Achieving Fault Tolerance in Reversible Computing

Md Asif Nashiry, Jacqueline E. Rice

Abstract— Reversible circuits have drawn the attention of researchers for their many advantages over traditional irreversible circuits. However traditional circuit design techniques can not always be translated to reversible circuits, and so new technique e.g. fault tolerance must be developed. Since a reversible circuit maintains a one-to-one relationship between inputs and outputs, achieving fault tolerance in such a system is not an easy task. A fault tolerant system can correctly perform its specified operations even in the presence of faults. This paper investigate the existing work on the design of fault tolerant reversible circuits and presents an efficient approach for achieving fault tolerance in reversible circuit.

Index Terms— Reversible logic, fault tolerance, logic gates, fault testing, majority voter circuit.

1 Introduction

IN recent years, reversible computing has established itself as a promising research area and emerging technology. This is motivated by a widely supported prediction that conventional computer hardware technologies will reach their limits in the near future. Limitations of traditional computing, such as heat dissipation, can become an obstacle for the further development of current technology [1]. Reversible computing [2] offers a solution to this potential deadlock of further development in traditional computing. In 1961, Landauer showed that KTln2 joules are dissipated each time an information bit is lost during a logical operation, where K is Boltzmann's constant and T is the operating temperature in Kelvin [1]. For instance, when a two-input AND gate produces a single bit of output, this amount of energy is dissipated as heat. As Moore's law, predicting a doubling of components every few years, has held true over the last several decades, this heat dissipation is becoming a major concern in traditional irreversible systems. However reversible systems are, by definition, bijective, and thus there is no information loss. Other reasons for research into reversible systems includes connections to quantum computing, and applications in cryptography, nano-computing technologies and digital processing [2].

Since a reversible circuit maintains a one-to-one relationship between inputs and outputs, achieving fault tolerance in such a system is not an easy task. Fault tolerance is defined as an attribute of a system that enables the system to correctly perform its specified operations even in the presence of faults. In order to make a system fault tolerant, it is necessary to build in redundancy of some type, generally hardware redundancy, software redundancy, information redundancy and/or timing

Redundancy [3]. This added redundancy comes at a cost. A common approach to fault tolerance is to incorporate hardware redundancy by replicating one or more physical components of a system. The cost of this is initially high, but is amortized over the lifetime of the system. Hardware redundancy can offer an active approach, a passive approach or a combination of both [3]. An active approach to fault tolerance works by detecting a fault, locating the fault and recovering the system through some form of reconfiguration. However fault tolerance can also be achieved using a passive approach that does not require detecting or reconfiguring, rather masking the occurrence of faults. Masking contributes to a fault tolerant system by hiding faults from the final outcome of the system. Thus the fault may affect the system locally, but does not affect the global performance of a system. A majority voter is commonly used to achieve fault tolerance in traditional systems [3] by implementing triple modular redundancy (TMR) or N-modular redundancy. The basic idea of TMR is to triplicate the hardware and use a majority voter that determines the final output by observing the outputs from all the modules. If one of the modules become faulty, the majority voter can mask the faulty outcome by observing the outcomes of the remaining modules and correcting the faulty signal.

The rest of the paper is organized as follows: Section 2 provides the fundamentals of reversible computing; Section 3 describes some related works by focusing on the difference between testing and tolerance; fundamentals of achieving fault tolerance in reversible circuits are illustrated in Section 4; the existing majority voter circuits are introduced in Section 5; Section 6 describes our proposed majority voter circuit, and shows that the design of a fault tolerant reversible circuit using the proposed voter circuit; Section 7 draws the conclusion of this paper and provides future research direction.

2 Background

2.1 Reversible Logic

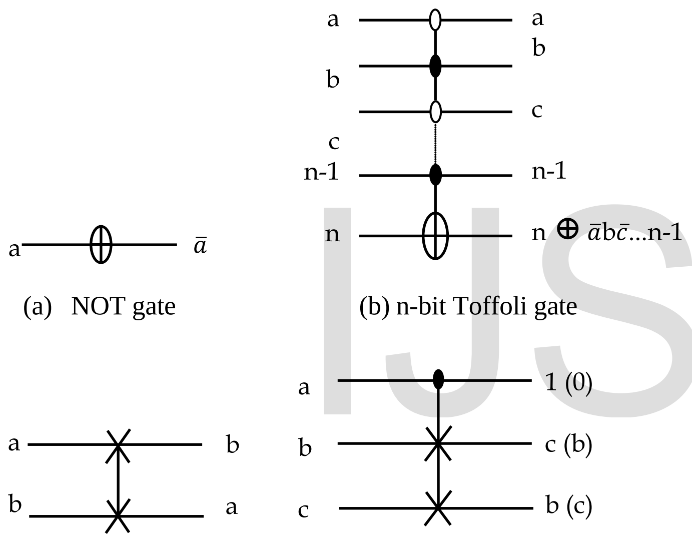
A reversible logic function has the form f:$B^n \rightarrow B^n$, where n is a non-negative integer and the domain B = {0,1}, with the key feature being that the function is bijective. More specifically, the number of inputs and the number of outputs of a reversible function are exactly the same. In particular, there is always a distinct output state for each of the possible input states [1, 2].

- *Md Asif Nashiry. Dept. of Mathematics and Computer Science, University of Lethbridge, Alberta, Canada. E-mail:asif.nashiry@uleth.ca*
- *Jacqueline E. Rice. Dept. of Mathematics and Computer Science, University of Lethbridge, Alberta, Canada. E-mail: j.rice@uleth.ca*

2.2 Reversible Logic Gates

Let $X \coloneqq \{x_1, x_2, \dots\dots, x_n\}$ be the set of Boolean variables. Then a reversible gate has the form $g(C, T)$, where $C = \{x_{i1}, \dots\dots, x_{ik}\} \in X$ is the set of control lines and T= $\{x_{j1}, \dots\dots, x_{jl}\} \in X$ with $C \cap T = \emptyset$ is the set of target lines [4]. Two commonly used reversible gates are Toffoli gates and Fredkin gates [4]. A Toffoli gate with no controls is a NOT gate i.e. $g(0, x_{j1})$.

Similarly, a Toffoli gate $g(x_{i1}, x_{j1})$ can be thought of as a controlled NOT (or CNOT) gate, and $g(\{x_{i1}, \dots\dots, x_{in}\}, x_{j1})$ is a n-bit Toffoli gate. A Fredkin gate with no controls is a SWAP gate $g(x_{j1}, x_{j2})$, which interchanges the two target input bits at output. A n-bit positive control Fredkin gate $g(\{x_{i1}, \dots\dots, x_{in}\}, x_{j1}, x_{j2})$ interchanges the two target bits at output when all the control inputs are equal to 1. A reversible gate may also have negative control. In this case the gate becomes active when negative control has a value of 0. Fig. 1 shows several commonly used reversible logic gates.



(a) NOT gate (b) n-bit Toffoli gate

(b) SWAP gate (d) 3-bit Fredkin gate

Figure 1: Some reversible logic gates

2.3 Cost Metrics

Two important metrics used to compare reversible circuit implementations are gate count and quantum cost. The gate count (GC) is the number of gates in a circuit and the quantum cost (QC) is the number of basic quantum gates required to implement macro-level reversible gates such as the Toffoli and Fredkin gates [12, 13]. For example, the QC of a CNOT gate is 1, and QC of a SWAP gate is 3. The QC of a (3 × 3) Toffoli and a (3 × 3) Fredkin gate is 5 [5].

3 RELATED WORKS: TESTING VS. TOLERANCE

Many works in the literature that address fault tolerant circuit designs in reversible logic use the term fault tolerant when really they are referring to testing. Works that fall into this category include [6 - 10]. In these works, the authors propose approaches which can fall into a category of testing. For example, Parhami [9] highlights the fact that most arithmetic and other processing functions do not preserve the parity of the input data at the output end. If the parity of input data is maintained throughout the computation, no intermediate error checking mechanism is required to detect faults in a circuit. The paper identifies some parity preserving reversible logic gates and presents a fault detection method based on parity preserving reversible gates. For example, Fredkin gates and double Feynman gates are parity preserving gates. A reversible circuit which is composed of only these two gates can detect an occurrence of fault as described by Parhami [9]. The paper also presents a design for a parity preserving full adder reversible circuit. The author concludes that fault tolerance can be achieved without adding any extra design effort if a reversible circuit is built using only parity reserving logic gates. Based on this concept, Islam et al. [7] proposed an approach to design what they claimed to be fault tolerant reversible circuits. The authors proposed a 4 x 4 parity preserving reversible gate, or IG gate, as shown in Fig. 2. They offer an implementation of a reversible full adder circuit with two IG gates and claim that their proposed design is fault tolerant, suggesting again that fault tolerance can be achieved without any extra design effort if a reversible circuit is built using parity preserving gates. Their proposed full adder circuit is presented in Fig. 3.

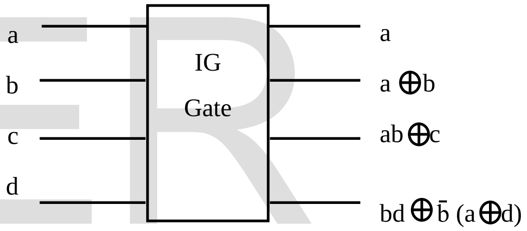


Figure 2: IG gate presented in [7].

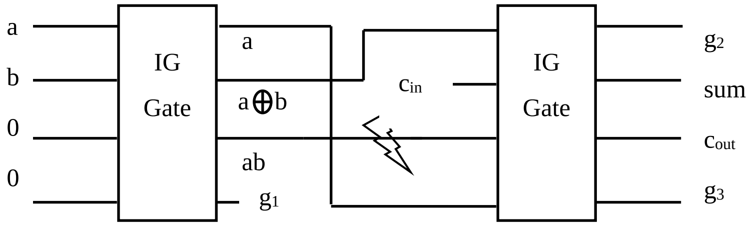


Figure 3: Parity Preserving Full Adder Circuit Composed of IG gates.

Since an IG gate is a parity preserving gate, the proposed full adder circuit also preserves the parity. The sum and the output carry of the full adder in [7] are defined as sum = (a ⊕ b ⊕ c_{in}) and c_{out} = (((a ⊕ b) c_{in} ⊕ ab). For fault free operation, when a = 1, b = 0 and c_{in} = 1, the sum and the output carry are calculated as sum = 1 ⊕ 0 ⊕ 1 = 0 and c_{out} = (1 ⊕ 0) x 1 ⊕ 1 x 0 = 1 ⊕ 0 = 1. Suppose a single bit fault occurs in the third line from the top of the circuit, as shown in Fig. 3. The third output line of the first IG gate, which generates ab, is connected to the third input line of the second IG gate. As an effect of a single bit fault, as shown in Fig. 3, the value of ab is inverted before contributing to the second IG gate. For example, when the full adder performs the addition of the three input bits, a = 1, b = 0 and c_{in} = 1; the third line from the top generates ab = 0. Due to the presence of the fault, the value of ab will be 1. This incorrect value of ab will be an input of the second IG gate, as shown in Figure 5.2. In this case the output carry is calculated as $cout = (a \oplus b)\ c_{in} \oplus ab = (1 \oplus 0) \times 1 \oplus 1 = 1 \oplus 1 = 0$, which is incorrect. As we see, the proposed full adder circuit in [7] has no ability to generate corrected output in the presence of a fault. Thus, this full adder cannot be called a fault tolerant full adder.

Similarly Haghparast el al. [6], the authors present a parity preserving Toffoli gate. According to this approach, a 3-bit Fredkin gate and a double Feynman gate are used to design a parity preserving Toffoli gate. Since a Feynman gate is already a parity preserving gate, the authors use two of their proposed parity preserving Toffoli gates and two double-Feynman gates to design what they claim is a fault tolerant full adder circuit. However, similar to the previous example, the full adder circuit presented in [6] has no ability to generated corrected output in the presence of faults in the circuit, and so is not fault tolerant.

As discussed above, these works focus on preserving parity in reversible circuits. After ensuring parity preservation in their designs, the authors of these works claim that their proposed approaches are fault tolerant. Parity preservation can be used as a testing approach, which may indicate whether the output of a circuit is correct or incorrect. However, a fault tolerant circuit must have the capability to supply corrected (intended) values at the output even in the presence of faults in the circuit. A parity preserving circuit does not guarantee that the circuit is fault tolerant, since the use of parity preservation offers only error detection. Parity preservation does not ensure that the circuit generates corrected output in the event of error. Thus the designs based on parity preservation cannot be categorized as offering fault tolerance. Other works that indeed fall into the category of fault tolerance are described later in this paper.

4 Fault Tolerance in Reversible Circuits

Fault tolerance is generally achieved through the addition of redundancy. Redundancy to achieve fault tolerance can take several forms, generally hardware redundancy, software redundancy, information redundancy and/or timing redundancy [3]. Redundancy is an additional resource in form of logical or physical components of a system, which helps to a system to achieve fault tolerance. The behavior of a system is an important factor to consider when choosing a particular type of redundancy to make the system fault tolerant. For example, the choice of redundancy that can be added to an information transfer system may not be the same as that of an information generation system. A common approach to achieve fault tolerance in an information generating system is to incorporate hardware redundancy by replicating one or more physical components of a system. In our work we focus on fault masking to achieve fault tolerance in reversible logic. The reasons for choosing fault masking as a tool for designing a fault tolerant reversible logic circuit is the simplicity and the dynamic behaviour of fault masking techniques. Fault masking allows a system to generate the correct output in real time. As a passive approach, a fault masking technique does not need to reconfigure or replace any physical component of a system. A system which is designed using a fault masking technique hides faults from the output, i.e. a fault cannot affect the final outcome of a system. Therefore, a fault may affect the system locally, but cannot affect the global performance of the system. Triple modular redundancy (TMR) has been used as one of the most common forms of passive hardware redundancy to design a fault tolerant system [3, 11, 12]. The basic idea of TMR is to use three exact copies of a module (system), and a majority voter is used to combine the outputs from the triple modules. The majority voter is designed in such a way that it always generates the bit which appears on majority of the input lines. Therefore, if one of the modules generates an incorrect output, the majority voter will still generate the correct output by masking the output of the faulty module. Sometimes more than three modules are used to form n-modular redundancy, where n replicas of a module are connected to a majority voter.

5 Existing Reversible Majority Voter Circuits

There are only a few attempts in the literature to design majority voter circuits in reversible logic. In [13], the authors proposed two designs based on triple modular redundancy. In their proposed design, a voter circuit takes three inputs from the output of three copied circuits and generates three data outputs. In order to maintain reversibility, the voter used in their design is a 5-bit circuit with two garbage lines and two constant input lines. However, our analysis suggests that this circuit does not generate the intended output.

A simplified majority voter circuit, as shown in Fig. 4, is presented in [11]. The authors describe a reversible fault tolerant multiplexing scheme using a 3-bit repetition code. The voter consists of two CNOT gates and one Toffoli gate, and the majority output value is taken from the line labeled 'a_o'. The majority voter has a GC of 3 and a QC of 1+1+5 = 7. The majority voter proposed in [11] can be used in designing fault tolerant reversible circuits by following the principle of triple modular redundancy.

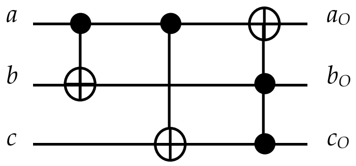


Figure 4: A reversible majority voter as proposed in [11]

6 A Proposed Reversible Majority Voter

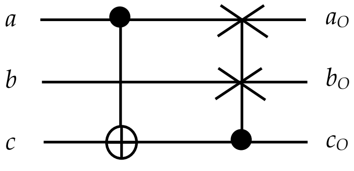


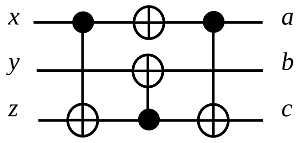
Figure 5: A three input reversible majority voter.

Table 1: Truth table for the reversible voter circuit.

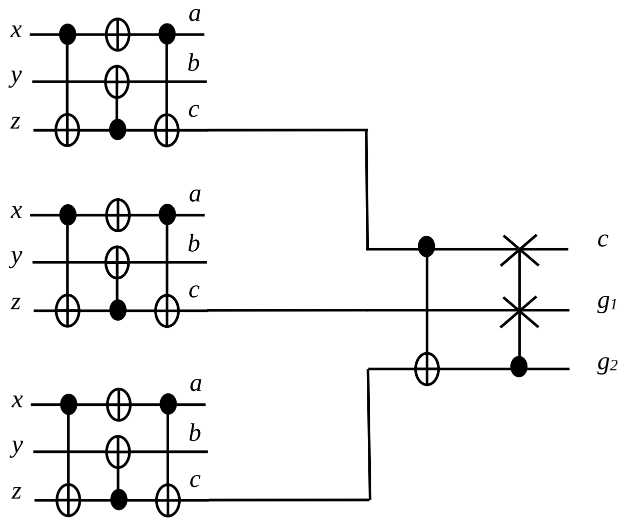
	After 1st gate	After 2nd gate
abc	a/b/c/	$a_0b_0c_0$
000	000	**0**00
001	001	**0**01
010	010	**0**10
011	011	**1**01
100	101	**0**11
101	100	**1**00
110	111	**1**11
111	110	**1**10

In this section we present a simplified and cost effective approach for designing a majority voter in reversible logic [14]. To introduce our approach, a reversible three input majority voter is shown in Fig. 5. The behaviour of this circuit is characterized by the truth table in Table 1. The output of interest is ‘a_0’ which always gives the value appearing on the majority of the input lines. The other two lines are not used, and thus are considered garbage. The QC of this circuit is very low: 1 for the CNOT gate and 5 for the Fredkin gate, for a total of 6, which offers a small improvement as compared to the approach presented in [11]. As we see from Figure 5.8, the control of the Fredkin gate is (a ⊕ c). When a = c then (a ⊕ c) = 0, and the Fredkin gate becomes inactive and does not interchange the values of ‘a’ and ‘b’ at the output. In this case ‘a’ is the value used as the majority output. However, when a ≠ c then (a ⊕ c) = 1, and the Fredkin gate becomes active and swaps the value of ‘a’ and ‘b’. In this case the value of ‘b’ will be the majority output.

Our proposed majority voter circuit can be used to achieve fault tolerance in any reversible circuit. For example, consider a (3 x 3) reversible circuit shown in Fig. 6(a). Suppose the output line ‘c’ is the output of interest for this circuit and the other two outputs are garbage. Fig. 6(b) shows a fault tolerant design for the circuit from Fig. 6(a). As we see from Fig. 6(b), three copies of the circuit are used, since we are using the principle of TMR. The three output lines from the three copied modules are connected to our proposed majority voter. If any one of the three copied modules becomes faulty, the majority voter masks the fault and sends the correct output to ‘c’. The other two output lines ‘g1’ and ‘g2’ are the garbage output lines of the majority voter.



(a) A reversible circuit.



(b) Fault tolerant version of the circuit presented in Figure 5.9(a).

Figure 6: Achieving fault tolerance using TMR.

6.1 Fault Tolerant Reversible Full Adder Circuit

A fault tolerant full adder design based on our proposed majority voter is shown in Fig. 7. Since we have two outputs (sum and carry) of interest in a full adder, we need two majority voters to ensure that faults in either output can be masked. As we see from Fig. 7 there are three carry lines that are connected to the voter on top, while the bottom voter is connected to the three the sum lines. This design can generate corrected output in the presence of faults from any of the proposed fault models [15] as long as the fault affects at most one of the triplicated full adders. Similarly, this design can be applied to any type of circuit, albeit with a significant amount of hardware overhead. We have also proposed some designs to extend the majority voter which are presented in [14].

7 Conclusion and Future Direction

This chapter presents a 3-bit reversible majority voter circuit. The purpose of this circuit is to identify the bit value which appears more than any other bit value on the three input bits, assuming the use of Boolean (binary) values. Our proposed design is simpler and of lower cost in terms of gate count and quantum cost than existing designs in the literature. We also provide the designs for extending the voter from 3-bit up to 6-bits. Moreover, we demonstrate the application of our voter in fault tolerant reversible circuit design.

We provide an overview and analysis of existing works that term themselves to be fault tolerant but which do not meet the required characteristics to be categorized as such. Lastly, we present a design of a fault tolerant reversible full adder circuit. The proposed design can be used to make any reversible circuit fault tolerant. The proposed majority voter can be used to generate a corrected output in the presence of any type of fault as long as the fault affects a minority of the n input lines to the voter.

One of the crucial threats in designing a fault tolerant digital system using n-modular redundancy is the single point failure. Thus, making the majority voter robust is one of the main areas of further study. In addition, future work includes developing techniques for fault location as well as fault correction in reversible circuits.

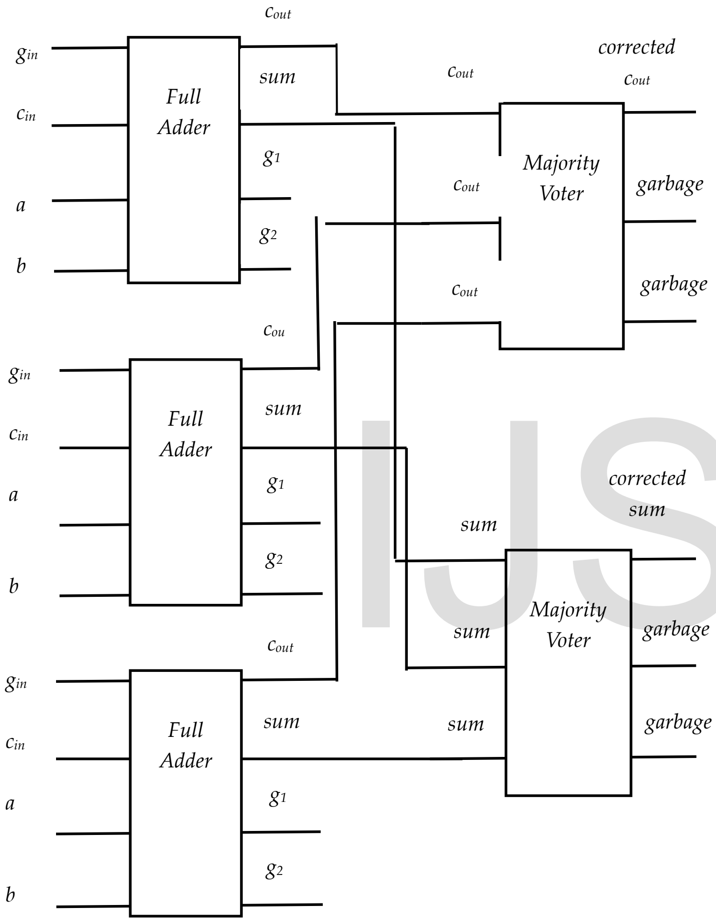


Fig. 7. A fault tolerant reversible full-adder circuit design.

REFERENCES

[1] Michael P Frank. “Approaching the physical limits of computing”. In Proceedings of 35th International Symposium on Multiple- Valued Logic, IEEE, pp. 168–185, 2005.

[2] Michael P Frank. “Introduction to reversible computing: motivation, progress, and challenges”. In Proceedings of the 2nd Conference on Computing Frontiers, ACM, pp. 385-390, 2005.

[3] B. W. Johnson, Design & Analysis of Fault Tolerant Digital Systems. Addison-Wesley Longman Publishing Co., Inc., 1988.

[4] T. Toffoli. “Reversible computing”, MIT, Tech. Rep. MIT/LCS/TM No. 151, 1980.

[5] Md Asif Nashiry, Mozammel H. A. Khan and Jacueline E. Rice. “Controlled and uncontrolled SWAP gates in reversible logic synthesis”, International Conference on Reversible Computation pp. 141-147, 2017.

[6] M. Haghparast and K. Navi. Design of a novel fault tolerant reversible full adder for nanotechnology based systems. World Applied Sciences Journal, 3(1):114–118, 2008.

[7] Md Saiful Islam, Muhammad Mahbubur Rahman, Zerina Begum, Mohd Zulfiquar Hafiz, and Abdullah Al Mahmud. Synthesis of fault tolerant reversible logic circuits. In Testing and Diagnosis, 2009. ICTD 2009. IEEE Circuits and Systems International Conference on, pages 1–4. IEEE, 2009.

[8] Sajib Kumar Mitra and Ahsan Raja Chowdhury. Minimum cost fault tolerant adder circuits in reversible logic synthesis. In VLSI Design (VLSID), 2012 25th International Conference on, pages 334–339. IEEE, 2012.

[9] Behrooz Parhami. Fault-tolerant reversible circuits. In Proceedings of the Fortieth Asilomar Conference on Signals, Systems and Computers (ACSSC), pages 1726–1729, Oct. 29–Nov. 1 2006.

[10] Bibhash Sen, Siddhant Ganeriwal, and Biplab K Sikdar. Reversible logic-based fault tolerant nanocircuits in QCA. ISRN Electronics, 2013.

[11] P Oscar Boykin and Vwani P Roychowdhury. Reversible fault-tolerant logic. In Dependable Systems and Networks, Proceedings. International Conference on, pp. 444-453. IEEE, 2005.

[12] Victor P. Nelson. Fault-tolerant computing: Fundamental concepts. Computer, 23(7):19–25, 1990.

[13] Masoud Zamani, Navid Farazmand, and Mehdi B. Tahoori. Fault masking and diagnosis in reversible circuits. In Proceedings of the 16th IEEE European Test Symposium (ETS), pages 69–74, May 2011.

[14] Md Asif Nashiry and Jacqueline E Rice. A reversible majority voter circuit and applications. In Communications, Computers and Signal Processing (PACRIM), 2017 IEEE Pacific Rim Conference on. IEEE, 2017.

[15] Ilia Polian, Thomas Fiehn, Bernd Becker, and John P Hayes. A family of logical fault models for reversible circuits. In Test Symposium, 2005. Proceedings. 14th Asian, pages 422–427. IEEE, 2005.