PERTANIKA JOURNALS

Review Article

A Comprehensive and Comparative Study on Online Testability for Reversible Logic

Hari Mohan Gaur[1]*, Ashutosh Kumar Singh[2] and Umesh Ghanekar[1]

[1]Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra, Haryana, India

[2]Department of Computer Applications, National Institute of Technology, Kurukshetra, Haryana, India

ABSTRACT

Reversible logic is one of the rising fields for low-power electronic devices. Testing of these devices is a significant issue where the researchers are at par with the latest innovations in the field. However, new technology gives birth to new challenges, and in this field too, several fault models have arisen. Several online testing methods have been proposed for their detection, which are scaled on various performance parameters. This paper provides a comparative study of online testability for reversible logic. We bring together a review of fault models, performance parameters and online testing strategies from the literature with the aim of obtaining a near optimal solution by efficiently exploring the entire search space. We critically analyze a range of online testing strategies reported by researchers using parity preservation and generation, dual-rail coding and concurrent error detection schemes. These strategies are presented in two broad classifications, namely designing with novel gates and designing with existing circuits. All the techniques are explained in detail with a brief mathematical illustration. A comparison of experimental results based on the available number of benchmarks and combinational logic circuits is presented. The best possible strategy is highlighted on behalf of performance parameters.

Keywords: Reversible logic, online testing, fault models, performance parameters, comparative study

INTRODUCTION

Testing of irreversible circuits has been extensively studied in the 70s; however, reversible logic has been studied by researchers only to some extent. More attention is required to meet the demand of high-speed low-power electronic devices. Reversible logic ensures nearly energy-free computation by preventing

loss of information in the form of heat as in irreversible operations (Bennett, 1973; Landauer, 1961). Since testing guarantees the true functioning of these devices, it is very important to attain the desired results. Testing of reversible logic circuits is achieved at the cost of an increase in gate count, quantum cost, ancilla inputs and garbage output. Although gate count and quantum cost have improved over the years, the argument for ancilla inputs and garbage output persists. One resembles an extra input bit and the other an extra output bit; both are sources of power loss in reversible circuits.

Reversible logic performs bijective functions, where a unique output state is obtained from every input state and therefore, the number of inputs should be the same as the number of outputs. The outcome is fully controllable and observable operations, which, in turn make the testing of reversible logic circuits easier than that of irreversible logic circuits. Also, some challenges have arisen as a consequence of new technology that should be considered during the entire testable design process. Some major challenges pertaining to online testability include:

- Design complexity due to high density of gates,

- A variety of new fault models; the maximum number of faults should be covered in one design,

- Garbage and ancilla input minimization, the outcomes of implementing a reversible function from an irreversible function,

- Very low signal levels as compared to that of irreversible logic circuits (Polian et al., 2005) and

- Fanout and feedback, a technology problem which is not permitted in reversible circuits.

There are two possible approaches of testing for any system, and in reversible logic, the first is online testing and the other is offline testing. A circuit is said to be online testable if it is able to find a fault within the circuit during its operation. In the case of offline testing, a number of test vectors are applied after taking out the circuit from its usual operation, for which correct output values are known. All the existing reversible logic-testing approaches that have been proposed previously are divided into these two major categories as shown in Figure 1. Further classification is given on the basis of different methodology of testing reversible logic circuits. Designing with novel gates and designing with the existing circuit are two methods in case of online testing. Test-set generation and compaction ATPG (Patel et al., 2004, Polian et al., 2005 & Tabei et al., 2005) and DFT methodology (Rahman et al., 2007, & Rahman et al., 2011) are found prominent in case of offline testing. We are concerned here with online testing approaches as online testing is gaining ground over offline testing. The latter approach entails utmost 2^n test vectors to test a circuit with n number of inputs, which probably becomes hugely time consuming and places the emphasis on categorizing various strategies on the basis of quantum cost, gate count, ancilla input and garbage output. Quantum cost is not reported by most researchers, but gate count has a proportional relationship with it.
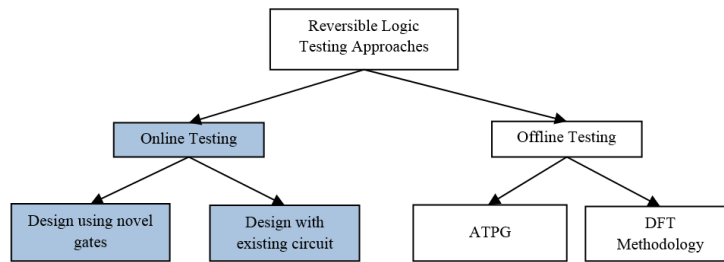
Figure 1. Classification of testing approaches.

In this paper, we first consider some background details, which include the basics of reversible logic, fault, fault models and performance parameters pertaining to online testing of reversible logic. Then, we demonstrate various testing approaches in online testing of reversible logic circuits from the literature by differentiating them into two categories. These approaches are then summarized and compared in a table based on their features, highlighting the optimum among them on behalf of performance parameters in two consecutive sections. Finally, the last section provides a comparative study of all the approaches on available work and future directions related to performance parameters with fault models.

BACKGROUND

Reversible Functions and Reversible Gates

A logic function with n Boolean variables is a reversible function if it maps distinct input to distinct output. There should be $n \times n$ bijective function mapping between input and output. In general, a truth table or a permutation can be used to represent a reversible function (Chua & Singh, 2014) as shown in the example in Table 1, where each output (x,y) is assigned to a unique input (a,b) without any replication.

Table 1
Reversible Function

Input		Output	
a b	Permutation	x y	Permutation
0 0	0	0 0	0
0 1	1	0 1	1
1 0	2	1 1	3
1 1	3	1 0	2

A reversible gate realizes a reversible function. A k input k output gate that produces k distinct output from its k distinct input functions is called a $k \times k$ reversible gate. There is one-to-one mapping between input and output vectors and therefore, the input state can be reconstructed from its output state. Various reversible gates are given in the literature, which is used for synthesizing reversible circuits using different algorithms (Shende et al., 2003).

Some of the commonly used are the NOT (Feynman, 1985), CNOT (Feynman, 1985), Toffoli (Toffoli, 1980), Fredkin (Fredkin & Toffoli, 1982) and Peres (Peres, 1985) gates, which are demonstrated in Figure 2, with inputs a, b and c and their respective response x, y and z shown in Table 2.
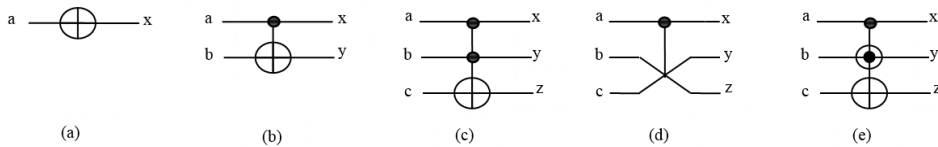

Figure 2. Reversible gates (a) NOT, (b) CNOT, (c) Toffoli, (d) Fredkin and (e) Peres.

Table 2
Responses of Reversible Gates for Their Corresponding Inputs

	x	y	z
NOT	a'	-	-
C-NOT	a	ab	-
Toffoli	a	b	$c\oplus ab$
Fredkin	a	a'b+ac	ab+a'c
Peres	a	$a\oplus b$	$ab\oplus c$

Fault and Fault Models

Faults are any kind of imperfection in a circuit that affects the functional behavior of a system. They are classified into two main categories (Hurst, 1998). The first category includes permanent faults that affect the functional behavior of a system due to incorrect interconnections, designing and masking, breakage etc. The second includes non-permanent faults that affect the functional behavior of a system for a finite period of time. Non-permanent faults occur due to environmental conditions like temperature, humidity, dust etc., which are called transient faults or are due to non-environmental conditions like ageing, loose connections, critical timing (hazards and race) etc. which are called intermittent faults. The online approach finds its application to detect permanent as well as non-permanent faults. In this review paper, we focused only on non-permanent faults that occurred only once during the operation of a circuit.

A fault model describes the type of fault that occurs in a circuit. It identifies the target of testing. There are several fault models proposed in the literature. In this paper, we focused only on structural fault models in reversible logic circuits. The following is a detailed discussion of fault models as depicted in Figure 3 from the left to the right.
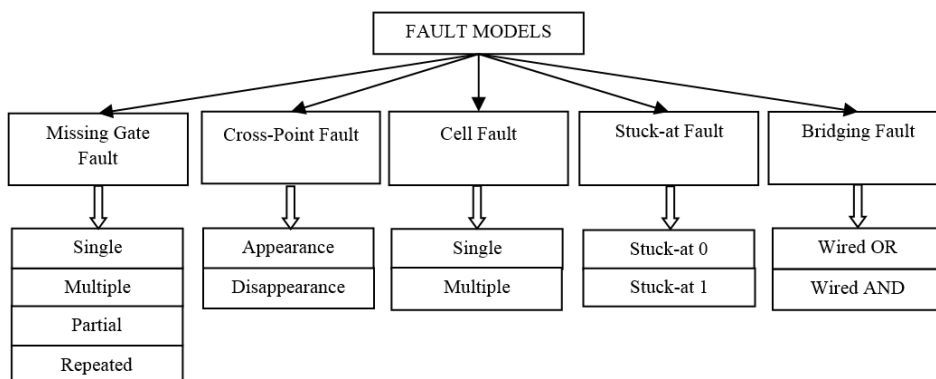
Figure 3. Fault models in reversible circuits.

Missing gate fault. A missing gate fault is defined as complete disappearance of a reversible gate from a circuit. If a single gate disappears from a circuit, it is called a 'single missing gate fault'. If two or more consecutive gates disappear, the fault is called a 'multiple missing gate fault'. Figure 4a shows that the first gate is missing and Figure 4b shows the first two consecutive gates missing. These faults are transient fault models that appear in a circuit due to short or mistuned input pulses (Polian, Hayes, Fiehn, & Becker, 2005).The un-faulty/ faulty value is written on the wire at each stage where the value is likely to be changed.

Missing gate faults may also occur as repeated gate faults when any gate replicates its functionality in certain instances, for instance, in a reversible circuit as shown in Figure 4c, where the second gate repeated its functionality twice. If the number of instances is even, it affects the circuit in a similar way as in the case of a single missing gate fault and if the instances are odd in number, then there is no change in the functionality of the circuit and a partial missing gate fault occurs when any control point loses its control from a gate in a reversible circuit as shown in Figure 5b.This fault model appears in a circuit due to long or duplicated input pulses.
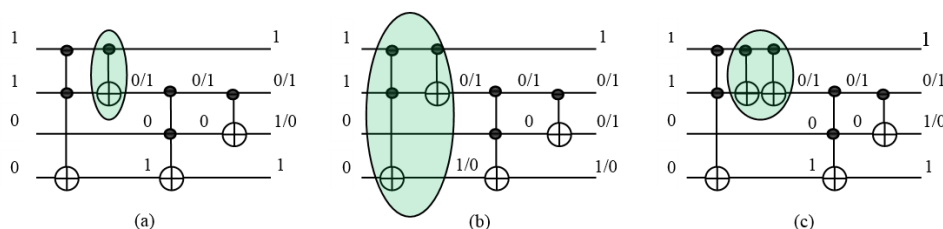

Figure 4. Illustration of fault models (a) Single missing gate fault (b) Single missing gate fault and (c) Repeated gate fault.

Cross-point fault. This fault model is related to the missing or extra connection at the cross-point or control point of a gate in a reversible circuit (Zhong, & Muzio, 2009). There are two types of cross-point faults that may occur in a circuit. They are the appearance fault,

which occurs when an extra control point is added on a gate, as shown in Figure 5a and the disappearance fault, which occurs when one or more control points are missing from a gate as shown in Figure 5b, similar to a partial missing gate fault.
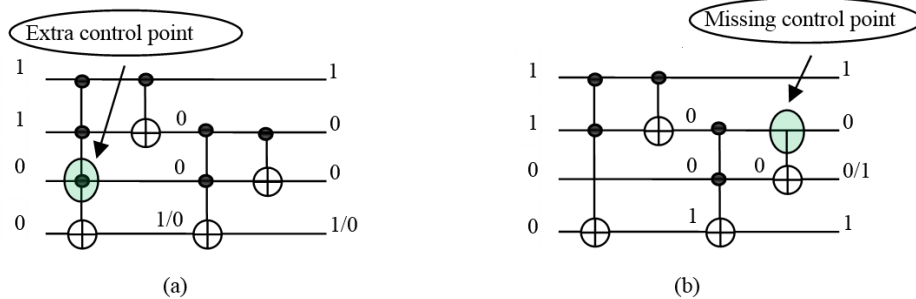

Figure 5. Illustration of fault models (a) Appearance fault and (b) Partial missing gate fault/ Disappearance fault.

Cell fault. This type of fault occurs due to malfunctioning of any gate in a reversible circuit such that it produces an incorrect output (Hayes et al., 2004). These faults may appear in a circuit in any form as in the case of repeated gate faults, where the output becomes faulty due to the repeated operation of a gate or a cell. These faults are based on the fault modeling of cellular logic arrays (Kautz, 1967) and that is why they are called cell faults.

Stuck-at fault. Like the traditional stuck-at fault model, this type of fault occurs in a circuit when a wire gets stuck or is fixed on a single value 0 or 1; these are called the stuck-at 0 or stuck-at 1 faults, respectively (Peres, 1985). The possible nine sites for the occurrence of this type of fault are represented by small circles shown in Figure 6a.
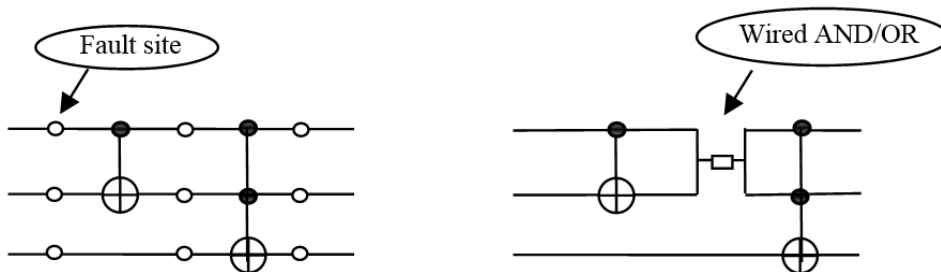

Figure 6. Illustration of fault models (a) Stuck-at fault (b) Bridging fault.

Bridging fault. Again, similar to the traditional fault model, this type of fault takes place when two adjacent wires in a circuit get physically bridged or shorted by means of wired AND/ OR interconnections. As a result, the response comes to an erroneous value (Hurst, 1998), as shown in Figure 6b.

Whenever any type of fault occurs in a circuit, the result is a change of single or multiple values of bits on any wire. This is referred to as a bit fault. When the value of a single line is changed, the fault is called a 'single bit fault' and if the values of two or more lines are changed,

the fault is called a 'multiple bit fault'. Figure 7a and Figure 7b demonstrate the respective faults where the propagation stuck-at-0 fault is shown by the highlighted box.
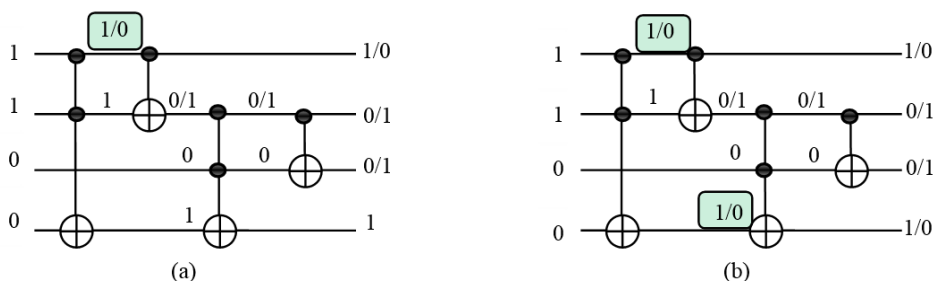

Figure 7. Illustration of fault models (a) Single bit fault and (b) Multiple bit fault.

Hence, the bit fault model is exclusively meant for online testing of reversible logic circuits as the detection of bit faults will detect all other fault models (Gaur et al., 2015). As a result, the complexity of designing for online testability reduces. In the case of offline testability, a separate algorithm is required for each fault.

Performance Metrics

It has been pointed out in the literature that the proposed testable design techniques in reversible logic circuits are justified by means of certain parameters. These parameters illustrate different performance measures for a testable design and on the basis of these measures; quality and performance can be evaluated. These performance measures are defined below:

Gate count. The total number of reversible gates required to realize a reversible circuit is its gate count. It is the straightforward function to find the cost of a reversible circuit, often called the logic cost (Golubitsky & Maslov, 2012). Figure 8a shows the rd32 benchmark circuit (Maslov et al., 2015) as an example where the gate count is 4. Sometimes, a gate count is analyzed on the basis of basic building blocks used to realize a reversible circuit as shown in Figure 8b where three *R* gates (Vasudevan et al., 2004) are required to implement an NOR function.
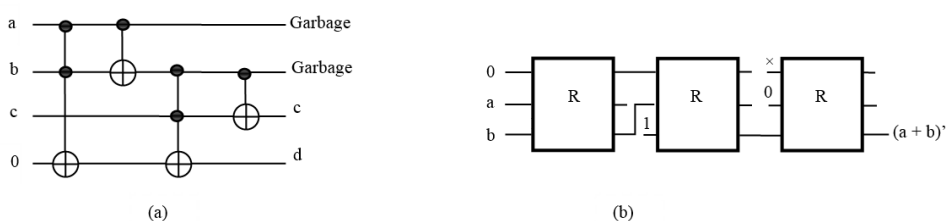

Figure 8. Example circuits (a) rd32 benchmark and (b) Implementation of NOR function using R gate.

Quantum cost. The majority of researchers use quantum cost to evaluate the quality of their design. The quantum cost of a quantum circuit is defined as the sum of elementary quantum gates (1×1, 2×2, Controlled-V and Controlled-V+ quantum gates) used to realize a circuit. In the case of circuits consisting of larger gates, one can substitute them with their equivalent elementary quantum implementation (Barenco et al., 1995; Golubitsky & Maslov, 2012). The quantum cost of some common reversible gates can be found in Table 3; for more information please refer to Maslov and Dueck (2015). The rd32 circuit in Figure 8a can be realized by eight elementary gates; hence, the quantum cost is 8.

Table 3
Quantum Cost of Common Reversible Gates

Name	Size	Quantum Cost
NOT	1×1	1
CNOT	2×2	1
Swap	2×2	3
Toffoli	3×3	5
Fredkin	3×3	3
Peres	3×3	4

Garbage. In order to make a function reversible, some additional output is added to the circuit to maintain equality between the number of outputs and inputs. The values of these additional outputs are not important in the realization of a circuit (Maslov & Dueck, 2015). There are two unused additional outputs in the rd32 benchmark in Figure 8a (marked garbage against them), which are referred to as garbage of garbage output. They are one of the causes of power loss in reversible circuits in the form of information loss, which should be minimized.

Ancilla inputs/Constant inputs. In the designing of a reversible circuit from an irreversible function, certain constant inputs are required whose values are set to constant 0 or 1 (Mohammadi & Eshghi, 2009). In quantum computation these are extra qubits added to the quantum circuit that are set as a constant state that is affected by the quantum operators in order to realize a reversible function. The number of inputs increases the number of qubits in a reversible circuit that increases the area/size of the circuit. These inputs are also referred to as branch cost (Farazmand et al., 2010) as a new branch is added with the addition of an ancilla input.The value of the last line in the rd32 circuit in Figure 8a is set to constant 0 and is referred to as ancilla input. The total number of inputs (sum of variable and constant inputs) in a reversible circuit is also a measure of performance as both are directly related to the area/size of a circuit.

The above-mentioned parameters have a direct relationship with area/size. If these parameters increase, they raise the size of the design and hence, power dissipation increases. Consequently, they should be minimized and should be considered as a matter for concern throughout the development of test strategies. The quantum cost has a proportional relationship with gate count, while ancilla input is an extra source of input power and garbage represents power loss. There are some technology-dependent measures like input to output delay, which gives computation rapidity while the fault coverage shows the quality of design.

Delay can be calculated as the product of depth and number of unit delays of the circuit (Mohammadi & Eshghi, 2009), where depth is the number of stages in a quantum circuit and unit delay is the duration of each stage when all the gates present in the circuit are replaced by their equivalent elementary quantum gates. Fault coverage is the ratio of the number of faults detected to the total number of faults in the circuit (Jha & Gupta, 2003).

DESIGN USING NOVEL TESTABLE GATES

Testing of the reversible circuit came into existence a decade ago and researchers have been curious about the idea of testing in the emerging field of reversible logic. The smart way to achieve online testability is to design new online testable gates, keeping the minimum number of inputs, quantum cost and garbage output. Only a few researchers have proposed this as well. These new gates do have some additional input and output other than purely functional ones to achieve testability during their operation.

Designing with R_1 and R_2 Online Testable Gates

The very first online testable 4×4 reversible gates, R_1 & R_2, in addition to a new reversible gate R, were introduced in the year 2004 (Vasudevan et al., 2004) depicted in Figure 9. The method is meant for the detection of single bit faults inside a reversible circuit. The gate R_1 is used to produce logical functions with a parity output bit, and gate R_2 is used to pass the output of gate R_1 directly to its output in addition to one more output parity bit. Both the reversible gates combine to form a testable block TB, as can be seen in Figure 10. The input p is set to logic 1 and r is set to logic 0 in order to achieve online testability. Test output q and s of TB should resulting complementary values that can be checked using the proposed dual-rail checker circuit composed of eight reversible R gates. Non-complementary output of TB implies a fault within the reversible gates R_1 or R_2.
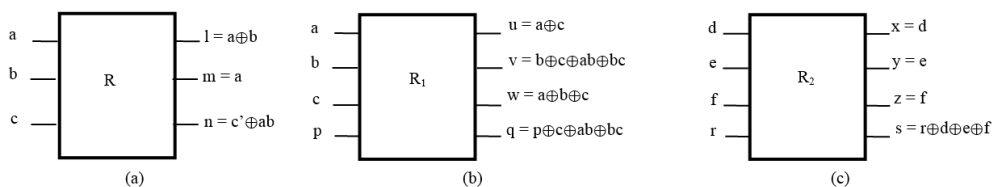

Figure 9. Schematic of proposed gates Gates (Vasudevan, 2004; Vasudevan et al., 2006) (a) R gate, (b) R_1 gate and (c) R_2 gate.
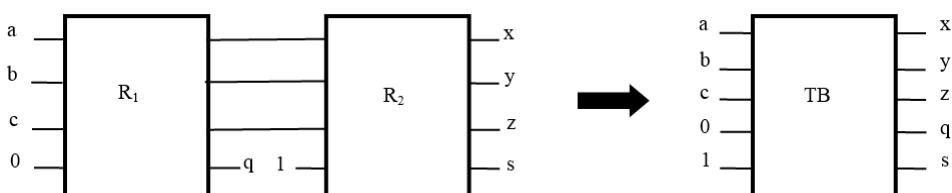

Figure 10. Construction of testable block (TB) (Vasudevan, 2004; Vasudevan et al. 2006)

Experiments were done using a set of benchmarks. Results were analyzed on the basis of the number gates used to synthesize a circuit, which includes a total number of TBs (R_1 and R_2) and R gates (used to implement the required number of rail checker circuits) as gate count (GC) and garbage output (GO) as shown in Table 4. The quantum cost of TB and R is not indicated by the author; it would have been large as their output functions were complex. Also, the required number of checker circuits was equal to the number of TBs used to synthesize a circuit. Hence, the idea is better as far as GO is concerned. The idea was extended further and the authors implemented their proposed scheme in CMOS technology (Vasudevan et al., 2006) in VHDL using Xilinx ISE. The results showed the technology specific measures, power dissipation (PD) in microwatts and total number of transistors (NoT) required for realize a benchmark circuit.

Table 4
Operating Cost of Testable Design in Proposed Method (Vasudevan, 2004; Vasudevan et al., 2006)

Benchmark/Circuit	NoT	PD	GC	GO
xor5	81	69.35	4	8
t	81	54.21	8	16
b1	105	43.12	9	8
majority	129	62.12	11	13
con 1	285	93.42	24	36
clip	309	216.51	26	37
t481	3604	293.21	34	412
rd53	777	311.20	65	90
o64	1545	56.21	129	258
misex 1	1401	675.31	117	161
Full Adder without Propagate	-	59.32	4	3
Full Adder with Propagate	-	-	4	2

Illustration. The test bits q and s of TB shown in Figure 10 can be given as:

$$q = p \oplus c \oplus ab \oplus bc$$

$$s = r \oplus d \oplus e \oplus f = r \oplus u \oplus v \oplus w$$
$$= r \oplus a \oplus c \oplus b \oplus c \oplus ab \oplus bc \oplus a \oplus b \oplus c$$

$$= r \oplus c \oplus ab \oplus bc$$

Let $x = c \oplus ab \oplus bc$

This means that if $p = 0$ and $r = 1$, the value of q and s will be complementary values for correct operations.

Improved Rail-Check Circuit with R_1 and R_2

An efficient rail-check circuit, namely IRC (improved rail check), was proposed (Hasan et al., 2009) in the advancement of the former approach for the detection of a single bit fault. The method proved competent in terms of reduction in quantum cost and garbage output. The design consisted of only a Fredkin gate and a Feynman gate (Chua & Singh, 2014) shown in Figure 11 rather than using R gates, where x_1 and x_2 are test inputs. Test output d produces a high state for a faulty operation and G represents garbage value. Hence, quantum cost decreases as the quantum cost of IRC is only 5, regardless of the number of gates used to synthesize a circuit.
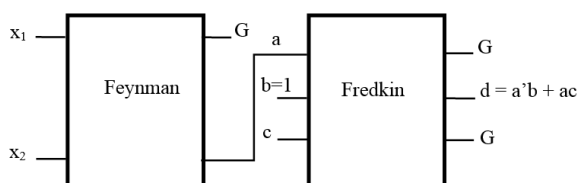

Figure 11. Schematic of IRC (Hasan et al., 2009)

A set of benchmark circuits and some sequential logic blocks like SR, D, JK and T flip flops were realized and performance was evaluated on the basis of the total number of gates, GC, and garbage output, GO. This is depicted in Table 5.

Illustration. In Figure 11, if $x_1 = x_2 = 1$ or $x_1 = x_2 = 0$ then output of the Feynman gate

$a = x_1 \oplus x_2 = 0$

This implies the output of the Fredkin gate:

$d = 1$ for $b = 1$ and $d = 0$ for $b = 0$

Table 5
Operating Cost of Testable Design in Proposed Method (Hasan et al., 2009)

Sequential Circuit	GO	GC	Benchmarks	GO	GC
RS FF	28	24	xor5	20	16
JK FF	36	32	rd53	180	156
D FF	33	28	hwb4	101	92
T FF	41	36	mod5adder	285	248
Master Slave FF	73	64	ham7	214	196
			4mod5	42	36
			rd32	51	44
			5mod5	181	156
			4_49	139	124
			hwb5	329	292
			2or5	172	148
			ham3	39	36
			3_17	63	56

This shows that output d is the decision parameter to check whether the input to the IRC circuit is complementary or not.

Designing with OTG

Thapliyal and Vinod (2007) proposed a 4×4 reversible online testable gate (OTG) shown in Figure 12, which can perform all Boolean functions. The OTG seems better with respect to the earlier proposed R_1 gate in terms of computational complexity. In view of online testability, OTG is used in combination with the 4×4 Feynman gate to form a testable block CTSG for the detection of a single bit fault, as mentioned in Figure 13. The output R and W of CTSG should give complementary logic values for the correct operation; otherwise, there will be a fault in OTG or in the Feynman gate. The value of E is set in accordance to the necessity of any logical function implementation.
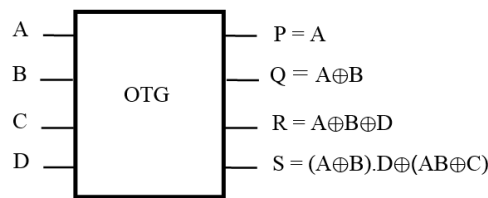
Figure 12. Schematic of OTG (Thapliyal & Vinod, 2007).
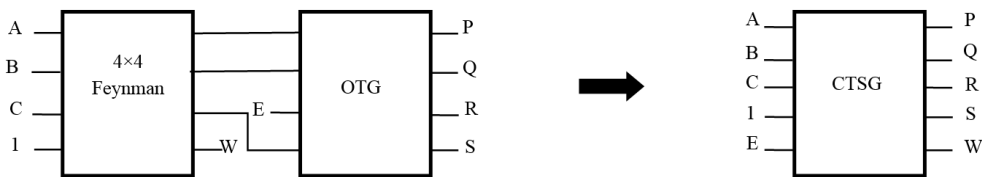
Figure 13. Construction of CTSG (Thapliyal & Vinod, 2007).

A number of testable combinational logic circuits like full adder, ripple carry adder and carry skip adder were synthesized using CTSG. The complementary values for W and R can be verified using the proposed improved two-pair two-rail checker per CTSG. The implementation was justified in terms of number of OTG used as GC, garbage output GO and unit delay (D). This is shown in Table 6. The quantum cost is high as the output functions of OTG are again complex as that of TB regardless of gate count.

Table 6
Operating Cost for Testable Design in Proposed Method (Thapliyal & Vinod, 2007)

Circuit	GO	GC	D
Full Adder with propagate	2	2	2
4-bit ripple carry adder	8	4	4
4-bit carry skip adder	17	22	-

Illustration. Output W and R of CTSG can be given as:

$W = A \oplus B \oplus C \oplus 1 = (A \oplus B \oplus C)'$

$R = (A \oplus B \oplus C)$; $D = C$ (connections shown in Figure 13)

This proves that W and R will hold complementary values that can be detected by the proposed dual rail-checker circuit to attain online testability.

Designing with Dual Rail-Coding-Scheme-Based Reversible Gates

In 2010 Farazmand et al. proposed an idea to reduce design complexity by eliminating the necessity of separate rail checker circuits. They projected two self-testable reversible gates by means of a dual-rail coding scheme for the detection of single bit faults. The schematic representation of the proposed gates as NAND gate and fanout branch is shown in Figure 14. The dual rail NAND gate was used to implement Boolean functions and the dual rail fanout was used to provide branching capability as they did not use reversible logic, which may require synthesizing a circuit.
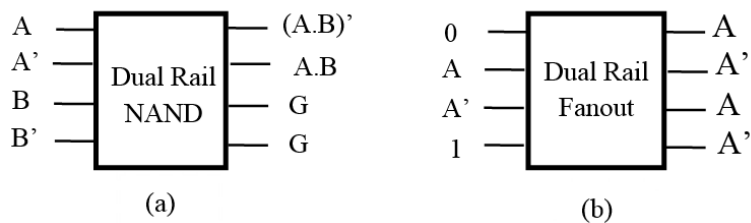
Figure 14. Schematic of dual rail reversible gates Farazmand et al., 2010 (a) as NAND gate and (b) as fanout branch.

Experiments were done on a set of benchmarks where the rail-checker circuit was used only at the final output stage. As a result, there was a large reduction of area in terms of branch cost/ancilla input (AI) and number of inputs (NoI), as seen in Table 7. The method was more efficient in terms of fault coverage and claims 100% coverage of single bit faults.

Table 7
Operating Cost of Testable Design in Proposed Method (Farazmand et al., 2010)

Benchmarks	GO	NoI	AI	Benchmarks	GO	NoI	AI
xor5	112	224	116	rd53	212	424	220
t	48	96	40	misex 1	134	268	118
b1	30	60	24	o64	258	516	0
Majority	18	36	20				
con 1	44	88	48				
clip	1250	2500	800				
t481	9534	19068	958				

Illustration. The dual rail-coding scheme in reversible logic empowers self-testability in the gate itself. The outputs of a reversible circuit implemented using these gates will always be complementary, which can be detected using a dual rail- checker circuit only at the final output stage as illustrated in the schematic of NAND gate realization in Figure 14.

Designing with LG, FIG and DCG

Further, in 2011 Zamani and Tahoori introduced an idea of development of gates in such a manner that they produced the necessary information at the output rather. This was an alternative to the parity generation method. In this approach, a detection line was added to gates whose value was inverted at every gate stage. The error can be detected by knowing the number of stages present in a circuit for the detection of missing gate faults and repeated gate faults. The proposed reversible gates are Logic Gate LG, Fanout/Inversion Gate FIG and Decision Collector Gate DCG as shown in Figure 15. The 4×4 LG gate is used to implement a Boolean function with target inputs A and B and outputs F_1 and F_2 where C is the control input, D is the detection line and G is garbage. The 3×3 FIG gate is used to make the fanout/inversion branch since a fanout is not allowed with reversible circuits. The control input is C set to logic 0/1 such that input A appears as A'/A at outputs F_1 and F_2 of FIG and D is again treated as the detection line. The error can be detected on the basis of the output at detection line D, which is collected by DCG. The value of D will be different for both even and odd numbers of stages in a circuit synthesized using LG and FIG. During any operation, initially all input D values were set to 0, the output value of D (say D_{odd}) was 1 for odd numbers of stages and the output value of D (say D_{even}) was 0 for even numbers of stages in a fault-free circuit. For a circuit consisting of both even and odd numbers of stages, the fault can be analyzed using the 3×3 DCG gate with inputs D_{even} and D_{odd} for even and odd numbers of stages, respectively, where C is the control input. The erroneous output is identified by the *Alarm* output of DCG.
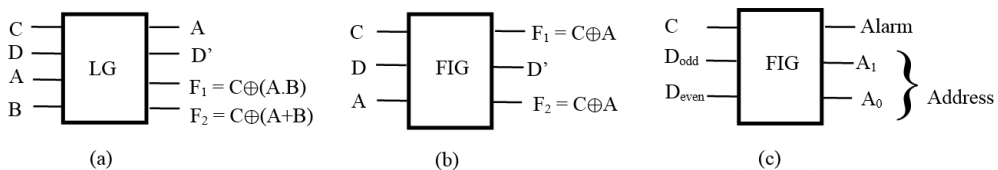
Figure 15. Schematic of proposed gates (Zamani et al., 2011) (a) LG, (b) FIG and (c) DCG.

The benchmark circuit made use of this approach and performance was analyzed on the basis of gate count (LG/FIG/DCG), garbage and delay as seen in Table 8. The method is more suitable in terms of delay, but is faulty because of an increase in design complexity, gate count and garbage output. The method was only meant for the detection of single missing and repeated gate faults.

Table 8

Operating Cost of Testable Design in Proposed Method (Zamani et al., 2011)

Benchmarks	GO	GC	D
Majority	25	17	6
t481	9712	7283	22
b9	815	587	13
alu2	1859	1385	18
term1	3835	2882	17
vda	2653	1948	17
x4	5203	3887	16

Illustration. In the example shown in Figure 16, the upper sub-circuit contains three stages (odd in number) and the lower contains two stages (even in number). For testability, initially the detection line D was set to value 0, hence $D_{odd} = 1$ and $D_{even} = 0$. For non-erroneous output the output on *Alarm* was 0, keeping $C = 1$. For other than these values of D_{odd} and D_{even}, a high signal value appeared at the *Alarm* output of DCG, revealing a fault.
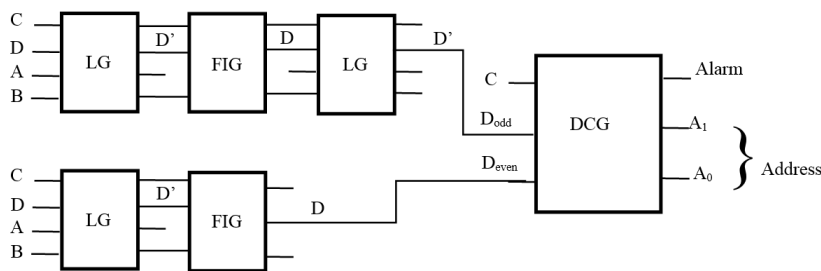

Figure 16. Illustration of method of online testing using LG, FIG and DCG.

Concurrent Error Detection Approach

Reversible functions are bijective functions where the input and output have a one-to-one relationship. Also, the inputs can be regenerated at the outputs if the output signal is re-applied at the inputs of a reversible gate/circuit. The problem of testing can be resolved by the comparison of original inputs and regenerated inputs by comparing these values. This scheme is termed 'concurrent error detection' of a reversible circuit.

Thapliyal and Ranganathan (2004) exploited the idea of concurrent error detection for online testing of reversible logic circuits and established the detection of multi-bit error at the outputs of a reversible circuit in a method called inverse and compare. As a result, the reduction on power dissipation could be achieved by minimizing the requirement of excessive garbage output in design for testability. The technique is illustrated in Figure 17a. The problem of fanout was avoided by the use of intermediate Feynman gate(s). The implementation of the concurrently testable full adder was proposed using OTG (Thapliyal & Vinod, 2007) and its inverse gate IOTG, which can be seen in Figure 17b. The fault can be detected by comparing primary inputs to OTG and outputs of IOTG. This results in zero garbage and a delay of 3

units using only three gates with a quantum cost of 20. The application of the concurrent error detection technique for quantum dot cellular automata QCA was also explained by Thapliyal and Vinod. Their method reduced the design complexity, but the quantum cost and garbage were much higher for bigger circuits as the number of gates used to synthesize a circuit was more than double. The ancilla input would be much higher with the addition of a Feynman gate for dealing with the output. Also, it is very complicated to insert a comparator circuit after each block in an electronic device.
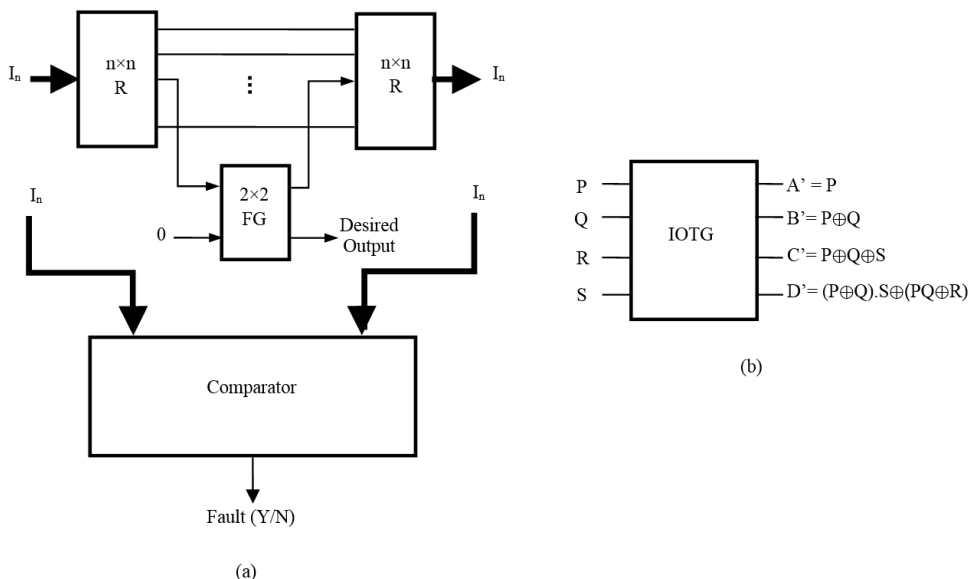
Figure 17. Concurrent error detection (a) Block diagram representation and (b) Schematic of IOTG.

Illustration. It is already proven that reversible gates produce input functions. The output functions can be seen in the equations given below for IOTG as shown in Figure 17b. The output of IOTG can be given as:

$$A' = P = A$$

$$B' = P \oplus Q = A \oplus (A \oplus B) = B$$

Similarly, $C' = C$ and $D' = D$

Hence, one can detect faults in a reversible circuit by comparing primary inputs and regenerated inputs as shown in the block diagram representation in Figure 17a.

Attributes of Designing with Novel Gates Methods

Extracted attributes of all the testing methods explained in this section are depicted in Table 9. These attributes appropriately compare all these methods. Comparison tables are also given. By comparison, it is found that the method by Vasudevan et. al., 2006 and Hasan et. al., 2009

proved better in performance in terms of minimum garbage output. The later approach has lower quantum cost and garbage than the first method.

Table 9
Attributes of Designing with Novel Gates Methods

Testing Method	Features	Tables for comparison
(Vasudevan et al., 2004 and Vasudevan et al., 2006)	Large GC (i.e. QC high), GO is comparatively small.	4,5,7 and 8
(M.Hasan et al., 2009)	QC and GO both are high	4,5,7 and 8
(Thapliyal et al., 2007)	QC is high and GO is comparatively small	4,6
(Farzamand et al., 2010)	QC , GO and NoI are high	4,5,7 and 8
(Zamani et al., 2011)	QC and GO both are high	4,5,7 and 8
(Thapliyal et al., 2004)	GC, QC, AI and GO all are high	-

DESIGNING WITH EXISTING CIRCUIT

Another way of achieving online testability in reversible logic circuits is to design with any existing standard circuits. In this approach, the original circuit is modified by adding a number of reversible gates and wires with constant inputs in order to attain online testability.

Designing with Derived Reversible Gates

In this regard Mahammad and Veezhinathan (2010) proposed a two-step procedure of conversion of a reversible gate in its testable form to detect single bit faults. A reversible gate URG was also proposed that used four basic logic functions efficiently in terms of quantum cost. In the first step a reversible gate R was converted into a deduced reversible gate DR_a such that for a $n \times n$ reversible gate with input vector I_{ia} and output vector O_{ia} was changed into $(n+1) \times (n+1)$ reversible gate with input vector I_{ia+1} and output vector O_{ia+1} without changing the original functionality of the circuit as depicted in Figure 18a.
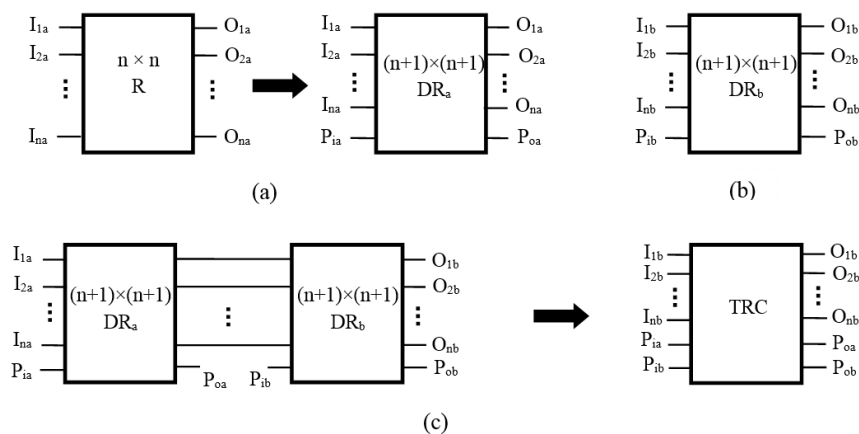

Figure 18. Basic fundamental used in proposed scheme (Mahammad et al., 2010) (a) Conversion of DR_a from reversible gate R, (b) Schematic of identity gate DR_b and (c) Construction of TRC.

If, $I_{ia} = (I_{1a}, I_{2a}...I_{na})$ and $O_{ia} = (O_{1a}, O_{2a}...O_{na})$, then $I_{ia+1} = (I_{1a}, I_{2a}... I_{na}, P_{ia})$ and $O_{ia+1} = (O_{1a}, O_{2a}...O_{na}, P_{oa})$ where P_{ia} and P_{oa} are the input and output parity bit corresponding to the first step, respectively and $P_{oa} = [O_{1a} \oplus O_{2a} \oplus ...O_{na} \oplus P_{ia}]$.

In the second step DR_a was cascaded with $(n+1) \times (n+1)$ identity reversible gate DR_b as shown in Figure 18b to form $(n+2) \times (n+2)$ testable reversible cell TRC as can be seen in Figure 18c. The unity reversible gate passes the output of DR_a directly to its output except parity P_{oa} and generates another parity bit P_{ob} with respect to the input parity bit P_{ib} such that for a $(n+1) \times (n+1)$ identity gate, the input vector and output vector are given by $I_{ib+1} = (I_{1b}, I_{2b}... I_{nb}, P_{ib})$ and $O_{ib+1} = (O_{1b}, O_{2b}...O_{nb}, P_{ob})$, respectively, where $(O_{1a}, O_{2a}...O_{na}) = (I_{1b}, I_{2b}... I_{nb}) = (O_{1b}, O_{2b}...O_{nb})$ and $P_{ob} = [O_{1b} \oplus O_{2b} \oplus ... O_{nb} \oplus P_{ib}]$.

The detection of faults can be done by assigning logic values to the input parity bits and checking the corresponding output parity bit. If P_{ia} and P_{ib} are assigned the same logic values, then P_{oa} and P_{ob} should be same or if P_{ia} and P_{ib} are assigned complementary logic values, then P_{oa} and P_{ob} should be complementary to prevent an erroneous result. Moreover, a testable cell TC is also proposed to detect faults consisting of multiple TRCs. The illustration below is based on implementing a testable Fredkin based 2×4 decoder. The results are given in terms of CMOS characteristics $2.249 \times 10^{-9} \mu W$ dynamic power using 1496 numbers of transistors with a delay of 13.2ns. The method is efficient in terms of quantum cost, but the number of ancilla inputs increases to twice the number of TRC used to synthesize a circuit.

Illustration. Using Figure 18c, if $P_{oa} = 0$ and $P_{ob} = 1$, then $P_{oa} = [O_{1a} \oplus O_{2a} \oplus ...O_{na} \oplus 0] = F_{oa} \oplus 0 = F_{oa}$ and $P_{ob} = [O_{1b} \oplus O_{2b} \oplus ...O_{nb} \oplus 1] = F_{ob} \oplus 1 = \sim F_{oa}$. Since $F_{ob} = F_{oa}$, this implies $Poa = \sim P_{ob}$. Hence, for a complementary input parity bit, output will always be complementary for correct operation of any circuit that can be checked using a dual-rail-checker circuit or proposed testable circuit called TC.

Design Using ETG for ESOP-Based Toffoli Circuits

Another idea for a testable design with existing circuit was proposed by Nayeem and Rice in 2011 for the detection of single and multiple bit faults. The idea is valid for the exclusive sum-of-product (ESOP)-based Toffoli circuits, which ensure a negligible increment in garbage on the cost of small increase quantum cost. The process of conversion of an ESOP circuit in its testable form unfolds in three steps. In the first step every n-bit Toffoli gate in the circuit is converted into $(n+1)$-bit ETG (Extended Toffoli Gate) (Chen et al., 2008) using an additional parity line without disturbing the original circuit. In the second step, CNOT gates are included from all the output lines to the parity line. In the third step, CNOT gates are again added from each input line to the parity line ahead and behind the complete circuit. This requires the addition of twice the number of CNOT gates as the number of input lines plus the number of outputs, which enhance quantum cost, but garbage output remains the same as in the original circuit. The steps of conversion of a given circuit in its testable form are shown in Figure 19. The fault is detected if the value of the output parity bit changes from 0 to 1 during its operation. This method is also valid for the ESOP circuit consisting of the inverted-Toffoli gate. The performance of this method is predicted in terms of quantum cost (QC) and garbage output (GO) by performing experiments on some benchmark circuits, as can be seen in Table 10.
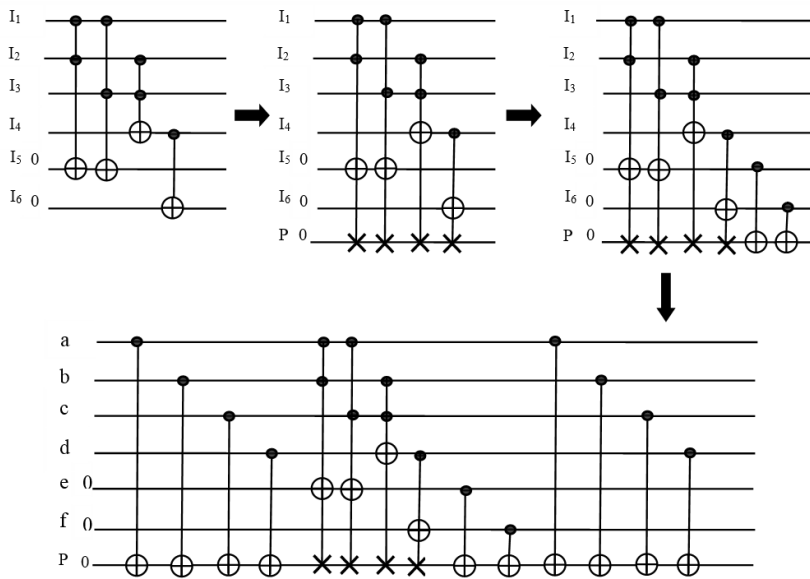
Figure 19. Conversion of ESOP-based circuit in its testable form (Nayeem & Rice, 2011).

Illustration. Mathematically, the output of the parity line P of build testable circuit shown in Figure 19 can be given as:

$$P \oplus a \oplus b \oplus c \oplus d \oplus ab \oplus ac \oplus bc \oplus (d \oplus bc) \oplus (e \oplus ab \oplus ac) \oplus (f \oplus d) \oplus a \oplus b \oplus c \oplus d$$

Since values of P, e and f are set logic 0 then,

$$a \oplus b \oplus c \oplus d \oplus ab \oplus ac \oplus bc \oplus (d \oplus bc) \oplus (ab \oplus ac) \oplus d \oplus a \oplus b \oplus c \oplus d$$

The solution gives the value of output of wire P as logic 0 if there are bit faults in the circuit or at the input and output of the circuit. If the value of output P changes to logic 1, a fault can be detected.

Design Using ETG for Toffoli-Based Circuits

The approach in previous section was extended by Nayeem and Rice (Vancouver, 2011) for any type of Toffoli circuit for the detection of single and multiple bit faults. An online testable circuit was constructed from a Toffoli circuit in similar manner except for the second step where, first a Toffoli gate was converted into an ETG connected to an additional parity line and then CNOT gates were inserted from all lines to parity lines before and after the original circuit. This required only double the number of lines as the number of additional CNOT gates compared to the previous method, as shown in Figure 20. The fault was also detected in the same manner by checking the logic value of output of parity line. Further, this technique was again extended for ternary Toffoli circuits (Nayeem & Rice, 2012). The detailed study was specified in Nayeem and Rice (2013), where a large number of benchmarks were used and a comparative study of proposed methods was given. Figures of merit are depicted in Table 10

in comparison to their previous approach.

The two methods in Nayeem and Rice (Canada, 2011; Vancouver, 2011) were efficient in terms of garbage output GO and ancilla input AI, but the quantum cost increased with the addition of CNOT gates. GO remained the same as that of the original circuit and only one AI was required to convert a circuit to its testable form. The basic difference between the respective methods was that the first was meant for ESOP-based circuits and the other was valid for all Toffoli circuits.
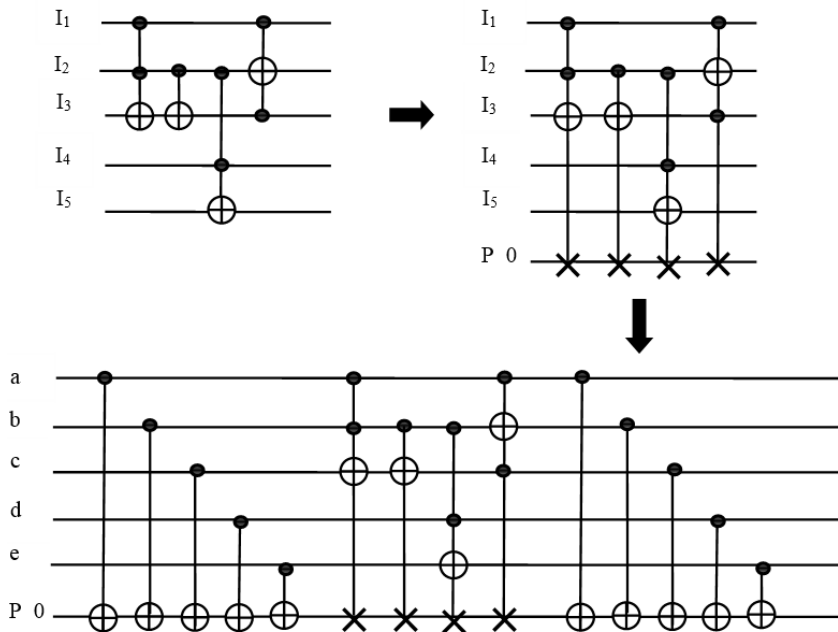
Figure 20. Conversion of a Toffoli-based circuit in its testable form Nayeem and Rice (Vancouver, 2011).

Table 10
Comparison and Operating Cost for Testable Design in Proposed Methods

Benchmarks	Nayeem et al. (Canada 2011; 2013)		Nayeem et al. (Vancouver 2011; 2013)	
	QC	GO	QC	GO
majority	154	5	154	5
con 1	184	7	184	7
misex 1	1040	8	439	8
ham7	259	7	162	7
rd32	49	3	45	3
alu2	4882	10	4551	10
9symml	11065	9	11066	9
alu4	48951	14	42071	14
nth prime3	607	7	575	7

Illustration. Mathematically, the output of the parity line *P* of the constructed testable circuit as shown in Figure 20 can be given as:

$$P \oplus a \oplus b \oplus c \oplus d \oplus e \oplus ab \oplus b \oplus bd \oplus a\,(c \oplus ab \oplus b) \oplus a \oplus [b \oplus a\,(c \oplus ab \oplus b)] \oplus (c \oplus ab \oplus b) \oplus d \oplus (bd \oplus e)$$

The solution gives a value of output of wire *P* as logic 0 when applying input logic 0 in the same manner if a bit fault had occurred in the circuit or at the input and output of the circuit. The value of output *P* changed to logic 1 from 0 and hence, a fault was detected.

Design Using Testable Toffoli and Peres

In this concern, Sen et al. (2012) also proposed a DFT methodology in which they demonstrated the idea of converting a reversible gate into its online testable form, ensuring detection of nearly all multiple bit faults. The authors introduced a DFT process algorithm based on parity generation using concurrent error detection. The algorithm renovated a reversible gate into its online testable form, and this resulted in three error detecting bits that were used to test a circuit. The testable 3×3 Toffoli gate was synthesized with input *A*, *B* and *C* and output *X*, *Y* and *Z* as shown in Figure 21. The fault can be noticed by error detecting bits P, Q and R. These bits should hold logic 0 values for a fault-free circuit operation. Similarly, a testable Peres gate was also proposed and the implementation of an online testable full adder was given for justification.

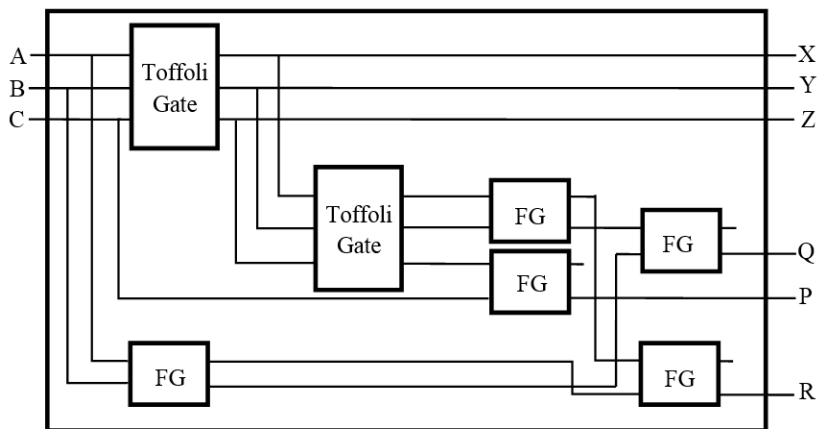*Figure 21.* Online testable 3×3 Toffoli gate (Sen et al., 2012).

The performance of the proposed DFT was evaluated by performing experiments on the existing reversible gates, three benchmark circuits, full adder and full subtractor circuit, as shown in Table 11. The testable Toffoli gate had a quantum cost of 11 and a garbage output of 3. Hence, the quantum cost was much higher regardless of the lower gate count.

Table 11

Operating Cost of Testable Design in Proposed Method (Sen et al., 2012).

Benchmark circuit/Gate	GC
Peres gate	7
Toffoli gate	7
Fredkin gate	11
Full Adder with propagate	14
Full subtractor	21
ham3	21
3_17	49
Nth prime3	28

Illustration. Error detecting bits for a testable Toffoli gate as shown in Figure 21 can be calculated as:

$$P = (A \oplus B) \oplus (A \oplus B) = 0$$

$$Q = C \oplus C = 0$$

$$R = A \oplus A = 0$$

If single/multiple bit faults occur in any gate or at the wire, any values of error detecting bits flip to logic 1 and the fault can be detected.

Extracted Attributes of Designing with Existing Circuit

Extracted attributes of all the testing methods explained in this section are depicted in Table 12. These attributes appropriately compare all these methods. Comparison tables are also provided. It is found that the method suggested by Nayeem & Rice, Canada, 2011 and Nayeem & Rice, Vancouver, 2011 was better in performance in terms of minimum garbage output. Also, the first method was more efficient in terms of quantum cost and garbage output.

Table 12

Attributes of Designing with Existing Circuit Methods

Testing Method	Features	Demonstration table number
Mahammad et al., 2010	AI and GO are large due to TC, QC is comparatively small	-
Nayeem et al., Canada, 2011	QC large but minimum GO	10 and 11
Nayeem) et al., Vancouver, 2011	QC large but minimum GO	10 and 11
Sen et al., 2012	QC, NoI and GO are large per testable gate	10 and 11

DISCUSSION AND CONCLUSION

There are various methods in two broad classifications of designing with proposed gates and designing with existing circuits. These methods are claimed to cover all the faults in the new family of fault models. Detection is achieved by the detection of single and multiple bit faults. These methods are illustrated by performing experiments on benchmark circuits and evaluating results in terms of performance parameters. Final attribute tables in two respective sections suggest the pros and cons of each method, which can be used to determine the suitability for designing under online testability. The evaluation is summarized in Table 13 with testability technique, considered fault models and metrics that can well define their identity.

Table 13

Comparison Table of Discussed Online Testing Approaches

S No.	Author	Online Testability Approach	Fault model	Computed parameters
1	Vasudevan et al. (2004)	Propose testable Reversible Gates R1 & R2 and Parity Generation method for testing is used	Single Bit Fault	GC, GO
2	Vasudevan et al. (2006)	CMOS realization with previous testable technique is provided	Single Bit Fault	GC, GO
3	Thapliyal et al. (2007)	Parity Generation method for testing is used Proposed reversible gate OTG with Feynman gate	Single Bit Fault	GC, GO, D
4	Chowdhury et al. (2009)	Improved Rail-check circuit is proposed and R1 and R2 are used for parity generation	Single Bit Fault	GC, GO
5	Mahammad et al. (2010)	Reversible gate URG and testable design with existing circuit using parity generation	Single Bit Fault	-
6	Tahoori et al. (2010)	Reversible Gates with dual rail I/O	Single Bit Fault	NoI, GO, AI
7	Thapliyal et. al. (2010)	Concurrent error detection using OTG and majority voter scheme in QCA	Single/Multiple Bit Fault	GC, GO, QC, D
8	Nayeem et. al. (Canada, 2011)	Testable design with existing circuit for ESOP-based circuit	Single/Multiple Bit Fault	QC, GO
9	Nayeem et. al. (Vancouver, 2011)	Testable design with existing circuit for Toffoli-based reversible circuit	Single/Multiple Bit Fault	QC, GO
10	Tahoori et. al. (2011)	Reversible gates, namely FIG, DCG and LG for testability	MGF and RGF	GC, GO, D
11	Sen et. al. (2012)	Concurrent error detection technique based on parity	Single/Multiple Bit Fault	GC
12	Nayeem et. al. (2013)	Testable design with existing circuit for ESOP and non-ESOP-based Toffoli circuits	Single/Multiple Bit Fault	QC, GO

A comparison of implemented online testable combination circuits can be seen in Table 14 with performance measures. Table 4 presents CMOS-technology dependent measures like number of transistors and power dissipation. The online testable sequential circuits with performance measures can be seen in Table 5. Quantum cost is not reported by most researchers, but gate count is proportionally related to it. If the gate count is low, the quantum cost can be high; this depends on the complexity of the output function of the respective gates.

Table 14

Comparison Table for Online Testable Combinational Logic Circuits

Circuit	Vasudevan et al., 2004 & Vasudevan et al., 2006			Thapliyal et al., 2007			Thapliyal et al., 2010			Sen et al., 2012		
	QC	GO	GC	QC	GO	GC	QC	GO	GC	QC	GO	GC
Full adder without propagate	–	3	4	–	–	–	–	–	–	–	–	–
Full Adder with propagate	–	2	4	–	2	2	30	3	8	–	–	14
4-bit ripple carry adder	–	–	–	–	8	4	–	–	–	–	–	–
4-bit carry skip adder	–	–	–	–	17	22	–	–	–	–	–	–
Full subtractor	–	–	–	–	–	–	–	–	–	–	–	21

Table 15 summarizes all the covered benchmarking circuits (Maslov et al., 2015; Reversible benchmarks, 2015) with online testability along with the comparison of all discussed techniques on the basis of performance parameters (QC, GO, GC and AI). The comparison clearly points out differences in area/size and consequently, the increment and decrement of power dissipation in designing with testability for reversible logic. The method with the best possible results among the depicted two categories is shown in the highlighted blocks.

Table 15

Comparison Table for all Discussed Approaches

Benchmarks	Vasudevan et al.				Hasan et al.				Farazmand et al.				Nayeem et al. Can.2011				Nayeem et al. Van 2011				Zamani et al.				Sen et al.			
	QC	GO	NoG	AI	QC	GO	NoG	AI	QC	GO	NoG	AI	QC	GO	NoG	AI	QC	GO	NoG	AI	QC	GO	NoG	BrC	QC	GO	NoG	AI
xor5	–	8	81	–	–	20	16	–	–	112	–	116	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
t	–	16	81	–	–	–	–	–	–	48	–	40	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
b1	–	8	105	–	–	–	–	–	–	30	–	24	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Majority	–	13	129	–	–	–	–	–	–	18	–	20	154	5	–	–	154	5	–	–	–	25	17	–	–	–	–	–
con 1	–	36	285	–	–	–	–	–	–	44	–	48	184	7	–	–	184	7	–	–	–	–	–	–	–	–	–	–
clip	–	37	309	–	–	–	–	–	–	1250	–	800	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
t481	–	412	3604	–	–	–	–	–	–	9534	–	958	–	–	–	–	–	–	–	–	–	9712	7283	–	–	–	–	–
rd53	–	90	777	–	–	180	156	–	–	212	–	220	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
misex 1	–	161	1401	–	–	–	–	–	–	134	–	118	1040	8	–	–	439	8	–	–	–	–	–	–	–	–	–	–
o64	–	258	1545	–	–	–	–	–	–	258	–	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
ham7	–	–	–	–	–	214	196	–	–	–	–	–	259	7	–	–	162	7	–	–	–	–	–	–	–	–	–	–
rd32	–	–	–	–	–	51	44	–	–	–	–	–	49	3	–	–	45	3	–	–	–	–	–	–	–	–	–	–
ham3	–	–	–	–	–	39	36	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	21	–
3_17	–	–	–	–	–	63	56	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	49	–
b9	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	815	587	–	–	–	–	–
alu2	–	–	–	–	–	–	–	–	–	–	–	–	4882	10	–	–	4551	10	–	–	–	1859	1385	–	–	–	–	–
nth prime3	–	–	–	–	–	–	–	–	–	–	–	–	607	7	–	–	575	7	–	–	–	–	–	–	–	–	28	–

It is observed that the bijective property in reversible logic is maximally utilised for online testability. One can easily generate or preserve information of input and output vectors in the form of parity. In reversible circuits TB (Vasudevan et al., 2004; Vasudevan et al. 2006) and CTSG (Thapliyal & Vinod, 2007), a fault was detected by generating two output parity bits. Sometimes, a reversible gate (Mahammad & Veezhinathan, 2010) or a circuit (Nayeem & Rice, Canada, 2011; Vancouver, 2011; 2013) was modified in order to generate parity bits by adding an extra wire and CNOT gates. Also, it was used along with a concurrent error detection technique (Sen et al., 2012) to generate error-detecting information. Moreover, the method of concurrent error detection in collaboration with parity generation was used in Thapliyal and Ranganathan (2004) and Sen et al. (2012). Dual-rail-coding (Farazmand et al., 2010) methods also utilised this fact for testability in reversible circuits. It was also observed that quantum cost was slightly improved, but the optimization of ancilla input and garbage output still required more effort by the researchers.

We have presented a comparative study of online testability in reversible logic circuits along with a brief illustration of each method by combining all possible fault models and performance metrics. The major challenges pertaining to online testability are presented. The paper also projects the role of garbage output and ancilla inputs in increasing power dissipation. Furthermore, the reliability of the method of parity preservation and generation, concurrent error detection and dual-rail coding techniques in designing for testability is discussed. Despite significant progress in online testability of reversible logic, new techniques will always arise to replace existing ones. This paper points out the considerable issues in this area and encourages the development of new strategies to face a new era of reversible logic because several challenges remain.

REFERENCES

Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., Weinfurter, H. (1995). Elementary gates for quantum computation. *Journal Physical Review A.52*(5), 3457-3467.

Bennett, C. H. (1973). Logical reversibility of computation. *Journal of Research and Development - IBM, 17*(6), 525-532.

Chen, J., Zhang, X., Wang, L., Wei, X., & Zhao, W. (2008). Extended Toffoli gate implementation with photons.*Proceedings of 9th International Conference on Solid-State and Integrated-Circuit Technology* (p.575-578).

Chua, S. C., & Singh, A. K. (2014). Heuristic synthesis of reversible logic – A comparative study. *Advances in Electrical and Electronics Engineering, 12*(3), 210-225.

Farazmand N., Zamani M., & Tahoori M. B. (2010).Online fault testing of reversible logic using dual rail coding. *Proceedings of 16th IEEE International Symposium on On-Line Testing (IOLTS)*(pp. 204-205).

Feynman, R. P. (1985).Quantum mechanical computers. *Optics News, 11*, 11-20.

Fredkin, E., & Toffoli, T. (1982).Conservative logic. *International of Journal Theoritical Physics*, *21*(3-4), 219-253.

Gaur, H.M., Singh, A. K., & Ghanekar, U. (2015). A review on online testability for reversible logic. *Procedia Computer Science, 70,* 384-391.

Golubitsky, O., & Maslov, D. A.(2012). Study of optimal 4-bit reversible Toffoli circuits and their synthesis. *IEEE Transactions on Computers, 61*(9), 1341-1353.

Hasan, M., Tauhidul, Islam, A. K. M., & Chowdhury, A. R. (2009).Design and analysis of online testability of reversible sequential circuits. *Proceedings of 12th IEEE International Conference on Computer and Information Technology* (pp. 180-185).

Hayes, J. P., Polian, I., & Becker, B. (2004).Testing for missing gate faults in reversible circuits. *Proceedings of 13th IEEE Asian Test Symposium (ATS)*(pp. 100-105).

Hurst, S. L. (1998). *VLSI testing – Digital and mixed analogue/digital techniques*. London, United Kingdom: The Institution of Electrical Engineers.

Jha N. K., & Gupta S. (2003). *Testing of digital systems*. Cambridge University Press.

Kautz, W. H. (1967). Testing for faults in combinational cellular logic arrays. *Proceedings IEEE Conference on Record of the Eighth Annual Symposium on Switching and Automata Theory* (pp. 161-174).

Landauer, R. (1961). Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, *5*(3), 183-219.

Mahammad, S. K. N., & Veezhinathan, K. (2010).Constructing online testable circuits usingreversible logic. *IEEE Transaction on Instrumentation & Meas*urement, *59*(1), 101-109.

Maslov, D., Dueck, G., & Scott, N. (2015, October). Reversible logic synthesis benchmark. Retrieved from http://www.cs.uvic.ca/~dmaslov.

Maslov, D., & Dueck, G. W. (2015, October). Garbage in reversible designs of multiple output functions. Available at http://cs.unb.ca/profs/gdueck/reversible/multiple.pdf.

Mohammadi M., & Eshghi M. (2009). On figures of merit in reversible logic design. *Springer Journal of Quantum Information Processing, 8*(4), 297-318.

Nayeem, N. M., & Rice, J. E. (2012). A new approach to online testing of TGFSOP-based Ternary Toffoli circuits. *42nd IEEE International Symposium on Multi-Valued Logic* (pp. 315-321).

Nayeem, N. M., & Rice, J. E. (2011). A shared-cube approach to ESOP-based synthesis of reversible logic. *Facta. Univ. Series. Electron Energetics, 24*(3), 385-402.

Nayeem N. M.,&Rice, J. E. (2011).A simple approach for designing testable reversible circuits. *Proceedings of IEEE Pacific Rim. Conference Comm. and Signal Processing (PACRIM)*(pp. 85-90). Victoria, Canada.

Nayeem, N. M., & Rice, J. E. (2011). Online fault detection in reversible logic.*Proceedings of IEEE SymposiumDefect and Fault Tolerance in VLSI and Nanotechnology system (DFT)*(pp. 426-434). Vancouver.

Nayeem, N. M., & Rice, J. E. (2013).Online testable approaches in reversible logic. *Journal of Electronic Testing, 29*(6), 763-778.

Patel, K. N., Hays, J. P., & Markov, I. L. (2004). Fault testing for reversible logic. *IEEE Transacton on CAD of Integrated Circuits and Systems, 23*(8), 1220-1230.

Peres, A. (1985).Reversible logic and quantum computers. *Physical Review A, 32*(6), 3266-3276.

Polian, I., Hayes, J. P., Fiehn, T., & Becker, B. (2005).A family of logical fault models for reversible circuits. *Proceedings of the 14th Asian Test Symposium (ATS)* (pp.422-427). Calcutta.

Rahaman, H., Kole, D., Das, D., & Bhattacharya, B. B. (2007).Optimum test set for bridging fault detection in reversible circuits. *Proceedings of 16th Asian Test Symposium* (p125-128).

Rahaman, H., Kole, D. K., Das, D. K., & Bhattacharya, B. B. (2011). Fault diagnosis in reversible circuits under missing-gate fault model. *Computers and Electrical Engineering, 37*(4), 475-485.

Sen, B., Das, J., & Sikdar, B. K. (2012). A DFT methodology targeting online testing of reversible circuit. *IEEE International Conference of Devices, Circuits and Systems,* pp.689-693.

Shende, V. V., Prasad, A. K., Markov, I. L., & Hayes, J. P. (2003). Synthesis of reversible logic circuits. *IEEE Transacton on CAD of Integrated Circuits and Systems 22*(6), 710-722.

Tabei, K., & Yamada, T. (2005). On generating test sets for reversible circuits. *Proceedings of 14th IEEE Asian Test Symposium* (p.422-427).

Thapliyal, H., & Vinod, A. P. (2007).Designing efficient online testable reversible adders with new reversible gate. *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)* (pp.1085-1088). New Orleans.

Thapliyal, H., & Ranganathan, N. (2004). Reversible logic based concurrent error detection methodology for emerging nanocircuit. *Proceedings of 10th IEEE International Conferencence on Nano Technology* (pp.217-222).

Toffoli, T. (1980). Reversible computing. *Proceedings of Automata, Lang. Program Conference* (pp. 632-644).

Vasudevan, D. P., Lala, P. K., Jia, D., & Parkerson, J. P. (2006). Reversible logic design with online testability. *IEEE Transaction on Instrumentation & Measurement, 55*(2), 406-414.

Vasudevan, D. P., Lala, P. K., & Parkerson, J. P. (2004).Online testable reversible logic circuit design using NAND blocks. *Proceedings of 19th IEEE International Symposium on Defect Fault Tolerance on VLSI Systems.*(pp. 324-331).

Zamani, M., & Tahoori, M. B. (2011). Online missing/repeated gate faults detection in reversible circuits. *Proceedings of IEEE Int. Symp. Defect and Fault Tolerance in VLSI and Nanotechnology Systems* (pp. 435-442).

Zhong, J., & Muzio, J. C. (2006) Analyzing fault models for reversible logic circuits. *Proceedings of IEEE Congress on Evolutionary Computation (CEC)* (pp. 2422-2427).