ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design of Online Testing Approach for Fault Models in Reversible Circuits using Verilog HDL

Modala Kavitha Rani
PG Scholar,
Dept of ECE,
Vignan Institute of Technology and Science,
Hyderabad, TS, India.

Vijaykumar R.Urkude
Associate Professor,
Dept of ECE,
Vignan Institute of Technology and Science,
Hyderabad, TS, India.

Abstract:

In order to continue the revolution in the computer hardware performance, we need to reduce the energy dissipated in each logic operation. Energy dissipation can be reduced by preventing information loss. This is achieved by designing the circuits using reversible logic gates. It has wider applications in the fields of quantum computing, nanotechnology, and many more. Testing such circuits is obviously an important issue. In this paper we propose an approach for the design of online testable reversible circuits. A reversible circuit composed of Toffoli gates can be made online testable by adding two sets of CNOT gates and a single parity line. The performance of the proposed approach for detecting a family of missing gate faults: 1.Single Missing Gate Fault (SMGF), 2.Partial Missing Gate Fault (PMGF), 3.Repeated Gate Fault (RGF), and Crosspoint fault has been observed. Discussion around the correctness of our approach and the overhead is also provided. All the designs are functionally verified using Xilinx ISE14.4 tool.

Keywords:

Revesible Logic, Online Testing, Fault Models, Performance measures.

I. INTRODUCTION:

In recent years reversible computation has established itself as a promising research area and emerging technology. This is motivated by a widely supported prediction that the conventional computer hardware technologies are going to reach their limits in the near future [2].

A fundamental limitation of conventional computing is that each time information is lost energy is dissipated regardless of the underlying technology. This is known as Landauer’s principle [5]. It was also shown by Bennett [1] that theoretical zero power dissipation can only be achieved if the circuit is logically reversible [1]. Reversible computing is bijective in nature, and by definition reversible circuits are theoretically information-lossless. Thus using reversible computation, the power dissipation which results according to Landauer’s principle can be decreased or even eliminated. In this paper we address the area of testing for reversible circuits, and propose an online testing approach to detect faults in reversible circuits.

II. BACKGROUND:

A reversible logic circuit is an acyclic combinational logic circuit in which all gates are reversible and are interconnected without fan-out. Moreover, feedback lines from the output to input are not allowed in reversible circuits [2]. In this paper we consider three types of reversible gates: NOT, CNOT (CNOT stands for Controlled NOT) and Toffoli gates. These three gates form the CNT (CNOT, NOT, Toffoli) gate library. Generally, we refer to the 0-CNOT gate as a NOT gate, to the 1-CNOT gate as Feynman gate and to the 2-CNOT gate as a Toffoli gate. A NOT gate (0-CNOT) has no control line and hence the input at the target line is always inverted at the output line.

1. Controlled-NOT Gate/ Feynman Gate:

It is a 2*2 reversible logic gate. CNOT Gate, also known as Feynman Gate and is used to overcome the fan-out problem since it can be used for copying the information. CNOT gate has unit quantum cost and unit delay.

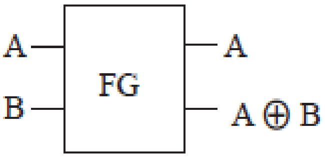


Fig.2. CNOT or Feynman Gate

2. Toffoli Gate:

Toffoli gate is a 3*3 reversible gate with quantum cost of 5 and delay of 5. It is called also universal reversible gate.

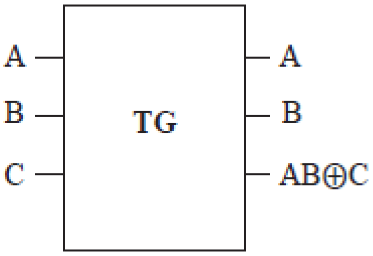


Fig.3. Toffoli gate

Types of Testing:

Testing is required to ensure quality, availability, and reliability of a circuit or device. There are two types of testing: offline testing and online testing [11]. In offline testing a circuit under test is taken out of its normal mode of operation. In contrast, online testing is carried out while the circuit is being used for normal operations. In this case additional circuitry is attached to the original circuit to determine whether the system is faulty or fault free. In this paper we focus on the latter approach. Our approach has been proposed to detect missing gate faults[10], repeated gate faults and crosspoint faults [14] .The crosspoint fault model focuses on faults that may occur on the control points of a reversible gate. When one or more control points are added erroneously to a gate then this is called an appearance crosspoint fault. A disappearance fault occurs when one or more control points of a gate do not work or disappear from a circuit.

The missing gate fault model is a package of three different fault models, including (a) the single missing gate fault (SMGF): a fault that is modeled by the disappearance of an entire gate; (b) the partial missing gate fault (PMGF): some of the control points of a gate are missing; (c) the repeated gate fault (RGF): an unwanted replacement of a gate by the several instances of the same gate. A PMGF turns a k-CNOT gate into a k -CNOT gate, where k referred to as the order of a PMGF.

III. LITERATURE SURVEY:

In [12] the authors proposed three new reversible gates. Two of the three gates are used to design an online testable block and the other gate is used to create a checker circuit. The purpose of the checker circuit is to compare the two parity bits produced by the online testable block, which will then detect a single bit fault. Similar to this approach, the authors in [6] proposed an improved approach for detecting a single bit fault. This design does not require an extra checker circuit to compare the parity bits. However both of these approaches have a common drawback. If a single bit fault occurs between cascaded blocks then the fault will go unnoticed Zhong et al. proposed both the cross point fault model and as well a testing approach to detect single appearance and disappearance cross point faults in a reversible circuit [14]; however their approach used offline testing. Authors in [10] proposed all the variants of the missing gate fault model and also detection conditions for each type of fault. Hayes et al. proposed a DFT (design for testability) offline approach for detecting single missing gate faults [3]. In [4] the authors proposed an online testing approach for the detection of single missing gate faults. In this paper we propose an online testing approach to detect single bit faults, crosspoint faults and missing gate faults.

IV. ONLINE TESTING APPROACH:

The reversible gates must be converted into a duplicate gate blocks for making the reversible circuit to online testable equivalent.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

1. Conversion of Toffoli gate into a DGB:

A Duplicate Gate Block (DGB) consists of two gates. In order to convert a Toffoli gate to a Duplicate Gate Block we add an additional Toffoli gate as shown in Figure 4.

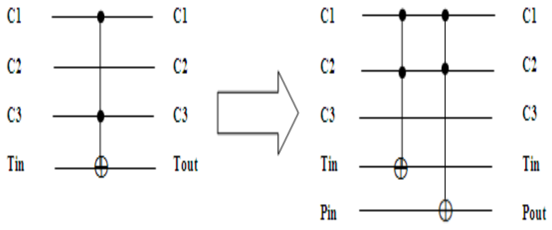


Fig. 4. Conversion of a Toffoli gate into a Duplicate Gate Block

The controls of the newly added gate (or duplicate gate) are on the same lines as that of the original gate. However, the target line of the duplicate gate is connected to the parity line.

2. Design:

To convert a reversible circuit to its online testable equivalent we first convert the k-CNOT gates of the circuit into Duplicate Gate Blocks. We also require the inclusion of a parity line P which is initialized with a logic 0. For each line in the circuit a 1-CNOT gate is inserted at the beginning and at the end of the original circuit. The targets of the additional CNOT gates are connected to the parity line. Given a reversible full adder circuit with L lines and N gates as shown in Figure 5, the first step to make it online testable is to add an extra line to the circuit. This line is the parity line, P, which is initialized with logic 0. We next convert each gate into its Duplicate Gate Block and the cascade the blocks in the same order that the gates appear in the original circuit. We now have a cascade of Duplicate Gate Blocks (DGB). The next step is to add 1-CNOT gates to each line at the input of the circuit. A total of L 1-CNOT gates are added.

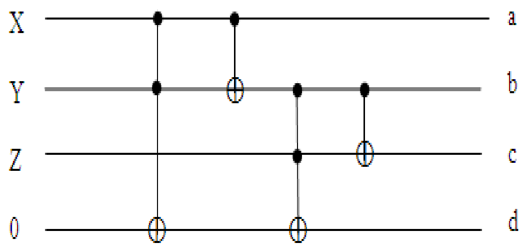


Fig. 5. A full adder reversible circuit

The target of each of these gates is connected to the parity line. We refer to this set of 1-CNOT gates as the Preamble Block. Similarly, we add another set of 1-CNOT gates which begins after the end of the cascaded DGB's, refer to this set of 1-CNOT gates as the Postamble Block. The entire circuit consists of three blocks in sequence: the Preamble Block, Duplicate Gate Block and Postamble Block.

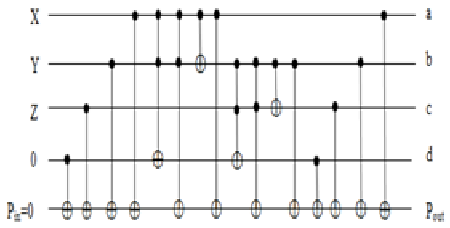


Fig. 6. Online testable equivalent full adder

3. Analysis:

A generalized diagram of an online testable reversible circuit as shown in Figure 7. Where Px and Qxy represent the parity line and the common lines for the corresponding level respectively. The target and the control lines are treated as common lines. We can determine the outputs at the Preamble Block as follows: Q11 = Q10, Q21 = Q20 ,. . . ,QL1 = QL0 and $P1 = P0 \oplus Q10 \oplus Q20 \oplus Q30,..., \oplus QL0$. The parity line is initialized to 0, thus P0 = 0 and $P1 = Q10 \oplus Q20 \oplus Q30,..., \oplus QL0$. From these equations we can say that the Preamble Block acts like a parity checker. That is, if the parity of the common lines at the input (level 0) is odd then after passing through the Preamble Block, the value on the parity line (P1) at level 1 will change to logic 1. If the parity of the common lines at the input (level 0) is even then the parity line (P1) at level 1 will remain logic 0. Also, the output values of the Preamble Block on the common lines will be equal to the input values. Thus the circuit will have a logic 1 at the parity line when the parity of the common lines of that level is odd. On the other hand, the parity bit will be at logic 0 if the parity of common lines of that level is even. We call this property the parity property.

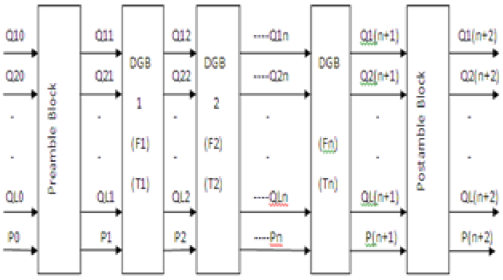


Fig. 7. Block Diagram of Online Testable Reversible Circuit

The output of the Preamble Block forms the input of the cascade of the Duplicate Gate Block. If there is no fault in the Preamble Block then the DGBs also follow the Parity Property. Let Fx be the output function of any Duplicate Gate Block. Let Tx and Px+1 be the two target lines of the original gate and the duplicate gate of a Duplicate Gate Block (DGB) respectively. The target line of the duplicate gate is always the parity line, whereas the target line of the original gate is one of the common lines. Then Tx is one of the lines amongst {Q1(x+1), Q2(x+1),...,QL(x+1)}. Let Tx = Qi(x+1) where i ∈ (1, 2, 3,...,L); then Tx = Fx⨁Qix and P(x+1) = Fx⨁Px. From the above two equations it is observed that if Fx is logic 1 then P(x+1) and Tx will toggle the input value (Px) and Qix respectively. If Fx is 0 then the output of the DGB will be equal to its input and no change will take place. The changes in Tx and P(x+1) take place simultaneously. In other words, the change in the parity of the common lines and P(x+1) take place simultaneously or they do not change. We refer to this property of the DGB as the Simultaneous Change Property.

The Simultaneous Change Property ensures that the Parity Property present at the input of the DGB remains consistent throughout the output of the circuit. Furthermore, if the input of the DGB violates the Parity Property then the violation is passed to the output of the DGB. The output of the cascade of the Duplicate Gate Block forms the input of the Postamble Block. If there is no fault in any of the previous blocks then the input of the Postamble Block will also satisfy the Parity Property. The output equations of the Postamble Block are: Q1(n+2) = Q1(n+1); Q2(n+2) = Q2(n+1) ,. . . ,QL(n+2) = QL(n+1). P(n+2) = P(n+1) ⊕ Q1(n+1) ⊕ Q2(n+1) ⊕ Q3(n+1),...,QL(n+1) From these equations it is seen that if the parity of the common lines is odd at level (n + 1) then the input parity of the Postamble Block, P(n+1) is logic 1. Hence the output parity P(n+2) will be logic 0.

V. DISCUSSION WITH FAULT MODELS:

We assume only one type of fault is presented at a time. Faults that have an effect on the output of the circuit will change the value of output parity bit from 0 to 1. A logic 1 at the output parity indicates that the operation of the circuit is faulty.

A. Missing Gate Fault Family:

In this section we observe the effect of missing gate fault. However we assume only one type of fault is presented at a time.

1. Single Missing Gate Fault:

Consider a random Duplicate Gate Block (DGBx) in the circuit. Suppose the original gate in this Duplicate Gate Block is missing. The missing gate is redundant if any of this gate’s control points are logic 0. Let us consider the situation when all the control points of the original gate are logic 1. As there is no fault in the Preamble Block, so the input of this DGB will follow the Parity Property. The output of the original gate is connected to Tx, so there will not be any change in the common lines. However, the output parity line will toggle its input bit.

This is because all the control lines are logic 1, so the duplicate gate in the DGB will toggle its target bit (the target of the duplicate gate is the parity line).

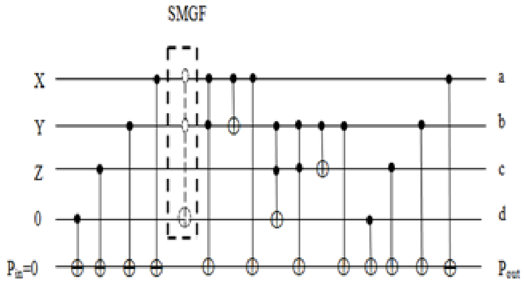


Fig. 8.SMGF in original gate

Thus the Parity Property would be violated at the output of this DGB. The parity will be forwarded to Postamble block. When the inputs of the Postamble Block do not follow the parity property then a logic 1 would be produced at the output parity line. A logic 1 at the P_{out} represents the presence of a fault. For example, consider the online testable circuit of a full adder as shown in Figure 8. When the input vector of (x, y, z, 0) is (1 1 0 0) and (1 1 1 0) then the output will be (1 0 0 0) and (1 0 1 0) instead of the correct output (1 0 0 1) and (1 0 1 1) respectively. Most importantly the parity output will be logic 1, which is the indication of faulty output.

Now consider the SMGF in the second gate i.e, duplicate gate of a Duplicate Gate Block as shown in Figure 9. In this case, the output of the common lines will change, because the output of the common lines depends on the original gate. However, as the target line of the faulty gate is connected to the parity line, the output parity line of the corresponding DGB will not be changed.For instance, when the control lines of the gates are at logic 1 then the target line, Tx of the original gate would toggle but the parity line would not toggle, which violates the Simultaneous Change Property.

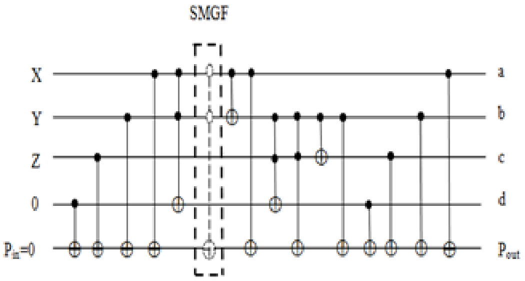


Fig. 9. SMGF in duplicate gate

This violation will also affect the Parity Property at the input of the Postamble Block. As a result the output parity of the circuit will be logic 1, which is a sufficient condition for the detection of a fault in the circuit.

2. Partial Missing Gate Fault:

If a partial missing gate fault occurs in the original gate as shown in Figure 10, then some of control points of the gate will be missing. For the fault to be detected at least one of the missing control points should be logic 0 and the rest of the control points of the faulty gate should be logic 1 [10]. Thus when the missing control point is logic 0 and all the non-missing control points are at logic 1 then the faulty gate would toggle the target line (Tx) of the DGB, which would give incorrect output and the parity of the common lines would be changed. However, all the control points are not at logic 1 for the duplicate gate in DGB. Therefore, the parity line (Px+1) of the DGB will not change. Hence, the Parity Property would be violated at the output of this DGB.

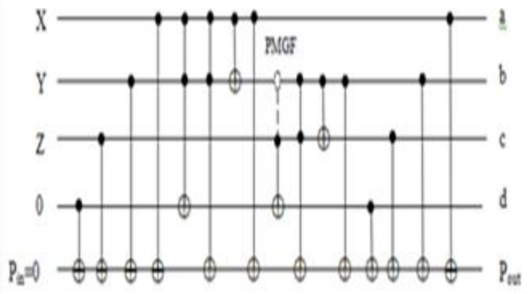


Fig.10. Partial Missing Gate Fault in original gate

According to the Simultaneous Change Property, this violation is transferred throughout the cascade of DGBs to the input of the Postamble Block. Faulty input at the Postamble Block produces logic 1 on the output parity line. Now consider the case where the control points of the duplicate gate are missing. When the non-missing control points are logic 1 and one of the missing control points is logic 0 then the parity line (where the target of the duplicate gate exists) would produce an output by toggling its input. However, the common lines simply pass the inputs of the original gate to the output. So the target line of the original gate would not toggle its input bit, which violates the parity property. As a consequence the output parity of the circuit would be logic 1, which identifies the fault.

3. Repeated Gate Fault:

As far as the Repeated Gate Fault is concerned, if the number of repetitions of a gate is odd then this fault does not affect the circuit output. However, if the number of repetitions is even then the effect of this fault is identical to that of a single missing gate fault i.e., the repeated gate is missed.

B. Cross point Fault:

If an appearance fault occurs in the original gate of the circuit then one or more extra control points are added to the gate as shown in figure 11. The fault is detectable if at least one of the extra control points has logic 0 while the other control points are logic 1.

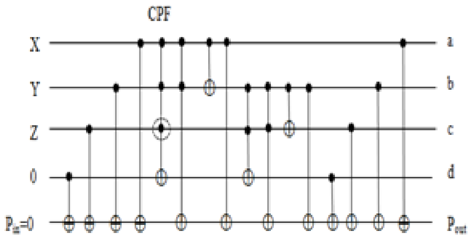


Fig.11. CPF fault occurs in the original gate

In this case the target line of the faulty gate will not toggle. However the target of the duplicate gate in the DGB will toggle its input bit. Therefore, a faulty output is generated and the output of DGB fails to satisfy the Parity Property.

VII. SIMULATION RESULTS:

The simulation results for SMGF in original gate are performed on Xilinx ISE 14.4 using Verilog HDL. The simulation results are shown below figures.

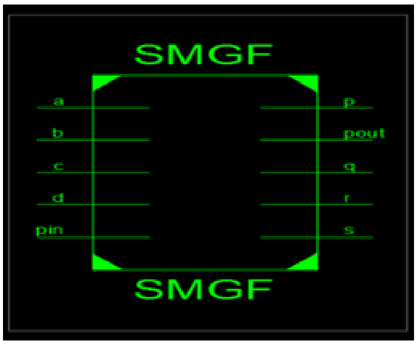
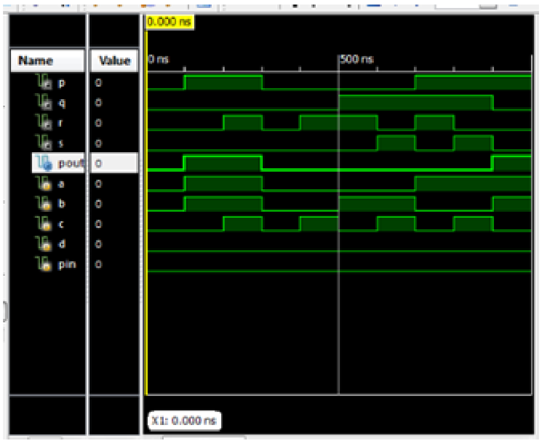


Figure-12. RTL schematic of Top-level of Single Missing Gate Fault



Name	Value
p	0
q	0
r	0
s	0
pout	0
a	0
b	0
c	0
d	0
pin	0

Figure -13. Simulation results of Single Missing Gate Fault in original gate.

VIII. CONCLUSION:

This paper presents an online testing approach for reversible circuits based on the CNT gate library. With the small modifications a circuit can be created such that computes its original functionality and in addition the circuit will detect missing gate faults and crosspoint faults. It can also detect a fault even if the fault occurs in the additional circuitry.

REFERENCES:

[1]C. H. Bennett. Logical reversibility of computation. IBM Journal of Research and Development, 6:525–532, 1973.

[2]M. P. Frank. Introduction to reversible computing: Motivation, progress, and challenges. In Proceedings of the 2nd Conference on Computing Frontiers, pages 385–390, Ischia, Italy, 2005. ACM Press.

[3]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, pages 100–105, Washington, DC, USA, 2004. IEEE Computer Society.

[4]D. K. Kole, H. Rahaman, D.K. Das, and B.B. Bhattacharya. Synthesis of online testable reversible circuit. In Proceedings of the IEEE 13th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), pages 277–280, 2010.

[5]R. Landauer. Irreversibility and heat generation in the computing process. IBM Journal of Research and Development, 5:183–191, 1961.

[6]Sk. N. Mahammad and K. Veezhinathan. Constructing online testable circuits using reversible logic. IEEE Transactions on Instrumentation and Measurement, 59(1):101–109, January 2010.

[7] D. Maslov. Reversible logic synthesis benchmarks page. 2011. http://webhome.cs.uvic.ca/ dmaslov/.

[8]N. M. Nayeem and J. E. Rice. Online fault detection in reversible logic. In Proceedings of the 26th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), pages 426–434, Vancouver, BC, Canada, October 2011.

[9]W. D. Pan and M. Nalasani. Reversible logic. Potentials, IEEE, 24(1):38–41, 2005.

[10]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), pages 422–427, 8–21 Dec., Calcutta, India, 2005.

[11]J. E. Rice. An overview of fault models and testing approached for reversible logic. In Proceedings of the 2013 Pacific Rim Conference of Communications, Computers and Signal Processings (PACRIM), pages 125–130, Victoria, Canada, Aug 2013.

[12]D. P. Vasudevan, P. K. Lala, J. Di, and J. P. Parkerson. Reversible-logic design with online testability. IEEE Transactions on Instrumentation and Measurement, 59(2):406–413, April 2006.

[13]R. Wille, D. Grope, L. Teuber, G. W. Dueck, and R. Drechsler. RevLib: An online resource for reversible functions and reversible circuits. In Int’l Symp. on Multi-Valued Logic, pages 220–225, 2008. RevLib is available at http://www.revlib.org.

[14]J. Zhong and J. C. Muzio. Analyzing fault models for reversible logic circuits. In IEEE Congress on Evolutionary Computation (CEC), pages 2422–2427, Vancouver, BC, Canada, 2006.