2013 26th International Conference on VLSI Design and the 12th International Conference on Embedded Systems

Design of a Fault Tolerant Reversible Compact Unidirectional Barrel Shifter

Md. Shamsujjoha[†], Hafiz Md. Hasan Babu[‡], Lafifa Jamal[§] and Ahsan Raja Chowdhury[✠]
[†‡§✠]Department of Computer Science and Engineering, University of Dhaka, Dhaka-1000, Bangladesh.
Email: [†]dishacse@yahoo.com, [‡]hafizbabu@hotmail.com, [§]lafifa@yahoo.com, [✠]farhan717@yahoo.com.

***Abstract*—This paper demonstrates reversible logic synthesis for (*n*, *k*) unidirectional logarithmic barrel shifters, where *n* is the number of data bits and *k*=log$_2$*n*. The circuits are designed using only reversible fault tolerant Fredkin gates. Thus, the entire scheme inherently becomes fault tolerant. Several lower bounds on the numbers of garbage outputs and constant inputs have been proposed. The comparative results show that the proposed method is much better in terms of numbers of gates, garbage outputs, quantum cost, hardware complexity and has significantly better scalability than the existing approaches.**

***Index Terms*—Barrel Shifter, Garbage Output, Hardware Complexity, Low power Design, Quantum Cost.**

I. Introduction

Conventional logic dissipates more power by losing bits of information whereas reversible circuitry avoid bit loss through unique mapping between input and output vectors [1]. In this regard, reversible logic plays an extensively important role in low power computing [2]. Moreover, reversible circuits are considered as a special case of quantum circuit as quantum evolution must be reversible [3]. Over the past few years, reversible circuitry gained remarkable interests in the field of optical computing [4], DNA-technology [5], nanotechnology [6], program debugging and testing [7], quantum dot cellular automata [8], discrete event simulation [9], modeling of biochemical systems and in the development of highly efficient algorithms [10], etc. Parity checking is a popular mechanism for detecting single level faults. An entire circuit can preserve parity if its individual gate is parity preserving [11]. Reversible fault tolerant circuit detect faulty signal in its primary outputs through parity checking [12]. In addition, data shift and rotate operations are widely used in arithmetic operations, bit-indexing and over all in fast encoding [13]. Barrel shifter shifts and rotates multiple bits in a cycle and hence, it attains great importance in designing processors [14], specially in digital signal processors [15] and low-density parity-check (LDPC) decoders [16]. Among the various barrel shifters, logarithmic one has the simplest structure and is more area efficient as it doesn't require any underneath decoder circuit.

In these consequences, this paper presents the generalized design methodologies of reversible fault tolerant unidirectional logarithmic barrel shifters. Reversible designs of unidirectional logarithmic barrel shifter was presented in [17] and [18], but none of the designs were generalized and compact.

II. Basic Definitions and Literature Review

This section defines reversible gates, garbage output, delay, hardware complexity and presents popular reversible and fault tolerant gates along with their quantum equivalent realizations.

A. Reversible and Fault Tolerant Gates

An $n \times n$ ***reversible gate*** is a data stripe block that uniquely maps between input vector $I_v = (I_0, I_1, ..., I_{n-1})$ and output vector $O_v = (O_0, O_1, \ldots, O_{n-1})$. The prime requirements for reversible circuit are as follows : "*There should be one-to-one correspondence among the input-output sequences, and thus, it must have equal number of inputs and outputs*"[1]∼[6].

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

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

B. Delay, Garbage Output, Hardware Complexity and Quantum Cost

The ***delay*** of a logic circuit is the delay of the critical path. Critical path is the path that consists of maximum number of gates from any input to any output [1].

Unused output of a reversible circuit is known as ***garbage output***, *i.e.*, output of any gate that neither used as primary output nor as input to another gate is the garbage output [2].

The number of basic operations of a circuit is referred to the ***hardware complexity***. 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 hardware complexity of the entire circuit is calculated with respect to α, β, and γ.

The ***quantum cost*** for all 1×1 and 2×2 reversible gates are considered as 0 and 1, respectively [1]∽[12]. Hence, the quantum cost of a reversible gate or circuit is the total number of 2×2 quantum gate used in it.

C. Popular Reversible and Fault Tolerant Gates

1) ***Toffoli Gate:***: Input and output vectors for 3×3 Toffoli gate (TG) are defined as follows : Input vector $I_v = (a, b, c)$ and output vector $O_v = (a, b, ab \oplus c)$. Block diagram of TG is shown in Fig. 1(a). Fig. 1(b), representing the quantum equivalent realization of TG, which shows that, TG is realized with five 2×2 primitive gates. So, the quantum cost of TG is five. TG is also known as a reversible universal gate as it can realize all the basic logic operations.

1063-9667/13 $26.00 © 2013 IEEE
DOI 10.1109/VLSID.2013.171

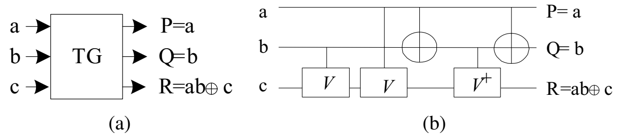


Fig. 1: Toffoli gate (a) Block diagram (b) Quantum realization

*2) **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. 2(a). Fig. 2(b) represents the quantum equivalent realization of $F2G$. From Fig. 2(b) we find that it is realized with two 2×2 Ex-OR gate. Thus, its quantum cost is two. *F2G* can be used as copying gate, for example, if its I_v be ($a, b = 0, c = 0$), then its O_v will be (a, a, a).

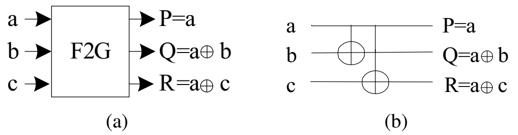


Fig. 2: Feynman double gate (a) Block diagram (b) Quantum realization

3) Fredkin Gate: The 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. 3(a). Fig. 3(b) represents the quantum realization of FRG. In Fig. 3(b), each rectangle is equivalent to a 2×2 quantum primitives, therefore its quantum cost is considered as one [1]∽[12]. Thus, total quantum cost of FRG is five.

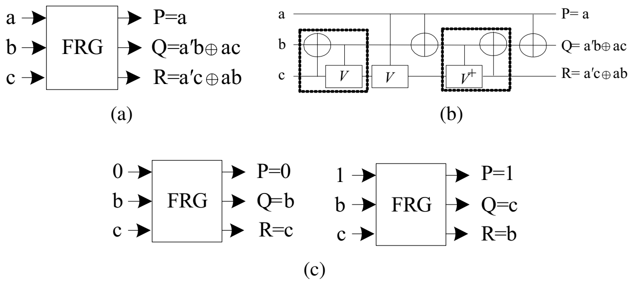


Fig. 3: Fredkin gate (a) Block diagram (b) Quantum realization (c) FRG Working as a 2:1 Mux

FRG can work as 2-to-1 Multiplexer (Mux). Referring to its input-output combinations, when *a* = 1, the inputs *b* and *c* will be swapped. The resulting value of the outputs are $Q = c$ and $R = b$. On the other hand, if *a* = 0, then outputs P, Q and R are directly connected to inputs *a*(= 0), *b*, and *c*, respectively (shown in Fig. 3(c)). FRG is also a universal gate like TG. FRG and $F2G$ are fault tolerant gate, *i.e.*, both comply with the condition of Eq. 1. The fault tolerant (parity preserving) properties of FRG and $F2G$ are shown in Table I.

TABLE I: Truth table for $F2G$ and FRG

Input			Output of $F2G$			Output of 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

Theorem 1: A $2n$-to-n reversible fault tolerant Mux requires at least $2n$ garbage outputs and no constant inputs, where n is number of data bits.

Proof: Let n be the number of data bits. Then, for a $2n$-to-n reversible fault tolerant Mux, there are $2n$ data inputs and n select inputs. So, to preserve the one-to-one mapping of reversibility, there should be at least $3n$ outputs. This input-output combinations preserve parity too. Among these $3n$ outputs, there are n primary outputs. Hence, there should be at least $2n$ garbage outputs for $2n$-to-n reversible fault tolerant Mux, which causes no constant input. □

Example 1: A $2n$-to-n MUX with n=1 has two data inputs (I_0 and I_1) and one control input (S_0). So, the total number of inputs for the 2-to-1 Mux is three. Thus, according to the property of reversibility, reversible 2-to-1 Mux must have at least three outputs. Among these three outputs, only one is the primary output, which means at least two outputs remain unused, *i.e.*, at least two garbage outputs are generated to realize 2-to-1 Mux reversibly. If we replace n with 1 in Theorem 1, we get two garbage outputs as well. Architecture of reversible 2-to-1 Mux is shown in Fig. 3(c). This architecture also has two garbage outputs, namely P and R. Table I proved that, Fig. 3(c) maintains the fault tolerant property of Eq. 1. Since, it is possible to realize the reversible fault tolerant 2-to-1 Mux with only three outputs, thus it will not require any constant input. Circuitry of Fig. 3(c) also has no constant input, that satisfies Theorem 1 for n=1.

D. Existing Unidirectional Barrel Shifters

Barrel shifter has n-input and n-output lines for data transmission and k control inputs, where k=log$_2n$. To our best knowledge, there are two reversible unidirectional barrel shifter in literature [17] and [18]. Both designs are logarithmic barrel shifter capable of performing left rotation only. Adaptive structure of the basic (n, k) unidirectional logarithmic barrel shifter is shown in Fig. 4. 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) or not to shift (or rotate) the input data 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.

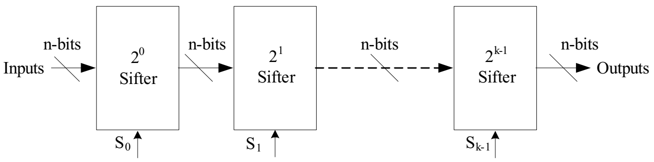


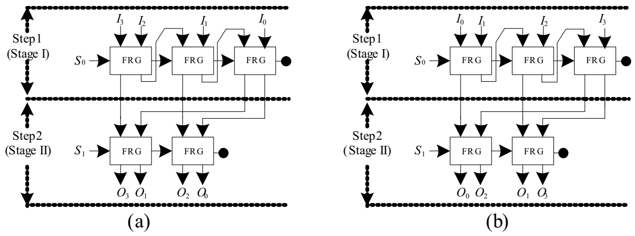
Fig. 4: Adaptive structure of (n, k) logarithmic barrel shifter

Theorem 2: 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 output. So, the number of garbage output is k, which result's no constant input are required. □

III. Proposed Reversible fault Tolerant Unidirectional Logarithmic Rotators

Fig. 5 and Fig. 6 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, the proposed rotators preserve parity at all stages.

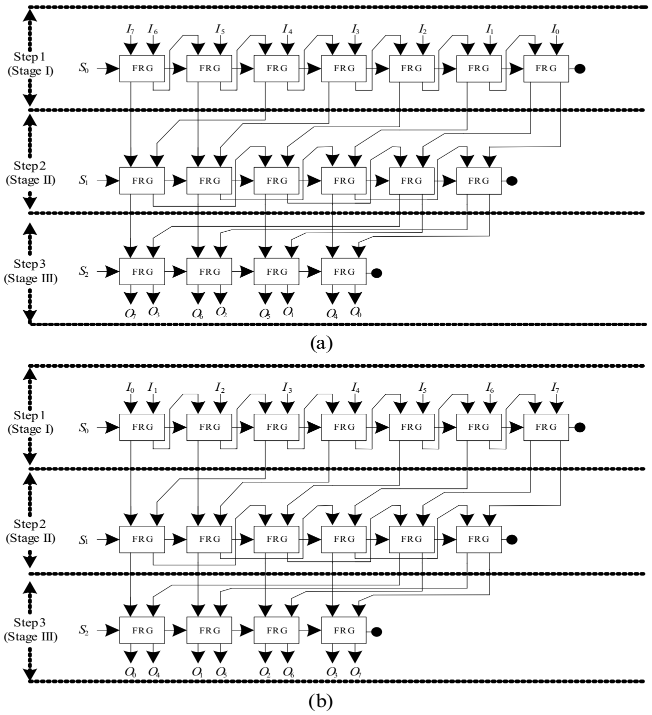


(a)

(b)

Fig. 5: Proposed $(4, 2)$ reversible fault tolerant unidirectional logarithmic barrel shifter (a) Circuit for left rotation (b) Circuit for right rotation.

Proposed unidirectional rotators follow the adaptive structure of logarithmic barrel shifter, *e.g.*, 1^{st} stage rotates 2^0 bit, 2^{nd} stage rotates 2^1 bits, 3^{rd} stage rotates 2^2 bits and so on. Rotation occurs only when control signal is set to high. If a control signal from any stage is set to low then instead of rotating, that stage just passes the input to the next stage. Design procedure of proposed reversible fault tolerant rotator is as follows : Let the data inputs for the proposed (n, k) rotator is $I_0, I_1, I_2, \ldots, I_{n-3}, I_{n-2}, I_{n-1}, I_{n-1}$ and the control inputs are $S_0, S_1, \ldots, S_{k-1}$; where, n represents the number of inputs and k is number of stages which equals to log$_2 n$. Each of these stages require a chain of $(n - 2^j)$ Fredkin gates (where j=0 to $k-1$). 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 $k - 1 \geq j \geq 0$.


Step1 (Stage I)
Step2 (Stage II)
Step3 (Stage III)
S_0
S_1
S_2
I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7
FRG

(a)

(b)

Fig. 6: Proposed $(8, 3)$ reversible fault tolerant unidirectional logarithmic barrel shifter (a) Circuit for left rotation (b) Circuit for right rotation.

The working procedure of the proposed reversible fault tolerant rotator is as follows[1] : Let n = 4. Then, we have a (4, 2) shifter which 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 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 operation 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 input sequence will remain unchanged. For example, when control input S_0 is set to low, the 1^{st} Fredkin gate (FRG) of first row will select input I_0, the 2^{nd} will select I_1 and 3^{rd} one will select I_2 also I_3 will be generated from another output of 3^{rd} FRG. Finally selected four outputs will be used as inputs of 1^{st} and 2^{nd} FRG at 2^{nd} row which are controlled by the control input S_1. Besides, if 1^{st} control input is set to high then the selection sequence will be I_1, I_2, I_3 and I_0 which are 1^{st}, 2^{nd} and 3^{rd} FRG at 1^{st} row, respectively.

Example 2: Let the value of n be 4. Then, we have (4, 2) barrel shifter. Fig. 5 shows that the proposed (4, 2) barrel shifter (circuits for rotation) has 2 two garbage outputs and no constant inputs. If we replace n with 4 in Theorem 2, 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 2.

[1] This is for right rotator's circuit. Unless mentioned explicitly, the design procedure elaborated in this section is for right rotator.

Lemma 1: Let ***FR*** be the required number of gates for (n, k) reversible fault tolerant unidirectional logarithmic rotator, where n and k (k= $\log_2 n$) are the number of data bits and control inputs, respectively. Then,

$$\boldsymbol{FR} = nk - \sum_{j=0}^{k-1} 2^j$$

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-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, where n is the number of data inputs and k is number of control inputs (k=$\log_2 n$). Then,

$$\boldsymbol{QC} = 5nk - 5\sum_{j=0}^{k-1} 2^j$$

Proof: In Lemma 1, we proved that the proposed reversible fault tolerant (n, k) rotator can be realized with $(nk - \sum_{j=0}^{k-1} 2^j)$ reversible fault tolerant Fredkin gates. According to our discussion in Sec.II, quantum cost of each Fredkin gate is 5. Therefore, total quantum cost of the (n, k) reversible fault tolerant unidirectional logarithmic rotator is $5 \times (nk - \sum_{j=0}^{k-1} 2^j) = (5nk - 5\sum_{j=0}^{k-1} 2^j)$. □

Lemma 3: Let α, β and γ be the hardware complexity for two-input Ex-OR, AND, NOT calculations, respectively. Let ***HC*** be the total hardware complexity for (n, k) reversible fault tolerant unidirectional logarithmic rotator, where n is the number of data inputs and k is number of control inputs which equals to $\log_2 n$. Then,

$$\boldsymbol{HC} = (2nk - \sum_{j=0}^{k-1} 2^{j+1})\alpha + (nk - \sum_{j=0}^{k-1} 2^j)(4\beta + \gamma)$$

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 (2α+4β+γ) [1]. Thus, hardware complexity of (n, k) reversible fault tolerant unidirectional rotator is $(2nk$-$\sum_{j=0}^{k-1} 2^{j+1})\alpha$+$(4nk$-$\sum_{j=0}^{k-1} 2^{j+2})\beta$+$(nk$-$\sum_{j=0}^{k-1} 2^j)\gamma$ = $(2nk$-$\sum_{j=0}^{k-1} 2^{j+1})\alpha$+$(nk$-$\sum_{j=0}^{k-1} 2^j)(4\beta$+$\gamma)$ □

Architecture of (n, k) reversible fault tolerant right rotator is shown in Fig. 7. From Fig. 5 and Fig. 6, we find that the proposed (4, 2) and (8, 3) reversible fault tolerant left and right rotators are very similar to each other, except the positions of input-output vectors. Thus, the reversible fault tolerant right rotator is shown for only the architecture of (n, k).

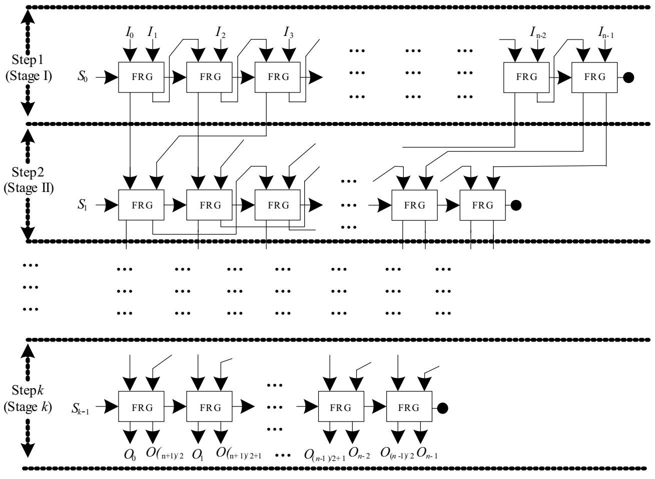


Fig. 7: Proposed (n, k) reversible fault tolerant unidirectional logarithmic right rotator

IV. Proposed Reversible Fault Tolerant Unidirectional Logarithmic 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, *i.e.*, here, S_{k-1} acts as the control signal for the 1^{st} stage, S_{k-2} for the 2^{nd} stage and so on. Thus, the k stages of 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 Fig. 8 and Fig. 9, respectively.

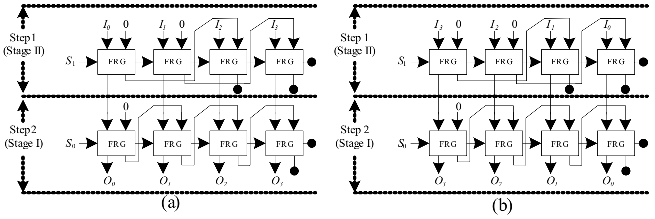
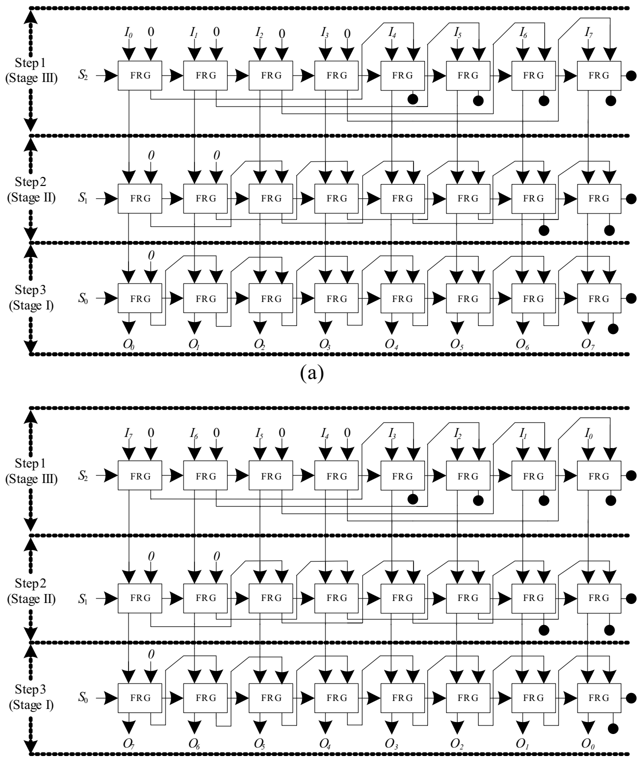


Fig. 8: Proposed $(4, 2)$ reversible fault tolerant unidirectional logarithmic logical shifter (a) Circuit for left logical shift (b) Circuit for right logical shift

Inputs to the Fredkin gates for the 1^{st} stage of the proposed logical shifter are as follows [2]:

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

[2] This design and working procedure is for the right logical shifter. Unless mentioned explicitly, the design procedure elaborated in this section is for right logical shifter.


Step1
(Stage III)
Step2
(Stage II)
Step3
(Stage I)
S_2
S_1
S_0
FRG
I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7
0
O_0 O_1 O_2 O_3 O_4 O_5 O_6 O_7
O_7 O_6 O_5 O_4 O_3 O_2 O_1 O_0

(a)

(b)

Fig. 9: Proposed $(8,3)$ reversible fault tolerant unidirectional logarithmic logical shifter (a) Circuit for left logical shift (b) Circuit for right logical shift

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, we have following values at the outputs:

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

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

$P(i,k-1)=S_{k-1}$, $\forall i=(n-1)\dots 0$
$Q(i,k-1)=i_i$, $\forall i=(n-1)\dots 0$
$R(i,k-1)=0$, $\forall i=(n-1)\dots 0$

The second stage of proposed (n,k) reversible fault tolerant logical shifters uses inputs from the first stage. From the outputs $Q(i,k-1)$ and $R(i,k-1)$ of the Fredkin gates in the first stage, all $Q(n,k-1)$ and some $R(n,k-1)$ outputs (first n/2 outputs) will be the useful outputs, while rest $R(n,k-1)$ outputs (last n/2 outputs) will work as the garbage outputs. Design methodology used for the second stage is as follows :

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

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,

$P(i,k-2)=S_{k-2}$, $\forall i=(n-1)\dots 0$
$R(i,k-2)=Q(i,k-1)$, $\forall i=(n-1)\dots 0$
$Q(i,k-2)=0$, $\forall i=(n-1)\dots(n-1)-2^{k-2}+1$
$Q(i,k-1)=Q(m,k-1)$, $\forall i=(n-1)-2^{k-2}\dots 0$ and $m=(n-1)\dots(n-1)-2^{k-2}+1$.

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

$P(i,k-2)=S_{k-2}$, $\forall i=(n-1)\dots 0$
$Q(i,k-2)=Q(i,k-1)$, $\forall i=(n-1)\dots 0$
$R(i,k-2)=Q(i,k-1)$, $\forall i=(n-1)\dots 0$.

Similar design strategy used for the remaining stages. Architecture of the (n, k) reversible fault tolerant right logical shifter is shown in Fig. 10. Since, there are almost no difference between the proposed logical left shifters and the right shifters, except for the input-output combination hence the architecture of (n, k) reversible fault tolerant left logical shifter is not shown here. From Fig. 8, Fig. 9 and Fig. 10, we find that, only reversible fault tolerant Fredkin gate is used here. Thus, designs of the proposed logical shifters also preserve parity.

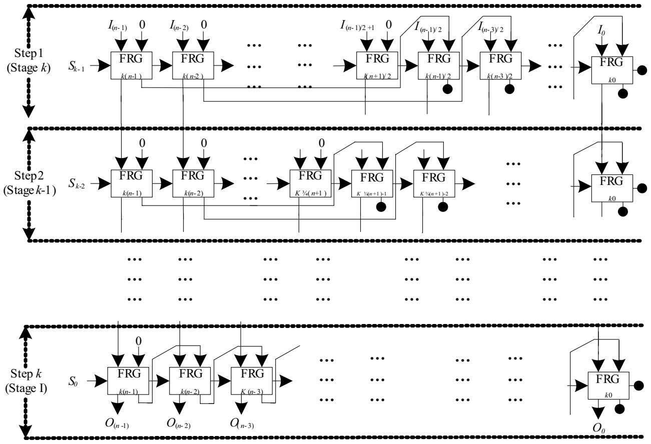


Fig. 10: Proposed (n,k) reversible fault tolerant logarithmic logical shifter (Circuit for right logical shift)

Lemma 4: Let $\boldsymbol{F}$ be the required number of gates for a (n,k) reversible fault tolerant logical shifter, where n is the number of data inputs and $k=log_2 n$. Then, $\boldsymbol{F}=nk$.

Proof: A (n,k) logical shifter has k stages (0 to k-1) and n input bits, where k=$log_2 n$. 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, where n is the number of data bits, k=$\log_2 n$; α, β and γ are the hardware complexity for two-input Exclusive-OR, AND, NOT operations, respectively.

TABLE II: Comparative study of reversible unidirectional logarithmic barrel shifter

	Proposed Rotator						Existing Rotator [18]						Proposed Logical Shifter					
(n,k)	GO	GT	QC	CI	D	HC	GO	GT	QC	CI	D	HC	GO	GT	QC	CI	D	HC
(4,2)	2	5	25	0	4	10α 20β 5γ	6	10	34	4	6	16α 24β 6γ	8	5	40	3	5	16α 32β 8γ
(8,3)	3	17	85	0	9	34α 68β 17γ	19	36	116	16	14	56α 80β 20γ	24	10	120	7	10	48α 96β 24γ
(16,4)	4	49	245	0	18	98α 196β 49γ	52	104	328	48	28	160α 224β 56γ	64	19	320	15	19	128α 256β 64γ
(32,5)	5	129	645	0	35	258α 516β 129γ	133	272	848	128	52	416α 456β 144γ	160	36	800	31	36	320α 640β 160γ
(64,6)	6	321	1605	0	68	642α 1284β 321γ	326	672	2080	320	98	1728α 1408β 352γ	384	69	1920	63	69	768α 1536β 384γ

*In this table, GO = No. of Garbage Outputs; GT = No of Gates; QC = Quantum cost; CI = No. of Constant Inputs, D = Critical Path Delay and HC = Hardware Complexity, where α, β and γ are the two-input Ex-or, AND and NOT operations, respectively.

Proof: In Lemma 4, we proved that the proposed reversible fault tolerant (n, k) logical shifter can be realized with nk reversible fault tolerant Fredkin gates. Similarly, like Lemmas 2 and 3 it can be proved that the reversible fault tolerant unidirectional logical shifter can be realized with $5nk$ quantum cost and ($2nk\alpha$+$4nk\beta$+$nk\gamma$) hardware complexity.□

V. Performance of the Proposed Methods

Existing work [18] greatly outperforms the work of [17], and hence we compare our proposed work with only with the method of [18]. Tables II show the performance of the proposed reversible fault tolerant scheme with the existing reversible one [18]. From Tables II, we find that the proposed rotators don't require any constant input whereas the existing method does. The overall results of Table II proves that the proposed scheme performs much better and are much more scalable than the existing scheme [18].

VI. Conclusions

In this paper, we presented the design methodologies of the $(n,\ k)$ reversible fault tolerant unidirectional barrel shifters, where n is the number of data bits and k=$\log_2 n$. We proposed several lower bounds on the numbers of garbage outputs and constant inputs. It has also been proved that the proposed rotator circuits are constructed with the optimum garbage outputs and constant inputs. Except the left rotators, all the designs are first ever proposed in the literature, to the best of our knowledge. By articulating several theoretical explanations, we have also proved the efficiency and supremacy of all the proposed schemes. Most interestingly, all the proposed designs have the capability of detecting errors at circuit's primary outputs which are not at all available in the existing circuits [17], [18].

References

[1] L. Jamal, M. Shamsujjoha, and H. M. Hasan Babu, "Design of optimal reversible carry look-ahead adder with optimal garbage and quantum cost," *International Journal of Engineering and Technology*, vol. 2, pp. 44–50, 2012.

[2] F. Sharmin, M. M. A. Polash, M. Shamsujjoha, L. Jamal, and H. M. Hasan Babu, "Design of a compact reversible random access memory," in *4th IEEE International Conference on Computer Science and Information Technology*, vol. 10, Chengdu, China, Jun. 2011, pp. 103–107.

[3] M. Nielsen and I. Chuang, *Quantum computation and quantum information*. New York, USA: Cambridge University Press, 2000.

[4] M. Perkowski, "Reversible computation for beginners," lecture series, Portland state university, 2000.

[5] M. P. Frank, "The physical limits of computing," *Computing in Science and Engg.*, vol. 4, no. 3, pp. 16–26, May 2002.

[6] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. Hasan Babu, "Efficient approaches for designing reversible binary coded decimal adders," *Microelectron. J.*, vol. 39, no. 12, pp. 1693–1703, Dec. 2008.

[7] S. N. Mahammad and K. Veezhinathan, "Constructing online testable circuits using reversible logic," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, pp. 101–109, 2010.

[8] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and weaknesses of quantum computing," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1510–1523, Oct. 1997.

[9] H. M. H. Babu, M. I. Zaber, M. M. Rahman, and M. R. Islam, "Implementation of multiple-valued flip-flips using pass transistor logic," in *Euromicro Symposium on DSD, Architectures, Methods and Tools, France*. IEEE Computer Society, 2004, pp. 603–606.

[10] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Proceedings of the 40th annual Design Automation Conference*, USA, 2003, pp. 318–323.

[11] R. K. James, T. K. Shahana, K. P. Jacob, and S. Sasi, "Fault tolerant error coding and detection using reversible gates," in *IEEE TENCON*, 2007, pp. 1–4.

[12] B. Parhami, "Fault-tolerant reversible circuits," in *Fortieth Asilomar Conference on Signals, Systems and Computers.*, 2006, pp. 1726 –1729.

[13] J. Muwa, G. Fettweis, and H. Ne, "Circuit for rotating, left shifting, or right shifting bits," *U.S. Patent : 5978822*, 1995.

[14] F. Worrell, "Microprocessor shifter using rotation and masking operations," *US Patent*, vol. 5, p. 482, 1998.

[15] Voyiatzis, D. Gizopoulos, and A. Paschalis, "Accumulator-based test generation for robust sequential fault testing in DSP cores in near-optimal time," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 1079–1086, 2005.

[16] J. Scott, L. Lee, J. Arends, and B. Moyer, "Designing the low-power mcore architecture," *in Proc. Intl. Symp. on Computer Architecture Power Driven Microarchitecture Workshop*, p. 145150, 1998.

[17] S. Gorgin and A. Kaivani, "Reversible barrel shifters," in *Computer Systems and Applications, ACS/IEEE International Conference on*. Los Alamitos, CA, USA: IEEE Computer Society, 2007, pp. 479–483.

[18] I. Hashmi and H. M. H. Babu, "An efficient design of a reversible barrel shifter," in *23rd International Conference on VLSI Design, 9th International Conference on Embedded Systems*. IEEE Computer Society, 2010, pp. 93–98.