

## Research Article

# First Steps in Creating Online Testable Reversible Sequential Circuits

Mozammel H. A. Khan<sup>1</sup> and Jacqueline E. Rice<sup>2</sup> 

<sup>1</sup>Department of Computer Science and Engineering, East West University, Aftabnagar, Dhaka 1212, Bangladesh

<sup>2</sup>Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, AB, Canada T1K 3M4

Correspondence should be addressed to Jacqueline E. Rice; [j.rice@uleth.ca](mailto:j.rice@uleth.ca)

Received 2 May 2017; Accepted 6 December 2017; Published 14 January 2018

Academic Editor: Marcelo Lubaszewski

Copyright © 2018 Mozammel H. A. Khan and Jacqueline E. Rice. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Synthesis of reversible sequential circuits is a very new research area. It has been shown that such circuits can be implemented using quantum dot cellular automata. Other work has used traditional designs for sequential circuits and replaced the flip-flops and the gates with their reversible counterparts. Our earlier work uses a direct feedback method without any flip-flops, improving upon the replacement technique in both quantum cost and ancilla inputs. We present here a further improved version of the direct feedback method. Design examples show that the proposed method produces better results than our earlier method in terms of both quantum cost and ancilla inputs. We also propose the first technique for online testing of single line faults in sequential reversible circuits.

## 1. Introduction

Moore's law [1] predicted a continuous rise in the number of transistors per chip due to the continuous reduction in feature size. The popular thought is that Moore's law will no longer hold true in about a decade when the feature size will approach the atomic level. DeBenedictis [2] showed that in CMOS technology feature size is no longer the primary challenge; rather energy dissipation is the new limiting factor. Current CMOS circuits implement Boolean logic networks, where the main contributor to heat generation is the energy in the 0 and 1 signals stored on wires. Each time a signal changes from one state to another and back again,  $CV^2$  joules of energy are turned into heat, where  $C$  is the capacitance between the wire and the ground and  $V$  is the power supply voltage. The most common number cited is  $CV^2 \geq 100kT$ , where  $k$  is Boltzmann's constant and  $T$  is the ambient temperature in kelvin [2]. DeBenedictis [2] also stated that adiabatic and reversible logic essentially recycle  $CV^2$  joules of signal energy many times before dissipating as heat. Consequently, the energy drawn from the power supply could be reduced by as much as 100 times. However, signal energy recycling cannot take place in traditional Boolean networks.

Therefore reversible logic may be the next promise for CMOS technology.

In a theoretical study, Landauer [3] stated that irreversible logic operations dissipate  $kT \ln 2$  of heat energy when a bit of information is lost, where again  $k$  is Boltzmann's constant and  $T$  is the operating temperature in kelvin. In another theoretical study, Bennett [4] showed that, from a thermodynamic point of view, if a circuit is both logically and physically reversible, then  $kT \ln 2$  heat will not be dissipated. Experiments have shown that dissipation of no heat is not achievable; however, it has been experimentally demonstrated that reversible logic dissipates less heat than the thermodynamic limit of  $kT \ln 2$  in physically irreversible low-power CMOS logic [5] and physically reversible superconductor flux logic (SFL) [6]. Thus, reversible logic is a favorable choice for low-power emerging computing technologies.

Reversible logic has become realizable in many emerging computing technologies such as superconductor flux logic (SFL) technology [6, 7], optical technology [8, 9], quantum dot cellular automata technology [10, 11], and nanotechnology [12]. In addition, quantum circuits are inherently reversible [13]. This is another reason why reversible logic has become

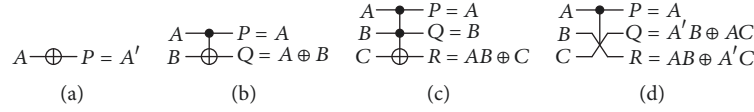


FIGURE 1: Commonly used reversible gates. (a) NOT gate, (b) CNOT or Feynman gate, (c) Toffoli gate, and (d) controlled swap or Fredkin gate.

a promising choice for low-power emerging computing technologies.

Reversible logic synthesis attempts have mostly focused on reversible combinational logic synthesis. Some of the most important but nonexhaustive work in this area is [14–23]. However, very few works have investigated reversible *sequential* logic synthesis. The major problem behind this lack of work is the prevailing thought that feedback is not possible in reversible logic and therefore reversible sequential circuits are not possible. However, in 1980, Toffoli [24] argued that if the feedback is provided through a delay element, then the feedback information will be available as the input to the reversible combinational circuit in the next clock cycle and thus a reversible sequential circuit is possible. Moreover, Thapliyal et al. [25] showed that reversible sequential circuits can be implemented using quantum dot cellular automata (QCA) technology by providing appropriate feedback timing by managing the clock of the QCA wire providing the feedback.

This paper presents a further improvement of our earlier work in designing a sequential circuit using direct feedback instead of flip-flops [26]. In addition, to our knowledge no other work has offered any technique for online testing of sequential circuits. Thus we also propose the first method of testing for single line faults in sequential circuits.

The rest of the paper is organized as follows. In Section 2, we discuss background for our proposed design method and related topics. We discuss related work on design of sequential circuits and online testing methods for combinational circuits in Section 3. In Section 4, we present our proposed method of designing sequential circuits. We show design examples in Section 5. In Section 6, we introduce our proposed method of online testing for single line faults in sequential circuits. We also present results for several benchmark circuits. Finally, in Section 8, we conclude the paper.

## 2. Background

**2.1. Reversible Logic.** A reversible function is a bijective mapping  $F : \{0, 1\}^n \mapsto \{0, 1\}^n$ , where  $n$  is the number of input variables. A reversible gate (or a circuit) implements a reversible function; that is, it maps the input combination to the output combination in a bijective manner. Thus a reversible gate (or a circuit) has the same number of inputs and outputs. In addition every input combination produces a unique output combination in a reversible circuit, and therefore for every output combination, the corresponding input combination can be uniquely determined.

A reversible circuit is constructed using reversible gates. The commonly used reversible gates are shown in Figure 1.

Figure 1(a) shows the symbol of a NOT gate. It complements the input at the output; that is,  $P = A'$ . Figure 1(b) shows the symbol of a controlled-NOT (CNOT) or Feynman gate. The input  $A$  is called the control input and its value is passed unchanged to the output; that is,  $P = A$ . The input  $B$  is called the target input. The target output has the value  $Q = A \oplus B$ . Figure 1(c) shows the symbol of a three-input Toffoli gate (sometimes referred to as a controlled-controlled-NOT gate or CCNOT gate). The inputs  $A$  and  $B$  are called the control inputs and their values are passed unchanged to the outputs; that is,  $P = A$  and  $Q = B$ . The input  $C$  is called the target input and the target output has the value  $R = AB \oplus C$ . Toffoli gates may have more than three inputs, in which case they may be referred to as multiple-controlled Toffoli gates. In an  $n$ -input Toffoli gate, the first  $(n-1)$  inputs (say  $x_1, x_2, \dots, x_{n-1}$ ) are the control inputs and the last input (say  $x_n$ ) is the target input. The value of the target output is  $x_1 x_2 \cdots x_{n-1} \oplus x_n$ . Figure 1(d) shows the symbol of a controlled swap gate, or Fredkin gate. Input  $A$  is the control input and inputs  $B$  and  $C$  are targets. When the control input value is  $A = 0$ , then the target inputs  $B$  and  $C$  are passed unchanged to the outputs; that is,  $Q = B$  and  $R = C$ . When the control input value is  $A = 1$ , then the target inputs are swapped at the outputs; that is,  $Q = C$  and  $R = B$ . The outputs  $Q$  and  $R$  can be expressed as  $Q = A' B \oplus AC$  and  $R = AB \oplus A' C$ . When  $A = 0$ , then  $Q = 0' \cdot B \oplus 0 \cdot C = B$  and  $R = 0 \cdot B \oplus 0' \cdot C = C$ . When  $A = 1$ , then  $Q = 1' \cdot B \oplus 1 \cdot C = C$  and  $R = 1 \cdot B \oplus 1' \cdot C = B$ .

Quantum cost and the number of *ancilla inputs* are widely used as metrics for comparing reversible circuits. Quantum cost refers to the number of primitive quantum gates required to realize the circuit, while ancilla inputs are the constant-initialized working inputs, in addition to the function inputs, that are required for the functionality of the circuit.

NOT and CNOT gates are technology realizable primitive gates, and their quantum costs are assumed to be one. Toffoli and controlled swap gates are macro-level gates and must be realized through a combination of primitive quantum gates. The three-input Toffoli gate and the controlled swap gate can be realized using five primitive quantum gates [27, 28] and thus their quantum cost is five each.

Realization of multiple-controlled Toffoli gates from primitive quantum gates is presented in [29], where quantum costs for up to 16-input Toffoli gates are reported. The quantum costs for 4-input, 5-input, and 6-input Toffoli gates are 14, 20, and 32, respectively.

In [30], extended Toffoli gates (ETG) are proposed, which have two target outputs. ETGs are very useful in online testing of reversible combinational circuits. In Figure 2(a), the extended Feynman gate (EFG) and its logic level implementation are shown. The quantum cost of the

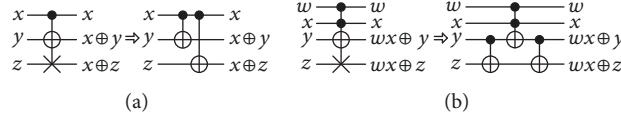


FIGURE 2: The symbol and logic level implementations for (a) the extended Feynman gate and (b) a 2-control extended Toffoli gate.

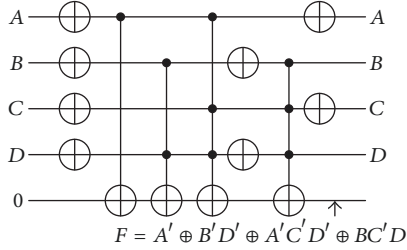


FIGURE 3: Reversible circuit implementing (1).

EFG is 2. In Figure 2(b), a 2-control ETG and its logic level implementation are shown. The quantum cost of an ETG is 2 plus the quantum cost of the corresponding Toffoli gate. The quantum cost of the ETG in Figure 2(b) is  $5 + 2 = 7$ .

**2.2. Reversible Logic Synthesis Using ESOP Expressions.** An exclusive-OR sum of products (ESOP) expression is similar to a conventional sum of products (SOP) expression, except that product terms are combined with exclusive-OR (EXOR) operators rather than OR operators [31]. A common technique in reversible logic synthesis is to express the logic function as an ESOP expression and then realize the ESOP expression as a cascade of NOT, CNOT, and Toffoli gates [32]. An example ESOP expression is given in (1) and its reversible realization as a cascade of NOT, CNOT, and Toffoli gates is shown in Figure 3.

$$F(A, B, C, D) = \overline{A} \oplus \overline{B} \overline{D} \oplus \overline{A} \overline{C} \overline{D} \oplus B \overline{C} D. \quad (1)$$

**2.3. Sequential Circuits.** The classical model of a sequential circuit is shown in Figure 4. A sequential circuit has a memory whose content during the present clock cycle is called the present state. The combinational circuit produces the output as a function of the present state and the input. The combinational circuit also produces the next state as a function of the present state and the input; this computed next state becomes the present state during the next clock cycle. The behavior of a sequential circuit is described using a state transition diagram. In describing a sequential circuit the present state is represented by  $Q$  and the next state is represented by  $Q^+$ . An example sequential circuit is shown in the state transition diagram in Figure 4. It has 1-bit input ( $x$ ), 1-bit output ( $z$ ), and 2-bit state ( $Q_1 Q_0$ ). The reset state is 00.

**2.4. Testing of Reversible Circuits.** Testing is very important to ensure quality, availability, and reliability of a circuit. There are two approaches to circuit testing: offline testing and online testing [33]. In offline testing, a circuit is tested when it is not

in normal operation. In online testing, however, the circuit must be able to continue normal operations while testing is carried out. For online testing additional circuitry is generally added to the original circuit to determine whether the circuit is faulty or not. There are three fault models: line faults (or bit faults) [33], missing control faults, which is one instance of crosspoint faults [34], and missing gate faults [35]. In a line fault, the logic value of a line is flipped and produces faulty output. In a missing control fault, the control point of a gate either is not working or disappears from the gate, resulting in an incorrect value for the target output. In a missing gate fault, the entire gate does not work or the gate disappears from the circuit causing faulty output.

### 3. Related Work

In this section, we discuss related work on reversible sequential circuit design and on online testing of reversible combinational circuits.

**3.1. Related Work on Reversible Sequential Circuit Design.** There are many attempts at designing reversible latches, flip-flops, and sequential circuits. Some important but nonexhaustive work on reversible sequential logic design includes [25, 26, 36–42]. References [36–39] present reversible designs for latches and flip-flops using reversible gates and suggest that sequential circuits be synthesized by replacing the latches, flip-flops, and gates of the combinational part of the traditional irreversible designs with their reversible counterparts. Following this replacement approach, the work in [40] presents a design for a four-bit falling-edge-triggered universal register and the work in [41] presents a design for a four-bit level-triggered up counter with asynchronous parallel load. The work in [25] again uses the replacement technique and demonstrates that reversible circuits can be implemented using QCA technology with offline testing of the QCA-based sequential circuit. The work in [42] offers the first attempt to synthesize a sequential circuit using direct feedback of the state output and without using any flip-flops. Reference [42] also presents a design for a level-triggered up counter using positive polarity Reed-Muller expressions for representing the next states. The up counter design is more efficient than the replacement design in [41] in terms of both quantum cost and garbage outputs (outputs that are not used for the intended circuit realization). The next step to this work is presented in [26], which details the direct design of arbitrary sequential circuits using pseudo-Reed-Muller expressions for representing the next states. This work also introduces modifications making the circuit falling-edge triggered and asynchronous loadable. The up/down counter in [26] is better than that in [41] in terms of both quantum

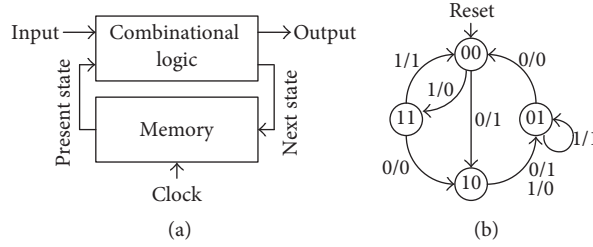


FIGURE 4: (a) Model of a sequential circuit. (b) State transition diagram of an example sequential circuit.

cost and garbage outputs, and the universal register in [26] improves upon the design in [40] again for both quantum cost and garbage output.

A further improvement of the work in [26] is presented in [43]. In [43], a new technique for representing the next states using exclusive-OR sum-of-product (ESOP) expressions is presented. Using this technique, designs for a four-bit falling-edge triggered up/down counter with asynchronous loading and a four-bit falling-edge triggered universal register are presented. Both designs in [43] offer improvements over those from [26] in terms of both quantum cost and ancilla input. This work is an extended version of [43].

**3.2. Related Work on Testing of Reversible Combinational Circuits.** Several works offer techniques for online testing of reversible combinational circuits. Some propose new testable gates which are then used to construct online testable reversible circuits. In [17] three new reversible gates are proposed, two of which are used to design an online testable block and the other is used to create a checker circuit. The checker circuit compares the two parity bits produced by the online testable blocks in order to test a single bit fault. In [23] an improved approach to single bit fault testing is proposed that does not require a checker circuit.

The most generalized approaches offer online testing of reversible circuits synthesized using NOT, CNOT, and Toffoli gates. In [44], all of the Toffoli gates are replaced by extended Toffoli gates, and two sets of CNOT gates and one additional parity line are added to the original circuit to achieve online testability for single line faults. In [45], a DFT- (design for testability-) based offline approach is proposed for detecting single missing gate faults. In [46], an online fault detection approach is proposed for detecting single missing gate faults. In [47], each of the Toffoli gates is accompanied by a duplicate Toffoli gate of the same size with the same control lines, but the target is placed on an additional parity line to make the Toffoli gate a testable Toffoli block. Two sets of CNOT gates are also used, for which the targets are the parity line. This approach detects all three types of faults.

To the best of our knowledge, no work has been reported in the literature on offline or online testing of reversible sequential circuits.

## 4. Synthesis of Sequential Reversible Circuits

In this section, we present our proposed method for synthesis of sequential reversible circuits.

**4.1. Representing the Next State.** Let  $Q$  and  $Q^+$  be the present state and the next state of a sequential circuit, respectively.  $Q^+$  can be expressed as a function of  $Q$  and  $Q^+$  as follows:

$$Q^+ = Q \oplus Q \oplus Q^+ = Q \oplus Q^*, \quad (2)$$

where

$$Q^* = Q \oplus Q^+. \quad (3)$$

We call  $Q^*$  the modified next state. In our proposed technique, the modified next state is determined using (3) and is expressed as a minimized ESOP expression of a function of the inputs and the present state. The next state is then expressed using (2).

**4.2. Synthesis Model.** Our model for the reversible synthesis of sequential circuits is shown in Figure 5. First the modified next state  $Q^*$  is generated, as a function of the inputs and present state, in the Modified Next State Logic section. The next state is generated using (2) in the Next State Logic section. The generated next state is loaded to the state output as the present state at the falling-edge of the clock in the Falling-Edge Trigger section. The asynchronous reset or load is carried out in the Asynchronous Load section. The feedback of the present state is generated in the Feedback section and the present state is fed back to the Modified Next State Logic section. Finally, in the Output Logic section, the output is generated as a function of the input and the present state. The following examples explain the use of this model.

## 5. Design Examples

In this section, we use three examples to illustrate the design model discussed in Section 4.2.

**5.1. Example 1: Sequential Circuit from Figure 4(b).** The truth table representing the next states ( $Q_1^+Q_0^+$ ), the modified next states ( $Q_1^*Q_0^*$ ), and the output ( $z$ ) of the sequential circuit in Figure 4(b) is shown in Table 1. The modified next states ( $Q_1^*Q_0^*$ ) and the output ( $z$ ) are minimized as ESOP expressions as follows:

$$Q_1^* = Q_0' \oplus xQ_1Q_0, \quad (4)$$

$$Q_0^* = 1 \oplus x'Q_1' \oplus Q_1'Q_0, \quad (5)$$

$$z = Q_0 \oplus x'. \quad (6)$$

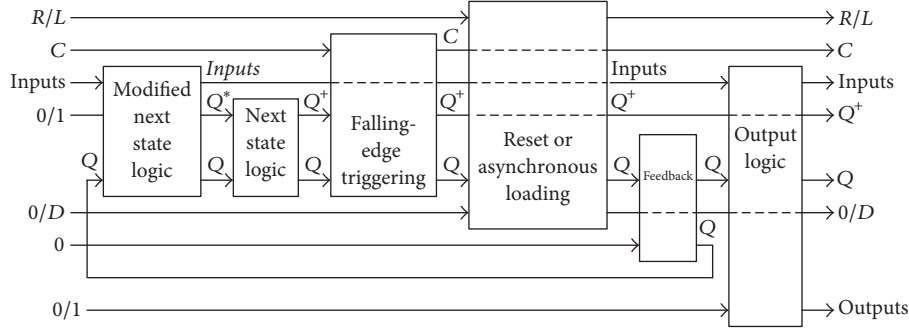


FIGURE 5: Model of reversible synthesis of sequential circuit.

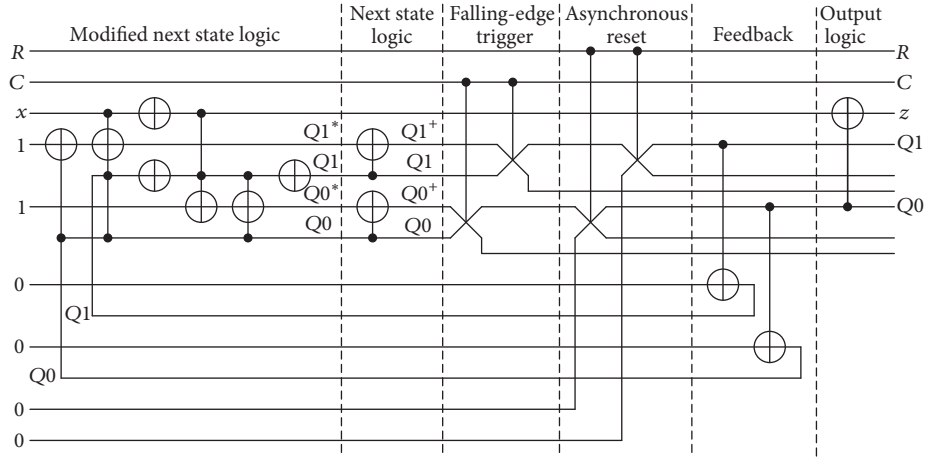


FIGURE 6: Reversible realization of the sequential circuit described by Figure 4(b).

TABLE 1: Truth table representing the next states, the modified next states, and the output of the sequential circuit from Figure 4(b).

$xQ_1Q_0$	$Q_1^+Q_0^+$	$Q_1^*Q_0^*$	$z$
000	10	10	1
001	00	01	0
010	01	11	1
011	10	01	0
100	11	11	0
101	01	00	1
110	01	11	0
111	00	11	1

Equations (4), (5), and (6) are then used to implement the state machine from Figure 4(b), resulting in the circuit shown in Figure 6. Each section of this circuit is explained below.

(1) *Modified Next State Logic.* The modified next states  $Q_1^*$  and  $Q_0^*$  (see (4) and (5)) are realized as functions of the input  $x$  and the fed-back present states  $Q_1$  and  $Q_0$  as a cascade of NOT, CNOT, and Toffoli gates. The quantum cost of this section is 28 and 2 ancilla inputs are required.

(2) *Next State Logic.* The next states are realized as  $Q_1^+ = Q_1 \oplus Q_1^*$  and  $Q_0^+ = Q_0 \oplus Q_0^*$  using the generated modified next states  $Q_1^*$  and  $Q_0^*$  and fed-back present states  $Q_1$  and  $Q_0$ . This requires two CNOT gates, and thus the quantum cost of this section is exactly equal to the number of bits in the state of the sequential circuit, since for each bit a CNOT gate is required and the quantum cost of the CNOT gate is 1. No ancilla inputs are required.

(3) *Falling-Edge Trigger.* The falling-edge triggering is achieved using two controlled swap gates controlled by the clock  $C$ . When the clock  $C = 1$  and reset  $R = 0$ , the fed-back states are passed to the state outputs, maintaining the state outputs unchanged. When the clock is set to  $C = 0$  and  $R = 0$  is maintained, then the generated next states are transmitted to the state outputs. Just after the next states arrive at the state output and the feedback of the present states arrives at the Falling-Edge Trigger section, the clock must be set to  $C = 1$  in order to maintain the new present state at the output. Thus, the changes of the present states occur at the falling-edge of the clock  $C$  and the duration of the period of  $C = 0$  has to be very carefully determined to avoid malfunction of the circuit. The quantum cost of this section is 5 times the number of bits required for the state of the sequential circuit, since for each bit a controlled swap gate is required and the quantum cost of



the controlled swap gate is 5. Thus the quantum cost of this section is 10, and no ancilla inputs are required.

(4) *Reset*. The reset is achieved using two controlled swap gates controlled by the reset input  $R$ . In the sequential circuit represented by Figure 4, the reset state is 00. During synchronous operation, the clock must be set to  $C = 1$  and the reset must be cleared to  $R = 0$  to maintain the present state unchanged. For the falling-edge triggering action, the clock and the reset must both be cleared to  $CR = 00$ . The reset action takes place asynchronously when the clock and the reset are both set to  $CR = 11$ . In this case, the constant inputs 00 are transmitted to the state outputs. At this point, the reset must be set to  $R = 0$  in order to maintain the state outputs unchanged. Again, the duration of the reset value  $R = 1$  must be carefully determined to avoid malfunction of the sequential circuit. The quantum cost of this section is also 5 times the number of bits in the state of the sequential circuit, since for each bit a controlled swap gate is required and the quantum cost of the controlled swap gate is 5. This section requires ancilla inputs exactly equal to the number of bits in the state. Thus the quantum cost of this section is 10 and 2 ancilla inputs are required.

(5) *Feedback*. The feedback of the present state is generated using two CNOT gates used as copying gates and is achieved by setting the target input to be 0. The quantum cost of this section is exactly equal to the number of the bits in the state of the sequential circuit, since one CNOT gate is required for each bit of the state and the quantum cost of the CNOT gate is 1. This section requires ancilla inputs exactly equal to the number of bits in the state of the sequential circuit, since for each bit of the state one constant-initialized input is required. The quantum cost of this section is 2 and 2 ancilla inputs are required.

(6) *Output Logic*. The output  $z$  is realized as a function of the input  $x$  and the present state  $Q0$  using (6).

The total quantum cost of the circuit in Figure 6 is 53 and 6 ancilla inputs are required. These values are compared with those of the replacement design approach in [26] and the direct design in [26] in Table 2. The % improvement over previous designs is calculated using the formula

$$\% \text{ Improvement} = \frac{\text{Previous} - \text{Present}}{\text{Previous}} \times 100. \quad (7)$$

From Table 2, we see that our new present design technique saves significantly on quantum cost and ancilla inputs as compared to both previous replacement design technique and the direct design technique presented in [26].

**5.2. Example 2: 4-Bit Up/Down Counter.** In this section, we illustrate the application of our technique to the design of a four-bit falling-edge triggered up/down counter with asynchronous load. The truth table representing the next states and the modified next states of the counter is shown in Table 3. The modified next states of the four-bit up counter from Table 3 are minimized as ESOP expressions as follows:

$$Q3^* = Q2Q1Q0, \quad (8)$$

TABLE 2: Comparison of reversible realization of the sequential circuit in Figure 6 with replacement design method and direct design method in [26].

	Quantum cost	Ancilla input
Replacement design [26]	96	18
Direct design [26]	88	9
Present design	53	6
% improvement over replacement design [26]	44.79	66.67
% improvement over direct design [26]	39.77	33.33

TABLE 3: Truth table representing the next states and the modified next states of a four-bit up counter [43].

Present state $Q3Q2Q1Q0$	Next state $Q3^+Q2^+Q1^+Q0$	Modified next state $Q3^*Q2^*Q1^*Q0^*$
0000	0001	0001
0001	0010	0011
0010	0011	0001
0011	0100	0111
0100	0101	0001
0101	0110	0011
0110	0111	0001
0111	1000	1111
1000	1001	0001
1001	1010	0011
1010	1011	0001
1011	1100	0111
1100	1101	0001
1101	1110	0011
1110	1111	0001
1111	0000	1111

$$Q2^* = Q1Q0, \quad (9)$$

$$Q1^* = Q0, \quad (10)$$

$$Q0^* = 1. \quad (11)$$

Similarly, the modified next states of the four-bit down counter are minimized as ESOP expressions as follows:

$$Q3^* = Q2'Q1'Q0', \quad (12)$$

$$Q2^* = Q1'Q0', \quad (13)$$

$$Q1^* = Q0', \quad (14)$$

$$Q0^* = 1. \quad (15)$$

The complete counter is implemented using the ESOP expressions (8) to (15) and the resulting reversible circuit is shown in Figure 7 where  $C$  is the clock input; the input  $L$  performs the asynchronous load with  $L = 0$  for normal operation and  $L = 1$  for asynchronous load; and the input  $M$  determines the count direction with  $M = 0$  for up and  $M = 1$  for down. The

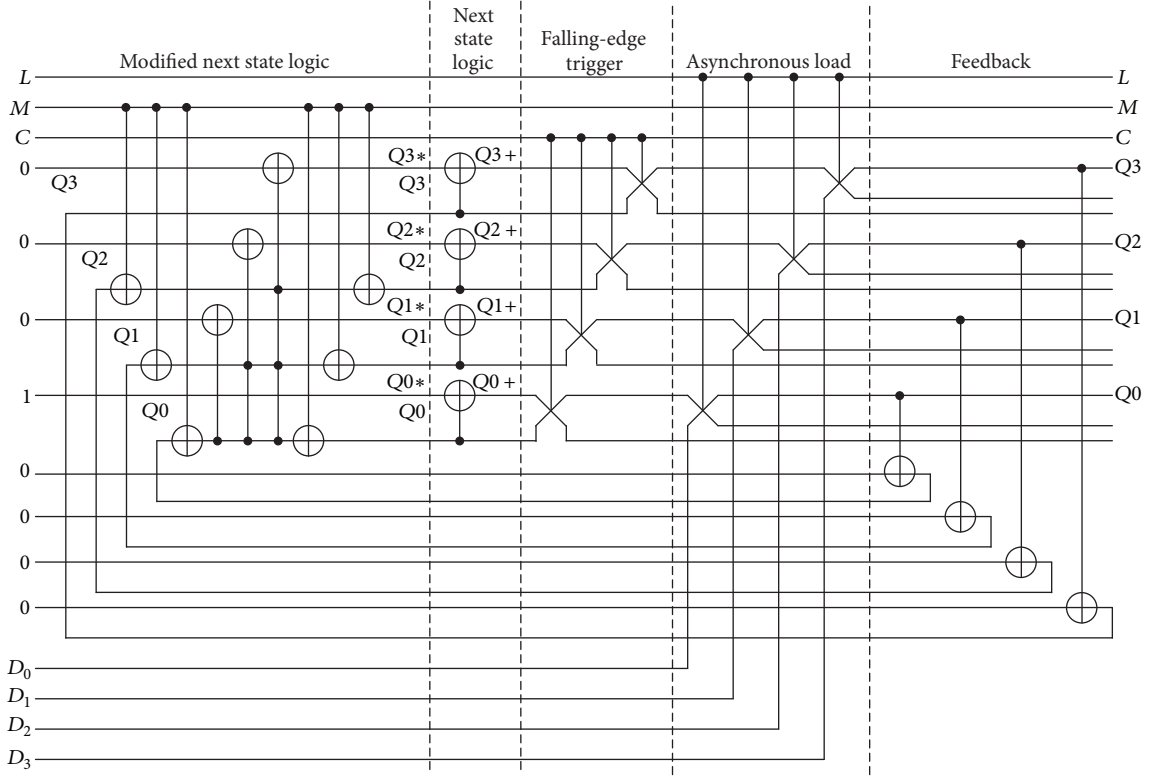


FIGURE 7: Reversible realization of the four-bit falling-edge triggered up/down counter with asynchronous load [43].

design is similar to that of the previous example (Example 1). The operation of the circuit is discussed below.

(1)  $C = 1$  and  $L = 1$ : *Asynchronous Load*. The data inputs  $D_3$ ,  $D_2$ ,  $D_1$ , and  $D_0$  are loaded to the present state outputs  $Q_3$ ,  $Q_2$ ,  $Q_1$ , and  $Q_0$  (respectively) through the controlled swap gates that make up the Asynchronous Load section of the circuit. After loading the input data, we set  $L = 0$  to maintain the present state output.

(2)  $M = 0$ : *Modified Next State Generation for Up Count*. The six  $M$ -controlled CNOT gates of the Modified Next State Logic section will not modify the logic values of their targets and the fed-back state values  $Q_3$ ,  $Q_2$ ,  $Q_1$ , and  $Q_0$  remain unchanged. These fed-back state values are used in the Modified Next State Logic section to generate modified next states  $Q_3^*$ ,  $Q_2^*$ ,  $Q_1^*$ , and  $Q_0^*$  (see (8) to (11)).

(3)  $M = 1$ : *Modified Next State Generation for Down Count*. The first three CNOT gates controlled by  $M$  complement the fed-back present states  $Q_3$ ,  $Q_2$ ,  $Q_1$ , and  $Q_0$ . These complemented values are used in the Modified Next State Logic section to generate modified next states  $Q_3^*$ ,  $Q_2^*$ ,  $Q_1^*$ , and  $Q_0^*$  (see (12) to (15)). The last three CNOT gates of the Modified Next State Logic section restore the fed-back present states for use in the Next State Logic section of the circuit.

The circuit in Figure 7 has a quantum cost of 74 and requires 8 ancilla inputs. The complexity of our design is

TABLE 4: Comparison of reversible realization of the four-bit falling-edge triggered up/down counter with asynchronous load with that in [26] (including data previously presented in [43]).

	Quantum cost	Ancilla input
Design of [26]	94	8
Present design	74	8
% improvement over design of [26]	21.28	0

compared with that in [26] as shown in Table 4. From the table, we see that the present design saves quantum cost with no increase of ancilla inputs.

**5.3. Example 3: 4-Bit Universal Register.** In this section, we illustrate the synthesis process for a four-bit falling-edge triggered universal register. The truth table representing the next states and the modified next states is shown in Table 5, where  $DR$  is the serial data input. The modified next states  $Q_3^*$ ,  $Q_2^*$ ,  $Q_1^*$ , and  $Q_0^*$  are minimized as ESOP expressions as follows:

$$Q_3^* = DR \oplus Q_3, \quad (16)$$

$$Q_2^* = Q_3 \oplus Q_2, \quad (17)$$

$$Q_1^* = Q_2 \oplus Q_1, \quad (18)$$

$$Q_0^* = Q_1 \oplus Q_0. \quad (19)$$

TABLE 5: Truth table representing the next states and the modified next states of a four-bit serial-in serial-out right-shift register [43].

Present state $DRQ3Q2Q1Q0$	Next state $Q3^+Q2^+Q1^+Q0$	Modified next state $Q3^*Q2^*Q1^*Q0^*$
00000	0000	0000
00001	0000	0001
00010	0001	0011
00011	0001	0010
00100	0010	0110
00101	0010	0111
00110	0011	0101
00111	0011	0100
01000	0100	1100
01001	0100	1101
01010	0101	1111
01011	0101	1110
01100	0110	1010
01101	0110	1011
01110	0111	1001
01111	0111	1000
10000	1000	1000
10001	1000	1001
10010	1001	1011
10011	1001	1010
10100	1010	1110
10101	1010	1111
10110	1011	1101
10111	1011	1100
11000	1100	0100
11001	1100	0101
11010	1101	0111
11011	1101	0110
11100	1110	0010
11101	1110	0011
11110	1111	0001
11111	1111	0000

Using (2), the next states corresponding to (16) to (19) can be determined as follows:

$$Q3^+ = DR \oplus Q3 \oplus Q3 = DR, \quad (20)$$

$$Q2^+ = Q3 \oplus Q2 \oplus Q2 = Q3, \quad (21)$$

$$Q1^+ = Q2 \oplus Q1 \oplus Q1 = Q2, \quad (22)$$

$$Q0^+ = Q1 \oplus Q0 \oplus Q0 = Q1. \quad (23)$$

Similarly, the next states of a four-bit serial-in serial-out left-shift register can be determined as follows, where  $DL$  is the serial data input:

$$Q3^+ = Q2, \quad (24)$$

$$Q2^+ = Q1, \quad (25)$$

$$Q1^+ = Q0, \quad (26)$$

$$Q0^+ = DL. \quad (27)$$

Equations (20) to (27) are implemented with reversible gates to build the circuit shown in Figure 8. Multiplexing is needed for implementing right-shift and left-shift: between  $DR$  and  $Q2$  for  $Q3^+$ , between  $Q3$  and  $Q1$  for  $Q2^+$ , between  $Q2$  and  $Q0$  for  $Q1^+$ , and between  $Q1$  and  $DL$  for  $Q0^+$ . This is implemented using four controlled swap gates controlled by the shift direction control input  $M$  in the Next State Logic section of the circuit. Implementations of the other sections are similar to those in Figure 7. The different modes of operations are explained below.

(1)  $C = 1$  and  $L = 0$ : *Serial-In Serial-Out Register*. If  $M = 0$ , expressions (20) to (23) are implemented by the Next State Logic section and data is shifted right. If  $M = 1$ , expressions (24) to (27) are implemented and data is shifted left. When  $C$  is changed to 0, the next states are passed to the present state outputs through the Asynchronous Load section of the circuit.

(2) *Serial-In Parallel-Out Register*. The operation is similar to step (1) and the present state outputs are taken in parallel.

(3)  $C = 1$  and  $L = 1$ : *Parallel-In Parallel-Out Register*. The parallel input values  $D3$ ,  $D2$ ,  $D1$ , and  $D0$  are loaded to the present state outputs through the Asynchronous Load section by setting  $L = 1$  and then changing to  $L = 0$ . Outputs are taken in parallel.

(4) *Parallel-In Serial-Out Register*. The asynchronous load operation is similar to step (3). After setting  $L = 0$ , if  $C$  is changed to 0, the states will be shifted to either right or left based on the value of  $M$ .

The quantum cost of the circuit in Figure 8 is 74 and 14 ancilla inputs are required. The circuit complexity is compared with that of previous work in Table 6. From the table, we see that the present design saves both quantum cost and ancilla inputs as compared to the replacement design in [40]. It is to be noted that in [40] the number of ancilla inputs is not mentioned. We count only the 0-initialized working inputs as ancilla inputs and do not include a large number of control inputs as ancilla inputs. The present design also saves on quantum cost as compared to the direct design in [26], but requires slightly more ancilla inputs.

## 6. Online Testable Design of Sequential Circuits

In this section, we introduce our proposed technique for online testing of single line faults in sequential circuit. We focus only on single line faults because some of the missing control fault and missing gate fault situations can be treated as single line faults. For example, if a missing control fault inverts the original target output of the gate, then that fault can be treated as a single line fault at the target output of the gate. Similarly, if a gate produces an inverted output at the



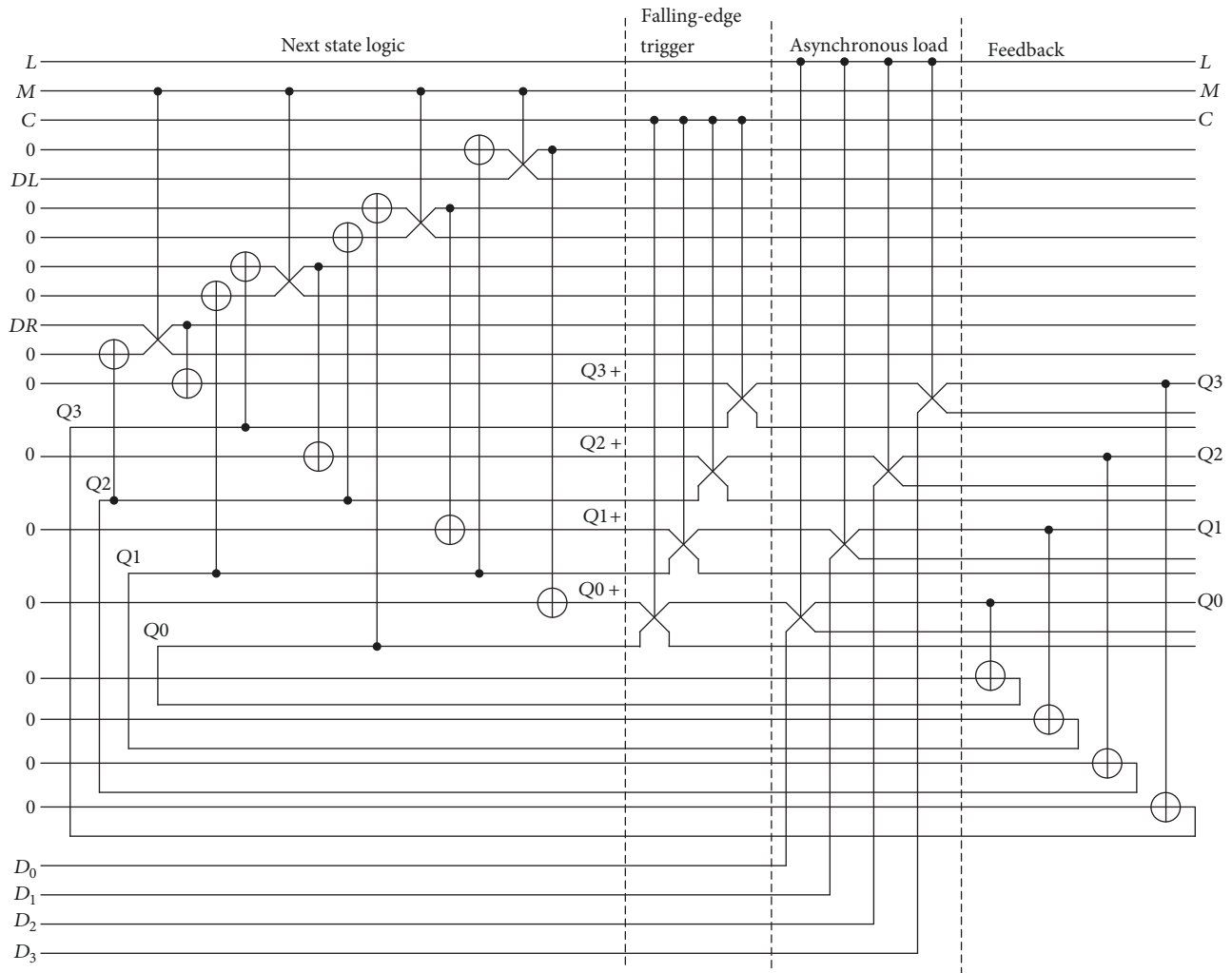


FIGURE 8: Reversible realization of the four-bit falling-edge triggered universal register [43].

TABLE 6: Comparison of reversible realization of the four-bit falling-edge triggered universal register with that of the replacement design in [40] and the direct design in [26].

	Quantum cost	Ancilla input
Replacement design [40]	220	18
Direct design [26]	112	12
Present design	74	14
% improvement over replacement design [40]	66.36	22.22
% improvement over direct design [26]	33.93	-16.67

target, then the missing gate will not produce that inversion and this condition can be treated as a single line fault at that position. Thus online detection of single line faults also detects some missing control and missing gate faults.

*6.1. Online Testing of Single Line Fault in Toffoli Circuits.* A Toffoli circuit is a cascade of NOT, CNOT, and Toffoli gates. In [44], a method for online detection of single line fault in Toffoli circuits is presented. The method in [44] is as follows:

- (1) A 0-initialized parity line  $L$  is added to the given circuit.
- (2) Assume that the given circuit has  $p$  inputs/outputs. At the beginning of the circuit,  $p$  CNOT gates are added, where the controls of the CNOT gates are the  $p$  input lines and the target of all the CNOT gates is the parity line  $L$ .
- (3) All CNOT and Toffoli gates of the given circuit are replaced by their corresponding extended versions (EFGs and ETGs, respectively). The second target of all gates is the parity line  $L$ .
- (4) The NOT gates in the given circuit are retained. If the number of NOT gates in the given circuit is odd, then an extra NOT gate is added at the end of the parity line  $L$ .

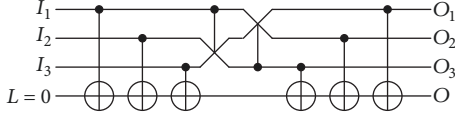


FIGURE 9: Online testing of a single line fault in a Fredkin circuit.

- (5) At the end of the circuit,  $p$  CNOT gates are added, where the controls of the CNOT gates are the  $p$  input lines and the target of all the CNOT gates is the parity line  $L$ .
- (6) If the circuit is fault free then the parity output will be 0, otherwise it will be 1.

**6.2. Online Testing of Single Line Faults in Fredkin Circuits.** A Fredkin circuit is a cascade consisting of only Fredkin (controlled swap) gates. To our knowledge, there is no existing work on offline or online testing of Fredkin circuits. We propose here a technique for online testing of single line faults in Fredkin circuits. The procedure is as follows:

- (1) A 0-initialized parity line  $L$  is added to the given circuit.
- (2) Assume that the given circuit has  $p$  inputs/outputs. At the beginning of the circuit,  $p$  CNOT gates are added, where the controls of the CNOT gates are the  $p$  input lines and the target of all the CNOT gates is the parity line  $L$ .
- (3) At the end of the circuit,  $p$  CNOT gates are added, where the controls of the CNOT gates are the  $p$  input lines and the target of all the CNOT gates is the parity line  $L$ .
- (4) If the circuit is fault free then the parity output will be 0, otherwise it will be 1.

A simple example of the above technique is shown in Figure 9. In a controlled swap gate the target inputs are either swapped or not swapped depending on the control value. Thus the outputs of a controlled swap gate are a permutation of the inputs. Similarly, the outputs of a Fredkin circuit are a permutation of the inputs. In Figure 9, the parity output is computed as

$$O = (I_1 \oplus I_2 \oplus I_3) \oplus (I_1 \oplus I_2 \oplus I_3). \quad (28)$$

If there is no single line fault in the circuit, then the parity output  $O$  of (28) will be 0, since each pair (input, output) will be canceled. If any single line fault occurs anywhere in the circuit, then any of  $I_n$  where  $n \in \{1, 2, 3\}$  will be inverted at the output. Thus, the parity output  $O$  will be 1, since  $I_n \oplus I'_n = 1$ . If a single line fault occurs at any point in the parity line the parity output  $O$  will also be 1, since if any subexpression in (28) is inverted, then  $O$  will be 1.

**6.3. Online Testing of Single Line Faults in Sequential Circuits.** In our proposed method for online testing of single line faults in sequential circuits, we use the technique in [44] for testing

the Toffoli circuits in the (i) Modified Next State Logic and Next State Logic sections together and test the (ii) Feedback section and (iii) Output Logic section separately. We then use our proposed technique for online testing of single line faults in Fredkin circuits in the Falling-Edge Trigger and Reset/Asynchronous Load sections together.

The online testable version of the reversible sequential circuit in Figure 6 is shown in Figure 10. The design of the testable circuit is described below:

- (1) The Modified Next State Logic and Next State Logic sections together are a Toffoli circuit. We make this circuit testable by adding 0-initialized parity input  $L_{MN}$ . This portion of the circuit has five active inputs:  $x$ , two 1-initialized inputs,  $Q1$ , and  $Q0$ . Five CNOT gates are added at the beginning of this portion and another five CNOT gates are added at the end of this portion. One CNOT gate and three Toffoli gates in the original circuit are replaced by their corresponding extended versions. There are three NOT gates in the original circuit, so an extra NOT gate is added along the parity line. If any single line fault occurs in this section, then the parity output  $O_{MN}$  will be 1, otherwise it will be 0. The overhead quantum cost of this section is 20.
- (2) The Falling-Edge Trigger and Asynchronous Reset sections together are a Fredkin circuit. We make this circuit testable by adding 0-initialized parity input  $L_{TL}$ . This portion of the circuit has eight active inputs:  $R$ ,  $C$ ,  $Q1^+$ ,  $Q1$ ,  $Q0^+$ ,  $Q0$ , and two 0-initialized inputs. Eight CNOT gates are added at the beginning of this portion and another eight CNOT gates are added at the end of this portion. If any single line fault occurs in this section then the parity output  $O_{TL}$  will be 1, otherwise it will be 0. The overhead quantum cost of this section is 16.
- (3) The Feedback section of the circuit is a Toffoli circuit. We make this circuit testable by adding 0-initialized parity input  $L_F$ . This portion of the circuit has four active inputs:  $Q1$ ,  $Q0$ , and two 0-initialized inputs. This portion of the circuit is made testable using the technique described in Section 6.1. If any single line fault occurs in this section then the parity output  $O_F$  will be 1, otherwise it will be 0. The overhead quantum cost of this section is 10.
- (4) The Output Logic section of the circuit is a Toffoli circuit. We make this circuit testable by adding 0-initialized parity input  $L_O$ . This portion of the circuit has two active inputs:  $x$  and  $Q0$ . This portion of the circuit is made testable using the technique described in Section 6.1. If any single line fault occurs in this section then the parity output  $O_O$  will be 1, otherwise it will be 0. The overhead quantum cost of this section is 5.
- (5) Multiple line faults in different sections can be detected simultaneously. Therefore, from the parity outputs, fault sections can be located.

TABLE 7: Results showing gate count (GC) and quantum cost (QC) for a selection of sequential benchmarks, both before and after adding gates for testability.

Filename	Inputs	Outputs	States	Total before testability			Total after testability			
				GC	QC	Number of lines	GC	QC	Number of lines	% increase in QC
bbara	4	2	10	46	1096	14	116	1242	18	13.30
bbsse	7	7	16	99	1361	22	185	1629	26	19.70
bbtas	2	2	6	21	203	10	73	285	14	40.40
beecount	3	4	7	53	577	13	111	729	17	26.30
cse	7	7	16	137	2922	22	223	3266	26	11.80
dk14	3	5	7	104	973	14	164	1229	18	26.30
dk512	3	5	4	58	491	12	104	645	16	31.40
donfile	2	1	24	67	877	13	145	1069	17	21.90
ex1	9	19	20	355	3979	38	483	4797	42	20.60
ex2	2	2	19	97	1924	14	177	2178	18	13.20
ex3	2	2	10	51	715	12	117	867	16	21.30
ex4	6	9	14	74	618	23	162	838	27	35.60
ex5	2	2	9	43	631	12	109	767	16	21.60
ex7	2	2	10	48	680	12	114	826	16	21.50
keyb	7	2	19	110	4946	19	200	5236	23	5.90
lion9	2	1	9	34	505	11	98	621	15	23.00
lion	2	1	4	15	79	7	51	137	11	73.40
planet1	7	19	48	646	4763	38	784	6169	42	29.50
planet	7	19	48	646	4763	38	784	6169	42	29.50
sla	8	6	20	97	3668	24	197	3942	28	7.50
s1	8	6	20	197	3864	24	297	4338	28	12.30
s8	4	1	5	36	1036	11	90	1150	15	11.00
sand	11	9	32	288	4941	30	400	5609	34	13.50
shiftreg	1	1	8	17	29	8	65	99	12	241.40
sse	7	7	16	98	1378	22	184	1644	26	19.30
styr	9	10	30	338	6553	29	448	7319	33	11.70
tav	4	4	4	23	403	12	69	487	16	20.80
tbk	6	3	32	110	3621	19	200	3911	23	8.00
train11	2	1	11	47	742	11	111	884	15	19.10
train4	2	1	4	18	103	7	54	167	11	62.10

The total overhead quantum cost of the circuit in Figure 10 is  $20 + 16 + 10 + 5 = 51$ . The quantum cost of the original circuit in Figure 6 is 53. Therefore, the quantum cost overhead of the testable circuit is only 96.23%.

## 7. Experimental Results

The process for designing the sequential reversible circuit and adding the gates for testability was applied to several sequential benchmarks from the MCNC suite of benchmarks [48]. As shown in Table 7, the additional circuitry required for making the sequential reversible circuit testable adds an average of 30% overhead, in terms of quantum cost. The highest percentage overhead occurs for the smallest circuits; this is a result of needing to copy the input and output lines several times. In larger circuits, the percentage overhead is significantly lower.

## 8. Conclusion

In this work, an improved synthesis approach for sequential reversible circuits is presented. Three design examples are demonstrated, including an arbitrary sequential circuit with 2-bit states, 1-bit input, and 1-bit output (Figure 4); a four-bit falling-edge triggered up/down counter with asynchronous load and a four-bit falling-edge triggered universal register are shown. We compare the quantum cost and the ancilla inputs of the three designs with both the replacement design technique and the direct design technique reported in [26]. The new design of the sequential circuit in Figure 4 saves 44.79% and 39.77% in quantum cost and 66.67% and 33.33% in ancilla inputs as compared to the replacement design and direct design in [26], respectively. The new counter design saves 21.28% in quantum cost with the same number of ancilla inputs as compared to the design in [26]. The new register design saves 66.36% in quantum cost and 22.22% in ancilla

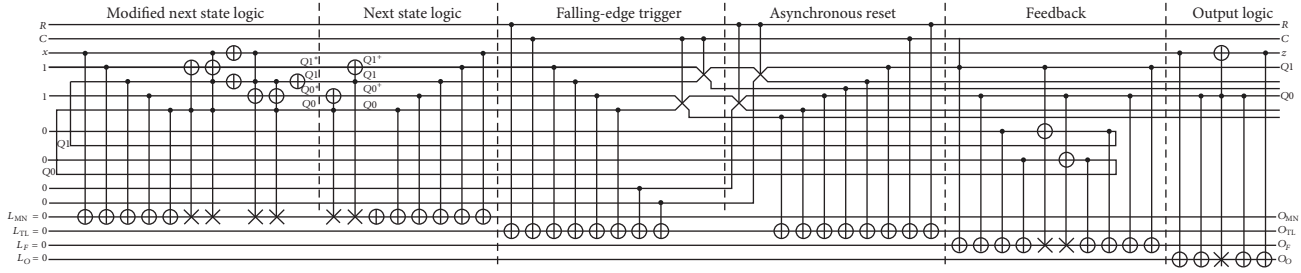


FIGURE 10: Online testable design of reversible sequential circuit of Figure 6 for single line fault testing.

inputs as compared to the replacement design in [40]. The register design also saves 33.93% in quantum cost with a 16.67% increase in ancilla inputs over that in [26].

We also present an online testable design for sequential reversible circuits for detecting single line faults. The technique presented in [44] is used for testing Toffoli-cascade portions of the circuit. For testing sections of the circuit built from controlled swap gates we propose a new testing technique. The testable design can simultaneously detect multiple single line faults in different sections of the circuit. As shown in Figure 10 the online testable version of Figure 6 requires only 96.23% quantum cost overhead. Furthermore, tests of several benchmarks indicate that as the circuit complexity grows the percentage overhead required for testability decreases.

Future work includes automation of the mapping process and consideration of online testability for all three fault models.

## Disclosure

The first author did most of his work at the University of Lethbridge while on sabbatical from East West University, Dhaka, Bangladesh.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The second author was supported by a grant from Canada's National Science and Engineering Research Council (NSERC) while pursuing this research.

## References

- [1] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, p. 8, 1965.
- [2] E. P. DeBenedictis, "The Boolean logic tax," *Computer*, vol. 49, no. 4, Article ID 7452299, pp. 79–82, 2016.
- [3] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM Journal of Research and Development*, vol. 44, pp. 183–191, 2000.
- [4] C. H. Bennett, "Logical reversibility of computation," *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525–532, 1973.
- [5] A. DeVos and Y. V. Rentergem, "Power consumption in reversible logic addressed by a ramp voltage," in *Proc. 15th Int. Workshop Power Timing Model., Optim. Simul.*, ser. LNCS 3728, pp. 207–216, 2005.
- [6] J. Ren and V. K. Semenov, "Progress with physically and logically reversible superconducting digital circuits," *IEEE Transactions on Applied Superconductivity*, vol. 21, no. 3, pp. 780–786, 2011.
- [7] J. Ren, V. K. Semenov, Y. A. Polyakov, D. V. Averin, and J.-S. Tsai, "Progress towards reversible computing with nsquid arrays," *IEEE Transactions on Applied Superconductivity*, vol. 19, no. 3, pp. 961–967, 2009.
- [8] N. Kostinski, M. P. Fok, and P. R. Prucnal, "Experimental demonstration of an all-optical fiber-based Fredkin gate," *Optics Express*, vol. 34, no. 18, pp. 2766–2768, 2009.
- [9] C. Taraphdar, T. Chattopadhyay, and J. N. Roy, "Mach-Zehnder interferometer-based all-optical reversible logic gate," *Optics & Laser Technology*, vol. 42, no. 2, pp. 249–259, 2010.
- [10] X. Ma, J. Huang, C. Metra, and F. Lombardi, "Reversible gates and testability of one dimensional arrays of molecular QCA," *Journal of Electronic Testing*, vol. 24, no. 1-3, pp. 1244–1245, 2008.
- [11] X. Ma, J. Huang, C. Metra, and F. Lombardi, "Detecting multiple faults in one-dimensional arrays of reversible QCA gates," *Journal of Electronic Testing*, vol. 25, no. 1, pp. 39–54, 2009.
- [12] S. Bandyopadhyay, "Nanoelectric implementation of reversible and quantum logic," *Superlattices Microstruct.*, vol. 23, no. 3-4, pp. 445–464, 1998.
- [13] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 1st edition, 2000.
- [14] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 6, pp. 710–722, 2003.
- [15] D. Maslov and G. W. Dueck, "Reversible cascades with minimal garbage," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 11, pp. 1497–1509, 2004.
- [16] K. N. Patel, 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, no. 8, pp. 410–416, 2004.
- [17] D. P. Vasudevan, P. K. Lala, J. Di, and J. P. Parkerson, "Reversible logic design with online testability," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 2, pp. 406–414, 2006.



- [18] V. V. Shende, S. S. Bullock, and I. L. Markov, "Synthesis of quantum-logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1000–1010, 2006.
- [19] P. Gupta, A. Agarwal, and N. K. Jha, "An algorithm for synthesis of reversible logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 11, pp. 2317–2330, 2006.
- [20] A. K. Prasad, V. V. Shende, I. L. Markov, J. P. Hayes, and K. N. Patel, "Data structures and algorithms for simplifying reversible circuits," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 2, no. 4, pp. 277–293, 2006.
- [21] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne, "Quantum circuit simplification and level compaction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 3, pp. 436–444, 2008.
- [22] D. Große, R. Wille, G. W. Dueck, and R. Drechsler, "Exact multiple-control Toffoli network synthesis with SAT techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 5, pp. 703–715, 2009.
- [23] S. N. Mahammad and K. Veezhinathan, "Constructing online testable circuits using reversible logic," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 1, pp. 101–109, 2010.
- [24] T. Toffoli, "Reversible computing," MIT Lab. Comput. Sci, Cambridge, MA, USA, 1980.
- [25] H. Thapliyal, N. Ranganathan, and S. Kotiyal, "Design of testable reversible sequential circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 7, pp. 1201–1209, 2013.
- [26] M. H. A. Khan, "Design of reversible synchronous sequential circuits using pseudo Reed-Muller expressions," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 11, pp. 2278–2286, 2014.
- [27] A. Barenco, C. H. Bennett, R. Cleve et al., "Elementary gates for quantum computation," *Physical Review A: Atomic, Molecular and Optical Physics*, vol. 52, no. 5, pp. 3457–3467, 1995.
- [28] J. A. Smolin and D. P. DiVincenzo, "Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate," *Physical Review A: Atomic, Molecular and Optical Physics*, vol. 53, no. 4, pp. 2855–2856, 1996.
- [29] D. M. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-control Toffoli gates," in *Proceedings of the 41st IEEE International Symposium on Multiple-Valued Logic (ISMVL '11)*, pp. 288–293, Finland, May 2011.
- [30] J.-L. Chen, X.-Y. Zhang, L.-L. Wang, X.-Y. Wei, and W.-Q. Zhao, "Extended Toffoli gate implementation with photons," in *Proceedings of the 2008 9th International Conference on Solid-State and Integrated-Circuit Technology (ICSICT '08)*, pp. 575–578, China, October 2008.
- [31] T. Sasao, "Logic Synthesis and Optimization," in *ch. AND-EXOR Expressions and Their Optimization*, pp. 287–312, Kluwer, Norwell, MA, USA, 1993.
- [32] N. M. Nayeem and J. E. Rice, "Improved ESOP-based synthesis of reversible logic," in *Reed-Muller Workshop*, May 2011.
- [33] J. E. Rice, "An overview of fault models and testing approaches for reversible logic," in *Proceedings of the 14th IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing (PACRIM '13)*, pp. 125–130, Victoria, Canada, August 2013.
- [34] J. Zhong and J. C. Muzio, "Analyzing fault models for reversible logic circuits," in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 2422–2427, Vancouver, BC, Canada, 2006.
- [35] I. Polian, J. P. Hayes, T. Fiehn, and B. Becker, "A family of logical fault models for reversible circuits," in *Proceedings of the 14th Asian Test Symposium (ATS '05)*, pp. 422–427, Kolkata, India, December 2005.
- [36] J. E. Rice, "A new look at reversible memory elements," in *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems (ISCAS '06)*, pp. 243–246, May 2006.
- [37] S. K. S. Hari, S. Shroff, S. N. Mahammad, and V. Kamakoti, "Efficient building blocks for reversible sequential circuit design," in *Proceedings of the 2006 49th Midwest Symposium on Circuits and Systems (MWSCAS '06)*, pp. 437–441, IEEE, Puerto Rico, August 2006.
- [38] H. Thapliyal and A. P. Vinod, "Design of reversible sequential elements with feasibility of transistor implementation," in *Proceedings of the 2007 IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 625–628, USA, May 2007.
- [39] M.-L. Chuang and C.-Y. Wang, "Synthesis of reversible sequential elements," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 3, no. 3, pp. 1–19, 2008.
- [40] H. Thapliyal and N. Ranganathan, "Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 6, no. 4, pp. 14:1–14:35, 2008.
- [41] M. Haghparast and M. S. Gharajeh, "Design of a nanometric reversible 4-bit binary counter with parallel load," *Australian Journal of Basic and Applied Sciences*, vol. 5, no. 7, pp. 63–71, 2011.
- [42] M. H. A. Khan and M. Perkowski, "Synthesis of reversible synchronous counters," in *Proceedings of the 41st IEEE International Symposium on Multiple-Valued Logic (ISMVL '11)*, pp. 242–247, Finland, May 2011.
- [43] M. H. A. Khan and J. E. Rice, "Improved synthesis of reversible sequential circuits," in *Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS '16)*, pp. 2302–2305, Canada, May 2016.
- [44] N. M. Nayeem and J. E. Rice, "Online fault detection in reversible logic," in *Proceedings of the 2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT '11)*, pp. 426–434, Vancouver, BC, Canada, October 2011.
- [45] J. P. Hayes, I. Polian, and B. Becker, "Testing for missing-gate faults in reversible circuits," in *Proceedings of the 13th Asian Test Symposium (ATS '04)*, pp. 100–105, Washington, DC, USA, 2004.
- [46] D. K. Kole, H. Rahaman, D. K. Das, and B. B. Bhattacharya, "Synthesis of online testable reversible circuit," in *Proceedings of the 13th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS '10)*, pp. 277–280, Austria, April 2010.
- [47] M. A. Nashiry, G. G. Bhaskar, and J. E. Rice, "Online testing for three fault models in reversible circuits," in *Proceedings of the 45th IEEE International Symposium on Multiple-Valued Logic (ISMVL '15)*, pp. 8–13, Waterloo, Canada, May 2015.
- [48] P. Fiser, *Collection of Digital Design Benchmarks*, 2017, <https://ddd.fit.cvut.cz/prj/Benchmarks/>.



