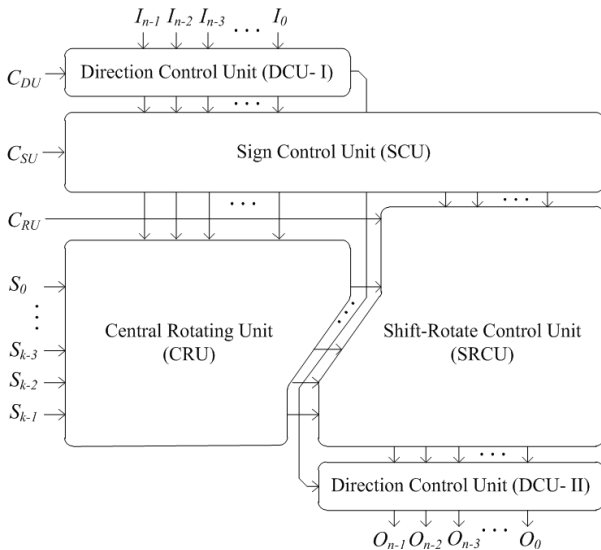


Fig. 4. Block diagram of the proposed (n, k) Reversible Bi-directional Fault Tolerant Logarithmic Barrel Shifter

Rest of the part of this section has discussed the modular organization of reversible bidirectional (n, k) logarithmic barrel shifter and the working principle of every modules.

B. Working Principle of the Proposed Module

We split the entire proposed work into several sub-modules for further analysis. Here, the step-by-step design methodology are described in details.

1) *Construction of Central Rotating Unit:* Central Rotating Unit, designed by using only Fredkin (FR) gates, rotates input data $(I_{n-1}, I_{n-2}, \dots, I_0)$ and emits through output lines $(O_{n-1}, O_{n-2}, \dots, O_0)$. The connection of FRs to data input lines (I_3, I_2, I_1, I_0) , output lines (O_3, O_2, O_1, O_0) and selection lines (S_1, S_0) of a $(4, 2)$ unidirectional rotating circuit is shown in Fig. 5. Each rotating state can be uniquely identified by using the combination of selection inputs $(S_1 S_0)$. In every state of the $(4, 2)$ CRU performs 1-bit rotating operations depending on the value of selection inputs $(S_1 S_0)$. For example, when $S_0 = S_1 = 0$, FR gates in both levels transfer data in crossing mode as projected connectivity of the $(4, 2)$ CRU. The FR gates, which are controlled by S_0 and S_1 pass data directly and crossways, when $S_0 = 1$ and $S_1 = 0$, respectively. Alternately, the FR gates are controlled by S_0 and S_1 passes data crossways and directly, respectively when $S_0 = 0$ and $S_1 = 1$. Finally, every FR gates of CRU pass data directly when selection inputs, $S_0 = S_1 = 1$.

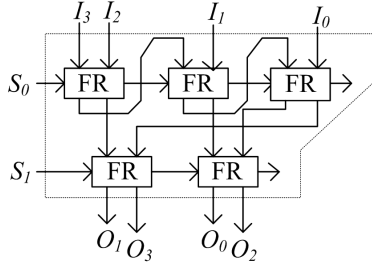


Fig. 5. Designing Reversible $(4, 2)$ Unidirectional Rotator

For the proposed reversible (n, k) Central Rotating Unit, the calculated cost factors (GA_{CRU} , GB_{CRU} , QC_{CRU} , AI_{CRU} and HC_{CRU}) and delay (DL_{CRU}) can be shown as follows:

$$\begin{aligned} GA_{CRU} &= nk - 2^k + 1; & GB_{CRU} &= k; \\ QC_{CRU} &= 5(nk - 2^k + 1); & AI_{CRU} &= 0; \\ HC_{CRU} &= (nk - 2^k + 1)(2\alpha + 4\beta + \gamma) \end{aligned}$$

2) *Designing Unidirectional Shift-Rotator Circuit:* Attachment of Shift-Rotate Control Unit (SRCU) to Central Rotating Unit (CRU) works as unidirectional shift-rotator controlled by C_{RU} . Fig. 6 shows the reversible $(4, 2)$ unidirectional shift-rotator circuit (USR) where vertical FR gates in block Shift-Rotate Control Unit (SRCU) propagates 0 (zero) instead of data inputs (I_3, I_2, I_1) and/or I_0 when C_{RU} be decided LOW. Now, we establish the theoretical underpinning of the proposed design approach with the following theorem and proposition:

Theorem 3.1: Let GA_{SRCU} , QC_{SRCU} and HC_{SRCU} be the number of gates, quantum cost, hardware complexity for (n, k) Shift-Rotate Control Unit where n and k are the number

of data bits and control bits, respectively then:

$$\begin{aligned} GA_{SRCU} &= 2^k + k - 1; & QC_{SRCU} &= 5(2^k + k - 1); \\ HC_{SRCU} &= (2^k + k - 1)(2\alpha + 4\beta + \gamma) \end{aligned}$$

Proof: A (n, k) Shift-Rotate Control Unit has n data bits and k states. Every stage requires single FR gate to select shift/rotate operation and any i^{th} stage needs 2^i FR gates (where $i = 0, 1, 2, \dots, k-1$) to propagate 0s' or data bits. So, required number of FR gates for (n, k) SRCU, $GA_{SRCU} = 2^k + k - 1$. The quantum cost and Hardware complexity of FR is 5 and $(2\alpha + 4\beta + \gamma)$, respectively, which summarizes the quantum cost and hardware complexity of (n, k) Shift-Rotate Control Unit as follows:

$$\begin{aligned} QC_{SRCU} &= 5(2^k + k - 1); \\ HC_{SRCU} &= (2^k + k - 1)(2\alpha + 4\beta + \gamma) \end{aligned}$$

Proposition 3.1: Let GA_{USR} , QC_{USR} and HC_{USR} be the number of gates, quantum cost, hardware complexity of proposed reversible (n, k) Unidirectional Shift-Rotator Circuit where n and k are the number of data bits and control bits, respectively then:

$$\begin{aligned} GA_{USR} &= k(n + 1); & QC_{USR} &= 5k(n + 1); \\ HC_{USR} &= 2k(n + 1)\alpha + k(n + 1)(4\beta + \gamma) \end{aligned}$$

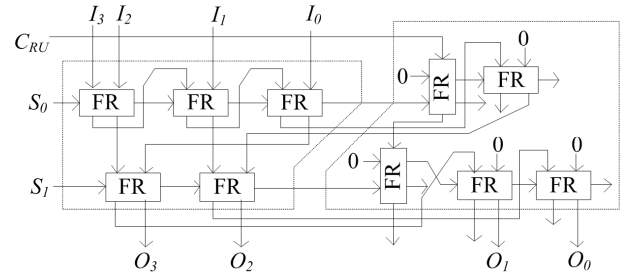


Fig. 6. Proposed design of controlled $(4, 2)$ Unidirectional (left) Fault Tolerant Shift-Rotator Circuit

3) *Migrating Design into Bidirectional Shift-Rotator:* The combination of Direction Control Unit (DCU) and Unidirectional shift-rotator forms the bi-directional shift-rotator circuit performs logical shift operation and rotation in both directions. Direction Control Unit alters data after getting from input lines and before sending to output lines based on the value of control line C_{DU} . Proposed design of reversible $(4, 2)$ Bi-directional Shift-Rotator is shown in Fig. 7 works as a left (right) shifter/rotator when C_{DU} is HIGH (LOW).

Theorem 3.2: Let GA_{DCU} , QC_{DCU} and HC_{DCU} be the number of gates, quantum cost, hardware complexity for (n, k) Direction Control Unit where n and k are the number of data bits and control bits, respectively, then:

$$\begin{aligned} GA_{DCU} &= n; & QC_{DCU} &= 5n; \\ HC_{DCU} &= n(2\alpha + 4\beta + \gamma) \end{aligned}$$

Proof: A (n, k) Directional Control Unit has n data input/output lines and two subunits i.e., DCU-I and DCU-II. For shifting $n/2$ -bit using one control input in DCU-I, the

required number of FR is $n/2$ summaries total number of FR gates, $GA_{DCU} = n$. So, the quantum cost and hardware complexity for realizing (n, k) DCU can be written as follows:

$$QC_{DCU} = 5n; HC_{DCU} = n(2\alpha + 4\beta + \gamma)$$

Proposition 3.2: Let GA_{BSR} , QC_{BSR} and HC_{BSR} be the number of gates, quantum cost and hardware complexity of proposed Reversible (n, k) Bi-directional Shift-Rotator Circuit (BSR) where n and k are the number of data bits and control bits, respectively then:

$$GA_{BSR} = k(n+1) + n; QC_{BSR} = 5k(n+1) + 5n;$$

$$HC_{BSR} = (nk + n + k)(2\alpha + 4\beta + \gamma)$$

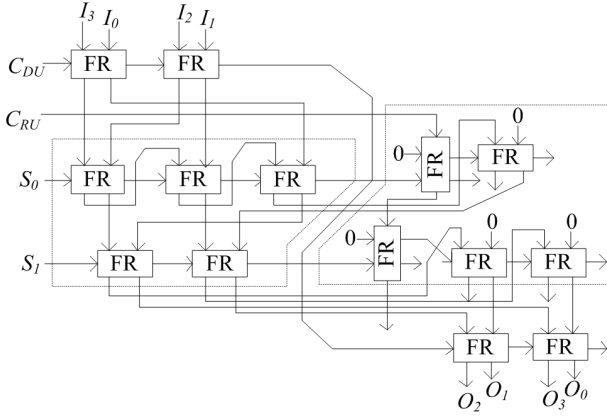


Fig. 7. Proposed design of Reversible Fault Tolerant (4, 2) Bi-directional Shift-Rotator

4) Arithmetic/Logical Shifter and Rotator Circuit Design: Sign or arithmetic shifting (Shift Arithmetic Right) can be performed replacing 0 (Zeros) in shift logical operation by using sign bit (most significant bit). The proposed architecture of $(4, 2)$ reversible bi-directional arithmetic/logical barrel shifter/rotator is shown in Fig. 8. We have converted bi-directional shift-rotate architecture by inserting Sign Control Unit (SCU) which is controlled by C_{SU} and when C_{SU} is HIGH the circuit performs arithmetic shifting.

Theorem 3.3: Let GA_{SCU} , QC_{SCU} and HC_{SCU} be the number of gates, quantum cost and hardware complexity for (n, k) Sign Control Unit where n and k are the number of data bits and control bits, respectively then:

$$GA_{SCU} = \frac{n}{2} + 1, QC_{SCU} = n + 8$$

$$HC_{SCU} = (n + 2)\alpha + 8\beta + 2\gamma$$

Proof: A (n, k) Sign Control Unit has n data bit. According to our design, the required number of FR gates is constant i.e. 2. On the other hand, for generating $(n-1)$ copy of sign bit or 0 (zero) while shifting, required number of FD gates is: $n/2-1$. The quantum cost and hardware complexity of FD is 2 and 2α , respectively. So,

$$GA_{SCU} = n/2 + 1, QC_{SCU} = n + 8$$

$$HC_{SCU} = (n + 2)\alpha + 8\beta + 2\gamma$$

Proposition 3.3: Let GA_{total} , QC_{total} and HC_{total} be the number of gates, quantum cost and hardware complexity of proposed Reversible (n, k) Bi-directional Fault tolerant Arithmetic/Logical Barrel Shifter and Rotator where n and k are the number of data bits and control bits, respectively then:

$$GA_{total} = n(k + 3/2) + k + 1;$$

$$QC_{total} = 5k(n + 1) + 2(3n + 4);$$

$$HC_{total} = (2nk + 2k + 3n + 2)\alpha + (nk + k + n + 2)(4\beta + \gamma)$$

Proposition 3.4: Let GB_{total} , AI_{total} and DL_{total} be the number of garbage outputs, ancilla inputs and delay of proposed Reversible (n, k) Bi-directional Fault tolerant Arithmetic/Logical Barrel Shifter and Rotator where n and k are the number of data bits and control bits, respectively then:

$$GB_{total} = n + 2k + 3; AI_{total} = n + k;$$

$$DL_{total} = 2(n + k)$$

The Bi-directional $(4, 2)$ Arithmetic/Shifter circuit executes arithmetic right shifting when C_{DU} goes LOW. During arithmetic right shifting, the left most or 1^{st} FR gate in Sign Control Unit generates a copy of sign bit (D_7) and others FD gates are used to generate multiple copies of sign bit which is controlled by 2^{nd} FR gate as presented in Fig. 8.

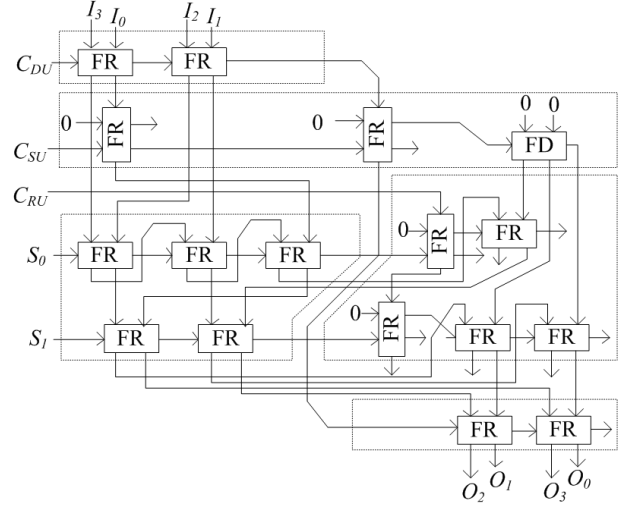


Fig. 8. Proposed design of Reversible Fault Tolerant Bi-directional Arithmetic/Logical Shift-Rotator circuits

IV. EVALUATION OF THE PROPOSED CIRCUIT

Here we present a comparison between proposed design and an existing design of [19] that performs bidirectional arithmetic/logical shifting including rotate operation. Table III shows that the proposed design has achieved excellent improvement over existing design about 200% in terms of Gate Count, about 200% to 500% in Garbage Count/Ancilla Inputs and $\sim 150\%$ in Delay. Fig. 9 shows the statistical analysis of outstanding result obtained from the comparative study on performance analysis shown in Table III. In case of Garbage Count and Ancilla Inputs, the improvement is growing up rapidly and the rate is proportional to the size of the shifting circuit (shown in Fig. 9). The proposed design methodology

not only incorporates the aforementioned feature in the developed circuits, but also outperforms the existing circuit [19] in terms of all the performance measures used to evaluate the reversible circuits. Quantum realization of proposed design produces better minimization and the improvement is more than 120% over [19]. Similarly, the hardware complexity analysis generate superior performance over the existing design and the improvement rate is more than 110% (as shown in Table III). Moreover, design methodology used minimum number of control inputs and area, and this number is lower than the existing method of [19]. Finally, the proposed design has presented better optimization of reversible shifting circuit among all the existing works on that area.

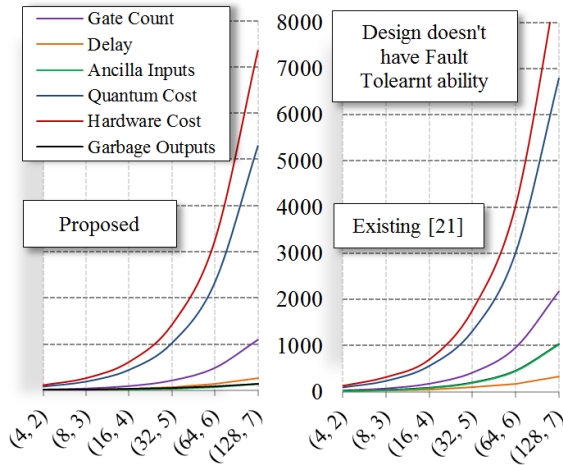


Fig. 9. Simulation Results, showing the evaluation of the proposed work has been compared with existing design [19].

TABLE III. PERFORMANCE ANALYSIS OF PROPOSED REVERSIBLE BI-DIRECTIONAL (n,k) BARREL SHIFTER COMPARED WITH EXISTING [19]

Cost Factors	Reversible (n, k) Barrel shifter and Rotator							
	Designs	(4, 2)	(8, 3)	(16, 4)	(32, 5)	(64, 6)	(128, 7)	
Total Gates	Proposed	17	40	93	214	487	1096	
	Existing [19]	29	73	177	417	961	2176	
Garbage	Proposed	11	17	27	45	79	145	
	Existing [19]	19	40	89	202	459	1036	
Quantum Cost	Proposed	82	191	444	1025	2342	5291	
	Existing [19]	97	237	560	1317	3013	6789	
Hardware Cost	Proposed	α	34	80	186	428	974	2192
		β	64	148	344	796	1824	4132
		γ	16	37	86	199	456	1033
	Existing [19]	α	46	114	274	642	1474	3330
		β	68	164	388	900	2052	4612
		γ	17	41	97	225	513	1153
Ancilla Inputs	Proposed	6	11	20	37	70	135	
	Existing [19]	13	33	81	193	449	1024	
Delay	Proposed	12	22	40	74	140	270	
	Existing [19]	16	28	50	92	174	336	

V. CONCLUSION

In this paper, a systematic approach to design Bi-directional Barrel shifter in reversible mode is presented. Exploiting the fault-tolerance capability of the reversible logic gates and modular design approach, the proposed technique is developed in such a way that exhibits all the features of a complete Reversible Fault-tolerant Bi-directional Barrel Shifter. None of the existing state-of-the-art methods [14]-[20] simultaneously include the features of bi-directionality, rotation capability and fault-tolerance in their designs. Currently, studies are being

performed to design other complex reversible circuits with the appropriate use of the proposed design.

REFERENCES

- [1] G. E. Moore, *Progress in Digital Integrated Electronics*, In IEEE Electron Devices Meeting, vol. 21, pp. 11–13, 1975.
- [2] R. Landauer, *Irreversibility and heat generation in the computing process*, IBM journal of research and development, Vol. 44(1), pp. 183–191, 1961.
- [3] K. N. Patel and J. P. Hayes and I. L. Markov, *Fault testing for reversible circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 23(8), pp. 1220–1230, 2004.
- [4] G. M. Tharakan and S. M. Kang, *A New Design of a Fast Barrel Switch Network*, IEEE Journal of Solid-State Circuits, Vol. 28(2), pp. 217–221, 1992.
- [5] N. Weste and D. Money, *CMOS VLSI Design*, AW Publisher, USA, 1993.
- [6] C. H. Bennett, *Logical Reversibility of Computation*, IBM Journal of Research and Development, Vol. 17(6), pp. 525–532, 1973.
- [7] R. P. Feynman, *Quantum Mechanical Computers*, Foundations of Physics, Vol. 16(6), pp. 507–531, 1986.
- [8] B. Parhami, *Fault Tolerant Reversible Circuits*, In Asilomar Conference on Signals, Systems and Computers, pp. 1726–1729, 2006.
- [9] E. Fredkin and T. Toffoli, *Conservative logic*, International Journal of Theoretical Physics, Kluwer Ac. Pub.-Plenum Pub., Vol. 21, pp. 219–253, 1982.
- [10] R. M. Pillmeier and M. J. Schulte and E. G. Walters III, *Design alternatives for barrel shifters*, In SPIE 4791, Advanced Signal Processing Algorithms, Architectures, and Implementations XII, 2002.
- [11] M. Perkowski and et al, *A hierarchical approach to computer-aided design of quantum circuits*. In International Symposium on Representations and Methodology of Future Computing Technology, pp. 201–209, 2003.
- [12] A. K. Biswas, Md. M. Hasan, A. R. Chowdhury and H. Md. H. Babu, *Efficient approaches for designing reversible Binary Coded Decimal adders*, Microelectronics Journal, Vol. 39(12), pp. 1693–1703, 2008.
- [13] S. K. Mitra and A. R. Chowdhury, *Minimum cost fault tolerant adder circuits in reversible logic synthesis*, In International Conference on VLSI Design, pp. 334–339, 2012.
- [14] H. Thapliyal and A. Bhatt and N. Ranganathan, *A New CRL Gate as Super Class of Fredkin Gate to Design Reversible Quantum Circuits*, In International Midwest Symposium on Circuits and Systems, pp. 1067–1070, 2013.
- [15] S. Gorgin and A. Kaivani, *Reversible barrel shifters*, In IEEE/ACS International Conference on Computer Systems and Applications, pp. 479–483, 2008.
- [16] I. Hashmi and H. Md. H. Babu, *An efficient design of a reversible barrel shifter*, In International Conference on VLSI Design, pp. 93–98, 2010.
- [17] S. Kotiyal and H. Thapliyal and N. Ranganathan, *Design of a reversible bidirectional barrel shifter*, In IEEE Conference on Nanotechnology, pp. 463–468, 2011.
- [18] E. Koushandeh and M. Haghparast, *A Beginning in the Design of Nanometric Fault Tolerant Reversible Barrel Shifter*, Australian Journal of Basic and Applied Sciences, Vol. 7(9), pp. 1110–1115, 2011.
- [19] O. Anjaneyulu and T. Pradeep and K. Reddy, *Design of An Efficient Reversible Logic Based Bidirectional Barrel Shifter*, International Journal of Electronics Signals and Systems, Vol. 2, pp. 48–53, 2012.
- [20] Md. Shamsujjoha and H. Md. H. Babu and L. Jamal and A. R. Chowdhury, *Design of a Fault Tolerant Reversible Compact Unidirectional Barrel Shifter*, In International Conference on VLSI Design, pp. 103–108, 2013.
- [21] A. R. Chowdhury, R. Nazmul, and H. M. H. Babu, “A new approach to synthesize multiple-output functions using reversible programmable logic array,” in *International Conference on VLSI Design*, pp. 311–316, 2006, .
- [22] S. Mitra and A. R. Chowdhury, “Minimum cost fault tolerant adder circuits in reversible logic synthesis,” in *International Conference on VLSI Design*, pp. 334–339, 2012.