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

Fault Detection in Parity Preserving Reversible Circuits

B.Mounika
M.Tech Student,
Siddhartha Institute of Technology and Science.

S.Rajitha, M.Tech
Assistant Professor,
Siddhartha Institute of Technology and Science.

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. Reversible logic computing is a rapidly developing research area. Testing such circuits is obviously an important issue. With the emergence of physical realizations, also the consideration of faults and faulttolerance became important. It has been suggested that parity preserving circuits would be ideal for fault detection, since here the parity of the inputs is the same as the parity of the outputs. Hence, if there is a fault on any single output, the parity should be flipped which would make the fault easy to detect. This paper however shows that this is not always the case. In fact, we provide and discuss examples showing that it is not sufficient to have parity preserving circuits when considering established fault models for reversible logic. As a result of our investigations, we can conclude that, even if a reversible circuit is parity preserving, it has to be checked against a particular fault model.Simulation and verification of this scheme is presented on Xilinx ISE 14.7.

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. Reversible computation enables several promising applications and, indeed, surpasses conventional computation paradigms in many domains including but not limited to quantum computation (see, e. g., [1]), certain aspects of low-power design (as experimentally observed, e. g., in [2]), the design of adiabatic circuits (see, e. g., [3]), encoding and decoding devices (see, e. g., [4]), or verification (see, e.g., [5]).

Accordingly, also the consideration of the design of reversible circuits received significant interest. In comparison to conventional circuit design, new concepts and paradigms have to be considered here. For example, fanout and feedback are not directly allowed. This affects the design of reversible circuits and requires alternative solutions. To this end, several approaches ranging from synthesis (see, e. g., [6], [7], [8], [9], [10]), optimization (see, e. g., [11], [12]), verification (see, e. g., [13], [14], [15]), and debugging (see, e. g., [16]) have been introduced. An overview of that is, e. g., provided in [17], [18].

In parallel, how to physically build reversible and quantum circuits is investigated and led to first promising results (see, e. g., [19], [20]). With this, also the question of how to prevent and detect faults in the physical realization became relevant. In particular for quantum computation, this is a crucial issue: Quantum systems are much more fault-prone than conventional

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

circuits, since the phenomenon of quantum decoherence forces the qubit states to decay – resulting in a loss of quantum information which, eventually, causes faults.

Besides the work on testing and test pattern generation (see, e. g., [21], [22], [23]), this also triggered the design of fault-detecting or fault-tolerant circuits – leading to the development of fault-tolerant libraries (see, e. g., [24], [25]) and corresponding design methods. The later includes the design of parity preserving reversible circuits which gained significant attention in the recent years and yielded several contributions such as [26], [27], [28], [29], [28], [30], [31], [32]. This development was motivated by the benefits of parity preservation in conventional circuits and aimed for adapting this to reversible circuits as well (this is discussed in more detail later in Section III).

However, no real investigation or discussion has been performed yet tackling the question whether parity preserving is indeed useful for reversible/quantum circuits. In this work, we are conducting such an investigation. To this end, we are considering two faults models that rank amongst the mostly considered models for reversible and quantum circuits: the single missing control fault and the single missing gate fault [22]. We explicitly apply previously proposed design methods for parity preserving reversible circuits and evaluate the resulting netlists with respect to faults from these fault models.

Our evaluations unveil that, although previously proposed techniques indeed guarantee parity preservation, this property is often not helpful with respect to the commonly assumed faults. In fact, parity preserving reversible/quantum circuits are frequently not capable of detecting or even tolerating missing control as well as missing gate faults. If they are, the respective method to make a circuit parity preserving yields a significant increase in the costs of the circuit. In this sense, our work motivates a re-evaluation of the need and usefulness of parityreversing reversible and quantum circuits.

The remainder of this work is structured as follows: The next section reviews the basics on reversible circuits and the fault models considered here, while, Section III provides an overview on recent work which has been published on parity preserving reversible circuits. Based on that, Section IV includes the evaluation and discussion on the resulting circuits with respect to fault detection and correction. Section V includes the synthesis and simulation on XilinxISE 14.7. The findings of the work are, eventually, summarized in the conclusions in Section VI.

PRELIMINARIES

To keep the remainder of the paper self-contained, this section introduces reversible circuits and provides a brief overview on the fault models considered in this work. A. Reversible Circuits A logic function $f:B_n \rightarrow B_m$ over inputs $X = \{x_1, ..., x_n\}$ is reversible iff (1) its number of inputs is equal to its number of outputs (i.e. $n = m$) and (2) it maps each input pattern to a unique output pattern. That is, reversible functions represent bijections. Reversible circuits are realizations of reversible functions. A reversible circuit G is a cascade of reversible gates gi, i.e. $G = g_1g_2...g_d$, where no fanout and feedback is allowed [1]. In this work, we consider the most widely used reversible gate, the Toffoli gate [33]

Definition 1. A Toffoli gate over the set of inputs $X = \{x_1, ...,x_n\}$ has the form g(C, t), where $C \subset X$ is the set of control lines and $t \in X \setminus C$ is the target line. In the following the target line has the index k, i. e., $t = x_k$. A single Toffoli gate g(C, t) realizes the bijective function

$$(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_{k-1}, x_k \oplus \bigwedge_{x_c \in C} x_c, x_{k+1}, ..., x_n)$$

That is, if all control line variables xc are assigned 1, the target line t is inverted. Under this assignment the gate is called activated. All other input values $x \in X \setminus \{t\}$ pass the gate unaltered. Note that the set of control lines may be empty. In this case, the gate works as a NOT gate, i. e., the target line is always inverted. Gates with a single control line are termed CNOT. When the term "Toffoli gate" is used in this paper, it is implied that it has two controls, unless otherwise noted.

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

Example 1. Fig. 1(a) shows a reversible circuit including three circuit lines and four Toffoli gates, i. e., n = 3 and d = 4. Control lines are denoted by a •, while the target line is denoted by ⊕. The annotated values demonstrate the computation of the respective gates for a certain input pattern. In this case, gates g_1 and g_3 are activated.

Note that reversible circuits usually provide a "blueprint" for quantum circuits. That is, in order to realize the logic design of a quantum circuit, the respective function is first synthesized as a reversible circuit and, afterwards, mapped to a quantum circuit structure (using methods such as [34], [35]). As the underlying fault models (covered in the next section) are similarly applied for both, reversible and quantum circuits, we omit an explicit introduction of quantum circuits and perform the discussion on reversible circuits only.

B. Fault Models

As in conventional circuits, faults may occur in reversible and quantum circuits (either caused by production or degradation). In order to abstract from the physical faults and, hence, allow for a logical consideration of the corresponding effects, usually discrete fault models are utilized. For reversible and quantum circuits, the single missing control fault and the single missing gate fault rank amongst the mostly considered models [22]. They are defined as follow:

Definition 2. Let g(C, t) be a Toffoli gate of a circuit G. Then,

1) a Single Missing Control Fault (SMCF) appears if instead of g'(C', t), a gate g (C , t) is executed, whereas C = C \ {xi} with xi ∈ C (i. e., a control line is removed).

2) a Single Missing Gate Fault (SMGF) appears if instead of g no gate is executed (i. e., g completely disappears). In order to detect a fault, the respective gates have to be activated so that the faulty behavior can be observed at the outputs of the circuit. Depending on the considered fault, this requires certain input assignments [22]. More precisely,

1) to detect an SMCF at gate g(C, t), all control lines in C (except the missing one) have to be assigned 1, while the missing control line xi ∈ C, respectively xi ∈/ C'' , has to be assigned 0. The assignment of the remaining lines can be arbitrarily chosen.

2) to detect an SMGF of gate g(C, t), i. e., the disappearance of g, all control lines in C have to be assigned 1, i. e., g simply has to be activated. The assignment of the remaining lines can be arbitrarily chosen.

Example 2. Fig. 1 illustrates an SMCF 1(b) and an SMGF 1(c), respectively, which can occur in the reversible circuit previously introduced in Fig. 1(a). The respective assignment needed to detect these faults are also given.

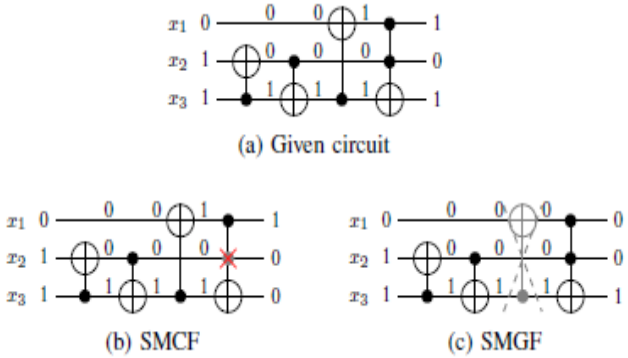


Fig. 1. Reversible circuits and possible faults

PARITY PRESERVING REVERSIBLE CIRCUITS

Parity bits have been used in electronic data transmission form the early days of computers, since it is easy to detect a fault in a single bit. Thornton [36] has shown that a parity bit can be efficiently embedded in adder circuits. Parhami [37] coined the term parity preserving in 2002. None of the papers considered the fault coverage of such a circuit. Furthermore, both papers consider conventional Boolean circuits that are not necessarily reversible.

Definition 3. A multiple-input multiple-output function f(X) is parity preserving if X = f(X), for all assignments of the input variables in X. In 2006, Parhami used the parity preserving property for reversible logic circuits [38]. He noted that any change in a single output will be detected.

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

Paul et al. [26] have shown that there are (2n−1!)2 reversible parity preserving function with n variables while the number of all reversible circuits is (2n)!. This means that less than one percent of all reversible circuits with four or more lines are parity preserving. But any reversible function can be made parity preserving by adding one more input and setting the corresponding output to preserve the parity of the input (a proof is also given in [26].) Furthermore, in [26] it is shown how parity preserving functions can be generated with exponential complexity. It is not known how the complexity increase of the function will affect the cost of the circuit.

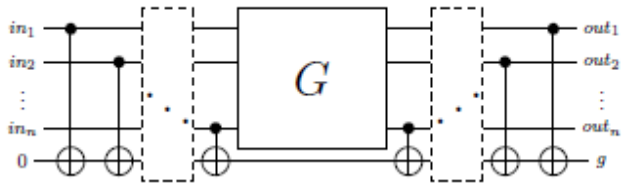


Fig. 2. A simple way to make any function/circuit parity preserving

Another way to embed a non parity preserving reversible circuit into an parity preserving reversible circuit is depicted in Fig. 2. By adding one line, with constant zero input, to the original circuit and calculating the parity before and after the original circuit on the added line, the new overall circuit will be parity preserving. Unfortunately, this circuit will always produce a parity preserving input-output relation independently of the original circuit C (assuming that all CNOTs gates before and after the original are fault-free). Hence, no fault in the original circuit C will be detected which does not motivate further studies.

Besides that, the following partial list of papers present other contributions towards reversible circuits aiming for parity preserving properties:

- In 2010, design of a full adder [29] using the building blocks from [28].
- In 2013, design of an ALU [30].
- In 2014, design a compressor [31].
- In 2015, design of a full adder [32].

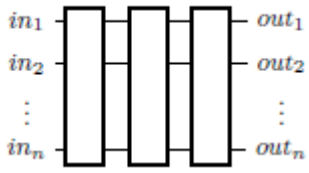
However, none of these papers include a study on the fault coverage of the presented design or a discussion of the additionally required costs for this purpose. Hence, it remains unknown whether the desired parity preserving characteristic of the proposed circuits is indeed helpful and efficient in order to detect faults. In this work, we will study this for the first time (to the best of our knowledge). To this end, a discussion about the usability and applicability of parity preserving design methods discussed above is provided next.

ANALYSIS OF ESTABLISHED FAULT MODELS

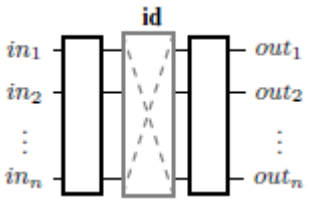
In the following, the parity preserving reversible circuits proposed in the past will be considered with respect to the two major fault models reviewed in Sect. II-B. As most currently presented approaches are using parity preserving elementary building blocks, the two fault models will be considered for such blocks at first in Sect. IV-A. As these results are not convincing, in Sect. IV-B realizations of the elementary building blocks using two straightforward methods will be investigated (again by applying the two fault models). Finally, Sect. IV-C deals with a quantum realization as presented in [29].

Fault Detection in Parity Preserving Building Blocks

For this subsection, we assume that a circuit has been realized by using parity preserving elementary building blocks only as depicted in Fig. 3(a).



(a) Circuit with parity preserving building blocks



(b) Circuit with a single missing gate fault

Fig. 3. Parity Preserving building blocks

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

Single Missing Gate Fault:

A missing gate fault means that the missing parity preserving gate/block behaves as an identity function and, thus, no change will be applied, see Fig. 3(b). Clearly, the identity function is parity preserving and consequently, the complete circuit will still be parity preserving because the composition of two parity preserving blocks is still parity preserving.

In conclusion, parity preserving elementary building block/gates do not help in detecting a fault according to the the missing gate fault model.

Single Missing Control Fault:

Let us now consider the missing control fault model. As the elementary building blocks/gates in other papers are presented as black boxes, we can not really apply a missing control fault anywhere. Hence, it remains unknown how to model a missing control fault within a building block. In fact, the realization of the block must be known before the missing control faults can be analyzed. One exception holds for [29], where the authors do not only describe the elementary parity preserving building blocks, but also present a quantum realization of an embedded Toffoli gate. A detailed analysis of this block will later be conducted in Sect. IV-C.

Parity Preserving Building Blocks

As shown in the last subsection, faults of the considered models will not be detected in circuits with parity preserving elementary building blocks. Thus, we have to analyze the quantum realizations of such building blocks. Unfortunately, of the standard gates, only the Fredkin gate is parity preserving. The NOT, CNOT and Toffoli gate are not parity preserving. However, an additional line is needed to embed any of the last-named gates into a parity preserving block. In nearly all of the papers presented in the last section, exactly one line with a constant zero input is added to implement these gates.

Two straightforward methods to make a Toffoli gate (independently from the number of control lines) parity preserving are shown in Fig. 4. Here, either

- a double gate as shown in Fig. 4(a) or
- two additional CNOT gates as shown in Fig. 4(b) are applied

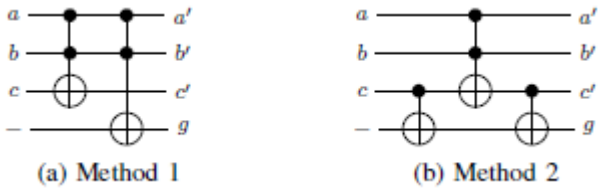


Fig. 4. Parity preserving Toffoli gate blocks

Note that for either method only one additional line is needed, because the value of the bottom line in the figure of both realizations is only used to preserve the parity.

Concerning the circuits cost, it should be mentioned that if the first method is used, twice as many gates are needed and the quantum cost doubles. While the second method will add twice as many CNOT gates as the number of used Toffoli gates. This also means, that the quantum costs will increase by two times the number of Toffoli gates, since a CNOT gates has quantum cost 1.

Furthermore, note that we will ignore faults that will cause a non-Boolean values at any control or output.

Double gate method:

Let us now assume that the double gate (see Fig. 4(a)) method has been used. If exactly one of the two Toffoli gates is missing, the building block will not be parity preserving. Thus, the whole circuit will also be not parity preserving and it is theoretically possible to detect this fault provided that we know a corresponding input/output pattern. Besides a missing gate, it is also possible that exactly one control is missing. Fortunately, in this case, we can also find an input/output pattern that breaks the parity preserving feature/property. For example, if the first control (on line a) of the first Toffoli gate is missing, the input (1,1,0,0) will be mapped to the output (1,1,0,1). For all 4 missing control cases, such a pattern can be found. Hence, this design detects all faults of the two considered fault models, but as stated above at the expense of signifi- cantly increased costs.

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

Add 2 CNOTs method:

Assuming that all buildings blocks are made of 2 CNOTs around the original Toffoli gate (see Fig. 4(b)), three different gates can be missing: one of the two CNOTs or the original gate. In the first case, i. e., one of the CNOTs is missing, the whole block will lose its parity preserving feature. In the latter case, i. e., the original gate is missing, the two remaining CNOTs together are realizing the identify function and, thus, the whole block will be still parity preserving. So, in 2 of 3 cases a missing gate would be detected. If the missing control fault model is applied, there are exactly two good cases: One of the CNOT controls is missing. Then the whole block will lose its parity preserving feature. On the other side, if one control of the main Toffoli gate is missing, the whole block will still be parity preserving, even if the overall circuit will not realize the desired function. If multiple-controlled Toffoli gates are considered, then this design provides very poor fault coverage. To summarize this, method 2 will obtain only 2 of |C| + 2 SMCF instances. It can be concluded, that the additional cost is low, but the fault coverage is not satisfactory.

Quantum Realization for a Parity Preserving Toffoli Gate

Instead of only using parity preserving building blocks, the authors of [29] have also proposed a quantum realization of the parity preserving Toffoli gate. We now investigate whether such a quantum realization can be used to detect faults. To this end, consider the quantum realization of a Toffoli gate as proposed in [29] and shown in Fig. 51. Both of the introduced fault models can be applied to the realization. First, the SMCF model is considered, afterwards the SMGF model.

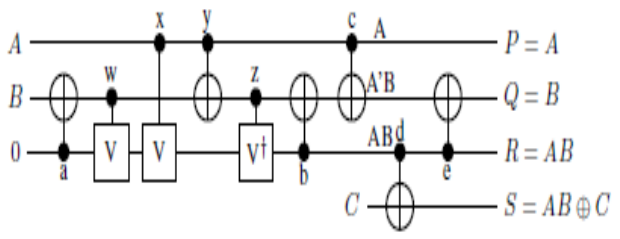


Fig. 5. Parity preserving Toffili gate from [29]

All possible SMCFs for CNOT gates that do not result in entanglement, i. e., do not trigger a control line with a nonBoolean value, are labeled a, b, c, d, e in Fig. 5. Those resulting in non-Boolean intermediate states are labeled w, x, y, z. Let us briefly consider the first set of faults:

a – Since the value of B will be always flip at the beginning, the parity is not preserved.

b – Since this fault will only flip B, it will not preserve the parity,

c – Similar to above.

d – Since only R is inverted, the parity is not preserved.

e – Since only Q is inverted, the parity is not preserved.

For all these SMCF instances, Tab. I gives one respective input/output pattern for the correct circuit as well as the output for the faulty circuit. This shows that the circuit is not parity preserving for any of the considered faults.

TABLE I POSSIBLE SMCFS WITHOUT ENTANGLEMENT PROBLEMS

	Input				Correct Output				Faulty Output			
SMCF	*A*	*B*	0	*C*	*P*	*Q*	*R*	*S*	*P*	*Q*	*R*	*S*
a	1	1	0	0	1	1	1	1	1	0	0	0
b	0	0	0	0	0	0	0	0	0	1	0	0
c	0	0	0	0	0	0	0	0	0	1	0	0
d	0	0	0	0	0	0	0	0	0	0	0	1
e	0	0	0	0	0	0	0	0	0	1	0	0

If one of the controls of w, x, y, z is missing, non-Boolean outputs can appear which require a measurement of the qubits. Unfortunately, these measurements are non-deterministic and we can not make any safe statement. To be more precise, if none of the controls of the gates are missing, a measurement it not necessary because all values will be pure Boolean. However, if any control is missing and the measurement states that one calculation is parity preserving, we can make any statement, but if the measurement shows us an input-output pattern which is not parity preserving, we can know for sure that the circuit is faulty.

Analyzing possible SMGFs has produced the results as shown in Tab. II. While a missing first gate does not

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

have any effect on the overall circuit, a missing gate at the position 6, 7, 8, or 9 would destroy the parity preservation. If one of the gates 2–5 is missing, again, non-Boolean values can be found at controls, thus, they are again omitted. As the tables show, using the quantum realization of Islam gives us a good fault detection rate by checking the parity preserving property. However, for each Toffoli gate one additional constant zero line has to be added which is impractical for larger circuits. Since the first CNOT gate of the quantum realization will never be activated because the control is on a constant zero input (unless the circuit has no faults), the quantum cost for the realization can be quantified with 8. Compared to the quantum cost of 5 of the normal Toffoli gate, the quantum cost of a final circuit would increased by a factor of 8/5 = 1.6.

TABLE II POSSIBLE SMGFS WITHOUT ENTANGLEMENT PROBLEMS

SMGF	Input A	B	0	C	Correct Output P	Q	R	S	Faulty Output P	Q	R	S
1	changes nothing, still parity preserving											
6	1	1	0	0	1	1	1	1	1	0	1	1
7	1	1	0	0	1	1	1	1	1	0	1	1
8	1	1	0	0	1	1	1	1	1	1	1	0
9	1	1	0	0	1	1	1	1	1	0	1	1

To the best of our knowledge, no method has been published that describes how, with only a fixed number of constant lines, more parity preserving Toffoli gates can be combined. In Fig. 6, we present such an embedding of a Toffoli gate. Unfortunately, all possible SMCF as well as SMGF would again cause non-Boolean signals, but without the usage of at least some V or V† gates, it would be impossible to realize the functionality.

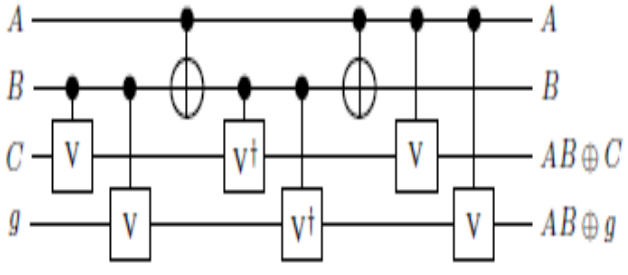


Fig. 6. Another parity preserving TG embedding

SIMULATION RESULTS

All the synthesis and simulation results of the Proposed Single Missing Gate Fault (SMGF) are performed using Verilog HDL. The synthesis and simulation are performed on Xilinx ISE 14.7. The corresponding simulation results of the Proposed Single Missing Gate Fault (SMGF) are shown below.

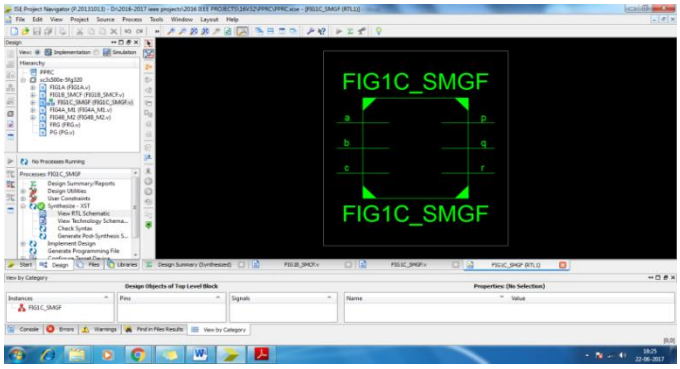


Fig.7: RTL schematic of Top-level of Proposed Single Missing Gate Fault (SMGF)

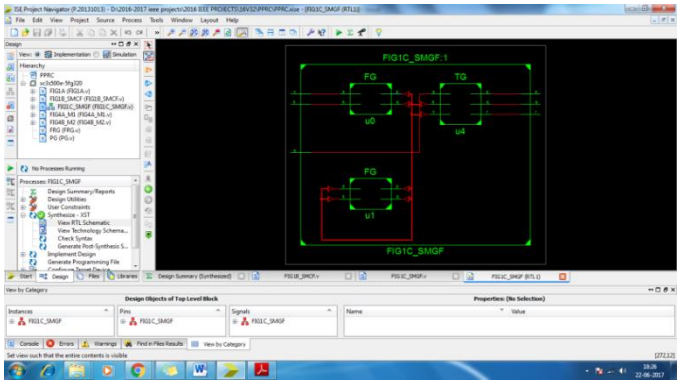


Fig.8: RTL schematic of Internal block of Proposed Single Missing Gate Fault (SMGF)

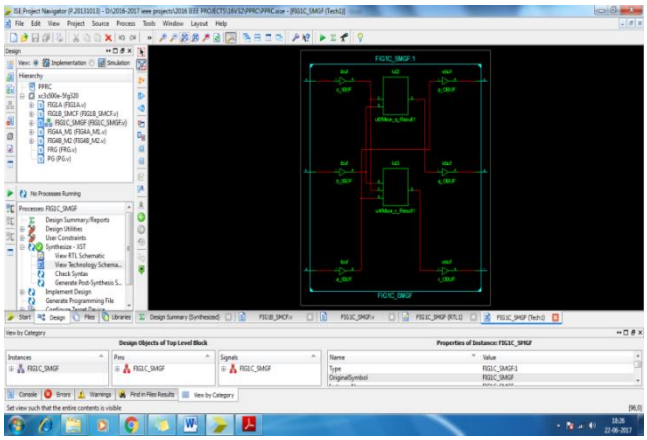


Fig.9: Technology schematic of Proposed Single Missing Gate Fault (SMGF)

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

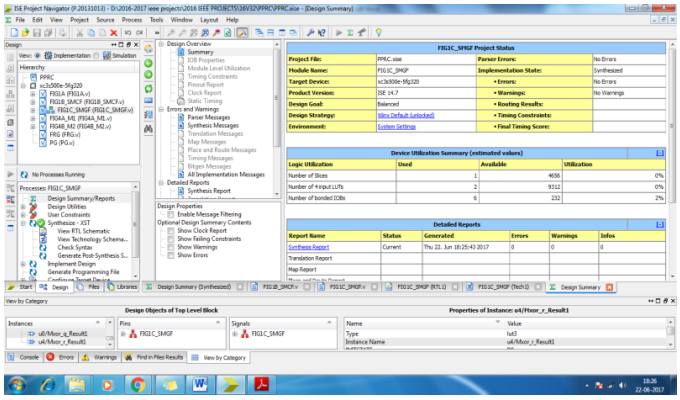


Fig.10: Synthesis report of Proposed Single Missing Gate Fault (SMGF)

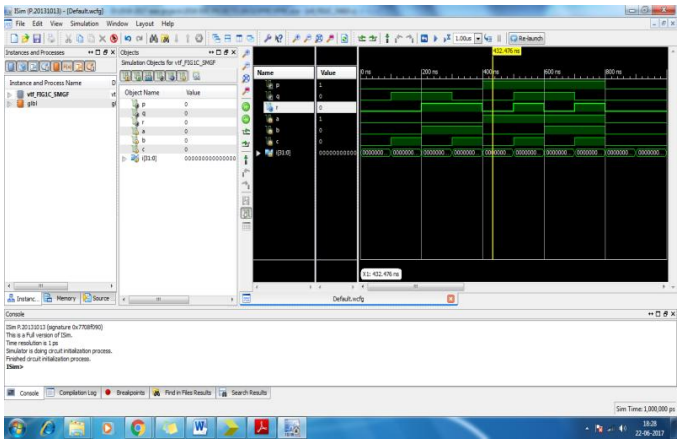


Fig.11: Simulated outputs for Proposed Single Missing Gate Fault (SMGF)

CONCLUSIONS

In this work, we considered the effectiveness of parity preserving reversible circuits with respect to established fault models. Our investigations show that parity preserving techniques are inadequate to cover single missing gate faults and single missing control faults. Deficiencies can be observed in terms of both, costs (i. e., they introduce large costs) as well fault coverage. Examples have been provided which show that barely half of the faults are covered. The lesson that we can learn from this investigation is that parity preserving circuits do no automatically ensure a reasonable fault coverage which always should be validated with a specific fault model. In the future, parity preserving might be beneficial when new (more realistic) faults models become available. But thus far, their benefit to reversible and quantum circuits is rather limited.

REFERENCES

[1] M. Nielsen and I. Chuang, Quantum Computation and Quantum Information.Cambridge Univ. Press, 2000.

[2] A. Berut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, and E. Lutz, “Experimental verification of landauer’s principle linking information and thermodynamics,” Nature, vol. 483, pp. 187–189, 2012.

[3] A. De Vos, Reversible Computing: Fundamentals, Quantum Computing and Applications. Wiley, 2010.

[4] R. Wille, R. Drechsler, C. Osewold, and A. G. Ortiz, “Automatic design of low-power encoders using reversible circuit synthesis,” in Design, Automation and Test in Europe, 2012, pp. 1036–1041.

[5] L. Amaru, P.-E.Gaillardon, R. Wille, and G. D. Micheli, “Exploiting inherent characteristics of reversible circuits for faster combinational equivalence checking,” in Design, Automation and Test in Europe, 2016.

[6] D. M. Miller, D. Maslov, and G. W. Dueck, “A transformation based algorithm for reversible logic synthesis,” in Design Automation Conf., 2003, pp. 318–323.

[7] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, “Synthesis of reversible logic circuits,” IEEE Trans. on CAD, vol. 22, no. 6, pp. 710–722, 2003.

[8] R. Wille and R. Drechsler, “BDD-based synthesis of reversible logic for large functions,” in Design Automation Conf., 2009, pp. 270–275.

[9] R. Wille, E. Schöborn, M. Soeken, and R. Drechsler, “SyReC: A hardware description language for the specification and synthesis of reversible circuits,” vol. 53, no. 3, pp. 39–53, 2016.

[10] M. Soeken, R. Wille, C. Hilken, N. Przigoda, and R. Drechsler, “Synthesis of reversible circuits with

minimal lines for large functions," in ASP Design Automation Conf., 2012, pp. 85–92.

[11] D. Y. Feinstein, M. A. Thornton, and D. M. Miller, "Partially redundant logic detection using symbolic equivalence checking in reversible and irreversible logic circuits," in Design, Automation and Test in Europe, 2008, pp. 1378–1381.

[12] D. M. Miller, R. Wille, and R. Drechsler, "Reducing reversible circuit cost by adding lines," in Int'l Symp. on Multi-Valued Logic, 2010, pp. 217–222.

[13] G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Checking equivalence of quantum circuits and states," in Int'l Conf. on CAD, 2007, pp. 69–74.

[14] S.-A. Wang, C.-Y.Lu, I.-M.Tsai, and S.-Y. Kuo, "An XQDD-based verification method for quantum circuits," IEICE Transactions, vol. 91- A, no. 2, pp. 584–594, 2008.

[15] J. Seiter, M. Soeken, R. Wille, and R. Drechsler, "Property checking of quantum circuits using quantum multiple-valued decision diagrams," in Reversible Computation 2012, ser. Lecture Notes in Computer Science, vol. 7581, 2012, pp. 183–196.

[16] R. Wille, D. Große, S. Frehse, G. W. Dueck, and R. Drechsler, "Debugging of Toffoli networks," in Design, Automation and Test in Europe, 2009, pp. 1284–1289.

[17] R. Drechsler and R. Wille, "From truth tables to programming languages: Progress in the design of reversible circuits," in Int'l Symp. on Multi-Valued Logic, 2011, pp. 78–85.

[18] M. Saeedi and I. L. Markov, "Synthesis and optimization of reversible circuits - a survey," ACM Computing Surveys, 2011.

[19] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," Nature, vol. 414, p. 883, 2001.

[20] B. Desoete and A. D. Vos, "A reversible carry-look-ahead adder using control gates," INTEGRATION, the VLSI Jour., vol. 33, no. 1-2, pp. 89–104, 2002.

[21] J. P. Hayes, I. Polian, and B. Becker, "Testing for missing-gate faults in reversible circuits," in Asian Test Symp., 2004, pp. 100–105.

[22] I. Polian, T. Fiehn, B. Becker, and J. P. Hayes, "A Family of Logical Fault Models for Reversible Circuits," in Asian Test Symposium. IEEE Computer Society, Dec. 2005, pp. 422–427.

[23] R. Wille, H. Zhang, and R. Drechsler, "ATPG for reversible circuits using simulation, Boolean satisfiability, and pseudo Boolean optimization," in IEEE Annual Symposium on VLSI, 2011.

[24] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 32, no. 6, pp. 818–830, 2013.

[25] P. Niemann, S. Basu, A. Chakrabarti, N. K. Jha, and R. Wille, "Synthesis of quantum circuits for dedicated physical machine descriptions," in Conference on Reversible Computation, July 2015.

[26] G. Paul, A. Chattopadhyay, and C. Chandak, "Designing parity preserving reversible circuits," CoRR, vol. abs/1308.0840, 2013. [Online]. Available: http://arxiv.org/abs/1308.0840

[27] M. Haghparast and K. Navi, "A novel fault tolerant reversible gate for nanotechnology based systems," American Journal of Applied Sciences, vol. 5, pp. 519–523, 2008.

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

[28] M. S. Islam, M. M. Rahman, Z. Begum, M. Z. Hafiz, and A. Al Mahmud, "Synthesis of fault tolerant reversible logic circuits," in Testing and Diagnosis, 2009. ICTD 2009. IEEE Conference on Circuits and Systems International, 2009, pp. 1–4.

[29] M. S. Islam, M. M. Rahman, Z. Begum, and M. Z. Hafiz, "Realization of a novel fault tolerant reversible full adder circuit in nanotechnology." Int. Arab J. Inf. Technol., vol. 7, no. 3, pp. 317–323, 2010.

[30] R. Saligram, S. S. Hegde, S. A. Kulkarni, H. Bhagyalakshmi, and M. Venkatesha, "Design of parity preserving logic based fault tolerant reversible arithmetic logic unit," arXiv preprint arXiv:1307.3690, 2013.

[31] S. Shoaei and M. Haghparast, "Novel designs of nanometric parity preserving reversible compressor," Quantum Information Processing, vol. 13, no. 8, pp. 1701–1714, 2014.

[32] M. Haghparast and S. Shoaei, "Design of a new parity preserving reversible full adder," Journal of Circuits, Systems and Computers, vol. 24, no. 01, 2015.

[33] T. Toffoli, "Reversible computing," in Automata, Languages and Programming, W. de Bakker and J. van Leeuwen, Eds. Springer, 1980, p. 632, technical Memo MIT/LCS/TM-151, MIT Lab. for Comput. Sci.

[34] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," Phys. Rev. A, vol. 52, no. 5, pp. 3457–3467, Nov. 1995.

[35] D. M. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-control toffolli gates," in Int'l Symp. onMultiValued Logic, May 2011, pp. 217–222.

[36] M. A. Thornton, "Signed binary addition circuitry with inherent even parity outputs," IEEE Trans. Comput., vol. 46, no. 7, pp. 811–816, July 1997.

[37] B. Parhami, "Parity-preserving transformations in computer arithmetic," Proc. SPIE, vol. 4791, pp. 403–411, 2002.

[38] ——, "Fault-tolerant reversible circuits," in Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on, 2006, pp. 1726–1729.