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

Design of Fault Models in Reversible circuits using Verilog HDL

Maragani Laxmi Prasanna
M.Tech(VLSISD),
Department of ECE,
Anurag Engineering College.

T.Sreedevi, M.Tech
Assistant Professor,
Department of ECE,
Anurag Engineering College.

M.Basha, M.Tech
Associate Professor & HOD,
Department of ECE,
Anurag Engineering College.

Abstract:

Programmable reversible logic design is trending as a prospective logic design style for implementation in recent nanotechnology and quantum computing with low impact on circuit heat generation. Reversible logic has emerged as a possible low cost alternative to conventional logic in terms of speed, power consumption and computing capability. 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 single bit fault, a crosspoint fault and the family of missing gate faults has been observed. Discussion around the correctness of our approach and the overhead is also provided.

I. INTRODUCTION:

The original motivation for the study of reversible circuits is the possibility of nearly energy-free computation. Landauer [14] showed that traditional irreversible circuits necessarily dissipate energy due to the erasure of information. It was later shown that, in principle, it was possible to perform reversible computation with arbitrarily small energy dissipation [2, 8]. Though the fraction of the power consumption in current VLSI circuits attributed to information loss is negligible, this is expected to change as increasing packing densities force the power consumption per gate operation to decrease, making reversible computation an attractive alternative. 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 three types of faults in reversible circuits. The organization of this paper is as follows: Section II presents the fundamentals of reversible logic and concepts of testing approaches and fault models; Section III describes some related work; Section IV introduces our proposed approach; detection of three types of faults using the proposed approach are presented in Section V; Section VI presents the shortcomings of our approach and Section VII concludes the paper and provides future directions.

II. REVERSIBLE LOGIC GATES

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. Generelly, we refer to the 0-CNOT gate as a NOT gate, to the 1-CNOT gate as a Feynman gate and to the 2-CNOT gate as a Toffoli gate. The traditional NOT gate is a reversible gate, since it is possible to restore the input of a NOT gate from its output. A NOT gate (0-CNOT) has no control line and hence the input at the target line is always inverted at the output line. However in a k-CNOT gate, there are k control inputs c1,...,ck and one target input, t.

The k-CNOT gate maps the vector (c1,...,ck, t) to the vector (c1,...,ck, t⊕c1c2 ...ck). This means the value at the target input is inverted if and only if all the values at the control inputs are 1 [9]. Reversible circuits are formed by cascading reversible gates. 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 faults described by three models: single bit faults [11], crosspoint faults [14] and missing gate fault [10]. A single bit fault is reflected on exactly one output of a gate, changing the correct value of the output to a faulty value because of the change in a bit on some line. 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 four 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 repeated gate fault (RGF): an unwanted replacement of a gate by the several instances of the same gate; (c) the multiple missing gate fault (MMGF): when several gates go missing from a circuit and (d) the partial missing gate fault (PMGF): some of the control points of a gate are missing. 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. In [8] the authors provided an improved online single bit fault testing approach. In this approach all the Toffoli gates of the circuit are changed to Extended Toffoli gates, and two sets of CNOT gates and one additional parity line are added to achieve online testability. This approach is effective as long as a single bit fault occurs in the original portion of the circuit. If a single bit fault occurs in the additional circuitry (any of the CNOT blocks) then the fault will go undetected. Zhong et al. proposed both the crosspoint fault model and as well a testing approach to detect single appearance and disappearance crosspoint 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:

A. 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.

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 1. 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. In the case of a 0-CNOT gate there is no control line, hence the Duplicate Gate Block would consist of two 0-CNOT gates: one on the same line as that of the original reversible gate and another on the parity line.

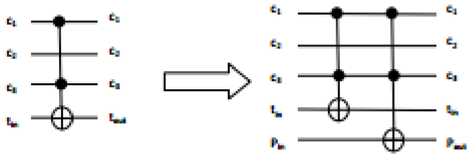


Fig. 1. Conversion of a Toffoli gate into a Duplicate Gate Block

Given a reversible circuit with L lines and N gates, 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 of the circuit into its Duplicate Gate Block and cascade the blocks in the same order that the gates appear in the original circuit. We now have a cascade of Duplicate Gate Blocks. 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. 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 Duplicate Gate Blocks. We refer to this set of 1-CNOT gates as the Postamble Block. Figure 2 illustrates the conversion. The entire circuit consists of three blocks in sequence: the Preamble Block, Duplicate Gate Block and Postamble Block. If the quantum cost of the original circuit is Q and the circuit has L lines then the quantum cost of the circuit's online testable equivalent will be 2L+2Q, since the quantum cost of a 1-CNOT gate is 1 [7].

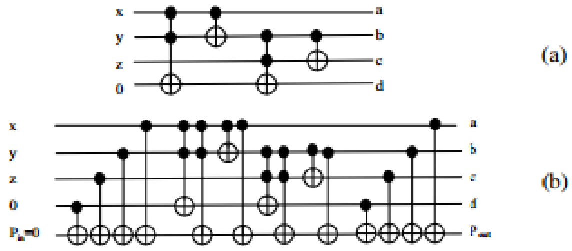


Fig. 2. (a) A full adder reversible circuit, and (b) its online testable equivalent

B. Analysis:

Figure 3 shows a generalized diagram of an online testable reversible circuit. 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 ⊕ Q10 ⊕ Q20 ⊕ Q30,..., ⊕QL0. The parity line is initialized to 0, thus P0 = 0 and P1 = Q10 ⊕ Q20 ⊕ Q30,..., ⊕QL0. From the above equation 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.

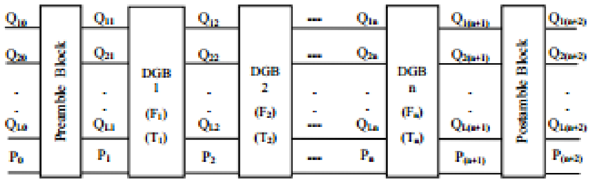


Fig. 3. Block Diagram of Online Testable Reversible Circuit

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. That is, if the parity of the common lines is even at level (n + 1) in Figure 3, then the input parity (Pn+1) would be logic 0 or vice versa. 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 the above equation 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.

On the other hand, if the parity of the common lines at level (n + 1) is even then the input parity P(n+1) is logic 0. Hence the output parity P(n+2) will be logic 0. In a nutshell, in a fault-free circuit operation the input of the Postamble Block will preserve the parity property and the final output parity P(n+2) of the circuit will be logic 0.

IV. FAULTS MODELS

In this section we consider scenarios for different types of faults in the three different blocks. 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 different types of missing gate faults in three different blocks in the circuit. Figure 2 shows the SMGF and PMGF. However we assume only one type of fault is presented at a time.

1) Single Missing Gate Fault and Repeated 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). Thus the Parity Property would be violated at the output of this DGB. According to the Simultaneous Change Property, this violation will be forwarded to the input of the 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. In this way, a faulty output which is generated due to a missing gate is notified by the parity line.

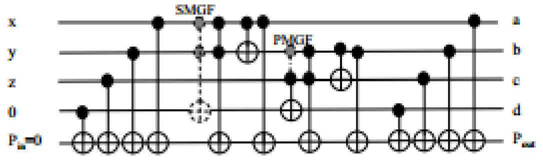


Fig. 4. Single Missing Gate Fault and Partial Missing Gate Fault

For example, consider the online testable circuit of a full adder as shown in Figure 4. Suppose the original gate which is indicated as a dotted line is missing. 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 (duplicate gate) of a Duplicate Gate Block. 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. 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.Now consider the case where one of the gates in the Preamble Block is missing. The parity lines of all the 1- CNOT gates appear in the parity line, so a fault in the Preamble Block will not affect the outputs of the common lines. But the fault will affect the parity output P1, which is given by P1 = Q10 ⊕ Q20 ⊕ Q30,...,QL0. When a fault affects the circuit output then the fault also changes the P1 to the opposite logic value of what we expect from the Parity Property. According to the Parity Property the output parity bit of the Preamble Block should be 0 if the parity of the common lines is even and the parity output should be 1 if the parity of the common lines is odd. But as one of the 1-CNOT gates is missing, so the fault will violate the parity property. This violation will affect the output parity line of the circuit and the high value at the output parity will indicate the faulty operation.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 [10]. Thus, similar faulty output would be generated for a repeated gate fault and the high output parity bit would indicate the presence of the fault in the circuit.

2) Partial Missing Gate Fault:

If a partial missing gate fault occurs in the original gate 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. 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. Thus, the output of the circuit would be erroneous, which is indicated by logic 1 on the output parity line For instance, assume that the control point as shown in Figure 4 is missing.

Now, when the input is (0 0 1 0) and (1 1 1 0) for an input vector (x, y, z, 0) then the output would be (0 0 1 1) and (1 0 1 0) instead of the fault-free value (0 0 1 0) and (1 0 1 1) respectively. For these two particular input vectors the value of the output parity would go high, which indicates that the operation is faulty. In this way a partial missing gate fault which appears in any of the original gates can be detected. 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. This is due to the fact that there is no PMGF in the original gate and all the input bits are not logic 0. So the target line of the original gate would not toggle its input bit. Thus, there is a change in the value of the parity line but parity of the common lines remains the same, which violates the simultaneous change property and also the parity property. As a consequence the output parity of the circuit would be logic 1, which identifies the fault. If a PMGF occurs in the Preamble Block then a 1-CNOT gate would become a 0-CNOT gate. A 0-CNOT gate will change every bit on its input. Thus the output parity bit of the Preamble Block will always be the opposite of the actual true value. As in previous cases this faulty parity will propagate to the successive blocks of the circuit and the effect will appear at the circuit output.

B. Crosspoint Fault:

The disappearance fault is identical to the partial missing gate fault, thus the effect of disappearance fault and its detection mechanism is the same as that of PMGF. If an appearance fault occurs in the original gate of the circuit then one or more extra control points are added to the gate. The fault is detectable if at least one of the extra control points have logic 0 while the other control points are logic 1. 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. When the appearance fault occurs in the duplicate gate of the DGB then the fault would be detectable if any of the extra added control points is at logic 0 and all other control points are at logic 1. In this case the original gate will toggle the output of the target line Tx. However the duplicate gate would not have all its control points at logic 1, so the parity line output would be the same as its input. Consequently, a change in the parity of common lines and no change in the value of the parity line would violate the Simultaneous Change Property and also the Parity Property would be lost. If an extra control point appears on a 1-CNOT gate of the Preamble Block then the fault will have an effect on the circuit output only when a new control point has a logic 0 and the old control point is at logic 1. Because of the presence of the fault the gate would not toggle. Thus we get the wrong parity at the output of the Preamble Block, which does not satisfy the Parity Property.

C. Single Bit Fault

In a single bit fault model, exactly one output of a circuit is faulty because of the change in a bit on some line. Thus if any single bit fault occurs in our model then the output of the common lines will not follow the Parity Property. Due to the nature of Simultaneous Change Property of the DGBs this violation will be propagated to the input of the Postamble Block. The input of the Postamble Block in turn would not follow the Parity Property and hence the parity output would be logic 1, which would indicate that there is a fault in the circuit. Consider a single bit fault occuring between the second and third DGB of the circuit in Figure 5. When the input vector (1 0 0 0 0) is applied to the circuit then the presence of the fault causes the value on line b to change from 1 to 0, which violates the Parity Property. The expected/correct output of the circuit should be (1 1 0 0 0); however the actual output, reflecting the fault, is (1 0 0 0 1). The violation of the Parity Property is carried through the circuit and the value of the output parity line becomes high, indicating the presence of the fault.

If there is no fault in the Preamble Block and the Duplicate Gate Block then the input of the Postamble Block will satisfy the Parity Property. The Postamble Block is the same in architecture as the Preamble Block. If any fault occurs in the Postamble Block then the effect of this fault will be the same as that of the Preamble Block. If any fault occurs in the Postamble Block the parity output will go high. A logic 1 at the output parity indicates the presence of the fault. To summarize, if there exists a fault with the original circuit then the output of the corresponding DGB cannot satisfy the Parity Property. Moreover, due to the Simultaneous Change Property of the DGBs, the violation of the Parity Property will be propagated to the input of the Postamble Block. When the input of the Postamble Block does not follow the Parity Property then the output of the Postamble Block produces logic 1 on the parity line, indicating that a fault exists. On the other hand, if an error occurs in any of the additional circuitry and affects the parity line then the parity line output will go high. The other outputs will not change. Thus if the parity line is high and the common line outputs are the same as expected then it indicates that the fault has occurred in the extra circuitry.

VI. COMPARISON AND LIMITATIONS

A. Comparison

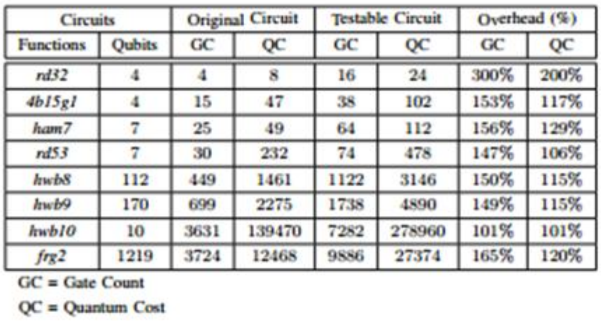
In this section we compare our proposed approach with two other online testing approaches. In [8], the authors proposed an online testing strategy for detection of single bit faults. They used two sets of CNOT gates and a single parity line to make a reversible circuit online testable. In their approach all the k-Toffoli gates (or k-CNOT gates) of the original circuit are changed to (k + 1)-Extended Toffoli Gates (ETG). We observed that using their approach to implement the full adder circuit presented in Figure 2 the resulting testable circuit has a gate count of 12 and a quantum cost of 28. Using our proposed approach (presented in Figure 2) the testable circuit has a gate count of 16 and quantum cost of 32. The quantum cost and the gate count of our approach are slightly higher as compared to their approach. However, the previous approach only considered single bit faults and our approach can detect three types of faults. We next compared our approach with the online testing approach presented in [4]. Their approach requires a single parity line and each k-CNOT gate of the original circuit will be transformed to its corresponding Augmented Reversible Gate (ARG). An ARG contains four gates: three additional gates and the original gate. Thus with their strategy, four gates are required to represent a single gate. Therefore, in order to implement the full adder circuit in Figure 2, their approach requires a testable circuit with a gate count 16, which is the same as that of our approach. The quantum cost of their testable circuit is 32, which is also the same as that of our approach. However, their approach was designed to detect only single missing gate faults.

Our approach is well suited for even a circuit with a large number of gates. In Table I we present the gate count and the quantum cost of the testable circuit after applying our approach to selected benchmark circuits [13] [7]. From this table we can see that for circuits with a larger number of gates our proposed approach actually results in a lower overhead (in terms of percentage of the original size). If we observe the first two benchmark circuits (with same number of inputs) from the table then we find that the circuit overhead is significantly lower for the circuit (4b15g1) with higher gate count. The reason behind this reduction is that the number of additional gates in the preamble and the postamble blocks does not depend on the number of gates of the original circuit, rather this number depends on the number of qubits (inputs) of the circuit. Circuits-rd32 and 4b15g1 have the same number of qubits, however 4b15g1 has almost four times more gates. Compared with [4], our approach also does better for circuits with higher gate count when considering quantum cost. For instance, if we add a single 1-CNOT gate to an original circuit presented in Figure 2 then for our approach the quantum cost increases by 2 (since the quantum cost of a 1-CNOT gate is 1, and we duplicate the gate).

However, using the approach in [4], the quantum cost would increase by 4. This is because as long as the number of qubits in a circuit does not increase, our approach only includes a duplicate gate for each original gate.

TABLE I OVERHEAD FOR SELECTED

BENCHMARK CIRCUITS



Circuits		Original Circuit		Testable Circuit		Overhead (%)	
Functions	Qubits	GC	QC	GC	QC	GC	QC
rd32	4	4	8	16	24	300%	200%
4b15g1	4	15	47	38	102	153%	117%
ham7	7	25	49	64	112	156%	129%
rd53	7	30	232	74	478	147%	106%
hwb8	112	449	1461	1122	3146	150%	115%
hwb9	170	699	2275	1738	4890	149%	115%
hwb10	10	3631	139470	7282	278960	101%	101%
frg2	1219	3724	12468	9886	27374	165%	120%

GC = Gate Count
QC = Quantum Cost

B. Limitations:

Firstly, the proposed online approach considers only the gates from the CNT gate library. So it is not currently applicable for other reversible gates. The proposed approach can not detect a single bit fault if the fault occurs in the Preamble Block. A single bit fault in the Preamble Block causes a reversible circuit to produce a faulty output. However in this case the output parity line will be logic 0. Thus by observing the output parity bit we can not detect the single bit fault.

In addition, our approach fails to detect a particular case when dealing with multiple missing gate fault. Multiple missing gate faults occur when several consecutive gates go missing in a circuit [10]. Suppose that N consecutive gates in the original gates are missing. Here N might be even or odd. For a different combination of inputs, different gates amongst the missing gates would be irredundant. If we consider a case where an even number of missing gates are irredundant, then the Parity Property would be violated for an even number of times. When the parity property is violated an even number of times then the fault is redundant. Thus in those cases when the number of missing gates is even then the output will be incorrect but the Parity Property will be preserved. So the parity output will be logic 0 which does not indicate the error even though the output is incorrect. Now consider an input combination when the irredundant missing gates are odd. The Parity Property is violated an odd number times. Hence the parity output value converts to high which properly indicates the error in the output. Therefore, some but not all the possible MMGF faults are detectable by the our proposed online model.

VII.SIMULATION RESULTS:

The corresponding simulation results of the floating point adders are shown below. All the synthesis and simulation results are performed using Verilog HDL. The synthesis and simulation are performed on Xilinx ISE 14.4. The simulation results are shown below figures.

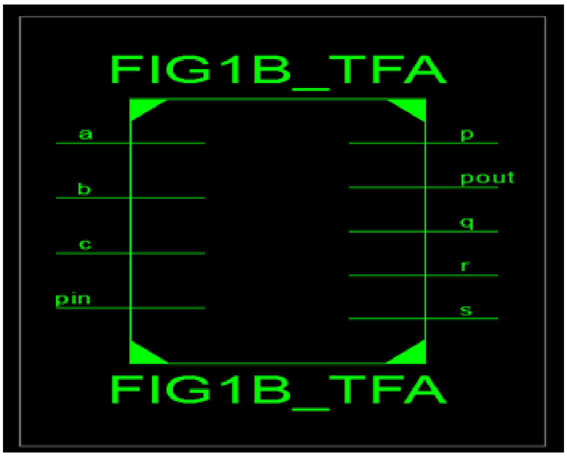


Figure-7: RTL schematic of full adder reversible circuit and its online testable equivalent

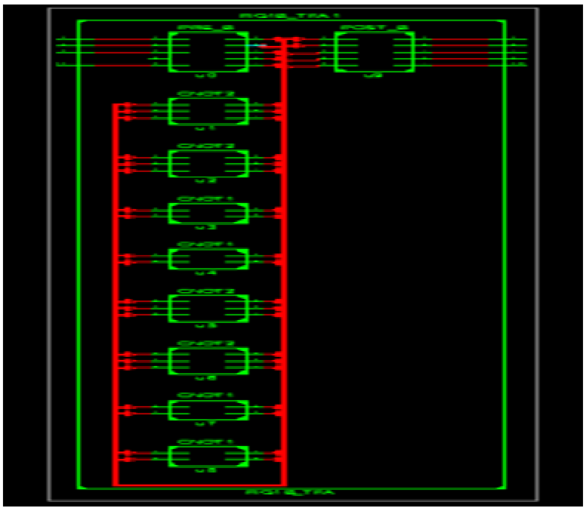


Figure 8: RTL schematic Internal block full adder reversible circuit and its online testable equivalent

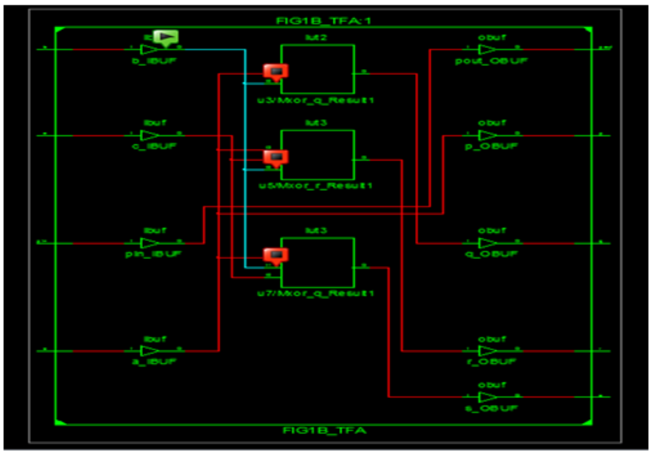


Figure 9: Technology schematic of full adder reversible circuit and its online testable equivalent

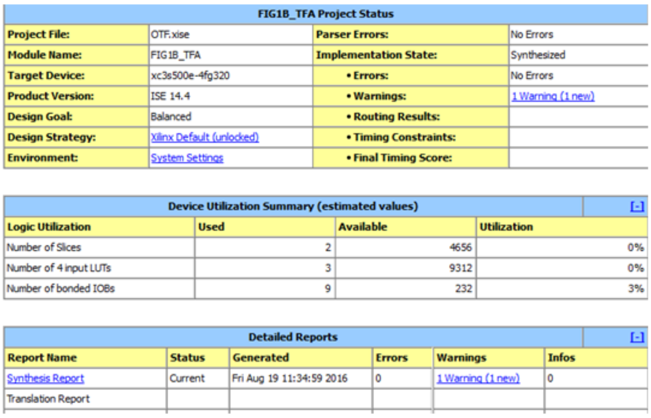


FIG1B_TFA Project Status			
Project File:	OTF.xise	**Parser Errors:**	No Errors
Module Name:	FIG1B_TFA	**Implementation State:**	Synthesized
Target Device:	xc3s500e-4fg320	**• Errors:**	No Errors
Product Version:	ISE 14.4	**• Warnings:**	1 Warning (1 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	2	4656	0%
Number of 4 input LUTs	3	9312	0%
Number of bonded IOBs	9	232	3%

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Fri Aug 19 11:34:59 2016	0	1 Warning (1 new)	0
Translation Report					

Figure 10: Synthesis report of full adder reversible circuit and its online testable equivalent

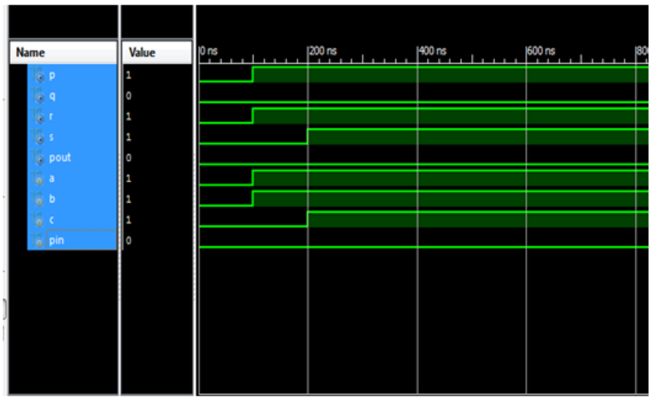


Figure 11: simulated outputs for full adder reversible circuit and its online testable equivalent

VII. CONCLUSION AND FUTURE WORKS:

We use the property of reversibility to simplify the testing problem for reversible circuits, and give conditions for an input set to fully test a reversible circuit under both the stuck-at and cell fault models. We develop some theoretical results on test set constructions. This paper presents an online testing approach for reversible circuits based on the CNT gate library. With this approach a reversible circuit can be converted to its online testable version by adding a set of CNOT gates and a single parity line in a well-defined manner. We considered different fault scenarios in a reversible circuit and observed the output. If a fault occurs in the original gate of a circuit then the output will be incorrect and the parity line will go high.

We also observe that if a fault occurs in any of the extra circuitry then the original output of the circuit will not be affected. However the parity line will go high which clearly indicates the presence of a fault in the circuit. Therefore, if the parity line is high and the output is same as expected then we can assume that the fault has occurred in the additional circuitry.Finally, though we have focused on testing of classical reversible circuits here, we hope extension of this approach to detect all the possibilities of single bit faults and multiple missing gate faults is the area of further research.

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. Große, 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.